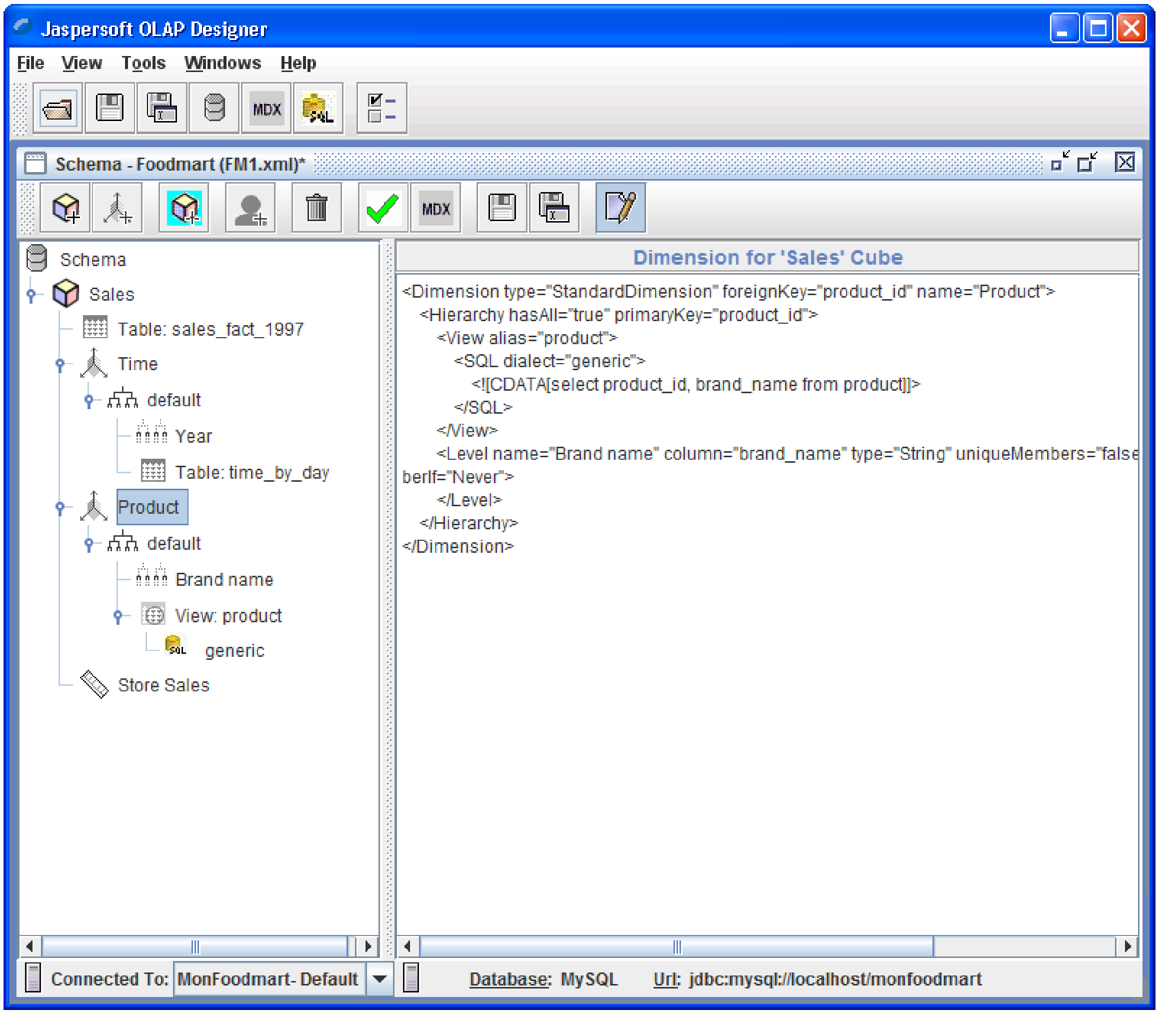
OLAP Workbench Guide
Version 4.0

Table of Contents

1. Introduction



```
<Dimension type="StandardDimension" foreignKey="product_id" name="Product">
   <Hierarchy hasAll="true" primaryKey="product_id">
      <View alias="product">
         <SQL dialect="generic">
            <![CDATA[select product_id, brand_name from product]]>
         </SQL>
      </View>
      <Level name="Brand name" column="brand_name" type="String" uniqueMembers="false
berIf="Never">
      </Level>
   </Hierarchy>
</Dimension>
```

Connected To: MonFoodmart- Default

Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

Figure 1 Jaspersoft OLAP Designer

The Jaspersoft OLAP Designer 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.
- MDX Query: Tests MDX queries against schema and database. Automatically generates example MDX queries.
- JDBC Explorer: Browses underlying database’s structure.

The Designer comes in two editions:

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

See the Mondrian Technical Guide for information about schemas. It is found in the /docs directory of the Designer installation. Once you have a schema XML file, you can upload it into any application that uses Mondrian, such as JasperReports Server.

2. Installing and Configuring the Designer

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

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

- 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 Designer installation, then select **Other** as the type of database when you define a connection.

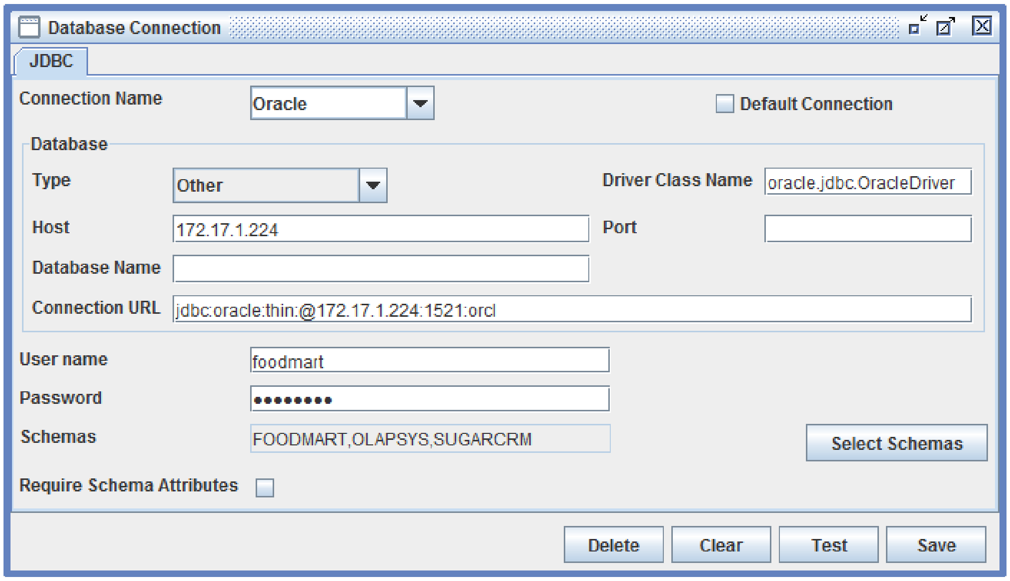
In the Jaspersoft OLAP Community Edition, JDBC drivers are provided for MySQL, PostgreSQL and Ingres. You can include your cube database's JDBC driver in the /drivers directory in the Designer installation, so that the Designer can use it.

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

Start the Designer by running Workbench.bat (Windows) or Workbench.sh (for Unix/Linux) in the directory where you installed the Designer. 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 .**



Connection Name: Oracle
☐ Default Connection

Database
Type: Other
Driver Class Name: oracle.jdbc.OracleDriver
Host: 172.17.1.224
Port:
Database Name:
Connection URL: jdbc:oracle:thin:@172.17.1.224:1521:orcl

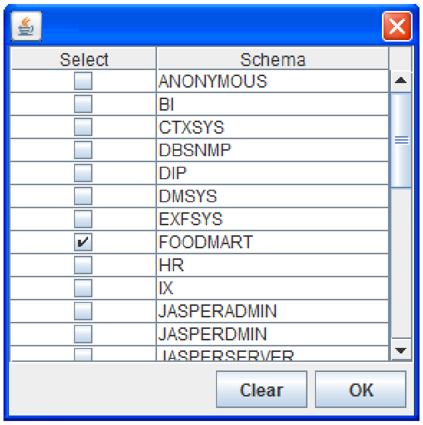
User name: foodmart
Password: ••••••••
Schemas: FOODMART,OLAPSYS,SUGARCRM
Require Schema Attributes ☐

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 drop-down, define the parameters, test the connection, and save it in the list.

By default, the Jaspersoft OLAP Designer automatically populates data for certain RDBMS types (such as MySQL and DB2). Selecting one 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 **Other** in the **Type** field to enter values specific to one of the databases. When you enter a connection URL, the Host, Port and Database Name values are updated, provided the Designer can insert the correct values automatically.

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



Select	Schema
□	ANONYMOUS
□	BI
□	CTXSYS
□	DBSNMP
□	DIP
□	DMSYS
□	EXFSYS
☑	FOODMART
□	HR
□	IX
□	JASPERADMIN
□	JASPERDMIN
□	JASPERSERVER

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 Designer requires the schema name to be entered for all tables and joins in the OLAP schema. This option is useful in several cases, including these:
    - The data described by your OLAP schema spans multiple database schemas.
    - You are using an RDBMS, such as Oracle and DB2, which creates a database schema for each connecting user. When the user connects, the RDBMS retrieves 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 Designer can access the data source properly. It looks for the correct database driver (in a JAR) in the Designer's /drivers directory and uses the properties entered in the Connection window to try connect to the database. This ensures that the connection can be used to validate schemas and explore tables and columns.
- The **Save** button saves the connection; click it once if 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 the .schemaWorkbench directory, including these files:

- 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 runtime properties. For details about these settings, see the Mondrian Technical Guide.

If you are using a using a database that requires a JDBC driver and the driver is not installed already, you must put the JDBC JAR files in the /drivers directory.

3. Using the Designer

3.1 Main Window

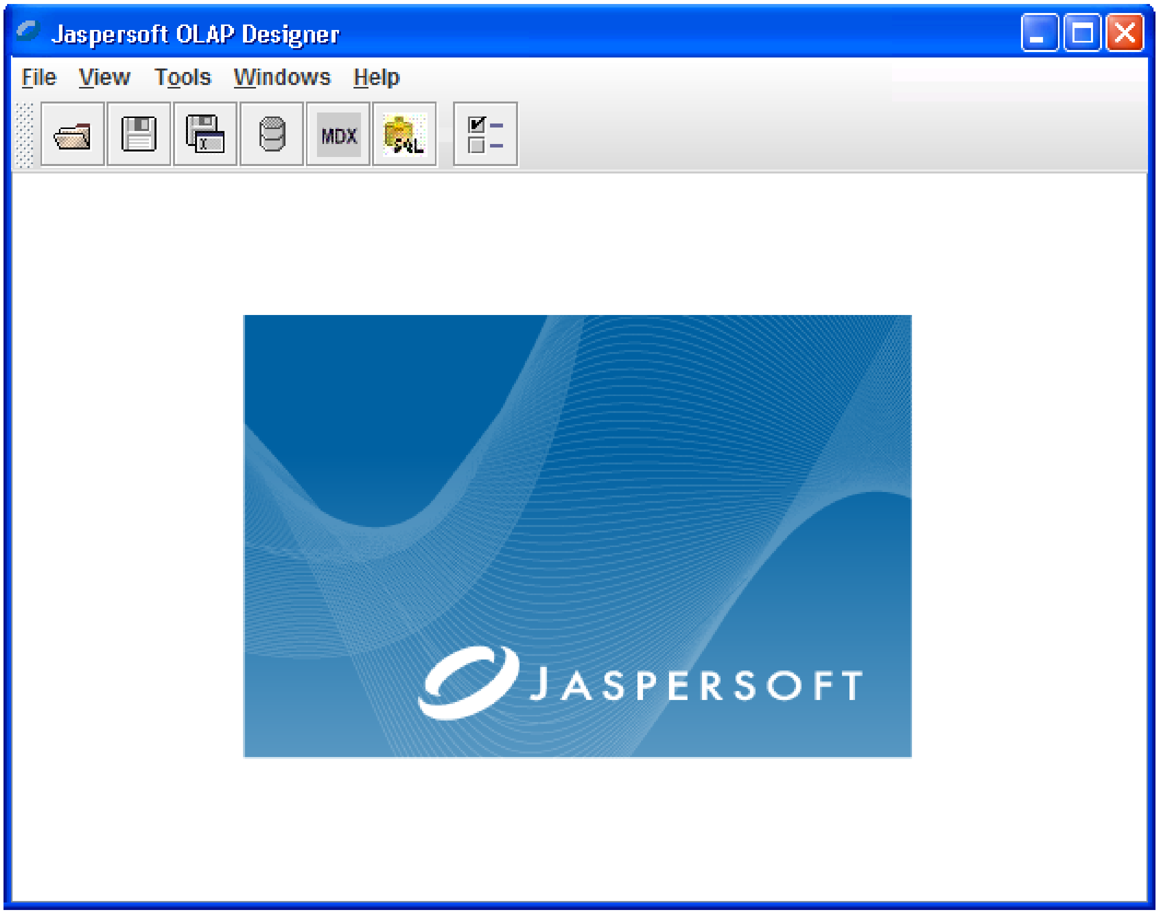


Figure 4 Main Window

The top-level menus and tool bars in the Designer provide commonly-used functions:

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 Designer
View	**View XML**	Switch between graphical and XML view of the current schema

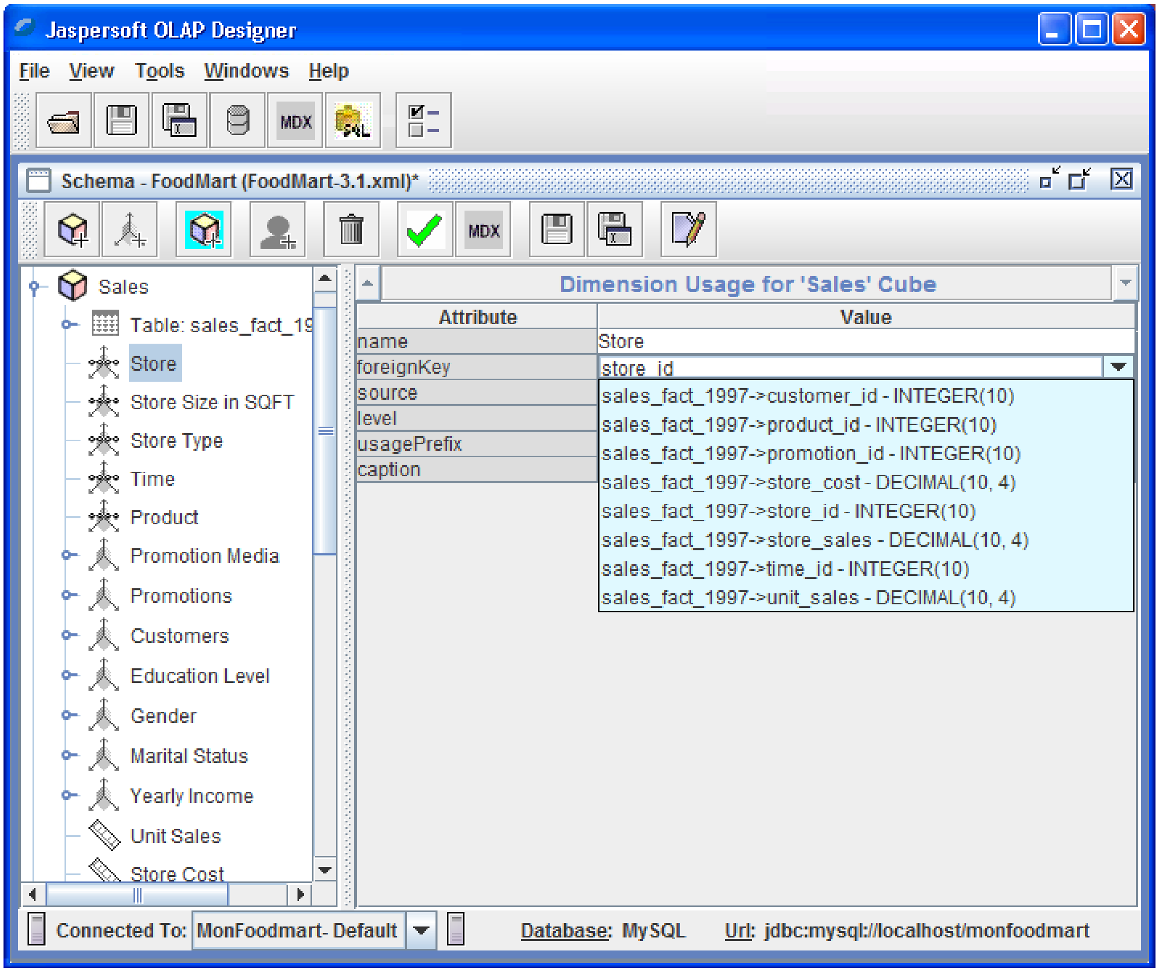
Menu	Option	Description
Tools	**Connection**	Maintain JDBC connections; this option is also available from the tool bar.
Windows	**List of Open Files**	Click a file name to maximize it and bring it to the front
	Cascade Windows	Arrange windows from top left to bottom right
	Tile Windows	Size and show all windows in current display
	Maximize All	Maximize all open windows
	Minimize All	Minimize all open windows
	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 and edit the OLAP schemas that Jaspersoft OLAP 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 required values and values selected from lists of values. For example, when you define a join, the drop-down only includes fields in tables in that join.

To launch the Schema Explorer, click **New > Schema**.



Attribute	Value
name	Store
foreignKey	store_id
source	
level	
usagePrefix	
caption	

sales_fact_1997->customer_id - INTEGER(10)
sales_fact_1997->product_id - INTEGER(10)
sales_fact_1997->promotion_id - INTEGER(10)
sales_fact_1997->store_cost - DECIMAL(10, 4)
sales_fact_1997->store_id - INTEGER(10)
sales_fact_1997->store_sales - DECIMAL(10, 4)
sales_fact_1997->time_id - INTEGER(10)
sales_fact_1997->unit_sales - DECIMAL(10, 4)

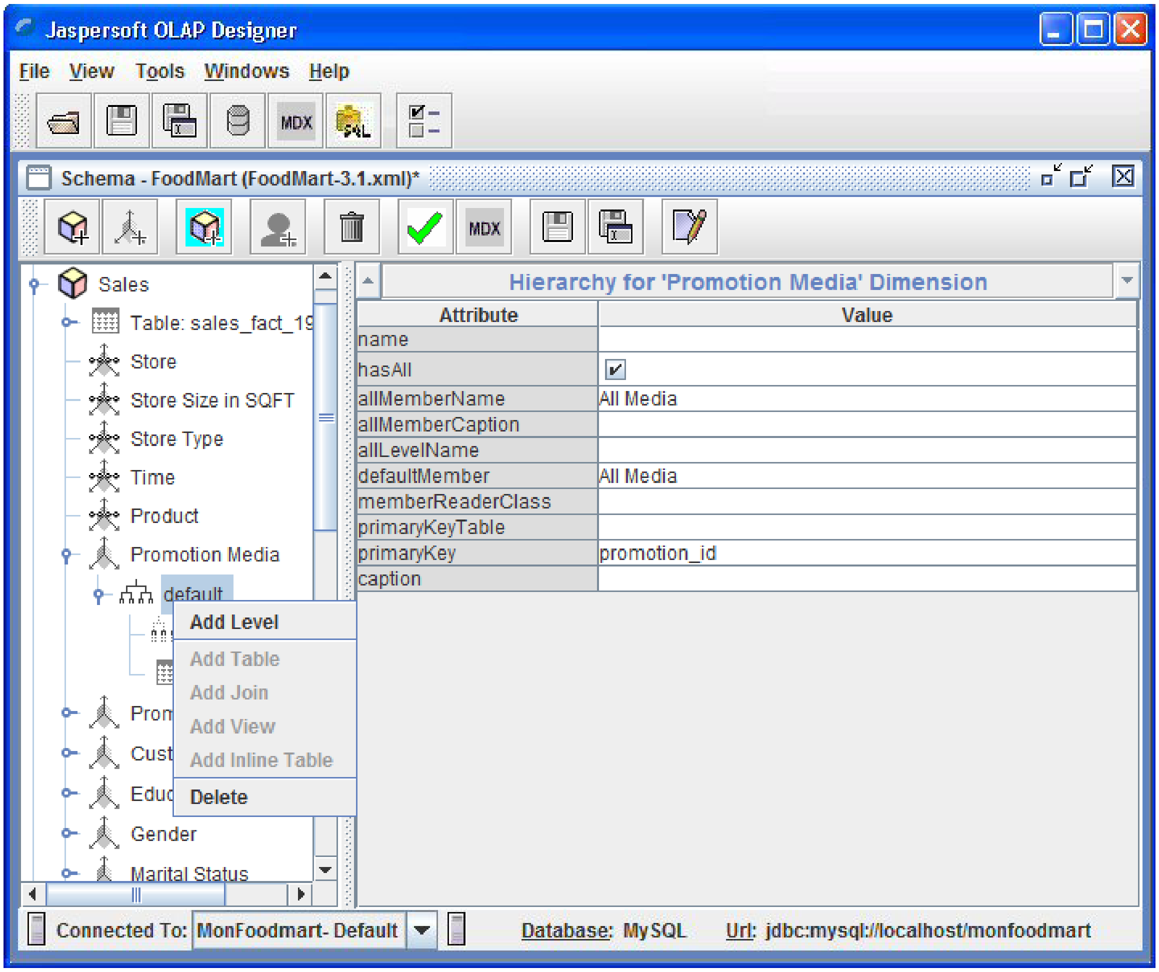
Connected To: MonFoodmart- Default Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

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

To edit an element, right-click it in the schema tree on the left side of the window. A pop-up context menu of actions relevant for the selected element appears.



Hierarchy for 'Promotion Media' Dimension

Attribute	Value
name	
hasAll	☑
allMemberName	All Media
allMemberCaption	
allLevelName	
defaultMember	All Media
memberReaderClass	
primaryKeyTable	
primaryKey	promotion_id
caption	

Connected To: MonFoodmart- Default

Database: MySQL

Url: jdbc:mysql://localhost/monfoodmart

Figure 6 Context Menu for **Sales > Promotion Media > default**

The final level of schema validation is the successful execution of MDX queries.

Use the MDX tool bar button to launch an MDX query editor against the current schema.

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 Designer, you are prompted to save it.

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

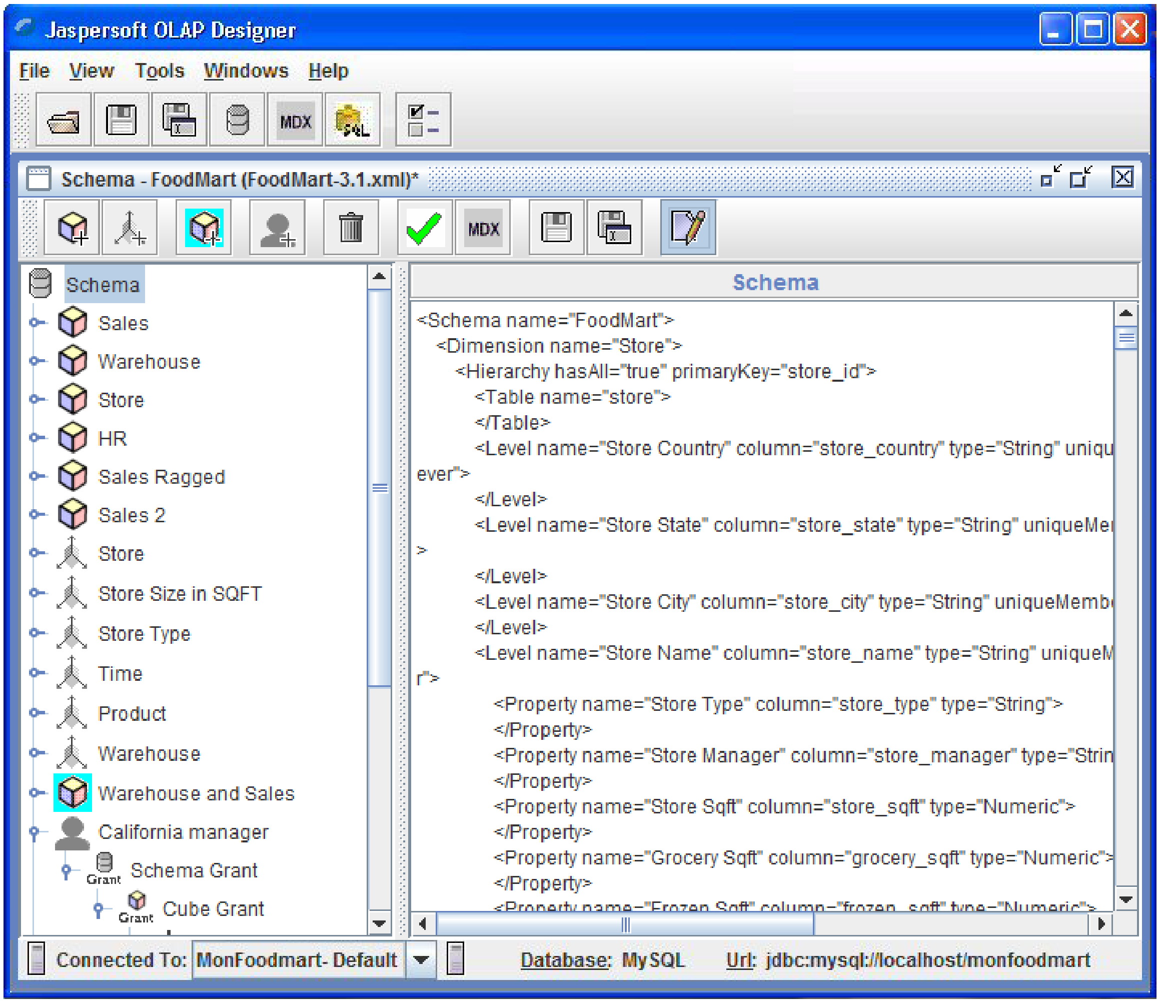
- Cube
- Dimension
- Virtual Cube
- Role - Role definitions can only be used in conjunction with the Professional Edition of Jaspersoft OLAP. They have no effect in Jaspersoft OLAP Community Edition.

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. This can also be toggled from the top level menu **View > XML**. You can't edit the XML directly.



```
<Schema name="FoodMart">
  <Dimension name="Store">
    <Hierarchy hasAll="true" primaryKey="store_id">
      <Table name="store">
      </Table>
      <Level name="Store Country" column="store_country" type="String" uniqu
ever">
      </Level>
      <Level name="Store State" column="store_state" type="String" uniqueMe
>
      </Level>
      <Level name="Store City" column="store_city" type="String" uniqueMemb
      </Level>
      <Level name="Store Name" column="store_name" type="String" uniqueM
r">
        <Property name="Store Type" column="store_type" type="String">
        </Property>
        <Property name="Store Manager" column="store_manager" type="Strin
        </Property>
        <Property name="Store Sqft" column="store_sqft" type="Numeric">
        </Property>
        <Property name="Grocery Sqft" column="grocery_sqft" type="Numeric">
        </Property>
        <Property name="Frozen Sqft" column="frozen_sqft" type="Numeric">
```

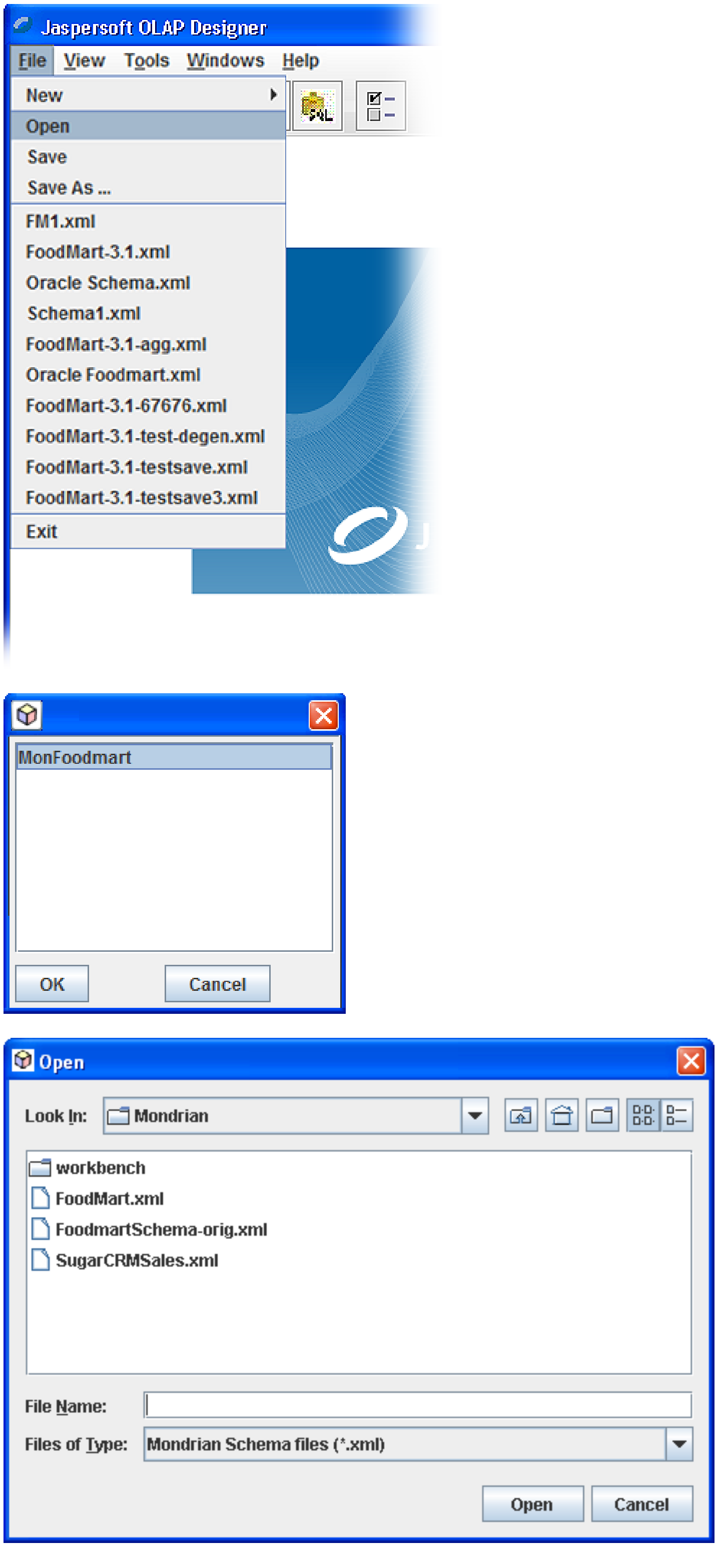
Connected To: MonFoodmart- Default   Database: MySQL   Url: jdbc:mysql://localhost/monfoodmart

Figure 7 XML View of an OLAP Schema

At the bottom of the window, a drop-down of available JDBC connections shows the current connection, the database type (determined from the JDBC driver), and the JDBC connection URL. You can change the connection you want to validate.

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.



Look In: Mondrian

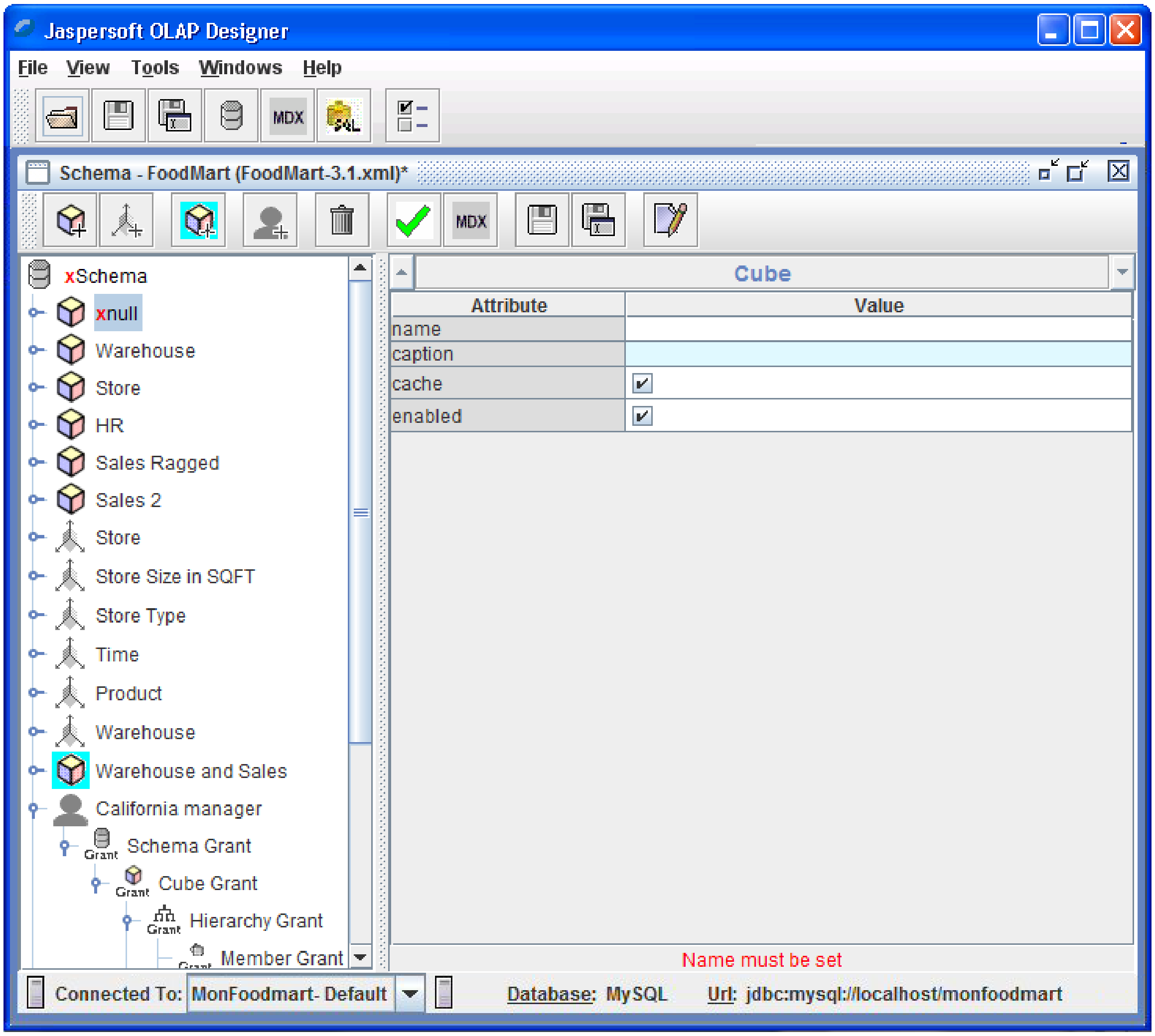
File Name:

Files of Type: Mondrian Schema files (*.xml)

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 Designer reports any errors it detects.

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



Attribute	Value
name	
caption	
cache	☑
enabled	☑

Name must be set

Connected To: MonFoodmart- Default

Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

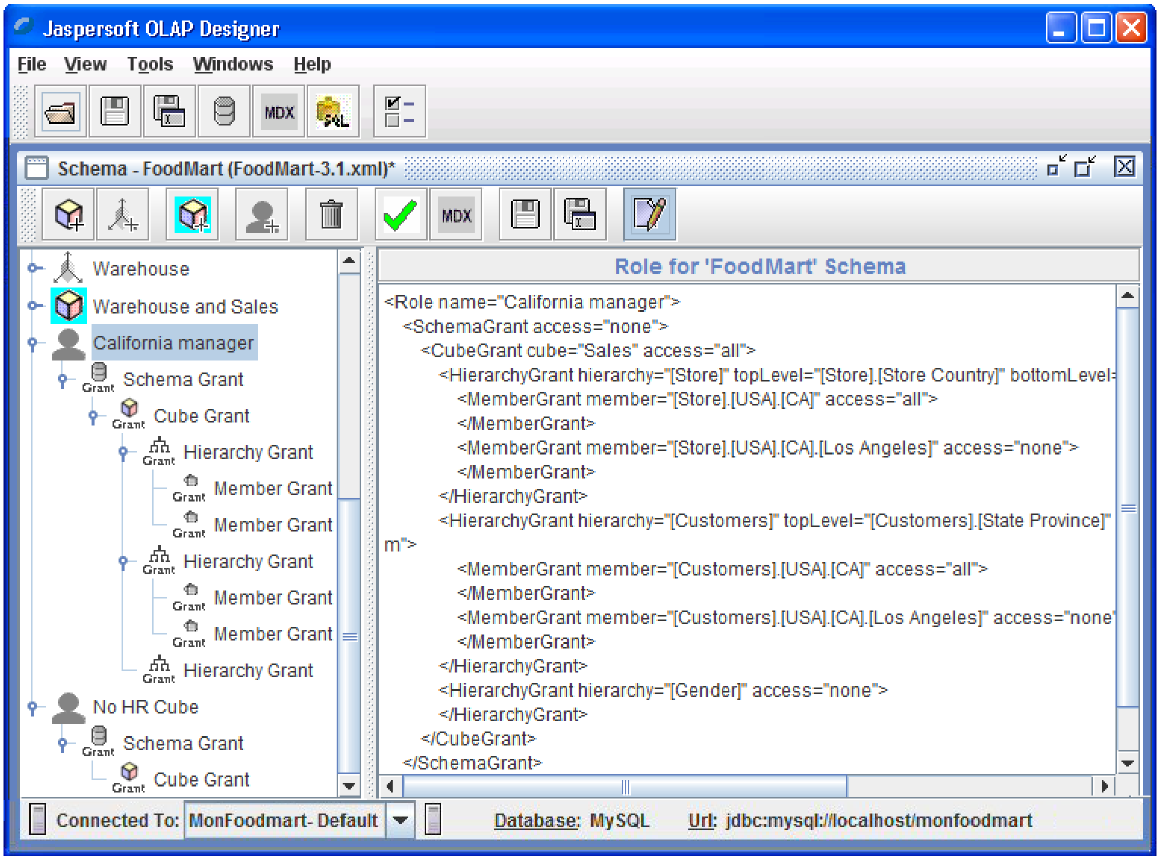
Figure 9 Status Message Showing an Error in Red

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 a sub-elements is in error, the message `<Element Type> is invalid` appears.

3.2.5 Access Grants for Jaspersoft OLAP Professional

This section describes functionality that is available only in JasperReports Server Professional and Enterprise editions. Contact Jaspersoft to obtain the software.

Jaspersoft OLAP 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 *Jaspersoft OLAP User Guide* and the *Jaspersoft OLAP Ultimate Guide* for details.



```
<Role name="California manager">
  <SchemaGrant access="none">
    <CubeGrant cube="Sales" access="all">
      <HierarchyGrant hierarchy="[Store]" topLevel="[Store].[Store Country]" bottomLevel=
        <MemberGrant member="[Store].[USA].[CA]" access="all">
        </MemberGrant>
        <MemberGrant member="[Store].[USA].[CA].[Los Angeles]" access="none">
        </MemberGrant>
      </HierarchyGrant>
      <HierarchyGrant hierarchy="[Customers]" topLevel="[Customers].[State Province]"
m">
        <MemberGrant member="[Customers].[USA].[CA]" access="all">
        </MemberGrant>
        <MemberGrant member="[Customers].[USA].[CA].[Los Angeles]" access="none"
        </MemberGrant>
      </HierarchyGrant>
      <HierarchyGrant hierarchy="[Gender]" access="none">
      </HierarchyGrant>
    </CubeGrant>
  </SchemaGrant>
```

Connected To: MonFoodmart- Default   Database: MySQL   Url: jdbc:mysql://localhost/monfoodmart

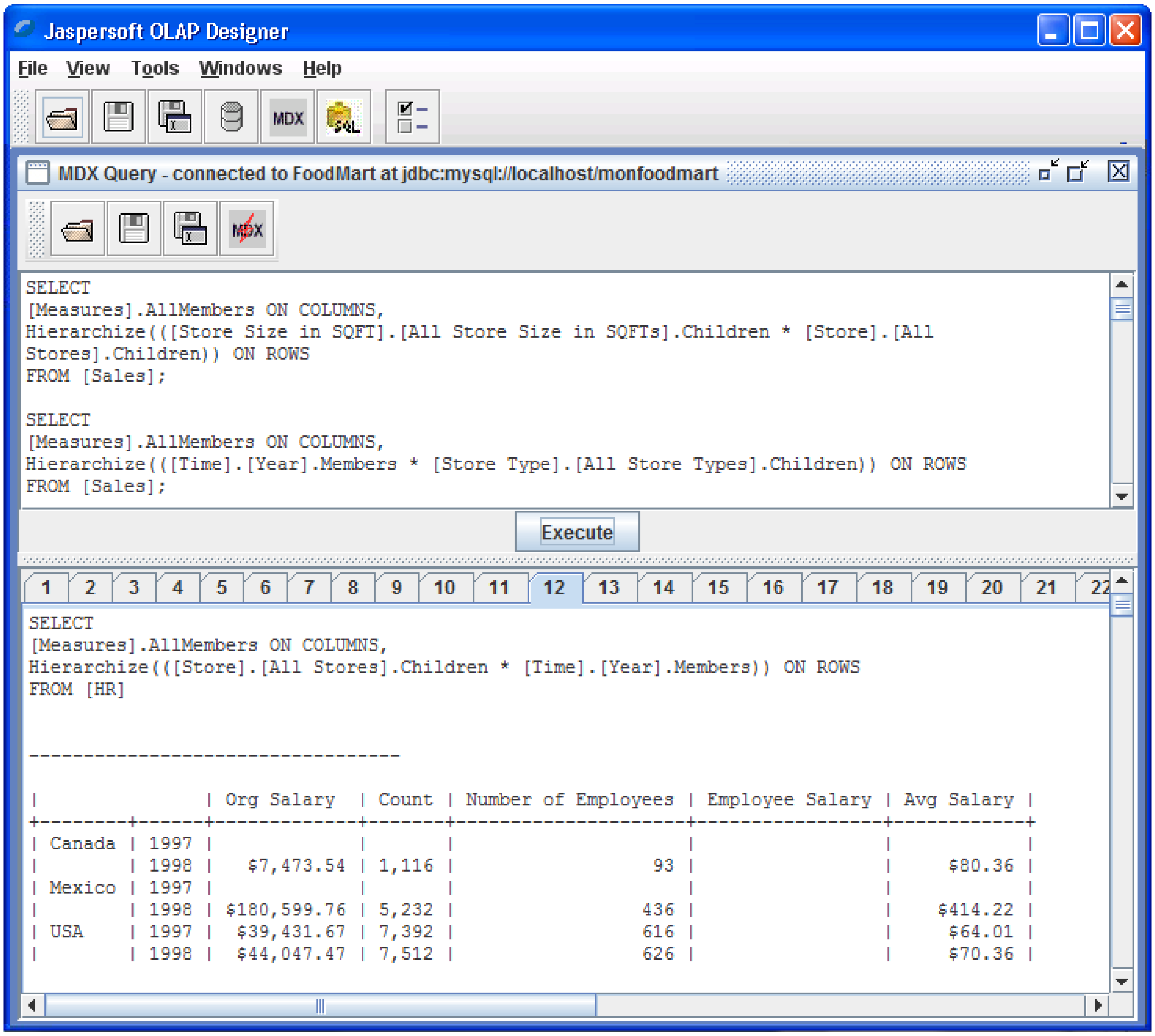
Figure 10 Security File for Jaspersoft OLAP Professional (AGXML Format)

Jaspersoft OLAP Professional relies on AGXML files to define role-based, data-level security. The Professional edition of the OLAP Designer 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 file separate from the OLAP schema itself, and it gives the file the correct file extension (.agxml). This file is saved in 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 OLAP view, or analysis view, in Jaspersoft OLAP Professional.

3.3 MDX Query

Save your schema and click **New > MDX Query** to test MDX queries. 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 and other results are displayed in the Designer window, in the console window, and in the Designer log file.



```
SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Store Size in SQFT].[All Store Size in SQFTs].Children * [Store].[All
Stores].Children)) ON ROWS
FROM [Sales];

SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Time].[Year].Members * [Store Type].[All Store Types].Children)) ON ROWS
FROM [Sales];
```

```
SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Store].[All Stores].Children * [Time].[Year].Members)) ON ROWS
FROM [HR]


----------------------------------

|               | Org Salary  | Count | Number of Employees | Employee Salary | Avg Salary |
+--------+------+-------------+-------+---------------------+-----------------+------------+
| Canada | 1997 |             |       |                     |                 |            |
|        | 1998 |   $7,473.54 | 1,116 |                  93 |                 |     $80.36 |
| Mexico | 1997 |             |       |                     |                 |            |
|        | 1998 | $180,599.76 | 5,232 |                 436 |                 |    $414.22 |
| USA    | 1997 |  $39,431.67 | 7,392 |                 616 |                 |     $64.01 |
|        | 1998 |  $44,047.47 | 7,512 |                 626 |                 |     $70.36 |
```

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 (#) at the very beginning on the comment. When **Execute** is clicked, each query is run and the results are displayed. Each query's results are displayed on a separate tab in the table format shown in Figure 11. Exceptions, such as bad syntax, are displayed.

The MDX Query Editor provides these features:

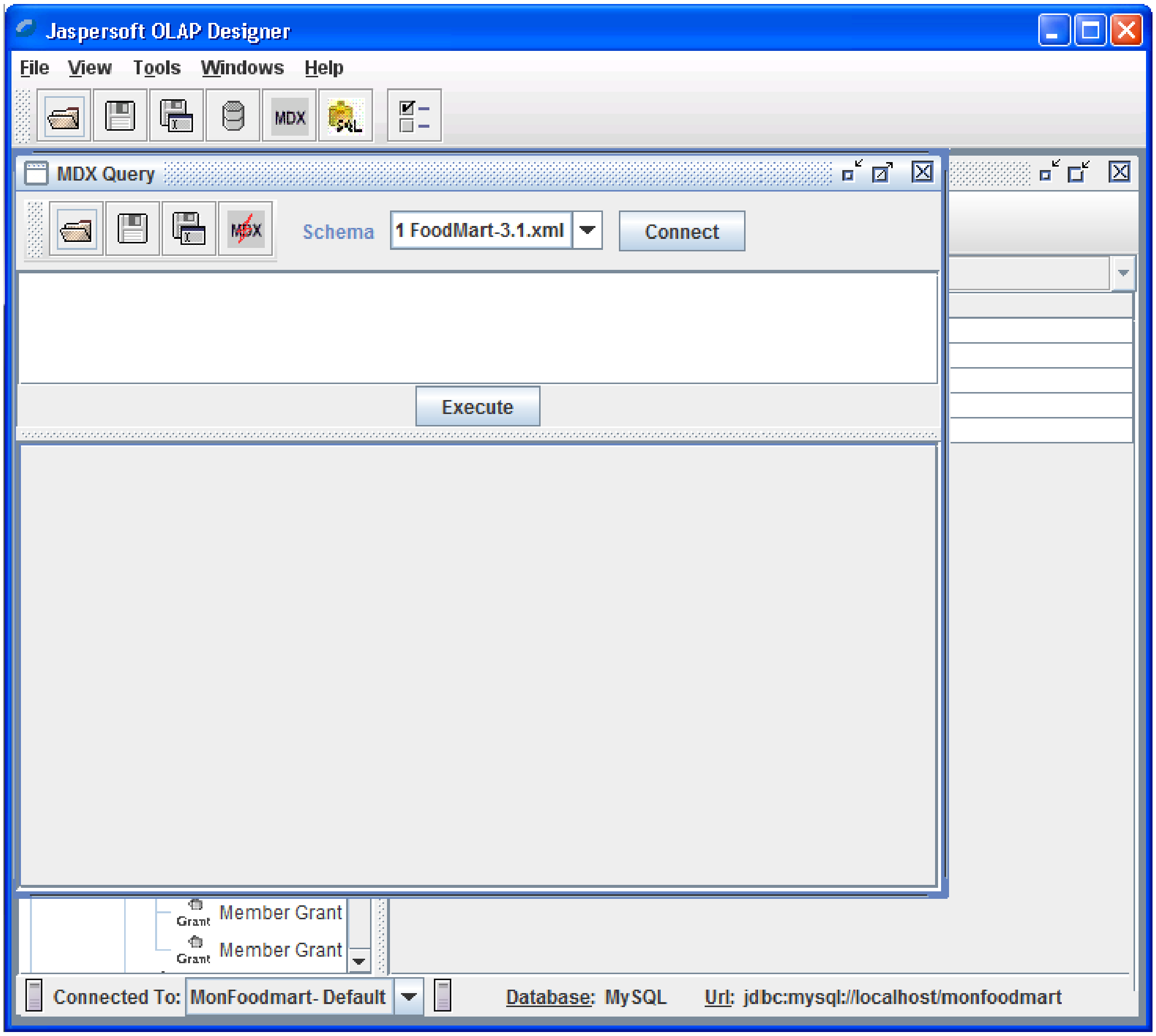
- **Open**, **Save**, and **Save As**.

  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.



1 FoodMart-3.1.xml

Connected To: MonFoodmart- Default

Database: MySQL

Url: jdbc:mysql://localhost/monfoodmart

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:

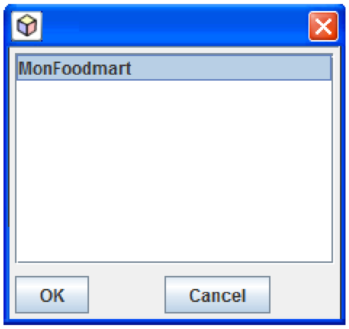
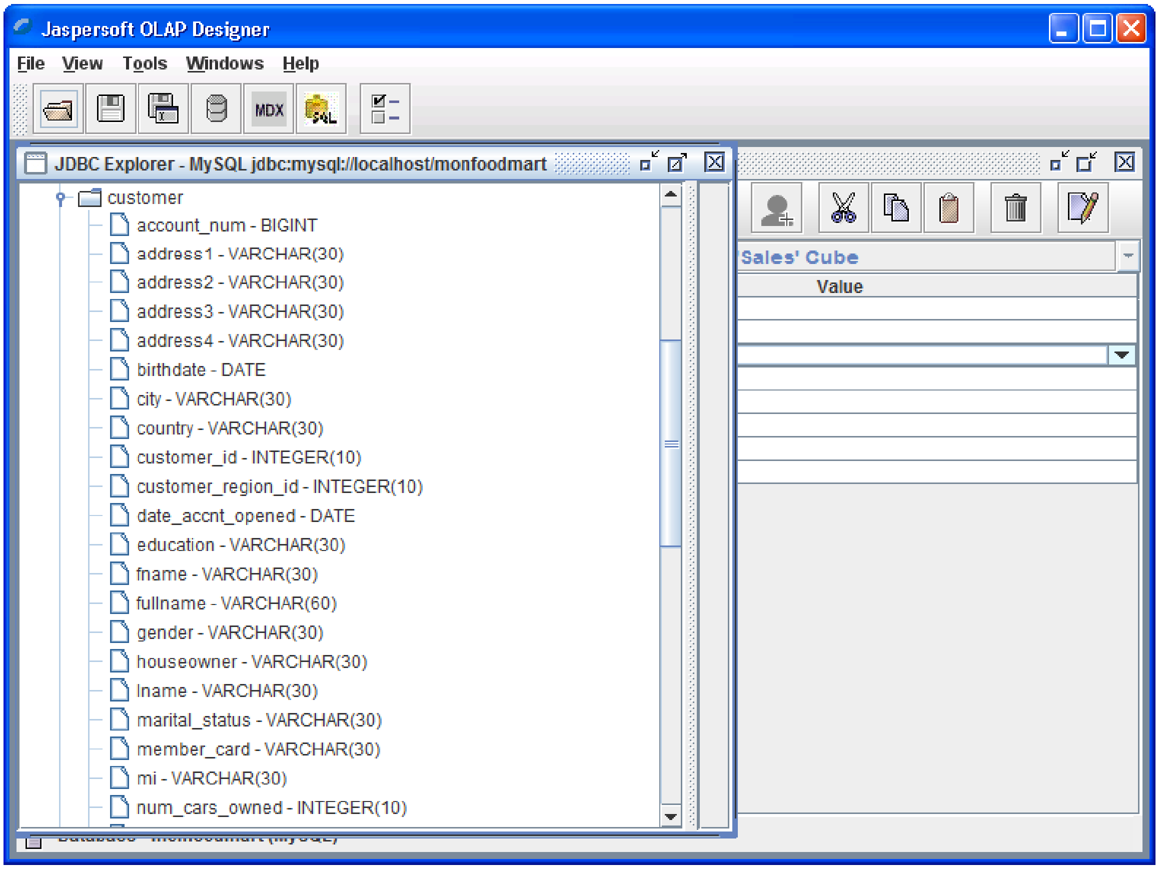

MonFoodmart

Figure 13 Selecting a Connection to Explore its Database

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


Jaspersoft OLAP Designer
File View Tools Windows Help
MDX
JDBC Explorer - MySQL jdbc:mysql://localhost/monfoodmart
Sales' Cube
Value

customer
- account_num - BIGINT
- address1 - VARCHAR(30)
- address2 - VARCHAR(30)
- address3 - VARCHAR(30)
- address4 - VARCHAR(30)
- birthdate - DATE
- city - VARCHAR(30)
- country - VARCHAR(30)
- customer_id - INTEGER(10)
- customer_region_id - INTEGER(10)
- date_accnt_opened - DATE
- education - VARCHAR(30)
- fname - VARCHAR(30)
- fullname - VARCHAR(60)
- gender - VARCHAR(30)
- houseowner - VARCHAR(30)
- lname - VARCHAR(30)
- marital_status - VARCHAR(30)
- member_card - VARCHAR(30)
- mi - VARCHAR(30)
- num_cars_owned - INTEGER(10)

Figure 14 Exploring a Database

3.5 Setting up Schemas

The process of creating an analysis view starts in the Designer and ends in Jaspersoft OLAP.

In the Designer:

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

In Jaspersoft OLAP:

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 Designer to populate the data source fields in JasperReports Server.
2. Create a Mondrian connection that points to the data source and includes the OLAP schema XML file.

   If you use Jaspersoft OLAP Professional, you can store the OLAP schema in the repository. Use the **New Resource** tool bar button and select **Other > XML File**, then upload the OLAP schema you created in the Designer.
3. If you use Jaspersoft OLAP Professional and need to secure your data, edit your Mondrian connection to include the AGXML file you created in the Designer.

4. Create an analysis view that points to your Mondrian connection Mondrian connection and includes one of the MDX queries you defined in the Designer.

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

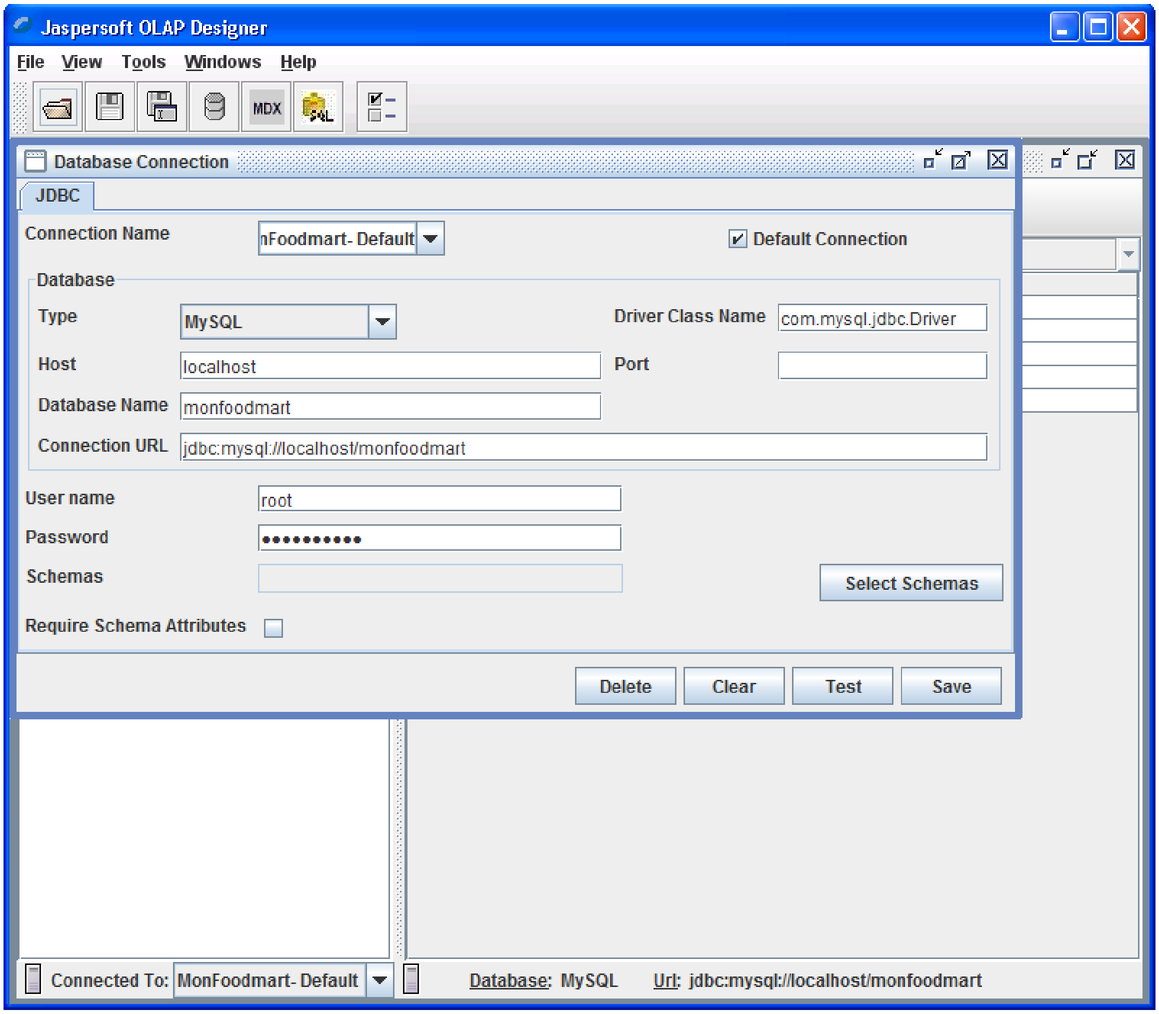
3.5.1 Getting Started

An OLAP schema defines multi-dimensional OLAP metadata in order to query a relational database that is structured as a star or snowflake schema. Such database structures have one or more fact tables with a large number of rows (whose values are known as “measures”) and dimension tables that categorize the facts.

This structure, or metadata, is used by the Mondrian engine to automatically generate SQL that satisfies the MDX queries. MDX is an OLAP-optimized query language for doing everything from simple totals across multiple categories to advanced statistical and time series analyses. 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 a Database Connection



Connection Name: nFoodmart- Default

☑ Default Connection

Database

Type: MySQL

Driver Class Name: com.mysql.jdbc.Driver

Host: localhost

Port:

Database Name: monfoodmart

Connection URL: jdbc:mysql://localhost/monfoodmart

User name: root

Password: ••••••••••

Schemas:

Require Schema Attributes □

Connected To: MonFoodmart- Default

Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

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.

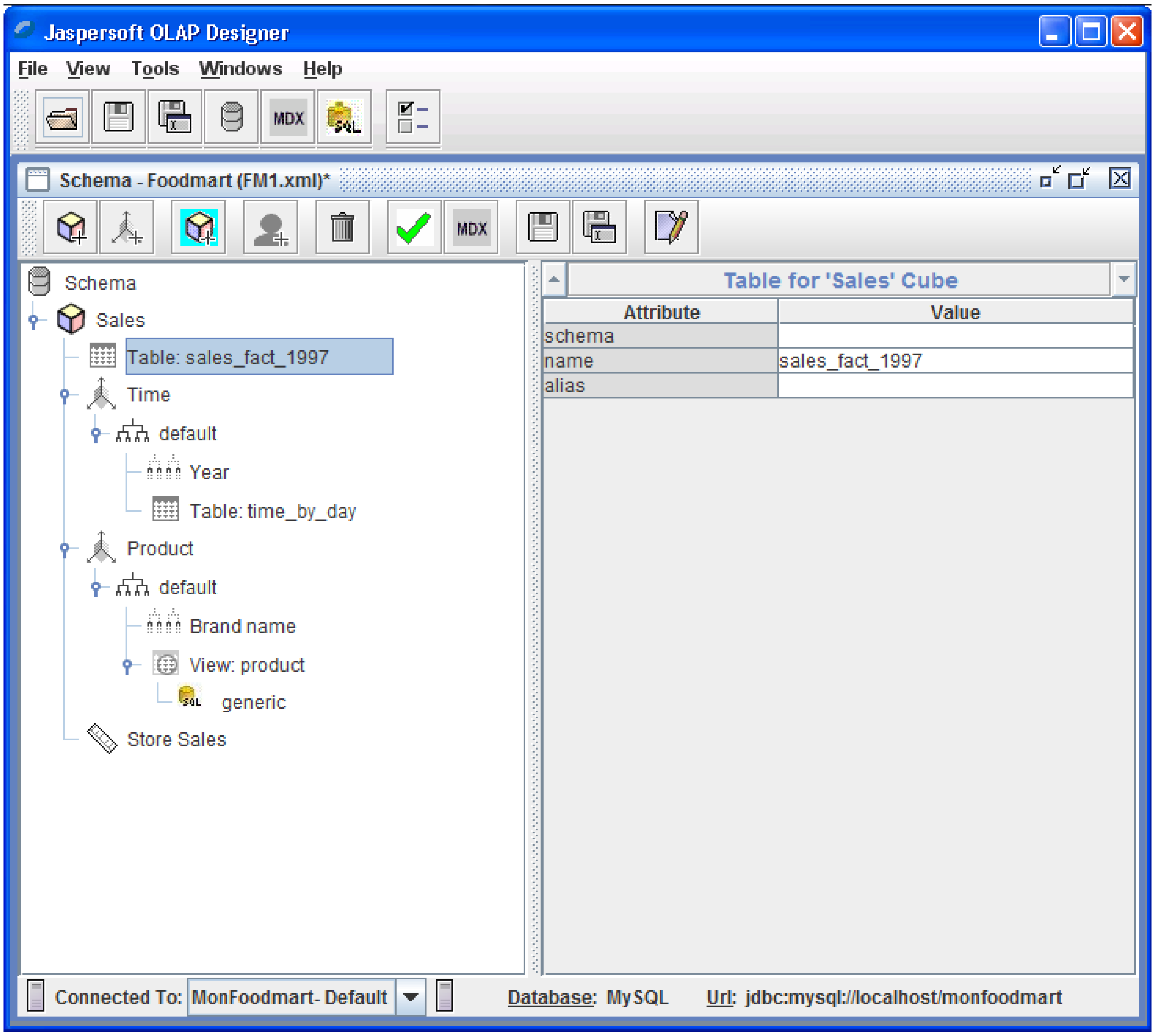


Table for 'Sales' Cube

Attribute	Value
schema	
name	sales_fact_1997
alias	

Connected To: MonFoodmart- Default

Database: MySQL

Url: jdbc:mysql://localhost/monfoodmart

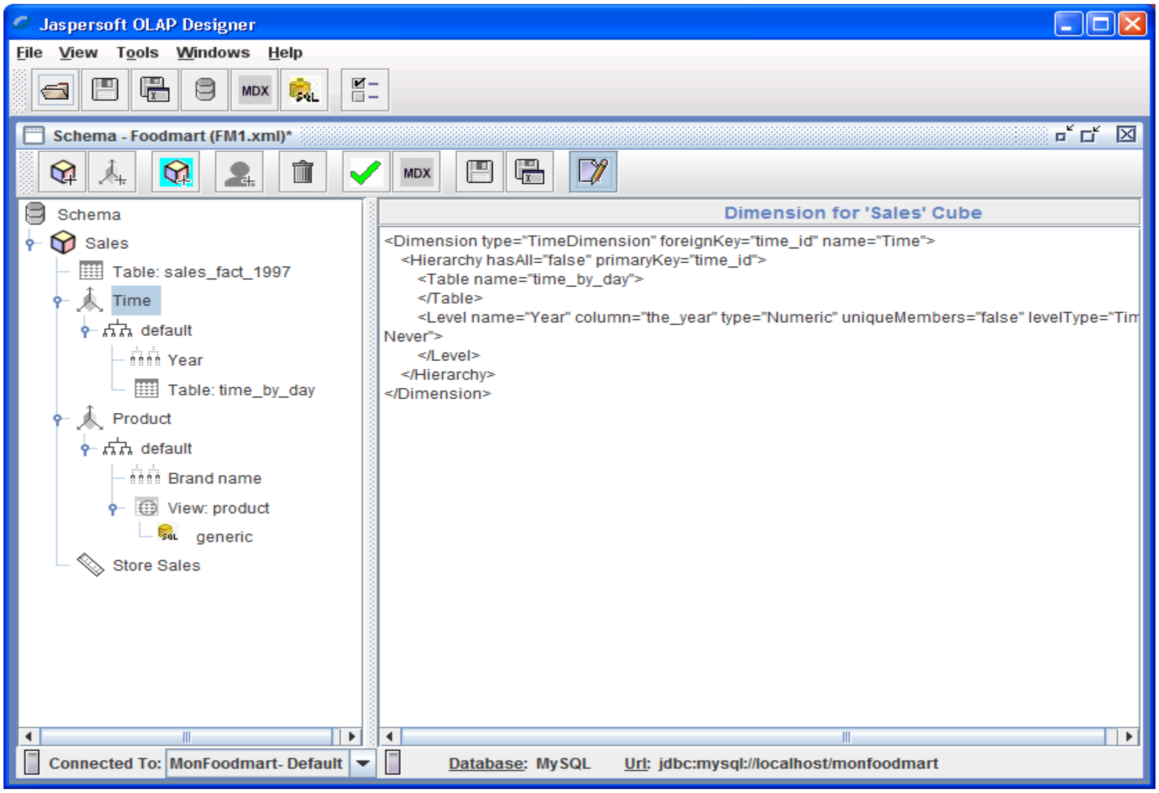
Figure 16 A Simple Dimension

Figure 16 shows a Sales cube. In Mondrian, each cube revolves around one fact table; in this case, it is 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 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. Unique MDX functions can provide quarter- or year-to-date summaries, as well as period-over-period calculations.

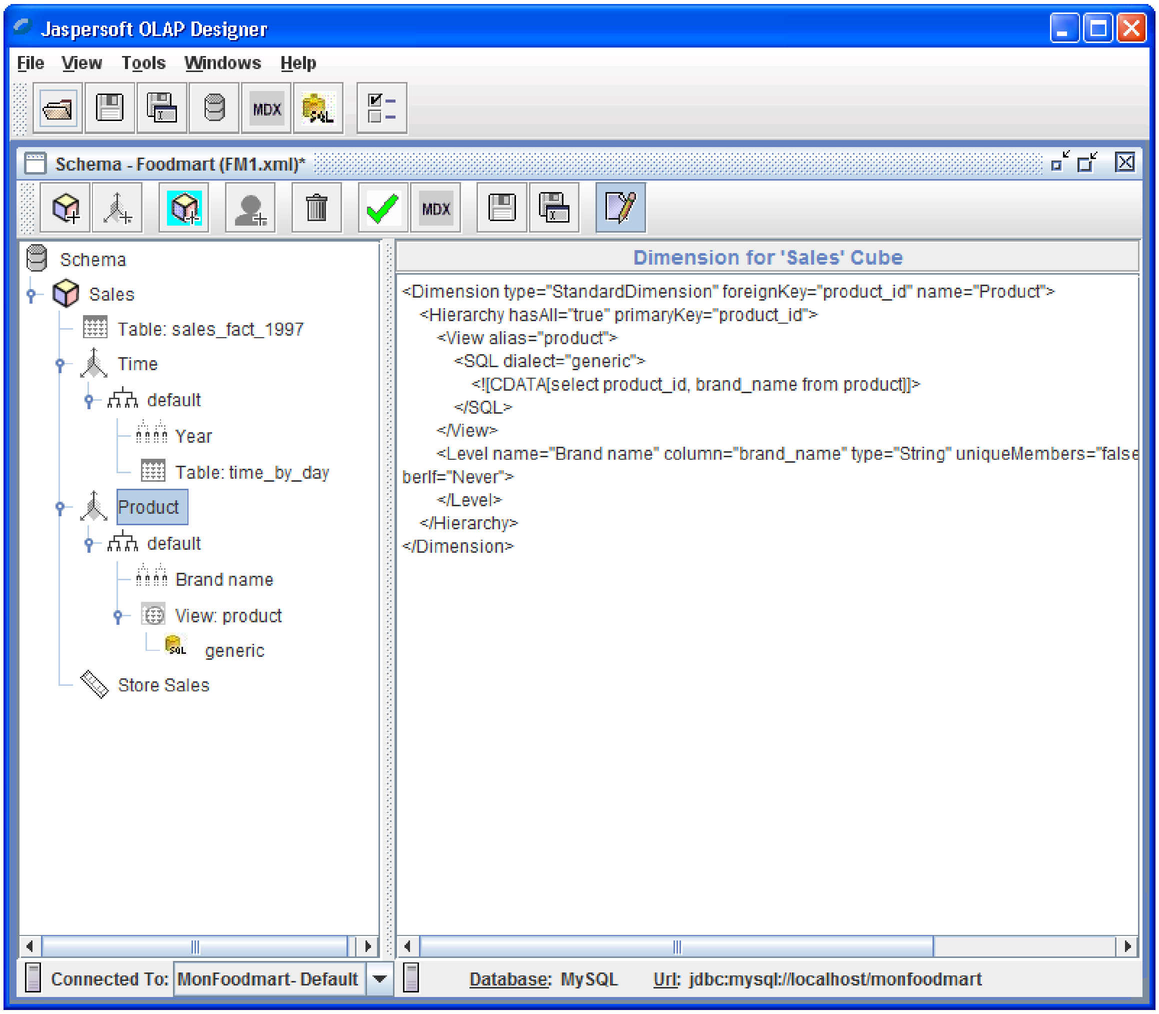


```
<Dimension type="TimeDimension" foreignKey="time_id" name="Time">
   <Hierarchy hasAll="false" primaryKey="time_id">
      <Table name="time_by_day">
      </Table>
      <Level name="Year" column="the_year" type="Numeric" uniqueMembers="false" levelType="Tim
Never">
      </Level>
   </Hierarchy>
</Dimension>
```

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, which is 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 (Brand name) to categorize facts; it is 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.



```
<Dimension type="StandardDimension" foreignKey="product_id" name="Product">
  <Hierarchy hasAll="true" primaryKey="product_id">
    <View alias="product">
      <SQL dialect="generic">
        <![CDATA[select product_id, brand_name from product]]>
      </SQL>
    </View>
    <Level name="Brand name" column="brand_name" type="String" uniqueMembers="false
berIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

Connected To: MonFoodmart- Default

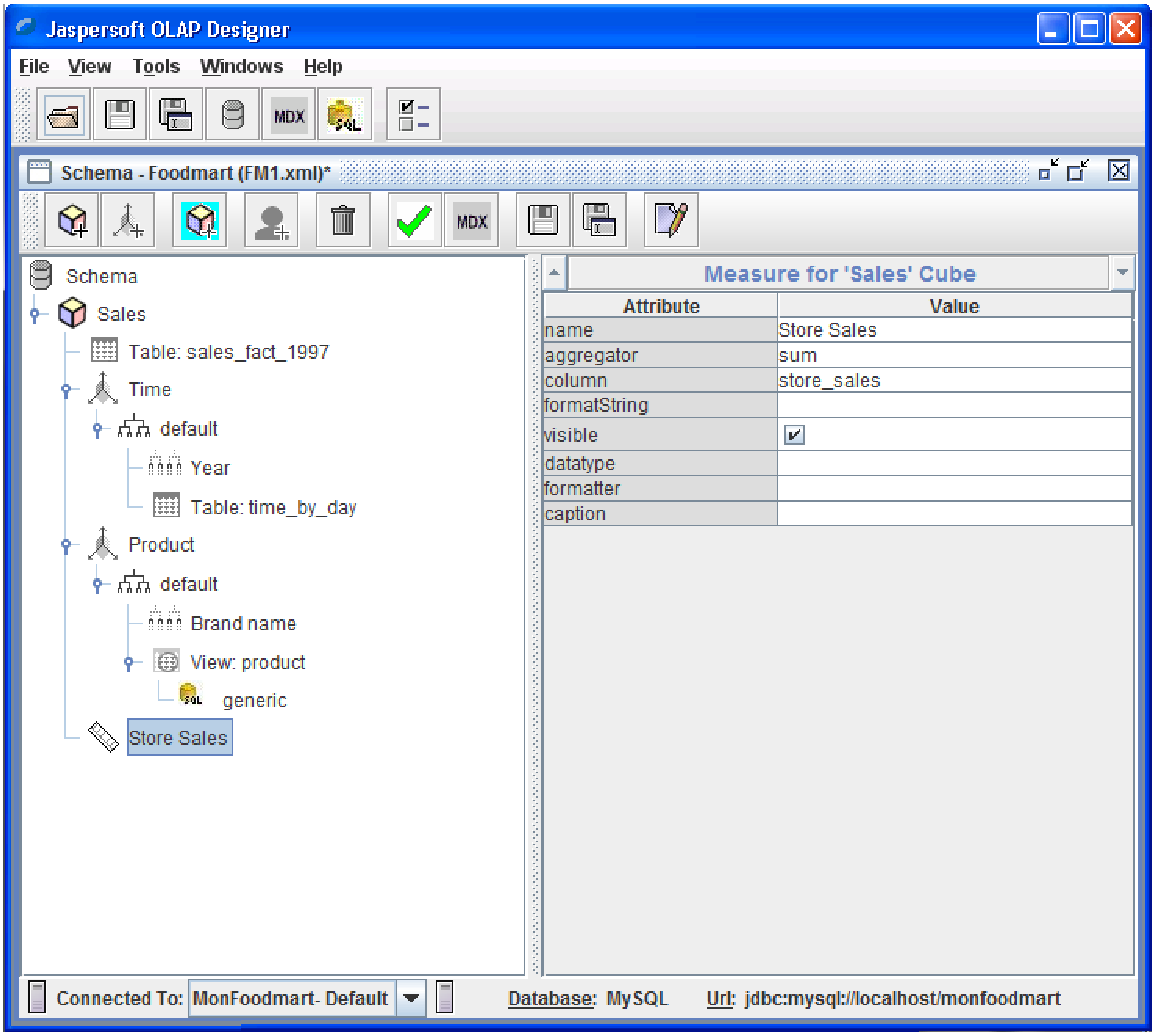
Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

Figure 18 Product Dimension

The Product dimension could have been modeled as a join, which expects that the tables that make up the snowflake are joined on a single column.

Measures

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



Attribute	Value
name	Store Sales
aggregator	sum
column	store_sales
formatString	
visible	☑
datatype	
formatter	
caption	

Connected To: MonFoodmart- Default Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

Figure 19 A Measure for Products

Testing the Schema

1. To test the schema, click ✓ **(Test Schema)**.
   If the test succeeds, a message appears indicating that the schema passed validation.

2. Click MDX (**New MDX Query**) to open the MDX editor.

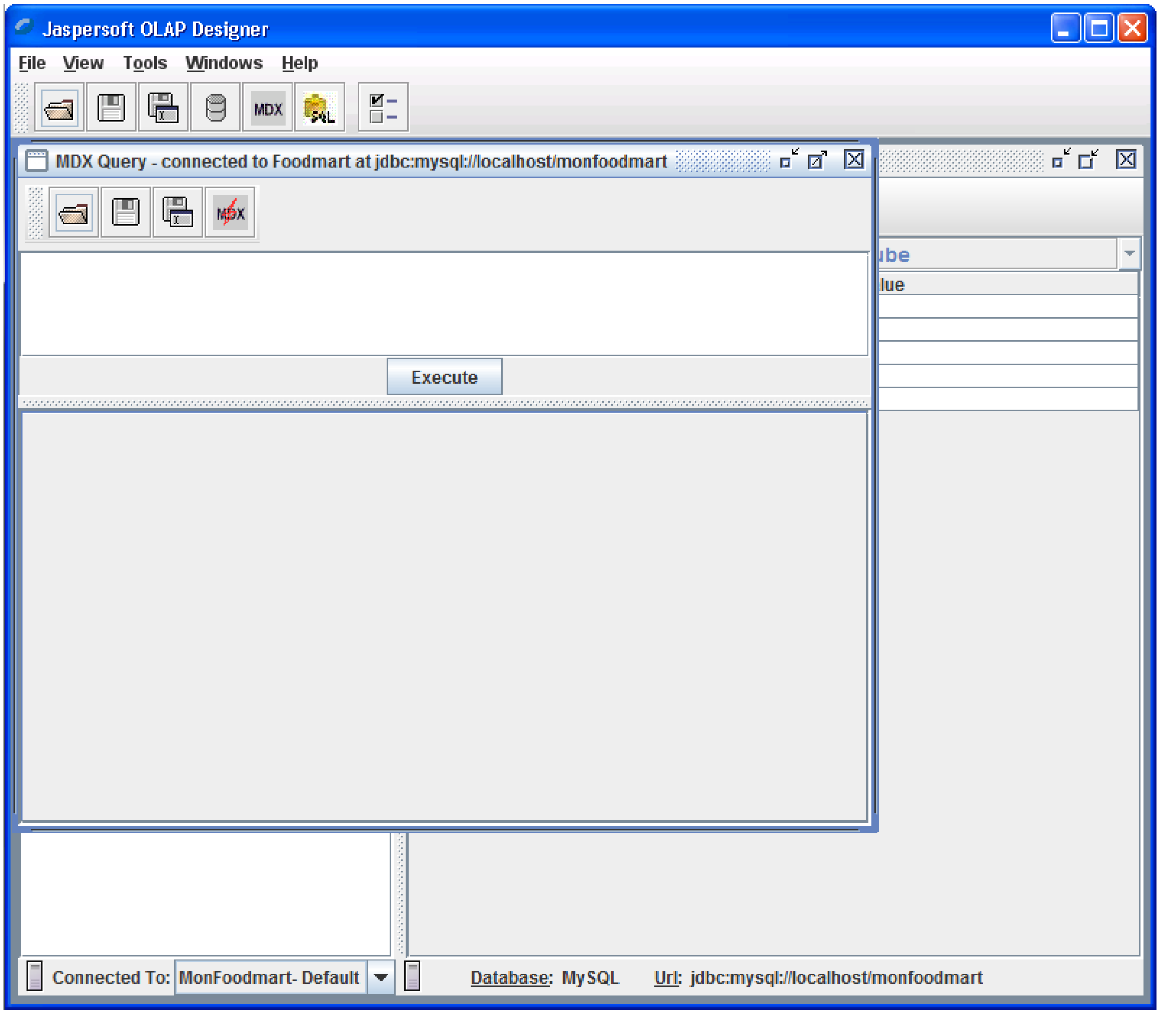
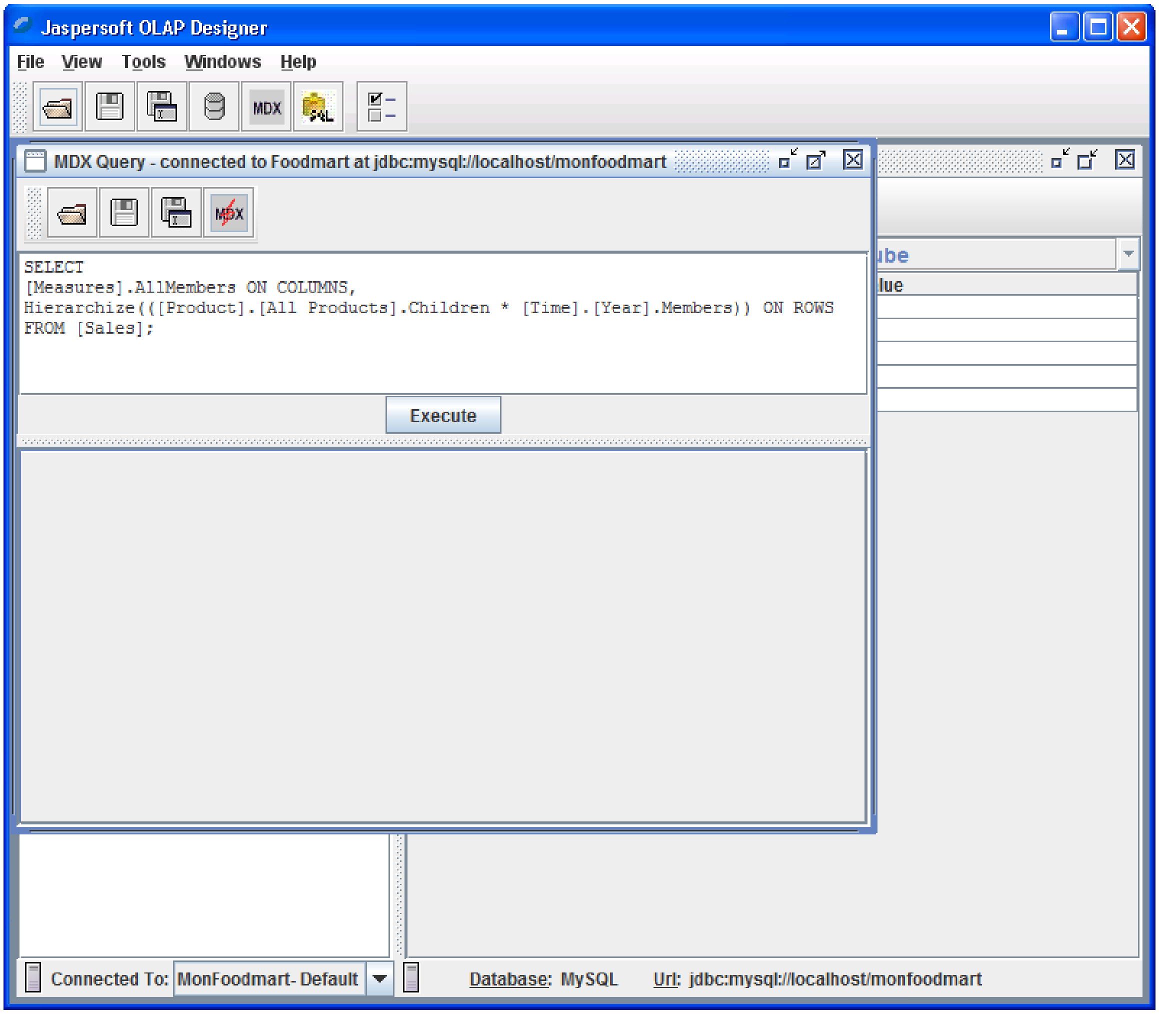

Jaspersoft OLAP Designer
File View Tools Windows Help
MDX
MDX Query - connected to Foodmart at jdbc:mysql://localhost/monfoodmart
Execute
ube
lue
Connected To: MonFoodmart- Default
Database: MySQL
Url: jdbc:mysql://localhost/monfoodmart

Figure 20 MDX Editor

3. Click **(Generate MDX Queries)** to automatically create MDX queries.



MDX Query - connected to Foodmart at jdbc:mysql://localhost/monfoodmart

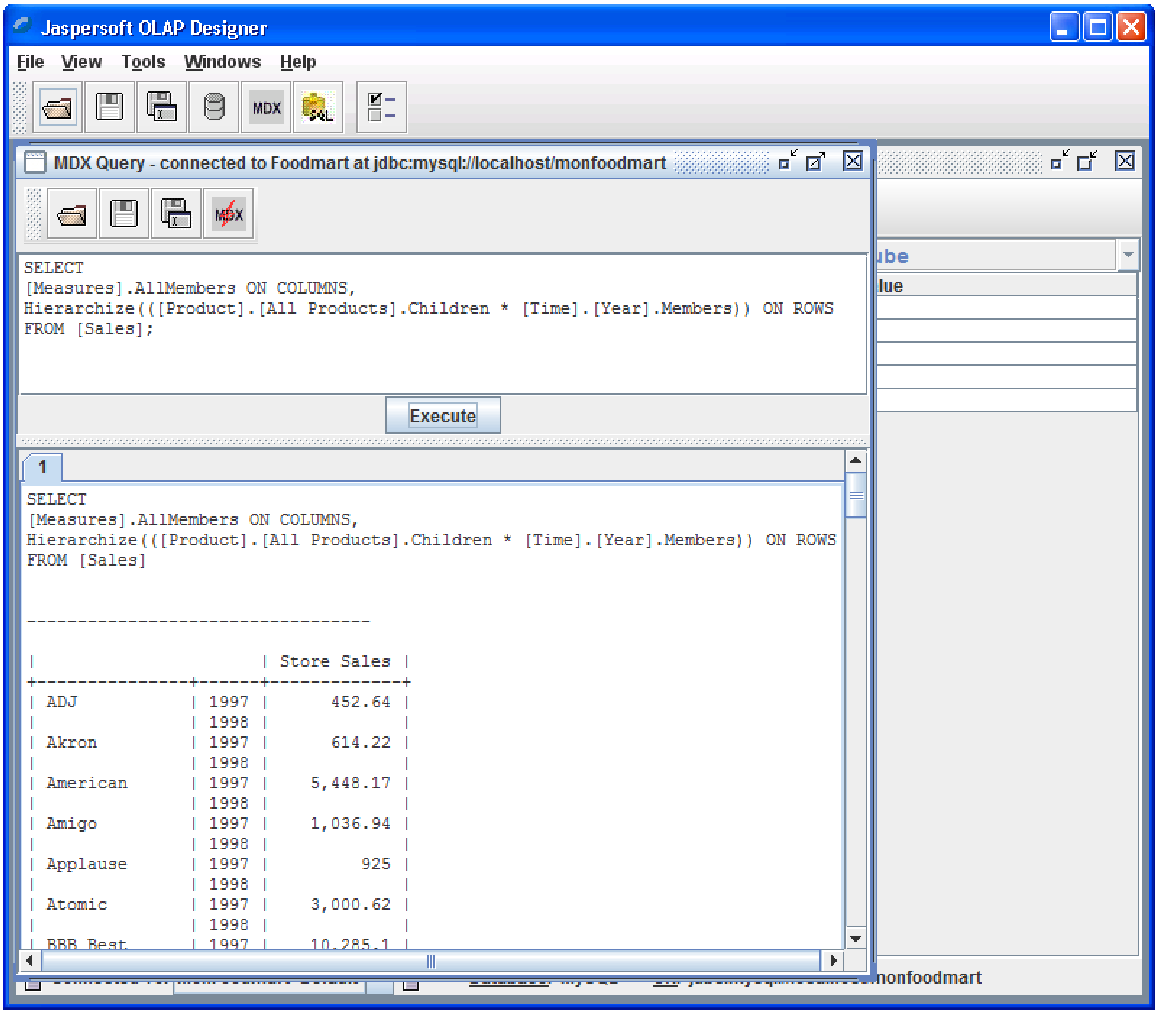
```
SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Product].[All Products].Children * [Time].[Year].Members)) ON ROWS
FROM [Sales];
```

Connected To: MonFoodmart- Default

Database: MySQL Url: jdbc:mysql://localhost/monfoodmart

Figure 21 A Generated MDX Query

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



Jaspersoft OLAP Designer
File View Tools Windows Help
MDX
MDX Query - connected to Foodmart at jdbc:mysql://localhost/monfoodmart
Execute
1
ıbe
lue
nonfoodmart

```
SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Product].[All Products].Children * [Time].[Year].Members)) ON ROWS
FROM [Sales];
```

```
SELECT
[Measures].AllMembers ON COLUMNS,
Hierarchize(([Product].[All Products].Children * [Time].[Year].Members)) ON ROWS
FROM [Sales]


----------------------------------

|                      | Store Sales |
+---------------+------+-------------+
| ADJ           | 1997 |      452.64 |
|               | 1998 |             |
| Akron         | 1997 |      614.22 |
|               | 1998 |             |
| American      | 1997 |    5,448.17 |
|               | 1998 |             |
| Amigo         | 1997 |    1,036.94 |
|               | 1998 |             |
| Applause      | 1997 |         925 |
|               | 1998 |             |
| Atomic        | 1997 |    3,000.62 |
|               | 1998 |             |
| BBB Best      | 1997 |    10,285.1 |
```

Figure 22 An MDX Query and Its Results

5. Close the MDX editor and click **Save** in the Schema Explorer.

Now you can upload the schema and create an analysis view that uses it in combination with one of the generated MDX queries and, in Jaspersoft OLAP Professional, an AGXML security file.

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