jdbc4olap user guide

Table of Contents

Overview

Objective

jdbc4olap's goal is, as its name says, to deliver a ready-to-use jdbc driver for OLAP databases. Jdbc is an interface that allows an application to connect to various relational environments, but not multidimensional ones. Indeed, that project implies to make a bridge between relational and multidimensional worlds, which are incompatible as they rely on two different models and involve different technologies.

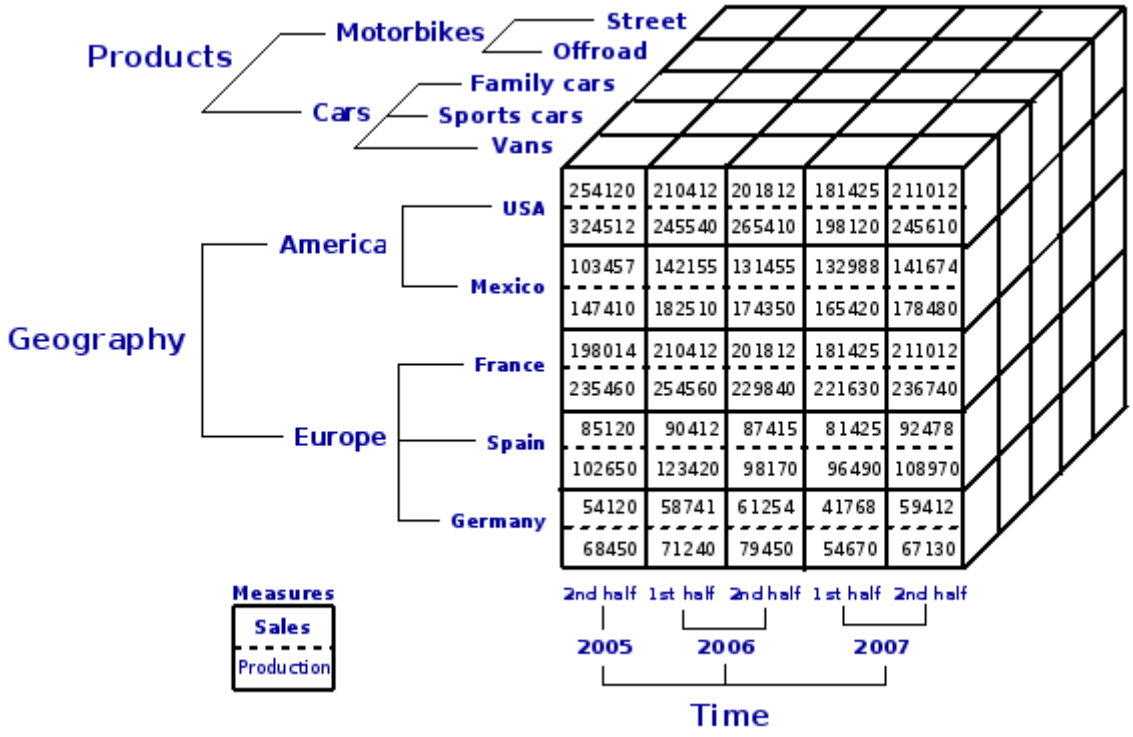
Relational vs Multidimensional

Relational databases, such as Oracle RDBMS, MySQL, MS SQL Server, enclose sets of tables, divided in columns and containing rows of data. Rows from different tables can be linked in order to represent an association between these sets of data. SQL is the standard language to dialog with a relational database server, in order to retrieve or manipulate data.

Multidimensional databases, such as SAP BW, MS SQL Server Analysis Services, Mondrian, consist of cubes of several dimensions (not only 3 like in a real cube), composed of one or more hierarchies of levels whose values, called members, are data properties. The data is located at the intersection of these dimensions, in cube cells that contain different values called measures. MDX emerged as the standard language for multidimensional databases, and despite some similarities with SQL, has its own syntax and mechanisms.

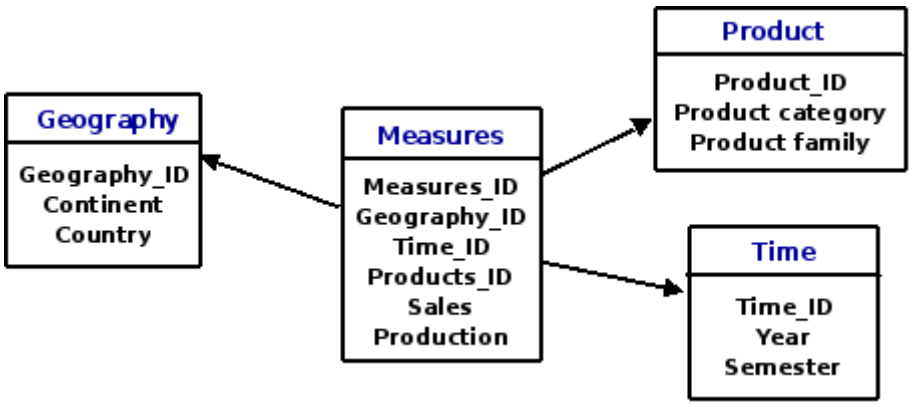
Jdbc4olap's model

In order to consider an OLAP base from a relational point of view, a correspondence must be established between the two models. Let's consider the example cube in the following illustration.



Are shown on this diagram 3 different dimensions: Products, Geography, and Time and their respective levels: {Product category, Product family}, {Continent, Country}, {Year, semester}. Two measures are available: Sales and Production.

In jdbc4olap, that cube would be represented as the following relational database:



That schema keeps the links between the dimensions and the measures. An important precision must be made: this example displays a cube whose dimensions include only one hierarchy. For example if the Geography dimension had different hierarchies corresponding to different ways of expressing a location, that would generate as many tables.

XML for Analysis

In order to retrieve data, we need a way to dialog with the server. Among the various possible ways to access OLAP databases, we chose XMLA. These initials indicate a standardization of the interface of access to multidimensional bases, defined by a consortium of major actors of the market. Here's the list of the principal decisional systems that use this standard:

- Hyperion Essbase 7
- Microsoft Analysis Services 2005
- Mondrian
- SAP BW from 3.0a

Based on XML, SOAP and HTTP standards, this technology consists of a webservice available on the OLAP server which dialogues with XMLA clients by exchanging SOAP messages. There are several ways to secure a web service and thus a XMLA access: SSL encoding on the transport layer, XML encoding on the message layer, and also access control with a proxy. And, the driver being exclusively in read-only mode, there are no possible data-damage issues.

SQL-MDX conversion

Cubes being represented as a relational model, and the driver aiming for the better JDBC compliancy, SQL was naturally the language to use. But as most OLAP servers only supports MDX, a convertor had to be designed and integrated to the query process. That's why there are limitations on the supported queries, as all SQL query concepts can't be translated in MDX.

Features

Connection

The first step in JDBC compliance is to conform to its connection mechanism. Indeed, once the driver installed, any jdbc connect attempt with a location starting with jdbc:jdbc4olap: will select jdbc4olap as the driver to use for the connection and further operations.

Metadata

The JDBC's DatabaseMetadata interface is fully implemented. First, that means that the driver's possibilities and limitations are indicated. But it also means that the user can interrogate a server to discover its data. Indeed he can get navigate through available catalogs, schemas, tables and columns. These features are usually used to help the user setup a query.

Queries

As explained above, there are limitations in the supported SQL queries, due to the conversion to MDX and the impossibility to adapt some concepts. The query must look like this :

```
query := 'SELECT' fieldList 'FROM' tableList 'WHERE' filterList
'GROUP BY' expressionList ';'

fieldList := (field (',' field)*) | '*'

field := tableStar | columnName

tableStar := (catalog.schema.table|table|tableAlias)'.*'

columnName := [catalog.schema.table.|table.|tableAlias.]column
[['AS'] fieldAlias]

tableList := table (',' table)*

table := catalog.schema.table [tableAlias]

filterList := sqlExpression ('AND' sqlExpression)*

sqlExpression := operand (relationalExpression | inClause)

operand := NUMBER | STRING | filterColumn | '?'

relationalExpression := '=' operand

inClause := 'IN' '(' expressionList ')'

expressionList := operand (',' operand)*

filterColumn := [catalog.schema.table.|table.|tableAlias.](column|
fieldAlias)
```

As we can see on this grammar, most of the limitations concern the filters, who can only be combined by AND operators, and consist of '=' or 'IN' clauses.

An important precision, in the actual release, the filters like 'column = column' are ignored. Indeed it's implied that the joins are set correctly to establish the model described before. So it's impossible to process a cartesian product.

If you look carefully at the grammar, you'll notice that '?' is supported in the filters. Indeed, not only statements but also prepared statements are supported in this release.

Setup

You need to add the different jar files from the zip archive to your classpath. The class used to setup your jdbc access is org.jdbc4olap.jdbc.OlapDriver

The url must start with 'jdbc:jdbc4olap:'. Here are different examples with servers tested with this release:

- MS SQL Server: jdbc:jdbc4olap:http://server:port/OLAP/msmdpump.dll
- Mondrian: jdbc:jdbc4olap:http://server:port/mondrian/xmla
- SAP BW: jdbc:jdbc4olap:http://server:port/sap/bw/soap/xmla?sap-client=number

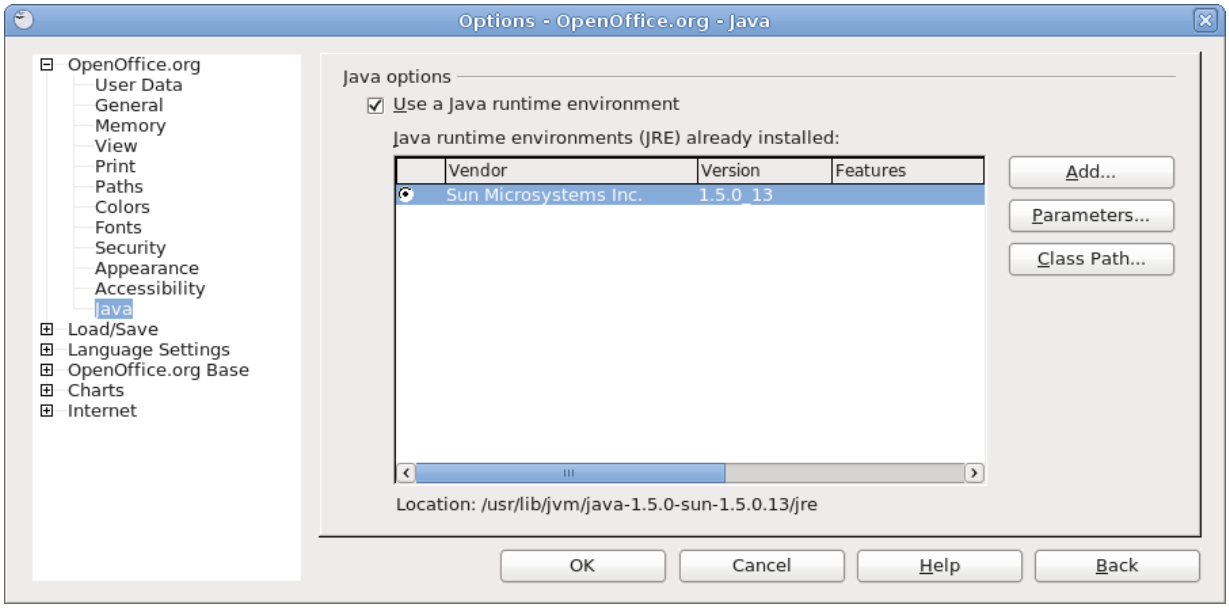
Open Office Base Mondrian scenario

Here are the different detailed steps in using the driver from Open Office Base to access a Mondrian OLAP database.

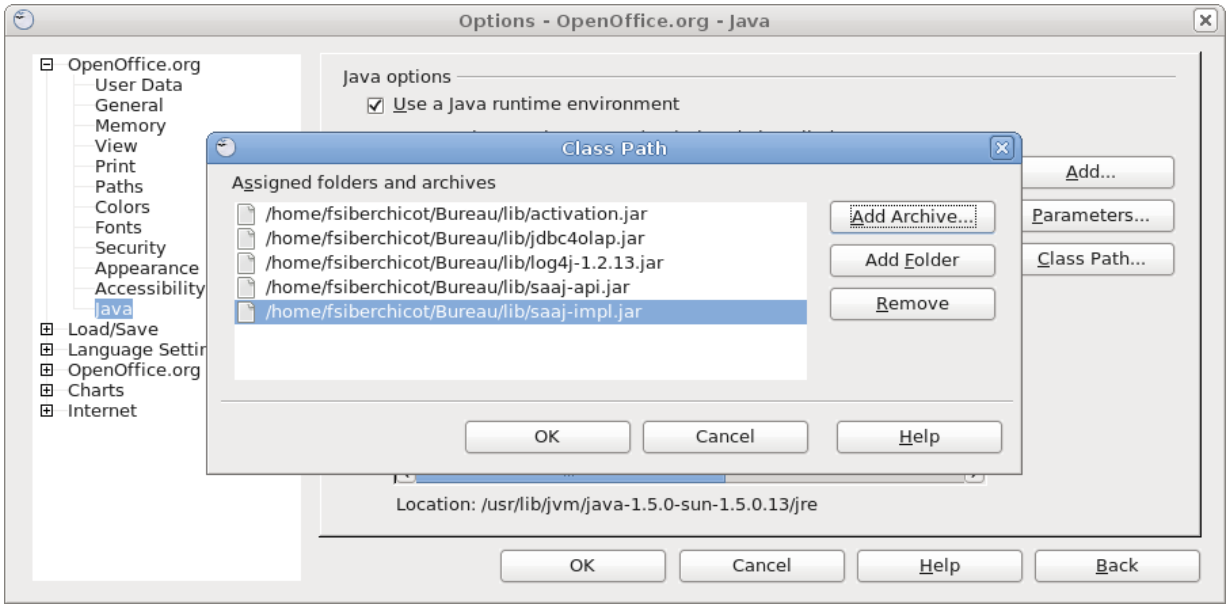
Start Open Office Base, and since a database must opened, create one if necessary:

In the next screen, you don't need to register but choose top open the database for editing. Finally, choose a location and filename for that base.

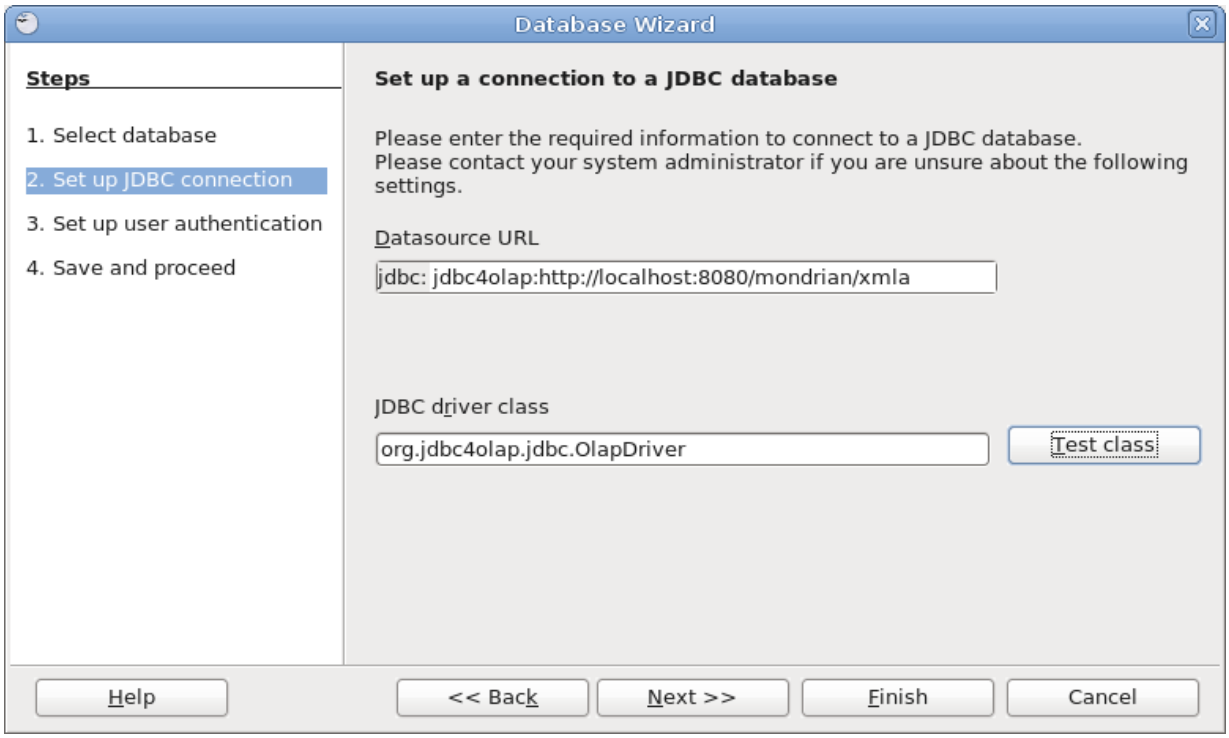
Go to Tools/Options and open Open Office.org/Java settings:



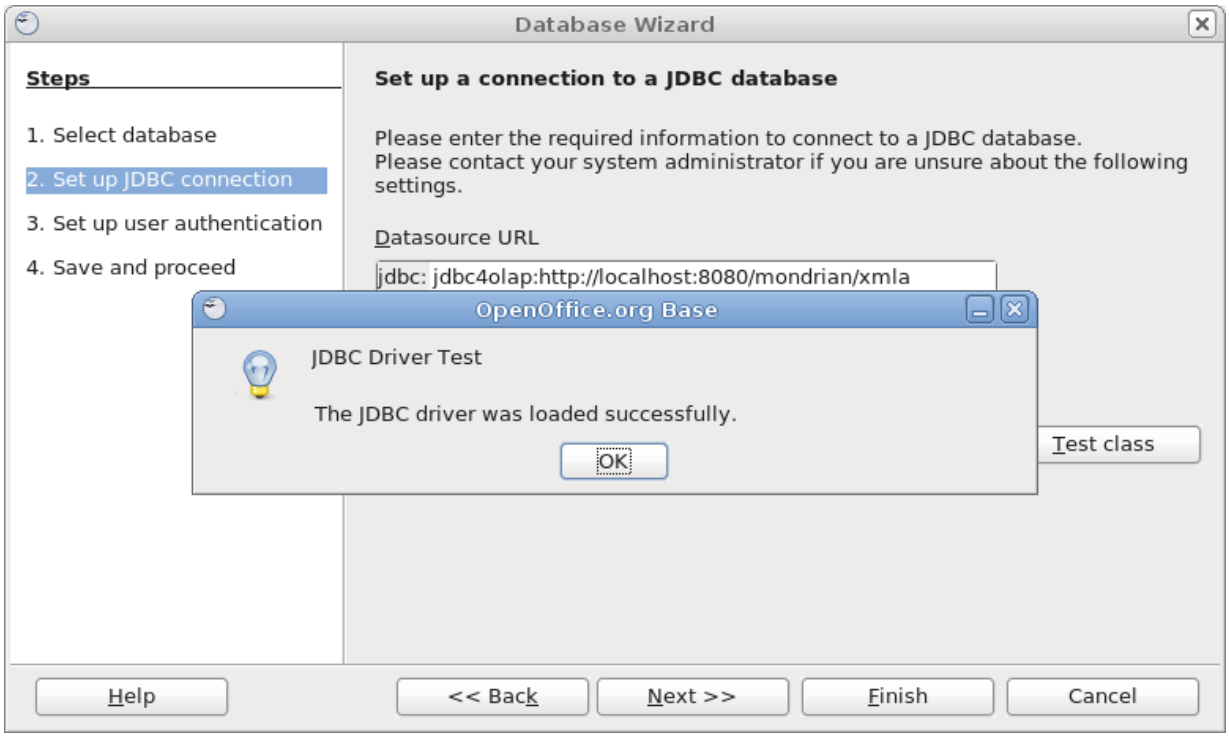
Select a JRE 1.5 or add one with the «Add... » button if none is available. Click on « Class Path... » and add all the jar files from the jdbc4olap archive:



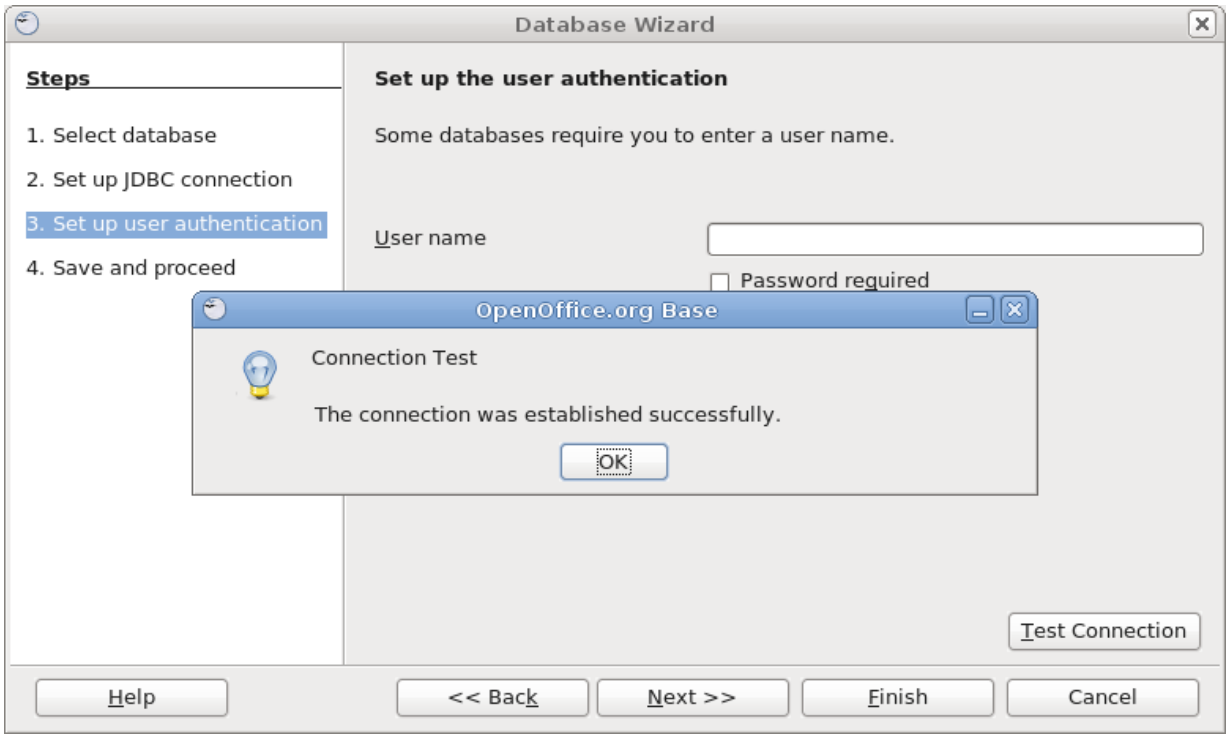
Everything is now correctly set up, and we don't need anymore the base we just created. We can restart Open Office to create a JDBC connection:



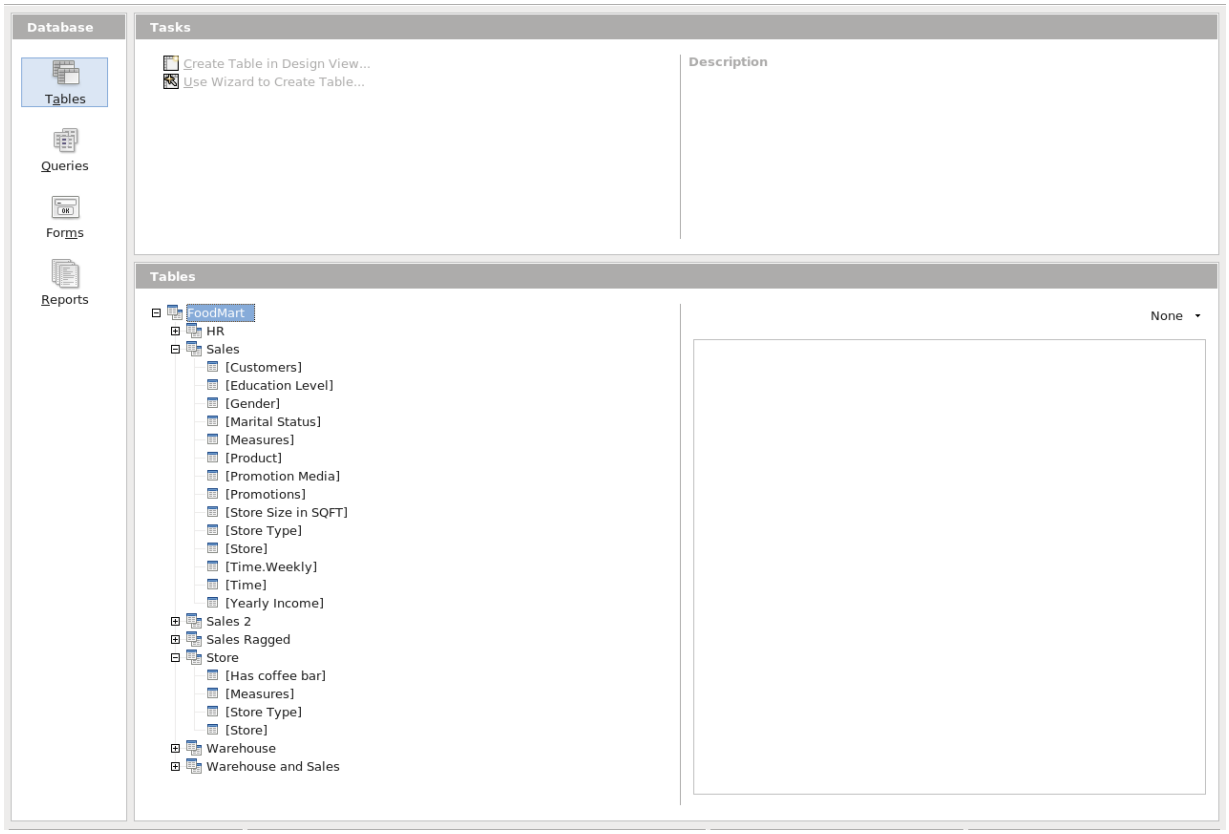
The «Test Class» button should give this:



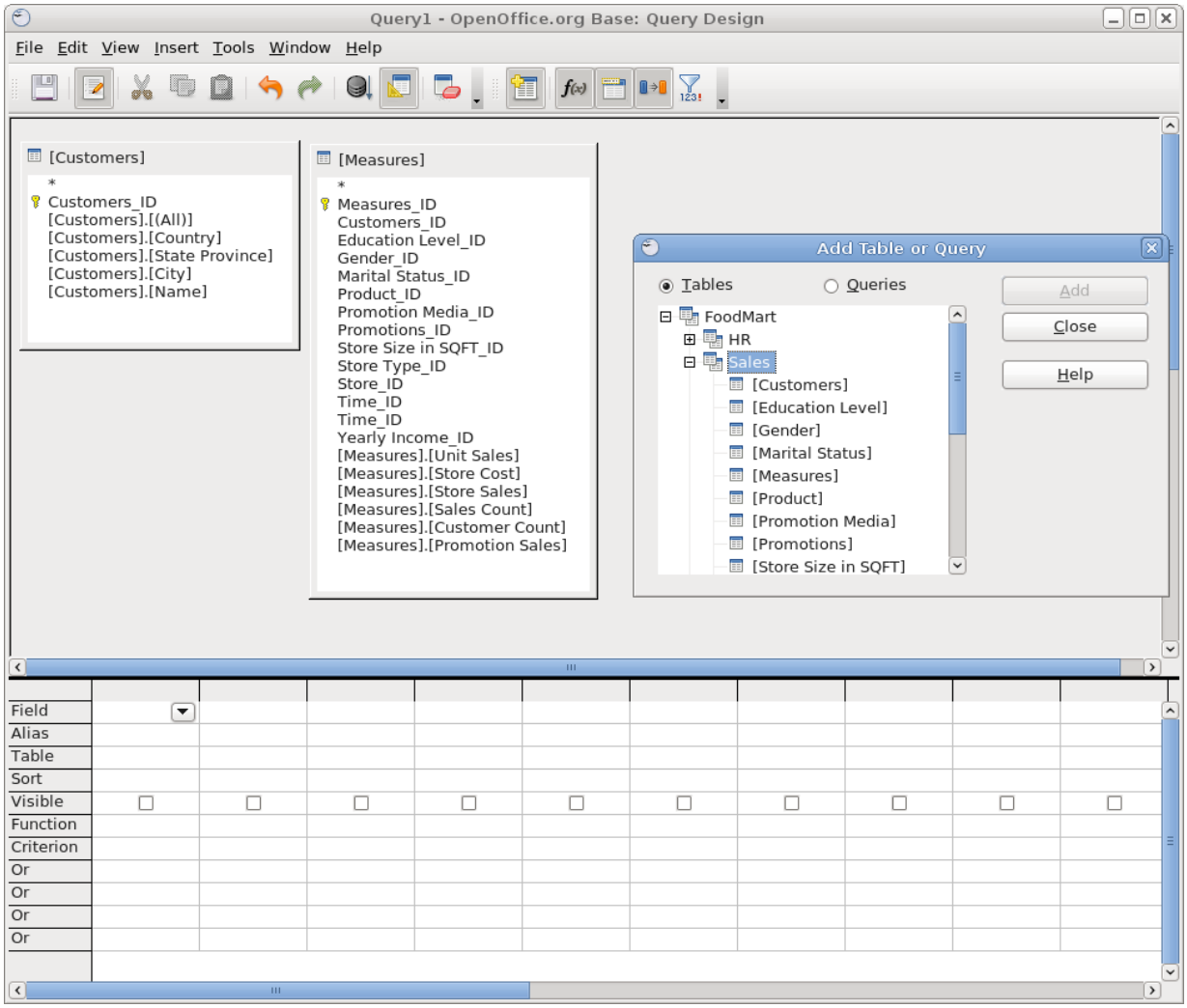
And the «Connection Test» from the next screen should end the same way :



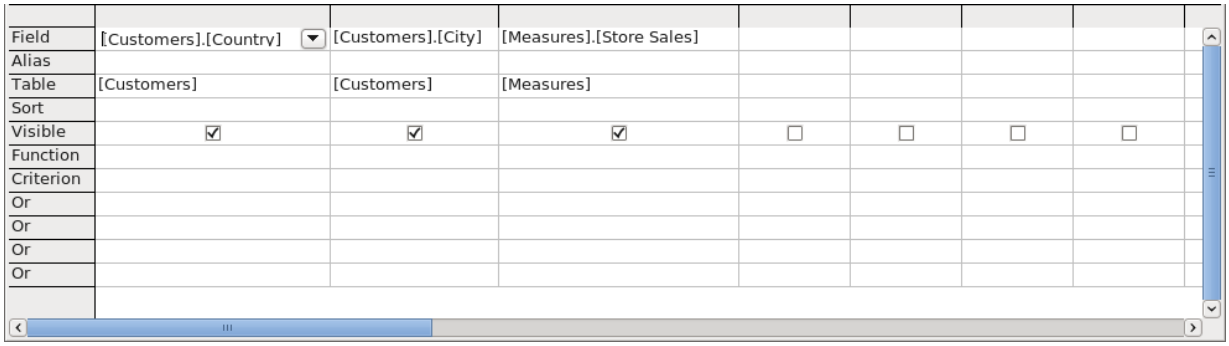
Once again, choose to register or not but opt for the editing mode, and choose a filename. Then go to Tables, and after a few seconds you'll see the database with the jdbc4olap relational model:



Now you can really use the database connection. All the functionnalities haven't been tested yet, but you can for example create a new query :



Only add tables from the same schema, because otherwise it would generate a query from different cubes, and that can't be done. Then select the columns you want on your query. A bug in Open Office can occur here: if you double click in a box (that appeared when you added a table) to select a column it works fine, but if you use the combo in the bottom part of the screen to make your selection it changes the name of the column and then the query fails.



You can finally launch the query to check its validity and its results :

	[Customers].[Country]	[Customers].[City]	[Measures].[Store Sales]
	Mexico	Guadalajara	0
	Mexico	Mexico City	0
	Mexico	Tlaxiaco	0
	Mexico	La Cruz	0
	Mexico	Orizaba	0
	Mexico	Merida	0
	Mexico	Camacho	0
	Mexico	Hidalgo	0
	USA	Altadena	5585.59
	USA	Arcadia	5136.59
	USA	Bellflower	6633.97
	USA	Berkeley	320.17
	USA	Beverly Hills	6194.37
	USA	Burbank	6577.33
	USA	Burlingame	407.38
	USA	Chula Vista	6284.3

Record 1 of 46 *