[MS-SSAS-T]:

SQL Server Analysis Services Tabular

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.
- **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.

Revision Summary

Date	Revision History	Revision Class	Comments
5/10/2016	1.0	New	Initial Availability
7/14/2016	2.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The SQL Server Analysis Services Tabular protocol provides the methods for a client to communicate with and perform operations on an **analysis server** that is using Tabular databases that are at compatibility level 1200 or higher. This protocol is an extension of the SQL Server Analysis Services protocol [MS-SSAS].

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

analysis server: A server that supports high performance and complex analytics for business intelligence applications.

attribute hierarchy: An implied single-level **hierarchy**, based on a single attribute, that consists of all the members of the attribute. An all-level member can optionally be enabled for an attribute hierarchy.

Data Analysis Expressions (DAX): A library of functions and operators that can be combined to build formulas and expressions in a data model.

data definition language (DDL): A subset of SQL or XMLA statements that defines all the attributes and properties of a database and its objects. DDL statements typically begin with CREATE, ALTER, or DROP.

hierarchy: A logical tree structure that organizes a record such that each member has one parent member and zero or more child members.

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC4627]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

key performance indicator (KPI): A predefined measure that is used to track performance against a strategic goal, objective, plan, initiative, or business process. A visual cue is frequently used to communicate performance against the measure.

level: A relative position in a hierarchy of data. A level is frequently used when describing how to navigate a hierarchy in an Online Analytical Processing (OLAP) database or a PivotTable report.

Multidimensional Expressions (MDX): A syntax that is used for defining multidimensional objects, and for querying and manipulating multidimensional data.

volatile: A condition of a formula in which the formula is calculated every time the workbook is calculated. This is unlike a non-volatile formula, which is calculated only when dependent values are changed.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring **XML schemas**.

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.

[JSON-SchemaVal] Internet Engineering Task Force (IETF), "JSON Schema: interactive and non interactive validation", January 2013, http://json-schema.org/latest/json-schema-validation.html

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

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006, http://www.rfc-editor.org/rfc/rfc4627.txt

[RFC793] Postel, J., Ed., "Transmission Control Protocol: DARPA Internet Program Protocol Specification", RFC 793, September 1981, http://www.rfc-editor.org/rfc/rfc793.txt

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation 27, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[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/

1.2.2 Informative References

[MS-CSDLBI] Microsoft Corporation, "Conceptual Schema Definition File Format with Business Intelligence Annotations".

[MSDN-DEFDETAILS] Microsoft Corporation, "DefaultDetails Element (CSDLBI)", https://msdn.microsoft.com/en-us/library/hh230808.aspx

[MSDN-FSCMDX] Microsoft Corporation, "FORMAT_STRING Contents (MDX)", http://msdn.microsoft.com/en-us/library/ms146084.aspx

[MSDN-SQLXML] Microsoft Corporation, "SQLXML", http://msdn.microsoft.com/en-us/library/aa286527.aspx

[MSFT-ENTITYTYPE] Microsoft Corporation, "EntityType Element (CSDLBI)", https://technet.microsoft.com/en-us/library/hh212976.aspx

[XMLA] Microsoft Corporation and Hyperion Solutions Corporation, "XML for Analysis Specification, Version 1.1", November 2002, http://xml.coverpages.org/xmlaV11-20021120.pdf

1.3 Overview

The Microsoft SQL Server Analysis Services protocol provides methods for a client to communicate with, and perform operations on, an analysis server. The Analysis Services protocol is based on SOAP and XML for Analysis (XMLA) [XMLA] and supports TCP/IP as an underlying transport mechanism in addition to HTTP/HTTPS.

The base communication details of this protocol are specified in [MS-SSAS]: SQL Server Analysis Services.

The SQL Server Analysis Services Tabular protocol is an extension of the SQL Server Analysis Services protocol. This extension protocol provides additional protocol messages for Tabular databases that are at compatibility level 1200 or higher.

Note For the purposes of this document, "Tabular database" refers only to a Tabular database that is at compatibility level 1200 or higher.

A Tabular database is administered by executing a set of commands that include, but are not limited to, the following.

- XMLA-based command extensions allow an application to perform operations such as the following.

    - Create an object.
    - Alter an object.
    - Delete an object.
    - Refresh the data in an object.
- **JavaScript Object Notation (JSON)**-based [RFC4627] commands can perform essentially the same operations. The JSON commands are sent as the string content of the **Statement** element in an XMLA command.
- A client application can obtain the metadata of a Tabular database by using a set of DISCOVER requests. For more information about DISCOVER requests, see [MS-SSAS] and [XMLA]. The metadata that is returned by these Discover requests is made up of the same objects and properties that are managed by the Create, Alter, Delete, Refresh, and so on commands.

Section 2.2.5 defines each of the metadata objects and their properties. Section 3.1.5 defines each of the commands and references the common objects and properties that are defined in section 2.2.5.

Notes on the objects, their properties, and the commands include the following.

- The JSON APIs use a different naming convention than the XMLA APIs. The JSON convention uses camel casing for names. For example:
    - "Name" would be "name".
    - "DefaultMode" would be "defaultMode".
    - Etc.

    Therefore, the case of the properties and objects can be ignored in the text of this document.
- Some of the properties are read-only and cannot be set explicitly by any of the commands. These properties appear only in the Discover operations for these objects. For example, the **ModifiedTime** and **RefreshedTime** properties are implicitly updated by different commands and cannot be explicitly changed.
- Some properties are documented as ID-based object references. These represent links to other objects in the object tree. For example, the **SortByColumnID** property represents a reference to another column in the same table. The actual representation of object references is different between the JSON and XMLA commands and is described in the corresponding section.
- Some properties are documented as enumerations. Their descriptions contain numeric values and strings for each accepted value. For example, **SummarizeBy** shows Default (1), None (2), Sum (3), etc. The XMLA commands and the TMSCHEMA Discover operations use the integer values, and the JSON commands use the string values.

1.3.1 Object Ownership

Metadata objects are owned by other objects. For example, a **Table** object owns a collection of **Column** objects.

The two classifications of object ownership relationships are as follows.

Strongly Typed: An object type can have a collection of child objects of a particular type. For example, a **Table** has a collection of objects of type **Column**. This in turn means that each **Column** object will have a well-defined **Table** parent object.

Weakly Typed: An object type can own a shared object type. For example, an **Annotation** object type can belong to a **Model** object, a **Table** object, a **Column** object, and so forth. This in turn means that the shared object type can belong to different parent types.

The importance of recognizing the distinction between these two ownership scenarios is that commands that reference the parent/child object also specify the type of the parent.

Similarly, objects can have reference links to other objects (e.g., a **PerspectiveTable** object can link to a **Table** object) and these links can also be strongly typed or weakly typed.

In addition, it is important to recognize that sometimes objects have collections of child objects (e.g., a **Table** that has a collection of columns), and sometimes objects can have a single child object (e.g., a **Column** that has a single **AttributeHierarchy** child object).

1.3.2 Object References

The table in section 2.2.5 defines the **hierarchy** of metadata objects in a Tabular database. One of the consequences of the hierarchy of objects is that the commands that reference a particular object are able to use the names of the ancestor objects to identify the path to the object.

For example, a command to delete a PerspectiveColumn will reference the name of the PerspectiveTable and the name of the PerspectiveColumn to uniquely identify the PerspectiveColumn object.

Similarly, altering a Partition object will use the name of the Table that the Partition belongs to and the name of the Partition.

For illustration, the following sample JSON command creates or replaces the **DimDate 2** partition object in the **DimDate** table in the **Adventure Works** database.

```
{
  "createOrReplace": {
    "object": {
      "database": "Adventure Works",
      "table": "DimDate",
      "partition": "DimDate 2"
    },
    "partition": {
      "name": "DimDate 2",
      "source": {
        "dataSource": "AdventureworksDW",
        "query": [
          "SELECT [dbo].[DimDate].* FROM [dbo].[DimDate]\r",
          "where CalendarYear=2009"
        ]
      }
    }
  }
}
```

In addition to the name-based paths, XMLA-based commands also support object references based on integer IDs. The integer IDs are identifiers assigned by the server for each object when they are created. These IDs can be discovered and used in subsequent XMLA-based commands.

The difference in the object references is illustrated as follows by using the schema of the XMLA command to alter a partition.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
```

```
            <xs:sequence>
              <xs:element type="row" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:complexType name="row">
          <xs:sequence>
            <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
            <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
            <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
            ...
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
```

In this example, the ID field represents the integer identifier of the partition. The fields ID.Table and ID.Partition represent the name-based path to the **Partition** object. In an XMLA command, either the integer-based ID or the name-based path can be used to refer to the object being manipulated.

In JSON commands, the integer-based object reference is not supported. Only name-based paths to the objects can be used. The following JSON-based **Alter** command is an example of a JSON schema for object references.

```
"object": {
    "description": "Path for object Partition",
    "type": "object",
    "properties": {
        "database": {
            "type": "string"
        },
        "table": {
            "type": "string"
        },
        "partition": {
            "type": "string"
        }
    },
    "additionalProperties": false
},
```

In this case, referring to a partition requires specifying the name of the database, the name of the table, and the name of the partition.

1.4 Relationship to Other Protocols

Analysis Services uses the SOAP messaging protocol for formatting requests and responses as specified either in [SOAP1.1] or in [SOAP1.2-1/2007] and [SOAP1.2-2/2007]. It transmits these messages by using HTTP [RFC2616], HTTPS [RFC2818], or TCP [RFC793].

The base messaging protocol used for Analysis Services is defined in [MS-SSAS]. This document extends that protocol to add support for messages that apply to databases in Tabular mode at compatibility level 1200 or higher.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol supports the exchange of messages between a client and an analysis server.

1.7 Versioning and Capability Negotiation

1.7.1 Versioning

This protocol includes capabilities for a client and a server to exchange versioning information by indicating whether XML elements that are sent or received need to be understood, or, if not understood, can be ignored. This is specified in [MS-SSAS] section 2.2.4.2.1.3.

1.7.2 Capability Negotiation

This protocol does explicit negotiation between the client and the server for use of binary XML and compression, as specified in [MS-SSAS] section 2.1.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The transport protocol for the messages in this specification is defined in [MS-SSAS].

2.2 Common Data Types

This section contains common data types used by the SQL Server Analysis Services Tabular protocol. The syntax of the definitions uses **XML schemas** as defined in [XMLSCHEMA1/2] and [XMLSCHEMA2/2] and **Web Services Description Language (WSDL)** as defined in [WSDL].

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms that are specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

The following table contains common definitions used by the SQL Server Analysis Services Tabular protocol. The syntax of the definitions uses XML schemas as defined in [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and Web Services Description Language as defined in [WSDL].

Prefix	Namespace URI	Reference
xsd	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2] [XMLSCHEMA2/2]
xsi	http://www.w3.org/2001/XMLSchema-instance	[XMLSCHEMA1/2] [XMLSCHEMA2/2]
sql	urn:schemas-microsoft-com:xml-sql	[MSDN-SQLXML]
xmla	urn:schemas-microsoft-com:xml-analysis	[XMLA]
xmla-ds	urn:schemas-microsoft-com:xml-analysis:mddataset	[XMLA]
xmla-rs	urn:schemas-microsoft-com:xml-analysis:rowset	[XMLA]
xmla-e	urn:schemas-microsoft-com:xml-analysis:empty	[XMLA]
xmla-x	urn:schemas-microsoft-com:xml-analysis:exception	[XMLA]
xmla-m	http://schemas.microsoft.com/analysisservices/2003/xmla-multipleresults	[MS-SSAS]
eng	http://schemas.microsoft.com/analysisservices/2003/engine	[MS-SSAS]
eng2	http://schemas.microsoft.com/analysisservices/2003/engine/2	[MS-SSAS]
eng2_2	http://schemas.microsoft.com/analysisservices/2003/engine/2/2	[MS-SSAS]
eng100	http://schemas.microsoft.com/analysisservices/2008/engine/100	[MS-SSAS]
eng100_100	http://schemas.microsoft.com/analysisservices/2008/engine/100/100	[MS-SSAS]
eng200	http://schemas.microsoft.com/analysisservices/2010/engine/200	[MS-SSAS]
eng200_200	http://schemas.microsoft.com/analysisservices/2010/engine/200/200	[MS-SSAS]

Prefix	Namespace URI	Reference
eng300	http://schemas.microsoft.com/analysisservices/2011/engine/300	[MS-SSAS]
eng300_300	http://schemas.microsoft.com/analysisservices/2011/engine/300/300	[MS-SSAS]
eng400	http://schemas.microsoft.com/analysisservices/2012/engine/400	[MS-SSAS]
eng400_400	http://schemas.microsoft.com/analysisservices/2012/engine/400/400	[MS-SSAS]
eng500	http://schemas.microsoft.com/analysisservices/2013/engine/500	[MS-SSAS]
eng500_500	http://schemas.microsoft.com/analysisservices/2013/engine/500/500	[MS-SSAS]
engtab	http://schemas.microsoft.com/analysisservices/2014/engine	

2.2.2 Elements

The protocol elements in section 2.2.5 follow the same structure and style as the XMLA protocol in [XMLA] and [MS-SSAS].

The syntax is element-based. The elements follow the PascalCase naming style. The specific element names and document layout are defined by the **XML schema definition (XSD)** in the appropriate subsections under section 3.1.5.

Some of the commands use the **rowset** data type described in [XMLA] and [MS-SSAS]. The **rowset** data type allows the schema of the rowset to be defined inline by using an XSD schema. The schema of the rowsets allowed for these commands are defined in the appropriate subsections under section 3.1.5.

2.2.3 Complex Types

2.2.3.1 AffectedObjects

An application that uses the Tabular Metadata commands described in section 3.1.5.1.1 can set the **ReturnAffectedObjects** XMLA property. When this property is set to 1, the command will return an object in the **return** element of the **ExecuteResponse** element (see [MS-SSAS] section 3.1.4.3.2.2.1) called **AffectedObjects**.

The **AffectedObjects** element has the following attributes.

Attribute	Type	Description
name	string	The name of the database that was affected by the operation.
BaseVersion	integer	The version of the Tabular model before this operation was performed.
CurrentVersion	integer	The version of the Tabular model after this operation was performed.

The **AffectedObjects** element has the following child elements.

Element	Type	Description
root	Array of **rowset** objects	Zero or more rowset objects. Each rowset contains rows representing metadata objects that were affected by the operation.

The **rowset** object type is defined in [MS-SSAS].

Each root element adds the following attribute.

Attribute	Type	Description
name	string	The type of object that was affected by the operation.

The **name** attribute identifies the type of object that was affected.

The columns of the rowset correspond to the columns defined by the Discover response for that object type. The columns for each Discover response that are specific to a particular Discover operation are described with the operation in section 3.1.5.1.1.

In addition, the following column is appended to each rowset.

Column	Type	Default	Description
ImpactType	integer	0	0: The object in the row was modified by the operation. 1: The object in the row was deleted by the operation.

A client application can use the **AffectedObjects** object to determine the new state of all objects that were changed on the server as a result of the operation. A request to the server can indirectly affect more objects than the ones explicitly specified in the request.

The following is an example of the **AffectedObjects** response.

```
<return xmlns="urn:schemas-microsoft-com:xml-analysis">
  <AffectedObjects xmlns="http://schemas.microsoft.com/analysisservices/2003/xmla-
multipleresults" name="TMTestDB" BaseVersion="1" CurrentVersion="2">
    <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla" name="Model">
      <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
        <xsd:element name="root">
          <xsd:complexType>
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
              <xsd:element name="row" type="row" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name="uuid">
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-
[0-9a-zA-Z]{12}" />
          </xsd:restriction>
        </xsd:simpleType>
        <xsd:complexType name="xmlDocument">
```

```
          <xsd:sequence>
            <xsd:any />
          </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="row">
          <xsd:sequence>
            <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0" />
            <xsd:element sql:field="Name" name="Name" type="xsd:string" minOccurs="0" />
            <xsd:element sql:field="Description" name="Description" type="xsd:string"
minOccurs="0" />
            <xsd:element sql:field="StorageLocation" name="StorageLocation" type="xsd:string"
minOccurs="0" />
            <xsd:element sql:field="DefaultMode" name="DefaultMode" type="xsd:long"
minOccurs="0" />
            <xsd:element sql:field="DefaultDataView" name="DefaultDataView" type="xsd:long"
minOccurs="0" />
            <xsd:element sql:field="Culture" name="Culture" type="xsd:string" minOccurs="0"
/>
            <xsd:element sql:field="Collation" name="Collation" type="xsd:string"
minOccurs="0" />
            <xsd:element sql:field="ModifiedTime" name="ModifiedTime" type="xsd:dateTime"
minOccurs="0" />
            <xsd:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xsd:dateTime" minOccurs="0" />
            <xsd:element sql:field="Version" name="Version" type="xsd:long" minOccurs="0" />
            <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:schema>
      <row>
        <ID>1</ID>
        <Name>Model</Name>
        <Description>Model description</Description>
        <DefaultMode>0</DefaultMode>
        <DefaultDataView>0</DefaultDataView>
        <Culture>en-US</Culture>
        <ModifiedTime>2016-01-31T00:01:24.016667</ModifiedTime>
        <StructureModifiedTime>2016-01-31T00:01:24.13</StructureModifiedTime>
        <Version>2</Version>
        <ImpactType>1</ImpactType>
      </row>
    </root>
  </AffectedObjects>
</return>
```

2.2.4 Simple Types

Any new simple types used by this protocol are specified in section 3.1.5.

2.2.5 Common Data Structures

This section describes the hierarchy of metadata objects that can be discovered, defined, and administered by using the APIs in this specification. Sections 2.2.5.1 through 2.2.5.23 document the metadata objects and their properties for a Tabular database at compatibility level 1200 or higher.

The root object of a Tabular database is a **Model**. All other metadata objects are descendants of the **Model** object.

The following table illustrates the hierarchy structure. With the exception of **AttributeHierarchy**, **KPI**, and **LinguisticMetadata**, each child object can be a collection of child objects. For example, the **Model** object can contain a child object named **Tables**, which is a collection of **Table** objects, with each of those **Table** objects containing a child object named **Columns**, which is a collection of

Column objects, and so on. The lowest level descendant of any parent object in this hierarchy is an **Annotation** object.

Root Object	Descendant Level 1	Descendant Level 2	Descendant Level 3	Descendant Level 4
Model				
	DataSource			
		Annotation		
	Table			
		Column		
			AttributeHierarchy	
				Annotation
			Annotation	
		Partition		
			Annotation	
		Measure		
			KPI	
				Annotation
			Annotation	
		Hierarchy		
			Level	
				Annotation
			Annotation	
		Annotation		
	Relationship			
		Annotation		
	Perspective			
		PerspectiveTable		
			PerspectiveColumn	
				Annotation
			PerspectiveHierarchy	
				Annotation
			PerspectiveMeasure	
				Annotation
			Annotation	

Root Object	Descendant Level 1	Descendant Level 2	Descendant Level 3	Descendant Level 4
		Annotation		
	Culture			
		ObjectTranslation		
			Annotation	
		LinguisticMetadata		
			Annotation	
		Annotation		
	Role			
		RoleMembership		
			Annotation	
		TablePermission		
			Annotation	
		Annotation		
	Annotation			

2.2.5.1 Model Object

The **Model** object represents the Tabular data model. It is a child of the **Database** object as defined in [MS-SSAS]. All other Tabular metadata objects are descendants of the **Model** object.

The **Model** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
Name	string	The name of the object.
Description	string	The description of the object.
StorageLocation<1>	string	The location on disk to place the model.
DefaultMode	long	The default method for making data available in the partition.
DefaultDataView	enumeration	Determines which partitions are to be selected to run queries against the model. The possible values are as follows. ▪ **Full** (0) – Partitions with DataView set to Default or Full are selected. ▪ **Sample** (1) – Partitions with DataView set to Default or Sample are selected. ▪ **SampleAndFull** (2) – All partitions are selected. ▪ **Default** (3) – Not applicable to Model. It can be set only on

Name	Type	Description
		Partitions.
Culture	string	The culture name to use for formatting.<2>
Collation	string	The collation sequence.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.
Version	long	The current version of the Model. The version number is incremented when any transaction on the Model is committed. This version number is set to 1 for any newly created Tabular databases and is always set to 1 for all Tabular databases when the server is restarted.

2.2.5.2 DataSource Object

The **DataSource** object represents an external source of data. It is a child of a **Model** object.

The **DataSource** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ModelID	unsignedLong	An ID-based reference to a **Model** object.
Name	string	The name of the object.
Description	string	The description of the object.
Type	enumeration	The type of **DataSource**. At present, the only possible value is as follows. ▪ **Provider** (1) - Normal connection string.
ConnectionString	string	A string that is used to open the connection to the data source.
ImpersonationMode	enumeration	A numeric value that specifies the credentials to use for impersonation. The enumeration values are as follows. ▪ **ImpersonateAccount** (2) - The server uses the specified user account. ▪ **ImpersonateAnonymous** (3) - The server uses the anonymous user account. ▪ **ImpersonateCurrentUser** (4) - The server uses the user account that the client is connecting as. ▪ **ImpersonateServiceAccount** (5) - The server uses the user account that the server is running as. ▪ **ImpersonateUnattendedAccount** (6) – The server uses an unattended user account.<3>
Account	string	The user account that is used for impersonation.<4>
Password	string	The password that is used to impersonate the specified user account.
MaxConnections	int	The maximum number of connections to be opened concurrently to

Name	Type	Description
		the data source.
Isolation	enumeration	The kind of isolation that is used when executing commands against the data source. The possible values are as follows. ▪ **ReadCommitted** (1) ▪ **Snapshot** (2)<5>
Timeout	int	The timeout in seconds for commands executed against the data source.
Provider	string	An optional string that identifies the name of the managed data provider for the data source.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.3 Table Object

The **Table** object represents a Table in the data model. It is a child of a **Model** object. The **Table** object is defined to have a set of columns, and the rows in the tables are based on **Partition** child objects.

The **Table** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ModelID	unsignedLong	An ID-based reference to a **Model** object.
Name	string	The name of the object.
DataCategory	string	Specifies the type of Table. The values in the following **TableType** enumeration will automatically be mapped to the DIMENSION_TYPE column defined in [MS-SSAS] section 3.1.4.2.2.1.3.6.1. All other unknown strings will map to UNKNOWN and be returned in the **Contents** attribute of the **EntityType** element of CSDLBI (see [MSFT-ENTITYTYPE]). ▪ **Regular** (1) - standard dimension ▪ **Time** (2) - time dimension ▪ **Geography** (3) - geography dimension ▪ **Organization** (4) - organization dimension ▪ **BillOfMaterials** (5) - bill of materials dimension ▪ **Accounts** (6) - accounts dimension ▪ **Customers** (7) - customers dimension ▪ **Products** (8) - products dimension ▪ **Scenario** (9) - scenario dimension ▪ **Quantitative** (10) - quantitative dimension ▪ **Utility** (11) - utility dimension ▪ **Currency** (12) - currency dimension ▪ **Rates** (13) - rates dimension ▪ **Channel** (14) - channel dimension ▪ **Promotion** (15) - promotion dimension

Name	Type	Description
Description	string	The description of the object.
IsHidden	boolean	A Boolean that indicates whether the table is treated as hidden by client visualization tools. True if the Table is treated as hidden; otherwise false.
TableStorageID	unsignedLong	An ID-based reference to a **TableStorage** object. The **TableStorage** object is reserved for internal use only.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.
SystemFlags	long	A bitmask that is used to identify the type of object. The possible values are as follows. ▪ **Bit 0 is set to 1**: The object is a system table that is defined and built internally by the system. ▪ **Bit 1 is set to 1**: The object is a user-created calculated table.

2.2.5.4 Column Object

The **Column** object represents a column in a Table. It is a child of a **Table** object. Each column has a number of properties defined on it that influence how client applications visualize the data in the column.

The **Column** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
ExplicitName	string	The user-specified name for the column. This element MUST be specified for calculated columns and columns bound to data. Columns in calculated tables can leave this unspecified, and then the name will be inferred from the expression.
InferredName	string	Specifies the engine-generated name for the column. It is valid only for columns of type **CalculatedTableColumn**.
ExplicitDataType	enumeration	The user-specified data type to be enforced on the contents of the column. The possible values are as follows. ▪ **Automatic** (1) – When calculated columns or calculated table columns set the value to **Automatic**, the type is automatically inferred. ▪ **String** (2) ▪ **Int64** (6) ▪ **Double** (8) ▪ **DateTime** (9) ▪ **Decimal** (10)

Name	Type	Description
		▪ **Boolean** (11) ▪ **Binary** (17) ▪ **Unknown** (19) - This value cannot be set on the **ExplicitDataType** field. It is set automatically by the engine on the **InferredDataType** field of a calculated column that is in a semantic error state.
InferredDataType	enumeration	Specifies the engine-generated data type for this column. It is valid only for columns of the type **CalculatedTableColumn** or **Calculated**.
DataCategory	string	The values in the following enumeration **PropertyType** will automatically be mapped to the LEVEL_TYPE column defined in [MS-SSAS] section 3.1.4.2.2.1.3.8.1. All other strings will map to EXTENDEDTYPE (0x12B1) and be returned as-is in the **Contents** property of CSDL for the column. ▪ **Invalid** (-1) ▪ **All** (1) ▪ **Regular** (2) ▪ **Image** (3) ▪ **ImageBMP** (4) ▪ **ImageGIF** (5) ▪ **ImageJPG** (6) ▪ **ImagePNG** (7) ▪ **ImageTIFF** (8) ▪ **ImageURL** (9) ▪ **Id** (10) ▪ **RelationToParent** (11) ▪ **Sequence** (12) ▪ **OrgTitle** (13) ▪ **Caption** (14) ▪ **ShortCaption** (15) ▪ **CaptionDescription** (16) ▪ **CaptionAbbreviation** (17) ▪ **WebURL** (18) ▪ **WebHTML** (19) ▪ **WebXMLOrXSL** (20) ▪ **WebmailAlias** (21)

Name	Type	Description
		▪ **Address** (22) ▪ **AddressStreet** (23) ▪ **AddressHouse** (24) ▪ **AddressCity** (25) ▪ **AddressStateOrProvince** (26) ▪ **AddressZIP** (27) ▪ **AddressQuarter** (28) ▪ **AddressCountry** (29) ▪ **AddressBuilding** (30) ▪ **AddressRoom** (31) ▪ **AddressFloor** (32) ▪ **AddressFax** (33) ▪ **AddressPhone** (34) ▪ **GeoCentroidX** (35) ▪ **GeoCentroidY** (36) ▪ **GeoCentroidZ** (37) ▪ **GeoBoundaryTop** (38) ▪ **GeoBoundaryLeft** (39) ▪ **GeoBoundaryBottom** (40) ▪ **GeoBoundaryRight** (41) ▪ **GeoBoundaryFront** (42) ▪ **GeoBoundaryRear** (43) ▪ **GeoBoundaryPolygon** (44) ▪ **PhysicalSize** (45) ▪ **PhysicalColor** (46) ▪ **PhysicalWeight** (47) ▪ **PhysicalHeight** (48) ▪ **PhysicalWidth** (49) ▪ **PhysicalDepth** (50) ▪ **PhysicalVolume** (51) ▪ **PhysicalDensity** (52) ▪ **PersonFullName** (53)

Name	Type	Description
		▪ **PersonFirstName** (54) ▪ **PersonLastName** (55) ▪ **PersonMiddleName** (56) ▪ **PersonDemographic** (57) ▪ **PersonContact** (58) ▪ **QtyRangeLow** (59) ▪ **QtyRangeHigh** (60) ▪ **FormattingColor** (61) ▪ **FormattingOrder** (62) ▪ **FormattingFont** (63) ▪ **FormattingFontEffects** (64) ▪ **FormattingFontSize** (65) ▪ **FormattingSubtotal** (66) ▪ **Date** (67) ▪ **DateStart** (68) ▪ **DateEnded** (69) ▪ **DateCanceled** (70) ▪ **DateModified** (71) ▪ **DateDuration** (72) ▪ **Version** (73) ▪ **Years** (74) ▪ **Quarters** (75) ▪ **Months** (76) ▪ **Weeks** (77) ▪ **Days** (78) ▪ **Hours** (79) ▪ **Minutes** (80) ▪ **Seconds** (81) ▪ **UndefinedTime** (82) ▪ **OrganizationalUnit** (83) ▪ **BomResource** (84) ▪ **Quantitative** (85)

Name	Type	Description
		▪ **Account** (86) ▪ **Customers** (87) ▪ **CustomerGroup** (88) ▪ **CustomerHousehold** (89) ▪ **Product** (90) ▪ **ProductGroup** (91) ▪ **Scenario** (92) ▪ **Utility** (93) ▪ **Person** (94) ▪ **Company** (95) ▪ **CurrencySource** (96) ▪ **CurrencyDestination** (97) ▪ **Channel** (98) ▪ **Representative** (99) ▪ **Promotion** (100) ▪ **Continent** (101) ▪ **Region** (102) ▪ **Country** (103) ▪ **StateOrProvince** (104) ▪ **County** (105) ▪ **City** (106) ▪ **PostalCode** (107) ▪ **Point** (108) ▪ **AccountType** (109) ▪ **AccountName** (110) ▪ **AccountNumber** (111) ▪ **ProjectName** (112) ▪ **ProjectCode** (113) ▪ **ProjectStartDate** (114) ▪ **ProjectEndDate** (115) ▪ **ProjectCompletion** (116) ▪ **CurrencyName** (117)

Name	Type	Description
		▪ **CurrencyIsOCode** (118) ▪ **PercentOwnership** (119) ▪ **PercentVoteright** (120) ▪ **Project** (121) ▪ **RateType** (122) ▪ **Rate** (123) ▪ **ProductSKU** (124) ▪ **ProductCategory** (125) ▪ **ProductBrand** (126) ▪ **DeletedFlag** (127) ▪ **ScdStatus** (128) ▪ **ScdEndDate** (129) ▪ **ScdOriginalID** (130) ▪ **ScdStartDate** (131) ▪ **DayOfMonthOrPeriod** (132) ▪ **WeekOfQuarter** (133) ▪ **WeekOfMonthOrPeriod** (134) ▪ **MonthOrPeriodOfQuarter** (135) ▪ **MonthOrPeriodOfYear** (136) ▪ **Trimesters** (137) ▪ **Halfyears** (138) ▪ **Tendays** (139) ▪ **DayOfWeek** (140) ▪ **DayOfTendays** (141) ▪ **DayOfMonth** (142) ▪ **DayOfQuarter** (143) ▪ **DayOfTrimester** (144) ▪ **DayOfHalfyear** (145) ▪ **DayOfYear** (146) ▪ **WeekOfYear** (147) ▪ **TendayOfMonth** (148) ▪ **TendayOfQuarter** (149)

Name	Type	Description
		▪ **TendayOfTrimester** (150) ▪ **TendayOfHalfyear** (151) ▪ **TendayOfYear** (152) ▪ **MonthOfTrimester** (153) ▪ **MonthOfQuarter** (154) ▪ **MonthOfHalfyear** (155) ▪ **MonthOfYear** (156) ▪ **TrimesterOfYear** (157) ▪ **QuarterOfHalfyear** (158) ▪ **QuarterOfYear** (159) ▪ **HalfyearOfYear** (160) ▪ **FiscalDate** (161) ▪ **FiscalDayOfWeek** (162) ▪ **FiscalDayOfMonth** (163) ▪ **FiscalDayOfQuarter** (164) ▪ **FiscalDayOfTrimester** (165) ▪ **FiscalDayOfHalfyear** (166) ▪ **FiscalDayOfYear** (167) ▪ **FiscalWeeks** (168) ▪ **FiscalWeekOfYear** (169) ▪ **FiscalWeekOfHalfyear** (170) ▪ **FiscalWeekOfQuarter** (171) ▪ **FiscalWeekOfTrimester** (172) ▪ **FiscalWeekOfMonth** (173) ▪ **FiscalMonths** (174) ▪ **FiscalMonthOfTrimester** (175) ▪ **FiscalMonthOfQuarter** (176) ▪ **FiscalMonthOfHalfyear** (177) ▪ **FiscalMonthOfYear** (178) ▪ **FiscalTrimesters** (179) ▪ **FiscalTrimesterOfYear** (180) ▪ **FiscalQuarters** (181)

Name	Type	Description
		▪ **FiscalQuarterOfYear** (182) ▪ **FiscalQuarterOfHalfyear** (183) ▪ **FiscalHalfyears** (184) ▪ **FiscalHalfyearOfYear** (185) ▪ **FiscalYears** (186) ▪ **ReportingDate** (187) ▪ **ReportingDayOfWeek** (188) ▪ **ReportingDayOfMonth** (189) ▪ **ReportingDayOfQuarter** (190) ▪ **ReportingDayOfTrimester** (191) ▪ **ReportingDayOfHalfyear** (192) ▪ **ReportingDayOfYear** (193) ▪ **ReportingWeeks** (194) ▪ **ReportingWeekOfYear** (195) ▪ **ReportingWeekOfHalfyear** (196) ▪ **ReportingWeekOfQuarter** (197) ▪ **ReportingWeekOfTrimester** (198) ▪ **ReportingWeekOfMonth** (199) ▪ **ReportingMonths** (200) ▪ **ReportingMonthOfTrimester** (201) ▪ **ReportingMonthOfQuarter** (202) ▪ **ReportingMonthOfHalfyear** (203) ▪ **ReportingMonthOfYear** (204) ▪ **ReportingTrimesters** (205) ▪ **ReportingTrimesterOfYear** (206) ▪ **ReportingQuarters** (207) ▪ **ReportingQuarterOfYear** (208) ▪ **ReportingQuarterOfHalfyear** (209) ▪ **ReportingHalfyears** (210) ▪ **ReportingHalfyearOfYear** (211) ▪ **ReportingYears** (212) ▪ **ManufacturingDate** (213)

Name	Type	Description
		▪ **ManufacturingDayOfWeek** (214) ▪ **ManufacturingDayOfMonth** (215) ▪ **ManufacturingDayOfQuarter** (216) ▪ **ManufacturingDayOfHalfyear** (217) ▪ **ManufacturingDayOfYear** (218) ▪ **ManufacturingWeeks** (219) ▪ **ManufacturingWeekOfYear** (220) ▪ **ManufacturingWeekOfHalfyear** (221) ▪ **ManufacturingWeekOfQuarter** (222) ▪ **ManufacturingWeekOfMonth** (223) ▪ **ManufacturingMonths** (224) ▪ **ManufacturingMonthOfQuarter** (225) ▪ **ManufacturingMonthOfHalfyear** (226) ▪ **ManufacturingMonthOfYear** (227) ▪ **ManufacturingTrimesters** (228) ▪ **ManufacturingTrimesterOfYear** (229) ▪ **ManufacturingQuarters** (230) ▪ **ManufacturingQuarterOfYear** (231) ▪ **ManufacturingQuarterOfHalfyear** (232) ▪ **ManufacturingHalfyears** (233) ▪ **ManufacturingHalfyearOfYear** (234) ▪ **ManufacturingYears** (235) ▪ **WinterSummerSeason** (236) ▪ **IsHoliday** (237) ▪ **IsWeekday** (238) ▪ **IsWorkingDay** (239) ▪ **IsPeakDay** (240) ▪ **ISO8601Date** (241) ▪ **ISO8601DayOfWeek** (242) ▪ **ISO8601DayOfYear** (243) ▪ **ISO8601Weeks** (244) ▪ **ISO8601WeekOfYear** (245)

Name	Type	Description
		▪ **ISO8601Years** (246) ▪ **RowNumber** (247) ▪ **ExtendedType** (248)
Description	string	The description of the object.
IsHidden	boolean	A Boolean that indicates whether a column is treated as hidden by client visualization tools. True if the column is treated as hidden by client visualization tools; otherwise false.
State	enumeration	Provides information on the state of the column. The possible values and their interpretation are as follows. ▪ **Ready** (1) – The column is queryable and has up-to-date data. ▪ **NoData** (3) – The column is queryable but has no data. This state is applicable only to columns of the type **Data**. ▪ **CalculationNeeded** (4) – The column is not queryable and needs to be refreshed (that is, recalculated) to become functional. This state applies only to columns of the type **Calculated** or **CalculatedTableColumn**. ▪ **SemanticError** (5) **-** The column is in an error state because of an invalid expression. The column is not queryable. This state applies only to columns of the type **Calculated** or **CalculatedTableColumn**. ▪ **EvaluationError** (6) – The column is in an error state because of an error during expression evaluation. The column is not queryable. This state applies only to columns of the type **Calculated** or **CalculatedTableColumn**. ▪ **DependencyError** (7) – The column is in an error state because some of its calculation dependencies are in an error state. The column is not queryable. This state applies only to columns of the type **Calculated** or **CalculatedTableColumn**. ▪ **Incomplete** (8) - Some parts of the column have no data, and the column needs to be refreshed to bring the data in. The column is queryable. This state applies only to columns of the type **Data**. ▪ **SyntaxError** (9) - The column is in an error state because of a syntax error in its expression. The column is not queryable. This state applies only to columns of the type **Calculated**.
IsUnique	boolean	A Boolean that indicates whether the column can contain duplicate values. True if the engine will validate that this column cannot contain duplicate values; otherwise false.
IsKey	boolean	A Boolean that indicates whether the column is a key of the table. True if the column is a key of the table; otherwise false.
IsNullable	boolean	A Boolean that indicates whether null values are allowed in the column. True if null values are allowed in the column; otherwise false.

Name	Type	Description
Alignment	enumeration	Specifies the text alignment of the column in report visualizations. It is returned as part of CSDL. The possible values are as follows. ▪ **Default** (1) ▪ **Left** (2) ▪ **Right** (3) ▪ **Center** (4)
TableDetailPosition	int	Provides the ability to place this column in the **DefaultDetails** collection of the **Table**. This collection is an ordered set of **Column** types. A positive value indicates participation in the collection. The collection is sorted in ascending order of this element. The **DefaultDetails** collection is returned as part of the CSDL metadata returned by the DISCOVER_CSDL_METADATA operation (see [MS-SSAS] section 3.1.4.2.2.1.3.61).<6>
IsDefaultLabel	boolean	A Boolean that indicates whether this column is included in the **DisplayKey** element in CSDL.
IsDefaultImage	boolean	A Boolean that indicates whether this column is returned as the **DefaultImage** property in CSDL.
SummarizeBy	enumeration	A value indicating the default function, if any, typically used to aggregate this field. The possible values are as follows. ▪ **Default** (1) ▪ **None** (2) ▪ **Sum** (3) ▪ **Min** (4) ▪ **Max** (5) ▪ **Count** (6) ▪ **Average** (7) ▪ **DistinctCount** (8) If omitted, **Default** is assumed for numeric fields, **None** for all other fields.
ColumnStorageID	unsignedLong	An ID-based reference to a **ColumnStorage** object. The **ColumnStorage** object is reserved for internal use only.
Type	enumeration	The type of **Column**. The possible values are as follows. ▪ **Data** (1) – The contents of this column come from a **DataSource**. ▪ **Calculated** (2) – The contents of this column are computed by using an expression after the **Data** columns have been populated. ▪ **RowNumber** (3) - The column is an internal column that represents the row number. ▪ **CalculatedTableColumn** (4) – Tables that are built based on

Name	Type	Description
		a calculated expression will automatically infer and generate the columns in the table. Such columns will have this **Type**. See section 2.2.5.6 for setting the type of partition to **Calculated**.
SourceColumn	string	The name of the column from which data will be retrieved. The name MUST match a column returned by the execution of the partition's **QueryDefinition** against the data source.
ColumnOriginID	unsignedLong	An ID-based reference to a **ColumnOrigin** object.
Expression	string	The **Data Analysis Expressions (DAX)** expression that is evaluated for the calculated column.
FormatString	string	A string that specifies the format of the column contents. For a description of the **FormatString** content, see [MSDN-FSCMDX].
IsAvailableInMDX	boolean	A Boolean that indicates whether the column can be excluded from usage in **Multidimensional Expressions (MDX)** query tools. False if the column can be excluded from usage in MDX query tools; otherwise true.
SortByColumnID	unsignedLong	Indicates that the column defining this property will be sorted by the values of the column referenced by this property.
AttributeHierarchyID	unsignedLong	An ID-based reference to an **AttributeHierarchy** object.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.
RefreshedTime	dateTime	The time that the object was last refreshed.
SystemFlags	long	A bitmask that is used to identify the type of object. The possible values are as follows. ▪ **Bit 0 is set to 1**: The object is a column that belongs to a system table. See **SystemFlags** on the **Table** object defined in section 2.2.5.3. ▪ **Bit 1 is set to 1**: The object is a column that belongs to a calculated table of the type **CalculatedTableColumn**.
KeepUniqueRows	boolean	A Boolean that indicates the grouping of rows. If false, client applications can group by this column. If true, client applications are encouraged to group by a more unique key for the column. For an example, see [MS-CSDLBI] section 2.1.13.3. These semantics correspond to the following behavior: ▪ False: return the values of MD_GROUPING_BEHAVIOR_ENCOURAGE in the GROUPING_BEHAVIOR column of the MDSCHEMA_HIERARCHIES schema rowset and GroupOnValue in the GroupingBehavior field of the Property element in the result of DISCOVER_CSDL_METADATA. ▪ True: return MD_GROUPING_BEHAVIOR_DISCOURAGE and GroupOnEntityKey.
DisplayOrdinal	int	Indicates the visual position of the column, defined as a relative ordering rather than a strict ordering (example: 10, 20, 40, 50). It

Name	Type	Description
		allows client applications to maintain a consistent column position.
ErrorMessage	string	A string that explains the error state associated with the current object. It is set by the engine only when the state of the object is one of these three values: SemanticError, DependencyError, or EvaluationError. It is applicable only to columns of the type **Calculated** or **CalculatedTableColumn**. It will be empty for other column objects.
SourceProviderType	string	The original data type of the column as defined in the language of the data source. This data type is used to generate queries directly against the data source, for example in Direct Query mode.
DisplayFolder	string	Defines the display folder for the column, for use by clients.

2.2.5.5 AttributeHierarchy Object

The **AttributeHierarchy** object represents the **attribute hierarchy** of a column in a table. It is an optional child object of a **Column** object and is implicitly created by the server. When the attribute hierarchy is present, the column becomes available as a hierarchy and can be queried by using the MDX language.

The **AttributeHierarchy** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ColumnID	unsignedLong	An ID-based reference to a **Column** object.
State	long	Provides information on the state of the **AttributeHierarchy** object. The possible values and their interpretation are as follows. ▪ **Ready** (1) – The Attribute Hierarchy is queryable and has up-to-date data. ▪ **NoData** (3) – Not applicable to Attribute Hierarchies. ▪ **CalculationNeeded** (4) – The Attribute Hierarchy does not contain any data because it was not refreshed. There is no error associated with the attribute hierarchy. ▪ **SemanticError** (5) - Not applicable to Attribute Hierarchies. ▪ **EvaluationError** (6) - Not applicable to Attribute Hierarchies. ▪ **DependencyError** (7) – The column that is associated with this Attribute Hierarchy is in an error state (SemanticError, EvaluationError, or DependencyError). ▪ **Incomplete** (8) - Not applicable to Attribute Hierarchies. ▪ **SyntaxError** (9) - Not applicable to Attribute Hierarchies.
AttributeHierarchyStorageID	unsignedLong	An ID-based reference to an **AttributeHierarchyStorage**

Name	Type	Description
		object. The **AttributeHierarchyStorage** object is reserved for internal use only.
ModifiedTime	dateTime	The time that the object was last modified.
RefreshedTime	dateTime	The time that the object was last refreshed.

2.2.5.6 Partition Object

The **Partition** object represents a partition in a table. It is a child of a **Table** object. The partitions in a table define the data from external data sources that become available when the table is queried.

The **Partition** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
Name	string	The name of the object.
Description	string	The description of the object.
DataSourceID	unsignedLong	An ID-based reference to a **DataSource** object.
QueryDefinition	string	The text of the query to be executed when populating data into the partition.
State	enumeration	Provides information on the state of the partition. The possible values and their interpretation are as follows. ▪ **Ready** (1) –The partition is queryable and has up-to-date data. ▪ **NoData** (3) –The partition is queryable but has no data. This state applies only to partitions with a type other than **Calculated**. ▪ **CalculationNeeded** (4) – The partition is not queryable and needs to be refreshed (that is, recalculated) to become functional. This state applies only to partitions of the type **Calculated**. ▪ **SemanticError** (5) **–** The partition is in an error state because of an invalid expression and is not queryable. This state applies only to partitions of the type **Calculated**. ▪ **EvaluationError** (6) – The partition is in an error state because of an error during expression evaluation. The partition is not queryable. This state applies only to partitions of the type **Calculated**. ▪ **DependencyError** (7) – The partition is in an error state because some of its calculation dependencies are in an error state. The partition is not queryable. This state applies only to partitions of the type **Calculated**. ▪ **Incomplete** (8) - Some parts of the partition have no data, and the partition needs to be refreshed to bring the data in.

Name	Type	Description
		The partition is queryable. This state applies only to partitions of a type other than **Calculated**. ▪ **SyntaxError** (9) - The partition is in an error state because of a syntax error in its expression. The partition is not queryable. This state applies only to partitions of the type **Calculated**.
Type	enumeration	The type of partition. The possible values are as follows. ▪ **Query** (1) – The data **in** this partition is retrieved by executing a query against a **DataSource**. ▪ **Calculated** (2) – The data in this partition is populated by executing a calculated expression. ▪ **None** (3) – The data in this partition is populated by pushing a rowset of data to the server as part of the **Refresh** operation.
PartitionStorageID	unsignedLong	An ID-based reference to a **PartitionStorage** object. The **PartitionStorage** object is reserved for internal use only.
Mode	enumeration	Defines the method for making data available in the partition. The possible values are as follows. ▪ **Import** (0) – Data will be imported from a data source. ▪ **DirectQuery** (1) – Data will be queried dynamically from a data source. ▪ **Default** (2) - Only partitions can use this value. When set, the partition will inherit the **DefaultMode** of the **Model**. ▪ **Push** (3) – Data will be pushed into the partition. The **Mode** of a **Partition** can be set to **Default** (2), in which case it will inherit its **Mode** from the **DefaultMode** of the **Model**.
DataView	enumeration	Determines which partitions are selected to run queries against the model. The possible values are as follows. ▪ **Full** (0) – Partitions with DataView set to Default or Full are selected. ▪ **Sample** (1) – Partitions with DataView set to Default or Sample are selected. ▪ **SampleAndFull** (2) – All partitions are selected. ▪ **Default** (3) – Inherits from the default DataView of the Model object.
ModifiedTime	dateTime	The time that the object was last modified.
RefreshedTime	dateTime	The time that the object was last refreshed.
SystemFlags	long	A bitmask used to identify the type of object. The possible values are as follows. ▪ **Bit 0 is set to 1**: The object is a partition that belongs to a system table that is not accessible to users through **data definition language (DDL)**.

Name	Type	Description
		▪ **Bit 1 is set to 1**: The object is a partition that belongs to a calculated table.
ErrorMessage	string	The string that explains the error state associated with the current object. It is set by the engine only when the state of the object is one of these three values: SemanticError, DependencyError, or EvaluationError. This element applies only to partitions of the type **Calculated**.

2.2.5.7 Relationship Object

The **Relationship** object represents a logical relationship between two **Table** objects. It is a child of a **Model** object.

The **Relationship** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ModelID	unsignedLong	An ID-based reference to a **Model** object.
Name	string	The name of the object.
IsActive	boolean	A Boolean that indicates whether the relationship is marked as Active or Inactive. An Active relationship is automatically used for filtering across tables. An Inactive relationship can be used explicitly by DAX calculations with the USERELATIONSHIP function.
Type	enumeration	The type of **Relationship**. At present, the only possible value is as follows. ▪ **SingleColumn** (1) - Normal column-column relationship.
CrossFilteringBehavior	enumeration	Indicates how relationships influence filtering of data. The enumeration defines the possible behaviors. The possible values are as follows. ▪ **OneDirection** (1) - The rows selected in the "To" end of the relationship will automatically filter scans of the table in the "From" end of the relationship. ▪ **BothDirections** (2) - Filters on either end of the relationship will automatically filter the other table. ▪ **Automatic** (3) - The engine will analyze the relationships and choose one of the behaviors by using heuristics.
JoinOnDateBehavior	enumeration	When joining two date time columns, indicates whether to join on date and time parts or on date part only. ▪ **DateAndTime** (1) - When joining two date time columns, join on date and time parts. ▪ **DatePartOnly** (2) - When joining two date time columns, join on date part only.

Name	Type	Description
RelyOnReferentialIntegrity	boolean	Unused; reserved for future use.
FromTableID	unsignedLong	An ID-based reference to a table at the "From" end of the relationship.
FromColumnID	unsignedLong	An ID-based reference to a column at the "From" end of the relationship.
FromCardinality	enumeration	Indicates whether the "From" end of the relationship has a cardinality of **One (1)** or **Many (2).**
ToTableID	unsignedLong	An ID-based reference to a table at the "To" end of the relationship.
ToColumnID	unsignedLong	An ID-based reference to a column at the "To" end of the relationship.
ToCardinality	enumeration	Indicates whether the "To" end of the relationship has a cardinality of **One (1)** or **Many (2).**
State	enumeration	Provides information on the state of the relationship. The possible values and their interpretation are as follows. ▪ **Ready** (1) – The relationship is queryable and has up-to-date data. ▪ **NoData** (3) – Not applicable to relationships. ▪ **CalculationNeeded** (4) – The relationship does not contain any data because it was not refreshed. There is no error associated with the relationship. ▪ **SemanticError** (5) - Not applicable to Relationship. ▪ **EvaluationError** (6) - Not applicable to Relationship. ▪ **DependencyError** (7) – A dependency associated with this relationship is in an error state (SemanticError, EvaluationError, or DependencyError). ▪ **Incomplete** (8) - Not applicable to relationships. ▪ **SyntaxError** (9) - Not applicable to relationships.
RelationshipStorageID	unsignedLong	An ID-based reference to a **RelationshipStorage** object. The **RelationshipStorage** object is reserved for internal use only.
RelationshipStorage2ID	unsignedLong	An ID-based reference to a second **RelationshipStorage** object.
ModifiedTime	dateTime	The time that the object was last modified.
RefreshedTime	dateTime	The time that the object was last refreshed.
SecurityFilteringBehavior	enumeration	Indicates how relationships influence filtering of data when evaluating row-level security expressions. The possible values are as follows. ▪ **OneDirection** (1) - The rows selected in the "To" end of the relationship will automatically filter scans of the table in the "From" end of the relationship. ▪ **BothDirections** (2) - Filters on either end of the relationship

Name	Type	Description
		will automatically filter the other table.

2.2.5.8 Measure Object

The **Measure** object represents a value that is calculated based on an expression. It is a child of a **Table** object.

The **Measure** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
Name	string	The name of the object.
Description	string	The description of the object.
DataType	enumeration	The data type of the measure. The possible values are as follows. ▪ **String** (2) ▪ **Int64** (6) ▪ **Double** (8) ▪ **DateTime** (9) ▪ **Decimal** (10) ▪ **Boolean** (11) ▪ **Binary** (17) ▪ **Unknown** (19) - a measure in an error state. ▪ **Variant** (20) - a measure with varying data type.
Expression	string	The DAX expression that is evaluated for the calculated measure.
FormatString	string	A string that specifies the format of the measure contents. For a description of the **FormatString** content, see [MSDN-FSCMDX].
IsHidden	boolean	A Boolean that indicates whether the measure is treated as hidden by client visualization tools. True if the measure is treated as hidden by client visualization tools; otherwise false.
State	enumeration	Provides information on the state of the measure. The possible values and their interpretation are as follows. ▪ **Ready (1)** – The measure is queryable and has up-to-date data. ▪ **NoData (3)** – Not applicable to Measure.

Name	Type	Description
		▪ **CalculationNeeded (4)** – Not applicable to Measure. ▪ **SemanticError (5) –** The measure expression has a semantic error. ▪ **EvaluationError (6) -** Not applicable to Measure. ▪ **DependencyError (7)** – A dependency associated with this measure is in an error state (SemanticError, EvaluationError, or DependencyError). ▪ **Incomplete (8)** - Not applicable to Measure. ▪ **SyntaxError (9)** – The **measure** has a syntax error in its expression.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.
KPIID	unsignedLong	An ID-based reference to a **KPI** object.
IsSimpleMeasure	boolean	A Boolean that indicates whether the measure is an implicit measure that is automatically created by client tools to aggregate a field. Client applications can hide measures that have this flag set.
ErrorMessage	string	The string that explains the error state associated with the current object. It is set by the engine only when the state of the object is one of these three values: SemanticError, DependencyError or EvaluationError.
DisplayFolder	string	Defines the display folder for the Measure, for use by clients.

2.2.5.9 Hierarchy Object

The **Hierarchy** object represents a collection of **levels** that provide a logical hierarchical drilldown path for client applications. It is a child of a **Table** object.

The **Hierarchy** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
Name	string	The name of the object.
Description	string	The description of the object.
IsHidden	boolean	A Boolean that indicates whether the hierarchy is treated as hidden by client visualization tools. True if the hierarchy is treated as hidden; otherwise false.
State	enumeration	Provides information on the state of the hierarchy. The possible values and their interpretation are as follows. ▪ **Ready (1)** – The hierarchy is queryable and has up-to-date

Name	Type	Description
		data. ▪ **NoData (3)** – Not applicable to Hierarchy. ▪ **CalculationNeeded** – The hierarchy does not contain any data because it was not refreshed. There is no error associated with the hierarchy. ▪ **SemanticError (5) –** Not applicable to Hierarchy. ▪ **EvaluationError (6) -** Not applicable to Hierarchy. ▪ **DependencyError (7)** – A dependency associated with the hierarchy is in an error state (SemanticError, EvaluationError, or DependencyError). ▪ **Incomplete (8)** - Not applicable to Hierarchy.
HierarchyStorageID	unsignedLong	An ID-based reference to a **HierarchyStorage** object. The **HierarchyStorage** object is reserved for internal use only.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.
RefreshedTime	dateTime	The time that the object was last refreshed.
DisplayFolder	string	Defines the display folder for the hierarchy, for use by clients.

2.2.5.10 Level Object

The **Level** object represents a level in a hierarchy that provides a logical hierarchical drilldown path for client applications. It is a child of a **Hierarchy** object. The level is based on the values in a column.

The **Level** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
HierarchyID	unsignedLong	An ID-based reference to a **Hierarchy** object.
Ordinal	int	The position of the level within the hierarchy. The levels in the hierarchy MUST be properly ordered, starting with 1 and increasing monotonically.
Name	string	The name of the object.
Description	string	The description of the object.
ColumnID	unsignedLong	An ID-based reference to a **Column** object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.11 Annotation Object

The **Annotation** object represents application-specific name/value pairs for the parent object. The Analysis Services server is not expected to interpret annotations. Annotations can generally be defined as child objects of any logical metadata object in the Tabular model, as listed for the **ObjectType** property in the following table.

The **Annotation** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ObjectID	unsignedLong	An ID-based reference to the object.
ObjectType	int	The data type of the object specified by **ObjectID**. The possible values are as follows. ▪ **TM_TYPEID_Model** (1) ▪ **TM_TYPEID_DataSource** (2) ▪ **TM_TYPEID_Table** (3) ▪ **TM_TYPEID_Column** (4) ▪ **TM_TYPEID_AttributeHierarchy** (5) ▪ **TM_TYPEID_Partition** (6) ▪ **TM_TYPEID_Relationship** (7) ▪ **TM_TYPEID_Measure** (8) ▪ **TM_TYPEID_Hierarchy** (9) ▪ **TM_TYPEID_Level** (10) ▪ **TM_TYPEID_Annotation** (11) ▪ **TM_TYPEID_KPI** (12) ▪ **TM_TYPEID_Culture** (13) ▪ **TM_TYPEID_ObjectTranslation** (14) ▪ **TM_TYPEID_LinguisticMetadata** (15) ▪ **TM_TYPEID_Perspective** (29) ▪ **TM_TYPEID_PerspectiveTable** (30) ▪ **TM_TYPEID_PerspectiveColumn** (31) ▪ **TM_TYPEID_PerspectiveHierarchy** (32) ▪ **TM_TYPEID_PerspectiveMeasure** (33) ▪ **TM_TYPEID_Role** (34) ▪ **TM_TYPEID_RoleMembership** (35) ▪ **TM_TYPEID_TablePermission** (36)
Name	string	The name of the object.

Name	Type	Description
Value	string	The value of the annotation.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.12 KPI Object

The **KPI** object represents a **key performance indicator (KPI)** object. It is a child of a **Measure** object.

The **KPI** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
MeasureID	unsignedLong	An ID-based reference to a **Measure** object.
Description	string	The description of the object.
TargetDescription	string	The description of the target value of the KPI.
TargetExpression	string	An expression that evaluates to a number and indicates the goal for the KPI.
TargetFormatString	string	The format string to be used when presenting the target value for the KPI.
StatusGraphic	string	The recommended graphic to represent the status of this KPI.<7>
StatusDescription	string	A description of the **Status** value for the KPI.
StatusExpression	string	An expression that is used to calculate the status of the KPI.
TrendGraphic	string	A string that identifies the graphic to show for the trend of the KPI.
TrendDescription	string	A description of the trend value of the KPI.
TrendExpression	string	An expression representing the trend of the KPI.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.13 Culture Object

The **Culture** object represents a user culture. It is a child of a **Model** object. The **Culture** object is used for translating strings and formatting values.

The **Culture** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ModelID	unsignedLong	An ID-based reference to a **Model** object.

Name	Type	Description
Name	string	The name of the object.
LinguisticMetadataID	unsignedLong	An ID-based reference to a **LinguisticMetadata** object.
ModifiedTime	dateTime	The time that the object was last modified.
StructureModifiedTime	dateTime	The time that the structure of the object was last modified.

2.2.5.14 ObjectTranslation Object

The **ObjectTranslation** object represents the translations of metadata properties for the **Culture** parent object. Properties like the Name and Description of a metadata object can be translated. If they are not translated, the properties specified on the main object are used.

The **ObjectTranslation** object has a weakly typed reference to the object that it is translating. For information on the distinction between strongly typed and weakly typed, see section 1.3.1.

The **ObjectTranslation** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
CultureID	unsignedLong	An ID-based reference to a **Culture** object.
ObjectID	unsignedLong	An ID-based reference to the object.
ObjectType	int	The data type of the object specified by **ObjectID**. The possible values are as follows. ▪ **TM_TYPEID_Model** (1) ▪ **TM_TYPEID_DataSource** (2) ▪ **TM_TYPEID_Table** (3) ▪ **TM_TYPEID_Column** (4) ▪ **TM_TYPEID_AttributeHierarchy** (5) ▪ **TM_TYPEID_Partition** (6) ▪ **TM_TYPEID_Relationship** (7) ▪ **TM_TYPEID_Measure** (8) ▪ **TM_TYPEID_Hierarchy** (9) ▪ **TM_TYPEID_Level** (10) ▪ **TM_TYPEID_Annotation** (11) ▪ **TM_TYPEID_KPI** (12) ▪ **TM_TYPEID_Culture** (13) ▪ **TM_TYPEID_ObjectTranslation** (14)

Name	Type	Description
		▪ **TM_TYPEID_LinguisticMetadata** (15) ▪ **TM_TYPEID_Perspective** (29) ▪ **TM_TYPEID_PerspectiveTable** (30) ▪ **TM_TYPEID_PerspectiveColumn** (31) ▪ **TM_TYPEID_PerspectiveHierarchy** (32) ▪ **TM_TYPEID_PerspectiveMeasure** (33) ▪ **TM_TYPEID_Role** (34) ▪ **TM_TYPEID_RoleMembership** (35) ▪ **TM_TYPEID_TablePermission** (36)
Property	enumeration	Specifies which property of the object is being translated. The possible values are as follows. ▪ **Invalid** (-1) - Default invalid value. ▪ **Caption** (1) - Object caption (shown instead of the name when available). ▪ **Description** (2) - Object description. ▪ **DisplayFolder** (3) – DisplayFolder property.
Value	string	The value of the translation.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.15 LinguisticMetadata Object

The **LinguisticMetadata** object is used to hold synonym information for the Tabular model. It is a child of a **Culture** object.

The **LinguisticMetadata** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
CultureID	unsignedLong	An ID-based reference to a **Culture** object.
Content	string	A string that contains the natural language synonyms.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.16 Perspective Object

The **Perspective** object defines a logical view over the **Model** and is a child of a **Model** object. It allows hiding Tables, Columns, Measures, and Hierarchies so that end users can look at a smaller subset of the large data model.

The **Perspective** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
ModelID	unsignedLong	An ID-based reference to a **Model** object.
Name	string	The name of the object.
Description	string	The description of the object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.17 PerspectiveTable Object

The **PerspectiveTable** object includes a Table into the Perspective. It is a child of a **Perspective** object. The PerspectiveColumns, PerspectiveMeasures, and PerspectiveHierarchies child objects allow customizing which parts of the Table are visible in the Perspective.

The **PerspectiveTable** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
PerspectiveID	unsignedLong	An ID-based reference to a **Perspective** object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
IncludeAll	boolean	A Boolean that indicates whether all **Column**, **Hierarchy**, and **Measure** objects in the **Table** object are automatically included into the perspective, If true, the objects are automatically included; otherwise, the **PerspectiveColumn**, **PerspectiveHierarchy** and **PerspectiveMeasure** objects need to be explicitly added to the **PerspectiveTable** object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.18 PerspectiveColumn Object

The **PerspectiveColumn** object includes a Column of a Table into the Perspective. It is a child of a **PerspectiveTable** object.

The **PerspectiveColumn** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.

Name	Type	Description
PerspectiveTableID	unsignedLong	An ID-based reference to a **PerspectiveTable** object.
ColumnID	unsignedLong	An ID-based reference to a **Column** object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.19 PerspectiveHierarchy Object

The **PerspectiveHierarchy** object includes a Hierarchy of a Table into the Perspective. It is a child of a **PerspectiveTable** object.

The **PerspectiveHierarchy** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
PerspectiveTableID	unsignedLong	An ID-based reference to a **PerspectiveTable** object.
HierarchyID	unsignedLong	An ID-based reference to a **Hierarchy** object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.20 PerspectiveMeasure Object

The **PerspectiveMeasure** object includes a Measure of a Table into the Perspective. It is a child of a **PerspectiveTable** object.

The **PerspectiveMeasure** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
PerspectiveTableID	unsignedLong	An ID-based reference to a **PerspectiveTable** object.
MeasureID	unsignedLong	An ID-based reference to a **Measure** object.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.21 Role Object

The **Role** object defines a set of user principals for whom security rules are applied. It is a child of a **Model** object.

The **Role** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.

Name	Type	Description
ModelID	unsignedLong	An ID-based reference to a **Model** object.
Name	string	The name of the object.
Description	string	The description of the object.
ModelPermission	enumeration	The level of access for this role. The possible values are as follows. ▪ **None** (1) - The role has no access to the model. ▪ **Read** (2) - The role can read metadata and data of the model. ▪ **ReadRefresh** (3) - The role has read and refresh permission. ▪ **Refresh** (4) - The role can refresh the data and calculations in the model. ▪ **Administrator** (5) - The role can administer the model.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.22 RoleMembership Object

The **RoleMembership** object defines a user principal that belongs to the Role. It is a child of a **Role** object.

The **RoleMembership** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
RoleID	unsignedLong	An ID-based reference to a **Role** object.
MemberName	string	The security name that identifies the user or group of the member.
MemberID	string	A string that uniquely identifies the member.
IdentityProvider	string	A string that defines the identity provider that MUST be used for authentication of a user.<8>
MemberType	enumeration	Indicates whether the particular member of a security role is an individual user or a group of users, or if the member is automatically detected.<9> The possible values are as follows. ▪ **Auto** (1) – Member of security role is automatically detected. ▪ **User** (2) – Member of security role is an individual user. ▪ **Group** (3) - Member of security role is a group of users.
ModifiedTime	dateTime	The time that the object was last modified.

2.2.5.23 TablePermission Object

The **TablePermission** object defines the security rules of the Role on the Table. It is a child of a **Role** object.

The **TablePermission** object has the following properties.

Name	Type	Description
ID	unsignedLong	A reference to the object.
RoleID	unsignedLong	An ID-based reference to a **Role** object.
TableID	unsignedLong	An ID-based reference to a **Table** object.
FilterExpression	string	The DAX expression that filters the rows in the table when this security role is in effect.
ModifiedTime	dateTime	The time that the object was last modified.
State	enumeration	Provides information on the state of the permission. The possible values and their interpretation are as follows. ▪ **Ready** (1) – The permission has a valid expression. ▪ **NoData** (3) – Not applicable. ▪ **CalculationNeeded** (4) – Not applicable. ▪ **SemanticError** (5) – The expression of the **TablePermission** object has a semantic error. The table expression cannot be executed and the role will not have access to the table. ▪ **EvaluationError** (6) **-** Not applicable. ▪ **DependencyError** (7) – A dependency associated with this **TablePermission** object is in an error state (SemanticError, EvaluationError, or DependencyError). The table expression cannot be executed and the role will not have access to the table. ▪ **Incomplete** (8) - Not applicable. ▪ **SyntaxError** (9) - The **TablePermission** object is in an error state because of a syntax error in its expression. The **TablePermission** object is not queryable. This state applies only to **TablePermission** objects of the type **Calculated**. The table expression cannot be executed and the role will not have access to the table.
ErrorMessage	string	A string that explains the error state associated with the current object. It is set by the engine only when the state of the object is one of these three values: SemanticError, DependencyError, or EvaluationError.

2.2.5.24 Common Restrictions for Discover Operations

One or more of the following restrictions can apply to a Discover operation.

Restriction	Type	Description
DatabaseName	string	The name of the database from which to return the metadata. If this restriction applies, the Discover operation returns the metadata objects from only the specified database. If this restriction is not specified, the current database of the session is used to restrict the results.
SystemObjectType	enumeration	A bitmask that specifies whether system objects are included or excluded. The possible values are as follows. ▪ 0x1: include user objects. This the default value. ▪ 0x2: include system objects.
ModifiedTimeOp	TimeRestrictionOp	Can apply to a Discover operation that includes one or more **dateTime** fields. The possible values are as follows. ▪ TIME_RESTRICTION_NEWER (0). This is the default value. ▪ TIME_RESTRICTION_OLDER (1).
StructureModifiedTimeOp	TimeRestrictionOp	Can apply to a Discover operation that includes one or more **dateTime** fields. The possible values are as follows. ▪ TIME_RESTRICTION_NEWER (0). This is the default value. ▪ TIME_RESTRICTION_OLDER (1).
RefreshedTimeOp	TimeRestrictionOp	Can apply to a Discover operation that includes one or more **dateTime** fields. The possible values are as follows. ▪ TIME_RESTRICTION_NEWER (0). This is the default value. ▪ TIME_RESTRICTION_OLDER (1).

The applicability of these restrictions is identified in the subsections of section 3.1.5.1.1.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

See [MS-SSAS] section 3.1.1.

3.1.2 Timers

None.

3.1.3 Initialization

See [MS-SSAS] section 3.1.3.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Discover

The Discover operation is used to find information about the server. For more information about the messaging protocol for Discover operations, see [MS-SSAS] section 3.1.4.2.

The Tabular Metadata Discover requests extend the types of objects that can be discovered to support objects that describe the Tabular Metadata.

The rowset type that is returned by all the Tabular Metadata Discovers inherits from the rowset type that is defined in [MS-SSAS] section 2.2.4.1.3 as follows.

```
<xs:element name="root" type="TabularDiscoverRowsetType" />
<xs:complexType name="TabularDiscoverRowsetType">
  <xs:complexContent>
    <xs:extension base="xmla-rs:rowset">
      <xs:attribute name="name" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

These schemas add to the set of schemas documented in [MS-SSAS] section 3.1.4.2.2.1.3.

3.1.5.1.1 Messages

The request and response messages for a Discover operation are defined in [MS-SSAS] section 3.1.4.2.1.

3.1.5.1.1.1 TMSCHEMA_MODEL

The TMSCHEMA_MODEL schema rowset specifies a model object in the database.

####### 3.1.5.1.1.1.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_MODEL. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.1.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.1.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.1.2.1 Columns

The TMSCHEMA_MODEL rowset contains the following columns.

Name	Restriction
ID	Yes
Name	Yes
Description	Yes
StorageLocation	Yes
DefaultMode	Yes
DefaultDataView	Yes
Culture	Yes
Collation	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes
Version	Yes

The name attribute in the TabularDiscoverRowsetType is set to Model. The XML schema definition (XSD) for the TMSCHEMA_MODEL rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverModelRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverModelRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="StorageLocation" name="StorageLocation" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="DefaultMode" name="DefaultMode" type="xs:long" minOccurs="0" />
      <xs:element sql:field="DefaultDataView" name="DefaultDataView" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="Culture" name="Culture" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Collation" name="Collation" type="xs:string" minOccurs="0" />
```

```
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
      <xs:element sql:field="Version" name="Version" type="xs:long" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.1.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_MODEL rowset.

- DatabaseName
- ModifiedTimeOp
- StructureModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.2 TMSCHEMA_DATA_SOURCES

The TMSCHEMA_DATA_SOURCES schema rowset provides information about the **DataSource** objects in the model.

3.1.5.1.1.2.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_DATA_SOURCES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.2.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.2.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.2.2.1 Columns

The TMSCHEMA_DATA_SOURCES rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
Description	Yes
Type	Yes
ConnectionString	Yes
ImpersonationMode	Yes
Account	Yes

Name	Restriction
Password	
MaxConnections	
Isolation	Yes
Timeout	Yes
Provider	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to DataSource. The XML schema definition (XSD) for the TMSCHEMA_DATA_SOURCES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverDataSourceRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverDataSourceRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="Type" name="Type" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ConnectionString" name="ConnectionString" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="ImpersonationMode" name="ImpersonationMode" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="Account" name="Account" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Password" name="Password" type="xs:string" minOccurs="0" />
      <xs:element sql:field="MaxConnections" name="MaxConnections" type="xs:int"
minOccurs="0" />
      <xs:element sql:field="Isolation" name="Isolation" type="xs:long" minOccurs="0" />
      <xs:element sql:field="Timeout" name="Timeout" type="xs:int" minOccurs="0" />
      <xs:element sql:field="Provider" name="Provider" type="xs:string" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.2.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_DATA_SOURCES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.3 TMSCHEMA_TABLES

The TMSCHEMA_TABLES schema rowset provides information about the **Table** objects in the model.

3.1.5.1.1.3.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_TABLES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.3.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.3.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.3.2.1 Columns

The TMSCHEMA_TABLES rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
DataCategory	Yes
Description	Yes
IsHidden	Yes
TableStorageID	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes
SystemFlags	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Table. The XSD for the TMSCHEMA_TABLES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverTableRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverTableRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="DataCategory" name="DataCategory" type="xs:string" minOccurs="0"
/>
```

```
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="IsHidden" name="IsHidden" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="TableStorageID" name="TableStorageID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
      <xs:element sql:field="SystemFlags" name="SystemFlags" type="xs:long" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.3.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_TABLES rowset.

- DatabaseName
- SystemObjectType
- ModifiedTimeOp
- StructureModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.4 TMSCHEMA_COLUMNS

The TMSCHEMA_COLUMNS schema rowset provides information about the **Column** objects in each table.

3.1.5.1.1.4.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_COLUMNS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.4.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.4.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.4.2.1 Columns

The TMSCHEMA_COLUMNS rowset contains the following columns.

Name	Restriction
ID	Yes
TableID	Yes
ExplicitName	Yes
InferredName	Yes
ExplicitDataType	Yes

Name	Restriction
InferredDataType	Yes
DataCategory	Yes
Description	Yes
IsHidden	Yes
State	Yes
IsUnique	Yes
IsKey	Yes
IsNullable	Yes
Alignment	Yes
TableDetailPosition	Yes
IsDefaultLabel	Yes
IsDefaultImage	Yes
SummarizeBy	Yes
ColumnStorageID	Yes
Type	Yes
SourceColumn	Yes
ColumnOriginID	Yes
Expression	Yes
FormatString	Yes
IsAvailableInMDX	Yes
SortByColumnID	Yes
AttributeHierarchyID	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes
RefreshedTime	Yes
SystemFlags	Yes
KeepUniqueRows	Yes
DisplayOrdinal	Yes
ErrorMessage	Yes
SourceProviderType	Yes
DisplayFolder	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Column. The XSD for the TMSCHEMA_COLUMNS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverColumnRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverColumnRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ExplicitName" name="ExplicitName" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="InferredName" name="InferredName" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="ExplicitDataType" name="ExplicitDataType" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="InferredDataType" name="InferredDataType" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="DataCategory" name="DataCategory" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="IsHidden" name="IsHidden" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="IsUnique" name="IsUnique" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="IsKey" name="IsKey" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="IsNullable" name="IsNullable" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="Alignment" name="Alignment" type="xs:long" minOccurs="0" />
      <xs:element sql:field="TableDetailPosition" name="TableDetailPosition" type="xs:int"
minOccurs="0" />
      <xs:element sql:field="IsDefaultLabel" name="IsDefaultLabel" type="xs:boolean"
minOccurs="0" />
      <xs:element sql:field="IsDefaultImage" name="IsDefaultImage" type="xs:boolean"
minOccurs="0" />
      <xs:element sql:field="SummarizeBy" name="SummarizeBy" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ColumnStorageID" name="ColumnStorageID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="Type" name="Type" type="xs:long" minOccurs="0" />
      <xs:element sql:field="SourceColumn" name="SourceColumn" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="ColumnOriginID" name="ColumnOriginID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="Expression" name="Expression" type="xs:string" minOccurs="0" />
      <xs:element sql:field="FormatString" name="FormatString" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="IsAvailableInMDX" name="IsAvailableInMDX" type="xs:boolean"
minOccurs="0" />
      <xs:element sql:field="SortByColumnID" name="SortByColumnID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="AttributeHierarchyID" name="AttributeHierarchyID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
      <xs:element sql:field="RefreshedTime" name="RefreshedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="SystemFlags" name="SystemFlags" type="xs:long" minOccurs="0" />
      <xs:element sql:field="KeepUniqueRows" name="KeepUniqueRows" type="xs:boolean"
minOccurs="0" />
      <xs:element sql:field="DisplayOrdinal" name="DisplayOrdinal" type="xs:int"
minOccurs="0" />
```

```
      <xs:element sql:field="ErrorMessage" name="ErrorMessage" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="SourceProviderType" name="SourceProviderType" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.4.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_COLUMNS rowset.

- DatabaseName
- SystemObjectType
- ModifiedTimeOp
- StructureModifiedTimeOp
- RefreshedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.5 TMSCHEMA_ATTRIBUTE_HIERARCHIES

The TMSCHEMA_ATTRIBUTE_HIERARCHIES schema rowset provides information about the AttributeHierarchy objects for a column.

3.1.5.1.1.5.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_ATTRIBUTE_HIERARCHIES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.5.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.5.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.5.2.1 Columns

The TMSCHEMA_ATTRIBUTE_HIERARCHIES rowset contains the following columns.

Name	Restriction
ID	Yes
ColumnID	Yes
State	Yes
AttributeHierarchyStorageID	Yes
ModifiedTime	Yes
RefreshedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to AttributeHierarchy. The XSD for the TMSCHEMA_ATTRIBUTE_HIERARCHIES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverAttributeHierarchyRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverAttributeHierarchyRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ColumnID" name="ColumnID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="AttributeHierarchyStorageID" name="AttributeHierarchyStorageID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="RefreshedTime" name="RefreshedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.5.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_ATTRIBUTE_HIERARCHIES rowset.

- DatabaseName
- ModifiedTimeOp
- RefreshedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.6 TMSCHEMA_PARTITIONS

The TMSCHEMA_PARTITIONS schema rowset provides information about the **Partition** objects in each table.

3.1.5.1.1.6.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PARTITIONS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.6.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.6.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.6.2.1 Columns

The TMSCHEMA_PARTITIONS rowset contains the following columns.

Name	Restriction
ID	Yes
TableID	Yes
Name	Yes
Description	Yes
DataSourceID	Yes
QueryDefinition	Yes
State	Yes
Type	Yes
PartitionStorageID	Yes
Mode	Yes
DataView	Yes
ModifiedTime	Yes
RefreshedTime	Yes
SystemFlags	Yes
ErrorMessage	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Partition. The XSD for the TMSCHEMA_PARTITIONS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPartitionRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPartitionRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="DataSourceID" name="DataSourceID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="QueryDefinition" name="QueryDefinition" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="Type" name="Type" type="xs:long" minOccurs="0" />
      <xs:element sql:field="PartitionStorageID" name="PartitionStorageID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Mode" name="Mode" type="xs:long" minOccurs="0" />
      <xs:element sql:field="DataView" name="DataView" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
```

```
      <xs:element sql:field="RefreshedTime" name="RefreshedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="SystemFlags" name="SystemFlags" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ErrorMessage" name="ErrorMessage" type="xs:string" minOccurs="0"
/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.6.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PARTITIONS rowset.

- DatabaseName
- SystemObjectType
- ModifiedTimeOp
- RefreshedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.7 TMSCHEMA_RELATIONSHIPS

The TMSCHEMA_RELATIONSHIPS schema rowset provides information about the **Relationship** objects in the model.

3.1.5.1.1.7.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_RELATIONSHIPS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.7.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.7.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.7.2.1 Columns

The TMSCHEMA_RELATIONSHIPS rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
IsActive	Yes
Type	Yes
CrossFilteringBehavior	Yes
JoinOnDateBehavior	Yes

Name	Restriction
RelyOnReferentialIntegrity	Yes
FromTableID	Yes
FromColumnID	Yes
FromCardinality	Yes
ToTableID	Yes
ToColumnID	Yes
ToCardinality	Yes
State	Yes
RelationshipStorageID	Yes
RelationshipStorage2ID	Yes
ModifiedTime	Yes
RefreshedTime	Yes
SecurityFilteringBehavior	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Relationship. The XSD for the TMSCHEMA_RELATIONSHIPS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverRelationshipRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverRelationshipRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="IsActive" name="IsActive" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="Type" name="Type" type="xs:long" minOccurs="0" />
      <xs:element sql:field="CrossFilteringBehavior" name="CrossFilteringBehavior"
type="xs:long" minOccurs="0" />
      <xs:element sql:field="JoinOnDateBehavior" name="JoinOnDateBehavior" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="RelyOnReferentialIntegrity" name="RelyOnReferentialIntegrity"
type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="FromTableID" name="FromTableID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="FromColumnID" name="FromColumnID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="FromCardinality" name="FromCardinality" type="xs:long"
minOccurs="0" />
      <xs:element sql:field="ToTableID" name="ToTableID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ToColumnID" name="ToColumnID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="ToCardinality" name="ToCardinality" type="xs:long" minOccurs="0"
/>
```

```
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="RelationshipStorageID" name="RelationshipStorageID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="RelationshipStorage2ID" name="RelationshipStorage2ID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="RefreshedTime" name="RefreshedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="SecurityFilteringBehavior" name="SecurityFilteringBehavior"
type="xs:long" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.7.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_RELATIONSHIPS rowset.

- DatabaseName
- ModifiedTimeOp
- RefreshedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.8 TMSCHEMA_MEASURES

The TMSCHEMA_MEASURES schema rowset provides information about the **Measure** objects in each table.

3.1.5.1.1.8.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_MEASURES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.8.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.8.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.8.2.1 Columns

The TMSCHEMA_MEASURES rowset contains the following columns.

Name	Restriction
ID	Yes
TableID	Yes
Name	Yes
Description	Yes
DataType	Yes

Name	Restriction
Expression	Yes
FormatString	Yes
IsHidden	Yes
State	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes
KPIID	Yes
IsSimpleMeasure	Yes
ErrorMessage	Yes
DisplayFolder	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Measure. The XSD for the TMSCHEMA_MEASURES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverMeasureRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverMeasureRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="DataType" name="DataType" type="xs:long" minOccurs="0" />
      <xs:element sql:field="Expression" name="Expression" type="xs:string" minOccurs="0" />
      <xs:element sql:field="FormatString" name="FormatString" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="IsHidden" name="IsHidden" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
      <xs:element sql:field="KPIID" name="KPIID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="IsSimpleMeasure" name="IsSimpleMeasure" type="xs:boolean"
minOccurs="0" />
      <xs:element sql:field="ErrorMessage" name="ErrorMessage" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.8.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_MEASURES rowset.

- DatabaseName
- ModifiedTimeOp
- StructureModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.9 TMSCHEMA_HIERARCHIES

The TMSCHEMA_HIERARCHIES schema rowset provides information about the **Hierarchy** objects in each table.

3.1.5.1.1.9.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_HIERARCHIES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.9.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.9.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.9.2.1 Columns

The TMSCHEMA_HIERARCHIES rowset contains the following columns.

Name	Restriction
ID	Yes
TableID	Yes
Name	Yes
Description	Yes
IsHidden	Yes
State	Yes
HierarchyStorageID	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes
RefreshedTime	Yes
DisplayFolder	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Hierarchy. The XSD for the TMSCHEMA_HIERARCHIES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
```

```
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverHierarchyRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverHierarchyRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="IsHidden" name="IsHidden" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="HierarchyStorageID" name="HierarchyStorageID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
      <xs:element sql:field="RefreshedTime" name="RefreshedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.9.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_HIERARCHIES rowset.

- DatabaseName
- ModifiedTimeOp
- StructureModifiedTimeOp
- RefreshedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.10 TMSCHEMA_LEVELS

The TMSCHEMA_LEVELS schema rowset provides information about the **Level** objects in each hierarchy.

3.1.5.1.1.10.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_LEVELS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.10.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.10.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.10.2.1 Columns

The TMSCHEMA_LEVELS rowset contains the following columns.

Name	Restriction
ID	Yes
HierarchyID	Yes
Ordinal	Yes
Name	Yes
Description	Yes
ColumnID	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Level. The XSD for the TMSCHEMA_LEVELS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverLevelRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverLevelRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="HierarchyID" name="HierarchyID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="Ordinal" name="Ordinal" type="xs:int" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="ColumnID" name="ColumnID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.10.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_LEVELS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.11 TMSCHEMA_ANNOTATIONS

The TMSCHEMA_ANNOTATIONS schema rowset provides information about the **Annotation** objects in the model.

3.1.5.1.1.11.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_ANNOTATIONS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.11.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.11.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.11.2.1 Columns

The TMSCHEMA_ANNOTATIONS rowset contains the following columns.

Name	Restriction
ID	Yes
ObjectID	Yes
ObjectType	Yes
Name	Yes
Value	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Annotation. The XSD for the TMSCHEMA_ANNOTATIONS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverAnnotationRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverAnnotationRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ObjectID" name="ObjectID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ObjectType" name="ObjectType" type="xs:int" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Value" name="Value" type="xs:string" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.11.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_ANNOTATIONS rowset.

- DatabaseName

- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.12 TMSCHEMA_KPIS

The TMSCHEMA_KPIS schema rowset provides information about the **KPI** objects in the model.

3.1.5.1.1.12.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_KPIS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.12.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.12.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.12.2.1 Columns

The TMSCHEMA_KPIS rowset contains the following columns.

Name	Restriction
ID	Yes
MeasureID	Yes
Description	Yes
TargetDescription	Yes
TargetExpression	Yes
TargetFormatString	Yes
StatusGraphic	Yes
StatusDescription	Yes
StatusExpression	Yes
TrendGraphic	Yes
TrendDescription	Yes
TrendExpression	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to KPI. The XSD for the TMSCHEMA_KPIS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
```

```
        <xs:element name="row" type="TabularDiscoverKPIRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverKPIRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="MeasureID" name="MeasureID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="TargetDescription" name="TargetDescription" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="TargetExpression" name="TargetExpression" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="TargetFormatString" name="TargetFormatString" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="StatusGraphic" name="StatusGraphic" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="StatusDescription" name="StatusDescription" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="StatusExpression" name="StatusExpression" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="TrendGraphic" name="TrendGraphic" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="TrendDescription" name="TrendDescription" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="TrendExpression" name="TrendExpression" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.12.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_KPIS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.13 TMSCHEMA_CULTURES

The TMSCHEMA_CULTURES schema rowset provides information about the **Culture** objects in the model.

3.1.5.1.1.13.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_CULTURES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.13.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.13.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.13.2.1 Columns

The TMSCHEMA_CULTURES rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
LinguisticMetadataID	Yes
ModifiedTime	Yes
StructureModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Culture. The XSD for the TMSCHEMA_CULTURES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverCultureRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverCultureRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="LinguisticMetadataID" name="LinguisticMetadataID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xs:dateTime" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.13.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_CULTURES rowset.

- DatabaseName
- ModifiedTimeOp
- StructureModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.14 TMSCHEMA_OBJECT_TRANSLATIONS

The TMSCHEMA_OBJECT_TRANSLATIONS schema rowset provides information about the translations of different objects for a culture. The object being translated is identified by the **ObjectType**, the **ObjectID**, and the **Property**.

3.1.5.1.1.14.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_OBJECT_TRANSLATIONS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.14.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.14.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.14.2.1 Columns

The TMSCHEMA_OBJECT_TRANSLATIONS rowset contains the following columns.

Name	Restriction
ID	Yes
CultureID	Yes
ObjectID	Yes
ObjectType	Yes
Property	Yes
Value	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to ObjectTranslation. The XSD for the TMSCHEMA_OBJECT_TRANSLATIONS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverObjectTranslationRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverObjectTranslationRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="CultureID" name="CultureID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ObjectID" name="ObjectID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ObjectType" name="ObjectType" type="xs:int" minOccurs="0" />
      <xs:element sql:field="Property" name="Property" type="xs:long" minOccurs="0" />
      <xs:element sql:field="Value" name="Value" type="xs:string" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
```

```
</xs:schema>
```

3.1.5.1.1.14.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_OBJECT_TRANSLATIONS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.15 TMSCHEMA_LINGUISTIC_METADATA

The TMSCHEMA_LINGUISTIC_METADATA schema rowset provides information about the synonyms for objects in the model for a particular culture.

3.1.5.1.1.15.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_LINGUISTIC_METADATA. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.15.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.15.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.15.2.1 Columns

The TMSCHEMA_LINGUISTIC_METADATA rowset contains the following columns.

Name	Restriction
ID	Yes
CultureID	Yes
Content	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to LinguisticMetadata. The XSD for the TMSCHEMA_LINGUISTIC_METADATA rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverLinguisticMetadataRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverLinguisticMetadataRowType">
    <xs:sequence>
```

```
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="CultureID" name="CultureID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="Content" name="Content" type="xmlDocument" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.15.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_LINGUISTIC_METADATA rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.16 TMSCHEMA_PERSPECTIVES

The TMSCHEMA_PERSPECTIVES schema rowset provides information about the **Perspective** objects in the model.

3.1.5.1.1.16.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PERSPECTIVES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.16.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.16.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.16.2.1 Columns

The TMSCHEMA_PERSPECTIVES rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
Description	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Perspective. The XSD for the TMSCHEMA_PERSPECTIVES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPerspectiveRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPerspectiveRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.16.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PERSPECTIVES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.17 TMSCHEMA_PERSPECTIVE_TABLES

The TMSCHEMA_PERSPECTIVE_TABLES schema rowset provides information about the **Table** objects in a perspective.

3.1.5.1.1.17.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PERSPECTIVE_TABLES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.17.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.17.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.17.2.1 Columns

The TMSCHEMA_PERSPECTIVE_TABLES rowset contains the following columns.

Name	Restriction
ID	Yes
PerspectiveID	Yes
TableID	Yes

Name	Restriction
IncludeAll	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to PerspectiveTable. The XSD for the TMSCHEMA_PERSPECTIVE_TABLES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPerspectiveTableRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPerspectiveTableRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="PerspectiveID" name="PerspectiveID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="IncludeAll" name="IncludeAll" type="xs:boolean" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.17.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PERSPECTIVE_TABLES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.18 TMSCHEMA_PERSPECTIVE_COLUMNS

The TMSCHEMA_PERSPECTIVE_COLUMNS schema rowset provides information about the **PerspectiveColumn** objects in each **PerspectiveTable** object.

3.1.5.1.1.18.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PERSPECTIVE_COLUMNS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.18.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.18.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.18.2.1 Columns

The TMSCHEMA_PERSPECTIVE_COLUMNS rowset contains the following columns.

Name	Restriction
ID	Yes
PerspectiveTableID	Yes
ColumnID	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to PerspectiveColumn. The XSD for the TMSCHEMA_PERSPECTIVE_COLUMNS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPerspectiveColumnRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPerspectiveColumnRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="PerspectiveTableID" name="PerspectiveTableID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ColumnID" name="ColumnID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.18.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PERSPECTIVE_COLUMNS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.19 TMSCHEMA_PERSPECTIVE_HIERARCHIES

The TMSCHEMA_PERSPECTIVE_HIERARCHIES schema rowset provides information about the **PerspectiveHierarchy** objects in each **PerspectiveTable** object.

3.1.5.1.1.19.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PERSPECTIVE_HIERARCHIES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.19.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.19.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.19.2.1 Columns

The TMSCHEMA_PERSPECTIVE_HIERARCHIES rowset contains the following columns.

Name	Restriction
ID	Yes
PerspectiveTableID	Yes
HierarchyID	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to PerspectiveHierarchy. The XSD for the TMSCHEMA_PERSPECTIVE_HIERARCHIES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPerspectiveHierarchyRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPerspectiveHierarchyRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="PerspectiveTableID" name="PerspectiveTableID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="HierarchyID" name="HierarchyID" type="xs:unsignedLong"
minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.19.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PERSPECTIVE_HIERARCHIES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.20 TMSCHEMA_PERSPECTIVE_MEASURES

The TMSCHEMA_PERSPECTIVE_MEASURES schema rowset provides information about the **PerspectiveMeasure** objects in each **PerspectiveTable** object.

3.1.5.1.1.20.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_PERSPECTIVE_MEASURES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.20.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.20.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.20.2.1 Columns

The TMSCHEMA_PERSPECTIVE_MEASURES rowset contains the following columns.

Name	Restriction
ID	Yes
PerspectiveTableID	Yes
MeasureID	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to PerspectiveMeasure. The XSD for the TMSCHEMA_PERSPECTIVE_MEASURES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverPerspectiveMeasureRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverPerspectiveMeasureRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="PerspectiveTableID" name="PerspectiveTableID"
type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="MeasureID" name="MeasureID" type="xs:unsignedLong" minOccurs="0"
/>
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.20.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_PERSPECTIVE_MEASURES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.21 TMSCHEMA_ROLES

The TMSCHEMA_ROLES schema rowset provides information about the **Role** objects in the model.

3.1.5.1.1.21.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_ROLES. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.21.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.21.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.21.2.1 Columns

The TMSCHEMA_ROLES rowset contains the following columns.

Name	Restriction
ID	Yes
ModelID	Yes
Name	Yes
Description	Yes
ModelPermission	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to Role. The XSD for the TMSCHEMA_ROLES rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverRoleRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverRoleRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="ModelID" name="ModelID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="Name" name="Name" type="xs:string" minOccurs="0" />
      <xs:element sql:field="Description" name="Description" type="xs:string" minOccurs="0"
/>
      <xs:element sql:field="ModelPermission" name="ModelPermission" type="xs:long"
minOccurs="0" />
```

```
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.21.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_ROLES rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.22 TMSCHEMA_ROLE_MEMBERSHIPS

The TMSCHEMA_ROLE_MEMBERSHIPS schema rowset provides information about the **RoleMembership** objects in each role.

3.1.5.1.1.22.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_ROLE_MEMBERSHIPS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.22.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.22.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.22.2.1 Columns

The TMSCHEMA_ROLE_MEMBERSHIPS rowset contains the following columns.

Name	Restriction
ID	Yes
RoleID	Yes
MemberName	Yes
MemberID	Yes
IdentityProvider	Yes
MemberType	Yes
ModifiedTime	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to RoleMembership. The XSD for the TMSCHEMA_ROLE_MEMBERSHIPS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverRoleMembershipRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverRoleMembershipRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="RoleID" name="RoleID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="MemberName" name="MemberName" type="xs:string" minOccurs="0" />
      <xs:element sql:field="MemberID" name="MemberID" type="xs:string" minOccurs="0" />
      <xs:element sql:field="IdentityProvider" name="IdentityProvider" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="MemberType" name="MemberType" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.22.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_ROLE_MEMBERSHIPS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.1.1.23 TMSCHEMA_TABLE_PERMISSIONS

The TMSCHEMA_TABLE_PERMISSIONS schema rowset provides information about the **TablePermission** objects in each role.

3.1.5.1.1.23.1 Request Body

The **RequestType** element of the DiscoverSoapIn message is TMSCHEMA_TABLE_PERMISSIONS. For the definition of the **RequestType** element, see [MS-SSAS] section 3.1.4.2.2.1.

3.1.5.1.1.23.2 Response Body

The rowset in the **DiscoverResponse** element of the DiscoverSoapOut message contains the columns specified in section 3.1.5.1.1.23.2.1. For the definition of the **DiscoverResponse** element, see [MS-SSAS] section 3.1.4.2.2.2.

3.1.5.1.1.23.2.1 Columns

The TMSCHEMA_TABLE_PERMISSIONS rowset contains the following columns.

Name	Restriction
ID	Yes
RoleID	Yes

Name	Restriction
TableID	Yes
FilterExpression	Yes
ModifiedTime	Yes
State	Yes
ErrorMessage	Yes

The name attribute in the **TabularDiscoverRowsetType** is set to TablePermission. The XSD for the TMSCHEMA_TABLE_PERMISSIONS rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" type="TabularDiscoverTablePermissionRowType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TabularDiscoverTablePermissionRowType">
    <xs:sequence>
      <xs:element sql:field="ID" name="ID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="RoleID" name="RoleID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="TableID" name="TableID" type="xs:unsignedLong" minOccurs="0" />
      <xs:element sql:field="FilterExpression" name="FilterExpression" type="xs:string"
minOccurs="0" />
      <xs:element sql:field="ModifiedTime" name="ModifiedTime" type="xs:dateTime"
minOccurs="0" />
      <xs:element sql:field="State" name="State" type="xs:long" minOccurs="0" />
      <xs:element sql:field="ErrorMessage" name="ErrorMessage" type="xs:string" minOccurs="0"
/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

3.1.5.1.1.23.2.2 Additional Restrictions

In addition to the column restrictions that are indicated in the preceding section, the following restrictions can apply to the TMSCHEMA_TABLE_PERMISSIONS rowset.

- DatabaseName
- ModifiedTimeOp

For a description of these restrictions, see section 2.2.5.24.

3.1.5.2 Execute

The **Execute** operation is used to execute commands on the server. For the messaging protocol for **Execute** operations, see [MS-SSAS] section 3.1.4.3.

This section defines the types of commands that can be executed to support operations that manipulate the Tabular Metadata.

This document defines the following two types of extension commands for the Tabular Metadata.

- XMLA-based commands, which extend the commands in [MS-SSAS] section 3.1.4.3.2.1.1.
- JSON-based commands, which are strings that are specified under the **Statement** element as defined in [MS-SSAS] section 3.1.4.3.2.1.1.2.

Many of the commands are available in both types. For example, a Table can be created, altered, or deleted by using either the XMLA-based format or the JSON-based format.

In this document, the XMLA-based commands are discussed in sections 3.1.5.2.1 through 3.1.5.2.1.9.2, and the JSON-based commands are discussed in sections 3.1.5.2.2 through 3.1.5.2.2.13.2.

3.1.5.2.1 XMLA-Based Tabular Metadata Commands

A server in Tabular mode can support databases that have the compatibility level set to 1200 and StorageEngineUsed set to TabularMetadata (see [MS-SSAS] section 2.2.4.2.2.5). The commands to manipulate these databases are documented in sections 3.1.5.2.1.1 through 3.1.5.2.1.9.2.

The request and response messages for these commands conform to the protocol of an **Execute** operation as defined in [MS-SSAS] section 3.1.4.3.1.

In particular, the XMLA **Command** element, defined in [MS-SSAS] section 3.1.4.3.2.1.1, is extended to allow the following Tabular Metadata commands.

```
<xsd:complexType name="Command">
  <xsd:choice>
    <xsd:element name="Create" type="mstns:TabularCommandType" minOccurs="0" />
    <xsd:element name="Alter" type="mstns:TabularCommandType" minOccurs="0" />
    <xsd:element name="Delete" type="mstns:TabularCommandType" minOccurs="0" />
    <xsd:element name="Rename" type="mstns:TabularCommandType" minOccurs="0" />
    <xsd:element name="Refresh" type="mstns:TabularRefreshCommandType" minOccurs="0" />
    <xsd:element name="MergePartitions" type="mstns: MergePartitionsTabular" minOccurs="0" />
    <xsd:element name="DBCC" type="mstns:DBCCTabular" minOccurs="0" />
    <xsd:element name="SequencePoint" type="mstns:SequencePointTabular" minOccurs="0" />
    <xsd:element name="Upgrade" type="mstns: UpgradeTabular" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

These command elements are documented in the Request sections below.

Each of the commands described in the following sections can use an object of type **TabularCommandType**, which contains objects of type xmla-rs:rowset, as described in [MS-SSAS] section 2.2.4.1.3.

The XSD for the **TabularCommandType** complex type is as follows.

```
<xs:complexType name="TabularCommandType">
  <xs:sequence>
    <xs:element name="DatabaseID" type="xs:string" />
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element name="Model" type="xmla-rs:rowset" />
        <xs:element name="DataSources" type="xmla-rs:rowset" />
        <xs:element name="Tables" type="xmla-rs:rowset" />
        <xs:element name="Columns" type="xmla-rs:rowset" />
        <xs:element name="Partitions" type="xmla-rs:rowset" />
        <xs:element name="Relationships" type="xmla-rs:rowset" />
        <xs:element name="Measures" type="xmla-rs:rowset" />
        <xs:element name="Hierarchies" type="xmla-rs:rowset" />
        <xs:element name="Levels" type="xmla-rs:rowset" />
        <xs:element name="Annotations" type="xmla-rs:rowset" />
        <xs:element name="Kpis" type="xmla-rs:rowset" />
```

```
          <xs:element name="Cultures" type="xmla-rs:rowset" />
          <xs:element name="ObjectTranslations" type="xmla-rs:rowset" />
          <xs:element name="LinguisticMetadata" type="xmla-rs:rowset" />
          <xs:element name="Perspectives" type="xmla-rs:rowset" />
          <xs:element name="PerspectiveTables" type="xmla-rs:rowset" />
          <xs:element name="PerspectiveColumns" type="xmla-rs:rowset" />
          <xs:element name="PerspectiveHierarchies" type="xmla-rs:rowset" />
          <xs:element name="PerspectiveMeasures" type="xmla-rs:rowset" />
          <xs:element name="Roles" type="xmla-rs:rowset" />
          <xs:element name="RoleMemberships" type="xmla-rs:rowset" />
          <xs:element name="TablePermissions" type="xmla-rs:rowset" />
        </xs:choice>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
```

The element name of each rowset identifies which type of object is represented by the rowset, as illustrated in the following example.

```
<Command>
  <Create xmlns="http://schemas.microsoft.com/analysisservices/2014/engine">
    <DatabaseID>Adventure Works</DatabaseID>
    <DataSources>
      <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:sql="urn:schemas-microsoft-com:xml-sql">
        <xs:element>
          <xs:complexType>
            <xs:sequence>
              <xs:element type="row" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:complexType name="row">
          <xs:sequence>
            <xs:element name="Name" type="xs:string" sql:field="Name"
                        minOccurs="0" />
            <xs:element name="Description" type="xs:string"
                        sql:field="Description" minOccurs="0" />
            <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
            <xs:element name="ConnectionString" type="xs:string"
                        sql:field="ConnectionString" minOccurs="0" />
            <xs:element name="ImpersonationMode" type="xs:long"
                        sql:field="ImpersonationMode" minOccurs="0" />
            <xs:element name="Account" type="xs:string"
                        sql:field="Account" minOccurs="0" />
            <xs:element name="Password" type="xs:string"
                        sql:field="Password" minOccurs="0" />
            <xs:element name="MaxConnections" type="xs:int"
                        sql:field="MaxConnections" minOccurs="0" />
            <xs:element name="Isolation" type="xs:long"
                        sql:field="Isolation" minOccurs="0" />
            <xs:element name="Timeout" type="xs:int"
                        sql:field="Timeout" minOccurs="0" />
            <xs:element name="Provider" type="xs:string"
                        sql:field="Provider" minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
      <row xmlns="urn:schemas-microsoft-com:xml-analysis:rowset">
        <Name>SqlServer sqlcldb2 AS_foodmart_2000</Name>
        <ConnectionString>Provider=SQLNCLI11;Data Source=...</ConnectionString>
        <ImpersonationMode>5</ImpersonationMode>
      </row>
    </DataSources>
    <Tables>
      <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                 xmlns:sql="urn:schemas-microsoft-com:xml-sql">
```

```
          <xs:element>
            <xs:complexType>
              <xs:sequence>
                <xs:element type="row" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:complexType name="row">
            <xs:sequence>
              <xs:element name="Name" type="xs:string"
                          sql:field="Name" minOccurs="0" />
              <xs:element name="DataCategory" type="xs:string"
                          sql:field="DataCategory" minOccurs="0" />
              <xs:element name="Description" type="xs:string"
                          sql:field="Description" minOccurs="0" />
              <xs:element name="IsHidden" type="xs:boolean"
                          sql:field="IsHidden" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:schema>
        <row xmlns="urn:schemas-microsoft-com:xml-analysis:rowset">
          <Name>Customer</Name>
          <Description>Customer information.</Description>
        </row>
      </Tables>
    </Create>
  </Command>
```

Each of these rowset objects follows the standard XMLA rowset format, as described in [MS-SSAS] section 2.2.4.1.3. Each object begins with an XSD followed by zero or more row objects. Each row in the rowset contains the properties of a new object that is to be created in the database. When a property is not specified, its default value will be used.

Note on Object References

In the following XMLA commands, Tabular Metadata objects can be identified in two ways: by integer object ID (such as **TableID**) and by name-based path (such as the equivalent **TableID.Table**). Each of the commands allows either form, but only one of them is expected to be present.

In general, it is preferable to use the integer ID if it is available. Otherwise, the name-based path can be used, and the integer ID will be derived from the name-based path. If it is necessary to use both, they need to refer to the same object, or conflicts might occur. In principle, specifying the name-based path is redundant if the integer ID is already specified.

3.1.5.2.1.1 Create Tabular Metadata

The **Create Tabular Metadata** command is used to create objects in a Tabular database that has the compatibility level set to 1200. The command requires a **DatabaseID** child element that identifies the database in which the Tabular metadata objects are to be created, followed by a set of rowsets that define the new objects that are to be created.

3.1.5.2.1.1.1 Request

The object types allowed are defined in the **TabularCommandType** object in section 3.1.5.2.1, and the schema of the rowsets for these object types is documented in the following subsections.

Creation of objects performs some basic validation. For example, references to parent objects, such as the table to which a **Column** object belongs, are validated during execution of the Create Tabular Metadata API. Other validations, such as syntax and semantic validation of DAX expressions, can be deferred until a later operation.

3.1.5.2.1.1.1.1 Create Model

The **Create Model** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="StorageLocation" type="xs:string" sql:field="StorageLocation"
minOccurs="0" />
          <xs:element name="Mode" type="xs:long" sql:field="Mode" minOccurs="0" />
          <xs:element name="Culture" type="xs:string" sql:field="Culture" minOccurs="0" />
          <xs:element name="Collation" type="xs:string" sql:field="Collation" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
Description	
StorageLocation	
Mode	
Culture	
Collation	

The properties correspond to the **Model** object defined in section 2.2.5.1.

3.1.5.2.1.1.1.2 Create DataSources

The **Create DataSources** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
```

```
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
          <xs:element name="ConnectionString" type="xs:string" sql:field="ConnectionString"
minOccurs="0" />
          <xs:element name="ImpersonationMode" type="xs:long" sql:field="ImpersonationMode"
minOccurs="0" />
          <xs:element name="Account" type="xs:string" sql:field="Account" minOccurs="0" />
          <xs:element name="Password" type="xs:string" sql:field="Password" minOccurs="0" />
          <xs:element name="MaxConnections" type="xs:int" sql:field="MaxConnections"
minOccurs="0" />
          <xs:element name="Isolation" type="xs:long" sql:field="Isolation" minOccurs="0" />
          <xs:element name="Timeout" type="xs:int" sql:field="Timeout" minOccurs="0" />
          <xs:element name="Provider" type="xs:string" sql:field="Provider" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
Description	
Type	"Provider"
ConnectionString	
ImpersonationMode	
Account	
Password	
MaxConnections	
Isolation	"ReadCommitted"
Timeout	
Provider	

The properties correspond to the **DataSource** object defined in section 2.2.5.2.

3.1.5.2.1.1.1.3 Create Tables

The **Create Tables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
```

```
          <xs:element name="DataCategory" type="xs:string" sql:field="DataCategory"
minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
DataCategory	
Description	
IsHidden	

The properties correspond to the **Table** object defined in section 2.2.5.3.

3.1.5.2.1.1.1.4 Create Columns

The **Create Columns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="ExplicitName" type="xs:string" sql:field="ExplicitName"
minOccurs="0" />
          <xs:element name="ExplicitDataType" type="xs:long" sql:field="ExplicitDataType"
minOccurs="0" />
          <xs:element name="DataCategory" type="xs:string" sql:field="DataCategory"
minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
          <xs:element name="IsUnique" type="xs:boolean" sql:field="IsUnique" minOccurs="0" />
          <xs:element name="IsKey" type="xs:boolean" sql:field="IsKey" minOccurs="0" />
          <xs:element name="IsNullable" type="xs:boolean" sql:field="IsNullable"
minOccurs="0" />
          <xs:element name="Alignment" type="xs:long" sql:field="Alignment" minOccurs="0" />
          <xs:element name="TableDetailPosition" type="xs:int"
sql:field="TableDetailPosition" minOccurs="0" />
          <xs:element name="IsDefaultLabel" type="xs:boolean" sql:field="IsDefaultLabel"
minOccurs="0" />
          <xs:element name="IsDefaultImage" type="xs:boolean" sql:field="IsDefaultImage"
minOccurs="0" />
          <xs:element name="SummarizeBy" type="xs:long" sql:field="SummarizeBy" minOccurs="0"
/>
```

```
          <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
          <xs:element name="SourceColumn" type="xs:string" sql:field="SourceColumn"
minOccurs="0" />
          <xs:element name="Expression" type="xs:string" sql:field="Expression" minOccurs="0"
/>
          <xs:element name="FormatString" type="xs:string" sql:field="FormatString"
minOccurs="0" />
          <xs:element name="IsAvailableInMDX" type="xs:boolean" sql:field="IsAvailableInMDX"
minOccurs="0" />
          <xs:element name="SortByColumnID" type="xs:unsignedLong" sql:field="SortByColumnID"
minOccurs="0" />
          <xs:element name="SortByColumnID.Table" type="xs:string"
sql:field="SortByColumnID.Table" minOccurs="0" />
          <xs:element name="SortByColumnID.Column" type="xs:string"
sql:field="SortByColumnID.Column" minOccurs="0" />
          <xs:element name="KeepUniqueRows" type="xs:boolean" sql:field="KeepUniqueRows"
minOccurs="0" />
          <xs:element name="DisplayOrdinal" type="xs:int" sql:field="DisplayOrdinal"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
TableID	
TableID.Table	
ExplicitName	
ExplicitDataType	
DataCategory	
Description	
IsHidden	
IsUnique	
IsKey	
IsNullable	
Alignment	"Default"
TableDetailPosition	
IsDefaultLabel	
IsDefaultImage	
SummarizeBy	"Default"
Type	"Data"
SourceColumn	
Expression	
FormatString	

Element	Default value
IsAvailableInMDX	
SortByColumnID	
SortByColumnID.Table	
SortByColumnID.Column	
KeepUniqueRows	
DisplayOrdinal	
DisplayFolder	

The properties correspond to the **Column** object defined in section 2.2.5.4.

3.1.5.2.1.1.1.5 Create Partitions

The **Create Partitions** schema definition is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element type="row" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="row">
    <xs:sequence>
      <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0" />
      <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
      <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
      <xs:element name="Description" type="xs:string" sql:field="Description" minOccurs="0"
/>
      <xs:element name="DataSourceID" type="xs:unsignedLong" sql:field="DataSourceID"
minOccurs="0" />
      <xs:element name="DataSourceID.DataSource" type="xs:string"
sql:field="DataSourceID.DataSource" minOccurs="0" />
      <xs:element name="QueryDefinition" type="xs:string" sql:field="QueryDefinition"
minOccurs="0" />
      <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
      <xs:element name="Mode" type="xs:long" sql:field="Mode" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Element	Default value
TableID	
TableID.Table	
Name	

Element	Default value
Description	
DataSourceID	
DataSourceID.DataSource	
QueryDefinition	
Type	"Query"
Mode	

The properties correspond to the **Partition** object defined in section 2.2.5.6.

3.1.5.2.1.1.1.6 Create Relationships

The **Create Relationships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="IsActive" type="xs:boolean" sql:field="IsActive" minOccurs="0" />
          <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
          <xs:element name="CrossFilteringBehavior" type="xs:long"
sql:field="CrossFilteringBehavior" minOccurs="0" />
          <xs:element name="JoinOnDateBehavior" type="xs:long" sql:field="JoinOnDateBehavior"
minOccurs="0" />
          <xs:element name="RelyOnReferentialIntegrity" type="xs:boolean"
sql:field="RelyOnReferentialIntegrity" minOccurs="0" />
          <xs:element name="FromTableID" type="xs:unsignedLong" sql:field="FromTableID"
minOccurs="0" />
          <xs:element name="FromTableID.Table" type="xs:string" sql:field="FromTableID.Table"
minOccurs="0" />
          <xs:element name="FromColumnID" type="xs:unsignedLong" sql:field="FromColumnID"
minOccurs="0" />
          <xs:element name="FromColumnID.Table" type="xs:string"
sql:field="FromColumnID.Table" minOccurs="0" />
          <xs:element name="FromColumnID.Column" type="xs:string"
sql:field="FromColumnID.Column" minOccurs="0" />
          <xs:element name="FromCardinality" type="xs:long" sql:field="FromCardinality"
minOccurs="0" />
          <xs:element name="ToTableID" type="xs:unsignedLong" sql:field="ToTableID"
minOccurs="0" />
          <xs:element name="ToTableID.Table" type="xs:string" sql:field="ToTableID.Table"
minOccurs="0" />
          <xs:element name="ToColumnID" type="xs:unsignedLong" sql:field="ToColumnID"
minOccurs="0" />
          <xs:element name="ToColumnID.Table" type="xs:string" sql:field="ToColumnID.Table"
minOccurs="0" />
          <xs:element name="ToColumnID.Column" type="xs:string" sql:field="ToColumnID.Column"
minOccurs="0" />
          <xs:element name="ToCardinality" type="xs:long" sql:field="ToCardinality"
minOccurs="0" />
```

```
          <xs:element name="SecurityFilteringBehavior" type="xs:long"
sql:field="SecurityFilteringBehavior" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
IsActive	
Type	
CrossFilteringBehavior	"OneDirection"
JoinOnDateBehavior	
RelyOnReferentialIntegrity	
FromTableID	
FromTableID.Table	
FromColumnID	
FromColumnID.Table	
FromColumnID.Column	
FromCardinality	
ToTableID	
ToTableID.Table	
ToColumnID	
ToColumnID.Table	
ToColumnID.Column	
ToCardinality	
SecurityFilteringBehavior	"OneDirection"

The properties correspond to the **Relationship** object defined in section 2.2.5.7.

3.1.5.2.1.1.1.7 Create Measures

The **Create Measures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
```

```
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="Expression" type="xs:string" sql:field="Expression" minOccurs="0"
/>
          <xs:element name="FormatString" type="xs:string" sql:field="FormatString"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
          <xs:element name="IsSimpleMeasure" type="xs:boolean" sql:field="IsSimpleMeasure"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
TableID	
TableID.Table	
Name	
Description	
Expression	
FormatString	
IsHidden	
IsSimpleMeasure	
DisplayFolder	

The properties correspond to the **Measure** object defined in section 2.2.5.8.

3.1.5.2.1.1.8 Create Hierarchies

The **Create Hierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
```

```
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
TableID	
TableID.Table	
Name	
Description	
IsHidden	
DisplayFolder	

The properties correspond to the **Hierarchy** object defined in section 2.2.5.9.

3.1.5.2.1.1.1.9 Create Levels

The **Create Levels** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="HierarchyID" type="xs:unsignedLong" sql:field="HierarchyID"
minOccurs="0" />
          <xs:element name="HierarchyID.Table" type="xs:string" sql:field="HierarchyID.Table"
minOccurs="0" />
          <xs:element name="HierarchyID.Hierarchy" type="xs:string"
sql:field="HierarchyID.Hierarchy" minOccurs="0" />
          <xs:element name="Ordinal" type="xs:int" sql:field="Ordinal" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="ColumnID" type="xs:unsignedLong" sql:field="ColumnID"
minOccurs="0" />
          <xs:element name="ColumnID.Table" type="xs:string" sql:field="ColumnID.Table"
minOccurs="0" />
          <xs:element name="ColumnID.Column" type="xs:string" sql:field="ColumnID.Column"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
```

```
</xs:schema>
```

Element	Default value
HierarchyID	
HierarchyID.Table	
HierarchyID.Hierarchy	
Ordinal	
Name	
Description	
ColumnID	
ColumnID.Table	
ColumnID.Column	

The properties correspond to the **Level** object defined in section 2.2.5.10.

3.1.5.2.1.1.1.10 Create Annotations

The **Create Annotations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ObjectID" type="xs:unsignedLong" sql:field="ObjectID"
minOccurs="0" />
          <xs:element name="ObjectID.DataSource" type="xs:string"
sql:field="ObjectID.DataSource" minOccurs="0" />
          <xs:element name="ObjectID.Table" type="xs:string" sql:field="ObjectID.Table"
minOccurs="0" />
          <xs:element name="ObjectID.Column" type="xs:string" sql:field="ObjectID.Column"
minOccurs="0" />
          <xs:element name="ObjectID.Partition" type="xs:string"
sql:field="ObjectID.Partition" minOccurs="0" />
          <xs:element name="ObjectID.Relationship" type="xs:string"
sql:field="ObjectID.Relationship" minOccurs="0" />
          <xs:element name="ObjectID.Measure" type="xs:string" sql:field="ObjectID.Measure"
minOccurs="0" />
          <xs:element name="ObjectID.Hierarchy" type="xs:string"
sql:field="ObjectID.Hierarchy" minOccurs="0" />
          <xs:element name="ObjectID.Level" type="xs:string" sql:field="ObjectID.Level"
minOccurs="0" />
          <xs:element name="ObjectID.Culture" type="xs:string" sql:field="ObjectID.Culture"
minOccurs="0" />
          <xs:element name="ObjectID.Perspective" type="xs:string"
sql:field="ObjectID.Perspective" minOccurs="0" />
```

```
          <xs:element name="ObjectID.PerspectiveTable" type="xs:string"
sql:field="ObjectID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveColumn" type="xs:string"
sql:field="ObjectID.PerspectiveColumn" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveHierarchy" type="xs:string"
sql:field="ObjectID.PerspectiveHierarchy" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveMeasure" type="xs:string"
sql:field="ObjectID.PerspectiveMeasure" minOccurs="0" />
          <xs:element name="ObjectID.Role" type="xs:string" sql:field="ObjectID.Role"
minOccurs="0" />
          <xs:element name="ObjectID.RoleMembership" type="xs:string"
sql:field="ObjectID.RoleMembership" minOccurs="0" />
          <xs:element name="ObjectID.TablePermission" type="xs:string"
sql:field="ObjectID.TablePermission" minOccurs="0" />
          <xs:element name="ObjectType" type="xs:int" sql:field="ObjectType" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Value" type="xs:string" sql:field="Value" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
ObjectID	
ObjectID.DataSource	
ObjectID.Table	
ObjectID.Column	
ObjectID.Partition	
ObjectID.Relationship	
ObjectID.Measure	
ObjectID.Hierarchy	
ObjectID.Level	
ObjectID.Culture	
ObjectID.Perspective	
ObjectID.PerspectiveTable	
ObjectID.PerspectiveColumn	
ObjectID.PerspectiveHierarchy	
ObjectID.PerspectiveMeasure	
ObjectID.Role	
ObjectID.RoleMembership	
ObjectID.TablePermission	
ObjectType	
Name	
Value	

The properties correspond to the **Annotation** object defined in section 2.2.5.11.

3.1.5.2.1.1.1.11 Create Kpis

The **Create Kpis** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="MeasureID" type="xs:unsignedLong" sql:field="MeasureID"
minOccurs="0" />
          <xs:element name="MeasureID.Table" type="xs:string" sql:field="MeasureID.Table"
minOccurs="0" />
          <xs:element name="MeasureID.Measure" type="xs:string" sql:field="MeasureID.Measure"
minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="TargetDescription" type="xs:string" sql:field="TargetDescription"
minOccurs="0" />
          <xs:element name="TargetExpression" type="xs:string" sql:field="TargetExpression"
minOccurs="0" />
          <xs:element name="TargetFormatString" type="xs:string"
sql:field="TargetFormatString" minOccurs="0" />
          <xs:element name="StatusGraphic" type="xs:string" sql:field="StatusGraphic"
minOccurs="0" />
          <xs:element name="StatusDescription" type="xs:string" sql:field="StatusDescription"
minOccurs="0" />
          <xs:element name="StatusExpression" type="xs:string" sql:field="StatusExpression"
minOccurs="0" />
          <xs:element name="TrendGraphic" type="xs:string" sql:field="TrendGraphic"
minOccurs="0" />
          <xs:element name="TrendDescription" type="xs:string" sql:field="TrendDescription"
minOccurs="0" />
          <xs:element name="TrendExpression" type="xs:string" sql:field="TrendExpression"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
MeasureID	
MeasureID.Table	
MeasureID.Measure	
Description	
TargetDescription	
TargetExpression	Empty
TargetFormatString	

Element	Default value
StatusGraphic	
StatusDescription	
StatusExpression	
TrendGraphic	
TrendDescription	
TrendExpression	

The properties correspond to the **KPI** object defined in section 2.2.5.12.

3.1.5.2.1.1.1.12 Create Cultures

The **Create Cultures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	

The properties correspond to the **Culture** object defined in section 2.2.5.13.

3.1.5.2.1.1.1.13 Create ObjectTranslations

The **Create ObjectTranslations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
```

```
          <xs:element name="CultureID" type="xs:unsignedLong" sql:field="CultureID"
minOccurs="0" />
          <xs:element name="CultureID.Culture" type="xs:string" sql:field="CultureID.Culture"
minOccurs="0" />
          <xs:element name="ObjectID" type="xs:unsignedLong" sql:field="ObjectID"
minOccurs="0" />
          <xs:element name="ObjectID.DataSource" type="xs:string"
sql:field="ObjectID.DataSource" minOccurs="0" />
          <xs:element name="ObjectID.Table" type="xs:string" sql:field="ObjectID.Table"
minOccurs="0" />
          <xs:element name="ObjectID.Column" type="xs:string" sql:field="ObjectID.Column"
minOccurs="0" />
          <xs:element name="ObjectID.Partition" type="xs:string"
sql:field="ObjectID.Partition" minOccurs="0" />
          <xs:element name="ObjectID.Relationship" type="xs:string"
sql:field="ObjectID.Relationship" minOccurs="0" />
          <xs:element name="ObjectID.Measure" type="xs:string" sql:field="ObjectID.Measure"
minOccurs="0" />
          <xs:element name="ObjectID.Hierarchy" type="xs:string"
sql:field="ObjectID.Hierarchy" minOccurs="0" />
          <xs:element name="ObjectID.Level" type="xs:string" sql:field="ObjectID.Level"
minOccurs="0" />
          <xs:element name="ObjectID.Culture" type="xs:string" sql:field="ObjectID.Culture"
minOccurs="0" />
          <xs:element name="ObjectID.Perspective" type="xs:string"
sql:field="ObjectID.Perspective" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveTable" type="xs:string"
sql:field="ObjectID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveColumn" type="xs:string"
sql:field="ObjectID.PerspectiveColumn" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveHierarchy" type="xs:string"
sql:field="ObjectID.PerspectiveHierarchy" minOccurs="0" />
          <xs:element name="ObjectID.PerspectiveMeasure" type="xs:string"
sql:field="ObjectID.PerspectiveMeasure" minOccurs="0" />
          <xs:element name="ObjectID.Role" type="xs:string" sql:field="ObjectID.Role"
minOccurs="0" />
          <xs:element name="ObjectID.RoleMembership" type="xs:string"
sql:field="ObjectID.RoleMembership" minOccurs="0" />
          <xs:element name="ObjectID.TablePermission" type="xs:string"
sql:field="ObjectID.TablePermission" minOccurs="0" />
          <xs:element name="ObjectType" type="xs:int" sql:field="ObjectType" minOccurs="0" />
          <xs:element name="Property" type="xs:long" sql:field="Property" minOccurs="0" />
          <xs:element name="Value" type="xs:string" sql:field="Value" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
CultureID	
CultureID.Culture	
ObjectID	
ObjectID.DataSource	
ObjectID.Table	
ObjectID.Column	
ObjectID.Partition	
ObjectID.Relationship	
ObjectID.Measure	

Element	Default value
ObjectID.Hierarchy	
ObjectID.Level	
ObjectID.Culture	
ObjectID.Perspective	
ObjectID.PerspectiveTable	
ObjectID.PerspectiveColumn	
ObjectID.PerspectiveHierarchy	
ObjectID.PerspectiveMeasure	
ObjectID.Role	
ObjectID.RoleMembership	
ObjectID.TablePermission	
ObjectType	
Property	"Invalid"
Value	

The properties correspond to the **ObjectTranslation** object defined in section 2.2.5.14.

3.1.5.2.1.1.1.14 Create LinguisticMetadata

The **Create LinguisticMetadata** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="CultureID" type="xs:unsignedLong" sql:field="CultureID"
minOccurs="0" />
          <xs:element name="CultureID.Culture" type="xs:string" sql:field="CultureID.Culture"
minOccurs="0" />
          <xs:element name="Content" type="xs:string" sql:field="Content" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
CultureID	

Element	Default value
CultureID.Culture	
Content	

The properties correspond to the **LinguisticMetadata** object defined in section 2.2.5.15.

3.1.5.2.1.1.1.15 Create Perspectives

The **Create Perspectives** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
Description	

The properties correspond to the **Perspective** object defined in section 2.2.5.16.

3.1.5.2.1.1.1.16 Create PerspectiveTables

The **Create PerspectiveTables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="PerspectiveID" type="xs:unsignedLong" sql:field="PerspectiveID"
minOccurs="0" />
          <xs:element name="PerspectiveID.Perspective" type="xs:string"
sql:field="PerspectiveID.Perspective" minOccurs="0" />
```

```
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="IncludeAll" type="xs:boolean" sql:field="IncludeAll"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
PerspectiveID	
PerspectiveID.Perspective	
TableID	
TableID.Table	
IncludeAll	false

The properties correspond to the **PerspectiveTable** object defined in section 2.2.5.17.

3.1.5.2.1.1.17 Create PerspectiveColumns

The **Create PerspectiveColumns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="PerspectiveTableID" type="xs:unsignedLong"
sql:field="PerspectiveTableID" minOccurs="0" />
          <xs:element name="PerspectiveTableID.Perspective" type="xs:string"
sql:field="PerspectiveTableID.Perspective" minOccurs="0" />
          <xs:element name="PerspectiveTableID.PerspectiveTable" type="xs:string"
sql:field="PerspectiveTableID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ColumnID" type="xs:unsignedLong" sql:field="ColumnID"
minOccurs="0" />
          <xs:element name="ColumnID.Table" type="xs:string" sql:field="ColumnID.Table"
minOccurs="0" />
          <xs:element name="ColumnID.Column" type="xs:string" sql:field="ColumnID.Column"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
PerspectiveTableID	

Element	Default value
PerspectiveTableID.Perspective	
PerspectiveTableID.PerspectiveTable	
ColumnID	
ColumnID.Table	
ColumnID.Column	

The properties correspond to the **PerspectiveColumn** object defined in section 2.2.5.18.

3.1.5.2.1.1.1.18 Create PerspectiveHierarchies

The **Create PerspectiveHierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="PerspectiveTableID" type="xs:unsignedLong"
sql:field="PerspectiveTableID" minOccurs="0" />
          <xs:element name="PerspectiveTableID.Perspective" type="xs:string"
sql:field="PerspectiveTableID.Perspective" minOccurs="0" />
          <xs:element name="PerspectiveTableID.PerspectiveTable" type="xs:string"
sql:field="PerspectiveTableID.PerspectiveTable" minOccurs="0" />
          <xs:element name="HierarchyID" type="xs:unsignedLong" sql:field="HierarchyID"
minOccurs="0" />
          <xs:element name="HierarchyID.Table" type="xs:string" sql:field="HierarchyID.Table"
minOccurs="0" />
          <xs:element name="HierarchyID.Hierarchy" type="xs:string"
sql:field="HierarchyID.Hierarchy" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
PerspectiveTableID	
PerspectiveTableID.Perspective	
PerspectiveTableID.PerspectiveTable	
HierarchyID	
HierarchyID.Table	
HierarchyID.Hierarchy	

The properties correspond to the **PerspectiveHierarchy** object defined in section 2.2.5.19.

3.1.5.2.1.1.1.19 Create PerspectiveMeasures

The **Create PerspectiveMeasures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="PerspectiveTableID" type="xs:unsignedLong"
sql:field="PerspectiveTableID" minOccurs="0" />
          <xs:element name="PerspectiveTableID.Perspective" type="xs:string"
sql:field="PerspectiveTableID.Perspective" minOccurs="0" />
          <xs:element name="PerspectiveTableID.PerspectiveTable" type="xs:string"
sql:field="PerspectiveTableID.PerspectiveTable" minOccurs="0" />
          <xs:element name="MeasureID" type="xs:unsignedLong" sql:field="MeasureID"
minOccurs="0" />
          <xs:element name="MeasureID.Table" type="xs:string" sql:field="MeasureID.Table"
minOccurs="0" />
          <xs:element name="MeasureID.Measure" type="xs:string" sql:field="MeasureID.Measure"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
PerspectiveTableID	
PerspectiveTableID.Perspective	
PerspectiveTableID.PerspectiveTable	
MeasureID	
MeasureID.Table	
MeasureID.Measure	

The properties correspond to the **PerspectiveMeasure** object defined in section 2.2.5.20.

3.1.5.2.1.1.1.20 Create Roles

The **Create Roles** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
```

```
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="ModelPermission" type="xs:long" sql:field="ModelPermission"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
Name	
Description	
ModelPermission	"None"

The properties correspond to the **Role** object defined in section 2.2.5.21.

3.1.5.2.1.1.1.21 Create RoleMemberships

The **Create RoleMemberships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="RoleID" type="xs:unsignedLong" sql:field="RoleID" minOccurs="0"
/>
          <xs:element name="RoleID.Role" type="xs:string" sql:field="RoleID.Role"
minOccurs="0" />
          <xs:element name="MemberName" type="xs:string" sql:field="MemberName" minOccurs="0"
/>
          <xs:element name="MemberID" type="xs:string" sql:field="MemberID" minOccurs="0" />
          <xs:element name="IdentityProvider" type="xs:string" sql:field="IdentityProvider"
minOccurs="0" />
          <xs:element name="MemberType" type="xs:long" sql:field="MemberType" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
RoleID	
RoleID.Role	
MemberName	

Element	Default value
MemberID	
IdentityProvider	
MemberType	"Auto"

The properties correspond to the **RoleMembership** object defined in section 2.2.5.22.

3.1.5.2.1.1.22 Create TablePermissions

The **Create TablePermissions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="RoleID" type="xs:unsignedLong" sql:field="RoleID" minOccurs="0"
/>
          <xs:element name="RoleID.Role" type="xs:string" sql:field="RoleID.Role"
minOccurs="0" />
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="FilterExpression" type="xs:string" sql:field="FilterExpression"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value
RoleID	
RoleID.Role	
TableID	
TableID.Table	
FilterExpression	

The properties correspond to the **TablePermission** object defined in section 2.2.5.23.

3.1.5.2.1.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.2 Alter Tabular Metadata

The **Alter Tabular Metadata** command is used to alter objects that already exist in a Tabular database that has the compatibility level set to 1200. The command requires a **DatabaseID** child element that identifies the database in which the Tabular metadata objects are to be altered, followed by a set of rowsets that define the properties of the objects that are to be altered. Properties that are not specified will remain unaltered, unless there are side-effects from altering other properties.

3.1.5.2.1.2.1 Request

The object types allowed are defined in the **TabularCommandType** object in section 3.1.5.2.1, and the schema of the rowsets for each of these object types is documented in sections 3.1.5.2.1.2.1.1 through 3.1.5.2.1.2.1.22.

The **Alter** command performs some basic validation. For example, references to objects, such as the Column referenced by a Level in a Hierarchy, are validated during execution of the Alter Tabular Metadata API. Other validations, such as syntax and semantic validation of DAX expressions, can be deferred until a later operation.

The object being altered is identified with a path based on the names of the parent objects (see section 3.1.5.2.1).

3.1.5.2.1.2.1.1 Alter Model

The **Alter Model** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="StorageLocation" type="xs:string" sql:field="StorageLocation"
minOccurs="0" />
          <xs:element name="Mode" type="xs:long" sql:field="Mode" minOccurs="0" />
          <xs:element name="Culture" type="xs:string" sql:field="Culture" minOccurs="0" />
          <xs:element name="Collation" type="xs:string" sql:field="Collation" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
Name
Description

Element
StorageLocation
Mode
Culture
Collation

The properties correspond to the **Model** object defined in section 2.2.5.1.

3.1.5.2.1.2.1.2 Alter DataSources

The **Alter DataSources** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.DataSource" type="xs:string" sql:field="ID.DataSource"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="ConnectionString" type="xs:string" sql:field="ConnectionString"
minOccurs="0" />
          <xs:element name="ImpersonationMode" type="xs:long" sql:field="ImpersonationMode"
minOccurs="0" />
          <xs:element name="Account" type="xs:string" sql:field="Account" minOccurs="0" />
          <xs:element name="Password" type="xs:string" sql:field="Password" minOccurs="0" />
          <xs:element name="MaxConnections" type="xs:int" sql:field="MaxConnections"
minOccurs="0" />
          <xs:element name="Isolation" type="xs:long" sql:field="Isolation" minOccurs="0" />
          <xs:element name="Timeout" type="xs:int" sql:field="Timeout" minOccurs="0" />
          <xs:element name="Provider" type="xs:string" sql:field="Provider" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.DataSource
Name
Description
ConnectionString
ImpersonationMode

Element
Account
Password
MaxConnections
Isolation
Timeout
Provider

The properties correspond to the **DataSource** object defined in section 2.2.5.2.

3.1.5.2.1.2.1.3 Alter Tables

The **Alter Tables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="DataCategory" type="xs:string" sql:field="DataCategory"
minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
Name
DataCategory
Description
IsHidden

The properties correspond to the **Table** object defined in section 2.2.5.3.

3.1.5.2.1.2.1.4 Alter Columns

The **Alter Columns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Column" type="xs:string" sql:field="ID.Column" minOccurs="0"
/>
          <xs:element name="ExplicitName" type="xs:string" sql:field="ExplicitName"
minOccurs="0" />
          <xs:element name="ExplicitDataType" type="xs:long" sql:field="ExplicitDataType"
minOccurs="0" />
          <xs:element name="DataCategory" type="xs:string" sql:field="DataCategory"
minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
          <xs:element name="IsUnique" type="xs:boolean" sql:field="IsUnique" minOccurs="0" />
          <xs:element name="IsKey" type="xs:boolean" sql:field="IsKey" minOccurs="0" />
          <xs:element name="IsNullable" type="xs:boolean" sql:field="IsNullable"
minOccurs="0" />
          <xs:element name="Alignment" type="xs:long" sql:field="Alignment" minOccurs="0" />
          <xs:element name="TableDetailPosition" type="xs:int"
sql:field="TableDetailPosition" minOccurs="0" />
          <xs:element name="IsDefaultLabel" type="xs:boolean" sql:field="IsDefaultLabel"
minOccurs="0" />
          <xs:element name="IsDefaultImage" type="xs:boolean" sql:field="IsDefaultImage"
minOccurs="0" />
          <xs:element name="SummarizeBy" type="xs:long" sql:field="SummarizeBy" minOccurs="0"
/>
          <xs:element name="SourceColumn" type="xs:string" sql:field="SourceColumn"
minOccurs="0" />
          <xs:element name="Expression" type="xs:string" sql:field="Expression" minOccurs="0"
/>
          <xs:element name="FormatString" type="xs:string" sql:field="FormatString"
minOccurs="0" />
          <xs:element name="IsAvailableInMDX" type="xs:boolean" sql:field="IsAvailableInMDX"
minOccurs="0" />
          <xs:element name="SortByColumnID" type="xs:unsignedLong" sql:field="SortByColumnID"
minOccurs="0" />
          <xs:element name="SortByColumnID.Table" type="xs:string"
sql:field="SortByColumnID.Table" minOccurs="0" />
          <xs:element name="SortByColumnID.Column" type="xs:string"
sql:field="SortByColumnID.Column" minOccurs="0" />
          <xs:element name="KeepUniqueRows" type="xs:boolean" sql:field="KeepUniqueRows"
minOccurs="0" />
          <xs:element name="DisplayOrdinal" type="xs:int" sql:field="DisplayOrdinal"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Column
ExplicitName
ExplicitDataType
DataCategory
Description
IsHidden
IsUnique
IsKey
IsNullable
Alignment
TableDetailPosition
IsDefaultLabel
IsDefaultImage
SummarizeBy
SourceColumn
Expression
FormatString
IsAvailableInMDX
SortByColumnID
SortByColumnID.Table
SortByColumnID.Column
KeepUniqueRows
DisplayOrdinal
DisplayFolder

The properties correspond to the **Column** object defined in section 2.2.5.4.

3.1.5.2.1.2.1.5 Alter Partitions

The **Alter Partitions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
```

```
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="DataSourceID" type="xs:unsignedLong" sql:field="DataSourceID"
minOccurs="0" />
          <xs:element name="DataSourceID.DataSource" type="xs:string"
sql:field="DataSourceID.DataSource" minOccurs="0" />
          <xs:element name="QueryDefinition" type="xs:string" sql:field="QueryDefinition"
minOccurs="0" />
          <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
          <xs:element name="Mode" type="xs:long" sql:field="Mode" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Partition
Name
Description
DataSourceID
DataSourceID.DataSource
QueryDefinition
Type
Mode

The properties correspond to the **Partition** object defined in section 2.2.5.6.

3.1.5.2.1.2.1.6 Alter Relationships

The **Alter Relationships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
```

```
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Relationship" type="xs:string" sql:field="ID.Relationship"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="IsActive" type="xs:boolean" sql:field="IsActive" minOccurs="0" />
          <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
          <xs:element name="CrossFilteringBehavior" type="xs:long"
sql:field="CrossFilteringBehavior" minOccurs="0" />
          <xs:element name="JoinOnDateBehavior" type="xs:long" sql:field="JoinOnDateBehavior"
minOccurs="0" />
          <xs:element name="RelyOnReferentialIntegrity" type="xs:boolean"
sql:field="RelyOnReferentialIntegrity" minOccurs="0" />
          <xs:element name="FromTableID" type="xs:unsignedLong" sql:field="FromTableID"
minOccurs="0" />
          <xs:element name="FromTableID.Table" type="xs:string" sql:field="FromTableID.Table"
minOccurs="0" />
          <xs:element name="FromColumnID" type="xs:unsignedLong" sql:field="FromColumnID"
minOccurs="0" />
          <xs:element name="FromColumnID.Table" type="xs:string"
sql:field="FromColumnID.Table" minOccurs="0" />
          <xs:element name="FromColumnID.Column" type="xs:string"
sql:field="FromColumnID.Column" minOccurs="0" />
          <xs:element name="FromCardinality" type="xs:long" sql:field="FromCardinality"
minOccurs="0" />
          <xs:element name="ToTableID" type="xs:unsignedLong" sql:field="ToTableID"
minOccurs="0" />
          <xs:element name="ToTableID.Table" type="xs:string" sql:field="ToTableID.Table"
minOccurs="0" />
          <xs:element name="ToColumnID" type="xs:unsignedLong" sql:field="ToColumnID"
minOccurs="0" />
          <xs:element name="ToColumnID.Table" type="xs:string" sql:field="ToColumnID.Table"
minOccurs="0" />
          <xs:element name="ToColumnID.Column" type="xs:string" sql:field="ToColumnID.Column"
minOccurs="0" />
          <xs:element name="ToCardinality" type="xs:long" sql:field="ToCardinality"
minOccurs="0" />
          <xs:element name="SecurityFilteringBehavior" type="xs:long"
sql:field="SecurityFilteringBehavior" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Relationship
Name
IsActive
Type
CrossFilteringBehavior
JoinOnDateBehavior
RelyOnReferentialIntegrity
FromTableID

Element
FromTableID.Table
FromColumnID
FromColumnID.Table
FromColumnID.Column
FromCardinality
ToTableID
ToTableID.Table
ToColumnID
ToColumnID.Table
ToColumnID.Column
ToCardinality
SecurityFilteringBehavior

The properties correspond to the **Relationship** object defined in section 2.2.5.7.

3.1.5.2.1.2.1.7 Alter Measures

The **Alter Measures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Measure" type="xs:string" sql:field="ID.Measure" minOccurs="0"
/>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="Expression" type="xs:string" sql:field="Expression" minOccurs="0"
/>
          <xs:element name="FormatString" type="xs:string" sql:field="FormatString"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
          <xs:element name="IsSimpleMeasure" type="xs:boolean" sql:field="IsSimpleMeasure"
minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Measure
Name
Description
Expression
FormatString
IsHidden
IsSimpleMeasure
DisplayFolder

The properties correspond to the **Measure** object defined in section 2.2.5.8.

3.1.5.2.1.2.1.8 Alter Hierarchies

The **Alter Hierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="IsHidden" type="xs:boolean" sql:field="IsHidden" minOccurs="0" />
      <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy

Element
Name
Description
IsHidden
DisplayFolder

The properties correspond to the **Hierarchy** object defined in section 2.2.5.9.

3.1.5.2.1.2.1.9 Alter Levels

The **Alter Levels** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
          <xs:element name="ID.Level" type="xs:string" sql:field="ID.Level" minOccurs="0" />
          <xs:element name="Ordinal" type="xs:int" sql:field="Ordinal" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="ColumnID" type="xs:unsignedLong" sql:field="ColumnID"
minOccurs="0" />
          <xs:element name="ColumnID.Table" type="xs:string" sql:field="ColumnID.Table"
minOccurs="0" />
          <xs:element name="ColumnID.Column" type="xs:string" sql:field="ColumnID.Column"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy
ID.Level
Ordinal
Name
Description

Element
ColumnID
ColumnID.Table
ColumnID.Column

The properties correspond to the **Level** object defined in section 2.2.5.10.

3.1.5.2.1.2.1.10 Alter Annotations

The **Alter Annotations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Value" type="xs:string" sql:field="Value" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
Name
Value

The properties correspond to the **Annotation** object defined in section 2.2.5.11.

3.1.5.2.1.2.1.11 Alter Kpis

The **Alter Kpis** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
```

```
            <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
            <xs:element name="ID.Measure" type="xs:string" sql:field="ID.Measure" minOccurs="0"
/>
            <xs:element name="ID.KPI" type="xs:string" sql:field="ID.KPI" minOccurs="0" />
            <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
            <xs:element name="TargetDescription" type="xs:string" sql:field="TargetDescription"
minOccurs="0" />
            <xs:element name="TargetExpression" type="xs:string" sql:field="TargetExpression"
minOccurs="0" />
            <xs:element name="TargetFormatString" type="xs:string"
sql:field="TargetFormatString" minOccurs="0" />
            <xs:element name="StatusGraphic" type="xs:string" sql:field="StatusGraphic"
minOccurs="0" />
            <xs:element name="StatusDescription" type="xs:string" sql:field="StatusDescription"
minOccurs="0" />
            <xs:element name="StatusExpression" type="xs:string" sql:field="StatusExpression"
minOccurs="0" />
            <xs:element name="TrendGraphic" type="xs:string" sql:field="TrendGraphic"
minOccurs="0" />
            <xs:element name="TrendDescription" type="xs:string" sql:field="TrendDescription"
minOccurs="0" />
            <xs:element name="TrendExpression" type="xs:string" sql:field="TrendExpression"
minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
```

Element
ID
ID.Table
ID.Measure
ID.KPI
Description
TargetDescription
TargetExpression
TargetFormatString
StatusGraphic
StatusDescription
StatusExpression
TrendGraphic
TrendDescription
TrendExpression

The properties correspond to the **KPI** object defined in section 2.2.5.12.

3.1.5.2.1.2.1.12 Alter Cultures

The **Alter Cultures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Culture" type="xs:string" sql:field="ID.Culture" minOccurs="0"
/>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Culture
Name

The properties correspond to the **Culture** object defined in section 2.2.5.13.

3.1.5.2.1.2.1.13 Alter ObjectTranslations

The **Alter ObjectTranslations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="Value" type="xs:string" sql:field="Value" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
Value

The properties correspond to the **ObjectTranslation** object defined in section 2.2.5.14.

3.1.5.2.1.2.1.14 Alter LinguisticMetadata

The **Alter LinguisticMetadata** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Culture" type="xs:string" sql:field="ID.Culture" minOccurs="0"
/>
          <xs:element name="ID.LinguisticMetadata" type="xs:string"
sql:field="ID.LinguisticMetadata" minOccurs="0" />
          <xs:element name="Content" type="xs:string" sql:field="Content" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Culture
ID.LinguisticMetadata
Content

The properties correspond to the **LinguisticMetadata** object defined in section 2.2.5.15.

3.1.5.2.1.2.1.15 Alter Perspectives

The **Alter Perspectives** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
        </xs:sequence>
```

```
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
Name
Description

The properties correspond to the **Perspective** object defined in section 2.2.5.16.

3.1.5.2.1.2.1.16 Alter PerspectiveTables

The **Alter PerspectiveTables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
          <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
          <xs:element name="IncludeAll" type="xs:boolean" sql:field="IncludeAll"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
TableID
TableID.Table
IncludeAll

The properties correspond to the **PerspectiveTable** object defined in section 2.2.5.17.

3.1.5.2.1.2.1.17 Alter PerspectiveColumns

The **Alter PerspectiveColumns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveColumn" type="xs:string"
sql:field="ID.PerspectiveColumn" minOccurs="0" />
          <xs:element name="ColumnID" type="xs:unsignedLong" sql:field="ColumnID"
minOccurs="0" />
          <xs:element name="ColumnID.Table" type="xs:string" sql:field="ColumnID.Table"
minOccurs="0" />
          <xs:element name="ColumnID.Column" type="xs:string" sql:field="ColumnID.Column"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
ID.PerspectiveColumn
ColumnID
ColumnID.Table
ColumnID.Column

The properties correspond to the **PerspectiveColumn** object defined in section 2.2.5.18.

3.1.5.2.1.2.1.18 Alter PerspectiveHierarchies

The **Alter PerspectiveHierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
```

```
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveHierarchy" type="xs:string"
sql:field="ID.PerspectiveHierarchy" minOccurs="0" />
          <xs:element name="HierarchyID" type="xs:unsignedLong" sql:field="HierarchyID"
minOccurs="0" />
          <xs:element name="HierarchyID.Table" type="xs:string" sql:field="HierarchyID.Table"
minOccurs="0" />
          <xs:element name="HierarchyID.Hierarchy" type="xs:string"
sql:field="HierarchyID.Hierarchy" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
ID.PerspectiveHierarchy
HierarchyID
HierarchyID.Table
HierarchyID.Hierarchy

The properties correspond to the **PerspectiveHierarchy** object defined in section 2.2.5.19.

3.1.5.2.1.2.1.19 Alter PerspectiveMeasures

The **Alter PerspectiveMeasures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
```

```
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveMeasure" type="xs:string"
sql:field="ID.PerspectiveMeasure" minOccurs="0" />
          <xs:element name="MeasureID" type="xs:unsignedLong" sql:field="MeasureID"
minOccurs="0" />
          <xs:element name="MeasureID.Table" type="xs:string" sql:field="MeasureID.Table"
minOccurs="0" />
          <xs:element name="MeasureID.Measure" type="xs:string" sql:field="MeasureID.Measure"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
ID.PerspectiveMeasure
MeasureID
MeasureID.Table
MeasureID.Measure

The properties correspond to the **PerspectiveMeasure** object defined in section 2.2.5.20.

3.1.5.2.1.2.1.20 Alter Roles

The **Alter Roles** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          <xs:element name="Description" type="xs:string" sql:field="Description"
minOccurs="0" />
          <xs:element name="ModelPermission" type="xs:long" sql:field="ModelPermission"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role
Name
Description
ModelPermission

The properties correspond to the **Role** object defined in section 2.2.5.21.

3.1.5.2.1.2.1.21 Alter RoleMemberships

The **Alter RoleMemberships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
          <xs:element name="ID.RoleMembership" type="xs:string" sql:field="ID.RoleMembership"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role
ID.RoleMembership

The properties correspond to the **RoleMembership** object defined in section 2.2.5.22.

3.1.5.2.1.2.1.22 Alter TablePermissions

The **Alter TablePermissions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
```

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:complexType name="row">
          <xs:sequence>
            <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
            <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
            <xs:element name="ID.TablePermission" type="xs:string"
sql:field="ID.TablePermission" minOccurs="0" />
            <xs:element name="TableID" type="xs:unsignedLong" sql:field="TableID" minOccurs="0"
/>
            <xs:element name="TableID.Table" type="xs:string" sql:field="TableID.Table"
minOccurs="0" />
            <xs:element name="FilterExpression" type="xs:string" sql:field="FilterExpression"
minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
```

Element
ID
ID.Role
ID.TablePermission
TableID
TableID.Table
FilterExpression

The properties correspond to the **TablePermission** object defined in section 2.2.5.23.

3.1.5.2.1.2.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.3 Delete Tabular Metadata

The **Delete Tabular Metadata** command is used to delete objects in a Tabular database that has the compatibility level set to 1200. The command requires a **DatabaseID** child element that identifies the database in which the Tabular metadata objects are to be deleted, followed by a set of rowsets that define the objects that are to be deleted.

3.1.5.2.1.3.1 Request

The object types allowed are defined in the **TabularCommandType** object in section 3.1.5.2.1, and the schema of the rowsets for these object types is documented in the following subsections.

Deletion of objects performs some basic validation. For example, references to parent objects, such as the table to which a **Column** object belongs, are validated during execution of the Delete Tabular Metadata API. Other validations, such as syntax and semantic validation of DAX expressions, can be deferred until a later operation.

The object being deleted is identified with a path based on the names of the parent objects (see section 3.1.5.2.1).

3.1.5.2.1.3.1.1 Delete Model

The **Delete Model** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence />
      </xs:complexType>
    </xs:schema>
```

3.1.5.2.1.3.1.2 Delete DataSources

The **Delete DataSources** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.DataSource" type="xs:string" sql:field="ID.DataSource"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.DataSource

3.1.5.2.1.3.1.3 Delete Tables

The **Delete Tables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table

3.1.5.2.1.3.1.4 Delete Columns

The **Delete Columns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Column" type="xs:string" sql:field="ID.Column" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Column

3.1.5.2.1.3.1.5 Delete Partitions

The **Delete Partitions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Partition

3.1.5.2.1.3.1.6 Delete Relationships

The **Delete Relationships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Relationship" type="xs:string" sql:field="ID.Relationship"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Relationship

3.1.5.2.1.3.1.7 Delete Measures

The **Delete Measures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Measure" type="xs:string" sql:field="ID.Measure" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Measure

3.1.5.2.1.3.1.8 Delete Hierarchies

The **Delete Hierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy

3.1.5.2.1.3.1.9 Delete Levels

The **Delete Levels** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
          <xs:element name="ID.Level" type="xs:string" sql:field="ID.Level" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy
ID.Level

3.1.5.2.1.3.1.10 Delete Annotations

The **Delete Annotations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID

3.1.5.2.1.3.1.11 Delete Kpis

The **Delete Kpis** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Measure" type="xs:string" sql:field="ID.Measure" minOccurs="0"
/>
          <xs:element name="ID.KPI" type="xs:string" sql:field="ID.KPI" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Measure
ID.KPI

3.1.5.2.1.3.1.12 Delete Cultures

The **Delete Cultures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Culture" type="xs:string" sql:field="ID.Culture" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Culture

3.1.5.2.1.3.1.13 Delete ObjectTranslations

The **Delete ObjectTranslations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID

3.1.5.2.1.3.1.14 Delete LinguisticMetadata

The **Delete LinguisticMetadata** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Culture" type="xs:string" sql:field="ID.Culture" minOccurs="0"
/>
          <xs:element name="ID.LinguisticMetadata" type="xs:string"
sql:field="ID.LinguisticMetadata" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Culture
ID.LinguisticMetadata

3.1.5.2.1.3.1.15 Delete Perspectives

The **Delete Perspectives** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective

3.1.5.2.1.3.1.16 Delete PerspectiveTables

The **Delete PerspectiveTables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
```

```
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable

3.1.5.2.1.3.1.17 Delete PerspectiveColumns

The **Delete PerspectiveColumns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveColumn" type="xs:string"
sql:field="ID.PerspectiveColumn" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
ID.PerspectiveColumn

3.1.5.2.1.3.1.18 Delete PerspectiveHierarchies

The **Delete PerspectiveHierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
```

```
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveHierarchy" type="xs:string"
sql:field="ID.PerspectiveHierarchy" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable
ID.PerspectiveHierarchy

3.1.5.2.1.3.1.19 Delete PerspectiveMeasures

The **Delete PerspectiveMeasures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="ID.PerspectiveTable" type="xs:string"
sql:field="ID.PerspectiveTable" minOccurs="0" />
          <xs:element name="ID.PerspectiveMeasure" type="xs:string"
sql:field="ID.PerspectiveMeasure" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
ID.PerspectiveTable

Element
ID.PerspectiveMeasure

3.1.5.2.1.3.1.20 Delete Roles

The **Delete Roles** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role

3.1.5.2.1.3.1.21 Delete RoleMemberships

The **Delete RoleMemberships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
          <xs:element name="ID.RoleMembership" type="xs:string" sql:field="ID.RoleMembership"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role
ID.RoleMembership

3.1.5.2.1.3.1.22 Delete TablePermissions

The **Delete TablePermissions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
          <xs:element name="ID.TablePermission" type="xs:string"
sql:field="ID.TablePermission" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role
ID.TablePermission

3.1.5.2.1.3.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.4 Rename Tabular Metadata

The **Rename Tabular Metadata** command is used to rename objects in a Tabular database that has the compatibility level set to 1200. The command requires a **DatabaseID** child element that identifies the database in which the Tabular metadata objects are to be renamed, followed by a set of rowsets

that define the new names of the objects. The Rename API will automatically update the references to the renamed objects in DAX expressions.

3.1.5.2.1.4.1 Request

The object types allowed are defined in the **TabularCommandType** object in section 3.1.5.2.1, and the schema of the rowsets for these object types is documented in the following subsections.

Renaming of objects performs some basic validation. For example, references to parent objects, such as the table to which a **Column** object belongs, are validated during execution of the Rename Tabular Metadata API. Other validations, such as syntax and semantic validation of DAX expressions, can be deferred until a later operation.

The object being renamed is identified with a path based on the names of the parent objects (see section 3.1.5.2.1).

3.1.5.2.1.4.1.1 Rename Model

The **Rename Model** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
Name

3.1.5.2.1.4.1.2 Rename DataSources

The **Rename DataSources** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.DataSource" type="xs:string" sql:field="ID.DataSource"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.DataSource
Name

3.1.5.2.1.4.1.3 Rename Tables

The **Rename Tables** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
Name

3.1.5.2.1.4.1.4 Rename Columns

The **Rename Columns** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
```

```
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Column" type="xs:string" sql:field="ID.Column" minOccurs="0"
/>
          <xs:element name="ExplicitName" type="xs:string" sql:field="ExplicitName"
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Column
ExplicitName

3.1.5.2.1.4.1.5 Rename Partitions

The **Rename Partitions** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Partition
Name

3.1.5.2.1.4.1.6 Rename Relationships

The **Rename Relationships** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Relationship" type="xs:string" sql:field="ID.Relationship"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Relationship
Name

3.1.5.2.1.4.1.7 Rename Measures

The **Rename Measures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Measure" type="xs:string" sql:field="ID.Measure" minOccurs="0"
/>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Measure

Element
Name

3.1.5.2.1.4.1.8 Rename Hierarchies

The **Rename Hierarchies** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy
Name

3.1.5.2.1.4.1.9 Rename Levels

The **Rename Levels** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Hierarchy" type="xs:string" sql:field="ID.Hierarchy"
minOccurs="0" />
          <xs:element name="ID.Level" type="xs:string" sql:field="ID.Level" minOccurs="0" />
```

```
            <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
```

Element
ID
ID.Table
ID.Hierarchy
ID.Level
Name

3.1.5.2.1.4.1.10 Rename Annotations

The **Rename Annotations** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
Name

3.1.5.2.1.4.1.11 Rename Cultures

The **Rename Cultures** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
```

```
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Culture" type="xs:string" sql:field="ID.Culture" minOccurs="0"
/>
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Culture
Name

3.1.5.2.1.4.1.12 Rename Perspectives

The **Rename Perspectives** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Perspective" type="xs:string" sql:field="ID.Perspective"
minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Perspective
Name

3.1.5.2.1.4.1.13 Rename Roles

The **Rename Roles** schema definition is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
```

```
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Role" type="xs:string" sql:field="ID.Role" minOccurs="0" />
          <xs:element name="Name" type="xs:string" sql:field="Name" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element
ID
ID.Role
Name

3.1.5.2.1.4.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.5 Refresh Tabular Metadata

The schema definition for the **Refresh** command is as follows.

```
<xs:complexType name="TabularRefreshCommandType">
    <xs:sequence>
      <xs:element name="MaxParallelism" type="xs:int" minOccurs="0" maxOccurs="1" />
      <xs:element name="DatabaseID" type="xs:string" />
      <xs:element name="PushedData" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="EndOfData" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:sequence minOccurs="1" maxOccurs="unbounded">
        <xs:choice minOccurs="1" maxOccurs="1">
          <xs:element name="Model" type="xmla-rs:rowset" />
          <xs:element name="Tables" type="xmla-rs:rowset" />
          <xs:element name="Partitions" type="xmla-rs:rowset" />
        </xs:choice>
      </xs:sequence>
      <xs:element name="Bindings" type="mstns:TabularBindingsType" />
    </xs:sequence>
</xs:complexType>
```

The **Refresh Tabular Metadata** command is used to refresh objects in a Tabular database that has the compatibility level set to 1200. The command requires a **DatabaseID** child element that identifies

the database in which the Tabular metadata objects are to be refreshed, followed by a set of rowsets that define the objects that are to be refreshed.

3.1.5.2.1.5.1 Request

The allowed object types are defined in the **TabularCommandType** object in section 3.1.5.2.1, and the schema of the rowsets for these object types is described in sections 3.1.5.2.1.5.1.1 through 3.1.5.2.1.5.1.5.

Refreshing objects performs some basic validation. For example, references to parent objects, such as the table to which a **Column** object belongs, are validated during execution of the Refresh Tabular Metadata API. Other validations, such as syntax and semantic validation of DAX expressions, can be deferred until a later operation.

The object being refreshed is identified with a path based on the names of the parent objects (see section 3.1.5.2.1).

The following table describes the elements of the **TabularRefreshCommandType** complex type.

Element	Type	Description
MaxParallelism	Integer	Optional. This value indicates the desired maximum parallelism for the refresh operation.
DatabaseID	String	The identifier of the database that is being refreshed
PushedData	String	Optional. The name of an XMLA parameter that contains a rowset to be pushed into a partition in the data model.
EndOfData	String	Optional. The name of an XMLA parameter that specifies whether a pushed rowset is the last rowset to be pushed into the partition.

3.1.5.2.1.5.1.1 Refresh Model

The schema definition for the **Model** element in **TabularRefreshCommandType** is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="RefreshType" type="xs:long" sql:field="RefreshType" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value	Description
RefreshType	"Full"	Specifies whether to refresh the data and recalculate or clear all dependents. The possible values are as follows.

Element	Default value	Description
		▪ **Full** (1) - For a regular partition, refresh data and recalculate all dependents. For a calculation partition, recalculate the partition and all its dependents. ▪ **ClearValues** (2) - Clear values in this object and all its dependents. ▪ **Calculate** (3) - Recalculate this object and all its dependents, but only if needed. This value does not force recalculation, except for **volatile** formulas. ▪ **DataOnly** (4) – Refresh data in this object and clear all dependents. ▪ **Automatic** (5) - If the object needs to be refreshed and recalculated, refresh and recalculate the object and all its dependents. Applies if the partition is in a state other than Ready (see section 2.2.5.6).

3.1.5.2.1.5.1.2 Refresh Tables

The schema definition for the **Tables** element in **TabularRefreshCommandType** is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="RefreshType" type="xs:long" sql:field="RefreshType" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value	Description
ID		An ID-based reference to a **Table** object.
ID.Table		A name-based path to the **Table** object specified by **ID**.
RefreshType	"Full"	Specifies whether to refresh the data and recalculate or clear all dependents. The possible values are as follows. ▪ **Full** (1) - For a regular partition, refresh data and recalculate all dependents. For a calculation partition, recalculate the partition and all its dependents. ▪ **ClearValues** (2) - Clear values in this object and all its dependents. ▪ **Calculate** (3) - Recalculate this object and all its dependents, but only if needed. This value does not force recalculation, except for volatile formulas. ▪ **DataOnly** (4) – Refresh data in this object and clear all dependents.

Element	Default value	Description
		▪ **Automatic** (5) - If the object needs to be refreshed and recalculated, refresh and recalculate the object and all its dependents. Applies if the partition is in a state other than Ready (see section 2.2.5.6).

3.1.5.2.1.5.1.3 Refresh Partitions

The schema definition for the **Partitions** element in **TabularRefreshCommandType** is as follows.

```
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="row" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="row">
        <xs:sequence>
          <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
          <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
          <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
          <xs:element name="RefreshType" type="xs:long" sql:field="RefreshType" minOccurs="0"
/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

Element	Default value	Description
ID		An ID-based reference to a **Partition** object.
ID.Table		The Table part of the name-based path to the **Partition** object specified by **ID**.
ID.Partition		The Partition part of the name-based path to the **Partition** object specified by **ID**.
RefreshType	"Full"	Specifies whether to refresh the data and recalculate or clear all dependents. The possible values are as follows. ▪ **Full** (1) - For a regular partition, refresh data and recalculate all dependents. For a calculation partition, recalculate the partition and all its dependents. ▪ **ClearValues** (2) - Clear values in this object and all its dependents. ▪ **Calculate** (3) - Recalculate this object and all its dependents, but only if needed. This value does not force recalculation, except for volatile formulas. ▪ **DataOnly** (4) – Refresh data in this object and clear all dependents. ▪ **Automatic** (5) - If the object needs to be refreshed and recalculated, refresh and recalculate the object and all its dependents. Applies if the partition is in a state other than Ready (see section 2.2.5.6). ▪ **Add** (6) - Append data to this partition and recalculate all dependents. This command is valid only for regular partitions and not for calculation partitions.

3.1.5.2.1.5.1.4 Out of Line Bindings

While issuing a **Refresh Tabular Metadata** command, users can use the **Bindings** element inside the **TabularRefreshCommandType** to change the properties of certain objects for the scope of the **Refresh** request. The properties that can be changed include DataSources, Partitions, and Columns.

The schema definition for the **Bindings** element is as follows.

```
<xs:complexType name="TabularBindingsType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Binding" type="mstns:TabularBindingType"/>
  </xs:sequence>
</xs:complexType>
```

Each **Binding** element will provide the values to override for the changed objects while refreshing the specific partition mentioned in the **Binding** element.

The schema definition for the **Binding** element is as follows.

```
<xs:complexType name="TabularBindingType">
      <xs:sequence>
            <xs:element name="ObjectID" type="xs:unsignedLong" minOccurs="0" />
            <xs:element name="TableName" type="xs:string" minOccurs="0" />
            <xs:element name="PartitionName" type="xs:string" minOccurs="0" />
            <xs:element name="DataSources" type="xmla-rs:rowset" />
            <xs:element name="Columns" type="xmla-rs:rowset" />
            <xs:element name="Partitions" type="xmla-rs:rowset" />
      </xs:sequence>
</xs:complexType>
```

The **Binding** element contains the following fields.

Element	Default value	Description
ObjectID		An ID-based reference to the **Partition** object for which the out-of-line bindings are to be applied before refreshing the partition.
TableName		The Table part of the name-based path to the **Partition** object for which the out-of-line bindings are to be applied before refreshing the partition.
PartitionName		The Partition part of the name-based path to the **Partition** object for which the out-of-line bindings are to be applied before refreshing the partition.

The schema definitions for the remaining elements of the **TabularBindingType** are as follows.

DataSources:

For possible values of the elements defined in this schema, see section 2.2.5.2.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element type="row" />
```

```
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="row">
    <xs:sequence>
      <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
      <xs:element name="ID.DataSource" type="xs:string" sql:field="ID.DataSource"
minOccurs="0" />
      <xs:element name="ConnectionString" type="xs:string" sql:field="ConnectionString"
minOccurs="0" />
      <xs:element name="ImpersonationMode" type="xs:long" sql:field="ImpersonationMode"
minOccurs="0" />
      <xs:element name="Account" type="xs:string" sql:field="Account" minOccurs="0" />
      <xs:element name="Password" type="xs:string" sql:field="Password" minOccurs="0" />
      <xs:element name="MaxConnections" type="xs:int" sql:field="MaxConnections"
minOccurs="0" />
      <xs:element name="Isolation" type="xs:long" sql:field="Isolation" minOccurs="0" />
      <xs:element name="Timeout" type="xs:int" sql:field="Timeout" minOccurs="0" />
      <xs:element name="Provider" type="xs:string" sql:field="Provider" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Partitions:

For possible values of the elements defined in this schema, see section 2.2.5.6.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element type="row" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="row">
    <xs:sequence>
      <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
      <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
      <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition" minOccurs="0"
/>
      <xs:element name="DataSourceID" type="xs:unsignedLong" sql:field="DataSourceID"
minOccurs="0" />
      <xs:element name="DataSourceID.DataSource" type="xs:string"
sql:field="DataSourceID.DataSource" minOccurs="0" />
      <xs:element name="QueryDefinition" type="xs:string" sql:field="QueryDefinition"
minOccurs="0" />
      <xs:element name="Type" type="xs:long" sql:field="Type" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Columns:

For possible values of the elements defined in this schema, see section 2.2.5.4.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element type="row" />
      </xs:sequence>
```

```
      </xs:complexType>
    </xs:element>
    <xs:complexType name="row">
      <xs:sequence>
        <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0" />
        <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0" />
        <xs:element name="ID.Column" type="xs:string" sql:field="ID.Column" minOccurs="0" />
        <xs:element name="SourceColumn" type="xs:string" sql:field="SourceColumn" minOccurs="0"
/>
      </xs:sequence>
    </xs:complexType>
  </xs:sequence>
```

See section 4.1 for an example of the **Refresh** command with out-of-line bindings.

3.1.5.2.1.5.1.5 Pushed Data

As part of **Tabular Refresh** command, users can submit data to be pushed into a partition. This can be achieved by using the **PushedData** and **EndOfData** elements in the **TabularRefreshCommandType**.

The data to be pushed into a partition is passed in as **Parameters** of the Execute request. For more information on **Parameters**, see [MS-SSAS] section 3.1.4.3.2.1.3.

The name of the parameter that will contain the data is specified in the **PushedData** element of the **Refresh** command. The name of the parameter that can be used to signal the end of data is passed in the **EndOfData** element of the **Refresh** command. The value of the parameter with the **PushedData** element name is a rowset, and the value of the parameter with the **EndOfData** element name is a Boolean. If that Boolean is true, any further parameters passed in the request with the name of the **PushedData** element will not be pushed into the partition. There can be more than one parameter with the **PushedData** element name. These parameters will be pushed into the partition in the same order as they are sent until the parameter with the **EndOfData** element name is encountered.

When data to be pushed into the partitions is passed in as part of the **Refresh** command, only one partition can be processed by using that data. If more than one partition tries to use the data, the engine will throw an error.

See section 4.1 for an example of the **Refresh** command with pushed data and out-of-line bindings.

3.1.5.2.1.5.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.6 MergePartitions Tabular Metadata

The **MergePartitions** command merges the data of the specified source partitions into a target partition.

3.1.5.2.1.6.1 Request

The **MergePartitions** schema definition is as follows.

```
<xs:complexType name="TabularMergePartitionCommandType">
  <xs:sequence>
    <xs:element name="DatabaseID" type="xs:string" minOccurs="1"/>
    <xs:element name="PartitionID" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="TableName" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:element name="PartitionName" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:sequence minOccurs="1" maxOccurs="1">
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element name="Partitions" type="xmla-rs:rowset" />
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
```

The XSD for the **Partitions** rowset is as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-
com:xml-sql">
  <xs:element>
    <xs:complexType>
      <xs:sequence>
        <xs:element type="row" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="row">
    <xs:sequence>
      <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0"/>
      <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table" minOccurs="0"/>
      <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

The XML elements that are included in the **MergePartitions** command are described in the following table.

Element	Default value	Description
DatabaseID		A mandatory reference to the database.
PartitionID		An optional reference to the target partition,
TableName		The Table part of the name-based path to the target **Partition** object.
PartitionName		The Partition part of the name-based path to the target **Partition** object.
Partitions		The source partitions whose data will be merged into the target partition. These partitions will be deleted at the end of the command.

3.1.5.2.1.6.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.7 DBCC for Tabular Metadata

The Database Consistency Check (DBCC) command is used to check consistency of objects on the server.

3.1.5.2.1.7.1 Request

The **DBCC** schema definition is as follows.

```
<xsd:complexType name="DBCC">
  <xsd:sequence>
    <xsd:element name="DatabaseID" type="string" />
    <xsd:element name="TableName" type="string" />
    <xsd:element name="PartitionName" type="string" />
  </xsd:sequence>
</xsd:complexType>
```

The following table shows the XML elements included in the DBCC command.

Element	Default value	Description
DatabaseID	[Required]	The ID of the database object to check for consistency.
TableName	[Optional]	The name of the table object to check for consistency.
PartitionName	[Optional]	The name of the partition object in the specified table to check for consistency.

The return result type for the DBCC command is **xmla-e:emptyresult** (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.1.7.2 Response

The response of a DBCC for Tabular Metadata request is an empty element.

3.1.5.2.1.8 SequencePoint

The **SequencePoint** command applies any pending changes in the current transaction. This process is referred to as the sequence point algorithm. The algorithm performs various actions including, but not limited to, analyzing all pending changes, applying validation rules, inferring object names, inferring data types, inferring calculated table schemas, and changing the states of the objects. The goal of this algorithm is to bring the data model into a consistent state so that it can be queried.

The **SequencePoint** command does not commit the transaction.

3.1.5.2.1.8.1 Request

The **SequencePoint** command requires a **DatabaseID** child element that identifies the database that has pending changes.

The **SequencePoint** schema definition is as follows.

```
<xsd:complexType name="SequencePoint">
  <xsd:sequence>
    <xsd:element name="DatabaseID" type="string" />
  </xsd:sequence>
```

```
</xsd:complexType>
```

The following table shows the XML elements included in the **SequencePoint** command.

Element	Default value	Description
DatabaseID	[Required]	The ID of the database object on which the sequence point algorithm is executed.

3.1.5.2.1.8.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

If the **ReturnAffectedObjects** XMLA property is set to 0, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

If the **ReturnAffectedObjects** XMLA property is set to 1, the response is an object of type **AffectedObjects**. The structure of the **AffectedObjects** element is defined in section 2.2.3.1.

3.1.5.2.1.9 Upgrade Tabular Metadata

The **Upgrade** command upgrades a Tabular database that has the compatibility level set below 1200 to compatibility level 1200.

3.1.5.2.1.9.1 Request

The command requires a **DatabaseID** child element that identifies the database in which the Tabular metadata objects are to be upgraded, followed by a set of rowsets that define the properties of the objects that are to be altered.

The allowed object types are defined in the **TabularCommandType** object (see section 3.1.5.2.1), and the schema of the rowsets for these object types is documented in section 3.1.5.2.1.1.

The Upgrade operation performs validations to ensure that the objects and properties of the model that is being upgraded properly match the objects in the original database.

3.1.5.2.1.9.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2 JSON-Based Tabular Metadata Commands

As described in section 3.1.5.2, a Tabular database can be administered by using two types of APIs, namely XMLA [XMLA] and JSON [RFC4627]. Sections 3.1.5.2.1 through 3.1.5.2.1.9.2 describe the XMLA-based APIs, and this section through section 3.1.5.2.2.13.2 describe the syntax of those APIs by using the JSON syntax [JSON-SchemaVal].

The JSON APIs are accepted as textual content under the **Statement** XMLA element as documented in [MS-SSAS].

Although there is significant overlap between the XMLA and JSON APIs, there are some commands that are unique to each one. Common guidelines that apply to the objects and properties for the JSON APIs include the following.

- JSON APIs use camel-casing for all object names and property names.
- JSON APIs always use name-based object references. The XMLA APIs support integer IDs, but JSON APIs are intended for end users and therefore use the more user-friendly name-based style.
- JSON APIs use textual enumeration values instead of integer enumeration values. As above, this is because the JSON APIs are targeted at end users and text-based enumerations are more user-friendly.
- The JSON APIs are naturally hierarchical. For example, creation of a Table defines the Columns as child nodes in the document structure.

Unless specified otherwise, each JSON command is performed with the following transactional semantics.

- If a transaction is already in progress, the command will execute but will commit when the application commits the transaction.
- If a transaction is not in progress, the command will execute and automatically commit.

3.1.5.2.2.1 Object Definitions in JSON Commands

The create, createOrReplace, and alter JSON commands accept new object definitions for the objects that are being created, replaced, or altered, respectively. Sections 3.1.5.2.2.1.1 through 3.1.5.2.2.1.22 document the structure and schema of these objects in JSON [JSON-SchemaVal].

3.1.5.2.2.1.1 Database

The JSON schema for the **database** object is as follows.

```
"database": {
  "description": "Database object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "description": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    },
    "compatibilityLevel": {
      "type": "integer"
    },
    "readWriteMode": {
      "enum": [
        "readWrite",
        "readOnly",
```

```
          "readOnlyExclusive"
        ]
      },
      "model": {
        "type": "object",
        ...
      }
    }
  }
```

The properties correspond to the properties of the **Database** object that is defined in [MS-SSAS].

In addition, this object can contain a child object named **model** which is of type **Model** (see section 3.1.5.2.2.1.2).

3.1.5.2.2.1.2 Model

The JSON schema for the **model** object is as follows.

```
"model": {
  "description": "Model object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "description": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    },
    "storageLocation": {
      "type": "string"
    },
    "defaultMode": {
      "enum": [
        "import",
        "directQuery",
        "default"
      ]
    },
    "defaultDataView": {
      "enum": [
        "full",
        "sample",
        "default"
      ]
    },
    "culture": {
      "type": "string"
    },
    "collation": {
      "type": "string"
    },
    "annotations": ": {
      "type": "array",
      "items": {
        ...
```

```
        }
      },
      "tables": {
        "type": "array",
        "items": {
          ...
        }
      },
      "relationships": {
        "type": "array",
        "items": {
          ...
        }
      },
      "dataSources": {
        "type": "array",
        "items": {
          ...
        }
      },
      "perspectives": {
        "type": "array",
        "items": {
          ...
        }
      },
      "cultures": {
        "type": "array",
        "items": {
          ...
        }
      },
      "roles": {
        "type": "array",
        "items": {
          ...
        }
      }
    }
  }
```

The properties correspond to the **Model** object that is defined in section 2.2.5.1. In addition, the object can include collections of child objects for Annotations, Tables, Relationships, DataSources, Perspectives, Cultures, and Roles.

3.1.5.2.2.1.3 DataSource

The JSON schema for the **dataSource** object is as follows.

```
  "dataSource": {
    "description": "ProviderDataSource object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "type": {
        "enum": [
          "provider"
        ]
      },
      "connectionString": {
        "type": "string"
```

```
      },
      "impersonationMode": {
        "enum": [
          "impersonateAccount",
          "impersonateAnonymous",
          "impersonateCurrentUser",
          "impersonateServiceAccount",
          "impersonateUnattendedAccount"
        ]
      },
      "account": {
        "type": "string"
      },
      "password": {
        "type": "string"
      },
      "maxConnections": {
        "type": "integer"
      },
      "isolation": {
        "enum": [
          "readCommitted",
          "snapshot"
        ]
      },
      "timeout": {
        "type": "integer"
      },
      "provider": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
      ...
        }
      }
    },
    "additionalProperties": false
  }
```

The properties correspond to the **DataSource** object that is defined in section 2.2.5.2. In addition, the object can include a collection of child objects for Annotations.

3.1.5.2.2.1.4 Table

The JSON schema for the **table** object is as follows.

```
  "table": {
    "description": "Table object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "dataCategory": {
        "type": "string"
      },
      "description": {
        "anyOf": [
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
```

```
              "type": "string"
            }
          }
        ]
      },
      "isHidden": {
        "type": "boolean"
      },
      "partitions": {
        "type": "array",
        "items": {
          ...
        }
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      },
      "columns": {
        "type": "array",
        "items": {
          ...
        }
      },
      "measures": {
        "type": "array",
        "items": {
          ...
        }
      },
      "hierarchies": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
```

The properties correspond to the **Table** object that is defined in section 2.2.5.3. In addition, the object can include collections of child objects for Partitions, Annotations, Columns, Measures, and Hierarchies.

3.1.5.2.2.1.5 Column

The JSON schema for the **column** object is as follows.

```
  "anyOf": [
    {
      "description": "DataColumn object of Tabular Object Model (TOM)",
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "dataType": {
          "enum": [
            "automatic",
            "string",
            "int64",
            "double",
            "dateTime",
            "decimal",
```

```
        "boolean",
        "binary",
        "unknown",
        "variant"
      ]
    },
    "dataCategory": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "isHidden": {
      "type": "boolean"
    },
    "isUnique": {
      "type": "boolean"
    },
    "isKey": {
      "type": "boolean"
    },
    "isNullable": {
      "type": "boolean"
    },
    "alignment": {
      "enum": [
        "default",
        "left",
        "right",
        "center"
      ]
    },
    "tableDetailPosition": {
      "type": "integer"
    },
    "isDefaultLabel": {
      "type": "boolean"
    },
    "isDefaultImage": {
      "type": "boolean"
    },
    "summarizeBy": {
      "enum": [
        "default",
        "none",
        "sum",
        "min",
        "max",
        "count",
        "average",
        "distinctCount"
      ]
    },
    "type": {
      "enum": [
        "data",
        "calculated",
        "rowNumber",
        "calculatedTableColumn"
      ]
    },
    "formatString": {
      "type": "string"
    },
    "isAvailableInMdx": {
      "type": "boolean"
    },
    "keepUniqueRows": {
      "type": "boolean"
```

```
      },
      "displayOrdinal": {
        "type": "integer"
      },
      "sourceProviderType": {
        "type": "string"
      },
      "displayFolder": {
        "type": "string"
      },
      "sourceColumn": {
        "type": "string"
      },
      "sortByColumn": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
            ...
        }
      }
    },
    "additionalProperties": false
  },
  {
    "description": "CalculatedTableColumn object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "dataType": {
        "enum": [
          "automatic",
          "string",
          "int64",
          "double",
          "dateTime",
          "decimal",
          "boolean",
          "binary",
          "unknown",
          "variant"
        ]
      },
      "dataCategory": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "isHidden": {
        "type": "boolean"
      },
      "isUnique": {
        "type": "boolean"
      },
      "isKey": {
        "type": "boolean"
      },
      "isNullable": {
        "type": "boolean"
      },
      "alignment": {
        "enum": [
          "default",
          "left",
          "right",
```

```
        "center"
      ]
    },
    "tableDetailPosition": {
      "type": "integer"
    },
    "isDefaultLabel": {
      "type": "boolean"
    },
    "isDefaultImage": {
      "type": "boolean"
    },
    "summarizeBy": {
      "enum": [
        "default",
        "none",
        "sum",
        "min",
        "max",
        "count",
        "average",
        "distinctCount"
      ]
    },
    "type": {
      "enum": [
        "data",
        "calculated",
        "rowNumber",
        "calculatedTableColumn"
      ]
    },
    "formatString": {
      "type": "string"
    },
    "isAvailableInMdx": {
      "type": "boolean"
    },
    "keepUniqueRows": {
      "type": "boolean"
    },
    "displayOrdinal": {
      "type": "integer"
    },
    "sourceProviderType": {
      "type": "string"
    },
    "displayFolder": {
      "type": "string"
    },
    "isNameInferred": {
      "type": "boolean"
    },
    "isDataTypeInferred": {
      "type": "boolean"
    },
    "sourceColumn": {
      "type": "string"
    },
    "sortByColumn": {
      "type": "string"
    },
    "columnOriginTable": {
      "type": "string"
    },
    "columnOriginColumn": {
      "type": "string"
    },
    "annotations": {
```

```
        "type": "array",
        "items": {
            ...
        }
      }
    },
    "additionalProperties": false
  },
  {
    "description": "CalculatedColumn object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "dataType": {
        "enum": [
          "automatic",
          "string",
          "int64",
          "double",
          "dateTime",
          "decimal",
          "boolean",
          "binary",
          "unknown",
          "variant"
        ]
      },
      "dataCategory": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "isHidden": {
        "type": "boolean"
      },
      "isUnique": {
        "type": "boolean"
      },
      "isKey": {
        "type": "boolean"
      },
      "isNullable": {
        "type": "boolean"
      },
      "alignment": {
        "enum": [
          "default",
          "left",
          "right",
          "center"
        ]
      },
      "tableDetailPosition": {
        "type": "integer"
      },
      "isDefaultLabel": {
        "type": "boolean"
      },
      "isDefaultImage": {
        "type": "boolean"
      },
      "summarizeBy": {
        "enum": [
          "default",
          "none",
          "sum",
```

```
            "min",
            "max",
            "count",
            "average",
            "distinctCount"
          ]
        },
        "type": {
          "enum": [
            "data",
            "calculated",
            "rowNumber",
            "calculatedTableColumn"
          ]
        },
        "formatString": {
          "type": "string"
        },
        "isAvailableInMdx": {
          "type": "boolean"
        },
        "keepUniqueRows": {
          "type": "boolean"
        },
        "displayOrdinal": {
          "type": "integer"
        },
        "sourceProviderType": {
          "type": "string"
        },
        "displayFolder": {
          "type": "string"
        },
        "isDataTypeInferred": {
          "type": "boolean"
        },
        "expression": {
          "type": "string"
        },
        "sortByColumn": {
          "type": "string"
        },
        "annotations": {
          "type": "array",
          "items": {
              ...
          }
        }
      },
      "additionalProperties": false
    }
  ]
```

The **column** object in JSON supports derived classes. The following derived types are supported.

- **data**: Data obtained from a column in the data source
- **calculated**: A column whose values are computed from a calculation expression
- **rowNumber**: An internally defined column that automatically generates a unique number for each row in the table
- **calculatedTableColumn**: A column whose values are computed from the result of a calculated table

Each of these derived types has the base properties, and a few extra properties that can apply. The extra properties are documented in the following tables.

For the **data** derived type:

Property	Description
sourceProviderType	SourceProviderType, described in section 2.2.5.4.
sourceColumn	SourceColumn, described in section 2.2.5.4.

For the **calculated** derived type:

Property	Description
isDataTypeInferred	A Boolean that indicates whether the data type is inferred or explicit. True if the data type is inferred from the expression; false if the data type is explicitly set.
expression	Expression, described in section 2.2.5.4.

The **rowNumber** derived type has no additional properties.

For the **calculatedTableColumn** derived type:

Property	Description
sourceProviderType	SourceProviderType, described in section 2.2.5.4.
isNameInferred	A Boolean that indicates whether the name of the column is inferred or explicit. True if the name is inferred from the calculated table; false if the name is explicitly specified.
isDataTypeInferred	A Boolean that indicates whether the data type is inferred or explicit. True if the data type is inferred from the calculated table expression; false if the data type is explicitly set.
sourceColumn	The name of the column in the calculated table expression that this column represents.
columnOriginTable	If the calculated table expression returns a column whose lineage can be determined, this property indicates the table from which the values are computed.
columnOriginColumn	If the calculated table expression returns a column whose lineage can be determined, this property indicates the column from which the values are computed.

3.1.5.2.2.1.6 Partition

The JSON schema for the **partition** object is as follows.

```
"partition": {
```

```
        "description": "Partition object of Tabular Object Model (TOM)",
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "description": {
            "anyOf": [
              {
                "type": "string"
              },
              {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
            ]
          },
          "mode": {
            "enum": [
              "import",
              "directQuery",
              "default"
            ]
          },
          "dataView": {
            "enum": [
              "full",
              "sample",
              "default"
            ]
          },
          "source": {
            "anyOf": [
              {
                "description": "QueryPartitionSource object of Tabular Object Model (TOM)",
                "type": "object",
                "properties": {
                  "type": {
                    "enum": [
                      "query",
                      "calculated",
                      "none"
                    ]
                  },
                  "query": {
                    "anyOf": [
                      {
                        "type": "string"
                      },
                      {
                        "type": "array",
                        "items": {
                          "type": "string"
                        }
                      }
                    ]
                  },
                  "dataSource": {
                    "type": "string"
                  }
                },
                "additionalProperties": false
              },
              {
                "description": "CalculatedPartitionSource object of Tabular Object Model
(TOM)",
                "type": "object",
```

```
            "properties": {
              "type": {
                "enum": [
                  "query",
                  "calculated",
                  "none"
                ]
              },
              "expression": {
                "anyOf": [
                  {
                    "type": "string"
                  },
                  {
                    "type": "array",
                    "items": {
                      "type": "string"
                    }
                  }
                ]
              }
            },
            "additionalProperties": false
          }
        ]
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
```

The **Partition** data type is expressed slightly differently compared with the common data structures described in section 2.2.5.

As described in section 2.2.5.6, all the properties of the **Partition** data type are flattened into the **Partition** object. However, the JSON representation has a child **Source** object that can be of type **query** or **calculated** or **none**.

The Source object of type **query** has a **dataSource** reference and a **query** property. The **query** property corresponds to the **QueryDefinition** property that is defined in section 2.2.5.6.

The Source object of type **calculated** has an **expression** property, which also corresponds to the **QueryDefinition** property that is defined in section 2.2.5.6.

If no Source is provided, the partition implicitly has a Source of type **none**.

3.1.5.2.2.1.7 Measure

The JSON schema for the **measure** object is as follows.

```
"measure": {
  "description": "Measure object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "description": {
      "anyOf": [
```

```
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        ]
      },
      "expression": {
        "anyOf": [
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        ]
      },
      "formatString": {
        "type": "string"
      },
      "isHidden": {
        "type": "boolean"
      },
      "isSimpleMeasure": {
        "type": "boolean"
      },
      "displayFolder": {
        "type": "string"
      },
      "kpi": {
        ...
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
```

The properties correspond to the **Measure** object that is defined in section 2.2.5.8. In addition, the JSON-based **measure** object has an optional child **kpi** object.

3.1.5.2.2.1.8 Hierarchy

The JSON schema for the **hierarchy** object is as follows.

```
  "hierarchies": {
    "type": "array",
    "items": {
      "description": "Hierarchy object of Tabular Object Model (TOM)",
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
```

```
        "description": {
          "anyOf": [
            {
              "type": "string"
            },
            {
              "type": "array",
              "items": {
                "type": "string"
              }
            }
          ]
        },
        "isHidden": {
          "type": "boolean"
        },
        "displayFolder": {
          "type": "string"
        },
        "annotations": {
          "type": "array",
          "items": {
             ...
          }
        },
        "levels": {
          "type": "array",
          "items": {
             ...
          }
        }
      },
      "additionalProperties": false
    }
  }
```

The properties correspond to the **Hierarchy** object that is defined in section 2.2.5.9. In addition, the JSON-based **hierarchy** object can have a collection of **level** objects.

3.1.5.2.2.1.9 Level

The JSON schema for the **level** object is as follows.

```
  "level": {
    "description": "Level object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "ordinal": {
        "type": "integer"
      },
      "name": {
        "type": "string"
      },
      "description": {
        "anyOf": [
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        ]
      },
```

```
      "column": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false

}
```

The properties correspond to the **Level** object that is defined in section 2.2.5.10.

3.1.5.2.2.1.10 Annotation

The JSON schema for the **annotation** object is as follows.

```
"annotation": {
  "description": "Annotation object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "value": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    }
  }
```

The properties correspond to the **Annotation** object that is defined in section 2.2.5.11.

3.1.5.2.2.1.11 KPI

The JSON schema for the **kpi** object is as follows.

```
"kpi": {
  "description": "KPI object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "description": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
```

```
      }
    ]
  },
  "targetDescription": {
    "anyOf": [
      {
        "type": "string"
      },
      {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    ]
  },
  "targetExpression": {
    "anyOf": [
      {
        "type": "string"
      },
      {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    ]
  },
  "targetFormatString": {
    "type": "string"
  },
  "statusGraphic": {
    "type": "string"
  },
  "statusDescription": {
    "anyOf": [
      {
        "type": "string"
      },
      {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    ]
  },
  "statusExpression": {
    "anyOf": [
      {
        "type": "string"
      },
      {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    ]
  },
  "trendGraphic": {
    "type": "string"
  },
  "trendDescription": {
    "anyOf": [
      {
        "type": "string"
      },
```

```
          {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        ]
      },
      "trendExpression": {
        "anyOf": [
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        ]
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false

  },
```

The properties correspond to the **KPI** object that is defined in section 2.2.5.12.

3.1.5.2.2.1.12 Culture

The JSON schema for the **culture** object is as follows.

```
  "culture": {
    "description": "Culture object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"


    },
    "linguisticMetadata": {
      "description": "LinguisticMetadata object of Tabular Object Model (TOM)",
      "type": "object",
      "properties": {
        ...
      },
      "additionalProperties": false
    },
    "translations": {


        "type": "object",
        "properties": {
          ...
        },
        "additionalProperties": false
      },
      "annotations": {
        "type": "array",
```

```
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
```

The properties correspond to the **Culture** object that is defined in section 2.2.5.13.

In addition, the JSON-based **culture** object can have two child objects, **linguisticMetadata** object and the **translations** object. The JSON-based **linguisticMetadata** object corresponds to the **LinguisticMetadata** object that is defined in section 2.2.5.15.

The **translations** object has a structure that is different than the **ObjectTranslation** object that is defined in section 2.2.5.14. For more details, see section 3.1.5.2.2.1.13.

3.1.5.2.2.1.13 Translations

The JSON schema for the **translations** object is as follows.

```
  "translations": {
    "type": "object",
    "properties": {
      "model": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "translatedCaption": {
            "type": "string"
          },
          "translatedDescription": {
            "anyOf": [
              {
                "type": "string"
              },
              {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
            ]
          },
          "tables": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "name": {
                  "type": "string"
                },
                "translatedCaption": {
                  "type": "string"
                },
                "translatedDescription": {
                  "anyOf": [
                    {
                      "type": "string"
                    },
                    {
                      "type": "array",
                      "items": {
```

```
          "type": "string"
        }
      }
    ]
  },
  "columns": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "translatedCaption": {
          "type": "string"
        },
        "translatedDescription": {
          "anyOf": [
            {
              "type": "string"
            },
            {
              "type": "array",
              "items": {
                "type": "string"
              }
            }
          ]
        },
        "translatedDisplayFolder": {
          "type": "string"
        }
      },
      "additionalProperties": false
    }
  },
  "measures": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "translatedCaption": {
          "type": "string"
        },
        "translatedDescription": {
          "anyOf": [
            {
              "type": "string"
            },
            {
              "type": "array",
              "items": {
                "type": "string"
              }
            }
          ]
        },
        "translatedDisplayFolder": {
          "type": "string"
        },
        "kpi": {
          "type": "object",
          "properties": {
            "translatedDescription": {
              "anyOf": [
                {
```

```
                    "type": "string"
                  },
                  {
                    "type": "array",
                    "items": {
                      "type": "string"
                    }
                  }
                ]
              }
            },
            "additionalProperties": false
          }
        },
        "additionalProperties": false
      }
    },
    "hierarchies": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "translatedCaption": {
            "type": "string"
          },
          "translatedDescription": {
            "anyOf": [
              {
                "type": "string"
              },
              {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
            ]
          },
          "translatedDisplayFolder": {
            "type": "string"
          },
          "levels": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "name": {
                  "type": "string"
                },
                "translatedCaption": {
                  "type": "string"
                },
                "translatedDescription": {
                  "anyOf": [
                    {
                      "type": "string"
                    },
                    {
                      "type": "array",
                      "items": {
                        "type": "string"
                      }
                    }
                  ]
                }
              },
```

```
                    "additionalProperties": false
                  }
                }
              },
              "additionalProperties": false
            }
          }
        },
        "additionalProperties": false
      }
    },
    "perspectives": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "translatedCaption": {
            "type": "string"
          },
          "translatedDescription": {
            "anyOf": [
              {
                "type": "string"
              },
              {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
            ]
          }
        },
        "additionalProperties": false
      }
    },
    "roles": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "translatedDescription": {
            "anyOf": [
              {
                "type": "string"
              },
              {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
            ]
          }
        },
        "additionalProperties": false
      }
    }
  },
  "additionalProperties": false
```

```
},
```

The underlying Microsoft SQL Server Analysis Services engine APIs express the translations of object properties by using flattened data structures. See the **ObjectTranslation** object that is defined in section 2.2.5.14.

However, the JSON representation of these translations is based on derived classes. As the above schema indicates, the hierarchical structure of the JSON document is used to identify each object. For example, the **column** object appears as a child of the **table** object.

Every object that can be translated or that has descendant objects that can be translated has its own object type, and the specific properties on that object that can be translated have their own member types. For example, **role** objects only allow translation of their description and are therefore defined as an object with the following two properties.

- **name**: Defines the name of the **Role** object that is being translated
- **translatedDescription**: Defines the translation of the description of the **Role** object

The three types of properties that can be translated are defined as follows.

Property	JSON Property Name
Name	translatedCaption
Description	translatedDescription
DisplayFolder	translatedDisplayFolder

3.1.5.2.2.1.14 LinguisticMetadata

The JSON schema for the **linguisticMetadata** object is as follows.

```
"linguisticMetadata": {
  "description": "LinguisticMetadata object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "content": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    },
    "annotations": {
      "type": "array",
      "items": {
        ...
      }
    }
  },
  "additionalProperties": false
},
```

The properties correspond to the **LinguisticMetadata** object that is defined in section 2.2.5.15.

3.1.5.2.2.1.15 Perspective

The JSON schema for the **perspective** object is as follows.

```
"perspectives": {
  "description": "Perspective object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "description": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    },
    "annotations": {
      "type": "array",
      "items": {
        ...
      }
    },
    "tables": {
      "type": "array",
      "items": {
        ...
      }
    }
  },
  "additionalProperties": false
}
```

The properties correspond to the **Perspective** object that is defined in section 2.2.5.16.

The **perspective** object can have a collection of **perspectiveTable** objects.

3.1.5.2.2.1.16 PerspectiveTable

The JSON schema for the **perspectiveTable** object is as follows.

```
"tables": {
  "type": "array",
  "items": {
    "description": "PerspectiveTable object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "includeAll": {
        "type": "boolean"
      },
      "annotations": {
        "type": "array",
        "items": {
```

```
            ...
          }
        },
        "columns": {
          "type": "array",
          "items": {
            ...
          }
        },
        "measures": {
          "type": "array",
          "items": {
            ...
          }
        },
        "hierarchies": {
          "type": "array",
          "items": {
            ...
          }
        }
      },
      "additionalProperties": false
    }
  }
```

The properties correspond to the **PerspectiveTable** object that is defined in section 2.2.11.17.

The **perspectiveTable** object can have collections of **perspectiveColumn**, **perspectiveHierarchy**, and **perspectiveMeasure** objects, which are defined in sections 3.1.5.2.2.1.17, 3.1.5.2.2.1.18, and 3.1.5.2.2.1.19, respectively.

3.1.5.2.2.1.17 PerspectiveColumn

The JSON schema for the **perspectiveColumn** object is as follows.

```
  "columns": {
    "type": "array",
    "items": {
      "description": "PerspectiveColumn object of Tabular Object Model (TOM)",
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "annotations": {
          "type": "array",
          "items": {
            ...
          }
        }
      },
      "additionalProperties": false
    }
  },
```

The properties correspond to the **PerspectiveColumn** object that is defined in section 2.2.5.18.

3.1.5.2.2.1.18 PerspectiveHierarchy

The JSON schema for the **perspectiveHierarchy** object is as follows.

```
"hierarchies": {
  "type": "array",
  "items": {
    "description": "PerspectiveHierarchy object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
}
```

The properties correspond to the **PerspectiveHierarchy** object that is defined in section 2.2.5.19.

3.1.5.2.2.1.19 PerspectiveMeasure

The JSON schema for the **perspectiveMeasure** object is as follows.

```
"measures": {
  "type": "array",
  "items": {
    "description": "PerspectiveMeasure object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
},
```

The properties correspond to the **PerspectiveMeasure** object that is defined in section 2.2.5.20.

3.1.5.2.2.1.20 Role

The JSON schema for the **role** object is as follows.

```
"role": {
  "description": "ModelRole object of Tabular Object Model (TOM)",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "description": {
      "anyOf": [
        {
          "type": "string"
```

```
        },
        {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      ]
    },
    "modelPermission": {
      "enum": [
        "none",
        "read",
        "readRefresh",
        "refresh",
        "administrator"
      ]
    },
    "annotations": {
      "type": "array",
      "items": {
        ...
      }
    },
    "members": {
      "type": "array",
      "items": {
        ...
      }
    },
    "tablePermissions": {
      "type": "array",
      "items": {
        ...
      }
    }
  },
  "additionalProperties": false
}
```

The properties correspond to the **Role** object that is defined in section 2.2.5.21.

The **role** object has two child collections. The **members** child collection is of type **RoleMembership**, and the **tablePermissions** child collection is of type **TablePermissions**.

3.1.5.2.2.1.21 RoleMembership

The JSON schema for the **member** object is as follows.

```
"anyOf": [
  {
    "description": "WindowsModelRoleMember object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "memberName": {
        "type": "string"
      },
      "memberId": {
        "type": "string"
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
```

```
      },
      "additionalProperties": false
    },
    {
      "description": "ExternalModelRoleMember object of Tabular Object Model (TOM)",
      "type": "object",
      "properties": {
        "memberName": {
          "type": "string"
        },
        "memberId": {
          "type": "string"
        },
        "identityProvider": {
          "type": "string"
        },
        "memberType": {
          "enum": [
            "auto",
            "user",
            "group"
          ]
        },
        "annotations": {
          "type": "array",
          "items": {
            ...
          }
        }
      },
      "additionalProperties": false
    }
  ]
```

The properties correspond to the **RoleMembership** object that is defined in section 2.2.5.22.

The JSON representation of a **RoleMembership** object has the following two derived classes.

- Windows
- External

The difference between the two derived types is that the External **member** object has an **identityProvider** property. If that property is present, then the **memberType** property can also be present.

3.1.5.2.2.1.22 TablePermission

The JSON schema for the **tablePermission** object is as follows.

```
  "tablePermissions": {
    "description": "TablePermission object of Tabular Object Model (TOM)",
    "type": "object",
    "properties": {
      "name": {
        "type": "string"
      },
      "filterExpression": {
        "anyOf": [
          {
            "type": "string"
          },
          {
            "type": "array",
            "items": {
```

```
              "type": "string"
            }
          }
        ]
      },
      "annotations": {
        "type": "array",
        "items": {
          ...
        }
      }
    },
    "additionalProperties": false
  }
```

The properties correspond to the **TablePermission** object that is defined in section 2.2.5.23. The **name** property refers to the name of the **Table** object on which the permission applies.

3.1.5.2.2.2 Create Command

The JSON **create** command creates the specified object and all the descendant objects that are specified. If the object already exists, the command raises an error.

3.1.5.2.2.2.1 Request

The JSON schema for the **create** command is as follows.

```
{
    "type": "object",
    "description": "Create command of Analysis Services JSON API",
    "properties": {
        "create": {
            "description": "Parameters of Create command of Analysis Services JSON API",
            "anyOf": [
            {
                "description": "Create command for Database object",
                "type": "object",
                "properties": {
                    "database": {
                        ...
                    }
                },
                "additionalProperties": false
            },
            {
                "description": "Create command for DataSource object",
                "type": "object",
                "properties": {
                    "parentObject": {
                        "description": "Path for object Database",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "dataSource": {
                        ...
                    }
                },
                "additionalProperties": false
            },
            {
```

```
                "description": "Create command for Table object",
                "type": "object",
                "properties": {
                    "parentObject": {
                        "description": "Path for object Database",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "table": {
                        ...
                    },
                    "additionalProperties": false
                }
            },
            {
                "description": "Create command for Partition object",
                "type": "object",
                "properties": {
                    "parentObject": {
                        "description": "Path for object Table",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            },
                            "table": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "partition": {
                        ...
                    }
                },
                "additionalProperties": false
            },
            {
                "description": "Create command for Role object",
                "type": "object",
                "properties": {
                    "parentObject": {
                        "description": "Path for object Database",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "role": {
                        ...
                    }
                },
                "additionalProperties": false
            }
            ]
        }
    },
    "additionalProperties": false
},
```

This schema indicates that the following objects can be created.

- Database
- DataSource
- Table
- Partition
- Role

Except for the **Database** object, the object being created is defined to be a child of a specified **parentObject**. The parent of the **Database** object is always the **Server** object, as described in [MS-SSAS].

3.1.5.2.2.2.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.3 CreateOrReplace Command

The JSON **createOrReplace** command creates the specified object and all the descendant objects that are specified. If the object already exists, the command replaces the object with the new definition.

3.1.5.2.2.3.1 Request

The JSON schema for the **createOrReplace** command is as follows.

```
{
    "type": "object",
    "description": "CreateOrReplace command of Analysis Services JSON API",
    "properties": {
        "createOrReplace": {
            "description": "Parameters of CreateOrReplace command of Analysis Services JSON
API",
            "anyOf": [
              {
                  "description": "CreateOrReplace command for Database object",
                  "type": "object",
                  "properties": {
                      "object": {
                          "description": "Path for object Database",
                          "type": "object",
                          "properties": {
                              "database": {
                                  "type": "string"
                              }
                          },
                          "additionalProperties": false
                      },
                      "database": {
                          ...
                      }
                  },
                  "additionalProperties": false
              },
              {
                  "description": "CreateOrReplace command for DataSource object",
                  "type": "object",
```

```
            "properties": {
                "object": {
                    "description": "Path for object DataSource",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        },
                        "dataSource": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                },
                "dataSource": {
                    ...
                }
            },
            "additionalProperties": false
        },
        {
            "description": "CreateOrReplace command for Table object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object Table",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        },
                        "table": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                },
                "table": {
                    ...
                }
            },
            "additionalProperties": false
        },
        {
            "description": "CreateOrReplace command for Partition object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object Partition",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        },
                        "table": {
                            "type": "string"
                        },
                        "partition": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                },
                "partition": {
                    ...
                }
            },
            "additionalProperties": false
        },
```

```
                {
                    "description": "CreateOrReplace command for Role object",
                    "type": "object",
                    "properties": {
                        "object": {
                            "description": "Path for object Role",
                            "type": "object",
                            "properties": {
                                "database": {
                                    "type": "string"
                                },
                                "role": {
                                    "type": "string"
                                }
                            },
                            "additionalProperties": false
                        },
                        "role": {
                            ...
                        }
                    },
                    "additionalProperties": false
                }
            ]
        }
    },
    "additionalProperties": false
},
```

This schema indicates that the following objects can be created or replaced.

- Database
- DataSource
- Table
- Partition
- Role

Except for the **Database** object, the object being created or replaced is defined to be a child of a specified **parentObject**. The parent of the **Database** object is always the **Server** object, as described in [MS-SSAS].

3.1.5.2.2.3.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.4 Alter Command

The JSON **alter** command alters the specified object. If the object does not exist, the command raises an error.

This command accepts only the object being altered. It does not accept child collections.

3.1.5.2.2.4.1 Request

The JSON schema for the **alter** command is as follows.

```
{
    "type": "object",
    "description": "Alter command of Analysis Services JSON API",
    "properties": {
        "alter": {
            "description": "Parameters of Alter command of Analysis Services JSON API",
            "anyOf": [
              {
                  "description": "Alter command for Database object",
                  "type": "object",
                  "properties": {
                      "object": {
                          "description": "Path for object Database",
                          "type": "object",
                          "properties": {
                              "database": {
                                  "type": "string"
                              }
                          },
                          "additionalProperties": false
                      },
                      "database": {
                          ...
                      }
                  },
                  "additionalProperties": false
              },
              {
                  "description": "Alter command for DataSource object",
                  "type": "object",
                  "properties": {
                      "object": {
                          "description": "Path for object DataSource",
                          "type": "object",
                          "properties": {
                              "database": {
                                  "type": "string"
                              },
                              "dataSource": {
                                  "type": "string"
                              }
                          },
                          "additionalProperties": false
                      },
                      "dataSource": {
                          ...
                      }
                  },
                  "additionalProperties": false
              },
              {
                  "description": "Alter command for Table object",
                  "type": "object",
                  "properties": {
                      "object": {
                          "description": "Path for object Table",
                          "type": "object",
                          "properties": {
                              "database": {
                                  "type": "string"
                              },
                              "table": {
                                  "type": "string"
                              }
                          },
                          "additionalProperties": false
                      },
                      "table": {
                          ...
```

```
                    }
                },
                "additionalProperties": false
            },
            {
                "description": "Alter command for Partition object",
                "type": "object",
                "properties": {
                    "object": {
                        "description": "Path for object Partition",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            },
                            "table": {
                                "type": "string"
                            },
                            "partition": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "partition": {
                        ...
                    }
                },
                "additionalProperties": false
            },
            {
                "description": "Alter command for Role object",
                "type": "object",
                "properties": {
                    "object": {
                        "description": "Path for object Role",
                        "type": "object",
                        "properties": {
                            "database": {
                                "type": "string"
                            },
                            "role": {
                                "type": "string"
                            }
                        },
                        "additionalProperties": false
                    },
                    "role": {
                        ...
                    }
                },
                "additionalProperties": false
            }
        ]
    }
  },
  "additionalProperties": false
},
```

This schema indicates that the following objects can be altered.

- Database
- DataSource
- Table

- Partition
- Role

The object being altered is specified by using the **object** path.

3.1.5.2.2.4.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.5 Delete Command

The JSON **delete** command deletes the specified object and all its child objects and collections. If the object does not exist, the command raises an error.

3.1.5.2.2.5.1 Request

The JSON schema for the **delete** command is as follows.

```
{
    "type": "object",
    "description": "Delete command of Analysis Services JSON API",
    "properties": {
    "delete": {
        "description": "Parameters of Delete command of Analysis Services JSON API",
        "anyOf": [
        {
            "description": "Delete command for Database object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object Database",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        {
            "description": "Delete command for DataSource object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object DataSource",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                         },
                            "dataSource": {
                                "type": "string"
                            }
                    },
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
```

```
        },
        {
            "description": "Delete command for Table object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object Table",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        },
                        "table": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        {
            "description": "Delete command for Partition object",
            "type": "object",
            "properties": {
                "object": {
                    "description": "Path for object Partition",
                    "type": "object",
                    "properties": {
                        "database": {
                            "type": "string"
                        },
                        "table": {
                            "type": "string"
                        },
                        "partition": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        {
            "description": "Delete command for Role object",
            "type": "object",
            "properties": {
                "object": {
                        "description": "Path for object Role",
                        "type": "object",
                        "properties": {
                        "database": {
                            "type": "string"
                        },
                        "role": {
                            "type": "string"
                        }
                    },
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        }
        ]
    }
    },
    "additionalProperties": false
```

```
},
```

This schema indicates that the following objects can be deleted.

- Database
- DataSource
- Table
- Partition
- Role

The object being deleted is specified by using the **object** path.

3.1.5.2.2.5.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.6 Refresh Command

The JSON **refresh** command refreshes the contents of the specified object and propagates changes to objects that depend on the affected objects. If the object does not exist, the command raises an error.

3.1.5.2.2.6.1 Request

The JSON schema for the **refresh** command is as follows.

```
{
    "type": "object",
    "description": "Refresh command of Analysis Services JSON API",
    "properties": {
    "refresh": {
        "description": "Parameters of Refresh command of Analysis Services JSON API",
        "properties": {
        "type": {
            "enum": [
            "full",
            "clearValues",
            "calculate",
            "dataOnly",
            "automatic",
            "add",
            "defragment"
            ]
        },
        "objects": {
            "type": "array",
            "items": {
            "anyOf": [
                {
                "description": "Path for object Database",
                "type": "object",
                "properties": {
                    "database": {
                    "type": "string"
                    }
                },
```

```
                    "additionalProperties": false
                    },
                    {
                    "description": "Path for object Table",
                    "type": "object",
                    "properties": {
                        "database": {
                        "type": "string"
                        },
                        "table": {
                        "type": "string"
                        }
                    },
                    "additionalProperties": false
                    },
                    {
                    "description": "Path for object Partition",
                    "type": "object",
                    "properties": {
                        "database": {
                        "type": "string"
                        },
                        "table": {
                        "type": "string"
                        },
                        "partition": {
                        "type": "string"
                        }
                    },
                    "additionalProperties": false
                    }
                ]
                }
            }
            }
        }
        },
        "additionalProperties": false
    },
```

The different types of refresh operations are as follows.

Refresh Type	Applies To	Description
full	Database Table Partition	For all partitions in the specified partition, table, or database, refresh data and recalculate all dependents. For a calculation partition, recalculate the partition and all its dependents.
clearValues	Database Table Partition	Clear values in this object and all its dependents.
calculate	Database Table Partition	Recalculate this object and all its dependents, but only if needed. This value does not force recalculation, except for volatile formulas.
dataOnly	Database Table Partition	Refresh data in this object and clear all dependents.
automatic	Database Table	If the object needs to be refreshed and recalculated, refresh and recalculate the object and all its dependents. Applies if the partition is in a state other than Ready (see section 2.2.5.6).

Refresh Type	Applies To	Description
	Partition	
add	Partition	Append data to this partition and recalculate all dependents. This command is valid only for regular partitions and not for calculation partitions.
defragment	Database Table	Defragment the data in the specified table. As data is added to or removed from a table, the dictionaries of each column can become polluted with values that no longer exist in the actual column values. The **defragment** option will clean up the values in the dictionaries that are no longer used.

For details on specific behaviors of these refresh types for each object, see section 3.1.5.2.1.5.

3.1.5.2.2.6.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.7 Sequence Command

The JSON **sequence** command enables execution of multiple JSON commands in one request. The commands are executed in a logically sequential manner. In addition, the analysis server can optimize the commands by automatically parallelizing some of them together.

3.1.5.2.2.7.1 Request

The JSON schema for the **sequence** command is as follows.

```
{
    "type": "object",
    "description": "Sequence command of Analysis Services JSON API",
    "properties": {
        "sequence": {
            "description": "Parameters of Sequence command of Analysis Services JSON API",
            "properties": {
                "maxParallelism": {
                    "type": "integer"
                },
                "operations": {
                    "type": "array",
                    "items": {
                        "anyOf": [
                        {
                            "type": "object",
                            "description": "Create command of Analysis Services JSON API",
                            "properties": {
                                "create": {
                                    "description": "Parameters of Create command of Analysis
Services JSON API",
                                    ...
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
```

```
                            "description": "CreateOrReplace command of Analysis Services JSON
API",
                            "properties": {
                                "createOrReplace": {
                                    "description": "Parameters of CreateOrReplace command of
Analysis Services JSON API",
                                    ...
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Alter command of Analysis Services JSON API",
                            "properties": {
                                "alter": {
                                    "description": "Parameters of Alter command of Analysis
Services JSON API",
                                    ...
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Delete command of Analysis Services JSON API",
                            "properties": {
                                "delete": {
                                    "description": "Parameters of Delete command of Analysis
Services JSON API",
                                    ...
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Refresh command of Analysis Services JSON API",
                            "properties": {
                                "refresh": {
                                    "description": "Parameters of Refresh command of Analysis
Services JSON API",
                                    "properties":
                                        ...
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Backup command of Analysis Services JSON API",
                            "properties": {
                                "backup": {
                                    "description": "Parameters of Backup command of Analysis
Services JSON API",
                                    "properties":
                                        ...
                                    "additionalProperties": false
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Restore command of Analysis Services JSON API",
                            "properties": {
                                "restore": {
                                    "description": "Parameters of Restore command of Analysis
Services JSON API",
```

```
                                    "properties":
                                        ...
                                    "additionalProperties": false
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Attach command of Analysis Services JSON API",
                            "properties": {
                                "attach": {
                                    "description": "Parameters of Attach command of Analysis
Services JSON API",
                                    "properties":
                                        ...
                                    "additionalProperties": false
                                }
                            },
                            "additionalProperties": false
                        },
                        {
                            "type": "object",
                            "description": "Detach command of Analysis Services JSON API",
                            "properties": {
                                "detach": {
                                    "description": "Parameters of Detach command of Analysis
Services JSON API",
                                    "properties":
                                        ...
                                    "additionalProperties": false
                                }
                            },
                            "additionalProperties": false
                        }
                        ]
                    }
                }
            }
        }
    },
    "additionalProperties": false
}
```

The following commands can be specified inside the **sequence** command.

- Create
- CreateOrReplace
- Alter
- Delete
- Refresh
- Backup
- Restore
- Attach
- Detach

The **sequence** command also accepts the maxParallelism integer property, which specifies the upper bound for the server to place on the parallelism of the sequence command operations. The server will attempt to limit the tasks that are executed concurrently, but the limit is not guaranteed to be enforced.

3.1.5.2.2.7.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.8 Backup Command

The JSON **backup** command creates a backup of the specified database. If the object does not exist, the command raises an error.

3.1.5.2.2.8.1 Request

The JSON schema for the **backup** command is as follows.

```
{
    "type": "object",
    "description": "Backup command of Analysis Services JSON API",
    "properties": {
    "backup": {
        "description": "Parameters of Backup command of Analysis Services JSON API",
        "properties": {
        "database": {
            "type": "string"
        },
        "file": {
            "type": "string"
        },
        "password": {
            "type": "string"
        },
        "allowOverwrite": {
            "type": "boolean"
        },
        "applyCompression": {
            "type": "boolean"
        }
        },
        "additionalProperties": false
    }
    },
    "additionalProperties": false
},
```

The properties accepted by the JSON **backup** command are as follows. They are similar to the properties accepted by the XMLA **Backup** command described in [MS-SSAS] section 3.1.4.3.2.1.1.17.

Property	Default value	Description
database	[Required]	The name of the database object to be backed up.
file	[Required]	The backup file name/path.
password	Empty	The password to use for encrypting the backup file.

Property	Default value	Description
allowOverwrite	False	A Boolean that, when true, indicates that a backup file that already exists will be overwritten; otherwise false.
applyCompression	True	A Boolean that, when true, indicates that backup files are compressed; otherwise false.

3.1.5.2.2.8.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.9 Restore Command

The JSON **restore** command restores the specified database from a backup file.

3.1.5.2.2.9.1 Request

The JSON schema for the **restore** command is as follows.

```
{
    "type": "object",
    "description": "Restore command of Analysis Services JSON API",
    "properties": {
    "restore": {
        "description": "Parameters of Restore command of Analysis Services JSON API",
        "properties": {
        "database": {
            "type": "string"
        },
        "file": {
            "type": "string"
        },
        "password": {
            "type": "string"
        },
        "dbStorageLocation": {
            "type": "string"
        },
        "allowOverwrite": {
            "type": "boolean"
        },
        "readWriteMode": {
            "enum": [
            "readWrite",
            "readOnly",
            "readOnlyExclusive"
            ]
        }
        },
        "additionalProperties": false
    }
    },
    "additionalProperties": false
},
```

The properties accepted by the JSON **restore** command are as follows. They are similar to the properties accepted by the XMLA **Restore** command described in [MS-SSAS] section 3.1.4.3.2.1.1.18.

Property	Default value	Description
database	[Required]	The name of the database object to be restored.
file	[Required]	The backup file name/path.
password	Empty	The password to use to decrypt the backup file.
dbStorageLocation	Empty	Storage location for the restored database.
allowOverwrite	False	A Boolean that, when true, indicates that a backup file that already exists will be overwritten; otherwise false.
readWriteMode	readWrite	An enumeration value that indicates the access modes allowed to the database. The enumeration values are as follows. ▪ **readWrite** – Read-write access is allowed. ▪ **readOnly** – Read-only access is allowed. ▪ **readOnlyExclusive** – Read-only exclusive access is allowed.

3.1.5.2.2.9.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.10 Attach Command

The JSON **attach** command attaches a detached database.

3.1.5.2.2.10.1 Request

The JSON schema for the **attach** command is as follows.

```
{
    "type": "object",
    "description": "Attach command of Analysis Services JSON API",
    "properties": {
    "attach": {
        "description": "Parameters of Attach command of Analysis Services JSON API",
        "properties": {
        "folder": {
            "type": "string"
        },
        "password": {
            "type": "string"
        },
        "readWriteMode": {
            "enum": [
            "readWrite",
            "readOnly",
            "readOnlyExclusive"
            ]
        }
        },
        "additionalProperties": false
```

```
        }
        },
        "additionalProperties": false
    },
```

The properties accepted by the JSON **attach** command are as follows. They are similar to the properties accepted by the XMLA **Attach** command described in [MS-SSAS] section 3.1.4.3.2.1.1.20.

Property	Default value	Description
database	[Required]	The name of the database object to be attached.
folder	[Required]	The folder that contains the detached database.
password	Empty	The password to use to decrypt secrets in the detached database.
readWriteMode	readWrite	An enumeration value that indicates the access modes allowed to the database. The enumeration values are as follows. ▪ **readWrite** – Read-write access is allowed. ▪ **readOnly** – Read-only access is allowed. ▪ **readOnlyExclusive** – Read-only exclusive access is allowed.

3.1.5.2.2.10.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.11 Detach Command

The JSON **detach** command detaches the specified database from the server. If the database does not exist, the command raises an error.

3.1.5.2.2.11.1 Request

The JSON schema for the **detach** command is as follows.

```
{
    "type": "object",
    "description": "Detach command of Analysis Services JSON API",
    "properties": {
    "detach": {
        "description": "Parameters of Detach command of Analysis Services JSON API",
        "properties": {
        "database": {
            "type": "string"
        },
        "password": {
            "type": "string"
        }
        },
        "additionalProperties": false
    }
    },
```

```
    "additionalProperties": false
}
```

The properties accepted by the JSON **detach** command are as follows. They are similar to the properties accepted by the XMLA **Detach** command described in [MS-SSAS] section 3.1.4.3.2.1.1.21.

Property	Default value	Description
database	[Required]	The name of the database object to be detached.
password	Empty	The password to use to encrypt secrets in the detached database.

3.1.5.2.2.11.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.12 Synchronize Command

The JSON **synchronize** command synchronizes the database from a source server.

3.1.5.2.2.12.1 Request

The JSON schema for the **synchronize** command is as follows.

```
{
  "type": "object",
  "description": "Synchronize command of Analysis Services JSON API",
  "properties": {
    "synchronize": {
      "description": "Parameters of Synchronize command of Analysis Services JSON API",
      "properties": {
        "database": {
          "type": "string"
        },
        "source": {
          "type": "string"
        },
        "synchronizeSecurity": {
          "enum": [
            "copyAll",
            "skipMembership",
            "ignoreSecurity"
          ]
        },
        "applyCompression": {
          "type": "boolean"
        }
      }
    }
  },
  "additionalProperties": false
},
```

The properties accepted by the JSON **synchronize** command are as follows. They are similar to the properties accepted by the XMLA **Synchronize** command described in [MS-SSAS] section 3.1.4.3.2.1.1.19.

Property	Default value	Description
database		The name of the database object.
source		The connection string to use to connect to the source server.
synchronizeSecurity	skipMembership	An enumeration value that specifies how to restore security definitions, including roles and permissions.
applyCompression	True	A Boolean that, when true, indicates that compression will be applied during the synchronization operation; otherwise false.

3.1.5.2.2.12.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.5.2.2.13 MergePartitions Command

The JSON **mergePartitions** command merges the partitions specified in the source array into the target partition.

3.1.5.2.2.13.1 Request

The JSON schema for the **mergePartitions** command is as follows.

```
{
  "type": "object",
  "description": "MergePartitions command of Analysis Services JSON API",
  "properties": {
    "mergePartitions": {
      "description": "Parameters of MergePartitions command of Analysis Services JSON API",
      "properties": {
        "target": {
          "description": "Path for object Partition",
          "type": "object",
          "properties": {
            "database": {
              "type": "string"
            },
            "table": {
              "type": "string"
            },
            "partition": {
              "type": "string"
            }
          },
          "additionalProperties": false
        },
        "sources": {
          "type": "array",
          "items": {
            "type": "string"
          },
          "minItems": 1,
```

```
            "uniqueItems": true
          }
        }
      }
    },
    "additionalProperties": false
  }
```

Property	Default value	Description
database		The name of the database object.
target.table		The name of the table that contains the partitions that are being merged.
target.partition		The name of the target partition.
sources		An array of strings that contains the names of the source partitions.

3.1.5.2.2.13.2 Response

If the request fails, an XMLA exception is returned in the response (see [MS-SSAS] section 2.2.4.1.5.1).

Otherwise, the response is an empty result (see [MS-SSAS] section 2.2.4.1.2).

3.1.6 Timer Events

None. All protocol requests are initiated by the client.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Refresh Tabular Metadata (XMLA)

In this example, the client sends an **XMLA Tabular Refresh Command** to the server with the **ReturnAffectedObjects** property.

4.1.1 Client Sends Request

The client sends the following request:

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Header>
    <Session xmlns="urn:schemas-microsoft-com:xml-analysis"
    SessionId="34B67555-85B9-46CE-8803-4BEC7D6AEE13" />
  </Header>
  <Body>
    <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
      <Command>
        <Refresh xmlns="http://schemas.microsoft.com/analysisservices/2014/engine">
          <DatabaseID>PushedDataDB</DatabaseID>
          <PushedData>InputRowset</PushedData>
          <EndOfData>EndOfInputRowset</EndOfData>
          <Partitions>
            <!-- Begin Refresh Partition schema -->
            <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-
microsoft-com:xml-sql">
              <xs:element>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="row" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:complexType name="row">
                <xs:sequence>
                  <xs:element name="ID" type="xs:unsignedLong" sql:field="ID" minOccurs="0"
/>
                  <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table"
minOccurs="0" />
                  <xs:element name="ID.Partition" type="xs:string" sql:field="ID.Partition"
minOccurs="0" />
                  <xs:element name="RefreshType" type="xs:long" sql:field="RefreshType"
minOccurs="0" />
                </xs:sequence>
              </xs:complexType>
            </xs:schema>
            <!-- End Refresh Partition schema -->
            <row xmlns="urn:schemas-microsoft-com:xml-analysis:rowset">
              <ID>13</ID>
              <RefreshType>4</RefreshType>
            </row>
          </Partitions>
          <Bindings>
            <Binding>
              <ObjectID>13</ObjectID>
              <Columns>
                <!-- Begin Bindings Column schema -->
                <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:sql="urn:schemas-microsoft-com:xml-sql">
                  <xs:element>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element type="row" />
                      </xs:sequence>
```

```
                    </xs:complexType>
                  </xs:element>
                  <xs:complexType name="row">
                    <xs:sequence>
                      <xs:element name="ID" type="xs:unsignedLong" sql:field="ID"
minOccurs="0" />
                      <xs:element name="ID.Table" type="xs:string" sql:field="ID.Table"
minOccurs="0" />
                      <xs:element name="ID.Column" type="xs:string" sql:field="ID.Column"
minOccurs="0" />
                      <xs:element name="SourceColumn" type="xs:string"
sql:field="SourceColumn" minOccurs="0" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:schema>
                <!-- End Bindings Column schema -->
                <row xmlns="urn:schemas-microsoft-com:xml-analysis:rowset">
                  <ID>14</ID>
                  <SourceColumn>a</SourceColumn>
                </row>
                <row xmlns="urn:schemas-microsoft-com:xml-analysis:rowset">
                  <ID>15</ID>
                  <SourceColumn>b</SourceColumn>
                </row>
              </Columns>
            </Binding>
          </Bindings>
        </Refresh>
      </Command>
      <Properties>
        <PropertyList>
          <ReturnAffectedObjects>2</ReturnAffectedObjects>
        </PropertyList>
      </Properties>
      <Parameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <Parameter>
          <Name>InputRowset</Name>
          <Value xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:simpleType name="uuid">
                <xsd:restriction base="xsd:string">
                  <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:complexType name="xmlDocument">
                <xsd:sequence>
                  <xsd:any />
                </xsd:sequence>
              </xsd:complexType>
              <xsd:complexType name="row">
                <xsd:sequence>
                  <xsd:element sql:field="a" name="a" type="xsd:int" minOccurs="0" />
                  <xsd:element sql:field="b" name="b" type="xsd:string" minOccurs="0" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
```

```
              <a>10</a>
              <b>b10</b>
            </row>
            <row>
              <a>20</a>
              <b>b20</b>
            </row>
          </Value>
        </Parameter>
        <Parameter>
          <Name>InputRowset</Name>
          <Value xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:simpleType name="uuid">
                <xsd:restriction base="xsd:string">
                  <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:complexType name="xmlDocument">
                <xsd:sequence>
                  <xsd:any />
                </xsd:sequence>
              </xsd:complexType>
              <xsd:complexType name="row">
                <xsd:sequence>
                  <xsd:element sql:field="a" name="a" type="xsd:int" minOccurs="0" />
                  <xsd:element sql:field="b" name="b" type="xsd:string" minOccurs="0" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <a>1</a>
              <b>b1</b>
            </row>
            <row>
              <a>2</a>
              <b>b2</b>
            </row>
          </Value>
        </Parameter>
        <Parameter>
          <Name>EndOfInputRowset</Name>
          <Value xsi:type="xsd:boolean">true</Value>
        </Parameter>
      </Parameters>
    </Execute>
  </Body>
</Envelope>
```

4.1.2 Server Response

The server responds with the results of the **XMLA Tabular Refresh Command**:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```
    <ExecuteResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <return>
        <AffectedObjects xmlns="http://schemas.microsoft.com/analysisservices/2003/xmla-
multipleresults" name="PushedDataDB" BaseVersion="0" CurrentVersion="5">
          <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla" name="Model">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:simpleType name="uuid">
                <xsd:restriction base="xsd:string">
                  <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:complexType name="xmlDocument">
                <xsd:sequence>
                  <xsd:any />
                </xsd:sequence>
              </xsd:complexType>
              <xsd:complexType name="row">
                <xsd:sequence>
                  <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0"
/>
                  <xsd:element sql:field="Name" name="Name" type="xsd:string" minOccurs="0"
/>
                  <xsd:element sql:field="Description" name="Description" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="StorageLocation" name="StorageLocation"
type="xsd:string" minOccurs="0" />
                  <xsd:element sql:field="DefaultMode" name="DefaultMode" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="DefaultDataView" name="DefaultDataView"
type="xsd:long" minOccurs="0" />
                  <xsd:element sql:field="Culture" name="Culture" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="Collation" name="Collation" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="ModifiedTime" name="ModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="Version" name="Version" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <ID>1</ID>
              <Name>Model</Name>
              <DefaultMode>0</DefaultMode>
              <DefaultDataView>0</DefaultDataView>
              <Culture>en-US</Culture>
              <ModifiedTime>2015-09-30T03:25:33.133333</ModifiedTime>
              <StructureModifiedTime>2015-09-30T03:25:33.306667</StructureModifiedTime>
              <Version>5</Version>
            </row>
          </root>
          <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla" name="Table">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:simpleType name="uuid">
                <xsd:restriction base="xsd:string">
                  <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:complexType name="xmlDocument">
                <xsd:sequence>
                  <xsd:any />
                </xsd:sequence>
              </xsd:complexType>
              <xsd:complexType name="row">
                <xsd:sequence>
                  <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0"
/>
                  <xsd:element sql:field="ModelID" name="ModelID" type="xsd:unsignedLong"
minOccurs="0" />
                  <xsd:element sql:field="Name" name="Name" type="xsd:string" minOccurs="0"
/>
                  <xsd:element sql:field="DataCategory" name="DataCategory" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="Description" name="Description" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="IsHidden" name="IsHidden" type="xsd:boolean"
minOccurs="0" />
                  <xsd:element sql:field="TableStorageID" name="TableStorageID"
type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element sql:field="ModifiedTime" name="ModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="SystemFlags" name="SystemFlags" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <ID>10</ID>
              <ModelID>1</ModelID>
              <Name>PastedTable</Name>
              <IsHidden>false</IsHidden>
              <TableStorageID>18</TableStorageID>
              <ModifiedTime>2015-09-30T03:25:33.306667</ModifiedTime>
              <StructureModifiedTime>2015-09-30T03:25:33.306667</StructureModifiedTime>
              <SystemFlags>0</SystemFlags>
            </row>
          </root>
          <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla" name="Column">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
```

```
              </xsd:complexType>
            </xsd:element>
            <xsd:simpleType name="uuid">
              <xsd:restriction base="xsd:string">
                <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
              </xsd:restriction>
            </xsd:simpleType>
            <xsd:complexType name="xmlDocument">
              <xsd:sequence>
                <xsd:any />
              </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="row">
              <xsd:sequence>
                <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0"
/>
                <xsd:element sql:field="TableID" name="TableID" type="xsd:unsignedLong"
minOccurs="0" />
                <xsd:element sql:field="ExplicitName" name="ExplicitName" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="InferredName" name="InferredName" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="ExplicitDataType" name="ExplicitDataType"
type="xsd:long" minOccurs="0" />
                <xsd:element sql:field="InferredDataType" name="InferredDataType"
type="xsd:long" minOccurs="0" />
                <xsd:element sql:field="DataCategory" name="DataCategory" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="Description" name="Description" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="IsHidden" name="IsHidden" type="xsd:boolean"
minOccurs="0" />
                <xsd:element sql:field="State" name="State" type="xsd:long" minOccurs="0"
/>
                <xsd:element sql:field="IsUnique" name="IsUnique" type="xsd:boolean"
minOccurs="0" />
                <xsd:element sql:field="IsKey" name="IsKey" type="xsd:boolean"
minOccurs="0" />
                <xsd:element sql:field="IsNullable" name="IsNullable" type="xsd:boolean"
minOccurs="0" />
                <xsd:element sql:field="Alignment" name="Alignment" type="xsd:long"
minOccurs="0" />
                <xsd:element sql:field="TableDetailPosition" name="TableDetailPosition"
type="xsd:int" minOccurs="0" />
                <xsd:element sql:field="IsDefaultLabel" name="IsDefaultLabel"
type="xsd:boolean" minOccurs="0" />
                <xsd:element sql:field="IsDefaultImage" name="IsDefaultImage"
type="xsd:boolean" minOccurs="0" />
                <xsd:element sql:field="SummarizeBy" name="SummarizeBy" type="xsd:long"
minOccurs="0" />
                <xsd:element sql:field="ColumnStorageID" name="ColumnStorageID"
type="xsd:unsignedLong" minOccurs="0" />
                <xsd:element sql:field="Type" name="Type" type="xsd:long" minOccurs="0" />
                <xsd:element sql:field="SourceColumn" name="SourceColumn" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="ColumnOriginID" name="ColumnOriginID"
type="xsd:unsignedLong" minOccurs="0" />
                <xsd:element sql:field="Expression" name="Expression" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="FormatString" name="FormatString" type="xsd:string"
minOccurs="0" />
                <xsd:element sql:field="IsAvailableInMDX" name="IsAvailableInMDX"
type="xsd:boolean" minOccurs="0" />
                <xsd:element sql:field="SortByColumnID" name="SortByColumnID"
type="xsd:unsignedLong" minOccurs="0" />
                <xsd:element sql:field="AttributeHierarchyID" name="AttributeHierarchyID"
type="xsd:unsignedLong" minOccurs="0" />
```

```
                  <xsd:element sql:field="ModifiedTime" name="ModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="StructureModifiedTime" name="StructureModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="RefreshedTime" name="RefreshedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="SystemFlags" name="SystemFlags" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="KeepUniqueRows" name="KeepUniqueRows"
type="xsd:boolean" minOccurs="0" />
                  <xsd:element sql:field="DisplayOrdinal" name="DisplayOrdinal"
type="xsd:int" minOccurs="0" />
                  <xsd:element sql:field="ErrorMessage" name="ErrorMessage" type="xsd:string"
minOccurs="0" />
                  <xs:element sql:field="SourceProviderType" name="SourceProviderType"
type="xs:string" minOccurs="0" />
                  <xs:element sql:field="DisplayFolder" name="DisplayFolder" type="xs:string"
minOccurs="0" />
                  <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <ID>11</ID>
              <TableID>10</TableID>
              <ExplicitName>RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61</ExplicitName>
              <ExplicitDataType>6</ExplicitDataType>
              <InferredDataType>19</InferredDataType>
              <IsHidden>true</IsHidden>
              <State>1</State>
              <IsUnique>true</IsUnique>
              <IsKey>true</IsKey>
              <IsNullable>false</IsNullable>
              <Alignment>1</Alignment>
              <TableDetailPosition>-1</TableDetailPosition>
              <IsDefaultLabel>false</IsDefaultLabel>
              <IsDefaultImage>false</IsDefaultImage>
              <SummarizeBy>1</SummarizeBy>
              <ColumnStorageID>23</ColumnStorageID>
              <Type>3</Type>
              <IsAvailableInMDX>true</IsAvailableInMDX>
              <AttributeHierarchyID>12</AttributeHierarchyID>
              <ModifiedTime>2015-09-30T03:25:33.31</ModifiedTime>
              <StructureModifiedTime>2015-09-30T03:25:33.306667</StructureModifiedTime>
              <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
              <SystemFlags>0</SystemFlags>
              <KeepUniqueRows>false</KeepUniqueRows>
              <DisplayOrdinal>0</DisplayOrdinal>
            </row>
            <row>
              <ID>14</ID>
              <TableID>10</TableID>
              <ExplicitName>x</ExplicitName>
              <ExplicitDataType>6</ExplicitDataType>
              <InferredDataType>19</InferredDataType>
              <IsHidden>false</IsHidden>
              <State>1</State>
              <IsUnique>false</IsUnique>
              <IsKey>false</IsKey>
              <IsNullable>true</IsNullable>
              <Alignment>1</Alignment>
              <TableDetailPosition>-1</TableDetailPosition>
              <IsDefaultLabel>false</IsDefaultLabel>
              <IsDefaultImage>false</IsDefaultImage>
              <SummarizeBy>1</SummarizeBy>
              <ColumnStorageID>27</ColumnStorageID>
              <Type>1</Type>
              <IsAvailableInMDX>true</IsAvailableInMDX>
              <AttributeHierarchyID>16</AttributeHierarchyID>
```

```
            <ModifiedTime>2015-09-30T03:25:33.31</ModifiedTime>
            <StructureModifiedTime>2015-09-30T03:25:33.31</StructureModifiedTime>
            <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
            <SystemFlags>0</SystemFlags>
            <KeepUniqueRows>false</KeepUniqueRows>
            <DisplayOrdinal>0</DisplayOrdinal>
          </row>
          <row>
            <ID>15</ID>
            <TableID>10</TableID>
            <ExplicitName>y</ExplicitName>
            <ExplicitDataType>2</ExplicitDataType>
            <InferredDataType>19</InferredDataType>
            <IsHidden>false</IsHidden>
            <State>1</State>
            <IsUnique>false</IsUnique>
            <IsKey>false</IsKey>
            <IsNullable>true</IsNullable>
            <Alignment>1</Alignment>
            <TableDetailPosition>-1</TableDetailPosition>
            <IsDefaultLabel>false</IsDefaultLabel>
            <IsDefaultImage>false</IsDefaultImage>
            <SummarizeBy>1</SummarizeBy>
            <ColumnStorageID>31</ColumnStorageID>
            <Type>1</Type>
            <IsAvailableInMDX>true</IsAvailableInMDX>
            <AttributeHierarchyID>17</AttributeHierarchyID>
            <ModifiedTime>2015-09-30T03:25:33.31</ModifiedTime>
            <StructureModifiedTime>2015-09-30T03:25:33.31</StructureModifiedTime>
            <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
            <SystemFlags>0</SystemFlags>
            <KeepUniqueRows>false</KeepUniqueRows>
            <DisplayOrdinal>0</DisplayOrdinal>
          </row>
        </root>
        <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla"
name="AttributeHierarchy">
          <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
            <xsd:element name="root">
              <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                  <xsd:element name="row" type="row" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:simpleType name="uuid">
              <xsd:restriction base="xsd:string">
                <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
              </xsd:restriction>
            </xsd:simpleType>
            <xsd:complexType name="xmlDocument">
              <xsd:sequence>
                <xsd:any />
              </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="row">
              <xsd:sequence>
                <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0"
/>
                <xsd:element sql:field="ColumnID" name="ColumnID" type="xsd:unsignedLong"
minOccurs="0" />
                <xsd:element sql:field="State" name="State" type="xsd:long" minOccurs="0"
/>
```

```
                  <xsd:element sql:field="AttributeHierarchyStorageID"
name="AttributeHierarchyStorageID" type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element sql:field="ModifiedTime" name="ModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="RefreshedTime" name="RefreshedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <ID>12</ID>
              <ColumnID>11</ColumnID>
              <State>4</State>
              <AttributeHierarchyStorageID>35</AttributeHierarchyStorageID>
              <ModifiedTime>2015-09-30T03:25:33.306667</ModifiedTime>
              <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
            </row>
            <row>
              <ID>16</ID>
              <ColumnID>14</ColumnID>
              <State>4</State>
              <AttributeHierarchyStorageID>36</AttributeHierarchyStorageID>
              <ModifiedTime>2015-09-30T03:25:33.31</ModifiedTime>
              <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
            </row>
            <row>
              <ID>17</ID>
              <ColumnID>15</ColumnID>
              <State>4</State>
              <AttributeHierarchyStorageID>37</AttributeHierarchyStorageID>
              <ModifiedTime>2015-09-30T03:25:33.31</ModifiedTime>
              <RefreshedTime>1699-12-31T00:00:00</RefreshedTime>
            </row>
          </root>
          <root xmlns="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msxmla="http://schemas.microsoft.com/analysisservices/2003/xmla" name="Partition">
            <xsd:schema targetNamespace="urn:schemas-microsoft-com:xml-analysis:rowset"
xmlns:sql="urn:schemas-microsoft-com:xml-sql" elementFormDefault="qualified">
              <xsd:element name="root">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element name="row" type="row" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:simpleType name="uuid">
                <xsd:restriction base="xsd:string">
                  <xsd:pattern value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-
Z]{4}-[0-9a-zA-Z]{12}" />
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:complexType name="xmlDocument">
                <xsd:sequence>
                  <xsd:any />
                </xsd:sequence>
              </xsd:complexType>
              <xsd:complexType name="row">
                <xsd:sequence>
                  <xsd:element sql:field="ID" name="ID" type="xsd:unsignedLong" minOccurs="0"
/>
                  <xsd:element sql:field="TableID" name="TableID" type="xsd:unsignedLong"
minOccurs="0" />
                  <xsd:element sql:field="Name" name="Name" type="xsd:string" minOccurs="0"
/>
                  <xsd:element sql:field="Description" name="Description" type="xsd:string"
minOccurs="0" />
```

```
                  <xsd:element sql:field="DataSourceID" name="DataSourceID"
type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element sql:field="QueryDefinition" name="QueryDefinition"
type="xsd:string" minOccurs="0" />
                  <xsd:element sql:field="State" name="State" type="xsd:long" minOccurs="0"
/>
                  <xsd:element sql:field="Type" name="Type" type="xsd:long" minOccurs="0" />
                  <xsd:element sql:field="PartitionStorageID" name="PartitionStorageID"
type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element sql:field="Mode" name="Mode" type="xsd:long" minOccurs="0" />
                  <xsd:element sql:field="DataView" name="DataView" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="ModifiedTime" name="ModifiedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="RefreshedTime" name="RefreshedTime"
type="xsd:dateTime" minOccurs="0" />
                  <xsd:element sql:field="SystemFlags" name="SystemFlags" type="xsd:long"
minOccurs="0" />
                  <xsd:element sql:field="ErrorMessage" name="ErrorMessage" type="xsd:string"
minOccurs="0" />
                  <xsd:element sql:field="ImpactType" name="ImpactType" type="xsd:int" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:schema>
            <row>
              <ID>13</ID>
              <TableID>10</TableID>
              <Name>partition</Name>
              <State>1</State>
              <Type>3</Type>
              <PartitionStorageID>20</PartitionStorageID>
              <Mode>2</Mode>
              <DataView>3</DataView>
              <ModifiedTime>2015-09-30T03:26:42.27</ModifiedTime>
              <RefreshedTime>2015-10-01T02:31:38.766667</RefreshedTime>
              <SystemFlags>0</SystemFlags>
            </row>
          </root>
        </AffectedObjects>
      </return>
    </ExecuteResponse>
  </soap:Body>
</soap:Envelope>
```

4.2 Refresh Tabular Metadata (JSON)

In this example, the client sends a **JSON Tabular Refresh Command** to the server that automatically refreshes the objects that need to be refreshed.

4.2.1 Client Sends Request

The client sends the following request.

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Header>
    <Session xmlns="urn:schemas-microsoft-com:xml-analysis"
    SessionId="34B67555-85B9-46CE-8803-4BEC7D6AEE13" />
  </Header>
  <Body>
    <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
      <Command>
        <Statement>
{
  "refresh": {
    "type": "automatic",
```

```
    "objects": [
      {
        "database": "Adventure Works "
      }
    ]
  }
}
        </Statement>
      </Command>
      <Properties>
        <PropertyList>
        </PropertyList>
      </Properties>
    </Execute>
  </Body>
</Envelope>
```

4.2.2 Server Response

The server responds with an empty result.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ExecuteResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <return>
         <root xmlns="urn:schemas-microsoft-com:xml-analysis:empty"/>
      </return>
    </ExecuteResponse>
  </soap:Body>
</soap:Envelope>
```

4.3 CreateOrReplace Tabular Metadata (JSON)

In this example, the client sends a **JSON Tabular Create Command** to the server to create or replace the specified partition and any descendants.

4.3.1 Client Sends Request

The client sends the following request.

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Header>
    <Session xmlns="urn:schemas-microsoft-com:xml-analysis"
    SessionId="34B67555-85B9-46CE-8803-4BEC7D6AEE13" />
  </Header>
  <Body>
    <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
      <Command>
        <Statement>
{
  "createOrReplace": {
    "object": {
      "database": "Adventure Works ",
      "table": "DimDate",
      "partition": "DimDate 2"
    },
    "partition": {
      "name": "DimDate 2",
      "source": {
        "dataSource": "localhost AdventureworksDW",
        "query": [
          "SELECT [dbo].[DimDate].* FROM [dbo].[DimDate]\r",
```

```
            "where CalendarYear=2009"
          ]
        }
      }
    }
}
        </Statement>
      </Command>
      <Properties>
        <PropertyList>
        </PropertyList>
      </Properties>
    </Execute>
  </Body>
</Envelope>
```

4.3.2 Server Response

The server responds with an empty result.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ExecuteResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <return>
         <root xmlns="urn:schemas-microsoft-com:xml-analysis:empty"/>
      </return>
    </ExecuteResponse>
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

The server could be returning potentially sensitive data in its responses. Therefore, it is strongly recommended that the server be configured to use GSS-API based encryption over TCP or Secure Sockets Layer (SSL) over HTTPS to ensure the integrity of the data and to prevent tampering and unauthorized access.

There are two strategies for reducing the impact of denial-of-service (DOS) attacks against the server:

- Turn on authentication and deny access to unauthenticated clients. This will allow a user to quickly disable access to rogue client machines.
- Make sure no single request takes too much processing time on the server. That will ensure that any attacker needs to maintain a steady stream of requests to deny access to the server. Therefore, a simple network trace will allow one to identify the offending machine and shut it down. This applies to requests sent by "spoof clients" (for example, a virus emulating a client that might try to pass an unbounded request or a long-running MDX query).

5.2 Index of Security Parameters

None.

6 Appendix A: 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.

- Microsoft SQL Server 2016

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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 2.2.5.1: Microsoft implementations do not support the **StorageLocation** property.

<2> Section 2.2.5.1: Microsoft SQL Server Analysis Services requires this value to be a valid Windows culture name, such as "en-US" or "de-DE".

<3> Section 2.2.5.2: In Analysis Services, the behavior of the value "Default" is dependent upon the context in which impersonation is used.

<4> Section 2.2.5.2: In Analysis Services, the user account is the Windows user account.

<5> Section 2.2.5.2: Microsoft implementations do not support the **Snapshot** value.

<6> Section 2.2.5.4: For more information on the use of **DefaultDetails**, see [MSDN-DEFDETAILS].

<7> Section 2.2.5.12: Analysis Services provides a list of known values; however, other values are possible.

<8> Section 2.2.5.22: In Analysis Services, the only supported value is an empty string, which indicates that Windows authentication is used.

<9> Section 2.2.5.22: This property is always "Auto" for the Windows identity provider.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- The removal of a document from the documentation set.

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 **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.5.1 Model Object	7512664 : Specified the type for the Version property.	Y	Content update.
2.2.5.2 DataSource Object	7512671 : Specified the type for the Password property.	Y	Content update.
2.2.5.6 Partition Object	7512680 : Specified the type for the ErrorMessage property.	Y	Content update.
2.2.5.23 TablePermission Object	7512685 : Specified the type for the ErrorMessage property.	Y	Content update.
3.1.5.2.1.1.1.2 Create DataSources	7512762 : Removed the default value for ImpersonationMode.	Y	Content update.
3.1.5.2.1.2.1.4 Alter Columns	7512782 : Removed TableID and TableID.Table from the list of elements.	Y	Content update.
3.1.5.2.1.6.1 Request	7512794 : Revised schema definition for MergePartitions and added XSD for Partitions rowset.	Y	Content update.
3.1.5.2.1.8.2 Response	7512811 : Added descriptions of responses to successful requests.	Y	Content update.
3.1.5.2.1.9.2 Response	7512828 : Added description of the response to a successful request.	Y	Content update.
4.1.2 Server Response	7512834 : Added SourceProviderType and DisplayFolder to the response.	Y	Content update.

8 Index