[MS-SMDL-Diff]:

Semantic Model Definition Language File Format

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
6/4/2010	0.1	Major	First release.
9/3/2010	0.1.1	Editorial	Changed language and formatting in the technical content.
2/9/2011	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/7/2011	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
11/3/2011	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
1/19/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
2/23/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
3/27/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
5/24/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
6/29/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
3/26/2013	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
6/11/2013	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
12/5/2013	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2014	1.0	Major	Updated and revised the technical content.
5/20/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
5/10/2016	1.0	None	No changes to the meaning, language, or formatting of the technical content.
8/16/2017	2.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
10/16/2019	3.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Semantic Model Definition Language (SMDL) file format is the file format for SQL Server SMDL, a file type that is used to represent the metadata for defining a semantic model.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

anchored expression: An expression or subexpression that is meaningful only in a specific entity context. An expression that is not an anchored expression is a floating expression. EntityRef elements are anchored in the entity to which they refer. AttributeRef elements are anchored in the entity that contains the attribute to which they refer if the attribute does not have an Attribute.IsAggregate element whose value is true. Function elements are anchored if any of their arguments are anchored, unless the argument takes a set of values and the function returns a scalar value. All other expressions or subexpressions, including Literal elements, Null elements, ParameterRef elements, and AttributeRef elements that refer to aggregate attributes are not anchored. An Expression.Path element that starts with a RolePathItem element anchors an otherwise floating expression. The RolePathItem element is anchored in the entity that contains the role that is referenced by the RolePathItem.RoleID child element.

attribute: A characteristic of some object or entity, typically encoded as a name/value pair.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

big-endian: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

common language runtime (CLR): The core runtime engine in the Microsoft .NET Framework for executing applications. The common language runtime supplies managed code with services such as cross-language integration, code access security, object lifetime management, and debugging and profiling support.

context entity: The context entity of an expression is a specific entity whose context is used to evaluate the expression. For example, within the expression, attribute references can only be to attributes that are contained by the context entity or contained by any entity in the same inheritance hierarchy as the context entity.

contextual name: A context-sensitive name for an attribute when the entity that contains the attribute is reached via a role.

culture: A part of a language identification tagging system, as described in [RFC1766]. Culture names adhere to the format "<languagecode2>-<country/regioncode2>."

data source: A physical data source.

data source view (DSV): A logical representation of database objects from a given data source specified by a schema.

drillthrough: A means in a client application to view a more detailed view of a subset of the data after clicking on a displayed instance of data. The displayed instance of data contains an action

with a drillthrough link, and clicking on the link executes a semantic query that returns more detailed results.

entity: A single business object about which data can be stored. It is the subject of a table in a relational database.

entity folder: A folder that can contain entities and further entity folders.

field: An attribute or role of an entity.

field folder: A folder that can contain fields and further field folders.

filter: A set of criteria that controls the set of records that is returned as a result set.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

inheritance: An inheritance relation describes how one entity inherits from another. The entity that inherits from another entity is known as the descendent entity, and the other entity is known as the ancestor entity. A direct descendent is called a child entity and a direct ancestor is called a parent entity.

inheritance hierarchy: An inheritance relation gives rise to a natural ordering among entity elements known as an inheritance hierarchy. An inheritance hierarchy is a logical tree structure that organizes a collection of entities such that each entity has zero or one parent entity and zero or more child entities. An entity that does not have a parent entity is a root node in the inheritance hierarchy, and an entity that does not have a child entity is a leaf node in the inheritance hierarchy.

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

MIME type: A method that is used by protocol clients to associate files of a certain type with applications that can open or access files of that type.

model: A semantic model.

model item: An element that defines an element of the semantic model. The model items are entities, entity folders, fields, and field folders.

non-anchored expression: An expression that is not an anchored expression.

path: An ordered list of roles to follow to reach a specific entity.

perspective: A subset of the semantic model to provide to users as a constrained submodel.

physical data source: A collection of information usually stored in a relational database from which data can be queried.

physical model: The physical model contains a description of the physical database with optional transformations to more closely correspond to the desired semantic model. An abstract physical model is represented by a concrete data source view (DSV) in an SMDL model.

primary key: A field or set of fields that uniquely identifies each record in a table. A primary key cannot contain a null value.

query: A semantic query, as defined in [MS-SMDL].

relation: A relation or relationship is an association between a pair of tables in a relational database, where one or more columns in one table is associated with one or more columns, which are typically primary keys, in the other table.

report definition language (RDL): A file type that is used to represent the metadata for defining a report.

role: An association between a pair of entities. One entity is the source or origin of the role and the other entity is the target or destination of the role. The role specifically describes how the target entity is related to the source entity. For each role there is a naturally occurring corresponding role called the related role, where the source entity of the related role is the target entity of the role and the target entity of the related role is the source entity of the role. The association repeats for each subsequent pair of entities, so that the target of one role becomes the source of the next role. In this way, entities in an information model may be associated through chains of roles that extend from one entity to the next throughout the information model.

semantic model: A user-friendly metadata description of a data source. A semantic model is an additional layer of information that maps database tables and views into concepts that are meaningful to business users. A semantic model is a collection of entities, their attributes, and their relationships (roles) that reflects the real-world relationships between business functions and processes. An example of this is how a product's entity relates to inventory and sales entities. Entities contain a set of attributes describing the entity, such as product ID, product name, and other product-related items, in addition to a set of roles that describe the relationship of an entity to another entity in the model.

semantic query: A query that is expressed using expressions and items from a semantic model.

static function: A function that is evaluated by the semantic modeling engine rather than being translated into the target query language. As a result, static functions are evaluated only at the start of query execution rather than being evaluated for each row in the query.

table: A two-dimensional object in a relational database that stores data in rows and columns.

variation: An attribute or role that is considered to be a variation of a field. For example, the attributes Order Year, Order Month, and Order Day are considered variations of the Order Date attribute.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[IEEE754] IEEE, "IEEE Standard for Binary Floating-Point Arithmetic", IEEE 754-1985, October 1985, http://ieeexplore.ieee.org/servlet/opac?punumber=2355

[ISO3166-1] ISO, "ISO 3166-1 decoding table", ISO 3166-1, http://www.iso.org/iso/iso-3166-1_decoding_table

[ISO639-2] ISO, "Codes for the representation of names of languages -- Part 2: Alpha-3 code", ISO 639-2:1998, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=4767

Note There is a charge to download the specification.

[ISO8601] ISO, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO 8601:2004, December 2004, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874

Note There is a charge to download the specification.

[MS-RSWSRE2005] Microsoft Corporation, "Report Server Web Service for Report Execution: ReportExecution2005".

[MS-RSWSRMNM2005] Microsoft Corporation, "Report Server Web Service for Report Management for Native Mode: ReportService2005".

[MS-RSWSRMSM2006] Microsoft Corporation, "Report Server Web Service for Report Management for SharePoint Mode: ReportService2006".

[MS-SSAS] Microsoft Corporation, "SQL Server Analysis Services Protocol".

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://www.rfc-editor.org/rfc/rfc2045.txt

[RFC2046] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996, http://www.rfc-editor.org/rfc/rfc2046.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC3548] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003, http://www.rfc-editor.org/rfc/rfc3548.txt

[RFC4646] Phillips, A., and Davis, M., Eds., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, http://www.rfc-editor.org/rfc/rfc4646.txt

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[XML10/5] Bray, T., Paoli, J., Sperberg-McQueen, C.M., et al., Eds., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, http://www.w3.org/TR/2008/REC-xml-20081126/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 (Updated Section) Informative References

[MSDN-BRM] Microsoft Corporation, "BinaryReader Class - Methods", ~~http://msdn~~https://docs.microsoft.com/en-us/~~library~~dotnet/api/system.io.binaryreader~~_~~#methods~~.aspx~~

[MSDN-BWM] Microsoft Corporation, "BinaryWriter Class - Methods", http://msdnhttps://docs.microsoft.com/en-us/librarydotnet/api/system.io.binarywriter_#methods.aspx

[MSDN-DTM] Microsoft Corporation, "DateTime Struct - Methods", http://msdnhttps://docs.microsoft.com/en-us/librarydotnet/api/system.datetime_#methods.aspx

[MSDN-DTOUTCP] Microsoft Corporation, "DateTimeOffset.UtcDateTime Property", http://msdnhttps://docs.microsoft.com/en-us/librarydotnet/api/system.datetimeoffset.utcdatetime.aspx

[MSDN-FDOW] Microsoft Corporation, "DateTimeFormatInfo.FirstDayOfWeek Property", http://msdnhttps://docs.microsoft.com/en-us/librarydotnet/api/system.globalization.datetimeformatinfo.firstdayofweek(v=vs.110).aspx

[MSDN-FTYPES] Microsoft Corporation, "Formatting Types in the .NET Framework", http://msdn", https://docs.microsoft.com/en-us/library/26etazsy.aspxdotnet/standard/base-types/formatting-types

[MSDN-GTBAM] Microsoft Corporation, "Guid.ToByteArray Method ()", http://msdn", https://docs.microsoft.com/en-us/librarydotnet/api/system.guid.tobytearray.aspx

[MSDN-TSFTM] Microsoft Corporation, "TimeSpan.FromTicks(Int64) Method (Int64)", http://msdn", https://docs.microsoft.com/en-us/librarydotnet/api/system.timespan.fromticks.aspx

[MSDN-TSTP] Microsoft Corporation, "TimeSpan.Ticks Property", http://msdnhttps://docs.microsoft.com/en-us/librarydotnet/api/system.timespan.ticks.aspx

[MSFT-RDL200501] Microsoft Corporation, "Report Definition Language (RDL) 2005", http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefinition/

[MSFT-RDL200801] Microsoft Corporation, "Report Definition Language (RDL) 2008", http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition/

[MSFT-RDL201001] Microsoft Corporation, "Report Definition Language (RDL) 2010", http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition

[MSFT-SMDL200410] Microsoft Corporation, "Semantic Modeling Schema", http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling/

1.3 Overview

This document specifies the file format for Semantic Model Definition Language (SMDL), a file type that is used to represent the metadata for defining a semantic model. A semantic model definition file is an XML file, as specified in [XML10/5].

1.3.1 Semantic Model Definition Language File Content

The Semantic Model Definition Language consists primarily of the following three kinds of information:

- **Semantic model**: This contains the user's model of the data, which is described in terms of objects with which the user would be familiar.
- **Physical model**: This contains a description of the physical database, with optional transformations to more closely correspond to the desired semantic model. Each physical model is a data source view (DSV).
- **Mapping**: A mapping describes how the semantic model is represented in a physical model, binding semantic objects to their corresponding physical objects.

An SMDL model contains exactly one semantic model, one physical model, and one mapping.

1.3.2 Byte Ordering

Some computer architectures order bytes in a binary word from left to right, which is referred to as big-endian. This documentation uses big-endian bit diagrams. Other architectures order the bytes in a binary word from right to left, which is referred to as little-endian. The underlying structures and fields in this document are little-endian.

Using big-endian and little-endian methods, the number 0x12345678 would be ordered as shown in the following table.

Byte order	Byte 0	Byte 1	Byte 2	Byte 3
Big-endian	0x12	0x34	0x56	0x78
Little-endian	0x78	0x56	0x34	0x12

Unless otherwise specified, all data specified in this document are ordered in little-endian format.

1.3.3 Document Structure

The root element of an SMDL document is the SemanticModel element.

Subelements are order-independent. That is, subelements can appear in any order. This includes items in collection elements unless the collection is explicitly defined as an ordered list.

Unless otherwise specified, a subelement can occur at most one time as a child of its parent element. A collection can have multiple instances of a subelement and this will be explicitly stated.

White space is not trimmed from values in the SMDL document.

All identifiers in SMDL are case-sensitive and match using the invariant culture.

1.3.3.1 XML Namespace

The namespace URI for SMDL is:

http://schemas.microsoft.com/sqlserver/yyyy/mm/semanticmodeling

The date component (yyyy/mm) indicates the date of the release of that version of SMDL. The date component is used throughout this document to identify one of the following specific versions of SMDL:

- 2004/10 [MSFT-SMDL200410]

The standard file name extension for SMDL files is .smdl.

The MIME type to use for SMDL files is text/xml.

1.3.4 Semantic Model Definition Overview Diagrams

This section contains diagrams that illustrate the schema of the 2004/10 (section 5) version of Semantic Model Definition Language.

The first diagram that follows illustrates the overall component model composed of a semantic model with semantic objects and bindings, a physical model and a data source. The diagrams that follow the

following diagram illustrate the semantic objects, the bindings between the semantic objects and the physical model, the expression objects, and the query objects.

Note that, for simplicity, certain related types are represented in the diagrams as abstract base types. For example, the abstract base type **Field** does not appear in SMDL. Only concrete derived types, such as Attribute and Role, appear in SMDL.

The following abstract base types are shown in the diagrams but do not appear in SMDL: **EntityFolderItem**, **ExpressionNode**, **Field** and **FieldFolderItem**.

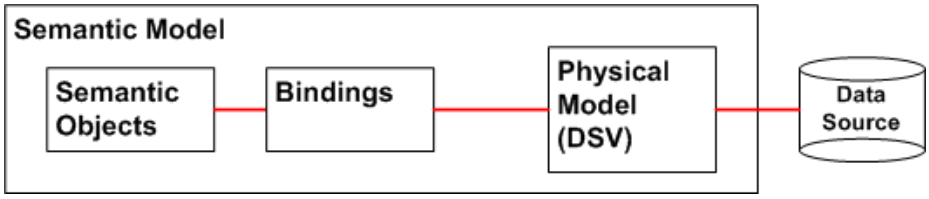


Figure 1: Model Components

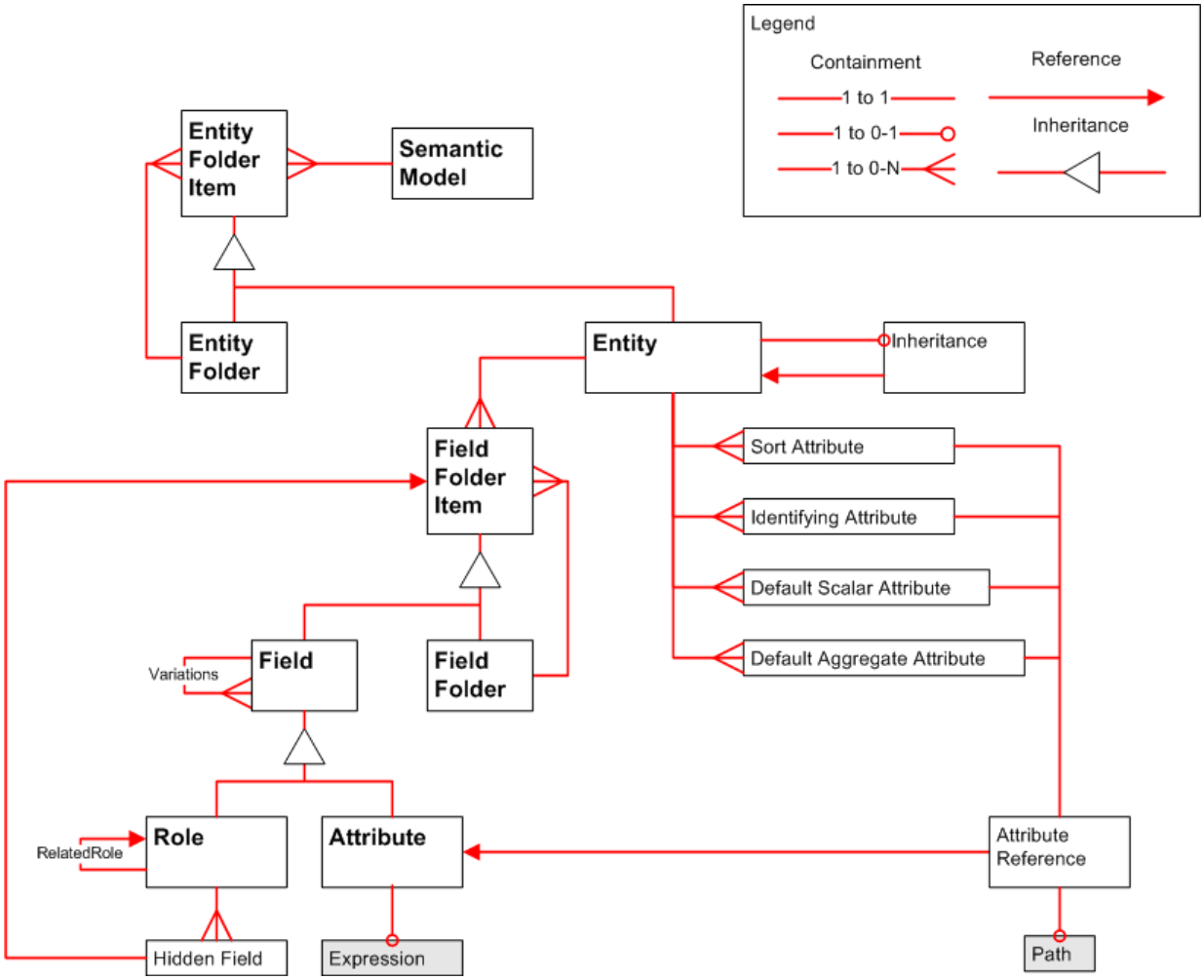


Figure 2: Semantic Objects

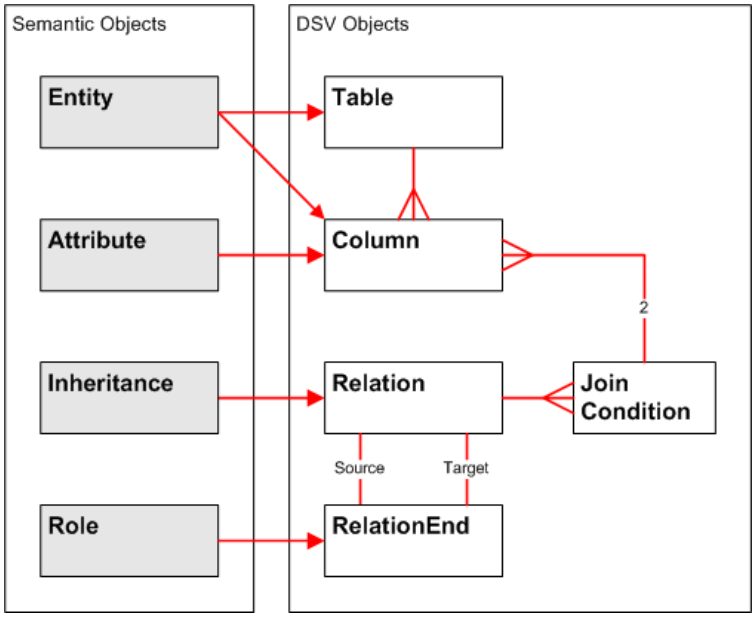


Figure 3: Bindings

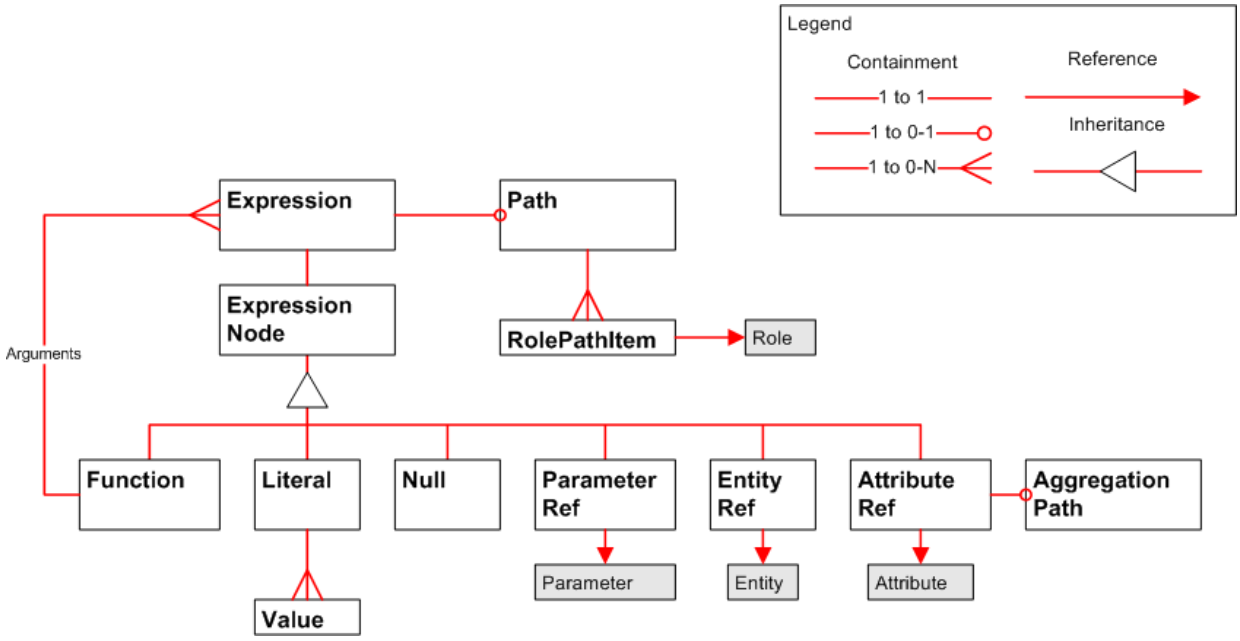


Figure 4: Expression Objects

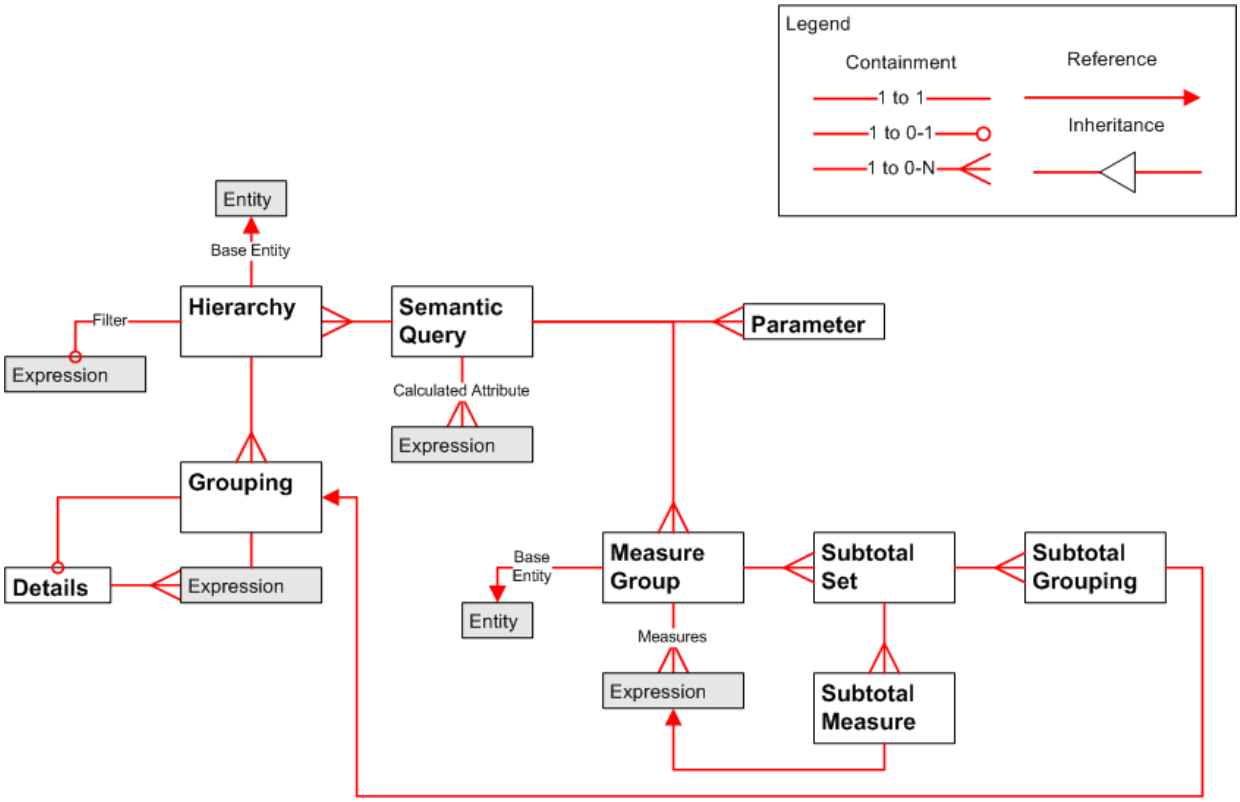


Figure 5: Query Objects

1.4 Relationship to Protocols and Other Structures

SMDL depends on the following protocols:

SQL Server Analysis Services Protocol [MS-SSAS]: Used to specify a DSV that specifies the physical data model for a semantic model.

SMDL is used as a payload in the following protocols:

Report Definition Language Schema for Version 2005/01 [MSFT-RDL200501]: Used when defining a report dataset query for a semantic model.

Report Definition Language Schema for Version 2008/01 [MSFT-RDL200801]: Used when defining a report dataset query for a semantic model.

Report Definition Language Schema for Version 2010/01 [MSFT-RDL201001]: Used when defining a report dataset query for a semantic model.

ReportExecution2005 Web Service Protocol [MS-RSWSRE2005]: Used when setting the definition of a report for execution.

ReportService2005 Web Service Protocol [MS-RSWSRMNM2005]: Used when setting or retrieving the definition of a report in the report catalog.

ReportService2006 Web Service Protocol [MS-RSWSRMSM2006]: Used when setting or retrieving the definition of a report in the report catalog.

1.5 Applicability Statement

This persistence format is applicable for use as a standalone representation of semantic model metadata and for publishing semantic model metadata from a client to a server.

1.6 Versioning and Localization

This document specifies the structures for the following version of SMDL:

- 2004/10 [MSFT-SMDL200410]

This version is defined in Appendix B: Product Behavior in this document.

There are no localization-dependent structures in the SMDL file format.

1.7 Vendor-Extensible Fields

SMDL is not an open schema. Only the CustomProperties element can be used for annotating the model, but entire XML substructures can be used as values in **CustomProperties** elements.

2 Structures

2.1 Introduction

As an XML file that conforms to an XML schema, SMDL is comprised of simple and complex elements. SMDL has other structural constraints and restrictions that cannot be expressed fully by an XML schema.

This section specifiies the elements that are defined in the SMDL, describing the meaning, restrictions, and constraints of each element and its value, if applicable; the relationships between the element and others; and the XML schema definition of the element.

The SMDL XML file is a valid XML file, as specified in [XML10/5], which conforms to one of the XSD specifications defined in section 1.3.3.1, depending on the SMDL version. This section specifies the structures of an SMDL XML file that conforms to this specification.

Some SMDL elements can be referenced by other SMDL elements. The SMDL elements that can be referenced are either given a name, which is a String with a value such as "Products", or they are given an ID, which is a QName with a value such as "G2e7933c5-4a48-4758-acb0-ce5fe8cab66e". Associating a name or ID with an SMDL element is accomplished by providing the SMDL element a **Name** attribute or child element or an **ID** attribute with an appropriate value.

The other SMDL elements that reference these SMDL elements have a child element with an element name, such as **GroupingName** or **AttributeID** and the value of that child element is the same as the name or ID of the element that it is referencing. When a reference is to an item that has a **Name** child element, that is called a "reference by name". When a reference is to an item that has an **ID** attribute, that is called a "reference by ID".

An Attribute is the only item in SMDL that can have either a reference by ID or a reference by name. The **Attribute** element has both an Attribute.Name child element and an Attribute.ID attribute.

2.2 Common SMDL Data Types

This section specifies the common data types that are used for the values of elements and attributes in SMDL. The SMDL data types are interpreted according to the XML Schema data types as specified in [XMLSCHEMA2] and [XMLSCHEMA2/2].

2.2.1 String

If the SMDL format specifies the value of an element or attribute as type **String**, the value MUST be a valid UNICODE string, as specified in [XMLSCHEMA2] section 3.2.1 for **xsd:string**.

2.2.2 Integer

If the SMDL format specifies the value of an element or attribute as type **Integer**, the value MUST be a valid int value, as specified in [XMLSCHEMA2] section 3.3.17 for **xsd:int**. A valid int value MUST be greater than -2147483649 and MUST be less than 2147483648.

2.2.3 Boolean

If the SMDL format specifies the value of an element or attribute as type **Boolean**, the value MUST be a valid Boolean value, as specified in [XMLSCHEMA2] section 3.2.2 for **xsd:boolean**.

2.2.4 DateTime

If the SMDL format specifies the value of an element or attribute as type **DateTime**, it MUST be a valid date and time-of-day value, as specified in [ISO8601] and in [XMLSCHEMA2] section 3.2.7 for **xsd:dateTime**.

2.2.5 Time

If the SMDL format specifies the value of an element or attribute as type **Time**, it MUST be a valid time-of-day value, as specified in [ISO8601] and in [XMLSCHEMA2] section 3.2.8 for **xsd:time**.

2.2.6 Decimal

If the SMDL format specifies the value of an element or attribute as type **Decimal**, the value MUST be a valid decimal value, as specified in [XMLSCHEMA2] section 3.2.3 for **xsd:decimal**. A valid decimal value MUST be greater than -79,228,162,514,264,337,593,543,950,335 and MUST be less than 79,228,162,514,264,337,593,543,950,335. The decimal value's fractional part, if present, MUST have 28 or fewer digits, and the unscaled decimal MUST be within the range specified.

2.2.7 Float

If the SMDL format specifies the value of an element or attribute as type **Float**, it MUST be a valid IEEE single-precision 32-bit floating point type [IEEE754], as specified in [XMLSCHEMA2] section 3.2.4 for **xsd:float**.

2.2.8 Binary

If the SMDL format specifies the value of an element or attribute as type **Binary**, it MUST be a valid Base64-encoded binary value [RFC3548], as specified in [XMLSCHEMA2] section 3.2.16 for **xsd:base64Binary**.

2.2.9 EntityKey

If the SMDL format specifies the value of an element or attribute as type **EntityKey**, it MUST be a valid Base64-encoded binary value [RFC3548], as specified in [XMLSCHEMA2] section 3.2.16 for **xsd:base64Binary**. The value MUST uniquely identify an instance of an entity.

An **EntityKey** is created from a composite of the key columns for the entity. The first byte contains a set of flags. The low seven flag bits are reserved and MUST be equal to zero. The high bit indicates whether the next set of bytes contains an optional null value bitmask. If the high bit is set to 1, the null value bitmask is present.

The null value bitmask, if present, has one bit per key column, rounded up to full bytes. Each bit set to 1 indicates that the corresponding column (keys ordered as above; high-to-low bits) contains a null.

The first byte and optional null value bitmask are followed by a binary serialization of the key columns of the table to which the entity is bound (or the column to which the entity is bound) in their native types. Key columns are serialized in the order in which they are defined in the primary key list in the physical model.

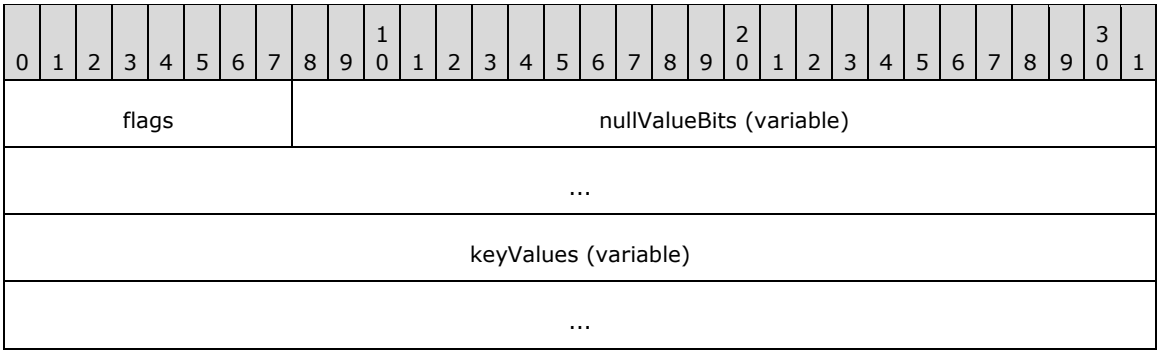
The structure of an **EntityKey** as a binary value is specified by the EntityKey structure and by the KeyValue structure, which specifies the structure of an individual key column. The description of these structures follows.

This document uses Augmented Backus-Naur Form (ABNF) as specified in [RFC5234] to specify the sequence of fields for the **EntityKey** and **KeyValue** structures that follow.

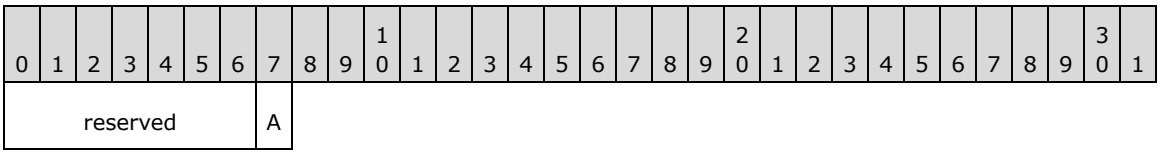
2.2.9.1 EntityKey Structure

The **EntityKey** structure specifies a collection of key column values corresponding to an instance of an entity.

```
EntityKey = flags [nullValueBits] *KeyValue
```



flags (1 byte): A byte field that specifies the structure of the EntityKey. This field is a bitmask and is specified by the following bit fields.



reserved (7 bits): The value MUST be zero.

A (1 bit): IncludeNullValues - A bit that specifies that the next set of bytes contains a null value bitmask array (**nullValueBits**). **IncludeNullValues** is the most significant bit (MSB) of the **flags** field.

Value	Meaning
0	The null value bitmask array (**nullValueBits**) is not present, and the next set of bytes is the **keyValues** array.
1	The null value bitmask array (**nullValueBits**) is present.

nullValueBits (variable): An optional bit array that specifies for each key column whether the value of the key column is NULL. If a bit in this array is set to 1, the value of the corresponding key column is NULL, and the value of the key column is omitted from the key values array (**keyValues**). If a bit in this array is set to 0, then the value of the corresponding key column is included in the key values array (**keyValues**).

The **nullValueBits** bit array is not present if **IncludeNullValues** is set to 0. If the number of key columns is not a multiple of 8, padding is placed in the least significant bits (LSB) of the last byte. These padding bits are not used and can contain any value.

The size of the **nullValueBits** field, in bytes, is calculated using the following formula:

size of **nullValueBits** in bytes = 1+floor(*n* / 8), where *n* is the number of key columns for the entity.

The padding of this field, in bits, is calculated using the following formula:

padding = size of **nullValueBits** in bits – *n*, where *n* is the number of key columns for the entity.

The position of each bit in the array corresponds to the position of each key column in the key column collection, arranged as follows:

- The most significant bit of the first byte corresponds to the first key column in the collection. The next most significant bit corresponds to the next key column in the collection. This pattern continues for each byte in the sequence.
- Any padding is placed in the least significant bits of the last byte.

The following tables demonstrate this order.

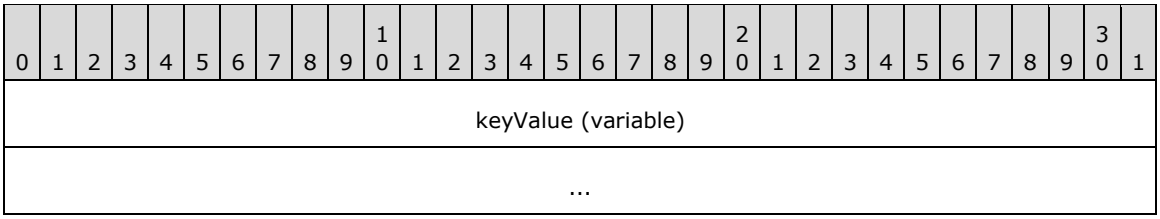
Byte 0							
0	1	2	3	4	5	6	7 (MSB)
Key 7	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	Key 0

Byte 1							
8 (LSB)	9	10	11	12	13	14	15
Padding				Key 11	Key 10	Key 9	Key 8

keyValues (variable): An array of KeyValue structures that specifies the values of the key columns. If **IncludeNullValues** is set to 0, the number of values in this array MUST equal the total number of key columns. If **IncludeNullValues** is set to 1, the number of values in this array MUST equal the number of 0 values in the **nullValueBits** bit array, excluding any padding.

2.2.9.2 KeyValue Structure

The **KeyValue** structure specifies the value of a key column.



keyValue (variable): A binary value that represents the value of a key column. The type of the binary value MUST be one of the types in the following table and is determined by the data type of the corresponding key column. For each type, the following table describes the size in bytes of the value and the method used to deserialize the binary value.

CLR data type	Size in bytes	Deserialization method
String	variable	BinaryReader.ReadString
Char	2	BinaryReader.ReadChar
Int32	4	BinaryReader.ReadInt32
Int16	2	BinaryReader.ReadInt16

CLR data type	Size in bytes	Deserialization method
UInt16	2	BinaryReader.ReadUInt16
Byte	1	BinaryReader.ReadByte
SByte	1	BinaryReader.ReadSByte
Decimal	16	BinaryReader.ReadDecimal
Int64	8	BinaryReader.ReadInt64
UInt64	8	BinaryReader.ReadUInt64
UInt32	4	BinaryReader.ReadUInt32
Double	8	BinaryReader.ReadDouble
Single	4	BinaryReader.ReadSingle
DateTime	8	
DateTimeOffset	8	
TimeSpan	8	
Boolean	1	BinaryReader.ReadBoolean
Guid	16	
ByteArray	variable	

Key column values are deserialized from a stream by applying a **BinaryReader** [MSDN-BRM] method as specified in the preceding table according to the type of the value. The five exceptions to this rule are key column values that have one of the following types: **DateTime**, **DateTimeOffset**, **TimeSpan**, **Guid**, and **ByteArray**.

Key column values that have the **DateTime** or **DateTimeOffset** type are deserialized by first applying the **BinaryReader.ReadInt64** [MSDN-BRM] method and then applying the **DateTime.FromBinary** [MSDN-DTM] method.

Key column values that have the **TimeSpan** type are deserialized by first applying the **BinaryReader.ReadInt64** [MSDN-BRM] method and then applying the **TimeSpan.FromTicks** [MSDN-TSFTM] method.

Key column values that have the **Guid** type are deserialized by first calling the **BinaryReader.ReadBytes(16)** [MSDN-BRM] method and then creating a new globally unique identifier (GUID) with the result of the method call.

Key column values that have the **ByteArray** type are deserialized by first deserializing the *length* of the byte array by using the **BinaryReader.ReadInt32** [MSDN-BRM] method and then deserializing the bytes by using the **BinaryReader.ReadBytes(*length*)** [MSDN-BRM] method.

Binary serialization of key column values to a stream is performed by applying the **BinaryWriter.Write** [MSDN-BWM] method to the key column value whose type is specified in the preceding table. The five exceptions to this rule are values that have one of the following types: **DateTime**, **DateTimeOffset**, **TimeSpan**, **Guid**, and **ByteArray**.

DateTime is serialized by first applying the **DateTime.ToBinary** [MSDN-DTM] method followed by **BinaryWriter.Write** [MSDN-BWM].

DateTimeOffset is serialized by first applying the **DateTimeOffset.UtcDateTime** [MSDN-DTOUTCP] and **DateTime.ToBinary** [MSDN-DTM] methods followed by **BinaryWriter.Write** [MSDN-BWM].

TimeSpan is serialized by first applying the **TimeSpan.Ticks** [MSDN-TSTP] method followed by **BinaryWriter.Write** [MSDN-BWM].

Guid is serialized by first applying the Guid.ToByteArray [MSDN-GTBAM] method followed by BinaryWriter.Write [MSDN-BWM].

ByteArray is serialized by first serializing the Int32 length of the byte array by using BinaryWriter.Write [MSDN-BWM] and then serializing the byte array by using BinaryWriter.Write [MSDN-BWM].

2.2.10 Variant

If the SMDL format specifies the value of an element as type **Variant**, the value of the element MUST be a valid simple XML type value, as specified in [XMLSCHEMA2] section 2.5.2 for **xsd:anySimpleType**.

If the SMDL data type **Variant** is referenced by the CustomProperty.Value element, the data type of the **Variant** is specified by the value of the **xsi:type** attribute on the **CustomProperty.Value** element.

If the SMDL data type **Variant** is referenced by the Literal.Value or Values.Value elements, and if the value of the related Literal.DataType element is not "EntityKey", the data type of the **Variant** is specified by the value of the **Literal.DataType** element. If the value of the **Literal.DataType** element is "EntityKey", the value of the element is not a **Variant**.

The **Variant** data type is referenced by the following elements.

Referenced by
CustomProperty.Value
Literal.Value
Values.Value

2.2.11 QName

If the SMDL format specifies the value of an element or attribute as type **QName**, the value MUST be a valid XML qualified name value, as specified in [XMLSCHEMA2] section 3.2.18 for **xsd:QName**. A further restriction on such XML qualified names is specified below.

An XML qualified name consists of a prefix string specifying a namespace, followed by a colon and then followed by a string specifying the local part of the qualified name. The prefix followed by a colon is optional, but the local part of the qualified name MUST be specified.

If the namespace for the qualified name is the SMDL namespace defined in section 1.3.3.1, the local part of the qualified name MUST be a GUID that has a "G" prepended. For example, the string "G81aad0b6-be6e-4ef1-a412-de56263e3bc5" is a **QName**. Such an XML qualified name uniquely identifies an object globally.

The **QName** data type is referenced by the following elements and attributes.

Referenced by
Attribute.DefaultAggregateAttributeID
Attribute.ID
AttributeRef.AttributeID
AttributeReference.AttributeID
BaseEntity.EntityID
CustomProperty.Name
Entity.ID
EntityFolder.ID
EntityRef.EntityID
FieldFolder.ID
HiddenFields.FieldFolderItemID
Inheritance.InheritsFromEntityID
ModelItems.ModelItemID
Perspective.ID
Role.ID
Role.RelatedRoleID
RolePathItem.RoleID
SemanticModel.ID

2.2.12 MIMEType

If the SMDL format specifies the value of an element or attribute as type **MIMEType**, the value MUST be a valid String ([XMLSCHEMA2] section 3.2.1) value that SHOULD<1> be a valid MIME [RFC2045] [RFC2046] type.

The **MIMEType** data type is referenced by the following element.

Referenced by
Attribute.MimeType

2.2.13 Language

If the SMDL format specifies the value of an element or attribute as type **Language**, the value MUST be a valid language value [RFC4646], as specified in [XMLSCHEMA2] section 3.3.3 for **xsd:language**.

The language code is a combination of the following:

- A two-letter lowercase culture code that is associated with a language, as specified in [ISO639-2].
- A two-letter uppercase subculture code that is associated with a country or region, as specified in [ISO3166-1].

For example, the string "en-US" is a **Language**.

The **Language** data type is referenced by the following elements.

Referenced by
Attribute.DataCulture
SemanticModel.Culture

2.3 SemanticModel

The **SemanticModel** element specifies all of the entities, attributes, and roles of a semantic model. This element is the top-level root node structure of the semantic model of data.

The following are the attributes and child elements of the **SemanticModel** element.

Attributes
SemanticModel.ID

Child elements
SemanticModel.Culture
SemanticModel.CustomProperties
SemanticModel.DataSourceView
SemanticModel.Description
SemanticModel.Entities
SemanticModel.Perspectives
SemanticModel.Version

The following is the XML Schema definition of the **SemanticModel** element.

```
<xsd:element name="SemanticModel">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Description" type="xsd:string" minOccurs="0" />
      <xsd:element name="Version" type="xsd:string" minOccurs="0" />
      <xsd:element name="Culture" type="xsd:language" minOccurs="0" />
      <xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
      <xsd:element name="Perspectives" type="PerspectivesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType"
                   minOccurs="0" />
      <xsd:element ref="dsv:DataSourceView" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
```

```
  </xsd:complexType>
  <xsd:unique name="ID">
    <xsd:selector xpath=".|.//*" />
    <xsd:field xpath="@ID" />
  </xsd:unique>
</xsd:element>
```

2.3.1 SemanticModel.ID

The **SemanticModel.ID** attribute specifies an immutable GUID of the semantic model.

The **SemanticModel.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **SemanticModel.ID** attribute.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.3.2 SemanticModel.Culture

The **SemanticModel.Culture** element specifies the primary language or culture used for the name properties in the semantic model.

The **SemanticModel.Culture** element is optional. If the **SemanticModel.Culture** element is present, its value MUST be a Language that is a valid language code string. If the **SemanticModel.Culture** element is not present, its value is interpreted as the value of the culture of the server.

The following is the parent element of the **SemanticModel.Culture** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.Culture** element.

```
<xsd:element name="Culture" type="xsd:language" minOccurs="0" />
```

2.3.3 SemanticModel.CustomProperties

The **SemanticModel.CustomProperties** element specifies a collection of custom properties for the semantic model.

The **SemanticModel.CustomProperties** element is optional. This element is of type CustomProperties. If the **SemanticModel.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticModel.CustomProperties** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.3.4 (Updated Section) SemanticModel.DataSourceView

The **SemanticModel.DataSourceView** element specifies the DSV that specifies the physical model physical model.

The **SemanticModel.DataSourceView** element is optional. If this element is not present, its value is interpreted as NULL.

The **SemanticModel.DataSourceView** element is of type **DataSourceView**. The **DataSourceView** element type is defined external to SMDL and is not described in this document; the **DataSourceView** element type is specified in [MS-SSAS] section 2.2.4.2.2.7 ~~of [MS-SSAS].~~.

The following is the parent element of the **SemanticModel.DataSourceView** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.DataSourceView** element.

```
<xsd:element ref="dsv:DataSourceView" minOccurs="0" />
```

2.3.5 SemanticModel.Description

The **SemanticModel.Description** element specifies a textual description of a semantic model.

The **SemanticModel.Description** element is optional. If this element is present, its value MUST be a String. If the **SemanticModel.Description** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticModel.Description** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.3.6 SemanticModel.Entities

The **SemanticModel.Entities** element specifies the collection of entities in the semantic model.

The **SemanticModel.Entities** element is optional. This element is of type Entities. If the **SemanticModel.Entities** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticModel.Entities** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.Entities** element.

```
<xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
```

2.3.7 SemanticModel.Perspectives

The **SemanticModel.Perspectives** element specifies the collection of subsets of the semantic model to provide to users as constrained submodels.

The **SemanticModel.Perspectives** element is optional. This element is of type Perspectives. If the **SemanticModel.Perspectives** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticModel.Perspectives** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.Perspectives** element.

```
<xsd:element name="Perspectives" type="PerspectivesType" minOccurs="0" />
```

2.3.8 SemanticModel.Version

The **SemanticModel.Version** element specifies a user-defined version string for the semantic model.

The **SemanticModel.Version** element is optional. If this element is present, its value MUST be a String. If the **SemanticModel.Version** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticModel.Version** element.

Parent elements
SemanticModel

The following is the XML Schema definition of the **SemanticModel.Version** element.

```
<xsd:element name="Version" type="xsd:string" minOccurs="0" />
```

2.4 CustomProperties

The **CustomProperties** element specifies a collection of CustomProperty elements that allows model design tools to store implementation-specific properties.

The **CustomProperties** element MUST contain at least one CustomProperties.CustomProperty element and can contain more.

The following are the parent and child elements of the **CustomProperties** element.

Parent elements
Attribute
Entity
EntityFolder
Expression
FieldFolder
Perspective
Role
SemanticModel
SemanticQuery

Child elements
CustomProperties.CustomProperty

The following is the XML Schema definition of the **CustomProperties** element.

```
<xsd:complexType name="CustomPropertiesType">
  <xsd:sequence>
    <xsd:element name="CustomProperty" type="CustomPropertyType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.4.1 CustomProperties.CustomProperty

The **CustomProperties.CustomProperty** element specifies a custom property for a model item that allows model design tools to store implementation-specific properties.

The **CustomProperties.CustomProperty** element MUST be specified at least once in its parent CustomProperties collection and can occur more than once. This element is of type CustomProperty.

The following is the parent element of the **CustomProperties.CustomProperty** element.

Parent elements
CustomProperties

The following is the XML Schema definition of the **CustomProperties.CustomProperty** element.

```
<xsd:element name="CustomProperty" type="CustomPropertyType"
maxOccurs="unbounded" />
```

2.5 CustomProperty

The **CustomProperty** element specifies application-specific data for a model item.

The following are the parent elements, attributes, and child elements of the **CustomProperty** element.

Parent elements
CustomProperties

Attributes
CustomProperty.Name

Child elements
CustomProperty.Value

The following is the XML Schema definition of the **CustomProperty** element.

```
<xsd:complexType name="CustomPropertyType">
  <xsd:all>
    <xsd:element name="Value" type="xsd:anySimpleType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.5.1 CustomProperty.Name

The **CustomProperty.Name** attribute specifies the name of the custom property.

The **CustomProperty.Name** attribute MUST be specified. The value of this attribute MUST be a fully qualified QName. Custom property names are not required to be unique.

The following is the parent element of the **CustomProperty.Name** attribute.

Parent elements
CustomProperty

The following is the XML Schema definition of the **CustomProperty.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:QName" use="required" />
```

2.5.2 CustomProperty.Value

The **CustomProperty.Value** element specifies the value of a custom property.

The **CustomProperty.Value** element is optional. The value of this element MUST be a Variant. The data type of the value of the **Variant** MUST be specified by the value of the **xsi:type** attribute on the **CustomProperty.Value** element. If the **xsi:type** attribute is not present on the

CustomProperty.Value element, the value of the **Variant** is interpreted as a String. If the **CustomProperty.Value** element is not present, its value is interpreted as NULL.

The following is the parent element of the **CustomProperty.Value** element.

Parent elements
CustomProperty

The following is the XML Schema definition of the **CustomProperty.Value** element.

```
<xsd:element name="Value" type="xsd:anySimpleType" minOccurs="0" />
```

2.6 Entities

The **Entities** element specifies a collection that is an ordered list of Entity and EntityFolder elements in the model.

The **Entities** element MUST contain at least one Entities.Entity or Entities.EntityFolder element and can contain more.

The following are the parent and child elements of the **Entities** element.

Parent elements
EntityFolder
SemanticModel

Child elements
Entities.Entity
Entities.EntityFolder

The following is the XML Schema definition of the **Entities** element.

```
<xsd:complexType name="EntitiesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Entity" type="EntityType" />
    <xsd:element name="EntityFolder" type="EntityFolderType" />
  </xsd:choice>
</xsd:complexType>
```

2.6.1 Entities.Entity

The **Entities.Entity** element specifies an entity in the model.

The **Entities.Entity** element is optional. This element is of type Entity. At least one **Entities.Entity** or Entities.EntityFolder element MUST be specified as a child of the parent Entities collection.

The following is the parent element of the **Entities.Entity** element.

Parent elements
Entities

The following is the XML Schema definition of the **Entities.Entity** element.

```
<xsd:element name="Entity" type="EntityType" />
```

2.6.2 Entities.EntityFolder

The **Entities.EntityFolder** element specifies a folder that can contain entities and further entity folders.

The **Entities.EntityFolder** element is optional. This element is of type EntityFolder. At least one Entities.Entity or **Entities.EntityFolder** element MUST be specified as a member of the parent Entities collection.

The following is the parent element of the **Entities.EntityFolder** element.

Parent elements
Entities

The following is the XML Schema definition of the **Entities.EntityFolder** element.

```
<xsd:element name="EntityFolder" type="EntityFolderType" />
```

2.7 Entity

The **Entity** element specifies an entity in the semantic model.

The following are the parent elements, attributes, and child elements of the **Entity** element.

Parent elements
Entities

Attributes
Entity.ID

Child elements
Entity.CollectionName
Entity.Column
Entity.CustomProperties
Entity.DefaultAggregateAttributes

Child elements
Entity.DefaultDetailAttributes
Entity.DefaultSecurityFilter
Entity.Description
Entity.DisjointInheritance
Entity.Fields
Entity.Hidden
Entity.IdentifyingAttributes
Entity.Inheritance
Entity.InstanceSelection
Entity.IsLookup
Entity.Name
Entity.SecurityFilters
Entity.SortAttributes
Entity.Table

The following is the XML Schema definition of the **Entity** element.

```
<xsd:complexType name="EntityType">
  <xsd:all>
    <!-- ModelItemType Start-->
    <xsd:element name="Name" type="NonEmptyString" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
    <!-- ModelItemType End-->
    <xsd:element name="CollectionName" type="NonEmptyString" minOccurs="0" />
    <xsd:element name="IdentifyingAttributes" type="AttributeReferencesType" />
    <xsd:element name="DefaultDetailAttributes" type="AttributeReferencesType"
                 minOccurs="0" />
    <xsd:element name="DefaultAggregateAttributes" type="AttributeReferencesType"
                 minOccurs="0" />
    <xsd:element name="SortAttributes" type="SortAttributesType" minOccurs="0" />
    <xsd:element name="InstanceSelection">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="FilteredList" />
          <xsd:enumeration value="Dropdown" />
          <xsd:enumeration value="List" />
          <xsd:enumeration value="MandatoryFilter" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="IsLookup" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Inheritance" minOccurs="0">
      <xsd:complexType>
        <xsd:all>
          <xsd:element name="InheritsFromEntityID" type="xsd:QName" />
          <xsd:element name="Relation" type="RelationType" minOccurs="0" />
        </xsd:all>
      </xsd:complexType>
```

```
    </xsd:element>
    <xsd:element name="DisjointInheritance" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="SecurityFilters" type="AttributeReferencesType"
                 minOccurs="0" />
    <xsd:element name="DefaultSecurityFilter" type="DefaultSecurityFilterType"
                 minOccurs="0" />
    <xsd:element name="Table" type="TableType" minOccurs="0" />
    <xsd:element name="Column" type="ColumnType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.7.1 Entity.ID

The **Entity.ID** attribute specifies an immutable globally unique identifier (GUID) of an entity.

The **Entity.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **Entity.ID** attribute.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.7.2 Entity.CollectionName

The **Entity.CollectionName** element specifies the name used to refer to a collection of instances of the entity, such as "customers" for the entity "customer" or "faculty" for the entity "instructor".

The **Entity.CollectionName** element is optional. If this element is present, its value MUST be a nonempty String. If this element is not present, its value is interpreted as the value of Entity.Name.

The following is the parent element of the **Entity.CollectionName** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.CollectionName** element.

```
<xsd:element name="CollectionName" type="NonEmptyString" minOccurs="0" />
```

2.7.3 Entity.Column

The **Entity.Column** element specifies the database object, which is a column in a table that represents the entity and to which the entity is bound.

The **Entity.Column** element is optional. This element is of type Entity. Exactly one **Entity.Column** or Entity.Table element MUST be specified as a child of the parent **Entity** element. If the **Entity.Column** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.Column** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Column** element.

```
<xsd:element name="Column" type="ColumnType" minOccurs="0" />
```

2.7.4 Entity.CustomProperties

The **Entity.CustomProperties** element specifies a collection of custom properties for an entity.

The **Entity.CustomProperties** element is optional. This element is of type CustomProperties. If the **Entity.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.CustomProperties** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.7.5 Entity.DefaultAggregateAttributes

The **Entity.DefaultAggregateAttributes** element specifies a collection of aggregate attributes for client applications to show when the parent Entity element is displayed as an aggregate.

The **Entity.DefaultAggregateAttributes** element is optional. This element is of type DefaultAggregateAttributes. If the **Entity.DefaultAggregateAttributes** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.DefaultAggregateAttributes** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.DefaultAggregateAttributes** element.

```
<xsd:element name="DefaultAggregateAttributes" type="AttributeReferencesType"
minOccurs="0" />
```

2.7.6 Entity.DefaultDetailAttributes

The **Entity.DefaultDetailAttributes** element specifies a collection of attributes for client applications to show when the parent Entity element is displayed.

The **Entity.DefaultDetailAttributes** element is optional. This element is of type DefaultDetailAttributes. If the **Entity.DefaultDetailAttributes** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.DefaultDetailAttributes** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.DefaultDetailAttributes** element.

```
<xsd:element name="DefaultDetailAttributes" type="AttributeReferencesType"
minOccurs="0" />
```

2.7.7 Entity.DefaultSecurityFilter

The **Entity.DefaultSecurityFilter** element specifies an attribute to use as a filter for the entity if no SecurityFilters are available (for example, because of user permissions).

The **Entity.DefaultSecurityFilter** element is optional. This element is of type DefaultSecurityFilter. If the **Entity.DefaultSecurityFilter** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.DefaultSecurityFilter** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.DefaultSecurityFilter** element.

```
<xsd:element name="DefaultSecurityFilter" type="DefaultSecurityFilterType"
minOccurs="0" />
```

2.7.8 Entity.Description

The **Entity.Description** element specifies a textual description of an entity.

The **Entity.Description** element is optional. If this element is present, its value MUST be a String. If the **Entity.Description** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.Description** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.7.9 Entity.DisjointInheritance

The **Entity.DisjointInheritance** element specifies that entities that inherit from this entity are mutually exclusive sets. For example, car and truck both inherit from vehicle but no vehicle is both a car and a truck. However, both employee and customer inherit from person and some employees might also be customers.

The **Entity.DisjointInheritance** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that inherited entities are mutually exclusive sets. If the **Entity.DisjointInheritance** element is not present, its value is interpreted as false.

The following is the parent element of the **Entity.DisjointInheritance** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.DisjointInheritance** element.

```
<xsd:element name="DisjointInheritance" type="xsd:boolean" minOccurs="0" />
```

2.7.10 Entity.Fields

The **Entity.Fields** element specifies a collection of fields as an ordered list of attributes and roles of the entity.

The **Entity.Fields** element is optional. This element is of type Fields. If the **Entity.Fields** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.Fields** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Fields** element.

```
<xsd:element name="Fields" type="FieldsType" minOccurs="0" />
```

2.7.11 Entity.Hidden

The **Entity.Hidden** element specifies that the entity is not to be displayed to the user.

The **Entity.Hidden** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the entity is not displayed to the user. If the **Entity.Hidden** element is not present, its value is interpreted as false.

The following is the parent element of the **Entity.Hidden** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

2.7.12 Entity.IdentifyingAttributes

The **Entity.IdentifyingAttributes** element specifies a collection of attributes for client applications to use to identify an instance of the entity to the user.

The **Entity.IdentifyingAttributes** element MUST be specified. This element is of type IdentifyingAttributes.

The following is the parent element of the **Entity.IdentifyingAttributes** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.IdentifyingAttributes** element.

```
<xsd:element name="IdentifyingAttributes" type="AttributeReferencesType" />
```

2.7.13 Entity.Inheritance

The **Entity.Inheritance** element specifies how the parent Entity element inherits from another entity.

The **Entity.Inheritance** element is optional. This element is of type Inheritance. If the **Entity.Inheritance** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.Inheritance** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Inheritance** element.

```
<xsd:element name="Inheritance" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="InheritsFromEntityID" type="xsd:QName" />
      <xsd:element name="Relation" type="RelationType" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.7.14 Entity.InstanceSelection

The **Entity.InstanceSelection** element specifies the behavior of the client application for selecting instances of the entity, based upon the expected number of total instances.

The **Entity.InstanceSelection** element MUST be specified. Its value MUST be a String that is one of the following:

Dropdown: Specifies that the number of instances is small enough to fit in a simple dropdown.

List: Specifies that the number of instances is too large for a simple dropdown but does not require pre-filtering.

FilteredList: Specifies that the number of instances is large enough to require the user to initially filter the values before selecting instances.

MandatoryFilter: Specifies that the number of instances is so large to prohibit users from building queries on this entity without filtering.

The following is the parent element of the **Entity.InstanceSelection** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.InstanceSelection** element.

```
<xsd:element name="InstanceSelection">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="FilteredList" />
      <xsd:enumeration value="Dropdown" />
      <xsd:enumeration value="List" />
      <xsd:enumeration value="MandatoryFilter" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.7.15 Entity.IsLookup

The **Entity.IsLookup** element specifies that client applications are to treat the entity as a lookup table rather than as a full-fledged entity. For example, roles that join to lookup entities are displayed as attributes instead of roles. Selecting the attribute shows the DefaultDetailAttributes for the lookup entity.

The **Entity.IsLookup** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that client applications can treat the entity as a lookup table. If the **Entity.IsLookup** element is not present, its value is interpreted as false.

The following is the parent element of the **Entity.IsLookup** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.IsLookup** element.

```
<xsd:element name="IsLookup" type="xsd:boolean" minOccurs="0" />
```

2.7.16 Entity.Name

The **Entity.Name** element specifies the name of the entity.

The **Entity.Name** element MUST be specified. Its value MUST be a nonempty String. The value of this element MUST be a unique name within the namespace of the semantic model.

The following is the parent element of the **Entity.Name** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" />
```

2.7.17 Entity.SecurityFilters

The **Entity.SecurityFilters** element specifies a collection of attributes to use as filters for the entity. If specified, these filters will restrict the rows from the entity that are available to users.

The **Entity.SecurityFilters** element is optional. This element is of type SecurityFilters. If the **Entity.SecurityFilters** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.SecurityFilters** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.SecurityFilters** element.

```
<xsd:element name="SecurityFilters" type="AttributeReferencesType" minOccurs="0" />
```

2.7.18 Entity.SortAttributes

The **Entity.SortAttributes** element specifies a collection of attributes on which to sort when a sort is requested by the entity.

The **Entity.SortAttributes** element is optional. This element is of type SortAttributes. If the **Entity.SortAttributes** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.SortAttributes** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.SortAttributes** element.

```
<xsd:element name="SortAttributes" type="SortAttributesType" minOccurs="0" />
```

2.7.19 Entity.Table

The **Entity.Table** element specifies the database object, which is a table that represents the entity and to which the entity is bound.

The **Entity.Table** element is optional. This element is of type Table. Exactly one Entity.Column or **Entity.Table** element MUST be specified as a child of the parent Entity element. If the **Entity.Table** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Entity.Table** element.

Parent elements
Entity

The following is the XML Schema definition of the **Entity.Table** element.

```
<xsd:element name="Table" type="TableType" minOccurs="0" />
```

2.8 Column

The **Column** element specifies the column, which is a database object to which an Entity or Attribute is bound.

The following are the parent elements and attributes of the **Column** element.

Parent elements
Attribute
Entity

Attributes
Column.Name
Column.TableName

The following is the XML Schema definition of the **Column** element.

```
<xsd:complexType name="ColumnType">
  <xsd:attribute name="TableName" type="NonEmptyString" use="optional" />
  <xsd:attribute name="Name" type="NonEmptyString" use="required" />
</xsd:complexType>
```

2.8.1 Column.Name

The **Column.Name** attribute specifies the name of the column in the physical data source to which the entity or attribute is bound.

The **Column.Name** attribute MUST be specified. Its value MUST be a nonempty String. The value of **Column.Name** MUST be the name of a column in the table specified by the value of Column.TableName. If the column is bound to an entity, the column specified by the value of the **Column.Name** attribute MUST NOT be a column with a binary value. If the column is bound to an attribute and the parent entity of the attribute is also bound to a column, the two columns MUST be the same, that is, the value of the **Column.Name** attribute MUST be equal to the value of the **Column.Name** attribute of the Column child element of the great-grandparent Entity element.

The following is the parent element of the **Column.Name** attribute.

Parent elements
Column

The following is the XML Schema definition of the **Column.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.8.2 Column.TableName

The **Column.TableName** attribute specifies the name of the table in the physical data source that contains the column.

The **Column.TableName** attribute is optional. If this attribute is present, its value MUST be a nonempty String.

If the **Column.TableName** attribute is not present, and the column is bound to an attribute of an entity, the value of the **Column.TableName** attribute is interpreted as the name of the table to which the entity is bound or as the name of the table containing the column to which the entity is bound.

If the **Column.TableName** attribute is present, and the column is bound to an attribute of an entity, the value of the **Column.TableName** attribute MUST match the name of the table of the parent entity. If the parent entity is bound to a table, the name of that table MUST match the value of the **Column.TableName** attribute. If the parent entity is bound to a column, the name of the table that contains that column MUST match the value of the **Column.TableName** attribute. In other words, the value of the **Column.TableName** attribute MUST match the value of either the Table.Name grandchild attribute or the **Column.TableName** grandchild attribute of the parent entity depending upon the binding of the parent entity.

If the column is not bound to an attribute of an entity, the **Column.TableName** attribute MUST be specified.

The following is the parent element of the **Column.TableName** attribute.

Parent elements
Column

The following is the XML Schema definition of the **Column.TableName** attribute.

```
<xsd:attribute name="TableName" type="NonEmptyString" use="optional" />
```

2.9 DefaultAggregateAttributes

The **DefaultAggregateAttributes** element specifies a collection of references to the aggregate attributes that are shown when a summary is displayed of multiple instances of the parent Entity element that the **DefaultAggregateAttributes** collection is bound to.

The **DefaultAggregateAttributes** element specifies a collection of AttributeReference elements. The **DefaultAggregateAttributes** element MUST contain at least one DefaultAggregateAttributes.AttributeReference element and can contain more.

The following are the parent and child elements of the **DefaultAggregateAttributes** element.

Parent elements
Entity

Child elements
DefaultAggregateAttributes.AttributeReference

The following is the XML Schema definition of the **DefaultAggregateAttributes** element.

```
<xsd:complexType name="AttributeReferencesType">
  <xsd:sequence>
    <xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.9.1 DefaultAggregateAttributes.AttributeReference

The **DefaultAggregateAttributes.AttributeReference** element specifies a reference to an aggregate attribute to be shown when a summary of multiple instances of the grandparent Entity element is displayed.

The **DefaultAggregateAttributes.AttributeReference** element MUST be specified at least once in its parent DefaultAggregateAttributes collection and can occur more than once. The **DefaultAggregateAttributes.AttributeReference** element is of type AttributeReference. The attribute that is referenced by the **DefaultAggregateAttributes.AttributeReference** element MUST have its Attribute.IsAggregate element set to true.

The following is the parent element of the **DefaultAggregateAttributes.AttributeReference** element.

Parent elements
DefaultAggregateAttributes

The following is the XML Schema definition of the **DefaultAggregateAttributes.AttributeReference** element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
```

2.10 AttributeReference

The **AttributeReference** element specifies a reference to an attributes along with an optional path by which to reach that attribute.

The following are the parent and child elements of the **AttributeReference** element.

Parent elements
DefaultAggregateAttributes
DefaultDetailAttributes
DefaultSecurityFilter
IdentifyingAttributes
SecurityFilters

Parent elements
SortAttribute

Child elements
AttributeReference.AttributeID
AttributeReference.Path

The following is the XML Schema definition of the **AttributeReference** element.

```
<xsd:complexType name="AttributeReferenceType">
  <xsd:all>
    <xsd:element name="Path" type="PathType" minOccurs="0" />
    <xsd:element name="AttributeID" type="xsd:QName" />
  </xsd:all>
</xsd:complexType>
```

2.10.1 AttributeReference.AttributeID

The **AttributeReference.AttributeID** element specifies the ID of an attribute in the semantic model.

The **AttributeReference.AttributeID** element MUST be specified. The value of this element MUST be a QName. The value of this element MUST match the value of the Attribute.ID attribute of one of the attributes in the semantic model.

The following is the parent element of the **AttributeReference.AttributeID** element.

Parent elements
AttributeReference

The following is the XML Schema definition of the **AttributeReference.AttributeID** element.

```
<xsd:element name="AttributeID" type="xsd:QName" />
```

2.10.2 AttributeReference.Path

The **AttributeReference.Path** element specifies the path by which to reach the attribute or, in the case of an aggregate attribute, the path by which to reach the aggregated expressions within the referenced attribute.

The **AttributeReference.Path** element is optional. This element is of type Path. If the **AttributeReference.Path** element is not present, its value is interpreted as NULL, and the referenced Attribute MUST be contained within the same Entity that contains the AttributeReference parent element.

If the reference is to an attribute whose value of the Attribute.IsAggregate element is false, the path MUST NOT contain a role whose Role.Cardinality element has the value "Many" or "OptionalMany".

The following is the parent element of the **AttributeReference.Path** element.

Parent elements
AttributeReference

The following is the XML Schema definition of the **AttributeReference.Path** element.

```
<xsd:element name="Path" type="PathType" minOccurs="0" />
```

2.11 Path

The **Path** element specifies a collection of RolePathItem elements as an ordered list. This list is the series of steps (typically roles) to follow to reach the desired entity.

The **Path** element MUST contain at least one Path.RolePathItem element and can contain more.

The following are the parent and child elements of the **Path** element.

Parent elements
AttributeReference
Expression

Child elements
Path.RolePathItem

The following is the XML Schema definition of the **Path** element.

```
<xsd:complexType name="PathType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="RolePathItem" type="RolePathItemType" />
  </xsd:choice>
</xsd:complexType>
```

2.11.1 Path.RolePathItem

The **Path.RolePathItem** element specifies a role in a path.

The **Path.RolePathItem** element MUST be specified at least once in its parent Path collection and can occur more than once. This element is of type RolePathItem.

The following is the parent element of the **Path.RolePathItem** element.

Parent elements
Path

The following is the XML Schema definition of the **Path.RolePathItem** element.

```
<xsd:element name="RolePathItem" type="RolePathItemType" />
```

2.12 RolePathItem

The **RolePathItem** element specifies a role in a path.

The following are the parent and child elements of the **RolePathItem** element.

Parent elements
Path
SelectedPath

Child elements
RolePathItem.RoleID

The following is the XML Schema definition of the **RolePathItem** element.

```
<xsd:complexType name="RolePathItemType">
  <xsd:all>
    <xsd:element name="RoleID" type="xsd:QName" />
  </xsd:all>
</xsd:complexType>
```

2.12.1 RolePathItem.RoleID

The **RolePathItem.RoleID** element specifies the ID of the role in the path.

The **RolePathItem.RoleID** element MUST be specified. The value of this element MUST be a QName. The value of the **RolePathItem.RoleID** element MUST match the value of the Role.ID attribute of one of the roles in the semantic model.

The following is the parent element of the **RolePathItem.RoleID** element.

Parent elements
RolePathItem

The following is the XML Schema definition of the **RolePathItem.RoleID** element.

```
<xsd:element name="RoleID" type="xsd:QName" />
```

2.13 DefaultDetailAttributes

The **DefaultDetailAttributes** element specifies a collection of AttributeReference elements that are references to the attributes that are shown when an instance of the parent Entity element is displayed.

The **DefaultDetailAttributes** element MUST contain at least one DefaultDetailAttributes.AttributeReference element and can contain more.

The following are the parent and child elements of the **DefaultDetailAttributes** element.

Parent elements
Entity

Child elements
DefaultDetailAttributes.AttributeReference

The following is the XML Schema definition of the **DefaultDetailAttributes** element.

```
<xsd:complexType name="AttributeReferencesType">
  <xsd:sequence>
    <xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.13.1 DefaultDetailAttributes.AttributeReference

The **DefaultDetailAttributes.AttributeReference** element specifies a reference to an attribute to show when an entity is displayed.

The **DefaultDetailAttributes.AttributeReference** element MUST be specified at least once in its parent DefaultDetailAttributes collection and can occur more than once. This element is of type AttributeReference.

The following is the parent element of the **DefaultDetailAttributes.AttributeReference** element.

Parent elements
DefaultDetailAttributes

The following is the XML Schema definition of the **DefaultDetailAttributes.AttributeReference** element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
```

2.14 DefaultSecurityFilter

The **DefaultSecurityFilter** element specifies a reference to an attribute that is used as a filter on the parent Entity element if no SecurityFilters elements are present either because no **SecurityFilters** elements are defined or because the user does not have permission to any of the **SecurityFilters** elements.

The following are the parent and child elements of the **DefaultSecurityFilter** element.

Parent elements
Entity

Child Elements
DefaultSecurityFilter.AttributeReference

The following is the XML Schema definition of the **DefaultSecurityFilter** element.

```
<xsd:complexType name="DefaultSecurityFilterType">
  <xsd:all>
    <xsd:element name="AttributeReference" type="AttributeReferenceType" />
  </xsd:all>
</xsd:complexType>
```

2.14.1 DefaultSecurityFilter.AttributeReference

The **DefaultSecurityFilter.AttributeReference** element specifies a reference to an attribute to apply as a filter to the grandparent Entity element.

The **DefaultSecurityFilter.AttributeReference** element MUST be specified. This element is of type AttributeReference. The attribute that this reference specifies MUST have the value of its Attribute.IsFilter element equal to true.

The following is the parent element of the **DefaultSecurityFilter.AttributeReference** element.

Parent elements
DefaultSecurityFilter

The following is the XML Schema definition of the **DefaultSecurityFilter.AttributeReference** element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType" />
```

2.15 Fields

The **Fields** element specifies a collection that is an ordered list of Attribute, Role, and FieldFolder elements. These elements represent the fields (attributes, and roles) and field folders for an entity.

The **Fields** element MUST contain at least one Fields.Attribute, Fields.Role, or Fields.FieldFolder child element and can contain more.

The following are the parent and child elements of the **Fields** element.

Parent elements
Entity
FieldFolder

Child elements
Fields.Attribute
Fields.FieldFolder

Child elements
Fields.Role

The following is the XML Schema definition of the **Fields** element.

```
<xsd:complexType name="FieldsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Attribute" type="AttributeType" />
    <xsd:element name="Role" type="RoleType" />
    <xsd:element name="FieldFolder" type="FieldFolderType" />
  </xsd:choice>
</xsd:complexType>
```

2.15.1 Fields.Attribute

The **Fields.Attribute** element specifies an Attribute element that represents an attribute for an ancestor entity.

The **Fields.Attribute** element is optional. At least one **Fields.Attribute**, Fields.Role, or Fields.FieldFolder element MUST be specified for the Fields parent element. The **Fields.Attribute** element is of type **Attribute**.

The following is the parent element of the **Fields.Attribute** element.

Parent elements
Fields

The following is the XML Schema definition of the **Fields.Attribute** element.

```
<xsd:element name="Attribute" type="AttributeType" />
```

2.15.2 Fields.FieldFolder

The **Fields.FieldFolder** element specifies a FieldFolder element that represents a folder that can contain more fields, such as attributes roles, and field folders.

The **Fields.FieldFolder** element is optional. At least one Fields.Attribute, Fields.Role, or **Fields.FieldFolder** element MUST be specified for the Fields parent element. The **Fields.FieldFolder** element is of type FieldFolder.

The following is the parent element of the **Fields.FieldFolder** element.

Parent elements
Fields

The following is the XML Schema definition of the **Fields.FieldFolder** element.

```
<xsd:element name="FieldFolder" type="FieldFolderType" />
```

2.15.3 Fields.Role

The **Fields.Role** element specifies a Role element that represents a role for the entity.

The **Fields.Role** element is optional. At least one Fields.Attribute, **Fields.Role**, or Fields.FieldFolder element MUST be specified for the Fields parent element. The **Fields.Role** element is of type **Role**.

The following is the parent element of the **Fields.Role** element.

Parent elements
Fields

The following is the XML Schema definition of the **Fields.Role** element.

```
<xsd:element name="Role" type="RoleType" />
```

2.16 Attribute

The **Attribute** element specifies an attribute of an entity.

The following are the parent elements, attributes, and child elements of the **Attribute** element.

Parent elements
Fields
Variations

Attributes
Attribute.ID

Child elements
Attribute.Alignment
Attribute.Column
Attribute.ContextualName
Attribute.CustomProperties
Attribute.DataCulture
Attribute.DataType
Attribute.DefaultAggregateAttributeID
Attribute.Description
Attribute.DiscourageGrouping
Attribute.EnableDrillthrough

Child elements
Attribute.Expression
Attribute.Format
Attribute.Hidden
Attribute.IsAggregate
Attribute.IsFilter
Attribute.MimeType
Attribute.Name
Attribute.Nullable
Attribute.OmitSecurityFilters
Attribute.SortDirection
Attribute.ValueSelection
Attribute.Variations
Attribute.Width

The following is the XML Schema definition of the **Attribute** element.

```
<xsd:complexType name="AttributeType">
  <xsd:all>
    <!-- ModelItemType Start-->
    <xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
    <!-- ModelItemType End-->
    <!-- FieldType Start-->
    <xsd:element name="Variations" type="VariationsType" minOccurs="0" />
    <!-- FieldType End-->
    <xsd:element name="DataType" type="DataTypeEnum" />
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
    <xsd:element name="IsAggregate" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="IsFilter" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitSecurityFilters" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="SortDirection" type="SortDirectionEnum" minOccurs="0" />
    <xsd:element name="Width" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:int">
          <xsd:minInclusive value="0" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Alignment" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="General" />
          <xsd:enumeration value="Left" />
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Right" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
```

```
    <xsd:element name="Format" type="xsd:string" minOccurs="0" />
    <xsd:element name="MimeType" type="NonEmptyString" minOccurs="0" />
    <xsd:element name="DataCulture" type="xsd:language" minOccurs="0" />
    <xsd:element name="DiscourageGrouping" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ContextualName" type="AttributeContextualNameEnum"
                 minOccurs="0" />
    <xsd:element name="DefaultAggregateAttributeID" type="xsd:QName"
                 minOccurs="0" />
    <xsd:element name="ValueSelection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Dropdown" />
          <xsd:enumeration value="List" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Column" type="ColumnType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.16.1 Attribute.ID

The **Attribute.ID** attribute specifies an immutable globally unique identifier (GUID) of an attribute.

The **Attribute.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **Attribute.ID** attribute.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.16.2 Attribute.Alignment

The **Attribute.Alignment** element specifies the default character alignment for client applications displaying an attribute.

The **Attribute.Alignment** element is optional. If this element is present, its value MUST be a String that is one of the following:

General (default): Specifies that the client application displays instances of the attribute with a general alignment.

Left: Specifies that the client application displays instances of the attribute with a left alignment.

Center: Specifies that the client application displays instances of the attribute with a center alignment.

Right: Specifies that the client application displays instances of the attribute with a right alignment.

If the **Attribute.Alignment** element is not present, its value is interpreted as "General".

The following is the parent element of the **Attribute.Alignment** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Alignment** element.

```
<xsd:element name="Alignment" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="General" />
      <xsd:enumeration value="Left" />
      <xsd:enumeration value="Center" />
      <xsd:enumeration value="Right" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.16.3 Attribute.Column

The **Attribute.Column** element specifies a database object, which is a column in a table that represents an attribute and to which the attribute is bound.

The **Attribute.Column** element is optional. This element MUST be specified if the sibling Attribute.Expression element is not specified. The **Attribute.Column** element is of type Column. If the **Attribute.Column** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.Column** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Column** element.

```
<xsd:element name="Column" type="ColumnType" minOccurs="0" />
```

2.16.4 Attribute.ContextualName

The **Attribute.ContextualName** element specifies the means for a client application to generate a context-sensitive name for an attribute when the entity that it is an attribute of is reached via a role. The context-sensitive name is referred to as the contextual name of the attribute.

The **Attribute.ContextualName** element is optional. If this element is present, its value MUST be a String that is one of the following:

Attribute (default): Specifies that the client application uses the attribute name as the contextual name.

Merge: Specifies that the client application merges the attribute name and role name to generate a contextual name. For example, in the English language, merging the attribute name with the role name is done by prefixing the attribute name with the role name. Merge behavior can be different in other languages.

Role: Specifies that the client application uses the role name as the contextual name if the attribute is the single identifying attribute of the entity. Otherwise, the "Role" value specifies that the client application generates a contextual name the same as if "Merge" were specified.

If the **Attribute.ContextualName** element is not present, its value is interpreted as "Attribute".

The following is the parent element of the **Attribute.ContextualName** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.ContextualName** element.

```
<xsd:element name="ContextualName" type="AttributeContextualNameEnum"
             minOccurs="0" />
```

2.16.5 Attribute.CustomProperties

The **Attribute.CustomProperties** element specifies a collection of custom properties for the attribute.

The **Attribute.CustomProperties** element is optional. This element is of type CustomProperties. If the **Attribute.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.CustomProperties** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.16.6 Attribute.DataCulture

The **Attribute.DataCulture** element specifies the primary language or culture to use for an attribute for culture-sensitive operations that impact the meaning of data (such as formatting with a currency symbol). Note that this culture is not used for simple display formatting, such as for date and numeric formats.

The **Attribute.DataCulture** element is optional. If this element is present, its value MUST be a Language that is a valid language code string. If the **Attribute.DataCulture** element is not present, its value is interpreted as the value of the culture of the semantic model, which is the value of the SemanticModel.Culture element.

The following is the parent element of the **Attribute.DataCulture** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.DataCulture** element.

```
<xsd:element name="DataCulture" type="xsd:language" minOccurs="0" />
```

2.16.7 Attribute.DataType

The **Attribute.DataType** element specifies the data type of an attribute.

The **Attribute.DataType** element MUST be specified. Its value MUST be a String and MUST match the data type of the value of the sibling Attribute.Expression element if present.

Its value MUST be one of the following:

Boolean: Specifies that the data type of the attribute is the SMDL data type Boolean corresponding to the common language runtime (CLR) data type **Boolean**.

DateTime: Specifies that the data type of the attribute is the SMDL data type DateTime corresponding to the CLR data types **DateTime** and **DateTimeOffset**.

Time: Specifies that the data type of the attribute is the SMDL data type Time corresponding to the CLR data type **Timespan**.

Integer: Specifies that the data type of the attribute is the SMDL data type Integer corresponding to the CLR data types **Int16**, **Int32**, **Int64**, **UInt16**, **UInt32**, **Byte**, and **SByte**.

Decimal: Specifies that the data type of the attribute is the SMDL data type Decimal corresponding to the CLR data types **Decimal** and **UInt64**.

Float: Specifies that the data type of the attribute is the SMDL data type Float corresponding to the CLR data types **Single** and **Double**.

String: Specifies that the data type of the attribute is the SMDL data type **String** corresponding to the CLR data types **String**, **Char**, and **Guid**.

Binary: Specifies that the data type of the attribute is the SMDL data type Binary corresponding to the CLR data type **Byte[]**.

EntityKey: Specifies that the data type of the attribute is the SMDL data type EntityKey, which is a tuple of key column values for an entity and does not correspond to any CLR data type.

The following is the parent element of the **Attribute.DataType** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.DataType** element.

```
<xsd:element name="DataType" type="DataTypeEnum" />
```

2.16.8 Attribute.DefaultAggregateAttributeID

The **Attribute.DefaultAggregateAttributeID** element specifies the ID of an attribute to be used as the default aggregate for an attribute.

The **Attribute.DefaultAggregateAttributeID** element is optional. If this element is present, its value MUST be a QName. The value of the **Attribute.DefaultAggregateAttributeID** element MUST match the value of the Attribute.ID attribute of one of the attributes in the sibling Attribute.Variations

element collection. Additionally, the value of the Attribute.IsAggregate element on the default aggregate MUST be true.

The **Attribute.DefaultAggregateAttributeID** MUST NOT be specified if the sibling **Attribute.IsAggregate** element has a value of true. If the **Attribute.DefaultAggregateAttributeID** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.DefaultAggregateAttributeID** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.DefaultAggregateAttributeID** element.

```
<xsd:element name="DefaultAggregateAttributeID" type="xsd:QName" minOccurs="0" />
```

2.16.9 Attribute.Description

The **Attribute.Description** element specifies a textual description of an attribute.

The **Attribute.Description** element is optional. If this element is present, its value MUST be a String. If the **Attribute.Description** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.Description** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.16.10 Attribute.DiscourageGrouping

The **Attribute.DiscourageGrouping** element specifies that the client application is to discourage the user from grouping on an attribute. This element is typically set true on attributes with highly unique values, such as a phone number. If the attribute is a key or an identifying attribute of an entity, the client application groups on the entity's key columns instead.

The **Attribute.DiscourageGrouping** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that grouping on this attribute is discouraged. If the **Attribute.DiscourageGrouping** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.DiscourageGrouping** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.DiscourageGrouping** element.

```
<xsd:element name="DiscourageGrouping" type="xsd:boolean" minOccurs="0" />
```

2.16.11 Attribute.EnableDrillthrough

The **Attribute.EnableDrillthrough** element specifies that the client application is to provide a drillthrough link to the containing entity of the attribute.

The **Attribute.EnableDrillthrough** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the client application provides a drillthrough link. If the **Attribute.EnableDrillthrough** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.EnableDrillthrough** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.EnableDrillthrough** element.

```
<xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
```

2.16.12 Attribute.Expression

The **Attribute.Expression** element specifies an expression that is used to calculate the value of an attribute.

The **Attribute.Expression** element is optional. This element MUST be specified if the sibling Attribute.Column element is not specified. The **Attribute.Expression** element is of type Expression. If this element is present, the expression MUST return a scalar value and MUST NOT return a value that is a set. If the **Attribute.Expression** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.Expression** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
```

2.16.13 Attribute.Format

The **Attribute.Format** element specifies the default format specifier string with which to format the value of an attribute. The value of this element is either a valid standard format specifier [MSDN-FTYPES] or one of the extended format specifiers specified in the following table.

The **Attribute.Format** element is optional. If this element is present, its value MUST be a String whose value is a valid standard format specifier [MSDN-FTYPES] or one of the valid format specifiers specified in the following tables. If the **Attribute.Format** element is not present, its value is interpreted as NULL.

In addition to the standard format specifiers [MSDN-FTYPES], the following extended format specifiers are supported:

Category	Format specifier	Meaning
Date integers	ddd	Abbreviated name of day of week
	dddd	Full name of day of week
	MMM	Abbreviated name of month
	MMMM	Full name of month
Booleans	truefalse	True and false
	yesno	Yes and No

In the preceding table, a client application displays the values of day and month numbers based on the Gregorian calendar.

In the preceding and following format specifier tables, a client application displays the values True, False, Yes, and No as localized strings.

Custom format specifiers for attributes that have the Boolean data type are also supported. These take the form "truevalue;falsevalue" with the values of "truevalue" and "falsevalue" being one of the format specifiers specified in the following table.

Format specifier	Meaning
true	True
false	False
yes	Yes
no	No
" and '	Characters enclosed in single or double quotes are copied to the result string literally and do not affect formatting.
\	Escape character: The following character is taken to be a literal character, not a format specifier.

The following is the parent element of the **Attribute.Format** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Format** element.

```
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
```

2.16.14 Attribute.Hidden

The **Attribute.Hidden** element specifies that an attribute is not displayed to the user.

The **Attribute.Hidden** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the attribute is not displayed to the user. If the **Attribute.Hidden** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.Hidden** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

2.16.15 Attribute.IsAggregate

The **Attribute.IsAggregate** element specifies that an attribute is an aggregate that can be calculated in the context of not just the containing entity, but also in the context of any other entity for which there is a one-to-many relationship with the containing entity.

The **Attribute.IsAggregate** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the attribute is an aggregate. A value of false specifies that the attribute is treated as a scalar.

If the value of the **Attribute.IsAggregate** element is true, the value of the sibling Attribute.Expression element MUST be a non-anchored expression. If the **Attribute.IsAggregate** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.IsAggregate** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.IsAggregate** element.

```
<xsd:element name="IsAggregate" type="xsd:boolean" minOccurs="0" />
```

2.16.16 Attribute.IsFilter

The **Attribute.IsFilter** element specifies that an attribute is intended for use as a named filter.

The **Attribute.IsFilter** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the attribute is intended for use as a named filter. If the value of the **Attribute.IsFilter** element is true, the value of the sibling Attribute.DataType element MUST be Boolean. If the **Attribute.IsFilter** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.IsFilter** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.IsFilter** element.

```
<xsd:element name="IsFilter" type="xsd:boolean" minOccurs="0" />
```

2.16.17 Attribute.MimeType

The **Attribute.MimeType** element specifies the MIME type of the attribute.

The **Attribute.MimeType** element is optional. If this element is present, its value MUST be a nonempty MIMEType. If the **Attribute.MimeType** element is not present, its value is interpreted as NULL. The value of this element MUST be ignored if the value of the sibling Attribute.DataType element is not Boolean.

The following is the parent element of the **Attribute.MimeType** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.MimeType** element.

```
<xsd:element name="MimeType" type="NonEmptyString" minOccurs="0" />
```

2.16.18 Attribute.Name

The **Attribute.Name** element specifies the name of an attribute.

The **Attribute.Name** element MUST be specified. Its value MUST be a nonempty String. The value of this element MUST be a unique name within the namespace of the containing Entity element.

The following is the parent element of the **Attribute.Name** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
```

2.16.19 Attribute.Nullable

The **Attribute.Nullable** element specifies that an attribute can have a value of NULL. The value of the attribute is specified by the sibling Attribute.Expression element.

The **Attribute.Nullable** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the attribute can have a value of NULL. If the **Attribute.Nullable** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.Nullable** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Nullable** element.

```
<xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
```

2.16.20 Attribute.OmitSecurityFilters

The **Attribute.OmitSecurityFilters** element specifies that the expression for an attribute is calculated without adding security filters. A typical usage for this element is for cases where detail data is secured based on security filters but aggregate data is public.

The **Attribute.OmitSecurityFilters** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the expression for the attribute is calculated without adding security filters.

The **Attribute.OmitSecurityFilters** element MUST be ignored if the parent Attribute element does not have a child Attribute.Expression element. If the **Attribute.OmitSecurityFilters** element is not present, its value is interpreted as false.

The following is the parent element of the **Attribute.OmitSecurityFilters** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.OmitSecurityFilters** element.

```
<xsd:element name="OmitSecurityFilters" type="xsd:boolean" minOccurs="0" />
```

2.16.21 Attribute.SortDirection

The **Attribute.SortDirection** element specifies the default direction on which to sort an attribute.

The **Attribute.SortDirection** element is optional. If this element is present, its value MUST be a String that is one of the following:

None (default): Specifies that a client application does not sort the instance values of the attribute in any particular order when displaying the instance values.

Ascending: Specifies that a client application sorts the instance values of the attribute in ascending order when displaying the instance values.

Descending: Specifies that a client application sorts the instance values of the attribute in descending order when displaying the instance values.

If this element is not present, its value is interpreted as "None".

The following is the parent element of the **Attribute.SortDirection** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.SortDirection** element.

```
<xsd:element name="SortDirection" type="SortDirectionEnum" minOccurs="0" />
```

2.16.22 Attribute.ValueSelection

The **Attribute.ValueSelection** element specifies the client application behavior for selecting values of an attribute, based on the expected number of unique values.

The **Attribute.ValueSelection** element is optional. If this element is present, its value MUST be a String. The **Attribute.ValueSelection** element MUST be ignored if the value of the sibling Attribute.IsAggregate is true.

If the **Attribute.ValueSelection** element is present, its value MUST be one of the following:

None (default): Specifies that a selection UI is not provided. Users type values directly.

Dropdown: Specifies that the number of unique values is small enough to fit in a simple dropdown list.

List: Specifies that the number of unique values is too large for a simple dropdown list, requiring a larger list-based UI.

If the **Attribute.ValueSelection** element is not present, its value is interpreted as "None".

The following is the parent element of the **Attribute.ValueSelection** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.ValueSelection** element.

```
<xsd:element name="ValueSelection" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Dropdown" />
      <xsd:enumeration value="List" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.16.23 Attribute.Variations

The **Attribute.Variations** element specifies a collection of attributes and roles that are considered variations of the attribute. For example, Year, Month, and Day are considered variations of Date.

The **Attribute.Variations** element is optional. This element is of type Variations. The **Attribute.Variations** element MUST not be specified if the parent Attribute element is a variation of another field; that is, it is contained within the **Attribute.Variations** or Role.Variations of another field. If the **Attribute.Variations** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Attribute.Variations** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Variations** element.

```
<xsd:element name="Variations" type="VariationsType" minOccurs="0" />
```

2.16.24 Attribute.Width

The **Attribute.Width** element specifies the default column width in characters for displaying the attribute.

The **Attribute.Width** element is optional. If this element is present, its value MUST be an Integer whose value MUST be greater than 0.

If this element is not present, its value is interpreted as one of the following based on the value of the sibling Attribute.DataType element.

Value of Attribute.DataType	Default value of Attribute.Width
Boolean	6
DateTime	10
Time	8
Integer	8
Decimal	8
Float	8
String	20
Binary	1024
EntityKey	128

The following is the parent element of the **Attribute.Width** element.

Parent elements
Attribute

The following is the XML Schema definition of the **Attribute.Width** element.

```
<xsd:element name="Width" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:int">
      <xsd:minInclusive value="0" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.17 Variations

The **Variations** element specifies a collection of fields which are considered variations of the parent element, the containing field. For example, the attributes Year, Month, and Day are considered variations of Date.

The **Variations** element MUST contain at least one Variations.Attribute child element or one Variations.Role child element and can contain more.

The **Variations** element MUST NOT be specified if its parent field element is a variation of another field.

The following are the parent and child elements of the **Variations** element.

Parent elements
Attribute
Role

Child elements
Variations.Attribute
Variations.Role

The following is the XML Schema definition of the **Variations** element.

```
<xsd:complexType name="VariationsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Attribute" type="AttributeType" />
    <xsd:element name="Role" type="RoleType" />
  </xsd:choice>
</xsd:complexType>
```

2.17.1 Variations.Attribute

The **Variations.Attribute** element specifies an attribute that is a variation of the grandparent element, which is the containing field of the **Variations.Attribute** element.

The **Variations.Attribute** element is optional. This element is of type Attribute. At least one **Variations.Attribute** element or Variations.Role element MUST be specified in the parent Variations element collection and these elements can be specified more than once.

The **Variations.Attribute** element MUST NOT be specified if its grandparent element is a variation of another field.

The following is the parent element of the **Variations.Attribute** element.

Parent elements
Variations

The following is the XML Schema definition of the **Variations.Attribute** element.

```
<xsd:element name="Attribute" type="AttributeType" />
```

2.17.2 Variations.Role

The **Variations.Role** element specifies a role that is a variation of the grandparent element, which is the containing field of the **Variations.Role** element.

The **Variations.Role** element is optional. This element is of type Role. At least one Variations.Attribute element or **Variations.Role** element MUST be specified in the parent Variations element collection and these elements can be specified more than once.

The **Variations.Role** element MUST NOT be specified if its grandparent element is a variation of another field.

The following is the parent element of the **Variations.Role** element.

Parent elements
Variations

The following is the XML Schema definition of the **Variations.Role** element.

```
<xsd:element name="Role" type="RoleType" />
```

2.18 FieldFolder

The **FieldFolder** element specifies a folder that can contain both fields and further nested field folders.

The following are the parent elements, attributes, and child elements of the **FieldFolder** element.

Parent elements
Fields

Attributes
FieldFolder.ID

Child elements
FieldFolder.CustomProperties
FieldFolder.Description
FieldFolder.Fields
FieldFolder.Hidden
FieldFolder.Name

The following is the XML Schema definition of the **FieldFolder** element.

```
<xsd:complexType name="FieldFolderType">
  <xsd:all>
    <!-- ModelItemType Start-->
    <xsd:element name="Name" type="NonEmptyString" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
    <!-- ModelItemType End-->
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.18.1 FieldFolder.ID

The **FieldFolder.ID** attribute specifies an immutable globally unique identifier (GUID) of the parent FieldFolder element.

The **FieldFolder.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **FieldFolder.ID** attribute.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.18.2 FieldFolder.CustomProperties

The **FieldFolder.CustomProperties** element specifies a collection of custom properties for the parent FieldFolder element.

The **FieldFolder.CustomProperties** element is optional. This element is of type CustomProperties. If the **FieldFolder.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **FieldFolder.CustomProperties** element.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.18.3 FieldFolder.Description

The **FieldFolder.Description** element specifies a textual description of a field folder.

The **FieldFolder.Description** element is optional. If this element is present, its value MUST be a String. If this element is not present, its value is interpreted as NULL.

The following is the parent element of the **FieldFolder.Description** element.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.18.4 FieldFolder.Fields

The **FieldFolder.Fields** element specifies a collection of fields as an ordered list of attributes, roles, and field subfolders.

The **FieldFolder.Fields** element is optional. This element is of type Fields. If the **FieldFolder.Fields** element is not present, its value is interpreted as NULL.

The following is the parent element of the **FieldFolder.Fields** element.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.Fields** element.

```
<xsd:element name="Fields" type="FieldsType" minOccurs="0" />
```

2.18.5 FieldFolder.Hidden

The **FieldFolder.Hidden** element specifies that a folder is not to be displayed to the user.

The **FieldFolder.Hidden** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the folder is not displayed to the user. If this element is not present, its value is interpreted as false.

The following is the parent element of the **FieldFolder.Hidden** element.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

2.18.6 FieldFolder.Name

The **FieldFolder.Name** element specifies the name of a folder.

The **FieldFolder.Name** element MUST be specified. Its value MUST be a nonempty String. The value of this element MUST be a unique name among the names of all sibling FieldFolder, Attribute, and Role elements under the common Fields grandparent element.

The following is the parent element of the **FieldFolder.Name** element.

Parent elements
FieldFolder

The following is the XML Schema definition of the **FieldFolder.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" />
```

2.19 Role

The **Role** element specifies a role in the semantic model of an entity. The **Role** element specifies the relationship between the entity and a related entity.

The following are the parent elements, attributes, and child elements of the **Role** element.

Parent elements
Fields
Variations

Attributes
Role.ID

Child elements
Role.Cardinality
Role.ContextualName
Role.CustomProperties
Role.Description
Role.ExpandInline
Role.Hidden
Role.HiddenFields
Role.Linguistics
Role.Name
Role.Preferred
Role.PromoteLookup
Role.RelatedRoleID
Role.Relation
Role.Variations

The following is the XML Schema definition of the **Role** element.

```
<xsd:complexType name="RoleType">
  <xsd:all>
    <!-- ModelItemType Start-->
    <xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
    <!-- ModelItemType End-->
```

```
    <!-- FieldType Start-->
    <xsd:element name="Variations" type="VariationsType" minOccurs="0" />
    <!-- FieldType End-->
    <xsd:element name="Linguistics" type="LinguisticsType" minOccurs="0" />
    <xsd:element name="RelatedRoleID" type="xsd:QName" />
    <xsd:element name="Cardinality">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="One" />
          <xsd:enumeration value="Many" />
          <xsd:enumeration value="OptionalOne" />
          <xsd:enumeration value="OptionalMany" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ContextualName" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Default" />
          <xsd:enumeration value="Role" />
          <xsd:enumeration value="Merge" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="HiddenFields" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="FieldFolderItemID" type="xsd:QName"
                       maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Preferred" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PromoteLookup" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ExpandInline" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Relation" type="RelationType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.19.1 Role.ID

The **Role.ID** attribute specifies an immutable globally unique identifier (GUID) of a role.

The **Role.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **Role.ID** attribute.

Parent elements
Role

The following is the XML Schema definition of the **Role.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.19.2 Role.Cardinality

The **Role.Cardinality** element specifies the cardinality of the role.

The **Role.Cardinality** element MUST be specified. Its value MUST be a String that is one of the following:

One: Specifies that the cardinality of the role is 1.

Many: Specifies that the cardinality of the role is 1 or more.

OptionalOne: Specifies that the cardinality of the role is 0 or 1.

OptionalMany: Specifies that the cardinality of the role is 0 or more.

The following is the parent element of the **Role.Cardinality** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Cardinality** element.

```
<xsd:element name="Cardinality">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="One" />
      <xsd:enumeration value="Many" />
      <xsd:enumeration value="OptionalOne" />
      <xsd:enumeration value="OptionalMany" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.19.3 Role.ContextualName

The **Role.ContextualName** element specifies that a client application is to override the value of the Attribute.ContextualName element with the value of the **Role.ContextualName** element for attributes in the role's target entity when the role is used to reach the entity. This override occurs only when the value of the **Role.ContextualName** element is not "Default".

The **Role.ContextualName** element is optional. If this element is present, its value MUST be a String that is one of the following:

Default (default): Specifies that the client application does not override the value of the **Attribute.ContextualName** element of the attribute.

Role: Specifies that the client application overrides the value of the **Attribute.ContextualName** element of the attribute with the value "Role".

Merge: Specifies that the client application overrides the value of the **Attribute.ContextualName** element of the attribute with the value "Merge".

If the **Role.ContextualName** element is not present, its value is interpreted as "Default".

The following is the parent element of the **Role.ContextualName** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.ContextualName** element.

```
<xsd:element name="ContextualName" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Default" />
      <xsd:enumeration value="Role" />
      <xsd:enumeration value="Merge" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.19.4 Role.CustomProperties

The **Role.CustomProperties** element specifies a collection of custom properties for a role.

The **Role.CustomProperties** element is optional. This element is of type CustomProperties. If the **Role.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Role.CustomProperties** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.19.5 Role.Description

The **Role.Description** element specifies a textual description of a role.

The **Role.Description** element is optional. If this element is present, its value MUST be a String. If the **Role.Description** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Role.Description** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.19.6 Role.ExpandInline

The **Role.ExpandInline** element specifies that the client application is not to display the role to the user and that instead the Fields of the related entity are to be displayed as if they were part of the containing entity of the role.

The **Role.ExpandInline** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the client application does not display the role to the user. If the **Role.ExpandInline** element is not present, its value is interpreted as false.

This element MUST be ignored if the value of the sibling Role.Cardinality element is "Many" or "OptionalMany". This element MUST be ignored if this role is a variation of another field.

The following is the parent element of the **Role.ExpandInline** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.ExpandInline** element.

```
<xsd:element name="ExpandInline" type="xsd:boolean" minOccurs="0" />
```

2.19.7 Role.Hidden

The **Role.Hidden** element specifies that the role is not to be displayed to the user.

The **Role.Hidden** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the role is not displayed to the user. If the **Role.Hidden** element is not present, its value is interpreted as false.

The following is the parent element of the **Role.Hidden** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

2.19.8 Role.HiddenFields

The **Role.HiddenFields** element specifies a set of fields and field folders that the client application does not display for the role's target entity when the role is used to reach the entity. For example, the AddressType attribute is hidden when Address is displayed via the HomeAddress role.

The **Role.HiddenFields** element is optional. This element is of type HiddenFields. If the **Role.HiddenFields** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Role.HiddenFields** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.HiddenFields** element.

```
<xsd:element name="HiddenFields" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="FieldFolderItemID" type="xsd:QName"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
```

2.19.9 Role.Linguistics

The **Role.Linguistics** element specifies the linguistic properties of the role.

The **Role.Linguistics** element is optional. This element is of type Linguistics.

The **Role.Linguistics** element MUST NOT be specified if the sibling Role.Name element is not specified. In the case when the sibling **Role.Name** element is not specified, the values of the Linguistics.SingularName and Linguistics.PluralName elements are interpreted as the values of the Entity.Name and Entity.CollectionName elements of the related entity.

If the sibling **Role.Name** element is specified and the **Role.Linguistics** element is not present, the values of the **Linguistics.SingularName** and **Linguistics.PluralName** elements are interpreted as the value of the **Role.Name** element.

The following is the parent element of the **Role.Linguistics** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Linguistics** element.

```
<xsd:element name="Linguistics" type="LinguisticsType" minOccurs="0" />
```

2.19.10 Role.Name

The **Role.Name** element specifies the name of the role.

The **Role.Name** element is optional. If this element is present, its value MUST be a nonempty String. The value of this element MUST be a unique name within the namespace of the containing Entity element.

If the **Role.Name** element is not present, its value is interpreted as the value of the Entity.Name element of the related entity if the value of the sibling Role.Cardinality element is "One" or "OptionalOne".

If the **Role.Name** element is not present, its value is interpreted as the value of the Entity.CollectionName element of the related entity if the value of the sibling **Role.Cardinality** element is not "One" or "OptionalOne".

The following is the parent element of the **Role.Name** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
```

2.19.11 Role.Preferred

The **Role.Preferred** element specifies that paths that contain this role are preferred over paths that contain no preferred roles. This also applies to nonidentical subpaths of otherwise identical paths.

The **Role.Preferred** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that paths containing this role are preferred. If this element is not present, its value is interpreted as false.

The following is the parent element of the **Role.Preferred** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Preferred** element.

```
<xsd:element name="Preferred" type="xsd:boolean" minOccurs="0" />
```

2.19.12 Role.PromoteLookup

The **Role.PromoteLookup** element specifies that the client application promotes the target lookup entity in the UI to be a peer of the containing lookup entity. In other words, wherever the containing entity is displayed as a lookup, the client application also displays the target entity as a lookup.

The **Role.PromoteLookup** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the client application promotes the target lookup entity. If the **Role.PromoteLookup** element is not present, its value is interpreted as false.

If either the role's containing entity or the target entity are not lookup entities, the **Role.PromoteLookup** element MUST NOT be specified, or if this element is present, the value of the **Role.PromoteLookup** element MUST be false.

The following is the parent element of the **Role.PromoteLookup** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.PromoteLookup** element.

```
<xsd:element name="PromoteLookup" type="xsd:boolean" minOccurs="0" />
```

2.19.13 Role.RelatedRoleID

The **Role.RelatedRoleID** element specifies the ID of the corresponding role on the target entity. The corresponding role, also known as the related role, is a role that relates the same two entities as the original role, but relates them in the opposite order. The containing entity of the original role is the target entity of the related role and vice versa.

The **Role.RelatedRoleID** element MUST be specified. Its value MUST be a QName, which is the value of the Role.ID attribute for some other role different from this one.

The value of the **Role.RelatedRoleID** element and the value of the sibling **Role.ID** attribute MUST NOT be the same.

The value of the **Role.RelatedRoleID** element of the role on the target entity MUST be equal to the value of the **Role.ID** attribute for the parent Role element.

The following is the parent element of the **Role.RelatedRoleID** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.RelatedRoleID** element.

```
<xsd:element name="RelatedRoleID" type="xsd:QName" />
```

2.19.14 Role.Relation

The **Role.Relation** element specifies a relation in the underlying DSV that represents this role.

The **Role.Relation** element is optional. If this element is present, it is of type Relation. If the **Role.Relation** element is not present, its value is interpreted as the containment relationship between a column and its containing table as specified by the following condition.

The **Role.Relation** element cannot be specified if one of the containing Entity or the related **Entity** elements is bound to a table and the other **Entity** is bound to a column that is contained in that table. If this is not the case, the **Role.Relation** element MUST be specified. Note that the related **Entity** is defined to be the **Entity** that contains the related Role, which is the role referred to by the Role.RelatedRoleID element.

The **Role.Relation** element also imposes some restrictions on the binding of both the containing **Entity** and the related **Entity**. The end of the relation specified by the sibling **Role.RelationEnd** element MUST bind to columns in the table to which the related **Entity** is bound or equal the column to which that **Entity** is bound.

This restriction on roles applied to the related role implies that the reverse is also imposed; that is, the other end of the relation specified by the sibling **Role.RelationEnd** element MUST bind to columns in the table to which the containing **Entity** is bound or equal the column to which that **Entity** is bound.

The following is the parent element of the **Role.Relation** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Relation** element.

```
<xsd:element name="Relation" type="RelationType" minOccurs="0" />
```

2.19.15 Role.Variations

The **Role.Variations** element specifies a collection of attributes and role that are considered variations of the parent role. For example: Year, Month, and Day are considered variations of Date.

The **Role.Variations** element is optional. This element is of type Variations. The **Role.Variations** element MUST NOT be specified if the parent Role is a variation of another field; that is, it is contained within the Attribute.Variations or **Role.Variations** of another field. If the **Role.Variations** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Role.Variations** element.

Parent elements
Role

The following is the XML Schema definition of the **Role.Variations** element.

```
<xsd:element name="Variations" type="VariationsType" minOccurs="0" />
```

2.20 HiddenFields

The **HiddenFields** element specifies a collection of references to the fields and field folders that are not displayed for a role's target entity when the role is used to reach the entity.

The **HiddenFields** element MUST contain at least one HiddenFields.FieldFolderItemID element and can contain more.

The following are the parent and child elements of the **HiddenFields** element.

Parent elements
Role

Child elements
HiddenFields.FieldFolderItemID

The following is the XML Schema definition of the **HiddenFields** element.

```
<xsd:element name="HiddenFields" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="FieldFolderItemID" type="xsd:QName"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.20.1 HiddenFields.FieldFolderItemID

The **HiddenFields.FieldFolderItemID** element specifies the ID of an item that the client application does not display. The item is either a field or a field folder.

The **HiddenFields.FieldFolderItemID** element MUST be specified at least once in its parent HiddenFields collection and can occur more than once. The value of the **HiddenFields.FieldFolderItemID** element MUST be a QName. The value of the **HiddenFields.FieldFolderItemID** element MUST match the value of the Attribute.ID attribute, the Role.ID attribute, or the FieldFolder.ID attribute of one of the fields or field folders in the semantic model.

The following is the parent element of the **HiddenFields.FieldFolderItemID** element.

Parent elements
HiddenFields

The following is the XML Schema definition of the **HiddenFields.FieldFolderItemID** element.

```
<xsd:element name="FieldFolderItemID" type="xsd:QName" maxOccurs="unbounded" />
```

2.21 Linguistics

The **Linguistics** element specifies the linguistic properties of a role.

The following are the parent and child elements of the **Linguistics** element.

Parent elements
Role

Child elements
Linguistics.PluralName
Linguistics.SingularName

The following is the XML Schema definition of the **Linguistics** element.

```
<xsd:complexType name="LinguisticsType">
  <xsd:all>
    <xsd:element name="SingularName" type="NonEmptyString" minOccurs="0" />
    <xsd:element name="PluralName" type="NonEmptyString" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

2.21.1 Linguistics.PluralName

The **Linguistics.PluralName** element specifies the plural name of the role.

The **Linguistics.PluralName** element is optional. If this element is present, its value MUST be a nonempty String.

If the **Linguistics.PluralName** element is not present, and the parent sibling Role.Name element is present, the value of the **Linguistics.PluralName** element is interpreted as the value of the **Role.Name**element. If the **Linguistics.PluralName** element is not present, and the parent sibling **Role.Name** element is not present, the value of the **Linguistics.PluralName** element is interpreted as the value of the Entity.CollectionName element of the related entity specified by the grandparent Role element.

The following is the parent element of the **Linguistics.PluralName** element.

Parent elements
Linguistics

The following is the XML Schema definition of the **Linguistics.PluralName** element.

```
<xsd:element name="PluralName" type="NonEmptyString" minOccurs="0" />
```

2.21.2 Linguistics.SingularName

The **Linguistics.SingularName** element specifies the singular name of the role.

The **Linguistics.SingularName** element is optional. If this element is present, its value MUST be a nonempty String.

If the **Linguistics.SingularName** element is not present, and the parent sibling Role.Name element is present, the value of the **Linguistics.SingularName** element is interpreted as the value of the **Role.Name** element. If the **Linguistics.SingularName** element is not present, and the parent sibling **Role.Name** element is not present, the value of the **Linguistics.SingularName** element is interpreted as the value of the Entity.Name element of the related entity specified by the grandparent Role element.

The following is the parent element of the **Linguistics.SingularName** element.

Parent elements
Linguistics

The following is the XML Schema definition of the **Linguistics.SingularName** element.

```
<xsd:element name="SingularName" type="NonEmptyString" minOccurs="0" />
```

2.22 Relation

The **Relation** element specifies a relation to which a role or inheritance is bound.

The following are the parent elements and attributes of the **Relation** element.

Parent elements
Inheritance
Role

Attributes
Relation.Name
Relation.RelationEnd

The following is the XML Schema definition of the **Relation** element.

```
<xsd:complexType name="RelationType">
  <xsd:attribute name="Name" type="NonEmptyString" use="required" />
  <xsd:attribute name="RelationEnd" use="optional">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Source" />
```

```
          <xsd:enumeration value="Target" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
```

2.22.1 Relation.Name

The **Relation.Name** attribute specifies the name of the relation in the physical data source to which the grandparent role or inheritance is bound.

The **Relation.Name** attribute MUST be specified. Its value MUST be a nonempty String. The value MUST match the name of a relation in the physical data source.

If the grandparent of the **Relation.Name** attribute is an Inheritance element, the columns of both the source and target ends of the relation in the physical data source MUST comprise a unique key for the table associated with that end.<2>

The following is the parent element of the **Relation.Name** attribute.

Parent elements
Relation

The following is the XML Schema definition of the **Relation.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.22.2 Relation.RelationEnd

The **Relation.RelationEnd** attribute specifies the end of the relation in the physical data source to which the role is bound. A relation in the physical data source has a source and a target end.

The **Relation.RelationEnd** attribute is optional. If this attribute is present, its value MUST be a String. The **Relation.RelationEnd** attribute MUST be specified if the grandparent of this attribute is a Role element. The **Relation.RelationEnd** attribute MUST be ignored if the grandparent of this attribute is an Inheritance element, because the value of the **Relation.RelationEnd** attribute is assumed to be "Source" in this case.

If the grandparent of this attribute is a **Role** element and the value of its Role.Cardinality child element is "One" or "OptionalOne", the columns of the end of the relation in the physical data source that is specified by the value of the **Relation.RelationEnd** attribute MUST comprise a unique key for the table associated with that end.<3>

If this attribute is present, its value MUST be one of the following:

Source: Specifies that the **Role** is bound to the source end of the relation.

Target: Specifies that the **Role** is bound to the target end of the relation.

The following is the parent element of the **Relation.RelationEnd** attribute.

Parent elements
Relation

The following is the XML Schema definition of the **Relation.RelationEnd** attribute.

```
<xsd:attribute name="RelationEnd" use="optional">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Source" />
      <xsd:enumeration value="Target" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

2.23 IdentifyingAttributes

The **IdentifyingAttributes** element specifies a collection of references to the attributes used to identify an instance of the entity to the user.

The **IdentifyingAttributes** element specifies a collection of AttributeReference elements. The **IdentifyingAttributes** element MUST contain at least one IdentifyingAttributes.AttributeReference element and can contain more.

The following are the parent and child elements of the **IdentifyingAttributes** element.

Parent elements
Entity

Child elements
IdentifyingAttributes.AttributeReference

The following is the XML Schema definition of the **IdentifyingAttributes** element.

```
<xsd:complexType name="AttributeReferencesType">
  <xsd:sequence>
    <xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.23.1 IdentifyingAttributes.AttributeReference

The **IdentifyingAttributes.AttributeReference** element specifies a reference to an attribute used to identify an entity to the user.

The **IdentifyingAttributes.AttributeReference** element MUST be specified at least once in its parent IdentifyingAttributes collection and can occur more than once. The **IdentifyingAttributes.AttributeReference** element is of type AttributeReference.Path. The attribute that this references MUST NOT be of type Binary.

The following is the parent element of the **IdentifyingAttributes.AttributeReference** element.

Parent elements
IdentifyingAttributes

The following is the XML Schema definition of the **IdentifyingAttributes.AttributeReference** element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
```

2.24 Inheritance

The **Inheritance** element specifies how an entity inherits from another entity. It specifies both the other entity and a relation that represents this inheritance.

The following are the parent and child elements of the **Inheritance** element.

Parent elements
Entity

Child elements
Inheritance.InheritsFromEntityID
Inheritance.Relation

The following is the XML Schema definition of the **Inheritance** element.

```
<xsd:element name="Inheritance" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="InheritsFromEntityID" type="xsd:QName" />
      <xsd:element name="Relation" type="RelationType" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.24.1 Inheritance.InheritsFromEntityID

The **Inheritance.InheritsFromEntityID** element specifies the ID of an entity in the semantic model from which the grandparent Entity element inherits.

The **Inheritance.InheritsFromEntityID** element MUST be specified. Its value MUST be a QName. The value of the **Inheritance.InheritsFromEntityID** element MUST match the value of the Entity.ID attribute of one of the entities in the semantic model. The entity specified by the **Inheritance.InheritsFromEntityID** element MUST NOT directly or indirectly inherit from the grandparent **Entity** element.

The following is the parent element of the **Inheritance.InheritsFromEntityID** element.

Parent elements
Inheritance

The following is the XML Schema definition of the **Inheritance.InheritsFromEntityID** element.

```
<xsd:element name="InheritsFromEntityID" type="xsd:QName" />
```

2.24.2 Inheritance.Relation

The **Inheritance.Relation** element specifies the database relation that represents the inheritance of the grandparent Entity element from another entity.

The **Inheritance.Relation** element MUST be specified. This element is of type Relation. The database relation specified by the **Inheritance.Relation** element MUST relate to either (1) a unique key of the table or column to which the grandparent **Entity** element is bound, or (2) a unique key of the table or column to which the **Entity** specified by the value of the Inheritance.InheritsFromEntityID child element of the grandparent **Entity** element is bound.<4>

The following is the parent element of the **Inheritance.Relation** element.

Parent elements
Inheritance

The following is the XML Schema definition of the **Inheritance.Relation** element.

```
<xsd:element name="Relation" type="RelationType" minOccurs="0" />
```

2.25 SecurityFilters

The **SecurityFilters** element specifies a collection of references to attributes which are used as filters on the entity. These filters will restrict the rows from the entity that are available to users. When executing any query that references the entity, even indirectly through a role, all of these filters are added to the query and OR-ed together. Note that the collection of filters that are applied is subject to security-based model subsetting specified by the client application. If security filters are defined and none are available to the user, all instances of the entity are filtered out.

The **SecurityFilters** element specifies a collection of AttributeReference elements. The **SecurityFilters** element MUST contain at least one SecurityFilters.AttributeReference element and can contain more.

The following are the parent and child elements of the **SecurityFilters** element.

Parent elements
Entity

Child elements
SecurityFilters.AttributeReference

The following is the XML Schema definition of the **SecurityFilters** element.

```
<xsd:complexType name="AttributeReferencesType">
  <xsd:sequence>
    <xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.25.1 SecurityFilters.AttributeReference

The **SecurityFilters.AttributeReference** element specifies a reference to an attribute to apply as a filter to the grandparent Entity element.

The **SecurityFilters.AttributeReference** element MUST be specified at least once in its parent SecurityFilters collection and can occur more than once. The **SecurityFilters.AttributeReference** element is of type AttributeReference. The attribute that is referenced by the **SecurityFilters.AttributeReference** element MUST have its Attribute.IsFilter element set to true.

The following is the parent element of the **SecurityFilters.AttributeReference** element.

Parent elements
SecurityFilters

The following is the XML Schema definition of the **SecurityFilters.AttributeReference** element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType"
maxOccurs="unbounded" />
```

2.26 SortAttributes

The **SortAttributes** element specifies a collection of references to attributes on which to sort instances of the parent Entity element.

The **SortAttributes** element specifies a collection of SortAttribute elements. The **SortAttributes** element MUST contain at least one SortAttributes.SortAttribute element and can contain more.

The following are the parent and child elements of the **SortAttributes** element.

Parent elements
Entity

Child elements
SortAttributes.SortAttribute

The following is the XML Schema definition of the **SortAttributes** element.

```
<xsd:complexType name="SortAttributesType">
  <xsd:sequence>
    <xsd:element name="SortAttribute" type="SortAttributeType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.26.1 SortAttributes.SortAttribute

The **SortAttributes.SortAttribute** element specifies a reference to an attribute on which to sort instances of the grandparent Entity element.

The **SortAttributes.SortAttribute** element MUST be specified at least once in its parent SortAttributes collection and can occur more than once. The **SortAttributes.SortAttribute** element is of type SortAttribute. The attribute that is referenced by the **SortAttributes.SortAttribute** element MUST have its Attribute.IsAggregate element set to true.

The following is the parent element of the **SortAttributes.SortAttribute** element.

Parent elements
SortAttributes

The following is the XML Schema definition of the **SortAttributes.SortAttribute** element.

```
<xsd:element name="SortAttribute" type="SortAttributeType" maxOccurs="unbounded" />
```

2.27 SortAttribute

The **SortAttribute** element specifies a reference to an attribute on which to sort instances of the grandparent Entity element.

The following are the parent and child elements of the **SortAttribute** element.

Parent elements
SortAttributes

Child elements
SortAttribute.AttributeReference
SortAttribute.SortDirection

The following is the XML Schema definition of the **SortAttribute** element.

```
<xsd:complexType name="SortAttributeType">
  <xsd:all>
    <xsd:element name="AttributeReference" type="AttributeReferenceType" />
    <xsd:element name="SortDirection" type="SortAttributeDirectionEnum"
minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

2.27.1 SortAttribute.AttributeReference

The **SortAttribute.AttributeReference** element specifies a reference to an attribute on which to sort instances of the greatgrandparent Entity element.

The **SortAttribute.AttributeReference** element MUST be specified. This element is of type AttributeReference.

The following is the parent element of the **SortAttribute.AttributeReference** element.

Parent elements
SortAttribute

The following is the XML Schema definition of the SortAttribute.AttributeReference element.

```
<xsd:element name="AttributeReference" type="AttributeReferenceType" />
```

2.27.2 SortAttribute.SortDirection

The **SortAttribute.SortDirection** element specifies the direction on which to sort instances of the greatgrandparent Entity element by the referenced attribute when the entity is sorted

The **SortAttribute.SortDirection** element is optional. If this element is present, its value MUST be a String that is one of the following:

Ascending (default): Specifies that the instances of the entity are sorted in ascending order.

Descending: Specifies that the instances of the entity are sorted in descending order.

If this element is not present, its value is interpreted as "Ascending".

The following is the parent element of the **SortAttribute.SortDirection** element.

Parent elements
SortAttribute

The following is the XML Schema definition of the **SortAttribute.SortDirection** element.

```
<xsd:element name="SortDirection" type="SortAttributeDirectionEnum"
             minOccurs="0" />
```

2.28 Table

The **Table** element specifies the table, which is a database object to which an Entity is bound.

The following are the parent elements and attributes of the **Table** element.

Parent elements
Entity

Attributes
Table.Name

The following is the XML Schema definition of the **Table** element.

```
<xsd:complexType name="TableType">
  <xsd:attribute name="Name" type="NonEmptyString" use="required" />
</xsd:complexType>
```

2.28.1 Table.Name

The **Table.Name** attribute specifies the name of the table in the physical data source to which the entity is bound.

The **Table.Name** attribute MUST be specified. Its value MUST be a nonempty String that is the name of a table in the data source.

The following is the parent element of the **Table.Name** attribute.

Parent elements
Table

The following is the XML Schema definition of the **Table.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.29 EntityFolder

The **EntityFolder** element specifies a folder that contains entities and entity folders.

The following are the parent elements, attributes, and child elements of the **EntityFolder** element.

Parent elements
Entities

Attributes
EntityFolder.ID

Child elements
EntityFolder.CustomProperties
EntityFolder.Description
EntityFolder.Entities
EntityFolder.Hidden
EntityFolder.Name

The following is the XML Schema definition of the **EntityFolder** element.

```
<xsd:complexType name="EntityFolderType">
  <xsd:all>
    <!-- ModelItemType Start-->
    <xsd:element name="Name" type="NonEmptyString" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
```

```
      <!-- ModelItemType End-->
      <xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
  </xsd:complexType>
```

2.29.1 EntityFolder.ID

The **EntityFolder.ID** attribute specifies an immutable globally unique identifier (GUID) of the parent EntityFolder element.

The **EntityFolder.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **EntityFolder.ID** attribute.

Parent elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.29.2 EntityFolder.CustomProperties

The **EntityFolder.CustomProperties** element specifies a collection of custom properties for the parent EntityFolder element.

The **EntityFolder.CustomProperties** element is optional. This element is of type CustomProperties. If the **EntityFolder.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **EntityFolder.CustomProperties** element.

Parent elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.29.3 EntityFolder.Description

The **EntityFolder.Description** element specifies a textual description of a folder.

The **EntityFolder.Description** element is optional. If this element is present, its value MUST be a String. If this element is not present, its value is interpreted as NULL.

The following is the parent element of the **EntityFolder.Description** element.

Parent elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.29.4 EntityFolder.Entities

The **EntityFolder.Entities** element specifies a collection of entities as an ordered list of entities and entity subfolders.

The **EntityFolder.Entities** element is optional. This element is of type Entities. If the **EntityFolder.Entities** element is not present, its value is interpreted as NULL.

The following is the parent element of the **EntityFolder.Entities** element.

Parent elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.Entities** element.

```
<xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
```

2.29.5 EntityFolder.Hidden

The **EntityFolder.Hidden** element specifies that a folder is not to be displayed to the user.

The **EntityFolder.Hidden** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the folder is not displayed to the user. If the **EntityFolder.Hidden** element is not present, its value is interpreted as false.

The following is the parent element of the **EntityFolder.Hidden** element.

Parent Elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

2.29.6 EntityFolder.Name

The **EntityFolder.Name** element specifies the name of a folder.

The **EntityFolder.Name** element MUST be specified. Its value MUST be a nonempty String. The value of this element MUST be a unique name among the names of all sibling EntityFolder and Entity elements under the common Entities grandparent element.

The following is the parent element of the **EntityFolder.Name** element.

Parent elements
EntityFolder

The following is the XML Schema definition of the **EntityFolder.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" />
```

2.30 Perspectives

The **Perspectives** element specifies a collection of perspectives of the semantic model.

The **Perspectives** element MUST contain at least one Perspectives.Perspective child element and can contain more.

The following are the parent and child elements of the **Perspectives** element.

Parent elements
SemanticModel

Child elements
Perspectives.Perspective

The following is the XML Schema definition of the **Perspectives** element.

```
<xsd:complexType name="PerspectivesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Perspective" type="PerspectiveType" />
  </xsd:choice>
</xsd:complexType>
```

2.30.1 Perspectives.Perspective

The **Perspectives.Perspective** element specifies a perspective of the semantic model.

The **Perspectives.Perspective** element MUST be specified at least once in its parent Perspectives collection and can occur more than once. The **Perspectives.Perspective** element is of type Perspective.

The following is the parent element of the **Perspectives.Perspective** element.

Parent elements
Perspectives

The following is the XML Schema definition of the **Perspectives.Perspective** element.

```
<xsd:element name="Perspective" type="PerspectiveType" />
```

2.31 Perspective

The **Perspective** element specifies a subset of the semantic model to provide to users as a submodel.

The following are the parent elements, attributes, and child elements of the **Perspective** element.

Parent elements
Perspectives

Attributes
Perspective.ID

Child elements
Perspective.CustomProperties
Perspective.Description
Perspective.ModelItems
Perspective.Name

The following is the XML Schema definition of the **Perspective** element.

```
<xsd:complexType name="PerspectiveType">
  <xsd:all>
    <xsd:element name="Name" type="NonEmptyString" />
    <xsd:element name="Description" type="xsd:string" minOccurs="0" />
    <xsd:element name="ModelItems" type="ModelItemsType" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
                 minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
```

2.31.1 Perspective.ID

The **Perspective.ID** attribute specifies an immutable globally unique identifier (GUID) of a perspective.

The **Perspective.ID** attribute MUST be specified. The value of this attribute MUST be a QName that is globally unique for all the objects in the semantic model.

The following is the parent element of the **Perspective.ID** attribute.

Parent elements
Perspective

The following is the XML Schema definition of the **Perspective.ID** attribute.

```
<xsd:attribute name="ID" type="xsd:QName" use="required" />
```

2.31.2 Perspective.CustomProperties

The **Perspective.CustomProperties** element specifies a collection of custom properties for a perspective.

The **Perspective.CustomProperties** element is optional. This element is of type CustomProperties. If the **Perspective.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Perspective.CustomProperties** element.

Parent elements
Perspective

The following is the XML Schema definition of the **Perspective.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.31.3 Perspective.Description

The **Perspective.Description** element specifies a textual description of a perspective.

The **Perspective.Description** element is optional. If this element is present, its value MUST be a String. If this element is not present, its value is interpreted as NULL.

The following is the parent element of the **Perspective.Description** element.

Parent elements
Perspective

The following is the XML Schema definition of the **Perspective.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

2.31.4 Perspective.ModelItems

The **Perspective.ModelItems** element specifies a collection of references to items in the semantic model to include in the perspective. The items being referenced include Entity, Attribute, Role, EntityFolder, and FieldFolder elements.

The **Perspective.ModelItems** element MUST be specified. This element is of type ModelItems.

The following is the parent element of the **Perspective.ModelItems** element.

Parent elements
Perspective

The following is the XML Schema definition of the **Perspective.ModelItems** element.

```
<xsd:element name="ModelItems" type="ModelItemsType" />
```

2.31.5 Perspective.Name

The **Perspective.Name** element specifies the name of the perspective.

The **Perspective.Name** element MUST be specified. Its value MUST be a nonempty String. The value of the **Perspective.Name** element MUST be a unique name within the collection of perspectives contained in the semantic model.

The following is the parent element of the **Perspective.Name** element.

Parent elements
Perspective

The following is the XML Schema definition of the **Perspective.Name** element.

```
<xsd:element name="Name" type="NonEmptyString" />
```

2.32 ModelItems

The **ModelItems** element specifies a collection of references to items in the semantic model that are included in the perspective.

The **ModelItems** element specifies a collection of ModelItems.ModelItemID elements. The **ModelItems** element MUST contain at least one **ModelItems.ModelItemID** element and can contain more.

The following are the parent and child elements of the **ModelItems** element.

Parent elements
Perspective

Child elements
ModelItems.ModelItemID

The following is the XML Schema definition of the **ModelItems** element.

```
<xsd:complexType name="ModelItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ModelItemID" type="xsd:QName" />
  </xsd:choice>
</xsd:complexType>
```

2.32.1 ModelItems.ModelItemID

The **ModelItems.ModelItemID** element specifies the ID of an item in the semantic model to include in the perspective.

The **ModelItems.ModelItemID** element MUST be specified. The value of this element MUST be a QName. The value of the **ModelItems.ModelItemID** element MUST match the value of the **ID** attribute of one of the model items in the semantic model; that is, it MUST match the value of one of the following attributes in the model:

- Entity.ID

- EntityFolder.ID
- Attribute.ID
- Role.ID
- FieldFolder.ID

The following is the parent element of the **ModelItems.ModelItemID** element.

Parent elements
ModelItems

The following is the XML Schema definition of the **ModelItems.ModelItemID** element.

```
<xsd:element name="ModelItemID" type="xsd:QName" />
```

2.33 SemanticQuery

The **SemanticQuery** element specifies a semantic query. This element can be used as the root node of an XML tree in the SMDL namespace. In this case, the XML represents a Semantic Model Query Language (SMQL) query.

The following are the child elements of the **SemanticQuery** element.

Child elements
SemanticQuery.CalculatedAttributes
SemanticQuery.CustomProperties
SemanticQuery.EnableDrillthrough
SemanticQuery.Hierarchies
SemanticQuery.MeasureGroups
SemanticQuery.Parameters

The following is the XML Schema definition of the **SemanticQuery** element.

```
<xsd:element name="SemanticQuery">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Hierarchies" type="HierarchiesType" minOccurs="0" />
      <xsd:element name="MeasureGroups" type="MeasureGroupsType" minOccurs="0" />
      <xsd:element name="CalculatedAttributes" type="ExpressionsType"
                   minOccurs="0" />
      <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
      <xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType"
                   minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.33.1 SemanticQuery.CalculatedAttributes

The **SemanticQuery.CalculatedAttributes** element specifies a collection of expressions that can be used throughout the query.

The **SemanticQuery.CalculatedAttributes** element is optional. This element is of type CalculatedAttributes. If the **SemanticQuery.CalculatedAttributes** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticQuery.CalculatedAttributes** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.CalculatedAttributes** element.

```
<xsd:element name="CalculatedAttributes" type="ExpressionsType" minOccurs="0" />
```

2.33.2 SemanticQuery.CustomProperties

The **SemanticQuery.CustomProperties** element specifies a collection of custom properties for a query.

The **SemanticQuery.CustomProperties** element is optional. This element is of type CustomProperties. If the **SemanticQuery.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticQuery.CustomProperties** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.33.3 SemanticQuery.EnableDrillthrough

The **SemanticQuery.EnableDrillthrough** element specifies whether the query SHOULD have optional **DrillthroughContext** and **DrillthroughSourceQuery** parameters that identify additional filter information for the query.

The **SemanticQuery.EnableDrillthrough** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that the query is to have optional DrillthroughContext and **DrillthroughSourceQuery** parameters. If the **SemanticQuery.EnableDrillthrough** element is not present, its value is interpreted as false.

The following is the parent element of the **SemanticQuery.EnableDrillthrough** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.EnableDrillthrough** element.

```
<xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
```

2.33.4 SemanticQuery.Hierarchies

The **SemanticQuery.Hierarchies** element specifies a collection of grouping hierarchies for the query.

The **SemanticQuery.Hierarchies** element is optional. This element is of type Hierarchies. If the **SemanticQuery.Hierarchies** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticQuery.Hierarchies** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.Hierarchies** element.

```
<xsd:element name="Hierarchies" type="HierarchiesType" minOccurs="0" />
```

2.33.5 SemanticQuery.MeasureGroups

The **SemanticQuery.MeasureGroups** element specifies a collection of measures for a client application to display.

The **SemanticQuery.MeasureGroups** element is optional. This element MUST be specified if the sibling **SemanticQuery.Hierarchies** element is not specified or if there is no **Hierarchies.Hierarchy** child element of the sibling **SemanticQuery.Hierarchies** element that contains a **Hierarchies.Groupings** child element.

The **SemanticQuery.MeasureGroups** element is of type MeasureGroups. If the **SemanticQuery.MeasureGroups** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticQuery.MeasureGroups** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.MeasureGroups** element.

```
<xsd:element name="MeasureGroups" type="MeasureGroupsType" minOccurs="0" />
```

2.33.6 SemanticQuery.Parameters

The **SemanticQuery.Parameters** element specifies a collection of parameters to the query.

The **SemanticQuery.Parameters** element is optional. This element is of type Parameters. If the **SemanticQuery.Parameters** element is not present, its value is interpreted as NULL.

The following is the parent element of the **SemanticQuery.Parameters** element.

Parent elements
SemanticQuery

The following is the XML Schema definition of the **SemanticQuery.Parameters** element.

```
<xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
```

2.34 CalculatedAttributes

The **CalculatedAttributes** element specifies a collection of expressions that can be used throughout the query.

The **CalculatedAttributes** element MUST contain at least one CalculatedAttributes.Expression element and can contain more.

The following are the parent and child elements of the **CalculatedAttributes** element.

Parent elements
SemanticQuery

Child elements
CalculatedAttributes.Expression

The following is the XML Schema definition of the **CalculatedAttributes** element.

```
<xsd:complexType name="ExpressionsType">
  <xsd:sequence>
    <xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.34.1 CalculatedAttributes.Expression

The **CalculatedAttributes.Expression** element specifies an expression to use in the query.

The **CalculatedAttributes.Expression** element MUST be specified at least once in its parent CalculatedAttributes collection and can occur more than once. The **CalculatedAttributes.Expression** element is of type Expression.

The following is the parent element of the **CalculatedAttributes.Expression** element.

Parent elements
CalculatedAttributes

The following is the XML Schema definition of the **CalculatedAttributes.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
```

2.35 Hierarchies

The **Hierarchies** element specifies a collection of grouping hierarchies for a query.

The **Hierarchies** element MUST contain exactly one Hierarchies.Hierarchy element.

The following are the parent and child elements of the **Hierarchies** element.

Parent elements
SemanticQuery

Child elements
Hierarchies.Hierarchy

The following is the XML Schema definition of the **Hierarchies** element.

```
<xsd:complexType name="HierarchiesType">
  <xsd:sequence>
    <xsd:element name="Hierarchy" type="HierarchyType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.35.1 Hierarchies.Hierarchy

The **Hierarchies.Hierarchy** element specifies a grouping hierarchy for the query.

The **Hierarchies.Hierarchy** element MUST be specified exactly once in its parent Hierarchies collection. The **Hierarchies.Hierarchy** element is of type Hierarchy.

The following is the parent element of the **Hierarchies.Hierarchy** element.

Parent elements
Hierarchies

The following is the XML Schema definition of the **Hierarchies.Hierarchy** element.

```
<xsd:element name="Hierarchy" type="HierarchyType" maxOccurs="unbounded" />
```

2.36 Hierarchy

The **Hierarchy** element specifies a grouping hierarchy for a query.

The following are the parent and child elements of the **Hierarchy** element.

Parent elements
Hierarchies

Child elements
Hierarchy.BaseEntity
Hierarchy.Filter
Hierarchy.Groupings

The following is the XML Schema definition of the **Hierarchy** element.

```
<xsd:complexType name="HierarchyType">
  <xsd:all>
    <xsd:element name="BaseEntity" type="BaseEntityType" />
    <xsd:element name="Groupings" type="GroupingsType" minOccurs="0" />
    <xsd:element name="Filter" minOccurs="0">
      <xsd:complexType>
        <xsd:all>
          <xsd:element name="Expression" type="ExpressionType" />
        </xsd:all>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
```

2.36.1 Hierarchy.BaseEntity

The **Hierarchy.BaseEntity** element specifies the base entity for the hierarchy. Each instance of the entity generates one detail row to be grouped in the hierarchy except for those instances that are filtered out.

The **Hierarchy.BaseEntity** element MUST be specified. This element is of type BaseEntity.

The following is the parent element of the **Hierarchy.BaseEntity** element.

Parent elements
Hierarchy

The following is the XML Schema definition of the **Hierarchy.BaseEntity** element.

```
<xsd:element name="BaseEntity" type="BaseEntityType" />
```

2.36.2 Hierarchy.Filter

The **Hierarchy.Filter** element specifies a filter to apply to the base entity for detail row filtering.

The **Hierarchy.Filter** element is optional. This element is of type Filter. If the **Hierarchy.Filter** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Hierarchy.Filter** element.

Parent elements
Hierarchy

The following is the XML Schema definition of the **Hierarchy.Filter** element.

```
<xsd:element name="Filter" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Expression" type="ExpressionType" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.36.3 Hierarchy.Groupings

The **Hierarchy.Groupings** element specifies a collection that is an ordered list of groupings for the hierarchy.

The **Hierarchy.Groupings** element is optional. This element is of type Groupings. If the **Hierarchy.Groupings** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Hierarchy.Groupings** element.

Parent elements
Hierarchy

The following is the XML Schema definition of the **Hierarchy.Groupings** element.

```
<xsd:element name="Groupings" type="GroupingsType" minOccurs="0" />
```

2.37 BaseEntity

The **BaseEntity** element specifies the base entity for a Hierarchy or MeasureGroup.

The following are the parent and child elements of the **BaseEntity** element.

Parent elements
Hierarchy
MeasureGroup

Child elements
BaseEntity.EntityID

The following is the XML Schema definition of the **BaseEntity** element.

```
<xsd:complexType name="BaseEntityType">
  <xsd:all>
    <xsd:element name="EntityID" type="xsd:QName" />
  </xsd:all>
</xsd:complexType>
```

2.37.1 BaseEntity.EntityID

The **BaseEntity.EntityID** element specifies the ID of the base entity.

The **BaseEntity.EntityID** element MUST be specified. Its value MUST be a QName. The value of this element MUST match the value of the Entity.ID attribute of one of the entities in the semantic model.

The following is the parent element of the **BaseEntity.EntityID** element.

Parent elements
BaseEntity

The following is the XML Schema definition of the **BaseEntity.EntityID** element.

```
<xsd:element name="EntityID" type="xsd:QName" />
```

2.38 Filter

The **Filter** element specifies the definition of a filter used in the query.

The following are the parent and child elements of the **Filter** element.

Parent elements
Hierarchy

Child elements
Filter.Expression

The following is the XML Schema definition of the **Filter** element.

```
<xsd:element name="Filter" minOccurs="0">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Expression" type="ExpressionType" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.38.1 Filter.Expression

The **Filter.Expression** element specifies a filter expression to apply to the query.

The **Filter.Expression** element MUST be specified. This element is of type Expression. The data type of the expression MUST be Boolean. If the **Expression** evaluates to true for a given entity instance, the instance will appear in the result set of the query, otherwise not.

The following is the parent element of the **Filter.Expression** element.

Parent elements
Filter

The following is the XML Schema definition of the **Filter.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" />
```

2.39 Groupings

The **Groupings** element specifies a collection of groupings for a hierarchy.

The **Groupings** element MUST contain at least one Groupings.Grouping element and can contain more.

The following are the parent and child elements of the **Groupings** element.

Parent elements
Hierarchy

Child elements
Groupings.Grouping

The following is the XML Schema definition of the **Groupings** element.

```
<xsd:complexType name="GroupingsType">
  <xsd:sequence>
    <xsd:element name="Grouping" type="GroupingType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.39.1 Groupings.Grouping

The **Groupings.Grouping** element specifies a grouping for a hierarchy.

The **Groupings.Grouping** element MUST be specified at least once in its parent Groupings collection and can occur more than once. The **Groupings.Grouping** element is of type Grouping.

The following is the parent element of the **Groupings.Grouping** element.

Parent elements
Groupings

The following is the XML Schema definition of the **Groupings.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" maxOccurs="unbounded" />
```

2.40 Grouping

The **Grouping** element specifies an expression by which to group the data in a hierarchy.

The following are the parent elements, attributes, and child elements of the **Grouping** element.

Parent elements
Groupings

Attributes
Grouping.Name

Child elements
Grouping.Details
Grouping.Expression

The following is the XML Schema definition of the **Grouping** element.

```
<xsd:complexType name="GroupingType">
  <xsd:all>
    <xsd:element name="Expression" type="ExpressionType" />
    <xsd:element name="Details" type="ExpressionsType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="Name" type="NonEmptyString" use="required" />
</xsd:complexType>
```

2.40.1 Grouping.Name

The **Grouping.Name** attribute specifies the name of the grouping.

The **Grouping.Name** attribute MUST be specified. Its value MUST be a nonempty String. The value of this attribute MUST be a unique name within all the groupings in the query.

The following is the parent element of the **Grouping.Name** attribute.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.40.2 Grouping.Details

The **Grouping.Details** element specifies a collection of expressions to display for the grouping.

The **Grouping.Details** element is optional. This element MUST NOT be specified if the sibling Grouping.Expression element expression is not an EntityRef object.

The **Grouping.Details** element is of type Details. If the **Grouping.Details** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Grouping.Details** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Details** element.

```
<xsd:element name="Details" type="ExpressionsType" minOccurs="0" />
```

2.40.3 Grouping.Expression

The **Grouping.Expression** element specifies an expression by which to group the data.

The **Grouping.Expression** element MUST be specified. This element is of type Expression. The expression cannot have a Binary data type.

The following is the parent element of the **Grouping.Expression** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" />
```

2.41 Details

The **Details** element specifies a collection of detail expressions to include in the results of the query. Detail expressions return additional data about the parent Grouping element.

The **Details** element specifies a collection of Expression elements. This element MUST contain at least one Details.Expression element and can contain more.

The following are the parent and child elements of the **Details** element.

Parent elements
Grouping

Child elements
Details.Expression

The following is the XML Schema definition of the **Details** element.

```
<xsd:complexType name="ExpressionsType">
  <xsd:sequence>
    <xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.41.1 Details.Expression

The **Details.Expression** element specifies an expression to include in the results of the query.

The **Details.Expression** element MUST be specified at least once in its parent Details collection and can occur more than once. The **Details.Expression** element is of type Expression.

The following is the parent element of the **Details.Expression** element.

Parent elements
Details

The following is the XML Schema definition of the **Details.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
```

2.42 MeasureGroups

The **MeasureGroups** element specifies collections of detail expressions to include in the results of the query.

The **MeasureGroups** element specifies a collection of MeasureGroup elements. The **MeasureGroups** element MUST contain exactly one MeasureGroups.MeasureGroup element.

The following are the parent and child elements of the **MeasureGroups** element.

Parent elements
SemanticQuery

Child elements
MeasureGroups.MeasureGroup

The following is the XML Schema definition of the **MeasureGroups** element.

```
<xsd:complexType name="MeasureGroupsType">
  <xsd:sequence>
    <xsd:element name="MeasureGroup" type="MeasureGroupType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.42.1 MeasureGroups.MeasureGroup

The **MeasureGroup.MeasureGroup** element specifies a collection of detail expressions to include in the results of the query.

The **MeasureGroup.MeasureGroup** element MUST be specified exactly once in its parent MeasureGroups collection. The **MeasureGroup.MeasureGroup** element is of type MeasureGroup.

The following is the parent element of the **MeasureGroup.MeasureGroup** element.

Parent elements
MeasureGroups

The following is the XML Schema definition of the **MeasureGroup.MeasureGroup** element.

```
<xsd:element name="MeasureGroup" type="MeasureGroupType" maxOccurs="unbounded" />
```

2.43 MeasureGroup

The **MeasureGroup** element specifies a collection of related aggregation expressions to include in the results of the query. Each of these expressions is derived from the same base entity specified by the child MeasureGroup.BaseEntity element.

The following are the parent and child elements of the **MeasureGroup** element.

Parent elements
MeasureGroups

Child elements
MeasureGroup.BaseEntity
MeasureGroup.Measures
MeasureGroup.SubtotalSets

The following is the XML Schema definition of the **MeasureGroup** element.

```
<xsd:complexType name="MeasureGroupType">
  <xsd:all>
    <xsd:element name="BaseEntity" type="BaseEntityType" />
    <xsd:element name="Measures" type="ExpressionsType" />
    <xsd:element name="SubtotalSets" type="SubtotalSetsType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

2.43.1 MeasureGroup.BaseEntity

The **MeasureGroup.BaseEntity** element specifies the base entity for the measure group.

The **MeasureGroup.BaseEntity** element MUST be specified. This element is of type BaseEntity. If the ancestor SemanticQuery element has a Hierarchies.Hierarchy child element, the **MeasureGroup.BaseEntity** element MUST refer to the same Entity element that the Hierarchy.BaseEntity element refers to. Every element in the sibling Measures element collection MUST be reachable via paths from the **MeasureGroup.BaseEntity** element.

The following is the parent element of the **MeasureGroup.BaseEntity** element.

Parent elements
MeasureGroup

The following is the XML Schema definition of the **MeasureGroup.BaseEntity** element.

```
<xsd:element name="BaseEntity" type="BaseEntityType" />
```

2.43.2 MeasureGroup.Measures

The **MeasureGroup.Measures** element specifies a collection of measure expressions to display for this measure group.

The **MeasureGroup.Measures** element MUST be specified. This element is of type Measures.

The following is the parent element of the **MeasureGroup.Measures** element.

Parent elements
MeasureGroup

The following is the XML Schema definition of the **MeasureGroup.Measures** element.

```
<xsd:element name="Measures" type="ExpressionsType" />
```

2.43.3 MeasureGroup.SubtotalSets

The **MeasureGroup.SubtotalSets** element specifies a collection of subtotals to calculate for measures in the measure group.

The **MeasureGroup.SubtotalSets** element is optional. This element is of type SubtotalSets. If the **MeasureGroup.SubtotalSets** element is not present, its value is interpreted as NULL.

The following is the parent element of the **MeasureGroup.SubtotalSets** element.

Parent elements
MeasureGroup

The following is the XML Schema definition of the **MeasureGroup.SubtotalSets** element.

```
<xsd:element name="SubtotalSets" type="SubtotalSetsType" minOccurs="0" />
```

2.44 Measures

The **Measures** element specifies a collection of measure expressions in a measure group.

The **Measures** element specifies a collection of Expression elements. The **Measures** element MUST contain at least one Measures.Expression element and can contain more.

The following are the parent and child elements of the **Measures** element.

Parent elements
MeasureGroup

Child elements
Measures.Expression

The following is the XML Schema definition of the **Measures** element.

```
<xsd:complexType name="ExpressionsType">
  <xsd:sequence>
    <xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.44.1 Measures.Expression

The **Measures.Expression** element specifies an expression to include in the measure group.

The **Measures.Expression** element MUST be specified at least once in its parent Measures collection and can occur more than once. The **Measures.Expression** element is of type Expression. Every **Measures.Expression** element in the parent **Measures** element collection MUST be reachable via paths from the MeasureGroup.BaseEntity child element of the grandparent MeasureGroup element.

The following is the parent element of the **Measures.Expression** element.

Parent elements
Measures

The following is the XML Schema definition of the **Measures.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
```

2.45 SubtotalSets

The **SubtotalSets** element specifies collections of subtotals to calculate for measures within a measure group.

The **SubtotalSets** element specifies a collection of SubtotalSet elements. The **SubtotalSets** element MUST contain at least one SubtotalSets.SubtotalSet element and can contain more.

The following are the parent and child elements of the **SubtotalSets** element.

Parent elements
MeasureGroup

Child elements
SubtotalSets.SubtotalSet

The following is the XML Schema definition of the **SubtotalSets** element.

```
<xsd:complexType name="SubtotalSetsType">
  <xsd:sequence>
```

```
    <xsd:element name="SubtotalSet" type="SubtotalSetType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.45.1 SubtotalSets.SubtotalSet

The **SubtotalSets.SubtotalSet** element specifies a collection of subtotals to calculate for a set of measures in the measure group.

The **SubtotalSets.SubtotalSet** element MUST be specified at least once in its parent SubtotalSets collection and can occur more than once. The **SubtotalSets.SubtotalSet** element is of type SubtotalSet.

The following is the parent element of the **SubtotalSets.SubtotalSet** element.

Parent elements
SubtotalSets

The following is the XML Schema definition of the **SubtotalSets.SubtotalSet** element.

```
<xsd:element name="SubtotalSet" type="SubtotalSetType" maxOccurs="unbounded" />
```

2.46 SubtotalSet

The **SubtotalSet** element specifies a collection of subtotals to calculate for measures in a measure group.

The following are the parent and child elements of the **SubtotalSet** element.

Parent elements
SubtotalSets

Child elements
SubtotalSet.SubtotalGroupings
SubtotalSet.SubtotalMeasures

The following is the XML Schema definition of the **SubtotalSet** element.

```
<xsd:complexType name="SubtotalSetType">
  <xsd:all>
    <xsd:element name="SubtotalGroupings" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="GroupingName" type="xsd:string"
                       maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="SubtotalMeasures" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
```

```
            <xsd:element name="MeasureName" type="xsd:string"
                         maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>
```

2.46.1 SubtotalSet.SubtotalGroupings

The **SubtotalSet.SubtotalGroupings** element specifies a collection of groupings by which to calculate subtotals.

The **SubtotalSet.SubtotalGroupings** element is optional. This element is of type SubtotalGroupings. If the **SubtotalSet.SubtotalGroupings** element is not present, its value is interpreted as NULL and a grand total is calculated by the client application.

The following is the parent element of the **SubtotalSet.SubtotalGroupings** element.

Parent elements
SubtotalSet

The following is the XML Schema definition of the **SubtotalSet.SubtotalGroupings** element.

```
<xsd:element name="SubtotalGroupings" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="GroupingName" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.46.2 SubtotalSet.SubtotalMeasures

The **SubtotalSet.SubtotalMeasures** element specifies a collection of measures to subtotal by the groups.

The **SubtotalSet.SubtotalMeasures** element is optional. This element is of type SubtotalMeasures. If the **SubtotalSet.SubtotalMeasures** element is not present, its value is interpreted as NULL and all measures in the measure group are subtotaled.

The following is the parent element of the **SubtotalSet.SubtotalMeasures** element.

Parent elements
SubtotalSet

The following is the XML Schema definition of the **SubtotalSet.SubtotalMeasures** element.

```
<xsd:element name="SubtotalMeasures" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="MeasureName" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.47 SubtotalGroupings

The **SubtotalGroupings** element specifies a collection of groupings by which to calculate subtotals.

The following are the parent and child elements of the **SubtotalGroupings** element.

Parent elements
SubtotalSet

Child elements
SubtotalGroupings.GroupingName

The following is the XML Schema definition of the **SubtotalGroupings** element.

```
<xsd:element name="SubtotalGroupings" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="GroupingName" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.47.1 SubtotalGroupings.GroupingName

The **SubtotalGroupings.GroupingName** element specifies the name of a grouping in the query.

The **SubtotalGroupings.GroupingName** element MUST be specified at least once in its parent SubtotalGroupings collection and can occur more than once. The value of the **SubtotalGroupings.GroupingName** element MUST be a String and MUST be equal to the value of a Grouping.Name element specified in a Groupings collection in the query.

The following is the parent element of the **SubtotalGroupings.GroupingName** element.

Parent elements
SubtotalGroupings

The following is the XML Schema definition of the **SubtotalGroupings.GroupingName** element.

```
<xsd:element name="GroupingName" type="xsd:string" maxOccurs="unbounded" />
```

2.48 SubtotalMeasures

The **SubtotalMeasures** element specifies a collection of measures to calculate subtotals for.

The **SubtotalMeasures** element specifies a collection of SubtotalMeasures.MeasureName elements. The **SubtotalMeasures** element MUST contain at least one **SubtotalMeasures.MeasureName** element and can contain more.

The following are the parent and child elements of the **SubtotalMeasures** element.

Parent elements
SubtotalSet

Child elements
SubtotalMeasures.MeasureName

The following is the XML Schema definition of the **SubtotalMeasures** element.

```
<xsd:element name="SubtotalMeasures" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="MeasureName" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.48.1 SubtotalMeasures.MeasureName

The **SubtotalMeasures.MeasureName** element specifies the name of a measure expression in the measure group.

The **SubtotalMeasures.MeasureName** element MUST be specified at least once in its parent SubtotalMeasures collection and can occur more than once. The value of the **SubtotalMeasures.MeasureName** element MUST be a String and MUST match the value of the Expression.Name attribute of one of the expressions in the MeasureGroup for the semantic query.

The following is the parent element of the **SubtotalMeasures.MeasureName** element.

Parent elements
SubtotalMeasures

The following is the XML Schema definition of the **SubtotalMeasures.MeasureName** element.

```
<xsd:element name="MeasureName" type="xsd:string" maxOccurs="unbounded" />
```

2.49 Parameters

The **Parameters** element specifies a collection of parameters whose values can be used throughout the query.

The **Parameters** element specifies a collection of Parameter elements. This element MUST contain at least one Parameters.Parameter element and can contain more.

The following are the parent and child elements of the **Parameters** element.

Parent elements
SemanticQuery

Child elements
Parameters.Parameter

The following is the XML Schema definition of the **Parameters** element.

```
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.49.1 Parameters.Parameter

The **Parameters.Parameter** element specifies a parameter to the query.

The **Parameters.Parameter** element MUST be specified at least once in its parent Parameters collection and can occur more than once. The **Parameters.Parameter** element is of type Parameter.

The following is the parent element of the **Parameters.Parameter** element.

Parent elements
Parameters

The following is the XML Schema definition of the **Parameters.Parameter** element.

```
<xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
```

2.50 Parameter

The **Parameter** element specifies a parameter to the query.

The following are the parent elements, attributes, and child elements of the **Parameter** element.

Parent elements
Parameters

Attributes
Parameter.Name

Child elements
Parameter.Cardinality
Parameter.DataType
Parameter.Expression
Parameter.Nullable

The following is the XML Schema definition of the **Parameter** element.

```
<xsd:complexType name="ParameterType">
  <xsd:all>
    <xsd:element name="DataType" type="LiteralDataTypeEnum" />
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Cardinality" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="One" />
          <xsd:enumeration value="Many" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="Name" type="NonEmptyString" use="required" />
</xsd:complexType>
```

2.50.1 Parameter.Name

The **Parameter.Name** attribute specifies the name of the parameter.

The **Parameter.Name** attribute MUST be specified. Its value MUST be a nonempty String that is unique among all **Parameter.Name** attribute values specified in the query, that is, those under the same SemanticQuery element. The value of the **Parameter.Name** attribute MUST NOT be one of the two reserved parameter names "DrillthroughContext" or "DrillthroughSourceQuery".

The following is the parent element of the **Parameter.Name** attribute.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.50.2 Parameter.Cardinality

The **Parameter.Cardinality** element specifies the cardinality of the parameter, which indicates whether this is a multivalue parameter that provides a set of values of the type specified by the value of the sibling **Parameter.Cardinality** element.

The **Parameter.Cardinality** element is optional. If this element is present, its value MUST be a String that is one of the following:

One (default): Specifies that the parameter provides a single value.

Many: Specifies that the parameter provides a set of values of the specified type.

If the **Parameter.Cardinality** element is not present, its value is interpreted as "One".

The value of the **Parameter.Cardinality** element MUST match the cardinality of the value of the sibling Parameter.Expression element if present, that is, the value of the Parameter.Expression element MUST be either a scalar or a set depending upon whether the value of the **Parameter.Cardinality** element is "One" or "Many".

The following is the parent element of the **Parameter.Cardinality** element.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Cardinality** element.

```
<xsd:element name="Cardinality" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="One" />
      <xsd:enumeration value="Many" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

2.50.3 Parameter.DataType

The **Parameter.DataType** element specifies the data type of the parameter.

The **Parameter.DataType** element MUST be specified. Its value MUST be a String. In a client application, a user MUST enter a value for the parameter that can be interpreted as the specified type. If a default value is provided for the parameter by the sibling Parameter.Expression element, the value of that sibling Parameter.Expression element MUST evaluate to the specified type.

The value of the **Parameter.DataType** element MUST be one of the following:

Boolean: Specifies that the data type of the parameter is the SMDL data type Boolean corresponding to the CLR data type **Boolean**.

DateTime: Specifies that the data type of the parameter is the SMDL data type DateTime corresponding to the CLR data types **DataTime** and **DateTimeOffset**.

Decimal: Specifies that the data type of the parameter is the SMDL data type Decimal corresponding to the CLR data types **Decimal** and **UInt64**.

EntityKey: Specifies that the data type of the parameter is the SMDL data type EntityKey, which is a tuple of key column values for an entity and does not correspond to any CLR data type.

Float: Specifies that the data type of the parameter is the SMDL data type Float corresponding to the CLR data types **Single** and **Double**.

Integer: Specifies that the data type of the parameter is the SMDL data type Integer corresponding to the CLR data types **Int16**, **Int32**, **Int64**, **UInt16**, **UInt32**, **Byte**, and **SByte**.

String: Specifies that the data type of the parameter is the SMDL data type **String** corresponding to the CLR data types **String**, **Char**, and **Guid**.

Time: Specifies that the data type of the parameter is the SMDL data type Time corresponding to the CLR data type **Timespan**.

The value of the **Parameter.DataType** element MUST NOT be Binary.

The following is the parent element of the **Parameter.DataType** element.

Parent Elements
Parameter

The following is the XML Schema definition of the **Parameter.DataType** element.

```
<xsd:element name="DataType" type="LiteralDataTypeEnum" />
```

2.50.4 Parameter.Expression

The **Parameter.Expression** element specifies the default value for the parameter, if no value is supplied by the user in the client application. If the **Parameter.Expression** element is omitted, a parameter value is required to be supplied by the user.

The **Parameter.Expression** element is optional. This element is of type Expression. If the **Parameter.Expression** element is present, its value MUST evaluate to the data type and cardinality specified by the sibling elements Parameter.DataType and Parameter.Cardinality.

Furthermore, the value MUST NOT be NULL if the parameter is not nullable, that is, the value of the Parameter.Nullable element is false. If the **Parameter.Expression** element is not present, its value is interpreted as NULL and the user MUST supply a value to the client application.

The **Parameter.Expression** element MUST have one of a static function Function, Literal, or Null child elements in addition to the optional CustomProperties child element. AttributeRef, EntityRef, ParameterRef, or Path child elements MUST NOT be specified.

The following is the parent element of the **Parameter.Expression** element.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
```

2.50.5 Parameter.Nullable

The **Parameter.Nullable** element specifies that NULL is a valid value for the parameter.

The **Parameter.Nullable** element is optional. If this element is present, its value MUST be a Boolean. A value of true specifies that NULL is a valid value for the parameter. If the **Parameter.Nullable** element is not present, its value is interpreted as false.

The following is the parent element of the **Parameter.Nullable** element.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Nullable** element.

```
<xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
```

2.51 Expression

The **Expression** element specifies an expression, along with an optional path by which to reach an entity whose context is used to evaluate the expression. Each expression is evaluated within the context of a specific entity called the context entity.

Exactly one of the following child elements, Expression.AttributeRef, Expression.EntityRef, Expression.Function, Expression.Literal, Expression.Null, or Expression.ParameterRef MUST be specified as a child of the **Expression** element.

The following are the parent elements, attributes, and child elements of the **Expression** element.

Parent elements
Arguments
Attribute
CalculatedAttributes
Details
Filter
Grouping
Measures
Parameter

Attributes
Expression.Name

Child elements
Expression.AttributeRef
Expression.CustomProperties
Expression.EntityRef
Expression.Function
Expression.Literal
Expression.Null
Expression.ParameterRef
Expression.Path

The following is the XML Schema definition of the **Expression** element.

```
<xsd:complexType name="ExpressionType">
  <xsd:all>
    <xsd:element name="Path" type="PathType" minOccurs="0" />
    <xsd:element name="Function" type="FunctionType" minOccurs="0" />
    <xsd:element name="AttributeRef" type="AttributeRefType" minOccurs="0" />
    <xsd:element name="EntityRef" type="EntityRefType" minOccurs="0" />
    <xsd:element name="ParameterRef" type="ParameterRefType" minOccurs="0" />
    <xsd:element name="Literal" type="LiteralType" minOccurs="0" />
    <xsd:element name="Null" type="NullType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
```

```
                minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="Name" type="NonEmptyString" use="optional" />
</xsd:complexType>
```

2.51.1 Expression.Name

The **Expression.Name** attribute specifies the name of the expression.

The **Expression.Name** attribute MUST be specified, unless the expression is used as an argument for another expression, as a default value for a Parameter, or as the expression for a Filter. Otherwise, the **Expression.Name** attribute is optional in the listed exceptional cases. If this attribute is present, its value MUST be a nonempty String. If the **Expression.Name** attribute is not present, its value is interpreted as NULL.

If the **Expression.Name** attribute is used as an argument, its value MUST be unique within the argument collection. If the **Expression.Name** attribute is mandatory, its value MUST be unique within the query.

The following is the parent element of the **Expression.Name** attribute.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="optional" />
```

2.51.2 Expression.AttributeRef

The **Expression.AttributeRef** element specifies an attribute reference node in a query expression tree.

The **Expression.AttributeRef** element is optional. This element is of type AttributeRef. If the **Expression.AttributeRef** element is not present, its value is interpreted as NULL. Exactly one of the following elements, **Expression.AttributeRef**, Expression.EntityRef, Expression.Function, Expression.Literal, Expression.Null, or Expression.ParameterRef MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.AttributeRef** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.AttributeRef** element.

```
<xsd:element name="AttributeRef" type="AttributeRefType" minOccurs="0" />
```

2.51.3 Expression.CustomProperties

The **Expression.CustomProperties** element specifies a collection of custom properties for the expression.

The **Expression.CustomProperties** element is optional. This element is of type CustomProperties. If the **Expression.CustomProperties** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Expression.CustomProperties** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.51.4 Expression.EntityRef

The **Expression.EntityRef** element specifies an entity reference node in a query expression tree.

The **Expression.EntityRef** element is optional. This element is of type EntityRef. If the **Expression.EntityRef** element is not present, its value is interpreted as NULL. Exactly one of the following elements, Expression.AttributeRef, **Expression.EntityRef**, Expression.Function, Expression.Literal, Expression.Null, or Expression.ParameterRef, MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.EntityRef** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.EntityRef** element.

```
<xsd:element name="EntityRef" type="EntityRefType" minOccurs="0" />
```

2.51.5 Expression.Function

The **Expression.Function** element specifies a function node in a query expression tree.

The **Expression.Function** element is optional. This element is of type Function. If the **Expression.Function** element is not present, its value is interpreted as NULL. Exactly one of the following elements, Expression.AttributeRef, Expression.EntityRef, **Expression.Function**, Expression.Literal, Expression.Null, or Expression.ParameterRef MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.Function** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.Function** element.

```
<xsd:element name="Function" type="FunctionType" minOccurs="0" />
```

2.51.6 Expression.Literal

The **Expression.Literal** element specifies a literal node in a query expression tree.

The **Expression.Literal** element is optional. This element is of type Literal. If this element is not present, its value is interpreted as NULL. Exactly one of the following elements, Expression.AttributeRef, Expression.EntityRef, Expression.Function, **Expression.Literal**, Expression.Null, or Expression.ParameterRef MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.Literal** element.

Parent Elements
Expression

The following is the XML Schema definition of the **Expression.Literal** element.

```
<xsd:element name="Literal" type="LiteralType" minOccurs="0" />
```

2.51.7 Expression.Null

The **Expression.Null** element specifies a null value node in a query expression tree.

The **Expression.Null** element is optional. This element is of type Null. If the **Expression.Null** element is not present, its value is interpreted as NULL. Exactly one of the following elements, Expression.AttributeRef, Expression.EntityRef, Expression.Function, Expression.Literal, **Expression.Null**, or Expression.ParameterRef MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.Null** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.Null** element.

```
<xsd:element name="Null" type="NullType" minOccurs="0" />
```

2.51.8 Expression.ParameterRef

The **Expression.ParameterRef** element specifies a parameter reference node in a query expression tree.

The **Expression.ParameterRef** element is optional. This element is of type ParameterRef. If the **Expression.ParameterRef** element is not present, its value is interpreted as NULL. Exactly one of the following elements, Expression.AttributeRef, Expression.EntityRef, Expression.Function,

Expression.Literal, Expression.Null, or **Expression.ParameterRef** MUST be specified as a child of the parent Expression element.

The following is the parent element of the **Expression.ParameterRef** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.ParameterRef** element.

```
<xsd:element name="ParameterRef" type="ParameterRefType" minOccurs="0" />
```

2.51.9 Expression.Path

The **Expression.Path** element specifies the path by which to reach an entity whose context is used to evaluate the expression.

The **Expression.Path** element is optional. This element is of type Path. If the **Expression.Path** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Expression.Path** element.

Parent elements
Expression

The following is the XML Schema definition of the **Expression.Path** element.

```
<xsd:element name="Path" type="PathType" minOccurs="0" />
```

2.52 AttributeRef

The **AttributeRef** element specifies an attribute reference node in a query expression tree.

Exactly one AttributeRef.AttributeID or AttributeRef.AttributeName element MUST be specified as a child of the **AttributeRef** element.

The following are the parent and child elements of the **AttributeRef** element.

Parent elements
Expression

Child elements
AttributeRef.AttributeID
AttributeRef.AttributeName

The following is the XML Schema definition of the **AttributeRef** element.

```
<xsd:complexType name="AttributeRefType">
```

```
  <xsd:all>
    <xsd:element name="AttributeID" type="xsd:QName" minOccurs="0" />
    <xsd:element name="AttributeName" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

2.52.1 AttributeRef.AttributeID

The **AttributeRef.AttributeID** element specifies the ID of an attribute that the **AttributeRef.AttributeID** element refers to.

The **AttributeRef.AttributeID** element is optional. Exactly one **AttributeRef.AttributeID** or AttributeRef.AttributeName child element MUST be specified as a child of the parent AttributeRef element.

If this element is present, its value MUST be a QName and MUST match the value of the Attribute.ID attribute of one of the attributes in the semantic model.

Additionally, the attribute that is referred to MUST be an attribute contained by the current context entity or by any entity in the same inheritance hierarchy as the current context entity. If the **AttributeRef.AttributeID** element is not present, its value is interpreted as NULL.

The following is the parent element of the **AttributeRef.AttributeID** element.

Parent elements
AttributeRef

The following is the XML Schema definition of the **AttributeRef.AttributeID** element.

```
<xsd:element name="AttributeID" type="xsd:QName" minOccurs="0" />
```

2.52.2 AttributeRef.AttributeName

The **AttributeRef.AttributeName** element specifies the name of a calculated attribute within the query.

The **AttributeRef.AttributeName** element is optional. Exactly one AttributeRef.AttributeID or **AttributeRef.AttributeName** child element MUST be specified as a child of the parent AttributeRef element. If the **AttributeRef.AttributeName** element is present, its value MUST be a String. If the **AttributeRef.AttributeName** element is not present, its value is interpreted as NULL.

If the **AttributeRef.AttributeName** element is present, its value MUST be equal to the value of the Expression.Name element of a member of the SemanticQuery.CalculatedAttributes collection. Circular references MUST NOT be specified. Thus, the calculated attribute that is referred to MUST NOT be an ancestor of the expression that contains the parent **AttributeRef** element.

The following is the parent element of the **AttributeRef.AttributeName** element.

Parent elements
AttributeRef

The following is the XML Schema definition of the **AttributeRef.AttributeName** element.

```
<xsd:element name="AttributeName" type="xsd:string" minOccurs="0" />
```

2.53 EntityRef

The **EntityRef** element specifies an entity reference node in a query expression tree.

The following are the parent and child elements of the **EntityRef** element.

Parent elements
Expression

Child elements
EntityRef.EntityID

The following is the XML Schema definition of the **EntityRef** element.

```
<xsd:complexType name="EntityRefType">
  <xsd:all>
    <xsd:element name="EntityID" type="xsd:QName" />
  </xsd:all>
</xsd:complexType>
```

2.53.1 EntityRef.EntityID

The **EntityRef.EntityID** element specifies the ID of an entity.

The **EntityRef.EntityID** element MUST be specified. Its value MUST be a QName and MUST match the value of the Entity.ID attribute of one of the entities in the semantic model. Additionally, the entity that is referred to MUST be the current context entity or one related by inheritence.

The following is the parent element of the **EntityRef.EntityID** element.

Parent Elements
EntityRef

The following is the XML Schema definition of the **EntityRef.EntityID** element.

```
<xsd:element name="EntityID" type="xsd:QName" />
```

2.54 Function

The **Function** element specifies a function node in a query expression tree.

The following are the parent and child elements of the **Function** element.

Parent elements
Expression

Child elements
Function.Arguments
Function.FunctionName

The following is the XML Schema definition of the **Function** element.

```
<xsd:complexType name="FunctionType">
  <xsd:all>
    <xsd:element name="FunctionName" type="xsd:string" />
    <xsd:element name="Arguments" type="ExpressionsType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

2.54.1 Function.Arguments

The **Function.Arguments** element specifies a collection that is an ordered list of arguments to a function.

The **Function.Arguments** element is optional. This element is of type Arguments. If the **Function.Arguments** element is not present, the data type and cardinality of each argument MUST match the requirements of the function. These requirements are listed in section 2.65, **SMDL Functions**. If the **Function.Arguments** element is not present, its value is interpreted as NULL.

The following is the parent element of the **Function.Arguments** element.

Parent elements
Function

The following is the XML Schema definition of the **Function.Arguments** element.

```
<xsd:element name="Arguments" type="ExpressionsType" minOccurs="0" />
```

2.54.2 Function.FunctionName

The **Function.FunctionName** element specifies the name of a function.

The **Function.FunctionName** element MUST be specified. Its value MUST be a String. The value of the **Function.FunctionName** element MUST be one of the function names listed in the table in section 2.65 **SMDL Functions**.

The following is the parent element of the **Function.FunctionName** element.

Parent elements
Function

The following is the XML Schema definition of the **Function.FunctionName** element.

```
<xsd:element name="FunctionName" type="xsd:string" />
```

2.55 Arguments

The **Arguments** element specifies a collection that is an ordered list of arguments to a function.

The **Arguments** element MUST contain at least one Arguments.Expression element and can contain more. There MUST be one **Arguments.Expression** element for each argument that is specified by the function specified by the Function.FunctionName element. The data type and cardinality of each argument MUST match the requirements of the function. These requirements are listed in section 2.64.2.

The following are the parent and child elements of the **Arguments** element.

Parent elements
Function

Child elements
Arguments.Expression

The following is the XML Schema definition of the **Arguments** element.

```
<xsd:complexType name="ExpressionsType">
  <xsd:sequence>
    <xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.55.1 Arguments.Expression

The **Arguments.Expression** element specifies an argument to a function.

The **Arguments.Expression** element MUST be specified at least once in its parent Arguments collection and can occur more than once. The **Arguments.Expression** element is of type Expression. The data type and cardinality of each **Arguments.Expression** element MUST match the requirements of the function. These requirements are listed in section 2.64.2.

The following is the parent element of the **Arguments.Expression** element.

Parent Elements
Arguments

The following is the XML Schema definition of the **Arguments.Expression** element.

```
<xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
```

2.56 Literal

The **Literal** element specifies a literal node in a query expression tree.

Exactly one Literal.Value or Literal.Values element MUST be specified as a child of the **Literal** element.

The following are the parent and child elements of the **Literal** element.

Parent elements
Expression

Child elements
Literal.DataType
Literal.Value
Literal.Values

The following is the XML Schema definition of the **Literal** element.

```
<xsd:complexType name="LiteralType">
  <xsd:all>
    <xsd:element name="DataType" type="LiteralDataTypeEnum" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Values" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
```

2.56.1 Literal.DataType

The **Literal.DataType** element specifies the data type of the value of the literal.

The **Literal.DataType** element MUST be specified. Its value MUST be a String. The value of the sibling Literal.Value element or the collection of values of the sibling Literal.Values element MUST be able to be interpreted as the data type specified by the value of the **Literal.DataType** element. In the case where the **Literal.DataType** element does not have the value "EntityKey", the interpretation of the data type conforms to the XSD standard. Additionally, the value of the **Literal.Value** element or the collection of values of the sibling **Literal.Values** element is treated as a Variant data type.

The value of the **Literal.DataType** element MUST be one of the following:

Boolean: Specifies that the data type of the attribute is the SMDL data type Boolean corresponding to the CLR data type **Boolean**.

DateTime: Specifies that the data type of the attribute is the SMDL data type DateTime corresponding to the CLR data types **DateTime** and **DateTimeOffset**.

Time: Specifies that the data type of the attribute is the SMDL data type Time corresponding to the CLR data type **Timespan**.

Integer: Specifies that the data type of the attribute is the SMDL data type Integer corresponding to the CLR data types **Int16**, **Int32**, **Int64**, **UInt16**, **UInt32**, **Byte**, and **SByte**.

Decimal: Specifies that the data type of the attribute is the SMDL data type Decimal corresponding to the CLR data types **Decimal** and **UInt64**.

Float: Specifies that the data type of the attribute is the SMDL data type Float corresponding to the CLR data types **Single** and **Double**.

String: Specifies that the data type of the attribute is the SMDL data type **String** corresponding to the CLR data types **String**, **Char**, and **Guid**.

EntityKey: Specifies that the data type of the attribute is the SMDL data type EntityKey, which is a tuple of key column values for an entity and does not correspond to any CLR data type.

The value of the **Literal.DataType** element MUST NOT be "Binary".

The following is the parent element of the **Literal.DataType** element.

Parent elements
Literal

The following is the XML Schema definition of the **Literal.DataType** element.

```
<xsd:element name="DataType" type="LiteralDataTypeEnum" />
```

2.56.2 Literal.Value

The **Literal.Value** element specifies the value of the literal. This element is optional.

This element is either of type Variant or of type EntityKey. The interpretation of the data type is specified by the value of the Literal.DataType element. If the **Literal.Value** element is not present, its value is interpreted as NULL. Exactly one **Literal.Value** or Literal.Values element MUST be specified as a child of the Literal element.

The following is the parent element of the **Literal.Value** element.

Parent elements
Literal

The following is the XML Schema definition of the **Literal.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

2.56.3 Literal.Values

The **Literal.Values** element specifies a collection of values of the literal.

The **Literal.Values** element is optional. This element is of type Values. If the **Literal.Values** element is not present, its value is interpreted as NULL. Exactly one Literal.Value or **Literal.Values** element MUST be specified as a child of the Literal element.

The following is the parent element of the **Literal.Values** element.

Parent elements
Literal

The following is the XML Schema definition of the **Literal.Values** element.

```
<xsd:element name="Values" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.57 Values

The **Values** element specifies a collection of literal values.

The following are the parent and child elements of the **Values** element.

Parent elements
Literal

Child elements
Values.Value

The following is the XML Schema definition of the **Values** element.

```
<xsd:element name="Values" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2.57.1 Values.Value

The **Values.Value** element specifies one of a collection of values of the literal.

The **Values.Value** element MUST be specified and can occur more than once in its parent Values collection. This element is either of type Variant or of type EntityKey. The interpretation of the data type is specified by the value of the Literal.DataType element.

The following is the parent element of the **Values.Value** element.

Parent elements
Values

The following is the XML Schema definition of the **Values.Value** element.

```
<xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
```

2.58 Null

The **Null** element specifies a null value node in a query expression tree.

The **Null** element has no properties; that is, it has no child elements, no attributes, and no assigned value. If the **Null** element is present, the parent Expression element has the value NULL.

The following is the parent element of the **Null** element.

Parent elements
Expression

The following is the XML Schema definition of the **Null** element.

```
<xsd:complexType name="NullType" />
```

2.59 ParameterRef

The **ParameterRef** element specifies a parameter reference node in a query expression tree.

The following are the parent and child elements of the **ParameterRef** element.

Parent elements
Expression

Child elements
ParameterRef.ParameterName

The following is the XML Schema definition of the **ParameterRef** element.

```
<xsd:complexType name="ParameterRefType">
  <xsd:all>
    <xsd:element name="ParameterName" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
```

2.59.1 ParameterRef.ParameterName

The **ParameterRef.ParameterName** element specifies the name of a parameter.

The **ParameterRef.ParameterName** element MUST be specified. Its value MUST be a String that matches the value of the Parameter.Name attribute of one of the parameters in the semantic query. "DrillthroughContext" and "DrillthroughSourceQuery" are reserved parameter names and MUST NOT be used as the value of a **ParameterRef.ParameterName** element.

The following is the parent element of the **ParameterRef.ParameterName** element.

Parent elements
ParameterRef

The following is the XML Schema definition of the **ParameterRef.ParameterName** element.

```
<xsd:element name="ParameterName" type="xsd:string" />
```

2.60 DrillthroughContext

The **DrillthroughContext** element specifies the value of an optional *DrillthroughContext* parameter that identifies additional filter information for the query specified by the *DrillthroughSourceQuery* parameter. The value of the *DrillthroughContext* parameter is a **DrillthroughContext** element and the value of the *DrillthroughSourceQuery* parameter is a SemanticQuery element. The DrillthroughContext.GroupingValues and DrillthroughContext.SelectedItems child elements of the **DrillthroughContext** element contain references to the Groupings, Measures, and Details elements of the **SemanticQuery** element specified by the *DrillthroughSourceQuery* parameter. The **DrillthroughContext** element is in the SMDL namespace as a top-level element.

The following are the child elements of the **DrillthroughContext** element.

Child elements
DrillthroughContext.GroupingValues
DrillthroughContext.SelectedItems
DrillthroughContext.SelectedPath

The following is the XML Schema definition of the **DrillthroughContext** element.

```
<xsd:element name="DrillthroughContext">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="SelectedItems" type="SelectedItemsType" />
      <xsd:element name="SelectedPath" type="SelectedPathType" minOccurs="0" />
      <xsd:element name="GroupingValues" type="GroupingValuesType" minOccurs="0">
        <xsd:unique name="GroupingNames">
          <xsd:selector xpath=".//*" />
          <xsd:field xpath="@Name" />
        </xsd:unique>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

2.60.1 DrillthroughContext.GroupingValues

The **DrillthroughContext.GroupingValues** element specifies a collection of **GroupingValues**. GroupingValues elements whose value is the value of the grouping for the selected item, and whose value of the attribute GroupValues.GroupingValue.Name is the name of a grouping whose value is needed to identify the selected instance of a selected item.

The **DrillthroughContext.GroupingValues** element is optional. This element is of type **GroupingValues**. If the **DrillthroughContext.GroupingValues** element is not present, its value is interpreted as NULL.

The following is the parent element of the **DrillthroughContext.GroupingValues** element.

Parent elements
DrillthroughContext

The following is the XML Schema definition of the **DrillthroughContext.GroupingValues** element.

```
<xsd:element name="GroupingValues" type="GroupingValuesType" minOccurs="0">
  <xsd:unique name="GroupingNames">
    <xsd:selector xpath=".//*" />
    <xsd:field xpath="@Name" />
  </xsd:unique>
</xsd:element>
```

2.60.2 DrillthroughContext.SelectedItems

The **DrillthroughContext.SelectedItems** element specifies a collection of references by name to Expression elements whose value determines what data is drilled into and displayed to the user.

The **DrillthroughContext.SelectedItems** element MUST be specified. This element is of type SelectedItems. The **SelectedItems** element MUST contain a reference to a single grouping expression, or MUST contain a collection of one or more references to expressions that are all in the same MeasureGroup or in the same Details collection.

The following is the parent element of the **DrillthroughContext.SelectedItems** element.

Parent elements
DrillthroughContext

The following is the XML Schema definition of the **DrillthroughContext.SelectedItems** element.

```
<xsd:element name="SelectedItems" type="SelectedItemsType" />
```

2.60.3 DrillthroughContext.SelectedPath

The **DrillthroughContext.SelectedPath** element specifies the XML that specifies the path from the selected item or items to the target drillthrough entity.

The **DrillthroughContext.SelectedPath** element is optional. This element is of type SelectedPath. If the **DrillthroughContext.SelectedPath** element is not present, its value is interpreted as the longest subpath from the base entity of the query that is shared by all DrillthroughContext.SelectedItems elements.

The following is the parent element of the **DrillthroughContext.SelectedPath** element.

Parent elements
DrillthroughContext

The following is the XML Schema definition of the **DrillthroughContext.SelectedPath** element.
<xsd:element name="SelectedPath" type="SelectedPathType" minOccurs="0" />

2.61 GroupingValues

The **GroupingValues** element specifies a collection of GroupingValues.GroupingValue elements whose value is the value of the grouping for the selected item(s), and whose value of the attribute GroupingValues.GroupingValue.Name is the name of a grouping whose value is needed to identify the selected instance of the selected item(s).

The **GroupingValues** element MUST contain at least one **GroupingValues.GroupingValue** child element and can contain more.

The following are the parent and child elements of the **GroupingValues** element.

Parent elements
DrillthroughContext

Child elements
GroupingValues.GroupingValue

The following is the XML Schema definition of the **GroupingValues** element.

```
<xsd:complexType name="GroupingValuesType">
  <xsd:sequence>
    <xsd:element name="GroupingValue" nillable="true" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="Name" type="NonEmptyString" use="required" />
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

2.61.1 GroupingValues.GroupingValue

The **GroupingValues.GroupingValue** element specifies the value of the grouping for the selected item(s).

The **GroupingValues.GroupingValue** element MUST be specified at least once in its parent GroupingValues collection and can occur more than once. The value of this element MUST be a nonempty String that specifies a value for the grouping.

To specify NULL as the value of a **GroupingValues.GroupingValue** element, the value MUST NOT be specified, but instead it is recommended that the attribute name and value **xsi:nil="true"** is specified on the **GroupingValues.GroupingValue** element.

The following is the parent element of the **GroupingValues.GroupingValue** element.

Parent elements
GroupingValues

The following is the XML Schema definition of the **GroupingValues.GroupingValue** element.

```
<xsd:element name="GroupingValue" nillable="true" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="Name" type="NonEmptyString" use="required" />
      </xsd:extension>
    </xsd:simpleContent>
```

```
  </xsd:complexType>
</xsd:element>
```

2.61.2 GroupingValues.GroupingValue.Name

The **GroupingValues.GroupingValue.Name** attribute specifies the name of a Grouping element whose value is needed to identify the selected instance of the selected item(s).

The **GroupingValues.GroupingValue.Name** attribute MUST be specified. Its value MUST be a nonempty String. The value of this attribute MUST match the value of the Grouping.Name attribute of one of the groupings in the query.

The following is the parent element of the **GroupingValues.GroupingValue.Name** attribute.

Parent elements
GroupingValues

The following is the XML Schema definition of the **GroupingValues.GroupingValue.Name** attribute.

```
<xsd:attribute name="Name" type="NonEmptyString" use="required" />
```

2.62 SelectedItems

The **SelectedItems** element specifies a collection of references by name to Expression elements whose value determines what data is drilled into and displayed to the user.

The **SelectedItems** element MUST contain at least one SelectedItems.SelectedItemName element and can contain more.

The **SelectedItems** element MUST contain a reference to a single grouping expression, or a collection of references to expressions that are all in the same MeasureGroup or in the same Details collection.

The following are the parent and child elements of the **SelectedItems** element.

Parent elements
DrillthroughContext

Child elements
SelectedItems.SelectedItemName

The following is the XML Schema definition of the **SelectedItems** element.

```
<xsd:complexType name="SelectedItemsType">
  <xsd:sequence>
    <xsd:element name="SelectedItemName" type="NonEmptyString"
                 maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.62.1 SelectedItems.SelectedItemName

The **SelectedItems.SelectedItemName** element specifies a reference by name to an Expression element whose value determines what data is drilled into and displayed to the user.

The **SelectedItems.SelectedItemName** element MUST be specified at least once in its parent SelectedItems collection and can occur more than once. The value of the **SelectedItems.SelectedItemName** element MUST be a nonempty String. The value of this element MUST match the value of the Expression.Name attribute of one of the expressions in the query.

If the **SelectedItems.SelectedItemName** element is specified just once, its value MUST be a reference to a single grouping, measure, or detail expression.

If the **SelectedItems.SelectedItemName** element is specified more than once, each **SelectedItems.SelectedItemName** element MUST contain a reference to expressions that are all in the same MeasureGroup or in the same Details collection.

The following is the parent element of the **SelectedItems.SelectedItemName** element.

Parent elements
SelectedItems

The following is the XML Schema definition of the **SelectedItems.SelectedItemName** element.

```
<xsd:element name="SelectedItemName" type="NonEmptyString" maxOccurs="unbounded" />
```

2.63 SelectedPath

The **SelectedPath** element specifies the XML that specifies the path from the selected item or items to the target drillthrough entity.

The **SelectedPath** element is optional. If this element is present, it MAY contain zero or more child SelectedPath.RolePathItem elements. If the **SelectedPath** element is not present, its value is interpreted as the longest subpath from the base entity of the query that is shared by all DrillthroughContext.SelectedItems elements.

The following are the parent and child elements of the **SelectedPath** element.

Parent elements
DrillthroughContext

Child elements
SelectedPath.RolePathItem

The following is the XML Schema definition of the **SelectedPath** element.

```
<xsd:complexType name="SelectedPathType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="RolePathItem" type="RolePathItemType" />
  </xsd:choice>
</xsd:complexType>
```

2.63.1 SelectedPath.RolePathItem

The **SelectedPath.RolePathItem** element specifies the XML that specifies the path from the selected item or items to the target drillthrough entity.

The **SelectedPath.RolePathItem** element is optional and can be specified more than once. This element is of type RolePathItem. If the **SelectedPath.RolePathItem** element is not present, the value of the parent SelectedPath element is interpreted as an explicit zero-length path.

The following is the parent element of the **SelectedPath.RolePathItem** element.

Parent Elements
SelectedPath

The following is the XML Schema definition of the **SelectedPath.RolePathItem** element.

```
<xsd:element name="RolePathItem" type="RolePathItemType" />
```

2.64 Expressions

2.64.1 Context Entity and Cardinality

Each expression is evaluated within the context of a specific entity. This entity is referred to as the context entity. Within the expression, attribute references (AttributeRef elements) can only be to attributes contained by the current entity context or by any entity in the same inheritance hierarchy as the current entity context. Similarly, entity references (EntityRef elements) can only be to the current entity context or one related by inheritance.

For grouping expressions, the context entity is the base entity of the hierarchy.

For grouping detail expressions, the context entity is the entity specified by the grouping expression, which is always an **EntityRef** element for grouping detail expressions.

For measure expressions, the context entity is the base entity of the measure group.

For hierarchy filter expressions, the context entity is the base entity of the hierarchy.

For calculated attributes, the context entity depends on where it is used in the query just like subexpressions.

For subexpressions, the context entity depends on the combination of the containing context entity and the path.

Paths to subcomponents of the expression start in the context entity for that expression. Each **PathItem** element MUST be meaningful in the current context. For the RolePathItem element, this means it MUST be a role contained in the context entity.

Each **PathItem** element changes the current context entity. For the **RolePathItem** element, the new context entity is the target entity of the role.

In addition, a **PathItem** element can change the cardinality of the expression’s value. For the **RolePathItem** element, if the cardinality of the role is "Many" or "OptionalMany", the expression’s value is a set. Note that if the contained value was already a set rather than a scalar, the resulting value is not a set of sets. Instead, the individual contained sets are grouped and unioned into larger sets.

2.64.2 Function Argument Data Types

Arguments to SMDL functions can take either a scalar value of a given data type or a set of values. Every argument that can take a set of values can also take a scalar of the same data type instead. That scalar is treated as a set of one item.

In addition, Integers can be used for arguments that only take Decimal and/or Float : The **Integer** is autocast to the appropriate type. **Decimal** is used if the argument supports **Decimal** or **Float** is used if the argument only supports **Float**. Similarly, **Decimal** can be used for arguments that only take **Float**: The **Decimal** is autocast to **Float** ; however, a warning is generated about potential loss of precision.

Arguments to SMDL functions can have the following data types: **Integer**, **Decimal**, **Float**, Boolean, String, DateTime, Time, EntityKey, and Language.Many of these functions have an argument that can have multiple data types. In the function descriptions that follow in sections 2.64.3 through section 2.64.4, the argument data types are described in the **Type** column of the table specifying the function. The following collections of multiple data types are specified: Numeric, Eq, Sort, and Any.These collections of data types are described in sections 2.64.2.1 through section 2.64.2.4.Many of the SMDL functions can have a return value with multiple data types. In this case, an asterisk appears in the **Type** column and the full specification of the return value **Type** appears in the **Specification** column.

2.64.2.1 Argument Type: Numeric

The **Numeric** argument data type is used for numeric data and can be one of the SMDL data types: Integer, Decimal, or Float.

2.64.2.2 Argument Type: Eq

The **Eq** argument data type is used for equateable data types and can be one of the SMDL data types Integer, Decimal, Float, Boolean, DateTime, Time, String, or EntityKey.

2.64.2.3 Argument Type: Sort

The **Sort** argument data type is used for sortable data types and can be one of the SMDL data types Integer, Decimal, Float, DateTime, Time, or String.

2.64.2.4 Argument Type: Any

The **Any** argument data type is used for any data type and can be one of the SMDL data types Integer, Decimal, Float, Boolean, String, DateTime, Time, EntityKey, and Language.

2.64.3 Scalar Functions

Scalar functions are those functions whose arguments are all scalar values, not sets, and whose return value is also a scalar.

2.64.3.1 Scalar Function: Add

The following are the arguments, types, and specification of the **Add** scalar function.

Function	Arguments	Type	Specifications
Add	*Return*	*	*Return* type is Float if either item is Float, otherwise Decimal if either item is Decimal, otherwise Integer.

Function	Arguments	Type	Specifications
	Item1	Numeric	The first item to add.
	Item2	Numeric	The second item to add.

2.64.3.2 Scalar Function: Subtract

The following are the arguments, types, and specification of the **Subtract** scalar function.

Function	Argument	Type	Specification
Subtract	*Return*	*	*Return* type is Float if either item is **Float**, otherwise Decimal if either item is **Decimal**, otherwise Integer.
	Item1	Numeric	The item from which to subtract.
	Item2	**Numeric**	The item to subtract

2.64.3.3 Scalar Function: Multiply

The following are the arguments, types, and specification of the **Multiply** scalar function.

Function	Argument	Type	Specification
Multiply	*Return*	*	*Return* type is Float if either item is **Float**, otherwise Decimal if either item is **Decimal**, otherwise Integer.
	Item1	Numeric	The first item to multiply.
	Item2	**Numeric**	The second item to multiply

2.64.3.4 Scalar Function: Divide

The following are the arguments, types, and specification of the **Divide** scalar function.

Function	Argument	Type	Specification
Divide	*Return*	*	*Return* type is Float if either item is **Float**, otherwise Decimal if either item is **Decimal**, otherwise Integer.
	Item1	Numeric	The item by which to divide.
	Item2	**Numeric**	The item to divide.

2.64.3.5 Scalar Function: Power

The following are the arguments, types, and specification of the **Power** scalar function.

Function	Argument	Type	Specification
Power	*Return*	*****	*Return* type is Float if either item is **Float**, otherwise Decimal if either item is **Decimal**, otherwise Integer.
	Base	Numeric	The base to take to a power.
	Exponent	**Numeric**	The exponent.

2.64.3.6 Scalar Function: Negate

The following are the arguments, types, and specification of the **Negate** scalar function.

Function	Argument	Type	Specification
Negate	*Return*	*****	*Return* type is the same as the Item.
	Item	Numeric	The item to negate.

2.64.3.7 Scalar Function: Mod

The following are the arguments, types, and specification of the **Mod** scalar function.

Function	Argument	Type	Specification
Mod	*Return*	Integer	The remainder of the division.
	Item1	**Integer**	The item to divide into.
	Item2	**Integer**	The item by which to divide.

2.64.3.8 Scalar Function: Equals

The following are the arguments, types, and specification of the **Equals** scalar function.

Function	Argument	Type	Specification
Equals	*Return*	Boolean	Indicates that the items are the same.
	Item1	Eq	The first item to compare.
	Item2	*****	The second item to compare. MUST have the same type as *Item1*

Null equals null. For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.9 Scalar Function: NotEquals

The following are the arguments, types, and specification of the **NotEquals** scalar function.

Function	Argument	Type	Specification
NotEquals	*Return*	Boolean	Indicates that the items are not the same.
	Item1	Eq	The first item to compare.
	Item2	*	The second item to compare. MUST have the same type as *Item1*

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.10 Scalar Function: GreaterThan

The following are the arguments, types, and specification of the **GreaterThan** scalar function.

Function	Argument	Type	Specification
GreaterThan	*Return*	Boolean	Indicates that the first item is greater than the second item.
	Item1	Sort	The first item to compare.
	Item2	*	The second item to compare. MUST have the same type as *Item1*

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.11 Scalar Function: GreaterThanOrEquals

The following are the arguments, types, and specification of the **GreaterThanOrEquals** scalar function.

Function	Argument	Type	Specification
GreaterThanOrEquals	*Return*	Boolean	Indicates that the first item is greater than or equal to the second item.
	Item1	Sort	The first item to compare.
	Item2	*	The second item to compare. MUST have the same type as *Item1*

Null equals null. For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.12 Scalar Function: LessThan

The following are the arguments, types, and specification of the **LessThan** scalar function.

Function	Argument	Type	Specification
LessThan	*Return*	Boolean	Indicates that the first item is less than the second item.
	Item1	Sort	The first item to compare.
	Item2	*	The second item to compare.

Function	Argument	Type	Specification
			MUST have the same type as *Item1*

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.13 Scalar Function: LessThanOrEquals

The following are the arguments, types, and specification of the **LessThanOrEquals** scalar function.

Function	Argument	Type	Specification
LessThanOrEquals	*Return*	Boolean	Indicates that the first item is less than or equal to the second item.
	Item1	Sort	The first item to compare.
	Item2	*	The second item to compare. MUST have the same type as *Item1*

Null equals null. For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.14 Scalar Function: And

The following are the arguments, types, and specification of the **And** scalar function.

Function	Argument	Type	Specification
And	*Return*	Boolean	True if both *Item1* and *Item2* are true.
	Item1	**Boolean**	The first condition.
	Item2	**Boolean**	The second condition. Not evaluated if *Item1* is false.

2.64.3.15 Scalar Function: Or

The following are the arguments, types, and specification of the **Or** scalar function.

Function	Argument	Type	Specification
Or	*Return*	Boolean	True if either *Item1* or *Item2* are true.
	Item1	**Boolean**	The first condition.
	Item2	**Boolean**	The second condition. Not evaluated if *Item1* is true.

2.64.3.16 Scalar Function: Not

The following are the arguments, types, and specification of the **Not** scalar function.

Function	Argument	Type	Specification
Not	*Return*	Boolean	True if *Item* is false.
	Item	**Boolean**	The condition to negate.

2.64.3.17 Scalar Function: Truncate

The following are the arguments, types, and specification of the **Truncate** scalar function.

Function	Argument	Type	Specification
Truncate	*Return*	*	*Return* type is the same as *Item*.
	Item	Decimal or Float	The item to truncate.
	Digits	Integer	The number of decimal digits to which to truncate. Negative numbers indicate truncation to the left of the decimal point.

2.64.3.18 Scalar Function: Round

The following are the arguments, types, and specification of the **Round** scalar function.

Function	Argument	Type	Specification
Round	*Return*	*	*Return* type is the same as *Item*.
	Item	Decimal or Float	The item to round.
	Digits	Integer	The number of decimal digits to which to round.

2.64.3.19 Scalar Function: Integer

The following are the arguments, types, and specification of the **Integer** scalar function.

Function	Argument	Type	Specification
Integer	*Return*	Integer	The item cast as an **Integer**.
	Item	Numeric or String	The item to cast.

Numeric values are truncated. The invariant locale is used to cast numeric String values. Period is the decimal separator. Thousands separator (comma) is not allowed.

2.64.3.20 Scalar Function: Decimal

The following are the arguments, types, and specification of the **Decimal** scalar function.

Function	Argument	Type	Specification
Decimal	*Return*	Decimal	The item cast as a **Decimal**.
	Item	Numeric or String	The item to cast.

Numeric values are rounded to match the resulting type. The invariant locale is used to cast numeric String values. Period is the decimal separator. Thousands separator (comma) is not allowed.

2.64.3.21 Scalar Function: Float

The following are the arguments, types, and specification of the **Float** scalar function.

Function	Argument	Type	Specification
Float	*Return*	Float	The item cast as a **Float**.
	Item	Numeric or String	The item to cast.

Numeric values are rounded to match the resulting type. The invariant locale is used to cast numeric String values. Period is the decimal separator. Thousands separator (comma) is not allowed.

2.64.3.22 Scalar Function: String

The following are the arguments, types, and specification of the **String** scalar function.

Function	Argument	Type	Specification
String	*Return*	String	The item cast as a **String**.
	Item	Numeric	The item to cast.

The invariant locale is used to cast numeric String values. Period is the decimal separator. Thousands separator (comma) is not allowed.

2.64.3.23 Scalar Function: Length

The following are the arguments, types, and specification of the **Length** scalar function.

Function	Argument	Type	Specification
Length	*Return*	Integer	Length of the string, in characters. 0 for empty or null strings.
	String	String	The string of which to determine the length.

2.64.3.24 Scalar Function: Find

The following are the arguments, types, and specification of the **Find** scalar function.

Function	Argument	Type	Specification
Find	*Return*	Integer	Position of the first instance of the substring within the string. 0 if not found or null string or substring.

Function	Argument	Type	Specification
	String	String	The string in which to search for a contained string.
	Substring	**String**	The substring to search for.

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.25 Scalar Function: Substring

The following are the arguments, types, and specification of the **Substring** scalar function.

Function	Argument	Type	Specification
Substring	*Return*	String	The substring of the string from *Start* to *Start*+*Length*
	String	**String**	The string of which to extract a substring.
	Start	Integer	The start position within the string (1-based).
	Length	**Integer**	The number of characters.

2.64.3.26 Scalar Function: Left

The following are the arguments, types, and specification of the **Left** scalar function.

Function	Argument	Type	Specification
Left	*Return*	String	The substring of the string from 1 to *Length*.
	String	**String**	String from which to obtain the leftmost characters.
	Length	Integer	The number of characters.

2.64.3.27 Scalar Function: Right

The following are the arguments, types, and specification of the **Right** scalar function.

Function	Argument	Type	Specification
Right	*Return*	String	The substring of the string from the length of the string-*Length*+1 to the length of the string.
	String	**String**	String from which to obtain the rightmost characters.
	Length	Integer	The number of characters.

2.64.3.28 Scalar Function: Concat

The following are the arguments, types, and specification of the **Concat** scalar function.

Function	Argument	Type	Specification
Concat	*Return*	String	The second string concatenated to the end of the first string.
	String1	**String**	The first string to concatenate. Null is treated as the empty string.
	String2	**String**	The second string to concatenate. Null is treated as the empty string.

2.64.3.29 Scalar Function: Lower

The following are the arguments, types, and specification of the **Lower** scalar function.

Function	Argument	Type	Specification
Lower	*Return*	String	The string with all uppercase characters converted to lowercase.
	String	**String**	The string to convert to lower case.

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.30 Scalar Function: Upper

The following are the arguments, types, and specification of the **Upper** scalar function.

Function	Argument	Type	Specification
Upper	*Return*	String	The string with all lowercase characters converted to uppercase.
	String	**String**	The string to convert to upper case.

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.31 Scalar Function: LTrim

The following are the arguments, types, and specification of the **LTrim** scalar function.

Function	Argument	Type	Specification
LTrim	*Return*	String	The string with all leading spaces removed.
	String	**String**	The string from which to trim leading spaces.

2.64.3.32 Scalar Function: RTrim

The following are the arguments, types, and specification of the **RTrim** scalar function.

Function	Argument	Type	Specification
RTrim	*Return*	String	The string with all leading spaces removed.

Function	Argument	Type	Specification
	String	**String**	The string from which to trim leading spaces.

2.64.3.33 Scalar Function: Replace

The following are the arguments, types, and specification of the **Replace** scalar function.

Function	Arguments	Type	Specification
Replace	*Return*	String	The string with all instances of *Find* replaced with *Replace*
	String	**String**	The string in which to replace all instances of one substring with another.
	Find	**String**	The substring to search for.
	Replace	**String**	The substring with which to replace the *Find* string.

For string functions that depend on collation, the underlying collation properties of the physical data store are used.

2.64.3.34 Scalar Function: Date 1

The following are the arguments, types, and specification of the **Date** scalar function.

Function	Arguments	Type	Specification
Date	*Return*	DateTime	A **DateTime** with the given year, month and day at 00:00:00.
	Year	Integer	The year for the date.
	Month	**Integer**	The month (1–12) for the date.
	Day	**Integer**	The day (1–31) for the date. The value MUST be a valid day of the given month and year.

2.64.3.35 Scalar Function: DateTime

The following are the arguments, types, and specification of the **DateTime** scalar function.

Function	Arguments	Type	Specification
DateTime	*Return*	DateTime	A **DateTime** with the given year, month, day, hour, minute and second.
	Year	Integer	The year for the date.
	Month	**Integer**	The month (1–12) for the date.
	Day	**Integer**	The day (1–31) for the date. The value MUST be a valid day of the given month and year.
	Hour	**Integer**	The hour (0–23) for the time.
	Minute	**Integer**	The minute (0–59) for the time.

Function	Arguments	Type	Specification
	Second	**Integer**	The second (0–60) for the time.

2.64.3.36 Scalar Function: Year

The following are the arguments, types, and specification of the **Year** scalar function.

Function	Arguments	Type	Specification
Year	*Return*	Integer	The year of the date and time.
	DateTime	DateTime	The date and time from which to extract the year.

2.64.3.37 Scalar Function: Quarter

The following are the arguments, types, and specification of the **Quarter** scalar function.

Function	Arguments	Type	Specification
Quarter	*Return*	Integer	The quarter (1–4) of the date and time.
	DateTime	DateTime	The date and time from which to extract the quarter.

2.64.3.38 Scalar Function: Month

The following are the arguments, types, and specification of the **Month** scalar function.

Function	Arguments	Type	Specification
Month	*Return*	Integer	The month (1–12) of the date and time.
	DateTime	DateTime	The date and time from which to extract the month.

2.64.3.39 Scalar Function: Day

The following are the arguments, types, and specification of the **Day** scalar function.

Function	Arguments	Type	Specification
Day	*Return*	Integer	The day (1–31) of the date and time.
	DateTime	DateTime	The date and time from which to extract the day.

2.64.3.40 Scalar Function: Hour

The following are the arguments, types, and specification of the **Hour** scalar function.

Function	Arguments	Type	Specification
Hour	*Return*	Integer	The hour (0–23) of the date and time.
	DateTime	DateTime or Time	The date and time from which to extract the hour.

2.64.3.41 Scalar Function: Minute

The following are the arguments, types, and specification of the **Minute** scalar function.

Function	Arguments	Type	Specification
Minute	*Return*	Integer	The minute (0–59) of the date and time.
	DateTime	DateTime or Time	The date and time from which to extract the minute.

2.64.3.42 Scalar Function: Second

The following are the arguments, types, and specification of the **Second** scalar function.

Function	Arguments	Type	Specification
Second	*Return*	Integer	The second (0–59) of the date and time.
	DateTime	DateTime or Time	The date and time from which to extract the second.

2.64.3.43 Scalar Function: Time

The following are the arguments, types, and specification of the **Time** scalar function.

Function	Arguments	Type	Specification
Time	*Return*	Time	The time of the date and time.
	DateTime	DateTime	The date and time from which to extract the time.

2.64.3.44 Scalar Function: DayOfYear

The following are the arguments, types, and specification of the **DayOfYear** scalar function.

Function	Arguments	Type	Specification
DayOfYear	*Return*	Integer	The day of year (1–366) of the date and time.

Function	Arguments	Type	Specification
	DateTime	DateTime	The date and time from which to extract the year.

2.64.3.45 Scalar Function: Week

The following are the arguments, types, and specification of the **Week** scalar function.

Function	Arguments	Type	Specification
Week	*Return*	Integer	The week (1–54) of the date and time. The first day of the week is determined by the default first day of week associated with the culture of the semantic model.
	DateTime	DateTime	The date and time from which to extract the week.

For more information, see [MSDN-FDOW].

2.64.3.46 Scalar Function: DayOfWeek

The following are the arguments, types, and specification of the **DayOfWeek** scalar function.

Function	Arguments	Type	Specification
DayOfWeek	*Return*	Integer	The day of week (1–7) of the date and time. *1 = Monday through 7 = Sunday.*
	DateTime	DateTime	The date and time from which to extract the day of week.

2.64.3.47 Scalar Function: Date 2

The following are the arguments, types, and specification of the **Date** scalar function.

Function	Arguments	Type	Specification
Date	*Return*	DateTime	The date and time with the time cleared (set to 00:00:00).
	DateTime	**DateTime**	The date and time from which to remove the time.

2.64.3.48 Scalar Function: Now

The following are the arguments, types, and specification of the **Now** scalar function.

Function	Arguments	Type	Specification
Now	*Return*	DateTime	The current date and time. **Now** is a static function.

2.64.3.49 Scalar Function: Today

The following are the arguments, types, and specification of the **Today** scalar function.

Function	Arguments	Type	Specification
Today	*Return*	DateTime	The current date and time with the time cleared (set to 00:00:00). **Today** is a static function.

2.64.3.50 Scalar Function: DateDiff

The following are the arguments, types, and specification of the **DateDiff** scalar function.

Function	Arguments	Type	Specification
DateDiff	*Return*	Integer	The difference between the *Start* date and time and the *End* date and time, in units of *Interval*. Negative if the *Start* date and time is after the *End* date and time.
	Interval	String	The units over which to compute the date and time difference. MUST be one of the following: "Year", "Quarter", "Month", "Day", "Hour", "Minute", "Second", "Week". MUST be a Literal element.
	Start	DateTime or Time	The start date and time.
	End	**DateTime** or **Time**	The end date and time. MUST have the same data type as *Start*.

2.64.3.51 Scalar Function: DateAdd

The following are the arguments, types, and specification of the **DateAdd** scalar function.

Function	Arguments	Type	Specification
DateAdd	*Return*	*	The date and time that is the result of adding the specified number of interval units to the original date and time. Will have the same data type as **Start**.
	Interval	String	The unit of time to add to the date and time. MUST be one of the following: "Year", "Quarter", "Month", "Day", "Hour", "Minute", "Second", "Week". MUST be a Literal element.
	Number	Integer	The number of interval units to add to the date and time.
	Start	DateTime or Time	The date and time to which to add.

2.64.4 Aggregate Functions

Aggregate functions are those functions that have an argument that is a set of values and whose return value is a scalar, the aggregate of the set of values.

2.64.4.1 Aggregate Function: Sum

The following are the arguments, types, and specification of the **Sum** aggregate function.

Function	Arguments	Type	Specification
Sum	*Return*	*	The sum of the non-null values of the items. *Return* type is the same as the *Items*.
	Items	Numeric	The set of items to sum.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.2 Aggregate Function: Avg

The following are the arguments, types, and specification of the **Avg** aggregate function.

Function	Arguments	Type	Specification
Avg	*Return*	*	The average of the non-null values of the items. *Return* type is Decimal if the items are **Decimal** or Integer, otherwise Float.
	Items	Numeric	The set of items to average.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.3 Aggregate Function: Max

The following are the arguments, types, and specification of the **Max** aggregate function.

Function	Arguments	Type	Specification
Max	*Return*	*	The maximum of the non-null values of the items. *Return* type is the same as the items.
	Items	Sort	The set of items of which to determine the maximum.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.4 Aggregate Function: Min

The following are the arguments, types, and specification of the **Min** aggregate function.

Function	Arguments	Type	Specification
Min	*Return*	*	The minimum of the non-null values of the items. *Return* type is the same as the items.

Function	Arguments	Type	Specification
	Items	Sort	The set of items of which to determine the minimum.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.5 Aggregate Function: Count

The following are the arguments, types, and specification of the **Count** aggregate function.

Function	Arguments	Type	Specification
Count	*Return*	Integer	The count of the non-null values of the items.
	Items	Any	The set of items to count.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.6 Aggregate Function: CountDistinct

The following are the arguments, types, and specification of the **CountDistinct** aggregate function.

Function	Arguments	Type	Specification
CountDistinct	*Return*	Integer	The count of the distinct non-null values of the items.
	Items	Any	The set of items to count. MUST NOT be of type*EntityKey.*

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.7 Aggregate Function: StDev

The following are the arguments, types, and specification of the **StDev** aggregate function.

Function	Arguments	Type	Specification
StDev	*Return*	Float	The standard deviation of the non-null values of the items.
	Items	Numeric	The set of items of which to determine the standard deviation.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.8 Aggregate Function: StDevP

The following are the arguments, types, and specification of the **StDevP** aggregate function.

Function	Arguments	Type	Specification
StDevP	*Return*	Float	The population standard deviation of the non-null values of the items.
	Items	Numeric	The set of items of which to determine the population standard deviation.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.9 Aggregate Function: Var

The following are the arguments, types, and specification of the **Var** aggregate function.

Function	Arguments	Type	Specification
Var	*Return*	Float	The variance of the non-null values of the item.
	Items	Numeric	The set of items of which to determine the variance.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.4.10 Aggregate Function: VarP

The following are the arguments, types, and specification of the **VarP** aggregate function.

Function	Arguments	Type	Specification
VarP	*Return*	Float	The population variance of the non-null values of the item.
	Items	Numeric	The set of items of which to determine the population variance.

The value of the *Items* argument MUST NOT be a literal set, that is, the *Items* argument MUST NOT be a Literal element with a Literal.Values child element.

2.64.5 Passthrough Functions

A passthrough function is a function where the returned values retain the same identity as the data handed to one of its arguments, but the data can be modified in some way, such as by applying scalar functions or removing specific rows. A passthrough function MUST have at least one argument, the passthrough argument, that takes a set of values and that MUST return a set of values of the same type. If there is more than one set argument, only one of them is the passthrough argument.

2.64.5.1 Passthrough Function: Evaluate

The following are the arguments, types, and specification of the **Evaluate** passthrough function.

Function	Arguments	Type	Specification
Evaluate	*Return*	*	The evaluated expression. *Return* type is the same as *Expression*.
	Expression	Any	The expression to evaluate.

The **Evaluate** function is used for controlling the entity context in which an expression is evaluated.

For example, compare the following expressions:

```
Avg([customer->order->product]Price) versus
Avg([customer->order]Evaluate([order->product]Price))
```

The first expression averages the prices for each distinct product the customer ordered, whereas the second expression averages the prices (obtained from the product entity) for each order the customer placed.

2.64.5.2 Passthrough Function: Filter

The following are the arguments, types, and specification of the **Evaluate** passthrough function.

Function	Arguments	Type	Specification
Filter	*Return*	*	*Return* type is the same as *FilterItems*.
	FilterItems	Any	The set of values to filter.
	FilterCondition	Boolean	The set of values that indicate whether to include the corresponding instances.

2.64.6 Information Functions

Information functions provide basic information about the user executing a query.

2.64.6.1 Information Function: GetUserID

The following are the arguments, types, and specification of the **GetUserID** function.

Function	Arguments	Type	Specification
GetUserID	*Return*	String	The user's user id. **GetUserID** is a static function.

2.64.6.2 Information Function: GetUserCulture

The following are the arguments, types, and specification of the **GetUserCulture** function.

Function	Arguments	Type	Specification
GetUserCulture	*Return*	Language	The user's language and locale. **GetUserCulture** is a static function.

2.64.7 Other Functions

2.64.7.1 Other Function: In

The following are the arguments, types, and specification of the **In** function.

Function	Arguments	Type	Specification
In	*Return*	Boolean	*Return* type is Float if either item is **Float**, otherwise Decimal if either item is **Decimal**, otherwise Integer.

Function	Arguments	Type	Specification
	Item	Eq	The item to check for set membership.
	Set	*	A Literal with no Path or a ParameterRef. MUST have the same type as *Item.*

2.64.7.2 Other Function: If

The following are the arguments, types, and specification of the **If** function.

Function	Arguments	Type	Specification
If	*Return*	*	*Return* type is the same as *TrueCase*.
	Condition	Boolean	The condition to test.
	TrueCase	Any	The value to return if the condition is true. Not evaluated if the condition is false.
	FalseCase	*	Value to return if the condition is false. MUST have the same type as *TrueCase.* Not evaluated if the condition is true.

2.64.7.3 Other Function: Switch

The following are the arguments, types, and specification of the **Switch** function.

Function	Arguments	Type	Specification
Switch	*Return*	*	*Return* type is the same as *Value1*. Returns Null if all conditions are false.
	Condition1	Boolean	The condition to test.
	Value1	Any	The value to return if *Condition1* is true. Not evaluated if *Condition1* is false. MUST NOT be of type EntityKey.
	Switch can have zero or more additional condition and value pairs.		
	ConditionN	**Boolean**	The condition to test. Not evaluated if any earlier condition is true.
	ValueN	*	Value to return if *ConditionN* is true. MUST have same type as *Value1.* Not evaluated if *ConditionN* is not evaluated or if *ConditionN* is false. MUST NOT be of type EntityKey.

2.64.7.4 Other Function: Aggregate

The following are the arguments, types, and specification of the **Aggregate** function.

Function	Arguments	Type	Specification
Aggregate	*Return*	*	*Return* type is the same as *Expression*.
	Expression	Any	The aggregate expression to evaluate.

The **Aggregate** function is used for controlling the entity context in which an aggregation is evaluated.

For example, consider an attribute named "TotalSales" defined on an entity named "Order". The expression to show the total sales within the context of an order would be TotalSales. The expression to calculate the total sales within the context of a customer would be:

```
Aggregate([customer->order]TotalSales)
```

The *Expression* argument MUST contain a non-anchored expression or one or more nested passthrough functions that MUST take any data type, where the innermost passthrough argument has a non-anchored node in a semantic query expression tree.

For example:

```
Aggregate([customer->order]Filter([order->product]Sum(UnitPrice),
Equals(Shipped,"True")))
```

2.65 Error Codes

2.65.1 InvalidDataSourceView

InvalidDataSourceView is a critical error that MUST occur when the SemanticModel.DataSourceView element is invalid and the error is detected while validating either the XML syntax or XSD Schema. *InvalidDataSourceView* MUST also occur if there is an error reading the DateTime values of the **CreateTimestamp** or **LastSchemaUpdate** child elements of the **SemanticModel.DataSourceView** element.

2.65.2 InvalidSemanticModel

InvalidSemanticModel is a critical error that MUST occur when the SemanticModel element is invalid and the error is detected while validating either the XML syntax or XSD Schema.

2.65.3 InvalidSemanticQuery

InvalidSemanticQuery is a critical error that MUST occur when the SemanticQuery element is invalid and the error is detected while validating either the XML syntax or XSD Schema.

2.65.4 InvalidDrillthroughContext

InvalidDrillthroughContext is a critical error that MUST occur when the DrillthroughContext element is invalid and the error is detected while validating either the XML syntax or XSD Schema.

2.65.5 InvalidCulture

InvalidCulture is a critical error that MUST occur when the value of either of the SemanticModel.Culture or the Attribute.DataCulture elements is not a valid ISO culture.

2.65.6 DuplicateItemID

DuplicateItemID is a critical error that MUST occur when more than one model item in the semantic model has the same ID attribute. Item IDs MUST be unique within a model.

2.65.7 InvalidEntityBinding

InvalidEntityBinding is a critical error that MUST occur when the Entity element has both of the following child elements: Entity.Table and Entity.Column. The **Entity** element MUST have no more than one of these two child elements.

2.65.8 NestedVariations

NestedVariations is a critical error that MUST occur when an Attribute or Role element is a variation of another element and the **Attribute** or **Role** element contains a nested variation. Variation fields MUST NOT be nested inside other variation fields.

2.65.9 InvalidLinguistics

InvalidLinguistics is a critical error that MUST occur when the Linguistics child element is specified for a Role, but the Role.Name element has been omitted. **Linguistics** MUST NOT be specified for a **Role** if the **Role.Name** has not been specified.

2.65.10 MissingRelationEnd

MissingRelationEnd is a critical error that MUST occur when a Relation element does not contain a **RelationEnd** attribute and the parent of the **Relation** element is a Role element.

2.65.11 InvalidExpression

InvalidExpression is a critical error that MUST occur when an Expression element has either more than one or none of the following child elements: Expression.Function, Expression.AttributeRef, Expression.EntityRef, Expression.ParameterRef, Expression.Literal, and Expression.Null. The **Expression** element MUST have exactly one of these six child elements.

2.65.12 InvalidFunctionName

InvalidFunctionName is a critical error that MUST occur when the value of a Function.FunctionName element is not one of the valid function names listed in section 2.64 through section 2.64.7.4.

2.65.13 InvalidAttributeRef

InvalidAttributeRef is a critical error that MUST occur when an Expression element contains an AttributeRef element that has either both or neither of the following child elements: AttributeRef.AttributeID and AttributeRef.AttributeName. The **AttributeRef** element MUST have exactly one of these two child elements.

2.65.14 InvalidLiteral

InvalidLiteral is a critical error that MUST occur when a Literal element has either both or neither of the following child elements: Literal.Value and Literal.Values. The **Literal** element MUST have exactly one of these two child elements.

2.65.15 InvalidLiteralValue

InvalidLiteralValue is a critical error that MUST occur when the value of a Literal.Value or Values.Value element cannot be converted to the data type specified by the Literal.DataType child element of the containing Literal element. The value of the **Literal.Value** and **Values.Value** elements MUST be a valid value of the specified data type.

2.65.16 ItemNotFound

ItemNotFound is an error that MUST occur when a reference to a model item cannot be found. References to model items are by ID; the ID has a value of type QName. References to model items occur as the value of the following elements:

- Attribute.DefaultAggregateAttributeID
- AttributeReference.AttributeID
- RolePathItem.RoleID
- Role.RelatedRoleID
- HiddenFields.FieldFolderItemID
- Inheritance.InheritsFromEntityID
- ModelItems.ModelItemID
- BaseEntity.EntityID
- AttributeRef.AttributeID
- EntityRef.EntityID

The values of these elements MUST be references to items that occur in the semantic model.

ItemNotFound is a generally a critical error, but it can occur as a noncritical error, a warning, if the reference is contained in a SemanticQuery element and the reference is excluded from the query because of query subsetting, when the server is able to safely exclude portions of a **SemanticQuery** without compromising the query results. Model item references that MAY be contained in a **SemanticQuery** include the following:

- **RolePathItem.RoleID**
- **BaseEntity.EntityID**
- **AttributeRef.AttributeID**
- **EntityRef.EntityID**

2.65.17 InvalidReferencedItem

InvalidReferencedItem is a critical error that MUST occur when a reference to a model item is to an item of the wrong type. For example, the element Role.RelatedRoleID MUST be a reference to a Role element. An example of an *InvalidReferencedItem* error would be if the value of the **Role.RelatedRoleID** element was a valid QName value, but that value was the value of an Entity.ID.

References to model items are by ID; the ID has a value of type **QName**. References to model items by ID occur as the value of the following elements: Attribute.DefaultAggregateAttributeID, AttributeReference.AttributeID, RolePathItem.RoleID, **Role.RelatedRoleID**, HiddenFields.FieldFolderItemID, Inheritance.InheritsFromEntityID, ModelItems.ModelItemID,

BaseEntity.EntityID, AttributeRef.AttributeID, and EntityRef.EntityID. The values of each of these elements MUST be references to items of the correct expected type that occurs in the semantic model.

2.65.18 CircularInheritance

CircularInheritance is a critical error that MUST occur when an Inheritance.InheritsFromEntityID element specifies a reference to an Entity that is a direct or indirect inheritance from itself. Loops in inheritance are not allowed.

2.65.19 SelfReferentialRole

SelfReferentialRole is a critical error that MUST occur when a Role.RelatedRoleID element specifies a reference to its own parent Role element. The **Role.RelatedRoleID** element MUST refer to a **Role** other than itself.

2.65.20 GroupingNotFound

GroupingNotFound is an error that MUST occur when a SubtotalGroupings.GroupingName element within a SemanticQuery element or a GroupingValues.GroupingValue.Name attribute within a DrillthroughContext element specifies a reference by name to a Grouping element that does not exist.

GroupingNotFound is a generally a critical error but it can occur as a noncritical error, a warning, if the reference is from a **SubtotalGroupings.GroupingName** element and the reference is excluded from the query because of query subsetting, when the server is able to safely exclude portions of a **SemanticQuery** without compromising the query results.

2.65.21 MeasureNotFound

MeasureNotFound is an error that MUST occur when a SubtotalMeasures.MeasureName element within a SemanticQuery specifies a reference by name to a measure Expression that does not exist.

MeasureNotFound is a generally a critical error but it can occur as a noncritical error, a warning, if the reference is excluded from the query because of query subsetting, when the server is able to safely exclude portions of a **SemanticQuery** without compromising the query results.

2.65.22 CalculatedAttributeNotFound

CalculatedAttributeNotFound is an error that MUST occur when an AttributeRef.AttributeName element within an Expression specifies a reference by name to a calculated Attribute that does not exist in the semantic query.

CalculatedAttributeNotFound is a generally a critical error but it can occur as a noncritical error, a warning, if the reference is excluded from the query because of query subsetting, when the server is able to safely exclude portions of a SemanticQuery without compromising the query results.

2.65.23 ParameterNotFound

ParameterNotFound is an error that MUST occur when a ParameterRef.ParameterName element within an Expression specifies a reference by name to a Parameter that does not exist in the semantic query.

ParameterNotFound is a generally a critical error but it can occur as a noncritical error, a warning, if the reference is excluded from the query because of a query subsetting, when the server is able to safely exclude portions of a SemanticQuery without compromising the query results.

2.65.24 ResultExpressionNotFound

ResultExpressionNotFound is a critical error that MUST occur when a SelectedItems.SelectedItemName element within a DrillthroughContext element specifies a reference by name to an Expression that does not exist in the semantic query.

2.65.25 MissingItemName

MissingItemName is a critical error that MUST occur when one of the following model item elements does not have a **Name** child element:

- Entity
- Attribute
- EntityFolder
- FieldFolder
- Perspective

MissingItemName is a also critical error that MUST occur when a Role element does not have a child Role.Name element and a name for the **Role** element cannot be autoderived from the **Name** or **CollectionName** child elements of the **Entity** containing the related **Role** that is specified by the Role.RelatedRoleID element.

2.65.26 IDLocalNameLengthExceeded

IDLocalNameLengthExceeded is a critical error that MUST occur when the QName value of an ID of an item has a local name whose length exceeds the maximum length of 250 characters.

The ID of the item is one of the following attributes or elements:

- Entity.ID
- EntityFolder.ID
- Inheritance.InheritsFromEntityID
- Attribute.ID
- Attribute.DefaultAggregateAttributeID
- AttributeReference.AttributeID
- Role.ID
- Role.RelatedRoleID
- RolePathItem.RoleID
- FieldFolder.ID
- HiddenFields.FieldFolderItemID
- SemanticModel.ID
- Perspective.ID
- ModelItems.ModelItemID

- EntityRef.EntityID
- AttributeRef.AttributeID
- BaseEntity.EntityID

2.65.27 IDNamespaceLengthExceeded

IDNamespaceLengthExceeded is a critical error that MUST occur when the QName value of an ID of an item has a namespace whose length exceeds the maximum length of 150 characters.

The ID of the item is one of the following attributes or elements:

- Entity.ID
- EntityFolder.ID
- Inheritance.InheritsFromEntityID
- Attribute.ID
- Attribute.DefaultAggregateAttributeID
- AttributeReference.AttributeID
- Role.ID
- Role.RelatedRoleID
- RolePathItem.RoleID
- FieldFolder.ID
- HiddenFields.FieldFolderItemID
- SemanticModel.ID
- Perspective.ID
- ModelItems.ModelItemID
- EntityRef.EntityID
- AttributeRef.AttributeID
- BaseEntity.EntityID

2.65.28 InvalidGuid

InvalidGuid is a critical error that MUST occur when the QName value of an ID of an item is in the empty (or SMDL) namespace, but it is not a GUID. IDs in the empty (or SMDL) namespace MUST match the following GUID format: Gxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx.

The ID of the item is one of the following attributes or elements: Entity.ID, EntityFolder.ID, Inheritance.InheritsFromEntityID, Attribute.ID, Attribute.DefaultAggregateAttributeID, AttributeReference.AttributeID, Role.ID, Role.RelatedRoleID, RolePathItem.RoleID, FieldFolder.ID, HiddenFields.FieldFolderItemID, SemanticModel.ID, or Perspective.ID, ModelItems.ModelItemID, EntityRef.EntityID, AttributeRef.AttributeID, and BaseEntity.EntityID.

2.65.29 DuplicateItemName

DuplicateItemName is a critical error that MUST occur when more than one item in a collection has the same name. Item names MUST be unique among immediate siblings. The collections that MUST cause the error *DuplicateItemName* if they have duplicate names are the following: SemanticModel.Entities, SemanticModel.Perspectives, Entity.Fields, Attribute.Variations, Role.Variations, EntityFolder.Entities, and FieldFolder.Fields.

2.65.30 DuplicateEntityName

DuplicateEntityName is a critical error that MUST occur when more than one Entity in the SemanticModel has the same name. **Entity** names MUST be unique within a **SemanticModel**. The entities include all of those in the SemanticModel.Entities collection, in addition to those in EntityFolder subfolders.

2.65.31 DuplicateFieldName

DuplicateFieldName is a critical error that MUST occur when more than one field (Attribute or Role in an Entity has the same name. Field names MUST be unique within an **Entity**. The fields include all of those in the Entity.Fields collection, in addition to those in FieldFolder subfolders, and also fields that are variations of these fields.

2.65.32 MissingIdentifyingAttributes

MissingIdentifyingAttributes is a critical error that MUST occur when an Entity element does not have any AttributeReference elements in its Entity.IdentifyingAttributes collection. An **Entity** MUST have at least one **AttributeReference** element in its IdentifyingAttributes collection.

2.65.33 InvalidSetAttributeReference

InvalidSetAttributeReference is a critical error that MUST occur when an AttributeReference.AttributeID element references an attribute that is not an aggregate attribute, but is an attribute that returns a set of values as specified by the path that is used to reach the attribute. References to attributes without Attribute.IsAggregate set to true MUST NOT have an AttributeReference.Path element that contains a reference to a Role with a Role.Cardinality element that has a value of "Many" or "OptionalMany".

2.65.34 InvalidAggregateAttributeReference

InvalidAggregateAttributeReference is a critical error that MUST occur when an AttributeReference.AttributeID element references an attribute that is an aggregate attribute, but a nonaggregate attribute is expected. This error occurs when the **AttributeReference.AttributeID** element is a child of one of the following elements: Entity.IdentifyingAttributes, Entity.SecurityFilters, or Entity.DefaultSecurityFilter and the reference is to an aggregate attribute. An aggregate attribute is an Attribute that has an Attribute.IsAggregate child element with a value of true.

2.65.35 InvalidScalarAttributeReference

InvalidScalarAttributeReference is a critical error that MUST occur when an AttributeReference.AttributeID element references an attribute that is a nonaggregate attribute, but an aggregate attribute is expected. This error occurs when the **AttributeReference.AttributeID** element is a child of the Entity.DefaultAggregateAttributes element and the reference is to a nonaggregate attribute. An aggregate attribute is an Attribute that has an Attribute.IsAggregate child element with a value of true.

2.65.36 InvalidNonFilterAttributeReference

InvalidNonFilterAttributeReference is a critical error that MUST occur when an AttributeReference.AttributeID element references an attribute that is not a filter attribute, but a filter attribute is expected. This error occurs when the **AttributeReference.AttributeID** element is a child of one of the following elements: Entity.SecurityFilters or Entity.DefaultSecurityFilter and the reference is not to a filter attribute. A filter attribute is an Attribute that has an Attribute.IsFilter child element with a value of true.

2.65.37 InvalidHiddenAttributeReference

InvalidHiddenAttributeReference is a noncritical error that can occur when an AttributeReference.AttributeID element references an attribute that is a hidden attribute, but a non-hidden attribute is expected. This warning occurs when the **AttributeReference.AttributeID** element is a child of one of the following elements: Entity.IdentifyingAttributes, Entity.DefaultDetailAttributes, or Entity.DefaultAggregateAttributes and the reference is to a hidden attribute. A hidden attribute is an Attribute that has an Attribute.Hidden child element with a value of true.

2.65.38 ExpressionDataTypeMismatch

ExpressionDataTypeMismatch is a critical error that MUST occur when the data type of an Attribute.Expression element does not match the data type specified by the value of the Attribute.DataType element.

2.65.39 ExpressionNullableMismatch

ExpressionNullableMismatch is a critical error that MUST occur when the result of an Attribute.Expression element is nullable, but the sibling Attribute.Nullable element either is not specified or it has a value of false. An expression is nullable if any Role referenced in its Expression.Path has a Role.Cardinality element with a value of "OptionalOne" or "OptionalMany". An expression is also nullable if a function used in the expression can introduce nulls, or the expression references another attribute that is nullable.

2.65.40 MissingMimeType

MissingMimeType is a noncritical error that can occur when an Attribute element has an Attribute.DataType child element that has a value of "Binary" but the Attribute.MimeType child element has not been specified. The **Attribute.MIMEType** element SHOULD be specified for an **Attribute** when the **Attribute.DataType** child element has a value of "Binary".

2.65.41 IsAggregateWithDefaultAggregate

IsAggregateWithDefaultAggregate is a critical error that MUST occur when an Attribute element has specified an Attribute.DefaultAggregateAttributeID element that references an attribute, but the sibling Attribute.IsAggregate element has been specified with a value of true. A default aggregate attribute MUST NOT be specified for aggregate attributes.

2.65.42 NonAggregateAsDefaultAggregate

NonAggregateAsDefaultAggregate is a critical error that MUST occur when an Attribute element has specified an Attribute.DefaultAggregateAttributeID that references an **Attribute** element that has not specified an Attribute.IsAggregate child element or whose child element has been specified with a value of false. A default aggregate attribute MUST be an aggregate attribute.

2.65.43 NonVariationAsDefaultAggregate

NonVariationAsDefaultAggregate is a critical error that MUST occur when an Attribute element has specified an Attribute.DefaultAggregateAttributeID that references an **Attribute** element that is not a member of the Attribute.Variations collection of the original **Attribute**. A default aggregate attribute MUST be a variation of the attribute.

2.65.44 MissingRelatedRole

MissingRelatedRole is a critical error that MUST occur when a Role element does not have a Role.RelatedRoleID child element. A **Role** MUST have a **Role.RelatedRoleID** child element.

2.65.45 RelatedRolesMismatch

RelatedRolesMismatch is a critical error that MUST occur when a role and its related role do not refer to each other. The Role.RelatedRoleID value of the related role MUST be a reference to the original role. The related role is the role that is referred to by the **Role.RelatedRoleID** element of the original Role.

2.65.46 InvalidOptionalityOfRoleForColumnBoundEntity

InvalidOptionalityOfRoleForColumnBoundEntity is a noncritical error that can occur when a Role has a child Role.Cardinality element with a value of "OptionalOne" or "OptionalMany" and one of the two following conditions exist:

- The entity containing this role is bound to a table and the entity containing the related role is bound to a column in that table.
- The entity containing this role is bound to a column and the entity containing the related role is bound to a table that contains that column.

2.65.47 InvalidModelItemInPerspective

InvalidModelItemInPerspective is a critical error that MUST occur when a ModelItems.ModelItemID member of the Perspective.ModelItems collection is a reference to a SemanticModel or a Perspective element. Model items in a perspective MUST be references to one of the following: Entity, Attribute, Role, EntityFolder or FieldFolder elements.

2.65.48 MissingDataSourceView

MissingDataSourceView is a critical error that MUST occur when the SemanticModel element is missing a SemanticModel.DataSourceView element and the client application requires one to be present.<5>

2.65.49 MissingBinding

MissingBinding is a critical error that MUST occur when an Entity element is missing both its Entity.Table and Entity.Column child elements or the **Entity** has an Inheritance child element that is missing its Inheritance.Relation child.

The *MissingBinding* error also MUST occur when an Attribute element is missing its Attribute.Column child element and it does not have an Attribute.Expression child element or when a Role element is missing its Role.Relation child element and it is not true that one of the containing **Entity** or the related **Entity** elements is bound to a table and the other **Entity** is bound to a column that is contained in that table.

2.65.50 InvalidBinding

InvalidBinding is a critical error that MUST occur when anEntity.Table, Entity.Column, Attribute.Column, Inheritance.Relationor Role.Relation element specifies the name of an object that does not exist in the SemanticModel.DataSourceView element.

2.65.51 InvalidColumnReferenceInColumnEntity

InvalidColumnReferenceInColumnEntity is a critical error that MUST occur in any of the following three cases:

- An Attribute.Column element is bound to a column in a data source view (DSV) and the containing Entity is bound to a different column.
- An Inheritance.Relation element is bound to a relation and either the containing **Entity**, or the **Entity** that it is inherited from, is bound to a column that is different from the columns of the corresponding end of the relation.
- A Role.Relation element is bound to a relation and the end of the relation specified by the Relation.RelationEnd element does not equal the column that the related **Entity** is bound to, if that **Entity** is bound to a column. The related **Entity** is that **Entity** that contains the Role that is referred to by the Role.RelatedRoleID element.

2.65.52 MissingColumnTableName

MissingColumnTableName is a critical error that MUST occur when a Column element does not contain a **TableName** attribute, or it has a **TableName** attribute that has an empty String value and a **TableName** MUST be specified. A **TableName** MUST be specified if the column is bound to an entity that is not bound to a table; in other words, the entity is bound to a column.

2.65.53 InvalidColumnTableName

InvalidColumnTableName is a critical error that MUST occur when an Attribute element has a Column child element with a Column.TableName attribute that refers to a table that does not match the table for the containing entity.

2.65.54 InvalidColumnDataType

InvalidColumnDataType is a critical error that MUST occur when an Entity or Attribute has a Column child element that refers to a column in the underlying data base that has an unsupported data type.

2.65.55 NonPrimaryDataSource

NonPrimaryDataSource is a critical error that MUST occur when an Entity has a Table child element that refers to a table in the underlying database that is not in the primary data source.

2.65.56 MissingPrimaryKey

MissingPrimaryKey is a critical error that MUST occur when an Entity has a Table child element that refers to a table in the underlying database that does not have a primary key.

2.65.57 BinaryEntityColumn

BinaryEntityColumn is a critical error that MUST occur when an Entity has a Column child element that refers to a column in the underlying data base that has a binary data type. The **Column** child element of an **Entity** MUST NOT reference a column that has a binary data type.

2.65.58 InvalidInheritanceRelationTable

InvalidInheritanceRelationTable is a critical error that MUST occur when an Inheritance.Relation child element of an Entity element specifies the name of a relation and either the source table of the relation does not match the table that the containing entity is bound to, or the target table of the relation does not match the table that the inheritor entity is bound to.

2.65.59 NonUniqueInheritanceRelationColumns

NonUniqueInheritanceRelationColumns is a noncritical error that MAY occur when an Inheritance.Relation child element of an Entity element specifies a relation that has an end that is not bound to a set of uniquely constrained columns for the table that is associated with that end of the relation.

2.65.60 ColumnDataTypeMismatch

ColumnDataTypeMismatch is a critical error that MUST occur when the value of the Attribute.DataType element does not match the data type of the column that the attribute is bound to via the Attribute.Column element. The only mismatch that is allowed and is not considered an error occurs if the value of **Attribute.DataType** is Decimal, but the data type of the column is Integer.

2.65.61 ColumnNullableMismatch

ColumnNullableMismatch is a critical error that MUST occur when the value of the Attribute.Nullable element is false and the column that the attribute is bound to via the Attribute.Column element is nullable.

2.65.62 IsAggregateWithColumn

IsAggregateWithColumn is a critical error that MUST occur when the value of the Attribute.IsAggregate element is true but the attribute is bound to a column via an Attribute.Column element; that is, it is not a calculated attribute. An aggregate attribute MUST be a calculated attribute.

2.65.63 PromoteLookupForNonLookupEntity

PromoteLookupForNonLookupEntity is a critical error that MUST occur when the value of a Role.PromoteLookup element is true but either the Entity that contains the Role or the related **Entity** (which is the one that contains the **Role** referred to by the Role.RelatedRoleID element) does not have an Entity.IsLookup element or has one that has a value of false.

When the value of a **Role.PromoteLookup** element is true, both the containing **Entity** and the related **Entity** MUST have **Entity.IsLookup** child elements that have a value of true.

2.65.64 RoleRelationsMismatch

RoleRelationsMismatch is a critical error that MUST occur when the relation referred to by the Relation.Name child element of the Role.Relation element does not match the relation that is referred to by the **Relation.Name** child element of the **Role.Relation** element for the related role, which is the role that is referred to by the Role.RelatedRoleID element of the first Role.

2.65.65 RoleRelationEndsMismatch

RoleRelationEndsMismatch is a critical error that MUST occur when the value of the Relation.RelationEnd child element of the Role.Relation element matches the value of the

Relation.RelationEnd child element of the **Role.Relation** element for the related role, which is the role that is referred to by the Role.RelatedRoleID element of the first Role.

The values of the **Relation.RelationEnd** elements MUST be different for the role and its related role. One MUST have a value of "Source" and the other MUST have a value of "Target".

2.65.66 InvalidRoleRelationTable

InvalidRoleRelationTable is a critical error that MUST occur when the table associated with the end of the relation that is specified by the value of the Relation.RelationEnd child element of the Role.Relation element does not match the table that is bound to the entity that contains the related role.

2.65.67 NonUniqueRoleRelationColumns

NonUniqueRoleRelationColumns is a noncritical error that MAY occur when a Role element has a Role.Cardinality child element that has a value of "One" or "OptionalOne" and the Role.Relation child element of the **Role** specifies an end of a relation that is not bound to a set of uniquely constrained columns for the table that is associated with that end of the relation. **Role** elements that have a **Role.Cardinality** that has a value of "One" or "OptionalOne" require relations that are bound to uniquely constrained columns of the table.

2.65.68 NonBooleanFilterAttribute

NonBooleanFilterAttribute is a critical error that MUST occur when an Attribute has an Attribute.IsFilter child element with "CompositePathItem" a value of true and the value of the Attribute.DataType child element is not "Boolean".

2.65.69 CyclicExpression

CyclicExpression is a critical error that MUST occur when an Expression element contains a direct or indirect reference to itself. Loops in expressions are not allowed.

2.65.70 FieldReferenceOutOfContext

FieldReferenceOutOfContext is a critical error that MUST occur when an AttributeReference.AttributeID, a HiddenFields.FieldFolderItemID, a RolePathItem.RoleID, or an AttributeRef.AttributeID element references a field or field folder that is contained in an Entity that is not the current context entity or one related by inheritance.

2.65.71 EntityReferenceOutOfContext

EntityReferenceOutOfContext is a critical error that MUST occur when an EntityRef.EntityID element within an expression references an Entity that is not the current context entity or one related by inheritance.

2.65.72 NonAggregateExpression

NonAggregateExpression is a critical error that MUST occur when an Attribute element has an Attribute.IsAggregate child element that has a value of true and the expression defined by the Attribute.Expression element is an anchored expression or when a Measures.Expression element is an anchored expression.

Anchored expressions are either expressions that have an Expression.Path element that contains at least one RolePathItem element or expressions that have one of the following child elements:

- EntityRef elements.

- AttributeRef elements that refer to **Attribute** elements that do not have an **Attribute.IsAggregate** element whose value is true.
- Function elements if any of their arguments are anchored expressions, unless the argument takes a set and the function returns a scalar.

2.65.73 AggregateWithNonAggregateArgument

AggregateWithNonAggregateArgument is a critical error that MUST occur when an expression contains the Aggregate function with a non-aggregate argument. The **Aggregate** function MUST have an aggregate argument.

2.65.74 WrongNumberOfArguments

WrongNumberOfArguments is a critical error that MUST occur when an expression contains a function with the wrong number of arguments.

2.65.75 ArgumentDataTypeMismatch

ArgumentDataTypeMismatch is a critical error that MUST occur when an expression contains a function with an argument that has the wrong data type than is expected by the function for that argument.

2.65.76 ArgumentCardinalityMismatch

ArgumentCardinalityMismatch is a critical error that MUST occur when an expression contains a function with an argument that has a cardinality other than what is expected by the function for that argument; that is, either the function definition specifies the argument to be a scalar value and the actual argument that is provided is a set of values, or vice versa.

2.65.77 ArgumentValueOutOfRange

ArgumentValueOutOfRange is a critical error that MUST occur when an expression contains a function with an argument that has a value outside the range of allowable values. For example, the function Substring requires that the *Start* argument have a value greater than or equal to 1, and the Date function requires that the value of the *Day* argument be in the range from 1 through 31.

2.65.78 InvalidDateIntervalArgument

InvalidDateIntervalArgument is a critical error that MUST occur when an expression contains the DateAdd scalar function or the DateDiff scalar function with an *Interval* argument that is an Expression that either does not have a Literal child element or the **Expression** has a **Literal** child element and also a Path child element. The *Interval* argument requires an **Expression** with a **Literal** child element and without a **Path** child element.

2.65.79 InvalidDateIntervalValue

InvalidDateIntervalValue is a critical error that MUST occur when an expression contains the DateAdd scalar function or the DateDiff scalar function with an *Interval* argument that is an Expression that has a Literal child element without a Path and the value of the Literal.DataType element is "String" and the **Literal** has a Literal.Value child element whose value is not one of the following: "Year", "Quarter", "Month", "Day", "Hour", "Minute", "Second", or "Week".

2.65.80 InvalidInSetArgument

InvalidInSetArgument is a critical error that MUST occur when an expression contains an In function with a *Set* argument that is not a non-nullable constant. The *Set* argument of the **In** function requires a value that is a ParameterRef, a static function, or a Literal whose value is a non-nullable constant. The Expression that specifies the value of the *Set* argument MUST have an empty or no Path child element.

2.65.81 InvalidLiteralSetArgument

InvalidLiteralSetArgument is a critical error that MUST occur when an expression contains any function with an argument that is an Expression that has a Literal child element that returns a set of values via its Literal.Values child element. The only exception to this rule is the *Set* argument of the In function. In general, function arguments MUST NOT take a literal set as a value except for the *Set* argument of the **In** function.

2.65.82 ImplicitDecimalCastToFloat

ImplicitDecimalCastToFloat is a noncritical error that MAY occur when an expression contains a function and the definition of that function specifies an argument that is a Float value and a Decimal value is provided instead. Casting a **Decimal** value as a **Float** MAY result in loss of precision.

2.65.83 EntityKeyTypeMismatch

EntityKeyTypeMismatch is a critical error that MUST occur when an expression contains a function with more than one argument that is an EntityKey and the **EntityKey** arguments identify instances of different Entity elements. All **EntityKey** arguments of a function MUST identify instances of the same **Entity**.

2.65.84 MissingExpressionName

MissingExpressionName is a critical error that MUST occur when an Expression.Name attribute is expected and one is not present in the Expression element. An **Expression.Name** attribute is optional if the expression is used as an argument for another expression, is a default value for a Parameter, or is the expression for a Filter.

In all other cases, the **Expression.Name** attribute is mandatory and it MUST have a nonempty String value. This includes the following:

- Grouping.Expression
- Details.Expression
- Measures.Expression
- CalculatedAttributes.Expression

2.65.85 TopLevelSetExpression

TopLevelSetExpression is a critical error that MUST occur when an expression returns a set of values for each instance of an entity and it MUST return a single scalar value for each instance of the entity. This error can occur only for the following elements: Attribute.Expression, Grouping.Expression, Details.Expression, Measures.Expression, and Filter.Expression.

2.65.86 EmptySemanticQuery

EmptySemanticQuery is a critical error that MUST occur when the SemanticQuery element does not contain any Grouping.Expression, Details.Expression, or Measures.Expression elements after query subsetting. The **SemanticQuery** element MUST contain at least one of these elements. Not having these elements after query subsetting usually occurs because the user does not have permission to the items referenced by the query or the items referenced by the query have been deleted in the underlying model.

2.65.87 MultipleHierarchies

MultipleHierarchies is a critical error that MUST occur when a SemanticQuery element contains more than one Hierarchy element in its SemanticQuery.Hierarchies collection.

2.65.88 MultipleMeasureGroups

MultipleMeasureGroups is a critical error that MUST occur when a SemanticQuery element contains more than one MeasureGroup element in its SemanticQuery.MeasureGroups collection.

2.65.89 DuplicateGroupingName

DuplicateGroupingName is a critical error that MUST occur when more than one Grouping element within a SemanticQuery has the same Grouping.Name value. **Grouping** names MUST be unique within a **SemanticQuery**.

2.65.90 DuplicateExpressionName

DuplicateExpressionName is a critical error that MUST occur when more than one of the following Expression elements within a SemanticQuery has the same Expression.Name value: Grouping.Expression, Details.Expression, Measures.Expression, and CalculatedAttributes.Expression. These **Expression** names MUST be unique within a **SemanticQuery**.

2.65.91 MissingBaseEntity

MissingBaseEntity is a critical error that MUST occur when a Hierarchy or MeasureGroup element is missing a BaseEntity child element. A **BaseEntity** child element MUST be specified for **Hierarchy** and **MeasureGroup** elements.

2.65.92 MissingGroupingName

MissingGroupingName is a critical error that MUST occur when a Grouping element is missing a Grouping.Name attribute or has one with an empty value. A nonempty **Grouping.Name** attribute MUST be specified for every **Grouping**.

2.65.93 MissingGroupingExpression

MissingGroupingExpression is a critical error that MUST occur when a Grouping element is missing a Grouping.Expression child element. A **Grouping.Expression** child element MUST be specified for every **Grouping**.

2.65.94 BinaryGroupingExpression

BinaryGroupingExpression is a critical error that MUST occur when a Grouping.Expression element has a Binary data type. A **Grouping.Expression** element MUST NOT have a **Binary** data type.

2.65.95 NonEntityGroupingWithDetails

NonEntityGroupingWithDetails is a critical error that MUST occur when a Grouping.Expression does not have an EntityRef child element and a nonempty Details collection is specified. A nonempty **Details** child element is allowed only if the **Grouping.Expression** is an **EntityRef**.

2.65.96 InvalidFilter

InvalidFilter is a critical error that MUST occur when a Filter.Expression element has a data type that is not Boolean. The data type of a **Filter.Expression** MUST be **Boolean**.

2.65.97 BaseEntityMismatch

BaseEntityMismatch is a critical error that MUST occur when a MeasureGroup.BaseEntity element refers to a different entity than the Hierarchy.BaseEntity element refers to when both are in the same SemanticQuery.

2.65.98 MissingMeasures

MissingMeasures is a critical error that MUST occur when a MeasureGroup element is missing a MeasureGroup.Measures child element or has an empty Measures collection. A nonempty **MeasureGroup.Measures** collection MUST be specified for each **MeasureGroup**.

2.65.99 DuplicateParameterName

DuplicateParameterName is a critical error that MUST occur when more than one Parameter element within a SemanticQuery has a Parameter.Name attribute with the same value. **Parameter** names MUST be unique within a **SemanticQuery**.

2.65.100 MissingParameterName

MissingParameterName is a critical error that MUST occur when a Parameter.Name attribute is missing from a Parameter element or has an empty String value. The **Parameter.Name** attribute MUST be specified.

2.65.101 InvalidParameterName

InvalidParameterName is a critical error that MUST occur when the value of a Parameter.Name attribute is "DrillthroughSourceQuery" or "DrillthroughContext". These two values are reserved names that MUST NOT be used.

2.65.102 InvalidParameterExpression

InvalidParameterExpression is a critical error that MUST occur when the default value Expression for a Parameter contains one or more of the following child elements: Path, a non-static Function, AttributeRef, EntityRef, or ParameterRef. The Parameter.Expression element MUST NOT contain these elements.

A static **Function** is one of the following four functions: Now, Today, GetUserID, and GetUserCulture. These functions MAY appear as the default value Expression for a Parameter and no others.

2.65.103 ParameterExpressionDataTypeMismatch

ParameterExpressionDataTypeMismatch is a critical error that MUST occur when the Expression that specifies the default value for a Parameter has a different data type than the data type specified by the Parameter.DataType element.

2.65.104 ParameterExpressionCardinalityMismatch

ParameterExpressionCardinalityMismatch is a critical error that MUST occur when the Expression that specifies the default value for a Parameter has a different cardinality than the cardinality specified by the Parameter.Cardinality element; that is, a *ParameterExpressionCardinalityMismatch* error occurs when a scalar default value was expected, but a multivalue parameter was provided.

2.65.105 ParameterExpressionNullableMismatch

ParameterExpressionNullableMismatch is a critical error that MUST occur when a Parameter.Nullable child element of a Parameter is not specified or has a value of false, but the default value Expression for the parent **Parameter** element is nullable. The nullability of the default value **Expression** MUST match the value of the **Parameter.Nullable** element for the **Parameter**.

An expression is nullable if any Role referenced in its Expression.Path has a Role.Cardinality element with a value of "OptionalOne" or "OptionalMany". An expression is also nullable if a function used in the expression can introduce nulls, or if the expression references another attribute that is nullable.

2.65.106 InvalidParameterValueType

InvalidParameterValueType is a critical error that MUST occur when the value provided for a Parameter at runtime cannot be converted to the data type that is specified by the Parameter.DataType element.

2.65.107 InvalidParameterValueCardinality

InvalidParameterValueCardinality is a critical error that MUST occur when the value provided for a Parameter has a different cardinality than the cardinality specified by the Parameter.Cardinality element; that is, a scalar value was expected and a multivalue parameter was provided.

2.65.108 NullParameterValue

NullParameterValue is a critical error that MUST occur when a Parameter.Nullable element is not specified or when the **Parameter.Nullable** element has a value of false, but the value provided for the parent Parameter element is NULL. A NULL value can be provided only for the **Parameter** if the value of the **Parameter.Nullable** element for the **Parameter** is true.

2.65.109 MissingParameterValue

MissingParameterValue is a critical error that MUST occur when a default value Parameter.Expression child element is not specified for a Parameter element and a runtime value for the **Parameter** has not been provided.

2.65.110 UnusedParameterValue

UnusedParameterValue is a noncritical error that MAY occur when a value is provided at runtime for a Parameter that has a given name, but a **Parameter** with this name is not defined in the SemanticQuery.

2.65.111 InvalidDrillSelectedItems

InvalidDrillSelectedItems is a critical error that MUST occur when the SelectedItems element of the **DrillthroughContext** parameter refers to items in more than one context. The **SelectedItems** element MUST contain either a reference to a single Grouping expression or references to expressions that are all in the same **MeasureGroup** or in the same **Details** collection.

2.65.112 InvalidDrillSelectedPath

InvalidDrillSelectedPath is a critical error that MUST occur when the SelectedPath child element of the DrillthroughContext element specifies a path that is not equal to or is not a subset of the longest allowed path from the base entity of the query that is shared by all SelectedItems.

2.65.113 InvalidDrillTargetEntity

InvalidDrillTargetEntity is a critical error that MUST occur when the BaseEntity of the target SemanticQuery refers to a different Entity than the target entity that is specified by the SelectedPath element of the DrillthroughContext or any **Entity** in the same inheritance tree as the target entity.

The target entity specified by the **SelectedPath** element is the **Entity** that contains the Role that is specified by the Role.RelatedRoleID element of the **Role** that is referred to by the last RolePathItem in the **SelectedPath** collection.

2.65.114 LoopInSecurityFilters

LoopInSecurityFilters is a critical error that MUST occur when a SecurityFilters.AttributeReference element indirectly depends on itself. Loops in security filters are not allowed.

An example of a **SecurityFilters.AttributeReference** element indirectly depending on itself would be when an Entity element has a **SecurityFilters.AttributeReference** child element that refers to an Attribute that is contained by a second **Entity** element. If this second **Entity** element has a **SecurityFilters.AttributeReference** child element that refers to an **Attribute** that is contained by the first **Entity** element, a loop in the security filters exists; that is, the two entities depend on each other.

3 Structure Examples

3.1 SMDL

The following sections provide examples for the structures defined in the SMDL file format specification. Examples of the SMDL file format include examples that demonstrate defining the major components of a semantic model. These include defining the SemanticModel, a Perspective, a SemanticQuery, and defining various model items, such as an Entity, an Attribute, and a Role. Additionally, an example of a DrillthroughContext, Variations elements of an Attribute, and Expression elements are also provided.

In the XML code provided for each example, the XML code is interspersed with textual descriptions of the code. Additional annotation has also been placed within the XML code in the form of comments to aid in understanding the examples. An XML comment begins with the string "<!—" and ends with the string "-->". In particular, references are usually preceded with an annotation that indicates the name of the object that the reference is to. As references are implemented with GUID, this additional annotation is necessary to clarify the names of the objects that are being referred to. The annotation helps with the readability of the code.

3.1.1 SemanticModel

The following is an example of the SMDL for a semantic model in SMDL schema 2004/10 [MSFT-SMDL200410]. It shows the use of the various properties of a SemanticModel. This example shows elements for describing the semantic objects: Entity, Attribute, and Role; elements for describing the bindings: Table, Column, and Relation; and the element for describing the DSV SemanticModel.DataSourceView. These elements are discussed in greater detail in the examples that follow this example.

```
<?xml version="1.0" encoding="utf-8"?>
<SemanticModel ID="G4e45f937-18e9-495b-9413-2ca744aeb2d1"
xmlns="http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <Description>This is the description for the sample NorthwindSlim model.</Description>
  <Culture>en-US</Culture>
  <Entities>
    <Entity ID="G2cdf6d03-0f8b-47f6-9338-d2dfa0270aae">
      <Name>Customer</Name>
      <CollectionName>Customers</CollectionName>
      <IdentifyingAttributes>
        <AttributeReference>
          <!--Customer ID-->
          <AttributeID>G423152da-ea1d-4cad-b402-8b56d0d7d538</AttributeID>
        </AttributeReference>
      </IdentifyingAttributes>
      <DefaultDetailAttributes>
        <AttributeReference>
          <!--Customer ID-->
          <AttributeID>G423152da-ea1d-4cad-b402-8b56d0d7d538</AttributeID>
        </AttributeReference>
      </DefaultDetailAttributes>
      <DefaultAggregateAttributes>
        <AttributeReference>
          <!--#Customers-->
          <AttributeID>G9e0d254b-792e-4547-90cb-5e71f7ca7b92</AttributeID>
        </AttributeReference>
      </DefaultAggregateAttributes>
      <InstanceSelection>Dropdown</InstanceSelection>
      <IsLookup>true</IsLookup>
      <Fields>
        <Attribute ID="G9e0d254b-792e-4547-90cb-5e71f7ca7b92">
          <Name>#Customers</Name>
```

```
        <DataType>Integer</DataType>
        <Expression>
          <Function>
            <FunctionName>Count</FunctionName>
            <Arguments>
              <Expression>
                <EntityRef>
                  <!--Customer-->
                  <EntityID>G2cdf6d03-0f8b-47f6-9338-d2dfa0270aae</EntityID>
                </EntityRef>
              </Expression>
            </Arguments>
          </Function>
        </Expression>
        <IsAggregate>true</IsAggregate>
        <SortDirection>Descending</SortDirection>
        <Format>n0</Format>
        <EnableDrillthrough>true</EnableDrillthrough>
      </Attribute>
      <Attribute ID="G423152da-ea1d-4cad-b402-8b56d0d7d538">
        <Name>Customer ID</Name>
        <DataType>String</DataType>
        <SortDirection>Ascending</SortDirection>
        <Width>5</Width>
        <DiscourageGrouping>true</DiscourageGrouping>
        <EnableDrillthrough>true</EnableDrillthrough>
        <ContextualName>Merge</ContextualName>
        <ValueSelection>Dropdown</ValueSelection>
        <Column Name="CustomerID" />
      </Attribute>
      <Role ID="G329b7ff3-e5be-441d-919a-f115ce013417">
        <!--Customer-->
        <RelatedRoleID>G747e2363-25e9-4ed7-98bd-05ef62c9abe9</RelatedRoleID>
        <Cardinality>OptionalMany</Cardinality>
        <Relation Name="dbo_Orders_FK_Orders_Customers" RelationEnd="Source" />
      </Role>
    </Fields>
    <Table Name="dbo_Customers" />
  </Entity>
  <Entity ID="G4e57ea8e-11f3-4025-9ee7-f7e0642e6bde">
    <Name>Order</Name>
    <CollectionName>Orders</CollectionName>
    <IdentifyingAttributes>
      <AttributeReference>
        <!--Order ID-->
        <AttributeID>Gdc127d64-26c5-460f-8122-ab3eff61e018</AttributeID>
      </AttributeReference>
      <AttributeReference>
        <Path>
          <RolePathItem>
            <!--Customer-->
            <RoleID>G747e2363-25e9-4ed7-98bd-05ef62c9abe9</RoleID>
          </RolePathItem>
        </Path>
        <!--Customer ID-->
        <AttributeID>G423152da-ea1d-4cad-b402-8b56d0d7d538</AttributeID>
      </AttributeReference>
    </IdentifyingAttributes>
    <DefaultDetailAttributes>
      <AttributeReference>
        <!--Order ID-->
        <AttributeID>Gdc127d64-26c5-460f-8122-ab3eff61e018</AttributeID>
      </AttributeReference>
      <AttributeReference>
        <Path>
          <RolePathItem>
            <!--Customer-->
            <RoleID>G747e2363-25e9-4ed7-98bd-05ef62c9abe9</RoleID>
          </RolePathItem>
        </Path>
```

```
          <!--Customer ID-->
          <AttributeID>G423152da-ea1d-4cad-b402-8b56d0d7d538</AttributeID>
        </AttributeReference>
      </DefaultDetailAttributes>
      <DefaultAggregateAttributes>
        <AttributeReference>
          <!--#Orders-->
          <AttributeID>G5348c7f9-d6ab-4e0a-b46b-40503022b6d7</AttributeID>
        </AttributeReference>
      </DefaultAggregateAttributes>
      <InstanceSelection>FilteredList</InstanceSelection>
      <IsLookup>true</IsLookup>
      <Fields>
        <Attribute ID="G5348c7f9-d6ab-4e0a-b46b-40503022b6d7">
          <Name>#Orders</Name>
          <DataType>Integer</DataType>
          <Expression>
            <Function>
              <FunctionName>Count</FunctionName>
              <Arguments>
                <Expression>
                  <EntityRef>
                    <!--Order-->
                    <EntityID>G4e57ea8e-11f3-4025-9ee7-f7e0642e6bde</EntityID>
                  </EntityRef>
                </Expression>
              </Arguments>
            </Function>
          </Expression>
          <IsAggregate>true</IsAggregate>
          <SortDirection>Descending</SortDirection>
          <Format>n0</Format>
          <EnableDrillthrough>true</EnableDrillthrough>
        </Attribute>
        <Attribute ID="Gdc127d64-26c5-460f-8122-ab3eff61e018">
          <Name>Order ID</Name>
          <DataType>Integer</DataType>
          <SortDirection>Descending</SortDirection>
          <Width>6</Width>
          <Format>g</Format>
          <DiscourageGrouping>true</DiscourageGrouping>
          <EnableDrillthrough>true</EnableDrillthrough>
          <ContextualName>Merge</ContextualName>
          <ValueSelection>List</ValueSelection>
          <Column Name="OrderID" />
        </Attribute>
        <Role ID="G747e2363-25e9-4ed7-98bd-05ef62c9abe9">
          <!--Orders-->
          <RelatedRoleID>G329b7ff3-e5be-441d-919a-f115ce013417</RelatedRoleID>
          <Cardinality>OptionalOne</Cardinality>
          <Relation Name="dbo_Orders_FK_Orders_Customers" RelationEnd="Target" />
        </Role>
      </Fields>
      <Table Name="dbo_Orders" />
    </Entity>
  </Entities>
  <DataSourceView xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
    <ID>NorthwindSlim</ID>
    <Name>NorthwindSlim</Name>
    <DataSourceID>NorthwindSlim</DataSourceID>
    <Schema>
      <xs:schema id="NorthwindSlim" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:msprop="urn:schemas-microsoft-
com:xml-msprop">
        <xs:element name="NorthwindSlim" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true" msprop:CompareInfo="&lt;CompareInfo&gt;&lt;Culture&gt;en-
US&lt;/Culture&gt;&lt;IgnoreCase&gt;true&lt;/IgnoreCase&gt;&lt;IgnoreKanaType&gt;true&lt;/Ign
oreKanaType&gt;&lt;IgnoreWidth&gt;true&lt;/IgnoreWidth&gt;&lt;/CompareInfo&gt;">
          <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
```

```
            <xs:element name="dbo_Customers" msprop:FriendlyName="Customers"
msprop:DbSchemaName="dbo" msprop:DbTableName="Customers" msprop:TableType="Table"
msprop:stats_RowCount="91">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="CustomerID" msprop:stats_StDevWidth="0"
msprop:stats_UniqueValuePercent="100" msprop:DbDataType="nchar" msprop:stats_MaxWidth="5"
msprop:stats_UniqueValueCount="91" msprop:stats_AvgWidth="5"
msprop:DbColumnName="CustomerID">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:maxLength value="5" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="dbo_Orders" msprop:FriendlyName="Orders"
msprop:DbSchemaName="dbo" msprop:DbTableName="Orders" msprop:TableType="Table"
msprop:stats_RowCount="830">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="OrderID" msprop:stats_StDevWidth="0"
msprop:stats_UniqueValuePercent="100" msprop:DbDataType="int" msprop:stats_MaxWidth="5"
msprop:stats_UniqueValueCount="830" msprop:stats_AvgWidth="5" msprop:DbColumnName="OrderID"
type="xs:int" />
                  <xs:element name="CustomerID" msprop:stats_StDevWidth="0"
msprop:stats_UniqueValuePercent="10" msprop:DbDataType="nchar" msprop:stats_MaxWidth="5"
msprop:stats_UniqueValueCount="89" msprop:stats_AvgWidth="5" msprop:DbColumnName="CustomerID"
minOccurs="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:maxLength value="5" />
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:choice>
        </xs:complexType>
        <xs:unique name="dbo_Customers_PK_Customers" msdata:PrimaryKey="true">
          <xs:selector xpath=".//dbo_Customers" />
          <xs:field xpath="CustomerID" />
        </xs:unique>
        <xs:unique name="dbo_Orders_PK_Orders" msdata:PrimaryKey="true">
          <xs:selector xpath=".//dbo_Orders" />
          <xs:field xpath="OrderID" />
        </xs:unique>
        <xs:keyref name="dbo_Orders_FK_Orders_Customers"
refer="dbo_Customers_PK_Customers">
          <xs:selector xpath=".//dbo_Orders" />
          <xs:field xpath="CustomerID" />
        </xs:keyref>
      </xs:element>
    </xs:schema>
  </Schema>
</DataSourceView>
</SemanticModel>

      <Role ID="G7e278a2d-57ba-43f8-9031-64beb618bb5d">
        <!--Contact-->
        <RelatedRoleID>G7816536d-a452-45dc-a20a-ed4504872d41</RelatedRoleID>
        <Cardinality>OptionalMany</Cardinality>
        <Relation Name="FK_StoreContact_Contact_ContactID" RelationEnd="Source" />
      </Role>
      <Role ID="G5a72bd81-43a5-49d7-b584-6adcc9ce3b44">
        <Name>Sales Orders</Name>
        <!--Contact-->
```

```
            <RelatedRoleID>G997c487b-a42f-47f7-afb5-f4f48b7e0f71</RelatedRoleID>
            <Cardinality>OptionalMany</Cardinality>
            <Relation Name="FK_SalesOrderHeader_Contact_ContactID" RelationEnd="Source" />
          </Role>
          <Role ID="Gb4017f2b-fb7a-4bf6-b17c-b280df5c3658">
            <!--Contact-->
            <RelatedRoleID>G2198b1f8-8d57-43a0-a7d8-7b30a36fd6a1</RelatedRoleID>
            <Cardinality>OptionalMany</Cardinality>
            <Relation Name="FK_Individual_Contact_ContactID" RelationEnd="Source" />
          </Role>
          <Role ID="G48bfe2fa-609a-47ee-99e5-013c087b5934">
            <!--Contact-->
            <RelatedRoleID>G460d3335-09fc-4d60-9bb4-3b07b604c1da</RelatedRoleID>
            <Cardinality>OptionalMany</Cardinality>
            <Relation Name="FK_ContactCreditCard_Contact_ContactID" RelationEnd="Source" />
          </Role>
          <Role ID="G4cd82b49-bbf0-4144-9675-c91b4bb57d4e">
            <!--Contact-->
            <RelatedRoleID>G84ce1a66-711d-4d85-a6a6-a3dd1362809e</RelatedRoleID>
            <Cardinality>OptionalMany</Cardinality>
            <Relation Name="FK_VendorContact_Contact_ContactID" RelationEnd="Source" />
          </Role>
          <Role ID="G19e86713-9c66-4313-8900-ad7ef3086575">
            <!--Contact-->
            <RelatedRoleID>G04d39c42-8329-4a30-835b-b92a0a7099b7</RelatedRoleID>
            <Cardinality>OptionalMany</Cardinality>
            <Relation Name="FK_Employee_Contact_ContactID" RelationEnd="Source" />
          </Role>
        </Fields>
        <Table Name="Person_Contact" />
      </Entity>
...
      <EntityFolder ID="G2e7933c5-4a48-4758-acb0-ce5fe8cab66e">
        <Name>Product Details</Name>
        <Entities>
          <Entity ID="G26a9fee6-cdcc-4799-a510-5f9633ae7620">
            <Name>Product Cost History</Name>
            <CollectionName>Product Cost Histories</CollectionName>
            <IdentifyingAttributes>
              <AttributeReference>
                <Path>
                  <RolePathItem>
                    <!--Product-->
                    <RoleID>G99ab42b5-46e5-484c-93e1-6b329ae2f050</RoleID>
                  </RolePathItem>
                </Path>
                <!--Name-->
                <AttributeID>G64343d16-f565-44c8-884c-681a8716c07e</AttributeID>
              </AttributeReference>
              <AttributeReference>
                <!--Start Date-->
                <AttributeID>G2021af9f-0e51-45ac-a8b7-9a023a727261</AttributeID>
              </AttributeReference>
              <AttributeReference>
                <!--Modified Date-->
                <AttributeID>G9d1627ff-ce0e-4115-b0c6-b754be8ada77</AttributeID>
              </AttributeReference>
            </IdentifyingAttributes>
            <DefaultDetailAttributes>
              <AttributeReference>
                <Path>
                  <RolePathItem>
                    <!--Product-->
                    <RoleID>G99ab42b5-46e5-484c-93e1-6b329ae2f050</RoleID>
                  </RolePathItem>
                </Path>
                <!--Name-->
                <AttributeID>G64343d16-f565-44c8-884c-681a8716c07e</AttributeID>
              </AttributeReference>
              <AttributeReference>
```

```
      <!--Standard Cost-->
      <AttributeID>G97882c08-7cfc-4e87-bf8a-5166f321fe6e</AttributeID>
    </AttributeReference>
    <AttributeReference>
      <!--Start Date-->
      <AttributeID>G2021af9f-0e51-45ac-a8b7-9a023a727261</AttributeID>
    </AttributeReference>
    <AttributeReference>
      <!--End Date-->
      <AttributeID>G79554b2a-81bd-42ad-9d5c-8de023209436</AttributeID>
    </AttributeReference>
  </DefaultDetailAttributes>
  <DefaultAggregateAttributes>
    <AttributeReference>
      <!--#Product Cost Histories-->
      <AttributeID>G4480065d-404f-4525-8219-f0701d7755cd</AttributeID>
    </AttributeReference>
  </DefaultAggregateAttributes>
  <InstanceSelection>Dropdown</InstanceSelection>
  <Fields>
    <Attribute ID="G4480065d-404f-4525-8219-f0701d7755cd">
      <Name>#Product Cost Histories</Name>
      <DataType>Integer</DataType>
      <Expression>
        <Function>
          <FunctionName>Count</FunctionName>
          <Arguments>
            <Expression>
              <EntityRef>
                <!--Product Cost History-->
                <EntityID>G26a9fee6-cdcc-4799-a510-5f9633ae7620</EntityID>
              </EntityRef>
            </Expression>
          </Arguments>
        </Function>
      </Expression>
      <IsAggregate>true</IsAggregate>
      <SortDirection>Descending</SortDirection>
      <Format>n0</Format>
      <EnableDrillthrough>true</EnableDrillthrough>
    </Attribute>
    <Role ID="G99ab42b5-46e5-484c-93e1-6b329ae2f050">
      <!--Product Cost Histories-->
      <RelatedRoleID>G47fe60f4-71c8-444f-886a-455dcf275efc</RelatedRoleID>
      <Cardinality>One</Cardinality>
      <Relation Name="FK_ProductCostHistory_Product_ProductID"
                RelationEnd="Target" />
    </Role>
    <Attribute ID="G2021af9f-0e51-45ac-a8b7-9a023a727261">
      <Name>Start Date</Name>
      <DataType>DateTime</DataType>
      <SortDirection>Descending</SortDirection>
      <Format>d</Format>
      <EnableDrillthrough>true</EnableDrillthrough>
      <ValueSelection>Dropdown</ValueSelection>
      <Variations>
        <Attribute ID="G1939ef5e-9069-4793-83f1-be12286a20d5">
          <Name>Start Day</Name>
          <DataType>Integer</DataType>
          <Nullable>true</Nullable>
          <Expression>
            <Function>
              <FunctionName>Day</FunctionName>
              <Arguments>
                <Expression>
                  <AttributeRef>
                    <!--Start Date-->
                    <AttributeID>G2021af9f-0e51-45ac-a8b7-9a023a727261</AttributeID>
                  </AttributeRef>
                </Expression>
```

```
                    </Arguments>
                  </Function>
                </Expression>
                <SortDirection>Ascending</SortDirection>
              </Attribute>
...
            </Variations>
            <Column Name="StartDate" />
          </Attribute>
          <Attribute ID="G79554b2a-81bd-42ad-9d5c-8de023209436">
            <Name>End Date</Name>
            <DataType>DateTime</DataType>
            <Nullable>true</Nullable>
            <SortDirection>Descending</SortDirection>
            <Format>d</Format>
            <ValueSelection>Dropdown</ValueSelection>
            <Variations>
              <Attribute ID="G77f4e91a-b380-4fee-8f6d-5dab7e5aade9">
                <Name>End Day</Name>
                <DataType>Integer</DataType>
                <Nullable>true</Nullable>
                <Expression>
                  <Function>
                    <FunctionName>Day</FunctionName>
                    <Arguments>
                      <Expression>
                        <AttributeRef>
                          <!--End Date-->
                          <AttributeID>G79554b2a-81bd-42ad-9d5c-8de023209436</AttributeID>
                        </AttributeRef>
                      </Expression>
                    </Arguments>
                  </Function>
                </Expression>
                <SortDirection>Ascending</SortDirection>
              </Attribute>
...
            </Variations>
            <Column Name="EndDate" />
          </Attribute>
          <Attribute ID="G97882c08-7cfc-4e87-bf8a-5166f321fe6e">
            <Name>Standard Cost</Name>
            <DataType>Decimal</DataType>
            <SortDirection>Descending</SortDirection>
            <Width>9</Width>
            <Format>g</Format>
            <!--Sum Standard Cost-->
            <DefaultAggregateAttributeID>Gc47a9596-49e7-44a9-90f0-
4692b8b9b36f</DefaultAggregateAttributeID>
            <ValueSelection>Dropdown</ValueSelection>
            <Variations>
              <Attribute ID="Gc47a9596-49e7-44a9-90f0-4692b8b9b36f">
                <Name>Sum Standard Cost</Name>
                <DataType>Decimal</DataType>
                <Nullable>true</Nullable>
                <Expression>
                  <Function>
                    <FunctionName>Sum</FunctionName>
                    <Arguments>
                      <Expression>
                        <AttributeRef>
                          <!--Standard Cost-->
                          <AttributeID>G97882c08-7cfc-4e87-bf8a-5166f321fe6e</AttributeID>
                        </AttributeRef>
                      </Expression>
                    </Arguments>
                  </Function>
                </Expression>
                <IsAggregate>true</IsAggregate>
                <SortDirection>Descending</SortDirection>
```

```
                  <EnableDrillthrough>true</EnableDrillthrough>
                </Attribute>
...
              </Variations>
              <Column Name="StandardCost" />
            </Attribute>
...
          </Fields>
          <Table Name="Production_ProductCostHistory" />
        </Entity>
        <Entity ID="G054a4c10-daa5-4399-bce2-fe646ff72bc3">
          <Name>Product Description</Name>
          <CollectionName>Product Descriptions</CollectionName>
          <IdentifyingAttributes>
            <AttributeReference>
              <!--Description-->
              <AttributeID>G0a7800c9-5772-4678-85ba-789af2d1cb54</AttributeID>
            </AttributeReference>
          </IdentifyingAttributes>
          <DefaultDetailAttributes>
            <AttributeReference>
              <!--Description-->
              <AttributeID>G0a7800c9-5772-4678-85ba-789af2d1cb54</AttributeID>
            </AttributeReference>
          </DefaultDetailAttributes>
          <DefaultAggregateAttributes>
            <AttributeReference>
              <!--#Product Descriptions-->
              <AttributeID>G50022fe3-d027-4032-a517-8b8a761069d7</AttributeID>
            </AttributeReference>
          </DefaultAggregateAttributes>
          <InstanceSelection>FilteredList</InstanceSelection>
          <IsLookup>true</IsLookup>
          <Fields>
            <Attribute ID="G50022fe3-d027-4032-a517-8b8a761069d7">
              <Name>#Product Descriptions</Name>
              <DataType>Integer</DataType>
              <Expression>
                <Function>
                  <FunctionName>Count</FunctionName>
                  <Arguments>
                    <Expression>
                      <EntityRef>
                        <!--Product Description-->
                        <EntityID>G054a4c10-daa5-4399-bce2-fe646ff72bc3</EntityID>
                      </EntityRef>
                    </Expression>
                  </Arguments>
                </Function>
              </Expression>
              <IsAggregate>true</IsAggregate>
              <SortDirection>Descending</SortDirection>
              <Format>n0</Format>
              <EnableDrillthrough>true</EnableDrillthrough>
            </Attribute>
            <Attribute ID="G933ad478-d646-4b4c-9060-028fa37ac996">
              <Name>Product Description ID</Name>
              <Hidden>true</Hidden>
              <DataType>Integer</DataType>
              <SortDirection>Descending</SortDirection>
              <Width>4</Width>
              <Format>g</Format>
              <DiscourageGrouping>true</DiscourageGrouping>
              <Column Name="ProductDescriptionID" />
            </Attribute>
            <Attribute ID="G0a7800c9-5772-4678-85ba-789af2d1cb54">
              <Name>Description</Name>
              <DataType>String</DataType>
              <Width>167</Width>
              <DiscourageGrouping>true</DiscourageGrouping>
```

```
              <EnableDrillthrough>true</EnableDrillthrough>
              <ContextualName>Role</ContextualName>
              <Column Name="Description" />
            </Attribute>
            <Attribute ID="Gc0947d13-121e-4f89-9596-2c2ffd1685ca">
              <Name>Rowguid</Name>
              <Hidden>true</Hidden>
              <DataType>String</DataType>
              <Width>36</Width>
              <DiscourageGrouping>true</DiscourageGrouping>
              <Column Name="rowguid" />
            </Attribute>
            <Attribute ID="Ge4fe0ed0-4e80-4404-915a-5f3449460e20">
              <Name>Modified Date</Name>
              <Hidden>true</Hidden>
              <DataType>DateTime</DataType>
              <SortDirection>Descending</SortDirection>
              <Format>d</Format>
              <ValueSelection>Dropdown</ValueSelection>
              <Variations>
                <Attribute ID="Ga7bf1945-d84e-4956-ba4d-2aeeb2722b28">
                  <Name>Modified Day</Name>
                  <DataType>Integer</DataType>
                  <Nullable>true</Nullable>
                  <Expression>
                    <Function>
                      <FunctionName>Day</FunctionName>
                      <Arguments>
                        <Expression>
                          <AttributeRef>
                            <!--Modified Date-->
                            <AttributeID>Ge4fe0ed0-4e80-4404-915a-5f3449460e20</AttributeID>
                          </AttributeRef>
                        </Expression>
                      </Arguments>
                    </Function>
                  </Expression>
                </Attribute>
...
              </Variations>
              <Column Name="ModifiedDate" />
            </Attribute>
            <Role ID="Gcbe16f38-f954-4730-b333-51eec4d83b3b">
              <!--Product Description-->
              <RelatedRoleID>Gbf14d8fb-4087-4085-a350-8aa6cfbc98ad</RelatedRoleID>
              <Cardinality>OptionalMany</Cardinality>
              <Relation
Name="FK_ProductModelProductDescriptionCulture_ProductDescription_ProductDescriptionID"
RelationEnd="Source" />
            </Role>
          </Fields>
          <Table Name="Production_ProductDescription" />
        </Entity>
...
      </Entities>
    </EntityFolder>
...

  </Entities>
  <Perspectives>
    <Perspective ID="G33871e88-dc21-4242-9e45-01d751069a52">
      <Name>Individual Customer Sales</Name>
      <Description>This perspective provides a small subset of Adventure Works focused only
on sales to individuals.</Description>
      <ModelItems>
        <!--Contact-->
        <ModelItemID>G993b08ba-b548-428b-bfab-7823a87640e1</ModelItemID>
        <!--Individuals-->
        <ModelItemID>Gb4017f2b-fb7a-4bf6-b17c-b280df5c3658</ModelItemID>
        <!--Individual-->
```

```
          <ModelItemID>G467deb7e-dc58-43c4-9f7f-87549f69c1af</ModelItemID>
          <!--#Individuals-->
          <ModelItemID>Gf6cf817c-fc9e-4629-9684-649485d2e735</ModelItemID>
          <!--Contact-->
          <ModelItemID>G2198b1f8-8d57-43a0-a7d8-7b30a36fd6a1</ModelItemID>
          <!--Modified Date-->
          <ModelItemID>G9eb16772-e96f-4ac4-9786-ffde3cd6809f</ModelItemID>
          <!--Modified Day-->
          <ModelItemID>G0003c639-b77e-46eb-a864-f237fa649769</ModelItemID>
          <!--Modified Month-->
          <ModelItemID>G76eed1ac-ccf2-40e7-92c3-6802d41302a6</ModelItemID>
          <!--Modified Year-->
          <ModelItemID>Gecba9972-9652-4277-8c2f-c2ac7f0d320a</ModelItemID>
          <!--Modified Quarter-->
          <ModelItemID>G96f57111-e63e-4974-b398-1072839daabc</ModelItemID>
          <!--First Modified Date-->
          <ModelItemID>Ged6f836b-6374-42d1-979e-981af5cc65c6</ModelItemID>
          <!--Last Modified Date-->
          <ModelItemID>G49578624-1998-496e-ac3a-1212dfb16f95</ModelItemID>
          <!--Product-->
          <ModelItemID>G73aed554-12c3-4750-8981-997fe69705ab</ModelItemID>
          <!--Sales-->
          <ModelItemID>G75db5378-b40a-49a6-8fac-53b563f471d8</ModelItemID>
          <!--Sale-->
          <ModelItemID>G7d048573-fb87-42c3-9660-f171d6645f3a</ModelItemID>
          <!--#Sales -->
          <ModelItemID>G8adb941a-9e66-4373-9dfe-f3c44285e6d8</ModelItemID>
          <!--Sales Order-->
          <ModelItemID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</ModelItemID>
          <!--Sales ID-->
          <ModelItemID>G530377c5-2430-4f59-8c0e-e126375b0f79</ModelItemID>
          <!--Carrier Tracking Number-->
          <ModelItemID>G29ce27c4-26a9-4541-bebd-bf6560faec77</ModelItemID>
          <!--Order Qty-->
          <ModelItemID>Gadadf016-6677-4e7d-b068-bfb6cdbd846f</ModelItemID>
          <!--Sum Order Qty-->
          <ModelItemID>G52bef145-cd14-4c05-bc89-228a3ab8065c</ModelItemID>
...
        </ModelItems>
      </Perspective>
    </Perspectives>
    <DataSourceView xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
      <ID>Adventure Works</ID>
      <Name>Adventure Works</Name>
      <CreatedTimestamp>0001-01-01T08:00:00Z</CreatedTimestamp>
      <LastSchemaUpdate>0001-01-01T08:00:00Z</LastSchemaUpdate>
      <Annotations xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
        <Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowFriendlyNames</Name>
          <Value>true</Value>
        </Annotation>
        <Annotation>
          <Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:CurrentLayout</Name>
          <Value>_ALL_TABLES_</Value>
        </Annotation>
...
      </Annotations>
      <DataSourceID>Adventure Works</DataSourceID>
      <Schema>
        <xs:schema id="Adventure_x0020_Works" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-
msdata" xmlns:msprop="urn:schemas-microsoft-com:xml-msprop">
...
        </xs:schema>
      </Schema>
    </DataSourceView>
  </SemanticModel>
```

3.1.2 Entity

The following XML is an example of an Entity element that specifies an entity in the semantic model. The **ID** attribute of the **Entity** element specifies the QName value of G333db059-a14a-46db-b58e-9ae10efc0f07 as a unique identifier that can be used to refer to this entity from elsewhere in the SemanticModel and SemanticQuery elements.

The **Name** element defines the name of the **Entity** as "Sales Order". The **CollectionName** element defines "Sales Order Headers" as the phrase to use to refer to a collection of instances of the entity.

```
<Entity ID="G333db059-a14a-46db-b58e-9ae10efc0f07">
  <Name>Sales Order</Name>
  <CollectionName>Sales Order Headers</CollectionName>
```

The following IdentifyingAttributes element specifies a collection of attributes for client applications to use to identify an instance of the entity to the user. In this example, the collection has only the single member AttributeReference. The **AttributeReference** element specifies a reference to an Attribute element. This reference is specified by the value of the **AttributeID** element, which is the **QName** value G5ca52962-8edd-40c7-a335-a235601f49f5. This value appears as the value of an Attribute.ID attribute for an **Attribute** element in the Entity.Fields collection of the "Sales Order" **Entity** of this example. That **Attribute** has the name "Sales Order Number". Thus the meaning of the **IdentifyingAttributes** element in this example is that an instance of the "Sales Order" **Entity** is identified to the user by the value of the "Sales Order Number" **Attribute**.

```
<IdentifyingAttributes>
  <AttributeReference>
    <!--Sales Order Number-->
    <AttributeID>G5ca52962-8edd-40c7-a335-a235601f49f5</AttributeID>
  </AttributeReference>
</IdentifyingAttributes>
```

The DefaultDetailAttributes element specifies a collection of attributes that client applications show when the parent **Entity** element is displayed. In this example, the collection has one **AttributeReference** member. As in the preceding explanation, **AttributeReference** elements reference **Attribute** elements via their **AttributeID** child elements. The meaning of the **DefaultDetailAttributes** in this example is that when the "Sales Order" **Entity** is displayed, the "Sales Order Number" **Attribute** is to be shown.

```
<DefaultDetailAttributes>
  <AttributeReference>
    <!--Sales Order Number-->
    <AttributeID>G5ca52962-8edd-40c7-a335-a235601f49f5</AttributeID>
  </AttributeReference>
</DefaultDetailAttributes>
```

The following **DefaultDetailAttributes** element specifies a collection of aggregate attributes that client applications show when the parent **Entity** element is displayed as an aggregate. In this example, the collection has one **AttributeReference** member. As in the preceding explanation, **AttributeReference** elements reference **Attribute** elements via their **AttributeID** child elements. The meaning of the **DefaultDetailAttributes** in this example is that when the "Sales Order" **Entity** is displayed as an aggregate, the "#Sales Orders" **Attribute** element is to be shown.

```
<DefaultAggregateAttributes>
  <AttributeReference>
    <!--#Sales Orders-->
    <AttributeID>G2818a809-2ea8-4a14-9591-75b52f4188e8</AttributeID>
  </AttributeReference>
```

```
</DefaultAggregateAttributes>
```

The following SortAttributes element specifies a collection of **SortAttribute** elements that specify attributes on which to sort when a sort is requested by the entity. In this example, the collection has a single **SortAttribute** member that contains an **AttributeReference** element. As in the preceding explanation, the **AttributeReference** element references an **Attribute** element via its **AttributeID** element. After looking up the names of the referenced **Attribute** element, the meaning of the **SortAttributes** element in this example is that when the "Sales Order" **Entity** is sorted, the "Sales Order Number" **Attribute** is to be used to sort the **Entity**.

```
<SortAttributes>
  <SortAttribute>
    <AttributeReference>
      <!--Sales Order Number-->
      <AttributeID>G5ca52962-8edd-40c7-a335-a235601f49f5</AttributeID>
    </AttributeReference>
    <SortDirection>Descending</SortDirection>
  </SortAttribute>
</SortAttributes>
```

The following Entity.InstanceSelection element has the value "MandatoryFilter", which specifies that when the client application selects instances of the entity, the client application uses a filter because the number of instances is so large that users are to be prohibited from building queries on this entity without filtering the results. Following the **Entity.InstanceSelection** is the Fields element. The **Fields** element specifies a collection of **Attribute**s and Roles that are associated with the "Sales Order" **Entity**.

```
<InstanceSelection>MandatoryFilter</InstanceSelection>
<Fields>
```

The first item in the **Fields** collection is the **Attribute** with **ID** G2818a809-2ea8-4a14-959-75b52f4188e8 and **Name** "#Sales Orders". The Attribute.DataType element with value "Integer" specifies that the data type of this attribute is Integer. The Expression element specifies an expression that is used to calculate the value of the attribute, which makes this a calculated **Attribute**. For examples of the **Expression** element, see sections 3.1.10 through 3.1.13. The expression uses the Aggregate Function: Count to return the count of "Sales Orders", which is the **Entity** that the EntityRef element is referring to.

The Attribute.IsAggregate element has a value of true, which specifies that this attribute is an aggregate attribute. The Attribute.SortDirection element has a value of "Descending", which specifies that groups of this attribute are to be sorted from most "Sales Orders" to least "Sales Orders". The Attribute.Format element has a value of "n0", which specifies that the value of attribute instances is to be displayed in numeric format with 0 decimal places. The Attribute.EnableDrillthrough element has a value of true, which specifies that the client application is to provide a drillthrough link to the containing entity of the attribute.

```
<Attribute ID="G2818a809-2ea8-4a14-9591-75b52f4188e8">
  <Name>#Sales Orders</Name>
  <DataType>Integer</DataType>
  <Expression>
    <Function>
      <FunctionName>Count</FunctionName>
      <Arguments>
        <Expression>
          <EntityRef>
            <!--Sales Order-->
            <EntityID>G333db059-a14a-46db-b58e-9ae10efc0f07</EntityID>
          </EntityRef>
```

```
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <IsAggregate>true</IsAggregate>
  <SortDirection>Descending</SortDirection>
  <Format>n0</Format>
  <EnableDrillthrough>true</EnableDrillthrough>
</Attribute>
```

The second item in the **Fields** collection is the **Attribute** with **ID** G5ca52962-8edd-40c7-a335-a235601f49f5 and **Name** "Sales Order Number". The **Attribute.DataType** element with a value of "String" specifies that the data type of this attribute is **String**. The **Attribute.SortDirection** element has a value of "Ascending", which specifies that instances of this attribute are to be sorted from lowest "Sales Order Number" to highest "Sales Order Number" in lexical order. The Attribute.Width element has a value of 7, which specifies that the "Sales Order Number" is displayed in a column the width of seven characters. The Attribute.DiscourageGrouping element has a value of true, which specifies that the client application is to discourage the user from grouping on this attribute. The Attribute.EnableDrillthrough element has a value of true, which specifies that the client application is to provide a drillthrough link to the containing entity of the attribute. The Column element specifies a binding of the **Attribute** to a column in the underlying database. The column is specified by its name, which is " SalesOrderNumber" in this case.

```
<Attribute ID="G5ca52962-8edd-40c7-a335-a235601f49f5">
  <Name>Sales Order Number</Name>
  <DataType>String</DataType>
  <SortDirection>Ascending</SortDirection>
  <Width>7</Width>
  <DiscourageGrouping>true</DiscourageGrouping>
  <EnableDrillthrough>true</EnableDrillthrough>
  <Column Name="SalesOrderNumber" />
</Attribute>
```

Next in the **Fields** collection is the **Role** element with **ID** G47dff9f6-a62b-4b60-b084-b077d528fed8 and **Name** "Sales". See sections 3.1.7 and 3.1.8 for a more detailed explanation of **Role** examples. This **Role** element specifies a relationship between the containing "Sales Order" **Entity** element and a target **Entity** element, which is not listed in this example, whose name is "Sale".

The Linguistics element specifies "Sales" as both the singular and plural names of the role. The Role.RelatedRoleID element specifies the **QName** value Gd807fa2d-8de1-4304-90d7-ea92ecf54d77 as a reference to the corresponding role on the target entity. The corresponding role, also known as the related role, is a role that relates the same two entities as the original role, but relates them in the opposite order. The containing entity of the original role is the target entity of the related role and vice versa. In this case, the related **Role** element specifies a relationship in the reversed order from its containing "Sale" **Entity** to its target "Sales Order" **Entity**, which is the containing element of this **Role**.

The Role.Cardinality element specifies the cardinality of the role. In this case, "OptionalMany" specifies that the cardinality of the role is 0 or more. The Relation element specifies a binding of the **Role** to a relation in the underlying database. The relation is specified by its name, which is "FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID". The **Role** is bound to the source end of the relation as specified by the value of the Relation.RelationEnd attribute, which is "Source".

```
<Role ID="G47dff9f6-a62b-4b60-b084-b077d528fed8">
  <Name>Sales</Name>
  <Linguistics>
    <SingularName>Sales</SingularName>
    <PluralName>Sales</PluralName>
  </Linguistics>
  <!--Sales Order-->
```

```
          <RelatedRoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RelatedRoleID>
          <Cardinality>OptionalMany</Cardinality>
          <Relation Name="FK_SalesOrderDetail_SalesOrderHeader_SalesOrderID"
RelationEnd="Source" />
        </Role>
...
```

Next in the **Fields** collection is another **Role** element with **ID** G997c487b-a42f-47f7-afb5-f4f48b7e0f71. See sections 3.1.7 and 3.1.8 for a more detailed explanation of **Role** examples. This **Role** element specifies a relationship between the containing "Sales Order" **Entity** element and another **Entity** element, which is not listed in this example, whose name is "Customer".

The **Role.RelatedRoleID** element specifies the **QName** value G5a72bd81-43a5-49d7-b584-6adcc9ce3b44 as a reference to the corresponding role on the target entity. The corresponding role, also known as the related role, is a role that relates the same two entities as the original role, but relates them in the opposite order. The containing entity of the original role is the target entity of the related role and vice versa. In this case, the related **Role** element specifies a relationship in the reversed order from its containing "Customer" **Entity** to its target "Sales Order" **Entity**, which is the containing element of this **Role**.

The **Role.Cardinality** element specifies the cardinality of the role. In this case, "One" specifies that the cardinality of the role is one to one. The HiddenFields element specifies a collection of references to field folder items that are **Attribute**, **Role**, and FieldFolder elements. Each reference is specified by the **QName** value of the **FieldFolderItemID** elements. These field folder items are items that the client application does not display for the role's target entity when the role is used to reach the entity.

The **Relation** element specifies a binding of the **Role** to a relation in the underlying database. The relation is specified by its name, which is "FK_SalesOrderHeader_Contact_ContactID" in this case. The **Role** is bound to the target end of the relation, as specified by the value of the **Relation.RelationEnd** attribute, which is "Target" in this case.

```
        <Role ID="G997c487b-a42f-47f7-afb5-f4f48b7e0f71">
          <!--Sales Orders-->
          <RelatedRoleID>G5a72bd81-43a5-49d7-b584-6adcc9ce3b44</RelatedRoleID>
          <Cardinality>One</Cardinality>
          <HiddenFields>
            <!--Store Contacts-->
            <FieldFolderItemID>G7e278a2d-57ba-43f8-9031-64beb618bb5d</FieldFolderItemID>
            <!--Individuals-->
            <FieldFolderItemID>Gb4017f2b-fb7a-4bf6-b17c-b280df5c3658</FieldFolderItemID>
            <!--Vendor Contacts-->
            <FieldFolderItemID>G4cd82b49-bbf0-4144-9675-c91b4bb57d4e</FieldFolderItemID>
            <!--Employees-->
            <FieldFolderItemID>G19e86713-9c66-4313-8900-ad7ef3086575</FieldFolderItemID>
          </HiddenFields>
          <Relation Name="FK_SalesOrderHeader_Contact_ContactID" RelationEnd="Target" />
        </Role>
```

This is the end of the **Fields** collection. The Table element specifies the name of a table, called "Sales_SalesOrderHeader", in the underlying data source view (DSV). This binds the "Sales Order" **Entity** to the database table.

```
      </Fields>
      <Table Name="Sales_SalesOrderHeader" />
    </Entity>
```

3.1.3 Attribute 1: Simple Attribute

The following is a simple example of an Attribute element. It has an **ID** attribute with a QName value of G64343d16-f565-44c8-884c-681a8716c07e, which associates a globally unique identifier (GUID) with this **Attribute**. The **Name** of this **Attribute** is "Name", which is the value of the Attribute.Name element. The Attribute.DataType element with value "String" specifies that the data type of this attribute is String. This **Attribute** is a member of the Fields collection of an Entity whose name is "Product". Thus, this attribute represents the "name of a product".

The Attribute.SortDirection element has a value of "Ascending", which specifies that instances of this attribute are sorted in alphabetical order. The Attribute.Width element has a value of 30, which specifies that the "Sales Order ID" is displayed in a column that has a width of 30 characters. The Attribute.DiscourageGrouping element has a value of true, which specifies that the client application prevents the user from grouping on this attribute. The Attribute.EnableDrillthrough element has a value of true, which specifies that the client application provides a drillthrough link to the containing entity of the attribute.

The Attribute.ContextualName element has a value of "Role", which specifies that the client application uses the role name as the contextual name if the attribute is the single identifying attribute of the entity. Otherwise, the "Role" value specifies that the client application generates a contextual name that is the same as if "Merge" was specified. The Attribute.ValueSelection element has a value of "List", which specifies that the number of unique instance values of this attribute is too large for a simple dropdown and requires a larger list-based UI in the client application. The Column element specifies a binding of the **Attribute** to a column in the underlying database. The column is specified by its name, which is "Name" in this case.

```
<Attribute ID="G64343d16-f565-44c8-884c-681a8716c07e">
  <Name>Name</Name>
  <DataType>String</DataType>
  <SortDirection>Ascending</SortDirection>
  <Width>30</Width>
  <DiscourageGrouping>true</DiscourageGrouping>
  <EnableDrillthrough>true</EnableDrillthrough>
  <ContextualName>Role</ContextualName>
  <ValueSelection>List</ValueSelection>
  <Column Name="Name" />
</Attribute>
```

3.1.4 Attribute 2: Calculated Attribute

The following is another example of an Attribute element. This **Attribute** element is of a calculated **Attribute**. It has an **ID** attribute with a QName value of Gb4a8fe12-4ec9-4fa9-8b33-82a9acfea68c, which associates a GUID with this **Attribute**. The **Name** of this **Attribute** is "Order Year", which is the value of the Attribute.Name element. The Attribute.DataType element with a value of "Integer" specifies that the data type of this attribute is Integer. This **Attribute** element is taken from a member of the Variations collection of an **Attribute** element in the Fields collection of an Entity whose name is "Sales Order". Thus, this attribute represents the "order year of a sales order". The Attribute.Nullable element with a value of true specifies that the attribute can have a value of NULL.

The Expression element specifies an expression that is used to calculate the value of the attribute, which makes this a calculated **Attribute**. For examples of the **Expression** element, see sections 3.1.10 through 3.1.13. The expression uses the scalar Year function to return the integer year of the "Order Date" attribute, which is the **Attribute** that the AttributeRef element is referring to.

The Attribute.SortDirection element has a value of "Ascending", which specifies that instances of this attribute are sorted in ascending numerical order from earliest "Order Year" to the latest. The Attribute.ValueSelection element has a value of "Dropdown", which specifies that the number of unique instance values is small enough to fit in a simple dropdown list.

```
<Attribute ID="Gb4a8fe12-4ec9-4fa9-8b33-82a9acfea68c">
  <Name>Order Year</Name>
  <DataType>Integer</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Year</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <SortDirection>Ascending</SortDirection>
  <ValueSelection>Dropdown</ValueSelection>
</Attribute>
```

3.1.5 Attribute 3: Aggregate Attribute

The following is another example of an Attribute element. This **Attribute** element is of an aggregate **Attribute**. It has an **ID** attribute with a QName value of G2818a809-2ea8-4a14-9591-75b52f4188e8, which associates a GUID with this **Attribute**. The **Name** of this **Attribute** is "#Sales Orders", which is the value of the Attribute.Name element. The Attribute.DataType element with a value of "Integer" specifies that the data type of this attribute is Integer. The Expression element specifies an expression that is used to calculate the value of the attribute, which makes this a calculated **Attribute**. For examples of the **Expression** element, see sections 3.1.10 through 3.1.13. The expression uses the aggregate Count function to return the count of "Sales Order", which is the Entity that the EntityRef element is referring to.

The Attribute.IsAggregate element has a value of true, which specifies that this attribute is an aggregate attribute. The Attribute.SortDirection element has a value of "Descending", which specifies that groups of this attribute are to be sorted from most "Sales Orders" to least "Sales Orders". The Attribute.Format element has a value of "n0", which specifies that the values of the attribute instances are displayed in a numeric format with 0 decimal places. The Attribute.EnableDrillthrough element has a value of true, which specifies that the client application is to provide a drillthrough link to the containing entity of the attribute.

```
<Attribute ID="G2818a809-2ea8-4a14-9591-75b52f4188e8">
  <Name>#Sales Orders</Name>
  <DataType>Integer</DataType>
  <Expression>
    <Function>
      <FunctionName>Count</FunctionName>
      <Arguments>
        <Expression>
          <EntityRef>
            <!--Sales Order-->
            <EntityID>G333db059-a14a-46db-b58e-9ae10efc0f07</EntityID>
          </EntityRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <IsAggregate>true</IsAggregate>
  <SortDirection>Descending</SortDirection>
  <Format>n0</Format>
  <EnableDrillthrough>true</EnableDrillthrough>
</Attribute>
```

3.1.6 Variations

The following is an example of a Variations element, which defines a collection of Attribute elements that are considered variations of the parent element. In this example, the parent element is an **Attribute**. First, the parent **Attribute** element is shown, and then a few lines later the **Variations** element begins. The parent element is an attribute that specifies the "Order Date" of the "Sales Order" entity as an attribute that has a data type of DateTime.

```
<Attribute ID="G1f88a445-ae2b-497e-8b0b-3f797fbc868c">
  <Name>Order Date</Name>
  <DataType>DateTime</DataType>
  <SortDirection>Descending</SortDirection>
  <Width>5</Width>
  <Format>d</Format>
  <Variations>
```

The first member of the **Variations** collection, which follows, is an **Attribute** whose name is "Order Day" and whose data type is Integer. This is a calculated attribute, and the calculation is given by the Expression, which calls the scalar function Day with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, extracts the day from the attribute value, and returns the day of the month as an **Integer**. Thus, the variation defines the "Order Day" **Attribute** as taking the day of the "Order Date" **Attribute** and returning it as an **Integer**.

```
<Attribute ID="G479ca591-f52f-4950-9084-bd62f1830e99">
  <Name>Order Day</Name>
  <DataType>Integer</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Day</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <SortDirection>Ascending</SortDirection>
  <ValueSelection>Dropdown</ValueSelection>
</Attribute>
```

The second member of the **Variations** collection, which follows, is an **Attribute** whose name is "Order Month" and whose data type is **Integer**. This is a calculated attribute, and the calculation is given by the **Expression**, which calls the scalar function Month with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, extracts the month from the attribute value, and returns the month as an **Integer**. Thus, the variation defines the "Order Month" **Attribute** as taking the month of the "Order Date" **Attribute** and returning it as an **Integer**.

```
<Attribute ID="G88c8374b-78c7-4729-9fe7-719499d8f6fc">
  <Name>Order Month</Name>
  <DataType>Integer</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Month</FunctionName>
      <Arguments>
```

```
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <SortDirection>Ascending</SortDirection>
  <ValueSelection>Dropdown</ValueSelection>
</Attribute>
```

The third member of the **Variations** collection, which follows, is an **Attribute** whose name is "Order Year" and whose data type is **Integer**. This is a calculated attribute, and the calculation is given by the **Expression** that calls the scalar function Year with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, extracts the year from the attribute value, and returns the year as an **Integer**. Thus, the variation defines the "Order Year" **Attribute** as taking the year of the "Order Date" **Attribute** and returning it as an **Integer**.

```
<Attribute ID="Gb4a8fe12-4ec9-4fa9-8b33-82a9acfea68c">
  <Name>Order Year</Name>
  <DataType>Integer</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Year</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <SortDirection>Ascending</SortDirection>
  <ValueSelection>Dropdown</ValueSelection>
</Attribute>
```

The fourth member of the **Variations** collection, which follows, is an **Attribute** whose name is "Order Quarter" and whose data type is **Integer**. This is a calculated attribute, and the calculation is given by the **Expression** that calls the scalar function Quarter with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, extracts the quarter from the attribute value, and returns the quarter as an **Integer**. Thus, the variation defines the "Order Quarter" **Attribute** as taking the quarter of the "Order Date" **Attribute** and returning it as an **Integer**.

```
<Attribute ID="Gd7e9b121-ae15-44be-9fec-0a0265c133f4">
  <Name>Order Quarter</Name>
  <DataType>Integer</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Quarter</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
```

```
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <SortDirection>Ascending</SortDirection>
</Attribute>
```

The fifth member of the **Variations** collection, which follows, is an **Attribute** whose name is "First Order Date" and whose data type is **DateTime**. This is a calculated attribute, and the calculation is given by the **Expression**, which calls the aggregate function Min with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, finds the minimal date in the set of instance values, and returns the minimum as a **DateTime**. Thus, the variation defines the "First Order Date" **Attribute** as taking the minimum of the "Order Date" **Attribute** and returning it as a **DateTime**.

```
<Attribute ID="G0d6ff6f6-c971-45c9-ae27-f698c7a9224c">
  <Name>First Order Date</Name>
  <DataType>DateTime</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Min</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
  </Expression>
  <IsAggregate>true</IsAggregate>
  <SortDirection>Descending</SortDirection>
  <Format>d</Format>
  <EnableDrillthrough>true</EnableDrillthrough>
</Attribute>
```

The sixth and last member of the **Variations** collection, which follows, is an **Attribute** whose name is "Last Order Date" and whose data type is **DateTime**. This is a calculated attribute, and the calculation is given by the **Expression** which calls the aggregate function Max with an argument referencing the parent element of the **Variations** collection, which is the "Order Date" **Attribute**. This function takes the "Order Date" **Attribute**, which has the **DateTime** data type, finds the maximum date in the set of instance values, and returns the maximum as a **DateTime**. Thus, the variation defines the "Last Order Date" **Attribute** as taking the maximum of the "Order Date" **Attribute** and returning it as a **DateTime**.

```
<Attribute ID="Gd78ec086-f2b9-4ab4-86c2-6136be989d59">
  <Name>Last Order Date</Name>
  <DataType>DateTime</DataType>
  <Nullable>true</Nullable>
  <Expression>
    <Function>
      <FunctionName>Max</FunctionName>
      <Arguments>
        <Expression>
          <AttributeRef>
            <!--Order Date-->
            <AttributeID>G1f88a445-ae2b-497e-8b0b-3f797fbc868c</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
```

```
        </Function>
      </Expression>
      <IsAggregate>true</IsAggregate>
      <SortDirection>Descending</SortDirection>
      <Format>d</Format>
      <EnableDrillthrough>true</EnableDrillthrough>
    </Attribute>
  </Variations>
  <Column Name="OrderDate" />
</Attribute>
```

3.1.7 Role 1: Role with Linguistics

The following is an example of a Role element, which specifies the relationship between an entity and a related entity. The **Role** has an **ID** attribute with a QName value of G0a9cb3d9-f4b2-48a6-8c8d-6324f65a3f26, which associates a GUID with this **Role**. The **Name** of this **Role** is "Sales Orders", which is the value of the Role.Name element.

The singular and plural forms of the noun representing this **Role** are given by the Linguistics.SingularName and Linguistics.PluralName elements as the nouns "Sales Order" and "Sales Orders", in this case.

The Role.RelatedRoleID element specifies a reference to the corresponding role on the target entity. The corresponding role, also known as the related role, is a role that relates the same two entities as the original role, but relates them in the opposite order. The containing entity of the original role is the target entity of the related role, and vice versa. The related role for this example is given in section 3.1.8. The Role.Cardinality element specifies the cardinality of the role. In this case, "OptionalMany" specifies that the cardinality of the role is 0 or more.

The Relation element specifies a binding of the **Role** to a relation in the underlying database. The relation is specified by its name, which is "FK_SalesOrderHeader_SalesTerritory_TerritoryID" in this case. The **Role** is bound to the source end of the relation as specified by the value of the Relation.RelationEnd attribute, which is "Source". **Role** elements occur in pairs, with one role bound to the source end of the relation and the other role bound to the target end of the same relation.

```
<Role ID="G0a9cb3d9-f4b2-48a6-8c8d-6324f65a3f26">
  <Name>Sales Orders</Name>
  <Linguistics>
    <SingularName>Sales Order</SingularName>
    <PluralName>Sales Orders</PluralName>
  </Linguistics>
  <!--Territory-->
  <RelatedRoleID>G1a56c6f9-fdf4-490d-9e49-ed03fc380b4a</RelatedRoleID>
  <Cardinality>OptionalMany</Cardinality>
  <Relation Name="FK_SalesOrderHeader_SalesTerritory_TerritoryID"
            RelationEnd="Source" />
</Role>
```

3.1.8 Role 2: Related Role

This second Role example has an **ID** attribute with a QName value of G1a56c6f9-fdf4-490d-9e49-ed03fc380b4a, which associates a GUID with this **Role**. The **Name** of this **Role** is "Territory", which is the value of the Role.Name element. The singular and plural forms of the noun representing this **Role** are not specified in this example.

The Role.RelatedRoleID element specifies a reference to the corresponding role on the target entity. The corresponding role, also known as the related role, is a role that relates the same two entities as the original role, but relates them in the opposite order. The containing entity of the original role is the target entity of the related role and vice versa. The related role for this example is given in

section 3.1.7. The Role.Cardinality element specifies the cardinality of the role. In this case, "OptionalOne" specifies that the cardinality of the role is 0 or 1.

The Relation element specifies a binding of the **Role** to a relation in the underlying database. The relation is specified by its name. which is "FK_SalesOrderHeader_SalesTerritory_TerritoryID" in this case. The **Role** is bound to the target end of the relation, as specified by the value of the Relation.RelationEnd attribute, which is "Target". **Role** elements occur in pairs with one role bound to the source end of the relation and the other role bound to the target end of the same relation.

```
<Role ID="G1a56c6f9-fdf4-490d-9e49-ed03fc380b4a">
  <Name>Territory</Name>
  <!--Sales Orders-->
  <RelatedRoleID>G0a9cb3d9-f4b2-48a6-8c8d-6324f65a3f26</RelatedRoleID>
  <Cardinality>OptionalOne</Cardinality>
  <Relation Name="FK_SalesOrderHeader_SalesTerritory_TerritoryID"
            RelationEnd="Target" />
</Role>
```

3.1.9 Perspectives

The following is an example of a Perspectives element, which specifies a subset of the semantic model to provide to users as a submodel. This **Perspectives** collection contains a single Perspective element, which has an **ID** attribute with a QName value of G33871e88-dc21-4242-9e45-01d751069a52, which associates a GUID with this **Perspective**. The Perspective.Name and Perspective.Description elements provide a user-friendly name and description of the perspective.

The ModelItems collection specifies a list of references to Entity, Attribute, Role, EntityFolder, and FieldFolder elements in the semantic model. Each reference is specified as a **ModelItemID** element with a **QName** value, which is the value of the **ID** attribute of a model item. There is no specific ordering of the entries in the **ModelItems** collection.

```
  <Perspectives>
    <Perspective ID="G33871e88-dc21-4242-9e45-01d751069a52">
      <Name>Individual Customer Sales</Name>
      <Description>This perspective provides a small subset of Adventure Works focused only
on sales to individuals.</Description>
      <ModelItems>
```

The first **ModelItemID** element that follows is a reference to an **Entity**. The **QName** value of G73aed554-12c3-4750-8981-997fe69705ab is the same as the **ID** attribute value of the **Entity** whose **Name** is "Product".

The second **ModelItemID** element is a reference to a **Role**. The **QName** value of G75db5378-b40a-49a6-8fac-53b563f471d8 is the same as the **ID** attribute value of the **Role** whose **Name** is "Sales".

The third **ModelItemID** element is a reference to an **Entity**. The **QName** value of G7d048573-fb87-42c3-9660-f171d6645f3a is the same as the **ID** attribute value of the **Entity** whose **Name** is "Sale".

The fourth **ModelItemID** element is a reference to an **Attribute**. The **QName** value of G8adb941a-9e66-4373-9dfe-f3c44285e6d8 is the same as the **ID** attribute value of the **Attribute** whose **Name** is "#Sales".

The fifth **ModelItemID** element is a reference to a **Role**. The **QName** value of Gd807fa2d-8de1-4304-90d7-ea92ecf54d77 is the same as the **ID** attribute value of the **Role** whose **Name** is "Sales Order".

The sixth **ModelItemID** element is a reference to an **Attribute**. The **QName** value of G530377c5-2430-4f59-8c0e-e126375b0f79 is the same as the **ID** attribute value of the **Attribute** whose **Name** is "Sales ID".

```
          <!--Product-->
          <ModelItemID>G73aed554-12c3-4750-8981-997fe69705ab</ModelItemID>
          <!--Sales-->
          <ModelItemID>G75db5378-b40a-49a6-8fac-53b563f471d8</ModelItemID>
          <!--Sale-->
          <ModelItemID>G7d048573-fb87-42c3-9660-f171d6645f3a</ModelItemID>
          <!--#Sales -->
          <ModelItemID>G8adb941a-9e66-4373-9dfe-f3c44285e6d8</ModelItemID>
          <!--Sales Order-->
          <ModelItemID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</ModelItemID>
          <!--Sales ID-->
          <ModelItemID>G530377c5-2430-4f59-8c0e-e126375b0f79</ModelItemID>
        </ModelItems>
      </Perspective>
    </Perspectives>
```

3.1.10 Expression 1: Attribute Expression

The following is an example of an Expression element that returns an Attribute value. The **Attribute** is specified by a reference from the AttributeID child element of the AttributeRef element. The reference is the QName with a value of ge11619e1-5e37-4146-9389-8a7bdc9fe2f3, which is the value of an Attribute.ID attribute of an attribute in the semantic model. The **Attribute** with this **ID** happens to have the **Name** "Middle Name".

The Path element specifies a path from the context entity of this **Expression** to the **Entity** that contains the "Middle Name" **Attribute**. These entities cannot be seen in the example below. However, in the containing semantic model, these two **Entity** elements have the names "Customer" and "Individual" respectively. In this example, the **Path** collection has a single RolePathItem member, which is a reference to the Role with Role.ID, whose value is the **QName** with a value of g2198b1f8-8d57-43a0-a7d8-7b30a36fd6a1 specified by the RolePathItem.RoleID element. This **Role** specifies the relationship between the "Customer" and "Individual" entities, and the **Path** specifies the path consisting of just one step linking these two entities.

```
<Expression>
  <Path>
    <RolePathItem>
      <!--Contact-->
      <RoleID>G2198b1f8-8d57-43a0-a7d8-7b30a36fd6a1</RoleID>
    </RolePathItem>
  </Path>
  <AttributeRef>
    <!--Middle Name-->
    <AttributeID>Ge11619e1-5e37-4146-9389-8a7bdc9fe2f3</AttributeID>
  </AttributeRef>
</Expression>
```

3.1.11 Expression 2: Literal Expression

The following is an example of an Expression element that returns a Literal value. The Literal.DataType element specifies that the value of the expression is a Variant of type Integer. The value of the expression is specified by the value of the Literal.Value element, which is 6 in this example. Therefore, this is an example of an **Expression** that returns the **Variant** of type **Integer** with a value of 6.

```
<Expression>
```

```
    <Literal>
      <DataType>Integer</DataType>
      <Value>6</Value>
    </Literal>
  </Expression>
```

3.1.12 Expression 3: Nested Function Expression

The following is an example of an Expression element that calls a complex set of nested Function and **Expression** elements to determine its return value. This example was taken from a hierarchy filter expression, so the context entity of this **Expression** is the BaseEntity of a Hierarchy, which is the "Sale" Entity in this example, so the evaluation of the **Expression** takes place within the context of the "Sale" **Entity**.

The top-level **Expression** element is given the name "expr1" via the value of the **Name** attribute. The **Function** child element specifies that the And function is called as the top-level function as seen by the value of the Function.FunctionName element. The value of this **Expression** is a Boolean because the top-level function **And** has a **Boolean** return type.

The final result of this expression is a set of **Boolean** values. Each **Boolean** value is true if an instance of the "Product Category" **Entity** is in the set of EntityKey values {AAEAAAA=, AAMAAAA=, AAQAAAA=} and the "Order Date" Attribute value of the instance is greater than or equal to the value of the "Order Date" Parameter, and is false otherwise. The detailed explanation of the **Expression** element components follows.

```
<Expression Name="expr1">
  <Function>
    <FunctionName>And</FunctionName>
    <Arguments>
```

The first argument to the preceding **And** function is the expression containing the function In, which determines whether a value is contained in a given set.

```
      <Expression>
        <Function>
          <FunctionName>In</FunctionName>
          <Arguments>
```

In has as its first argument an **Expression** that has a reference to the "Product Category" **Entity**. This reference is given by the EntityRef and **EntityID** elements. The QName gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0 is the value of the **EntityID** attribute of an **Entity** in the semantic model.

The Path child element specifies a path via a series of references to Role elements from the context entity, the "Sale" **Entity** to the "Product Category" **Entity**.

The first element of the **Path** is a reference to the "Product" **Role** via the RolePathItem element and its RolePathItem.RoleID child element with **QName** value gbc0575d8-c020-4b4a-be45-bbf88013aca9, which identifies the "Product" **Role**. Although this is apparent only by viewing the entire semantic model, the "Product" **Role** specifies a relationship between the "Sale" **Entity** and the "Product" **Entity**.

The second element of the **Path** is a reference to the "Product Subcategory" **Role** via the **RolePathItem** element and its **RoleID** child element with **QName** value g59f93d17-7b9a-4700-a101-9322fdb58109, which identifies the "Product Subcategory **Role**. Although this is apparent only by viewing the entire semantic model, the "Product Subcategory" **Role** specifies a relationship between the "Product" **Entity** and the "Product Subcategory" **Entity**.

The third and last element of the **Path** is a reference to the "Product Subcategory" **Role** via the **RolePathItem** element and its **RoleID** child element with **QName** value g4d860b0a-2b57-4967-8fb9-133beee07067, which identifies the "Product Category" **Role**. Although this is apparent only by viewing the entire semantic model, the "Product Category" **Role** specifies a relationship between the "Product Subcategory" **Entity** and the "Product Category" **Entity**.

```
<Expression>
  <Path>
    <RolePathItem>
      <!--Product-->
      <RoleID>Gbc0575d8-c020-4b4a-be45-bbf88013aca9</RoleID>
    </RolePathItem>
    <RolePathItem>
      <!--Product Subcategory-->
      <RoleID>G59f93d17-7b9a-4700-a101-9322fdb58109</RoleID>
    </RolePathItem>
    <RolePathItem>
      <!--Product Category-->
      <RoleID>G4d860b0a-2b57-4967-8fb9-133beee07067</RoleID>
    </RolePathItem>
  </Path>
  <EntityRef>
    <!--Product Category-->
    <EntityID>Gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0</EntityID>
  </EntityRef>
</Expression>
```

In has as second argument an **Expression** that specifies the literal set of three **EntityKey** values {AAEAAAA=, AAMAAAA=, AAQAAAA=}. The data type of this literal is specified as **EntityKey** by the Literal.DataType element, and the value of this literal is specified as the set of values of the three Values.Value elements.

```
      <Expression>
        <Literal>
          <DataType>EntityKey</DataType>
          <Values>
            <Value>AAEAAAA=</Value>
            <Value>AAMAAAA=</Value>
            <Value>AAQAAAA=</Value>
          </Values>
        </Literal>
      </Expression>
    </Arguments>
  </Function>
  <CustomProperties>
    <CustomProperty Name="qd:FilterCondition" />
  </CustomProperties>
</Expression>
```

The second argument to the **And** function in the preceding XML snippet is the **Expression** that contains the function GreaterThanOrEquals.

```
<Expression>
  <Function>
    <FunctionName>GreaterThanOrEquals</FunctionName>
    <Arguments>
```

GreaterThanOrEquals has as first argument an expression that contains the scalar function Date that has as its argument an expression that contains a reference to the "Order Date" **Attribute**. This reference is given by the AttributeRef and **AttributeID** elements. The **QName** value g1f88a445-ae2b-

497e-8b0b-3f797fbc868c is the value of an Attribute.ID attribute of the "Order Date" **Attribute** in the semantic model.

As in the preceding XML snippet, the **Path** child element specifies a path, via a series of references to **Role** elements, from the context entity, the "Sale" **Entity** in this case, to the "Sales Order" **Entity**, which contains the "Order Date" **Attribute**.

```
                    <Expression>
                      <Function>
                        <FunctionName>Date</FunctionName>
                        <Arguments>
                          <Expression>
                            <Path>
                              <RolePathItem>
                                <!--Sales Order-->
                                <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
                              </RolePathItem>
                            </Path>
                            <AttributeRef>
                              <!--Order Date-->
                              <AttributeID>G1f88a445-ae2b-497e-8b0b-
3f797fbc868c</AttributeID>
                            </AttributeRef>
                          </Expression>
                        </Arguments>
                      </Function>
                    </Expression>
```

The second argument of the **GreaterThanOrEquals** function is an expression that contains a reference to the "Order Date" **Parameter** via the ParameterRef element with child element ParameterRef.ParameterName with value "Order Date".

Also of interest in the following code extract are the CustomProperties collections. These allow model design tools to store implementation-specific properties and persist them within the model. The CustomProperty element with **Name** "qd:ContextEntityID" has a CustomProperty.Value child element that illustrates a Variant value of type String with value g7d048573-fb87-42c3-9660-f171d6645f3. Note that the **String** data type of the **Variant** is specified by the "xsi:type" attribute, which has a value of "xsd:string".

```
                    <Expression>
                      <ParameterRef>
                        <ParameterName>Order Date</ParameterName>
                      </ParameterRef>
                    </Expression>
                  </Arguments>
                </Function>
                <CustomProperties>
                  <CustomProperty Name="qd:FilterCondition" />
                </CustomProperties>
              </Expression>
            </Arguments>
          </Function>
          <CustomProperties>
            <CustomProperty Name="qd:Filter" />
            <CustomProperty Name="qd:ContextEntityID">
              <Value xsi:type="xsd:string">G7d048573-fb87-42c3-9660-f171d6645f3a</Value>
            </CustomProperty>
            <CustomProperty Name="qd:AutoChangeBaseEntity" />
            <CustomProperty Name="qd:Design">
              <Value xsi:type="xsd:string">expr2</Value>
            </CustomProperty>
          </CustomProperties>
        </Expression>
```

3.1.13 Expression 4: Parameter Expression

The following is an example of an Expression element that returns the value of a Parameter. The ParameterRef element with child element ParameterRef.ParameterName specifies a reference to the **Parameter** whose **Name** is "Order Date" in this case. This **Parameter** is defined elsewhere in the semantic query.

```
<Expression>
  <ParameterRef>
    <ParameterName>Order Date</ParameterName>
  </ParameterRef>
</Expression>
```

3.1.14 Semantic Query

The following is an example that shows how to define a semantic query that can be used for querying a semantic model. Such a query would typically be used in a **Query.CommandText** element of a **DataSet** report definition language (RDL) file when creating a report against a semantic model. The example shows a Hierarchies definition, which includes Grouping and Filter elements; a MeasureGroups definition, which includes Measures; and definitions for CalculatedAttributes and a Parameter.

The following example of a SemanticQuery element is extracted from an RDL file where it is used to make a semantic query against a semantic model. The **SemanticQuery** element has several attributes, one of which specifies the definition of the SMDL Schema namespace to be http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling. The other attributes specify various namespace prefixes.

The first child element in the **SemanticQuery** is a **Hierarchies** collection that contains a single Hierarchy member. This **Hierarchy** element contains a BaseEntity, a Groupings collection, and a **Filter** element. The **BaseEntity** references the "Sale" Entity via the **EntityID** that has a QName value of g7d048573-fb87-42c3-9660-f171d6645f3a. This value is the value of an Entity.ID attribute for an **Entity** in the semantic model that returns "Sales" data. The **Groupings** collection has five **Grouping** members named "Product Category", "Product Subcategory", "Product", "Order Year", and "Order Quarter".

```
<SemanticQuery xmlns="http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:qd="http://schemas.microsoft.com/sqlserver/2004/11/semanticquerydesign"
xmlns:rb="http://schemas.microsoft.com/sqlserver/2004/11/reportbuilder">
  <Hierarchies>
    <Hierarchy>
      <BaseEntity>
        <!--Sale-->
        <EntityID>G7d048573-fb87-42c3-9660-f171d6645f3a</EntityID>
      </BaseEntity>
      <Groupings>
```

The following **Grouping** element specifies a grouping of the "Sale" **Entity** data by the "Product Category" **Entity**. The name of the **Grouping** is specified by the value of the **Name** attribute, which is "Product Category". The Expression element specifies the expression by which to group the data. The Path collection has three RolePathItem members that reference the "Product", "Product Subcategory", and "Product Category" Role elements via the **RoleID** elements that have values that match the value of Role.ID attributes for **Role** elements in the semantic model.

The EntityRef element references the "Product Category" **Entity** via the **EntityID** that has a **QName** value of gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0. This value is the value of an **Entity.ID** attribute for an **Entity** in the semantic model that returns "Product Category" data.

The Details collection specifies a single **Expression** to display for the grouping. This **Expression** returns the **Name** of instances of the "Product Category" **Entity** because the AttributeRef element references the "Name" Attribute via the **AttributeID** that has a **QName** value of g06eeff71-1bf2-461f-a84b-ff6268e3f684. This value is the value of an Attribute.ID attribute for an attribute in the semantic model that returns "Name of Product Category" data.

```
<Grouping Name="Product Category">
  <Expression Name="Product Category">
    <Path>
      <RolePathItem>
        <!--Product-->
        <RoleID>Gbc0575d8-c020-4b4a-be45-bbf88013aca9</RoleID>
      </RolePathItem>
      <RolePathItem>
        <!--Product Subcategory-->
        <RoleID>G59f93d17-7b9a-4700-a101-9322fdb58109</RoleID>
      </RolePathItem>
      <RolePathItem>
        <!--Product Category-->
        <RoleID>G4d860b0a-2b57-4967-8fb9-133beee07067</RoleID>
      </RolePathItem>
    </Path>
    <EntityRef>
      <!--Product Category-->
      <EntityID>Gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0</EntityID>
    </EntityRef>
  </Expression>
  <Details>
    <Expression Name="Product Category1">
      <AttributeRef>
        <!--Name-->
        <AttributeID>G06eeff71-1bf2-461f-a84b-ff6268e3f684</AttributeID>
      </AttributeRef>
    </Expression>
  </Details>
</Grouping>
```

The following **Grouping** element specifies a grouping of the "Sale" **Entity** data by the "Product Subcategory" **Entity**. The explanation of this **Grouping** is similar to the preceding **Grouping** element.

```
<Grouping Name="Product Subcategory">
  <Expression Name="Product Subcategory">
    <Path>
      <RolePathItem>
        <!--Product-->
        <RoleID>Gbc0575d8-c020-4b4a-be45-bbf88013aca9</RoleID>
      </RolePathItem>
      <RolePathItem>
        <!--Product Subcategory-->
        <RoleID>G59f93d17-7b9a-4700-a101-9322fdb58109</RoleID>
      </RolePathItem>
    </Path>
    <EntityRef>
      <!--Product Subcategory-->
      <EntityID>G8f91f894-d044-4da9-a138-772900aee688</EntityID>
    </EntityRef>
  </Expression>
  <Details>
    <Expression Name="Product Subcategory2">
      <AttributeRef>
        <!--Name-->
        <AttributeID>G00b169b7-9f08-4f96-babd-5924781607b5</AttributeID>
      </AttributeRef>
    </Expression>
  </Details>
```

```
</Grouping>
```

The following **Grouping** element specifies a grouping of the "Sale" **Entity** data by the "Product" **Entity**. The explanation of this **Grouping** is similar to the preceding **Grouping** elements.

```
<Grouping Name="Product">
  <Expression Name="Product">
    <Path>
      <RolePathItem>
        <!--Product-->
        <RoleID>Gbc0575d8-c020-4b4a-be45-bbf88013aca9</RoleID>
      </RolePathItem>
    </Path>
    <EntityRef>
      <!--Product-->
      <EntityID>G73aed554-12c3-4750-8981-997fe69705ab</EntityID>
    </EntityRef>
  </Expression>
  <Details>
    <Expression Name="Product3">
      <AttributeRef>
        <!--Name-->
        <AttributeID>G64343d16-f565-44c8-884c-681a8716c07e</AttributeID>
      </AttributeRef>
    </Expression>
  </Details>
</Grouping>
```

The following **Grouping** element specifies a grouping of the "Sale" **Entity** data by the "Order Year" **Entity**. The name of the **Grouping** is specified by the value of the **Name** attribute, which is "Order Year". The **Expression** element specifies the expression by which to group the data. The **Path** collection has a single **RolePathItem** member, which references the "Sales Order" **Role** element via the **RoleID** element that has a value that matches the value of the **Role.ID** attribute for the "Sales Order" **Role** in the semantic model.

The **AttributeRef** element references the "Product Category" **Attribute** via the **AttributeID** that has a **QName** value gb4a8fe12-4ec9-4fa9-8b33-82a9acfea68c. This value is the value of an **Attribute.ID** attribute for an **Attribute** in the semantic model that returns "Product Category" data.

```
<Grouping Name="Order Year">
  <Expression Name="Order Year">
    <Path>
      <RolePathItem>
        <!--Sales Order-->
        <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
      </RolePathItem>
    </Path>
    <AttributeRef>
      <!--Order Year-->
      <AttributeID>Gb4a8fe12-4ec9-4fa9-8b33-82a9acfea68c</AttributeID>
    </AttributeRef>
  </Expression>
</Grouping>
```

The following **Grouping** element specifies a grouping of the "Sale" **Entity** data by the "Order Quarter" **Entity**. The explanation of this **Grouping** is similar to the preceding one.

```
<Grouping Name="Order Quarter">
  <Expression Name="Order Quarter">
```

```
        <Path>
          <RolePathItem>
            <!--Sales Order-->
            <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
          </RolePathItem>
        </Path>
        <AttributeRef>
          <!--Order Quarter-->
          <AttributeID>Gd7e9b121-ae15-44be-9fec-0a0265c133f4</AttributeID>
        </AttributeRef>
      </Expression>
    </Grouping>
  </Groupings>
```

The **Filter** element that follows has a single filter **Expression** that contains several nested functions. The value of this **Expression** is a Boolean because the top-level function And has a **Boolean** return type.

The result of this filter is to filter the detail data by selecting only those instances where the "Product Category" is in the set {AAEAAAA=, AAMAAAA=, AAQAAAA=} and the "Order Date" is greater than or equal to the value of the "Order Date" parameter. A detailed explanation of the **Filter** element components follows.

```
<Filter>
  <Expression Name="expr1">
    <Function>
      <FunctionName>And</FunctionName>
      <Arguments>
```

The first argument to the **And** function in the preceding XML snippet is the expression that contains the function In, which determines whether a value is contained in a given set.

```
        <Expression>
          <Function>
            <FunctionName>In</FunctionName>
            <Arguments>
```

In has as its first argument an **Expression** that has a reference to the "Product Category" **Entity**. This reference is given by the **EntityRef** and **EntityID** elements. The **QName** value gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0 is the value of the **Entity.ID** attribute of an **Entity** in the semantic model.

```
              <Expression>
                <Path>
                  <RolePathItem>
                    <!--Product-->
                    <RoleID>Gbc0575d8-c020-4b4a-be45-bbf88013aca9</RoleID>
                  </RolePathItem>
                  <RolePathItem>
                    <!--Product Subcategory-->
                    <RoleID>G59f93d17-7b9a-4700-a101-9322fdb58109</RoleID>
                  </RolePathItem>
                  <RolePathItem>
                    <!--Product Category-->
                    <RoleID>G4d860b0a-2b57-4967-8fb9-133beee07067</RoleID>
                  </RolePathItem>
                </Path>
                <EntityRef>
                  <!--Product Category-->
                  <EntityID>Gacb1e610-3ea3-4922-a2d8-e503e1d0a3d0</EntityID>
                </EntityRef>
```

```
                    </Expression>
```

In has as its second argument an **Expression** that specifies the literal set of three EntityKey values {AAEAAAA=, AAMAAAA=, AAQAAAA=}. The data type of this Literal element is specified as **EntityKey** by the Literal.DataType element, and the value of this literal is specified as the set of values of the three Values.Value elements.

```
                    <Expression>
                      <Literal>
                        <DataType>EntityKey</DataType>
                        <Values>
                          <Value>AAEAAAA=</Value>
                          <Value>AAMAAAA=</Value>
                          <Value>AAQAAAA=</Value>
                        </Values>
                      </Literal>
                    </Expression>
                  </Arguments>
                </Function>
                <CustomProperties>
                  <CustomProperty Name="qd:FilterCondition" />
                </CustomProperties>
              </Expression>
```

The second argument to the **And** function in the preceding XML snippet is the **Expression** that contains the function GreaterThanOrEquals.

```
              <Expression>
                <Function>
                  <FunctionName>GreaterThanOrEquals</FunctionName>
                  <Arguments>
```

GreaterThanOrEquals has as its first argument an expression that contains the scalar function Date that has as its argument an expression that contains a reference to the "Order Date" **Attribute**. This reference is given by the **AttributeRef** and **AttributeID** elements. The **QName** value g1f88a445-ae2b-497e-8b0b-3f797fbc868c is the value of an **Attribute.ID** attribute of the "Order Date" **Attribute** in the semantic model.

```
                    <Expression>
                      <Function>
                        <FunctionName>Date</FunctionName>
                        <Arguments>
                          <Expression>
                            <Path>
                              <RolePathItem>
                                <!--Sales Order-->
                                <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
                              </RolePathItem>
                            </Path>
                            <AttributeRef>
                              <!--Order Date-->
                              <AttributeID>G1f88a445-ae2b-497e-8b0b-
3f797fbc868c</AttributeID>
                            </AttributeRef>
                          </Expression>
                        </Arguments>
                      </Function>
                    </Expression>
```

The second argument of the **GreaterThanOrEquals** function is an expression that contains a reference to the "Order Date" **Parameter** via the ParameterRef element with child element ParameterRef.ParameterName with a value of "Order Date".

```
                    <Expression>
                      <ParameterRef>
                        <ParameterName>Order Date</ParameterName>
                      </ParameterRef>
                    </Expression>
                  </Arguments>
                </Function>
                <CustomProperties>
                  <CustomProperty Name="qd:FilterCondition" />
                </CustomProperties>
              </Expression>
            </Arguments>
          </Function>
          <CustomProperties>
            <CustomProperty Name="qd:Filter" />
            <CustomProperty Name="qd:ContextEntityID">
              <Value xsi:type="xsd:string">G7d048573-fb87-42c3-9660-f171d6645f3a</Value>
            </CustomProperty>
            <CustomProperty Name="qd:AutoChangeBaseEntity" />
            <CustomProperty Name="qd:Design">
              <Value xsi:type="xsd:string">expr2</Value>
            </CustomProperty>
          </CustomProperties>
        </Expression>
      </Filter>
    </Hierarchy>
  </Hierarchies>
```

The **MeasureGroups** collection contains a single MeasureGroup element with a **BaseEntity** child element that references the "Sale" **Entity** by the BaseEntity.EntityID element whose value is the **QName** with value g7d048573-fb87-42c3-9660-f171d6645f3a.

In the **MeasureGroup**, the **Measures** collection contains a single **Expression** that contains the aggregate function Sum with an **Expression** as its single argument that returns the values of the "Total Due" **Attribute** as specified by the AttributeRef.AttributeID element with the **QName** value g39091120-95cf-4639-8b56-be464861440c. Also of interest is that the Measures.Expression element has a CustomProperties.CustomProperty element that specifies implementation-specific properties for model-design tools.

The **Path** collection in the **Expression** that is the argument of the **Sum** function specifies the path by which to reach the entity whose context is used to evaluate the expression. The **Path** collection has a single member, a **RolePathItem** that contains a reference to a **Role** via the **RoleID** element whose value is the **QName** value gd807fa2d-8de1-4304-90d7-ea92ecf54d77. This is the GUID of a **Role** in the semantic model whose name is "Sales Order".

```
<MeasureGroups>
  <MeasureGroup>
    <BaseEntity>
      <!--Sale-->
      <EntityID>G7d048573-fb87-42c3-9660-f171d6645f3a</EntityID>
    </BaseEntity>
    <Measures>
      <Expression Name="Sum Total Due_Sum">
        <Function>
          <FunctionName>Sum</FunctionName>
          <Arguments>
            <Expression>
              <Path>
                <RolePathItem>
                  <!--Sales Order-->
```

```
                <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
              </RolePathItem>
            </Path>
            <AttributeRef>
              <!--Total Due-->
              <AttributeID>G39091120-95cf-4639-8b56-be464861440c</AttributeID>
            </AttributeRef>
          </Expression>
        </Arguments>
      </Function>
      <CustomProperties>
        <CustomProperty Name="qd:Design">
          <Value xsi:type="xsd:string">Sum Total Due</Value>
        </CustomProperty>
      </CustomProperties>
    </Expression>
  </Measures>
 </MeasureGroup>
</MeasureGroups>
```

The **CalculatedAttributes** collection contains a single **Expression** whose name is "Sum Total Due", as specified by the **Name** attribute. The **Expression** evaluates the function Aggregate, which has one argument in the Arguments collection.

The argument is an **Expression** that has a **Path** child element and an **AttributeRef** element. The **Expression** returns the "Sum Total Due" **Attribute** because the **AttributeRef** element references the "Sum Total Due" **Attribute** via the **AttributeID** that has a **QName** value of g61256a47-8571-4aae-b9f5-0389812e8a3e. This value is the value of an **Attribute.ID** attribute for an **Attribute** in the semantic model that returns "Sum Total Due" data.

The **Path** collection specifies the path by which to reach an entity whose context is used to evaluate the expression. The **Path** collection has a single member, a **RolePathItem** that contains a reference to a **Role** via the **RoleID** element whose value is the **QName** value gd807fa2d-8de1-4304-90d7-ea92ecf54d77. This value is the value of the **Role.ID** attribute for a **Role** in the semantic model whose name is "Sales Order".

```
<CalculatedAttributes>
  <Expression Name="Sum Total Due">
    <Function>
      <FunctionName>Aggregate</FunctionName>
      <Arguments>
        <Expression>
          <Path>
            <RolePathItem>
              <!--Sales Order-->
              <RoleID>Gd807fa2d-8de1-4304-90d7-ea92ecf54d77</RoleID>
            </RolePathItem>
          </Path>
          <AttributeRef>
            <!--Sum Total Due-->
            <AttributeID>G61256a47-8571-4aae-b9f5-0389812e8a3e</AttributeID>
          </AttributeRef>
        </Expression>
      </Arguments>
    </Function>
    <CustomProperties>
      <CustomProperty Name="qd:ContextEntityID">
        <Value xsi:type="xsd:string">G7d048573-fb87-42c3-9660-f171d6645f3a</Value>
      </CustomProperty>
    </CustomProperties>
  </Expression>
</CalculatedAttributes>
```

The **SemanticQuery** has a single **Parameter** in its Parameters collection. This **Parameter** has a name of "Order Date" specified by its **Name** attribute. The data type of the parameter is specified as "DateTime" by the Parameter.DataType element, and the parameter cannot have a NULL value, because the Parameter.Nullable element is not specified. Finally, the **Expression** specifies the default value of the **Parameter** element as being the DateTime value 2003-05-01T00:00:00. This is specified by the values of the **Literal.DataType** and Literal.Value elements for the **Literal** element that is the content of the **Expression** being evaluated.

```
  <Parameters>
    <Parameter Name="Order Date">
      <DataType>DateTime</DataType>
      <Expression>
        <Literal>
          <DataType>DateTime</DataType>
          <Value>2003-05-01T00:00:00</Value>
        </Literal>
      </Expression>
    </Parameter>
  </Parameters>
</SemanticQuery>
```

3.1.15 DrillthroughContext

The following is an example that shows how to define a DrillthroughContext element that specifies additional filter information for the semantic query specified by the **DrillthroughSourceQuery** parameter. The **DrillthroughContext** element example that follows references items in the SemanticQuery element from section 3.1.14. The **DrillthroughContext** element has an **xmlns** attribute to specify using the SMDL Schema namespace, which is **http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling**.

The first child element of the **DrillthroughContext** element is a SelectedItems collection that contains a single SelectedItem.SelectedItemName member. This **SelectedItem.SelectedItemName** element specifies a reference by name to an Expression in the semantic query whose value determines what data is drilled into and displayed to the user. In this case, the example refers to the **Expression** whose name is "Sum Total Due_Sum". This is the **Expression** in the **SemanticQuery** from section 3.1.14 that defines the Measure in the MeasureGroup.

The second child element of the **DrillthroughContext** is a GroupingValues collection that contains three **GroupingValue** members. These **GroupingValue** elements specify that the Grouping whose **Name** is "Product Category" selects the group instance with a value of "AAQAAAA=", the **Grouping** whose **Name** is "Product Subcategory" selects the group instance with a value of "AB.oAAAA=", and the **Grouping** whose **Name** is "Order Year" selects the group instance with a value of "2003". These three **Grouping** elements are found in the Hierarchies element in the **SemanticQuery** from section 3.1.14.

```
<DrillthroughContext xmlns="http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling">
  <SelectedItems>
    <SelectedItemName>Sum Total Due_Sum</SelectedItemName>
  </SelectedItems>
  <GroupingValues>
    <GroupingValue Name="Product Category">AAQAAAA=</GroupingValue>
    <GroupingValue Name="Product Subcategory">AB.oAAAA=</GroupingValue>
    <GroupingValue Name="Order Year">2003</GroupingValue>
  </GroupingValues>
</DrillthroughContext>
```

4 Security Considerations

Because Semantic Model Definition Language contains only metadata about the semantic model, there are no security considerations for the SMDL file itself.

5 Appendix A: Full XML Schema

The following is the full SMDL XML Schema for Version 2004/10.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling"elementFormD
efault="qualified"
            xmlns="http://schemas.microsoft.com/sqlserver/2004/10/semanticmodeling"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:dsv="http://schemas.microsoft.com/analysisservices/2003/engine">
    <xsd:annotation>
        <xsd:documentation>
            The following schema for Microsoft SQL Server 2005 describes the Semantic Model
Definition Language (SMDL) for defining semantic models.

Microsoft does not make any representation or warranty regarding the schema or any product or
item developed based on the schema. The schema is provided to you on an AS IS basis.
Microsoft disclaims all express, implied and statutory warranties, including but not limited
to the implied warranties of merchantability, fitness for a particular purpose, and freedom
from infringement. Without limiting the generality of the foregoing, Microsoft does not make
any warranty of any kind that any item developed based on the schema, or any portion of the
schema, will not infringe any copyright, patent, trade secret, or other intellectual property
right of any person or entity in any country. It is your responsibility to seek licenses for
such intellectual property rights where appropriate.

MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION
WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL,
CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT
MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.

            (c) Microsoft Corporation. All rights reserved.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:import namespace="http://schemas.microsoft.com/analysisservices/2003/engine"
schemaLocation="DataSourceView.xsd" />
    <xsd:element name="SemanticModel">
        <xsd:complexType>
            <xsd:all>
                <xsd:element name="Description" type="xsd:string" minOccurs="0" />
                <xsd:element name="Version" type="xsd:string" minOccurs="0" />
                <xsd:element name="Culture" type="xsd:language" minOccurs="0" />
                <xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
                <xsd:element name="Perspectives" type="PerspectivesType" minOccurs="0" />
                <xsd:element name="CustomProperties" type="CustomPropertiesType"
                             minOccurs="0" />
                <xsd:element ref="dsv:DataSourceView" minOccurs="0" />
            </xsd:all>
            <xsd:attribute name="ID" type="xsd:QName" use="required" />
        </xsd:complexType>
        <xsd:unique name="ID">
            <xsd:selector xpath=".|.//*" />
            <xsd:field xpath="@ID" />
        </xsd:unique>
    </xsd:element>
    <xsd:simpleType name="NonEmptyString">
        <xsd:restriction base="xsd:string">
            <xsd:minLength value="1" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CustomPropertiesType">
        <xsd:sequence>
            <xsd:element name="CustomProperty" type="CustomPropertyType"
                         maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
```

```
<xsd:complexType name="CustomPropertyType">
    <xsd:all>
        <xsd:element name="Value" type="xsd:anySimpleType" minOccurs="0"/>
    </xsd:all>
    <xsd:attribute name="Name" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:complexType name="EntitiesType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Entity" type="EntityType" />
        <xsd:element name="EntityFolder" type="EntityFolderType" />
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="EntityFolderType">
    <xsd:all>
        <xsd:element name="Name" type="NonEmptyString" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="Entities" type="EntitiesType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:complexType name="EntityType">
    <xsd:all>
        <xsd:element name="Name" type="NonEmptyString" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="CollectionName" type="NonEmptyString" minOccurs="0" />
        <xsd:element name="IdentifyingAttributes" type="AttributeReferencesType" />
        <xsd:element name="DefaultDetailAttributes" type="AttributeReferencesType"
                     minOccurs="0" />
        <xsd:element name="DefaultAggregateAttributes" type="AttributeReferencesType"
                     minOccurs="0" />
        <xsd:element name="SortAttributes" type="SortAttributesType" minOccurs="0" />
        <xsd:element name="InstanceSelection">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="FilteredList" />
                    <xsd:enumeration value="Dropdown" />
                    <xsd:enumeration value="List" />
                    <xsd:enumeration value="MandatoryFilter" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="IsLookup" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Inheritance" minOccurs="0">
            <xsd:complexType>
                <xsd:all>
                    <xsd:element name="InheritsFromEntityID" type="xsd:QName" />
                    <xsd:element name="Relation" type="RelationType" minOccurs="0" />
                </xsd:all>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="DisjointInheritance" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
        <xsd:element name="SecurityFilters" type="AttributeReferencesType"
                     minOccurs="0" />
        <xsd:element name="DefaultSecurityFilter" type="DefaultSecurityFilterType"
                     minOccurs="0" />
        <xsd:element name="Table" type="TableType" minOccurs="0" />
        <xsd:element name="Column" type="ColumnType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:complexType name="AttributeReferencesType">
    <xsd:sequence>
        <xsd:element name="AttributeReference" type="AttributeReferenceType"
                     maxOccurs="unbounded" />
    </xsd:sequence>
```

```
</xsd:complexType>
<xsd:complexType name="AttributeReferenceType">
    <xsd:all>
        <xsd:element name="Path" type="PathType" minOccurs="0" />
        <xsd:element name="AttributeID" type="xsd:QName" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="SortAttributesType">
    <xsd:sequence>
        <xsd:element name="SortAttribute" type="SortAttributeType"
                     maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SortAttributeType">
    <xsd:all>
        <xsd:element name="AttributeReference" type="AttributeReferenceType" />
        <xsd:element name="SortDirection" type="SortAttributeDirectionEnum"
                     minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:simpleType name="SortAttributeDirectionEnum">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Ascending" />
        <xsd:enumeration value="Descending" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="DefaultSecurityFilterType">
    <xsd:all>
        <xsd:element name="AttributeReference" type="AttributeReferenceType" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="FieldsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Attribute" type="AttributeType" />
        <xsd:element name="Role" type="RoleType" />
        <xsd:element name="FieldFolder" type="FieldFolderType" />
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="FieldFolderType">
    <xsd:all>
        <xsd:element name="Name" type="NonEmptyString" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:complexType name="VariationsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Attribute" type="AttributeType" />
        <xsd:element name="Role" type="RoleType" />
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="AttributeType">
    <xsd:all>
        <xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="Variations" type="VariationsType" minOccurs="0" />
        <xsd:element name="DataType" type="DataTypeEnum" />
        <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
        <xsd:element name="IsAggregate" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="IsFilter" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="OmitSecurityFilters" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="SortDirection" type="SortDirectionEnum" minOccurs="0" />
        <xsd:element name="Width" minOccurs="0">
            <xsd:simpleType>
```

```
                <xsd:restriction base="xsd:int">
                    <xsd:minInclusive value="0" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Alignment" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="General" />
                    <xsd:enumeration value="Left" />
                    <xsd:enumeration value="Center" />
                    <xsd:enumeration value="Right" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Format" type="xsd:string" minOccurs="0" />
        <xsd:element name="MimeType" type="NonEmptyString" minOccurs="0" />
        <xsd:element name="DataCulture" type="xsd:language" minOccurs="0" />
        <xsd:element name="DiscourageGrouping" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="ContextualName" type="AttributeContextualNameEnum"
                     minOccurs="0"/>
        <xsd:element name="DefaultAggregateAttributeID" type="xsd:QName" minOccurs="0" />
        <xsd:element name="ValueSelection" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Dropdown" />
                    <xsd:enumeration value="List" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Column" type="ColumnType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:simpleType name="DataTypeEnum">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="String" />
        <xsd:enumeration value="Integer" />
        <xsd:enumeration value="Decimal" />
        <xsd:enumeration value="Float" />
        <xsd:enumeration value="Boolean" />
        <xsd:enumeration value="DateTime" />
        <xsd:enumeration value="Time" />
        <xsd:enumeration value="Binary" />
        <xsd:enumeration value="EntityKey" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="SortDirectionEnum">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="None" />
        <xsd:enumeration value="Ascending" />
        <xsd:enumeration value="Descending" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AttributeContextualNameEnum">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Attribute" />
        <xsd:enumeration value="Role" />
        <xsd:enumeration value="Merge" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="RoleType">
    <xsd:all>
        <xsd:element name="Name" type="NonEmptyString" minOccurs="0" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="Variations" type="VariationsType" minOccurs="0" />
```

```
            <xsd:element name="Linguistics" type="LinguisticsType" minOccurs="0" />
            <xsd:element name="RelatedRoleID" type="xsd:QName" />
            <xsd:element name="Cardinality">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="One" />
                        <xsd:enumeration value="Many" />
                        <xsd:enumeration value="OptionalOne" />
                        <xsd:enumeration value="OptionalMany" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="ContextualName" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="Default" />
                        <xsd:enumeration value="Role" />
                        <xsd:enumeration value="Merge" />
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="HiddenFields" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="FieldFolderItemID" type="xsd:QName"
                                     maxOccurs="unbounded" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="Preferred" type="xsd:boolean" minOccurs="0" />
            <xsd:element name="PromoteLookup" type="xsd:boolean" minOccurs="0" />
            <xsd:element name="ExpandInline" type="xsd:boolean" minOccurs="0" />
            <xsd:element name="Relation" type="RelationType" minOccurs="0" />
        </xsd:all>
        <xsd:attribute name="ID" type="xsd:QName" use="required" />
    </xsd:complexType>
    <xsd:complexType name="LinguisticsType">
        <xsd:all>
            <xsd:element name="SingularName" type="NonEmptyString" minOccurs="0" />
            <xsd:element name="PluralName" type="NonEmptyString" minOccurs="0" />
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="TableType">
        <xsd:attribute name="Name" type="NonEmptyString" use="required" />
    </xsd:complexType>
    <xsd:complexType name="ColumnType">
        <xsd:attribute name="TableName" type="NonEmptyString" use="optional" />
        <xsd:attribute name="Name" type="NonEmptyString" use="required" />
    </xsd:complexType>
    <xsd:complexType name="RelationType">
        <xsd:attribute name="Name" type="NonEmptyString" use="required" />
        <xsd:attribute name="RelationEnd" use="optional">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Source" />
                    <xsd:enumeration value="Target" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="PerspectivesType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="Perspective" type="PerspectiveType" />
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="PerspectiveType">
        <xsd:all>
            <xsd:element name="Name" type="NonEmptyString" />
            <xsd:element name="Description" type="xsd:string" minOccurs="0" />
            <xsd:element name="ModelItems" type="ModelItemsType" />
```

```
            <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        </xsd:all>
        <xsd:attribute name="ID" type="xsd:QName" use="required" />
    </xsd:complexType>
    <xsd:complexType name="ModelItemsType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="ModelItemID" type="xsd:QName" />
        </xsd:choice>
    </xsd:complexType>
    <!-- Semantic Model Query Language (SMQL) elements -->
    <xsd:element name="SemanticQuery">
        <xsd:complexType>
            <xsd:all>
                <xsd:element name="Hierarchies" type="HierarchiesType" minOccurs="0" />
                <xsd:element name="MeasureGroups" type="MeasureGroupsType" minOccurs="0" />
                <xsd:element name="CalculatedAttributes" type="ExpressionsType"
                             minOccurs="0" />
                <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
                <xsd:element name="EnableDrillthrough" type="xsd:boolean" minOccurs="0" />
                <xsd:element name="CustomProperties" type="CustomPropertiesType"
                             minOccurs="0" />
            </xsd:all>
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="HierarchiesType">
        <xsd:sequence>
            <xsd:element name="Hierarchy" type="HierarchyType" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="HierarchyType">
        <xsd:all>
            <xsd:element name="BaseEntity" type="BaseEntityType" />
            <xsd:element name="Groupings" type="GroupingsType" minOccurs="0" />
            <xsd:element name="Filter" minOccurs="0">
                <xsd:complexType>
                    <xsd:all>
                        <xsd:element name="Expression" type="ExpressionType" />
                    </xsd:all>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="BaseEntityType">
        <xsd:all>
            <xsd:element name="EntityID" type="xsd:QName" />
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="GroupingsType">
        <xsd:sequence>
            <xsd:element name="Grouping" type="GroupingType" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="GroupingType">
        <xsd:all>
            <xsd:element name="Expression" type="ExpressionType" />
            <xsd:element name="Details" type="ExpressionsType" minOccurs="0" />
        </xsd:all>
        <xsd:attribute name="Name" type="NonEmptyString" use="required" />
    </xsd:complexType>
    <xsd:complexType name="MeasureGroupsType">
        <xsd:sequence>
            <xsd:element name="MeasureGroup" type="MeasureGroupType" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="MeasureGroupType">
        <xsd:all>
            <xsd:element name="BaseEntity" type="BaseEntityType" />
            <xsd:element name="Measures" type="ExpressionsType" />
            <xsd:element name="SubtotalSets" type="SubtotalSetsType" minOccurs="0" />
        </xsd:all>
```

```
</xsd:complexType>
<xsd:complexType name="SubtotalSetsType">
    <xsd:sequence>
        <xsd:element name="SubtotalSet" type="SubtotalSetType" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SubtotalSetType">
    <xsd:all>
        <xsd:element name="SubtotalGroupings" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="GroupingName" type="xsd:string"
                                 maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="SubtotalMeasures" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="MeasureName" type="xsd:string"
                                 maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="ParametersType">
    <xsd:sequence>
        <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ParameterType">
    <xsd:all>
        <xsd:element name="DataType" type="LiteralDataTypeEnum" />
        <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Cardinality" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="One" />
                    <xsd:enumeration value="Many" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Expression" type="ExpressionType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="Name" type="NonEmptyString" use="required" />
</xsd:complexType>
<xsd:complexType name="ExpressionType">
    <xsd:all>
        <xsd:element name="Path" type="PathType" minOccurs="0" />
        <xsd:element name="Function" type="FunctionType" minOccurs="0" />
        <xsd:element name="AttributeRef" type="AttributeRefType" minOccurs="0" />
        <xsd:element name="EntityRef" type="EntityRefType" minOccurs="0" />
        <xsd:element name="ParameterRef" type="ParameterRefType" minOccurs="0" />
        <xsd:element name="Literal" type="LiteralType" minOccurs="0" />
        <xsd:element name="Null" type="NullType" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="Name" type="NonEmptyString" use="optional" />
</xsd:complexType>
<xsd:complexType name="ExpressionsType">
    <xsd:sequence>
        <xsd:element name="Expression" type="ExpressionType" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PathType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="RolePathItem" type="RolePathItemType" />
    </xsd:choice>
</xsd:complexType>
```

```
<xsd:complexType name="RolePathItemType">
    <xsd:all>
        <xsd:element name="RoleID" type="xsd:QName" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="FunctionType">
    <xsd:all>
        <xsd:element name="FunctionName" type="xsd:string" />
        <xsd:element name="Arguments" type="ExpressionsType" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="AttributeRefType">
    <xsd:all>
        <xsd:element name="AttributeID" type="xsd:QName" minOccurs="0" />
        <xsd:element name="AttributeName" type="xsd:string" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="EntityRefType">
    <xsd:all>
        <xsd:element name="EntityID" type="xsd:QName" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="ParameterRefType">
    <xsd:all>
        <xsd:element name="ParameterName" type="xsd:string" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="LiteralType">
    <xsd:all>
        <xsd:element name="DataType" type="LiteralDataTypeEnum" />
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Values" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:all>
</xsd:complexType>
<xsd:simpleType name="LiteralDataTypeEnum">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="String" />
        <xsd:enumeration value="Integer" />
        <xsd:enumeration value="Decimal" />
        <xsd:enumeration value="Float" />
        <xsd:enumeration value="Boolean" />
        <xsd:enumeration value="DateTime" />
        <xsd:enumeration value="Time" />
        <xsd:enumeration value="EntityKey" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="NullType" />
<xsd:element name="DrillthroughContext">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="SelectedItems" type="SelectedItemsType" />
            <xsd:element name="SelectedPath" type="SelectedPathType" minOccurs="0" />
            <xsd:element name="GroupingValues" type="GroupingValuesType" minOccurs="0">
                <xsd:unique name="GroupingNames">
                    <xsd:selector xpath=".//*" />
                    <xsd:field xpath="@Name" />
                </xsd:unique>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>
<xsd:complexType name="SelectedItemsType">
    <xsd:sequence>
```

```
            <xsd:element name="SelectedItemName" type="NonEmptyString"
                         maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="SelectedPathType">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="RolePathItem" type="RolePathItemType" />
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="GroupingValuesType">
        <xsd:sequence>
            <xsd:element name="GroupingValue" nillable="true" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="Name" type="NonEmptyString"
                                           use="required" />
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

6 (Updated Section) Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include ~~released service packs~~updates to those products.

- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016
- Microsoft SQL Server 2017
- Microsoft SQL Server 2019

Exceptions, if any, are noted ~~below.~~in this section. If ~~a~~ an update version, service pack or ~~Quick Fix Engineering (QFE~~Knowledge Base (KB) number appears with ~~the~~a product ~~version,~~name, the behavior changed in that ~~service pack or QFE.~~update. The new behavior also applies to subsequent ~~service packs of the product~~updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2.12: The Microsoft SMDL reader does not validate the value of **MIMEType**.

<2> Section 2.22.1: The Microsoft SMDL reader does not enforce this requirement of the Relation.RelationEnd attribute. Non-uniqueness of the key generates a warning but not an error.

<3> Section 2.22.2: The Microsoft SMDL reader does not enforce this requirement of the Relation.RelationEnd attribute. Non-uniqueness of the key generates a warning but not an error.

<4> Section 2.24.2: The Microsoft SMDL reader does not enforce the restriction that the key of the table or column be unique. Non-uniqueness of the key generates a warning but not an error. However, the database relation still needs to relate the data or column to which the grandparent Entity is bound, to the key of the table or column to which the **Entity** specified by the value of the Inheritance.InheritsFromEntityID child element of the grandparent **Entity** element is bound.

<5> Section 2.65.48: In SQL Server Reporting Services (SSRS), the Report Server requires the SemanticModel.DataSourceView element to be present, and the Model Designer SSRS client tool does not require this element to be present in the SMDL.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
6 Appendix B: Product Behavior	Added SQL Server 2019 to the product applicability list.	Major

8 Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X