JasperAnalysis Workbench Guide
Version 3.7

All Rights reserved. Copyright 2010.
Jaspersoft Corporation

Table of Contents

1. INTRODUCTION 3

2. INSTALLING AND CONFIGURING THE WORKBENCH 4

2.1 Creating a Database Connection 4

2.2 Advanced Configuration 5

3. USING THE WORKBENCH 6

3.1 Main Window 6

3.2 Schema Explorer 7

3.2.1 Overview 7

3.2.2 Adding Elements 7

3.2.3 Navigating an OLAP Schema 9

3.2.4 Working with Existing OLAP Schemas 10

3.2.5 Access Grants for JasperAnalysis Professional 12

3.3 MDX Query 13

3.4 Exploring the Database 14

3.5 Setting up Schemas 15

3.5.1 Getting Started 16

3.5.2 Defining Database Connection 16

3.5.3 Creating a Simple Schema 17

1. Introduction

The JasperAnalysis Schema Workbench allows you to visually create and test Mondrian OLAP cube schemas. It provides these features:

- Schema Explorer: an editor integrated with the underlying data source for validation (Figure 1).
- MDX Query: Tests MDX queries against schema and database. Automatically generate example MDX queries.
- JDBC Explorer: Browses underlying databases structure.

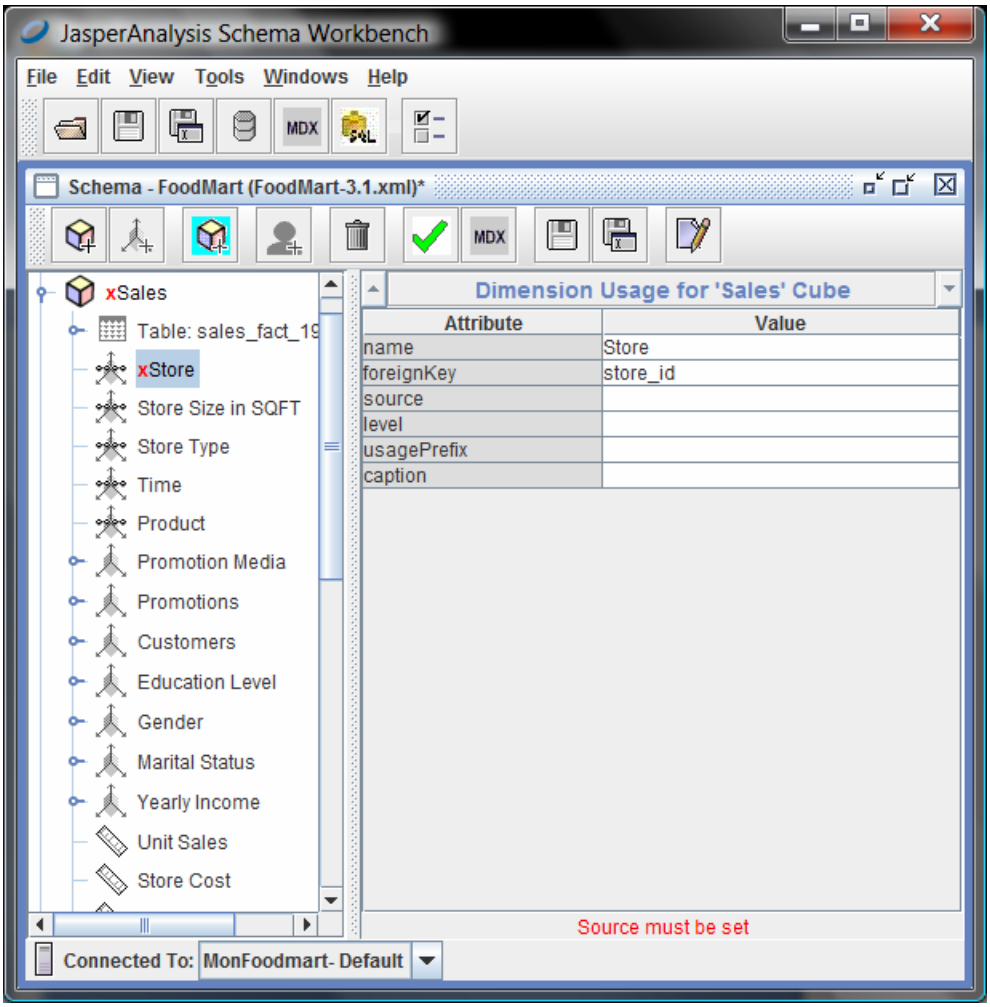

Figure 1 JasperAnalysis Schema Workbench

Note that the JasperAnalysis Schema Workbench comes in two editions:

- The Community Edition provides the majority of the functionality, such as support for OLAP schemas and MDX.
- The Professional Edition builds upon the Community Edition and adds support for creating security files that can be used in JasperAnalysis to secure data according to user roles.

See the Mondrian Technical Guide to understand schemas. It is found in the docs directory in the workbench installation. Once you have the schema XML file, you can upload this into your application that uses Mondrian, such as JasperServer.

2. Installing and Configuring the Workbench

To install the workbench, unzip its archive into a directory on your hard drive. Once installed, configure a JDBC connection that points to your data.

The workbench needs a JDBC connection to the database that stores the data you want to describe in OLAP terms. The RDBMS drivers Jaspersoft provides various with the edition of the workbench:

- The Community Edition includes drivers for MySQL, PostgreSQL and Ingres.
- The Professional Edition includes drivers for MySQL, PostgreSQL, Ingres, Oracle, Microsoft SQL Server, and DB2.

If your data is stored in a different RDBMS, you can add the appropriate JDBC driver to the /drivers directory in the workbench installation, then select Other as the Type of database when you define a connection.

In the JasperAnalysis Community Edition release, JDBC drivers are provided for MySQL, PostgreSQL and Ingres. You can include your cube database's JDBC driver in the drivers directory in the workbench installation, so the workbench can use it.

This document and the Mondrian Technical Guide are included in the docs directory.

Start the workbench by running workbench.bat (Windows) or workbench.sh (for Unix/Linux) where you installed the workbench. In Linux/Unix, you may need to set execute permissions on workbench.sh in order to run it.

2.1 Creating a Database Connection

Set properties for connection to your cube database by clicking **Tools** > **Connection** .

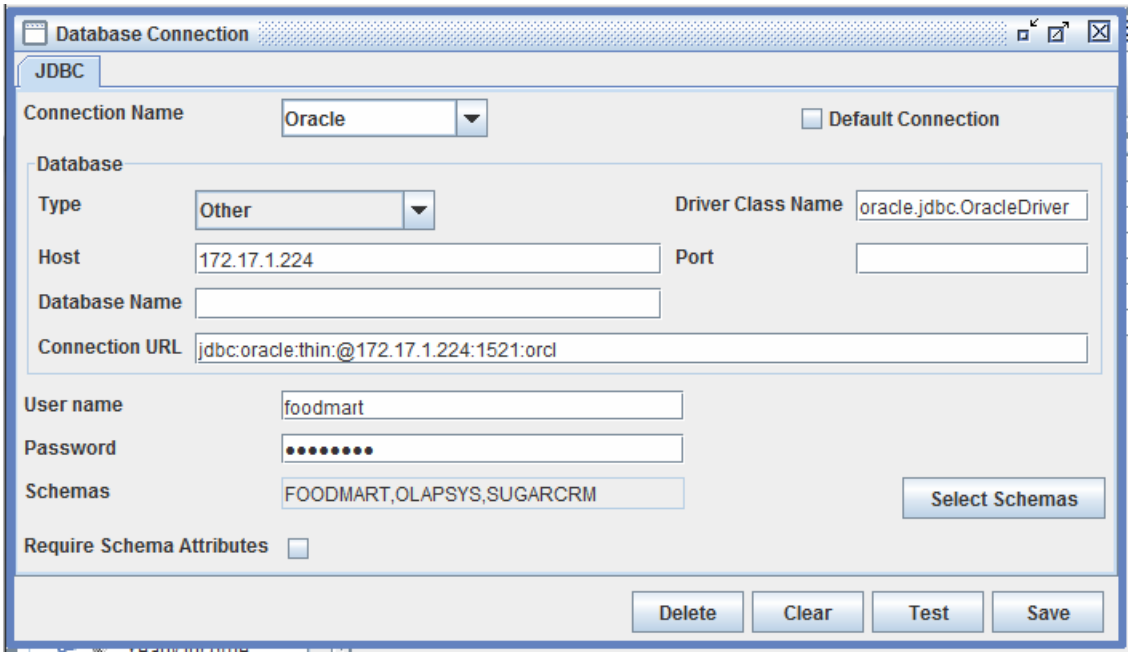

Figure 2 Database Connection window

You can have a number of JDBC connections defined. The list of connections you have defined is in the Connection name drop-down. To create a new connection, enter the name in the connection name drop-down, define the parameters, test the connection, and save it in the list.

By default, the workbench automatically populates data for certain RDBMS types (such as MySQL and DB2). Selecting one such in the **Database Type** drop-down populates the **Driver Class Name** field and automatically generates a simple connection string in the **Connection URL** field based on the **Host**, **Port** and **Database Name** values that you enter. You can add additional parameters in the Connection URL field. Select the Other database to you enter values specific to one of the . When you enter values in the **Connection URL**, the **Host**, **Port** and **Database Name** values are updated when the workbench can insert the correct value automatically.

The list of possible schemas is provided in all schema fields in the Schema Explorer. Note that the workbench may be slow to load information about the schemas, tables and columns in the database, depending the amount of schemas, tables and columns in your data source and the underlying performance of database information retrieval.

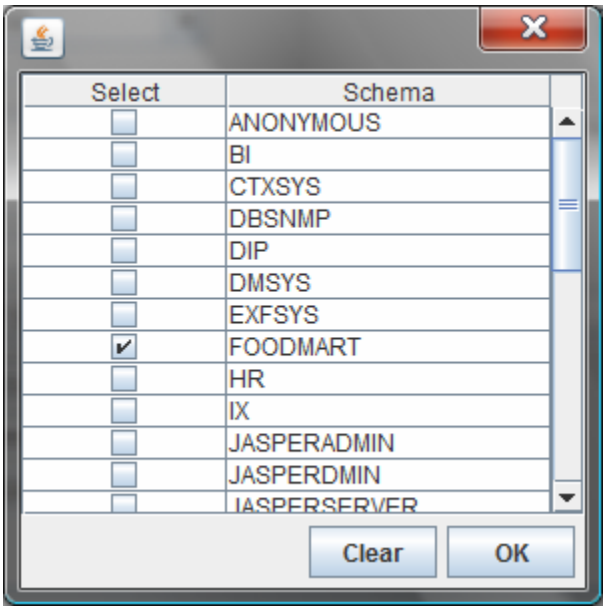

Figure 3 Selecting a Schema

After the text fields, the window displays a set of options:

- When the **Requires Schema Attributes** option is enabled, the workbench requires the schema name to be entered for all tables and joins in the OLAP schema. This option is useful in several cases, including:
    - The data described by your OLAP schema spans multiple database schemas.
    - You are using an RDBMS such as Oracle and DB2, which create a database schema for each connecting user. When the user connects, they retrieve data from this schema by default. However, the data you need to access is typically in a separate schema, so you must specify the schema where your data resides.
- The **Delete** button in the Connection window deletes the currently selected connection.
- The **Clear** button resets the values in the window.
- The **Test** button tests whether the workbench can access the data source properly. It looks for the correct database driver (in a JAR) in the workbench's drivers directory and uses the properties entered in the Connection window to try connect to the database. This ensures the connection can be used to validate schemas and explore tables and columns.
- The **Save** button saves the connection; click it once your test is successful.

Close the Connection window by clicking ☒ in its upper right-hand corner.

2.2 Advanced Configuration

A variety of configuration files are found under your user home directory in s directory called .schemaWorkbench.

- workbench.properties. Settings for the cube database preferences and the last ten schema files that have been edited.
- log4j.xml. Settings for the execution log file. Use it to see details about any problems you have with connections, schemas, or MDX queries. By default the log file is in <user_home>/.schemaWorkbench/workbench.log.
- mondrian.properties. Configuration of Mondrian run time properties used when the workbench is running. For details about these settings, see the Mondrian Technical Guide.

If you are using a using a database that requires a JDBC driver that is not already installed, you must put the JDBC JAR files into the drivers directory created when you installed the JasperAnalysis Schema Workbench.

3. Using the Workbench

3.1 Main Window

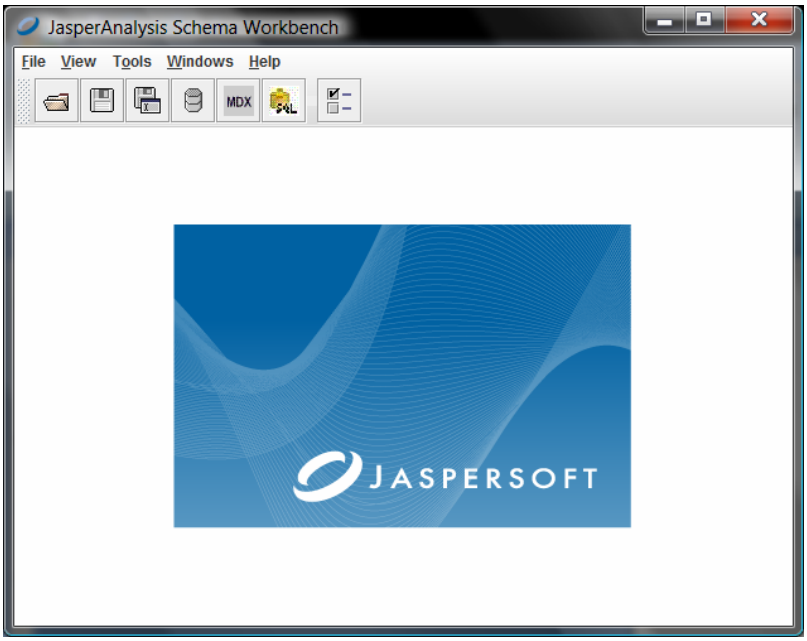

Figure 4 Main Window

The top level menus and tool bars in the workbench provide the most common functionality:

Menu	Option	Description
File	New Schema MDX Query JDBC Explorer	Create a new window for one of these; this option is also available from the tool bar.
	Open	Open an existing schema file; this option is also available from the tool bar.
	Save	Save current schema file; this option is also available from the tool bar.
	Save As	Save current schema file to a new name; this option is also available from the tool bar.
	List of Recent Files	Open previously edited schema files
	Exit	Exit the workbench
View	View XML	Switch between graphical and XML view of the current schema
Tools	Connection	Maintain JDBC connections; this option is also available from the tool bar.
Windows	List of Open Files	Click a file name to maximize it and bring it to the front
	Cascade Windows	Arrange windows from top left to bottom right
	Tile Windows	Size and show all windows in current display
	Maximize All	Maximize all open windows
	Minimize All	Minimize all open windows

Menu	Option	Description
	Close All	Close all open windows
Help	About	Shows the version and help information

3.2 Schema Explorer

3.2.1 Overview

The Schema Explorer allows you to create or edit the OLAP schemas JasperAnalysis needs. OLAP schemas are XML files that have a hierarchical structure; the Schema Explorer shows that structure. The Schema Explorer prompts for values and validates your schema against the cube database tables and column names, values and names in the schema, and other details, such as the required values and those selected from lists of values driven by your other selections. For example, when you define a join, the drop-down only includes fields in tables included in the join. Launch the Schema Explorer with **New** > **Schema**.

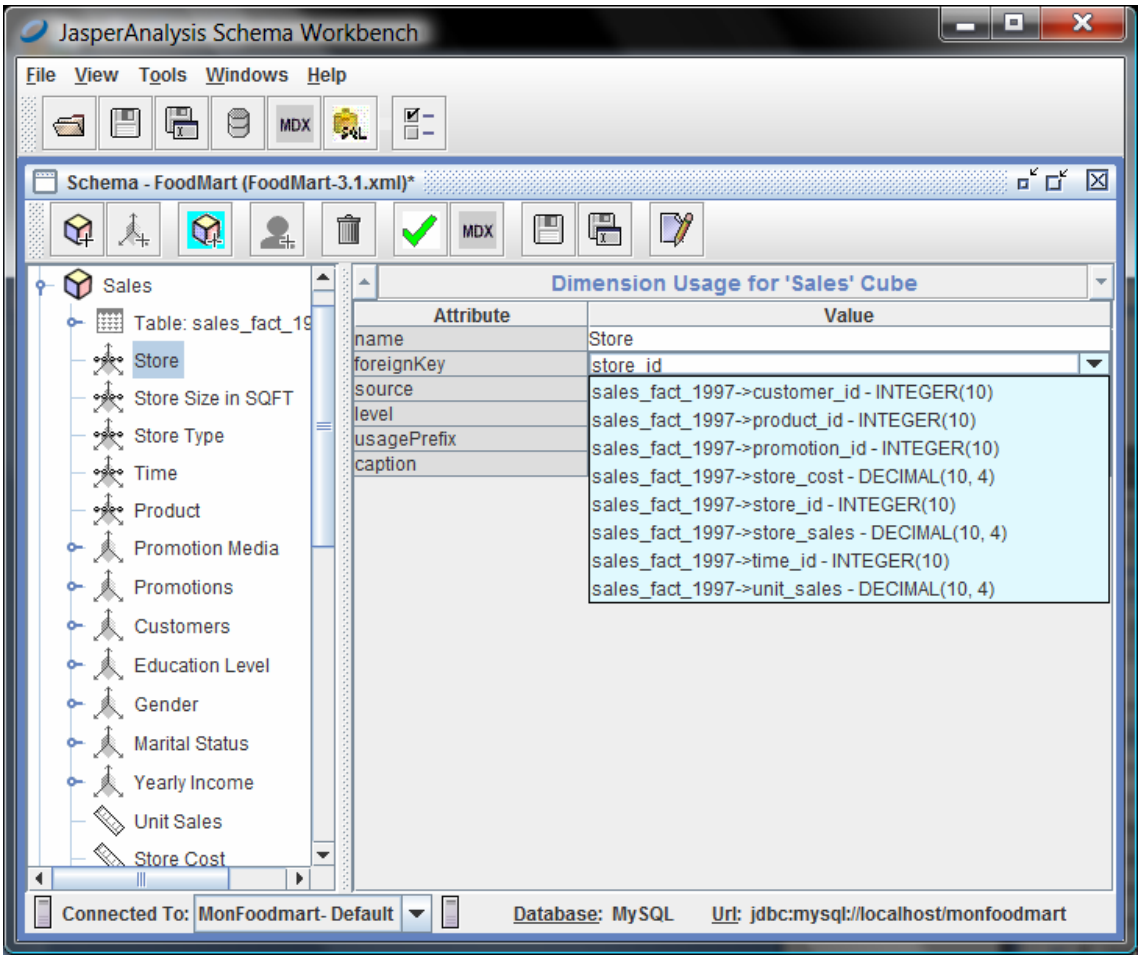

Figure 5 Selecting a Foreign Key for a Dimension

For complete validation against the underlying database, click **(Test Schema)** button in the tool bar.

3.2.2 Adding Elements

Edit elements by right-clicking one in the schema tree on the left hand side of the window to display a pop-up menu of actions relevant for the selected item.

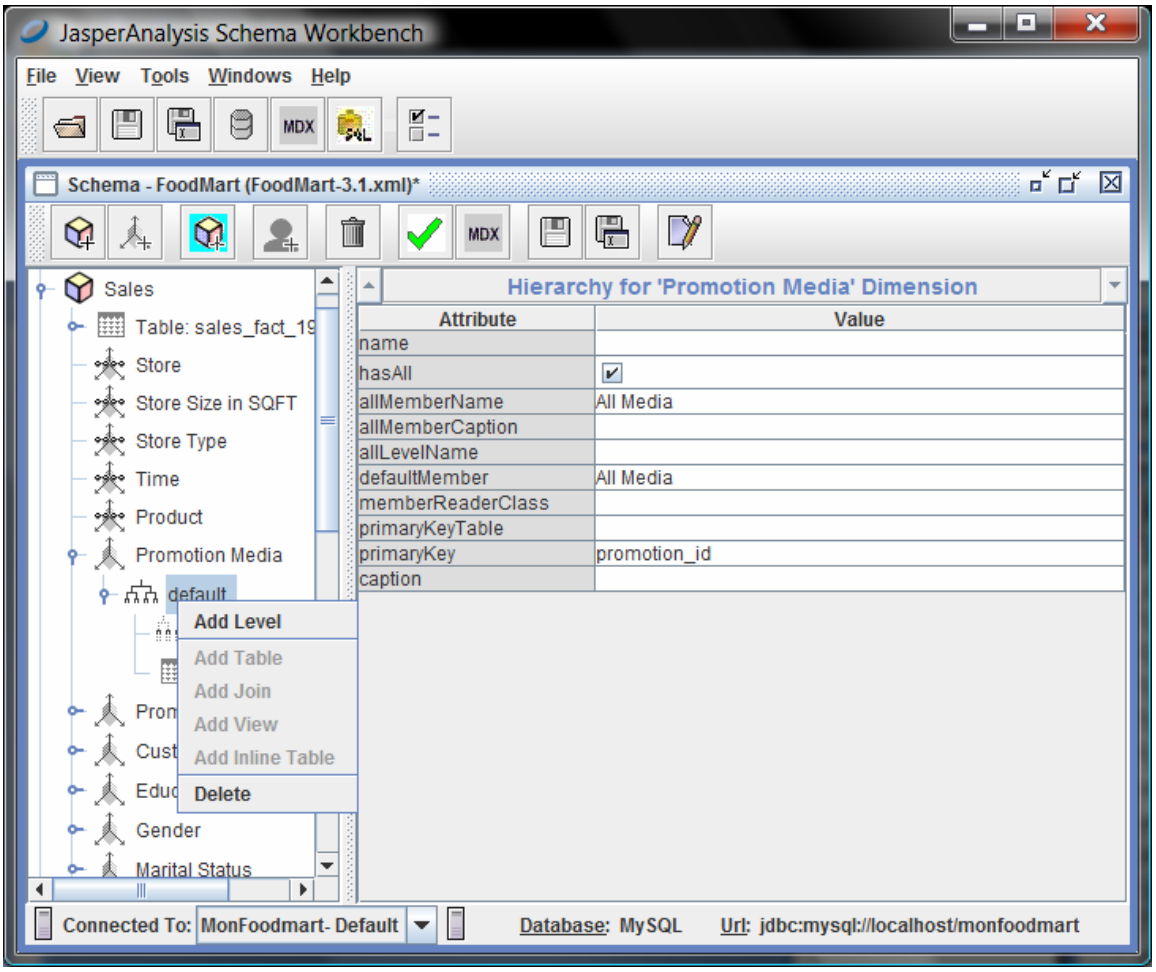

Figure 6 Right-click Context Menu

Use the MDX tool bar button to launch an MDX query editor against this schema. The final level of schema validation is the successful execution of MDX queries.

The schema can be saved to a file with the **Save** and **Save As** tool bar buttons. When you change a schema and either close the file or exit the workbench, you are prompted to save.

Use the tool bar buttons to add top level elements in the schema:

- Cube
- Dimension
- Virtual Cube
- Role

  Role definitions can only be used in conjunction with the Professional Edition of JasperAnalysis. They have no effect in JasperAnalysis Community Edition.

You can also add elements by right-clicking an element and selecting from the context menu. Click **Delete** on the context menu to delete the selected element.

Click the **Edit Mode** button to switch between the graphical view of the schema and an XML view. Clicking elements in the left-hand panel shows the XML for the selected element and the elements it contains.

3.2.3 Navigating an OLAP Schema

Selecting points in the schema in the left hand panel shows the XML for the selected element and all XML within that element. You can't edit the XML directly. This can also be toggled from the top level menu: **View > XML**.

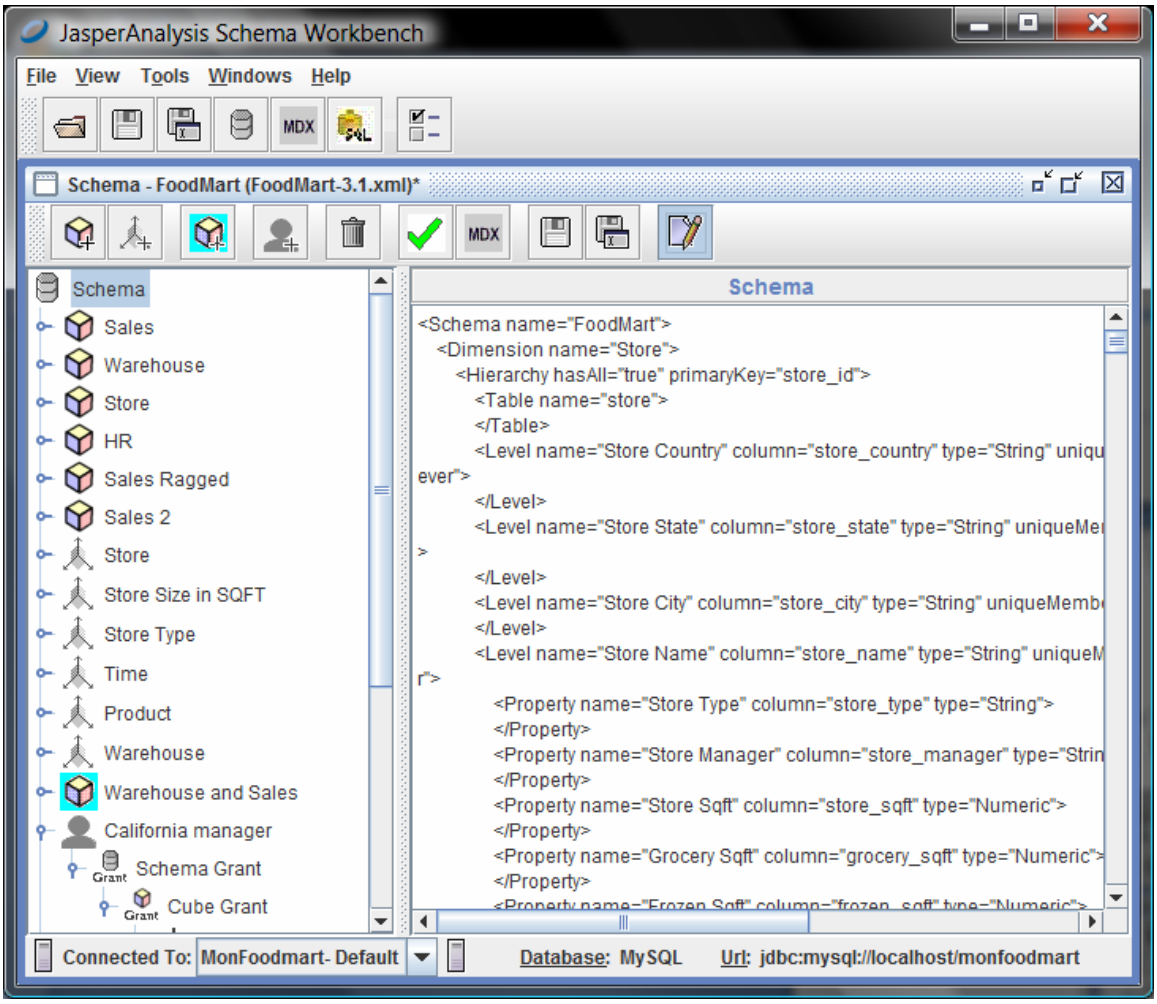

Figure 7 XML View of an OLAP Schema

At the bottom of the panel, a drop-down of available JDBC connections shows the current selected connection name, the database type determined from the JDBC driver, and the JDBC Connection URL. You can change the connection you want to validate against.

3.2.4 Working with Existing OLAP Schemas

When opening a schema XML file for the first time or creating a new schema, you are prompted for a connection against which to validate the schema against.

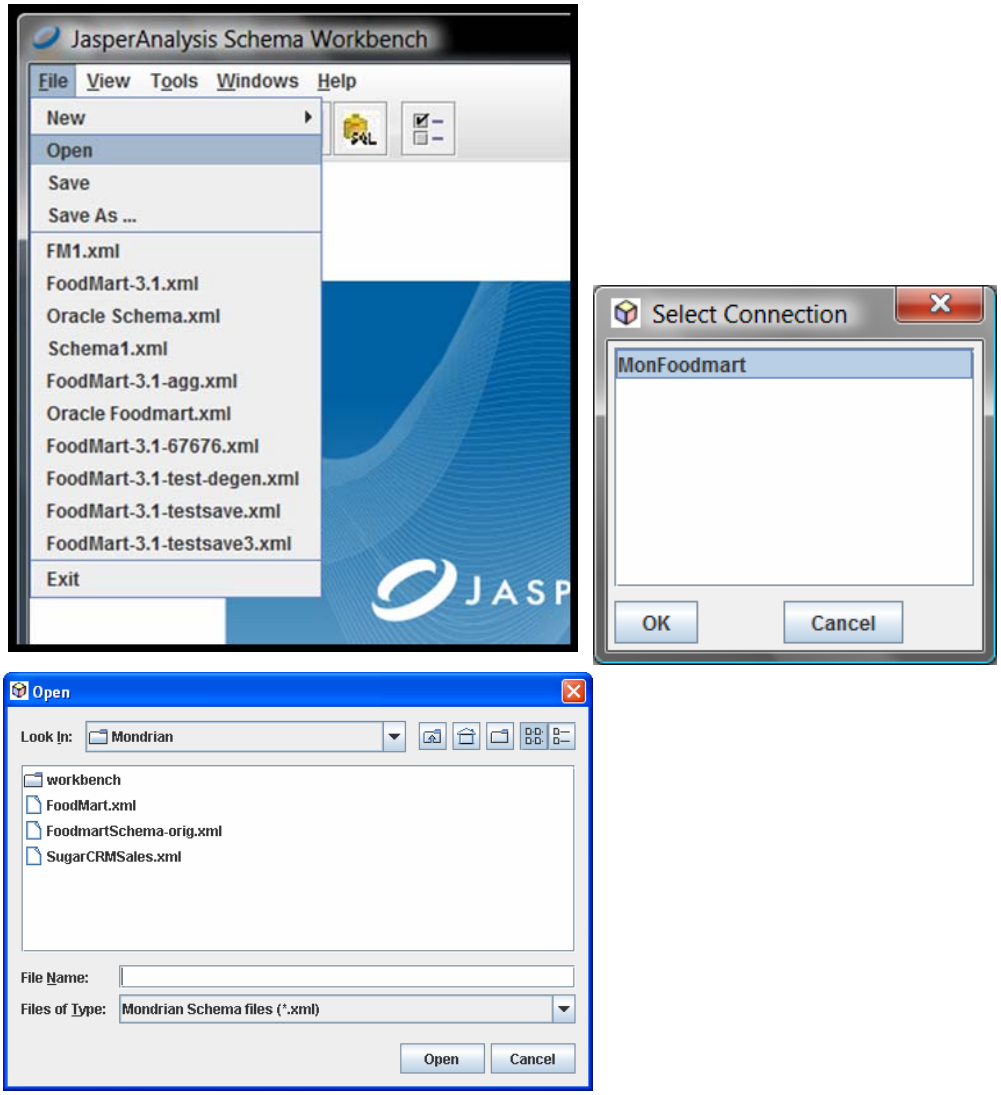

Figure 8 Window Sequence When Opening an Existing Schema

You can reopen an existing schema file. The schema is validated as it opens and the workbench reports any errors it detects.

The ten most-recently saved schemas are available from the **File** menu. The directory the schemas were in is tracked behind the scenes. When opened, a schema is validated against the JDBC connection parameters you last specified for it.

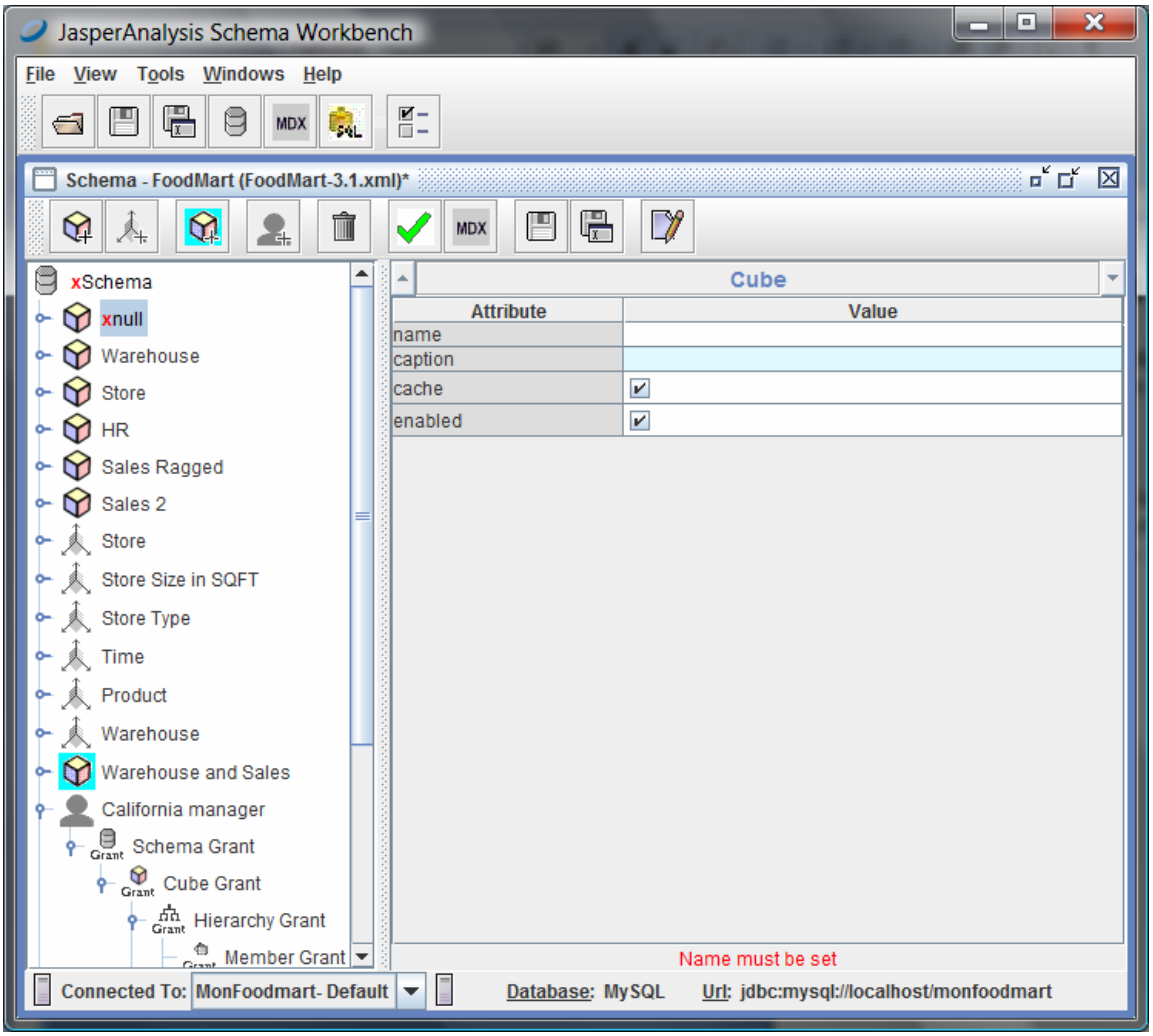

Figure 9 Status Message Showing an Error in Red

The first error detected in the current element is described in red at the bottom of the window and a red X is shown in the schema tree. When an element has sub-elements that are in error **<Element Type> is invalid** appears.

3.2.5 Access Grants for JasperAnalysis Professional

This section describes functionality that is available only in JasperAnalysis Professional. To get this software, contact Jaspersoft.

JasperAnalysis Professional supports the ability to define data-level security by extending the Roles concept in Mondrian. User role assignments and filters based on parameters from the profile of the currently logged in user. See the *JasperAnalysis Professional User Guide* and the *JasperAnalysis Ultimate Guide* for details.

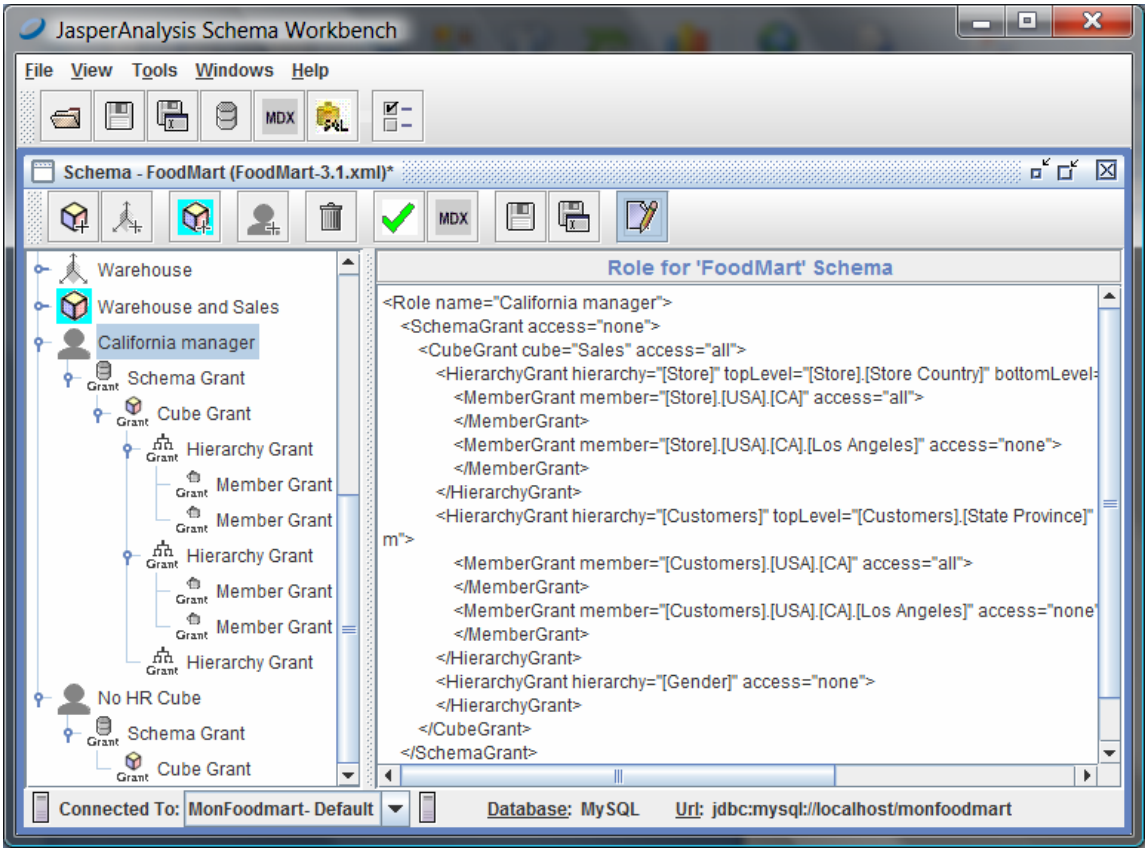

Figure 10 Security File for JasperAnalysis Professional (AGXML Format)

JasperAnalysis Professional relies on AGXML files to define role-based, data-level security. The Professional edition of the JasperAnalysis Schema Workbench supports the creation of these files by allowing you to add and edit role definitions in the Schema Explorer. The security information is maintained in a separate file than the OLAP schema itself, and gives the file the correct file extension (.agxml). This file is save d to the same directory as the schema XML file itself, with the same base file name. For example, if your schema's file name is MySchema.xml, the security file is named MySchema.agxml.

To use the security file, upload it as a part of the Mondrian connection used by your analysis view in JasperAnalysis Professional.

3.3 MDX Query

Save your schema and click **New > MDX Query** to run some MDX queries to test. You can also use the MDX buttons in the Main window and the Schema Explorer. Connecting to a schema completely validates the schema. Error messages or results are displayed in the workbench window, in the console window, and in the workbench log file.

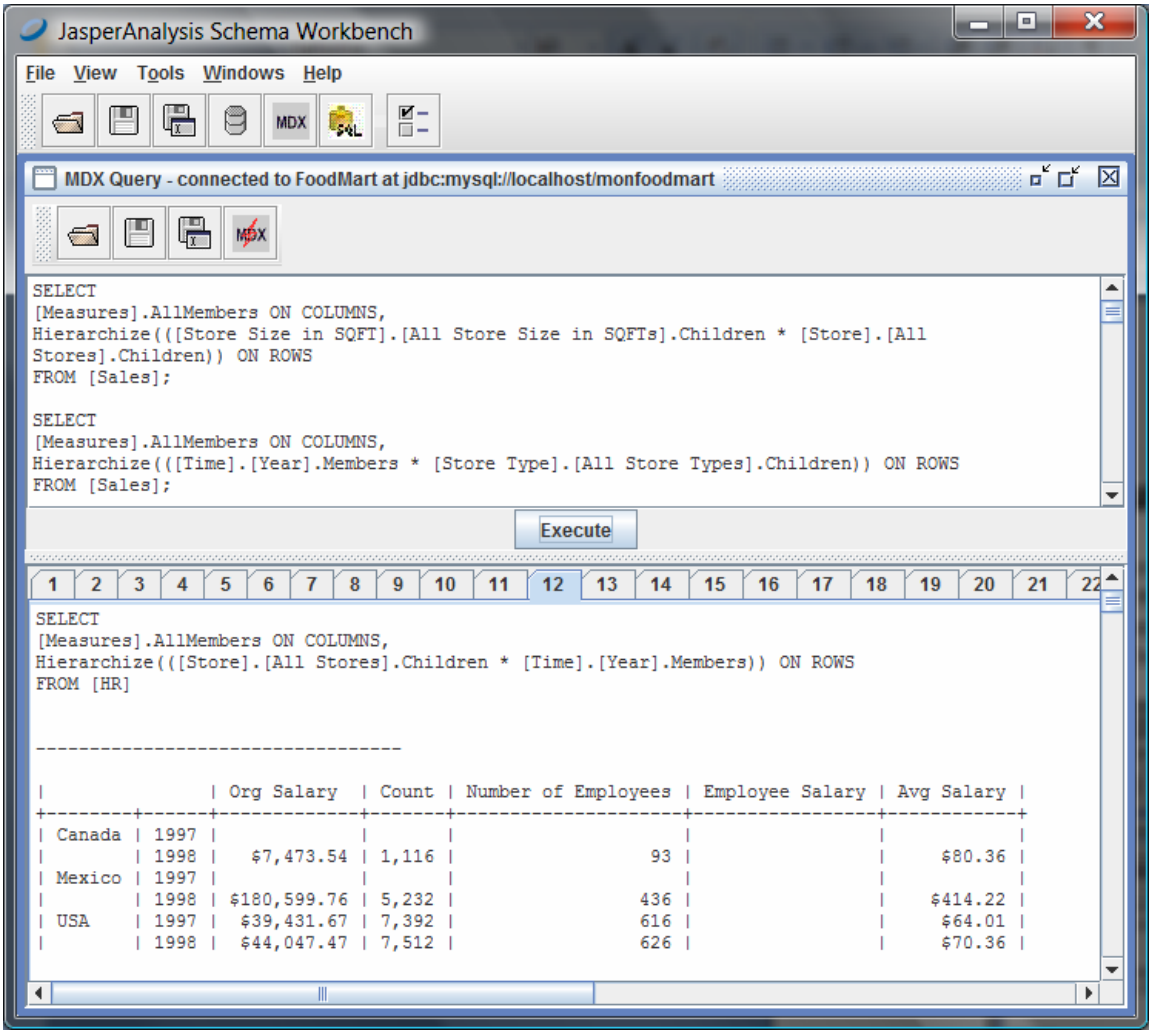

Figure 11 MDX Query and Results

The query panel can contain multiple MDX queries, each separated by a ; (semi-colon). Comments in a query are indicated by a # (number sign) as the first character of a line. When **Execute** is pressed, each query is run and the results are displayed, one tab per query result, in the table format shown in Figure 11. Any exceptions, such as bad syntax, are displayed.

The MDX Query Editor provides these features:

- **Open**, **Save**, and **Save As**.

  Maintain files of MDX queries in the query panel. MDX query files have an .mdx file extension.

- **Generate MDX Queries**

  Generate default MDX queries based on the underlying schema. This create at least one query per cube in the schema. Each query contains no more than two dimensions.

When you launch an MDX Query Window from the Main window, you can select one of the open schemas against which to run your MDX queries.

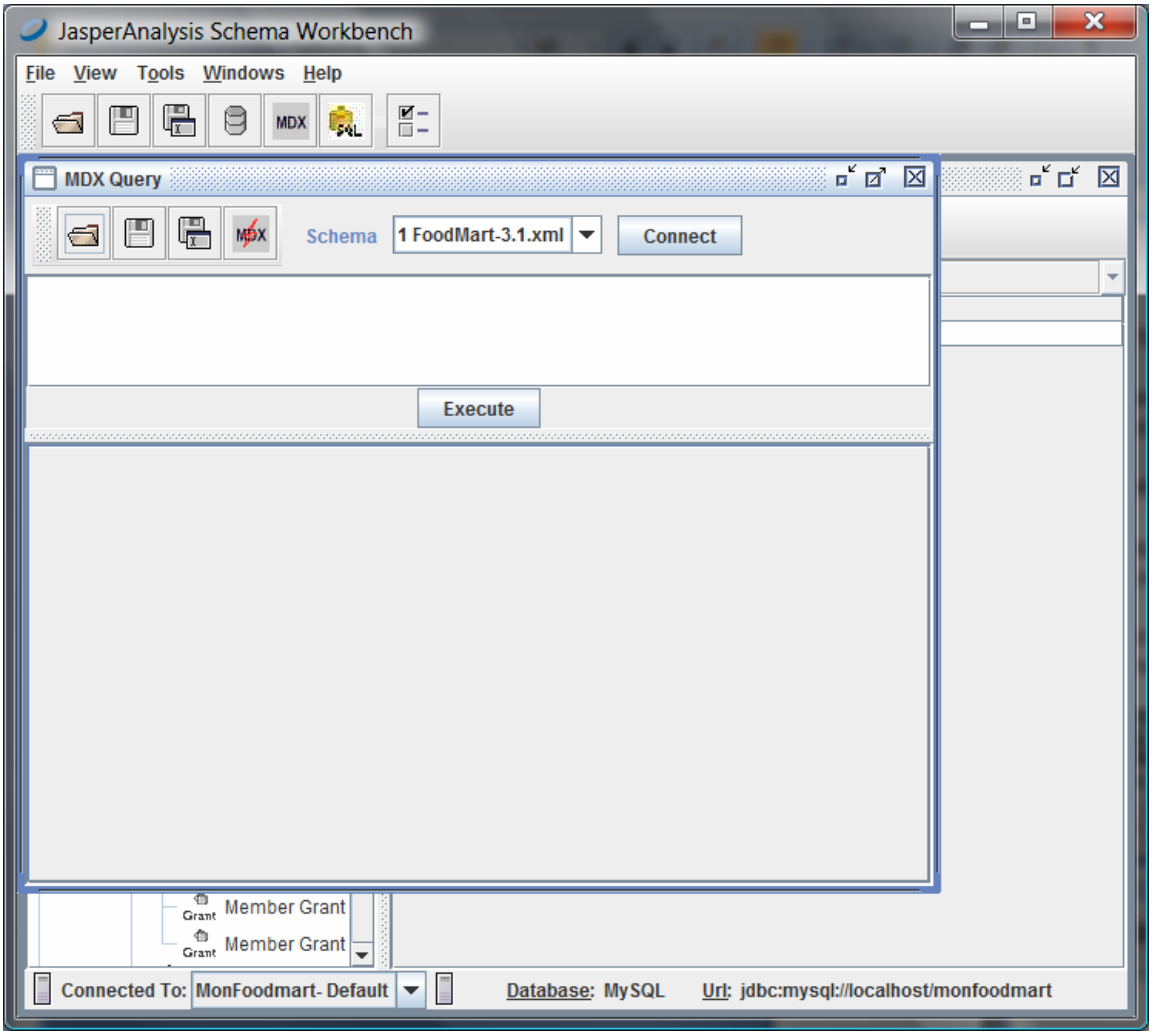

Figure 12 Selecting a Schema in the MDX Editor

3.4 Exploring the Database

Open the JDBC Explorer by clicking **New > JDBC Explorer**. This displays the structure of the current database. First, you are prompted for a connection to explore:

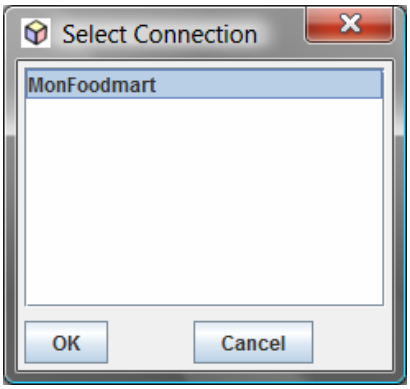

Figure 13 Selecting a Connection to Explore its Database

Select the connection that points to the data you want to explorer.

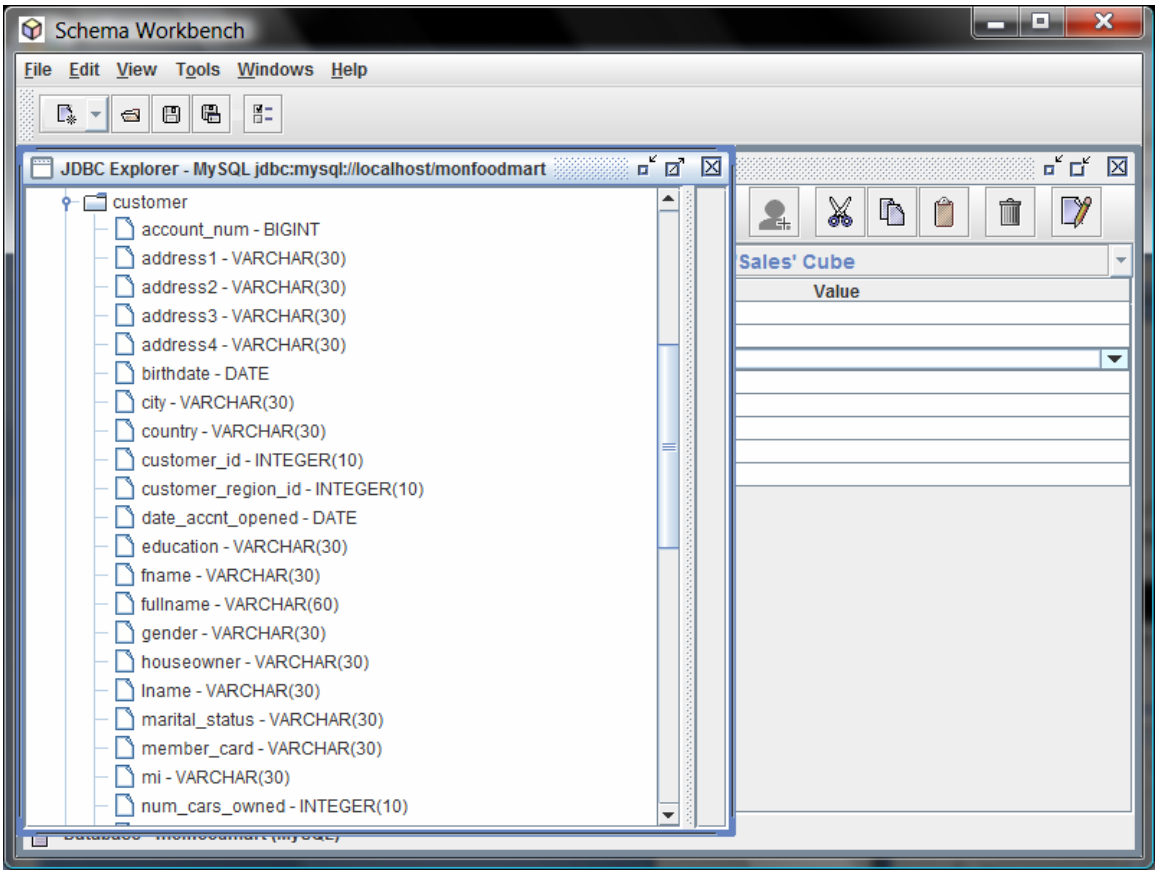

Figure 14 Exploring a Database

3.5 Setting up Schemas

The process of creating an analysis view starts in the workbench and ends in JasperAnalysis.

In the workbench:

1. Define your database connection.
2. Create a valid schema by adding elements using the Schema Explorer. Saving creates the <schema_name>.xml
3. Create access grants by adding roles to the schema in the Schema Explorer. Saving creates the <schema_name>.agxml when the workbench detects roles in the schema.
4. Test the schema against its connection.
5. Run MDX queries against schema + connection. The workbench can generate queries automatically. Saving creates a file of MDX queries. These can be pasted into JasperAnalysis.

In JasperAnalysis:

1. Create a data source that points to the data described by your OLAP schema. You can use the values defined in the database connection in the workbench to populate the data source fields in JasperServer.
2. Create a Mondrian connection that points to the data source and includes the OLAP schema XML file.

   If you use JasperAnalysis Professional, you can store the OLAP schema in the repository. Use the New Resource tool bar button and select Other >XML File, and upload the OLAP schema you created in the workbench.
3. If you use JasperAnalysis Professional and need to secure your data, edit your Mondrian connection to include the AGXML file you created in the workbench.
4. Create an analysis view that points to your Mondrian connection Mondrian connection and includes one of the MDX queries you defined in the workbench.

Note that, if you want to use XML/A to access this data, create a Mondrian XML/A connection in the remote instance of JasperAnalysis, and point it to the Mondrian connection created in step 2.

3.5.1 Getting Started

An OLAP schema defines multi-dimensional OLAP metadata in order to query a relational database structured as a star or snowflake schema. Such database structures have one or more fact tables with a large number of rows with a set of values, known as measures, and dimension tables that categorize those facts.

This metadata is used by the Mondrian engine to generate SQL automatically to satisfy MDX queries. MDX is an OLAP optimized query language for doing everything from simple totals across multiple categories to advanced statistical and time series analysis. A variety of Web browser and Excel tools can be used by end users to *slice and dice* the data in the underlying star schema database.

To implement a Mondrian based environment, you must define the database and populate it with the correct data. Often the source data is in one or more transactional systems that must be transformed and inserted into a database and structure that Mondrian can use. Note that Mondrian does not change any data in the underlying database.

3.5.2 Defining Database Connection

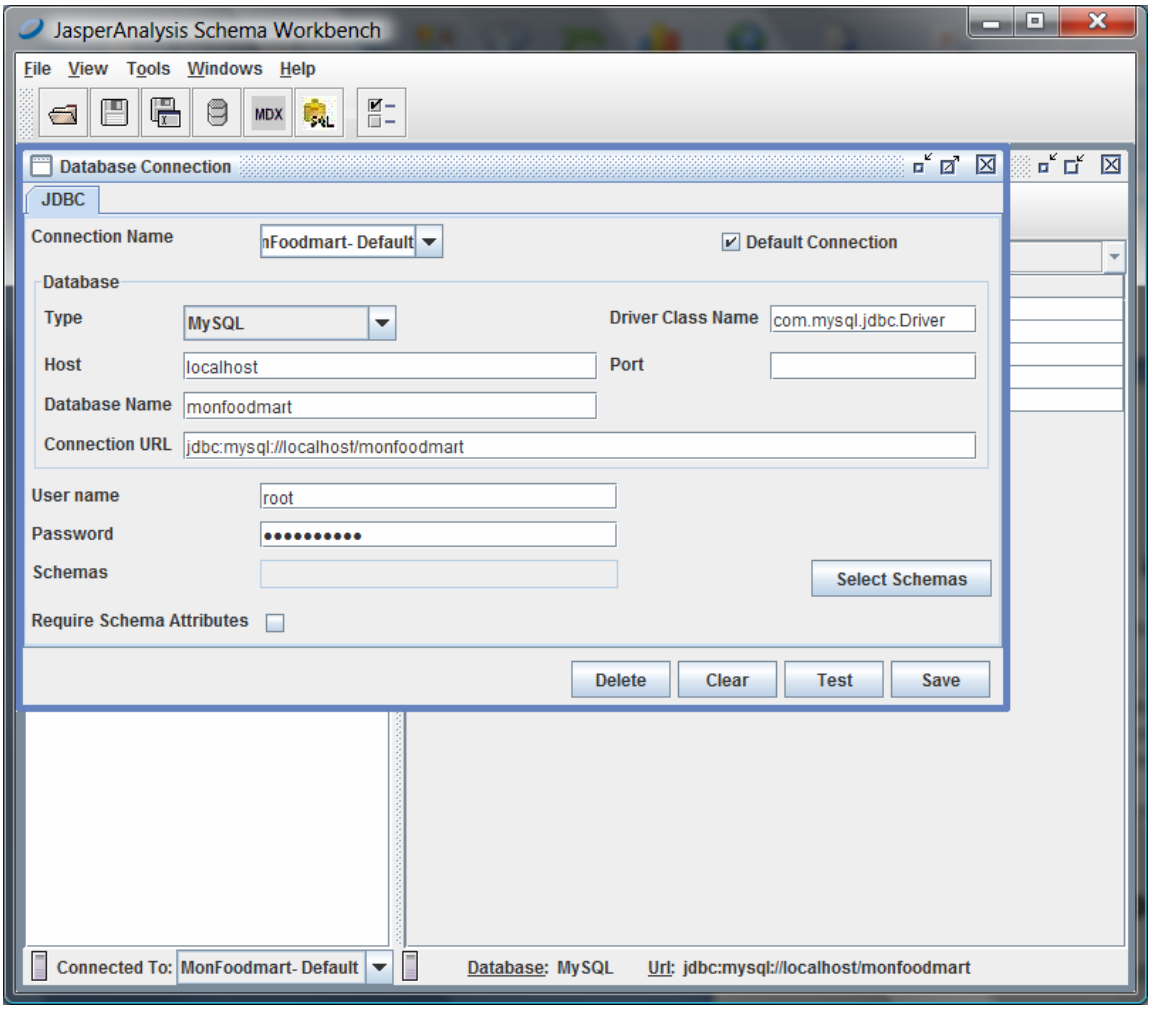

Figure 15 Setting Up a Database Connection

Select your database type. The **Type** drop-down lists the database types Mondrian supports. Specify the host, port and database name, or edit the **Connection URL** field directly. Enter the user name and password if needed, and specify schemas if your database supports them.

Name the connection, test it, and save it.

3.5.3 Creating a Simple Schema

Figure 16 shows a simple schema with one cube, two dimensions, and one measure.

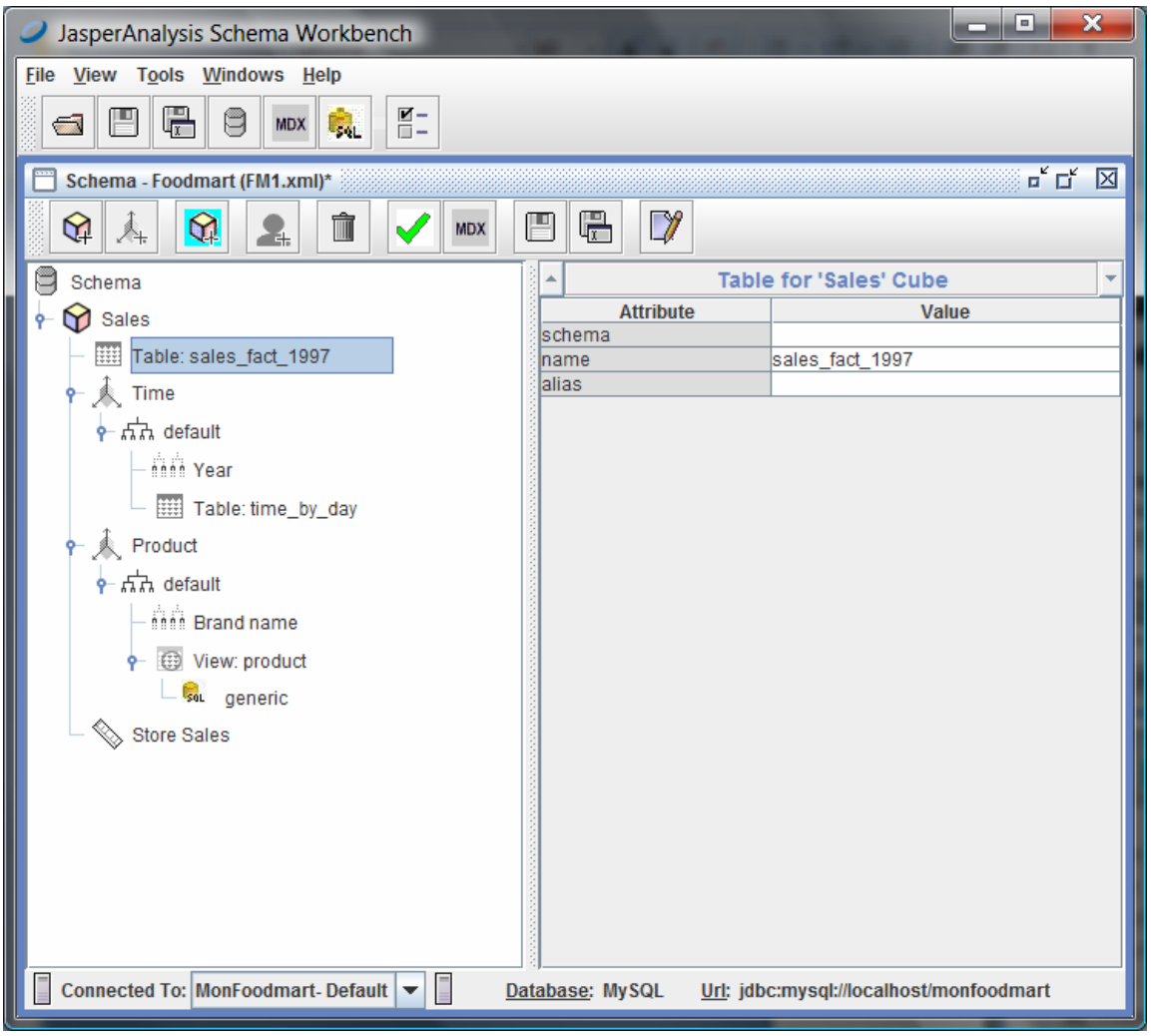

Figure 16 A Simple Dimension

Figure 16 shows a Sales cube. In Mondrian, each cube revolves around one fact table; in this case, it's sales_fact_1997. Time and Product are the dimensions. Store Sales is the measure.

Dimensions

The Time dimension is based on the time_by_day table:

- Its foreign key on the sales_fact_1997 table is time_id.
- Its primary key on the time_by_day table is time_id.
- It has one hierarchy, which does not need to be named.
- It has a single level to categorize facts (Year) that is based on the numeric the_year column.
- It does not have an *All Times* level – you can't summarize the facts across all time. While the All level is a commonly found in many dimensions, it doesn't always make sense to summarize data across an entire hierarchy; this is the case with the Time dimension.
- Time dimensions are special: special MDX functions can provide quarter- or year-to-date summaries, as well as period-over-period calculations.

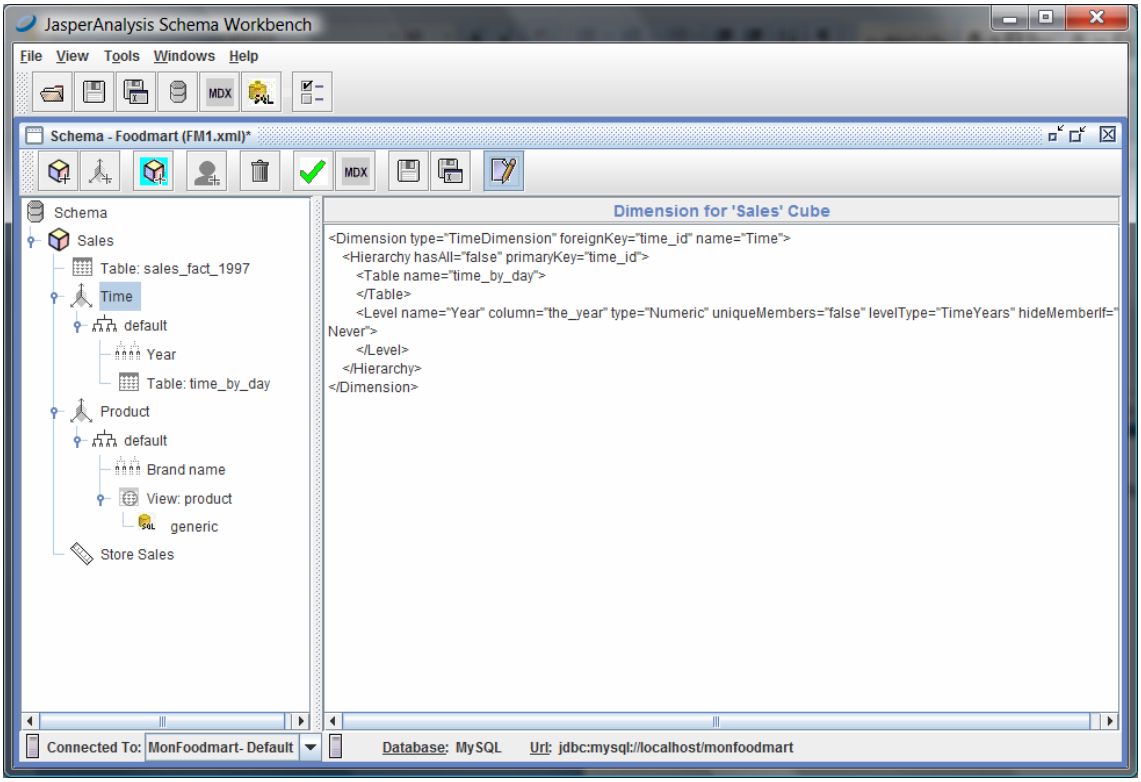

Figure 17 A Typical Time Dimension

The Product dimension is an example of a snowflake dimension, where the dimension is made up of several tables joined together. In this case, the dimension is made up of a View: a SQL query that joins the tables that make up the full dimension.

- Its foreign key on the sales_fact_1997 table is product_id.
- Its primary key in the SQL query is product _id.
- It has one hierarchy, which does not need to be named.
- It has a single level to categorize facts, Brand name, based on the String brand_name column in the SQL query result.
- It has an All Products level – you can summarize the facts across all products.

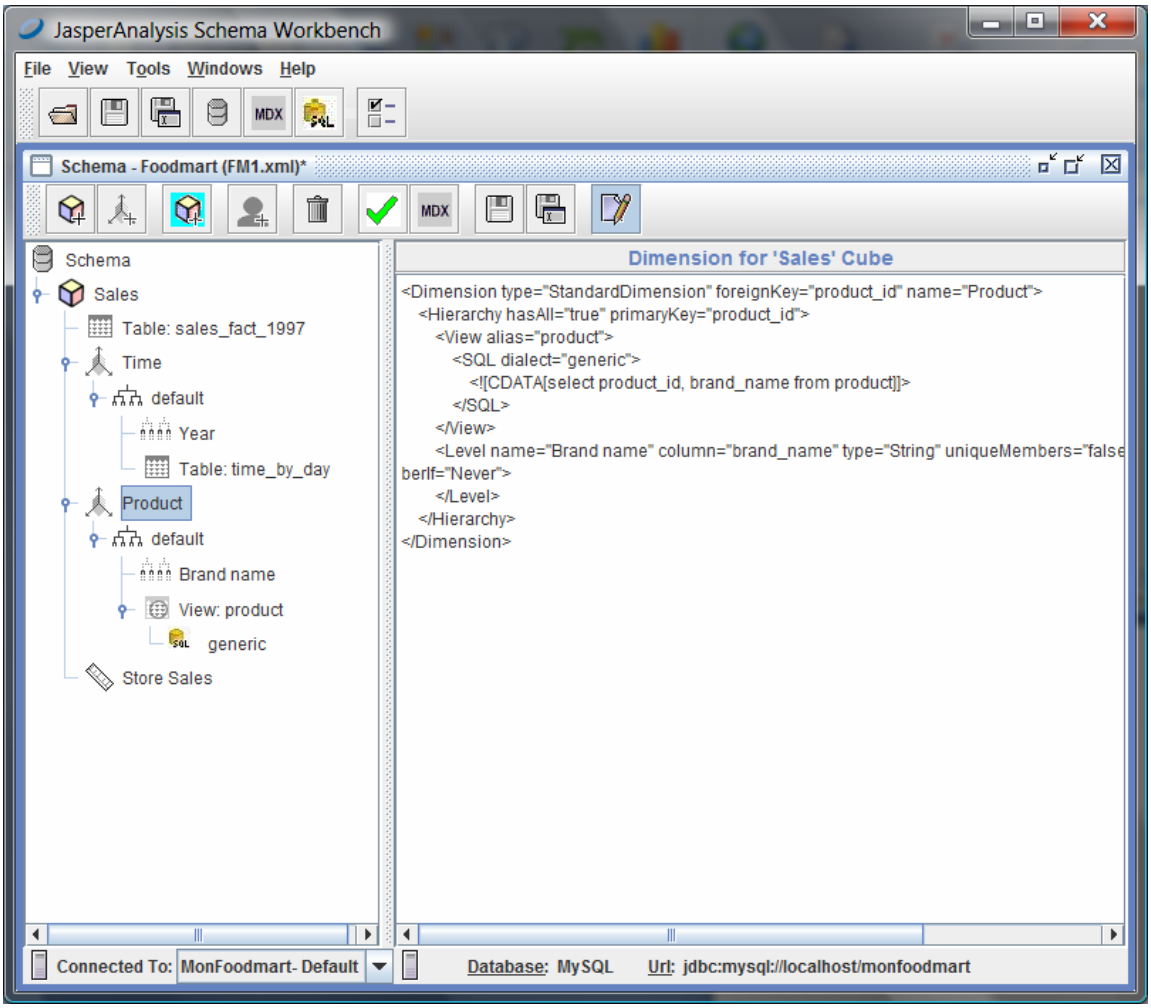

Figure 18 A Product Dimension

This could have been modeled as a Join, which expects that the tables that make up the snowflake are joined on a single column.

Measures

Store Sales is a measure on the fact table that can be summarized and manipulated according to the dimensions. It is based on the store_sales column and should be summed.

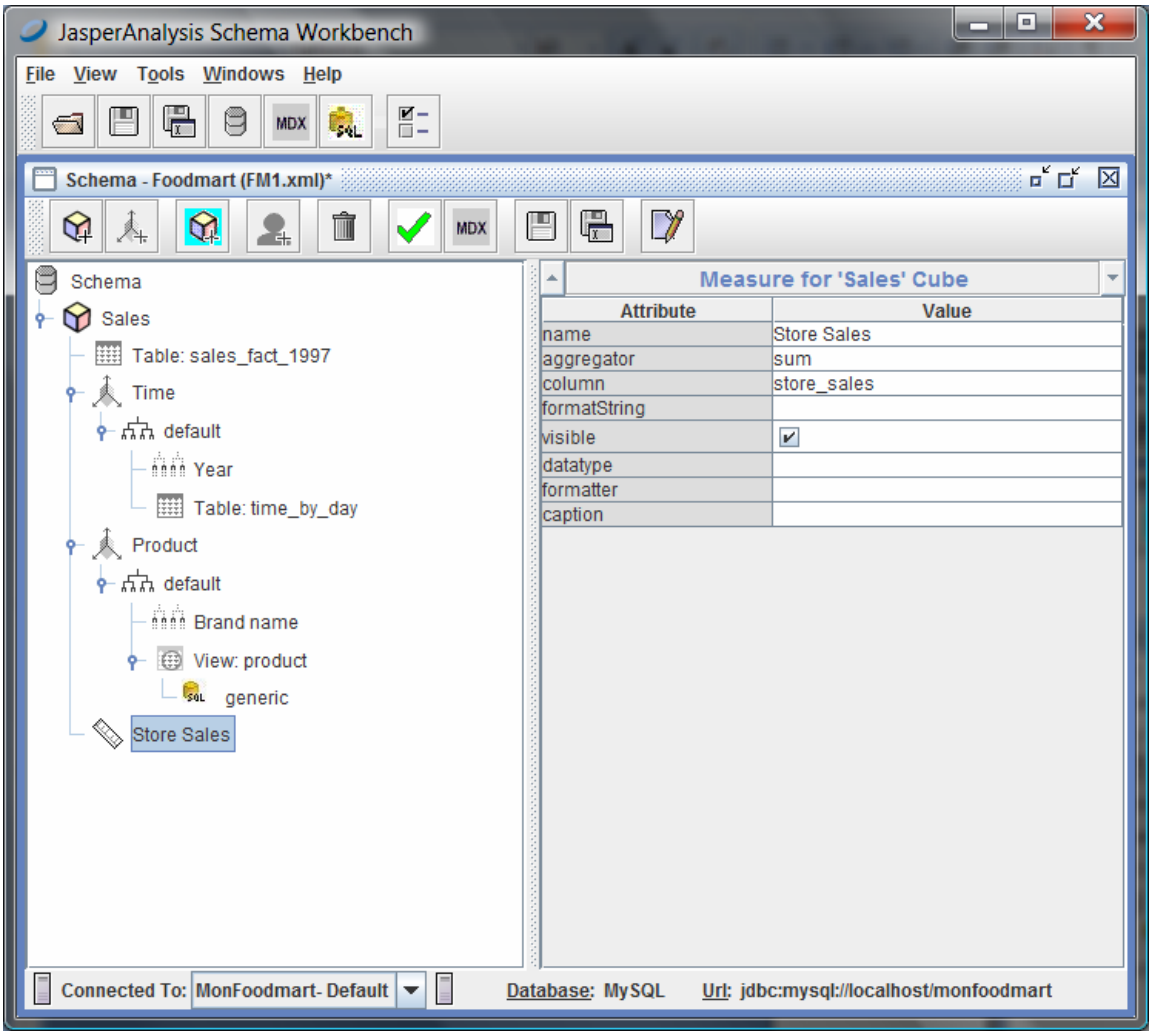

Figure 19 A Measure for Products

Testing the Schema

Click **(Test Schema)**; if the test succeeds, a message appears indicating that the schema passed validation:

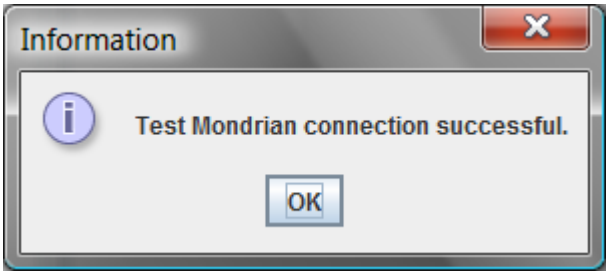

Figure 20 Connection Test Successful

Click

(**New MDX Query**) to open the MDX editor:

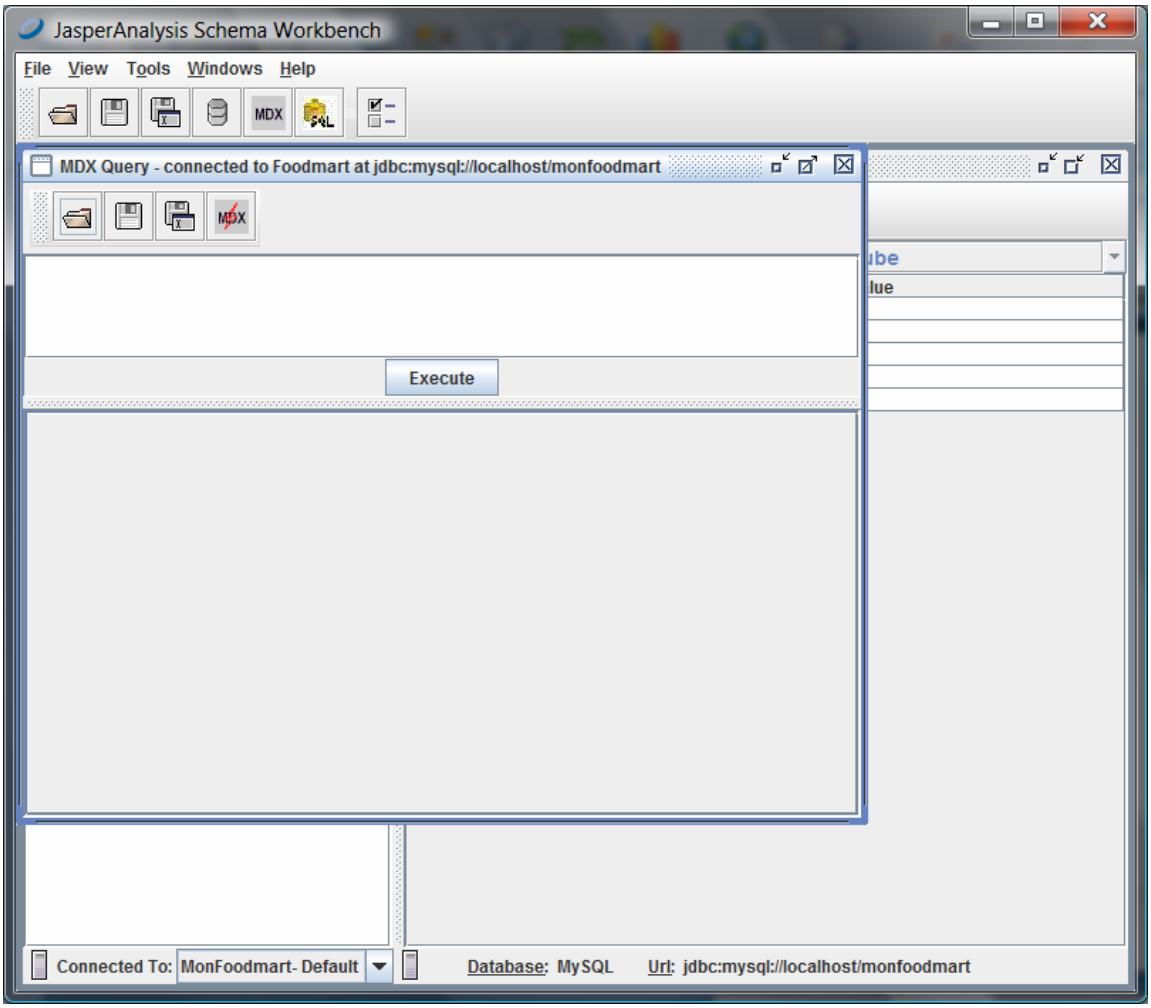

Figure 21 MDX Editor

Click **(Generate MDX Queries)** to automatically create MDX queries:

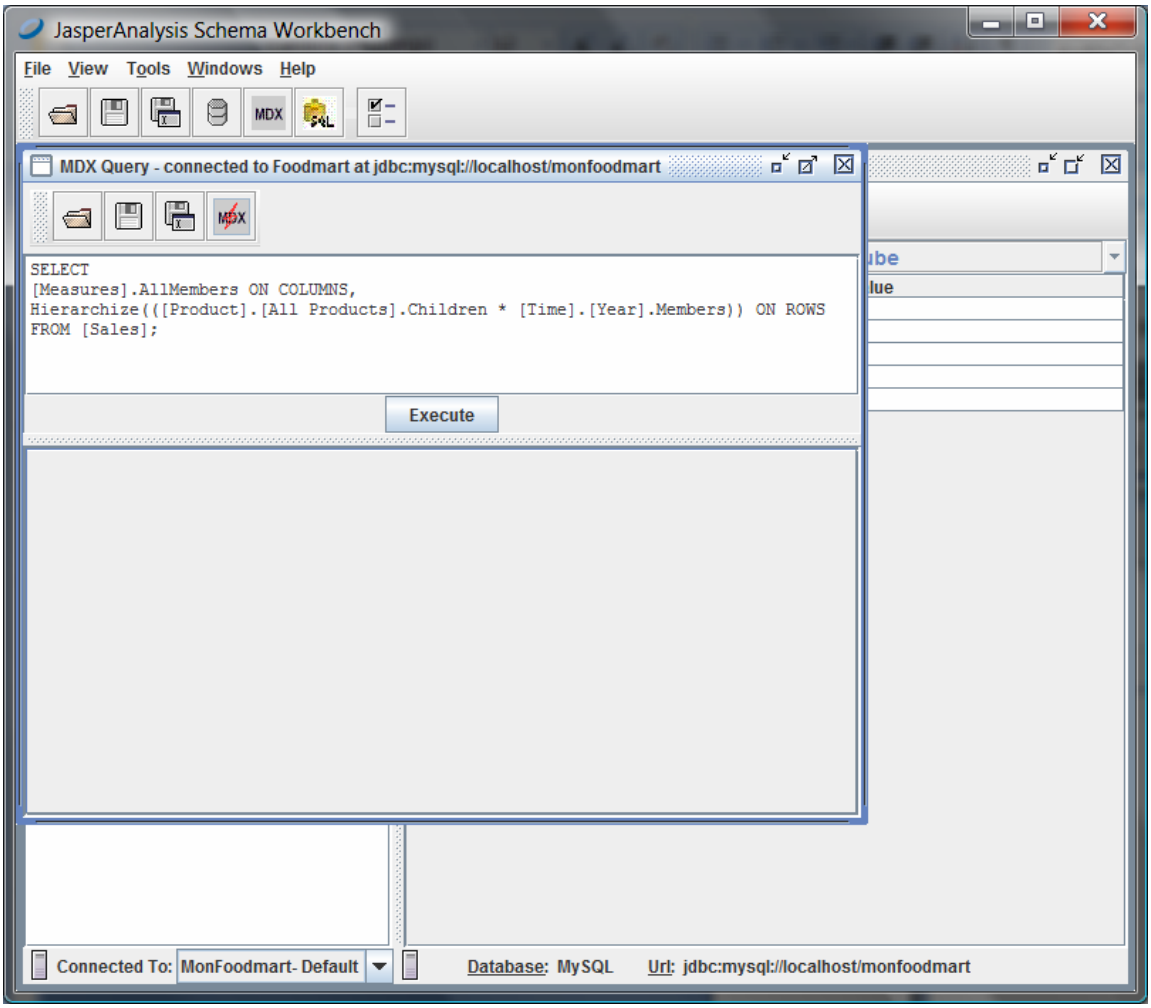

Figure 22 A Generated MDX Query

Click **Execute** to run the MDX query against your schema; results appear below the query:

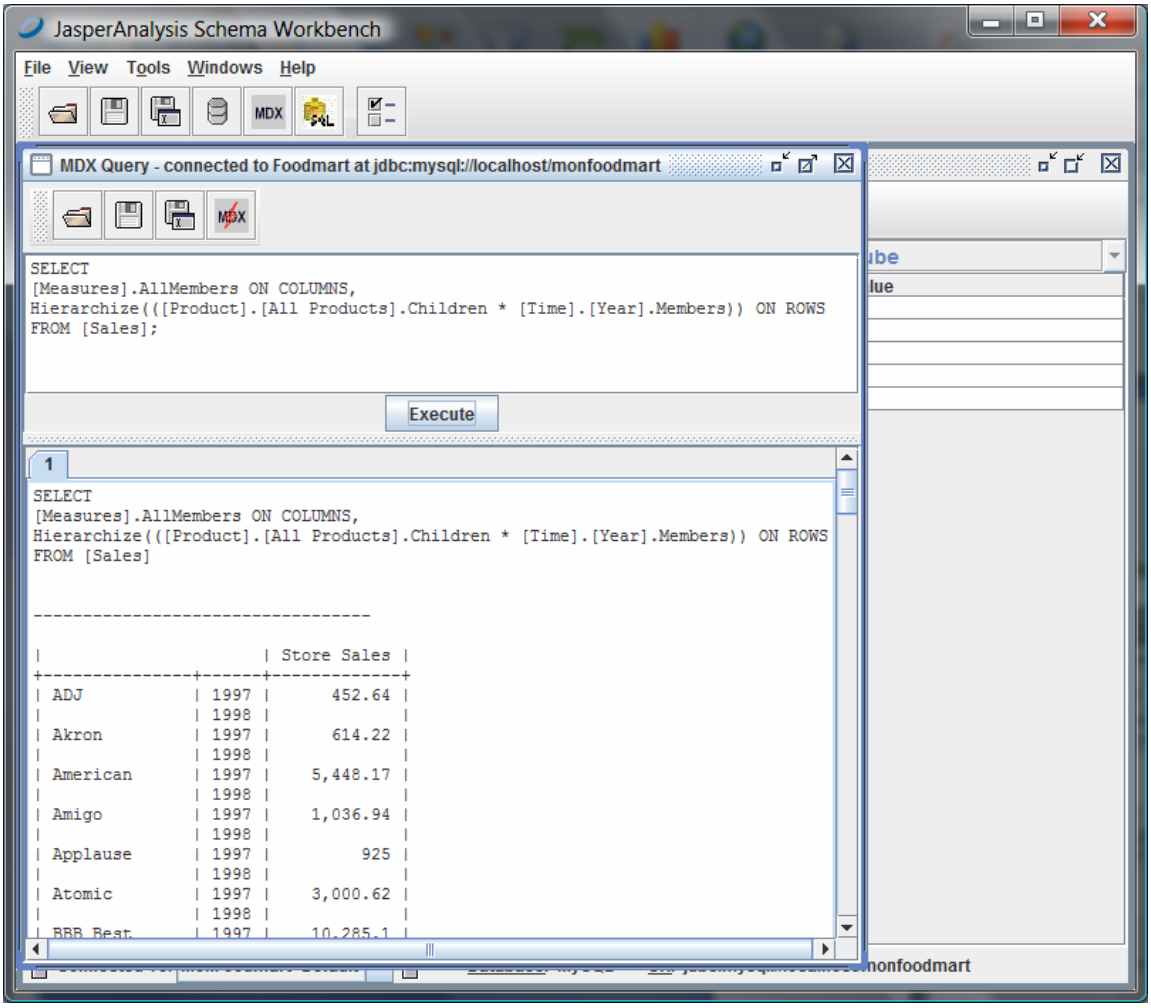

Figure 23 An MDX Query and its Results

Close the MDX editor and click **Save** in the Schema Explorer:

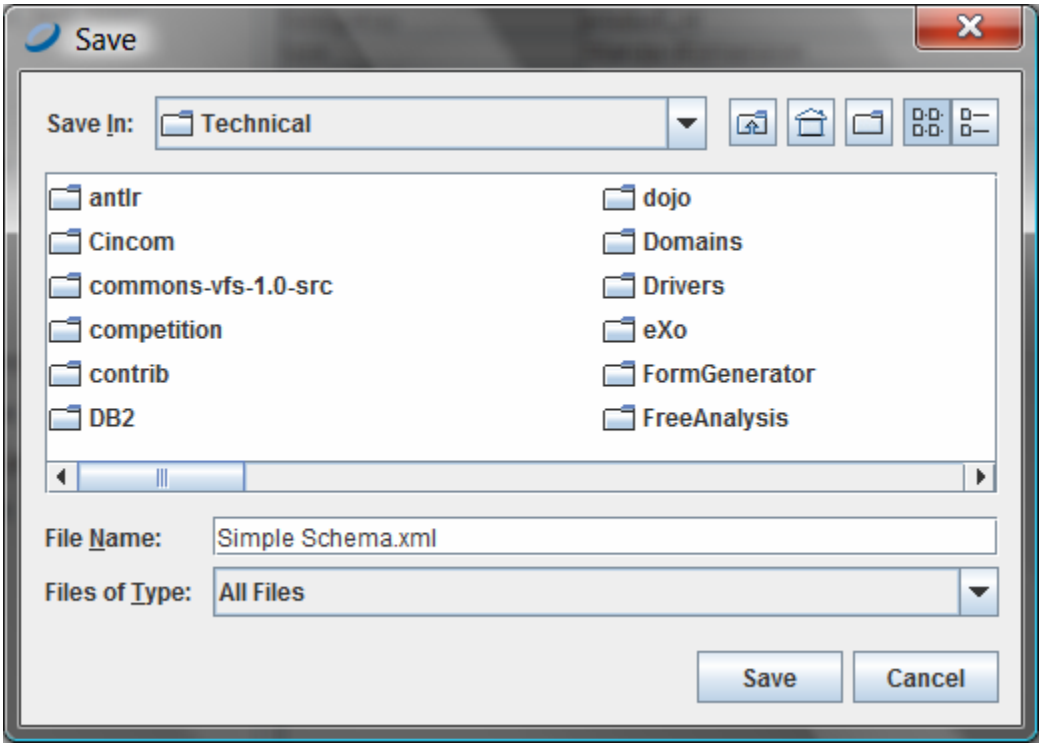

Figure 24 Saving an OLAP Schema

Now you can upload the schema and create an analysis view that uses it in combination with one of the generated MDX queries and, optionally, an AGXML security file (if you use the Professional editions of the tools.

For more information on how to create Mondrian connections and analysis view in JasperAnalysis, refer to the *JasperAnalysis Professional User Guide* and the *JasperAnalysis Ultimate Guide*.