

Pentaho BI Suite

Basic notions of OLAP cubes and their
implementation with Schema Workbench

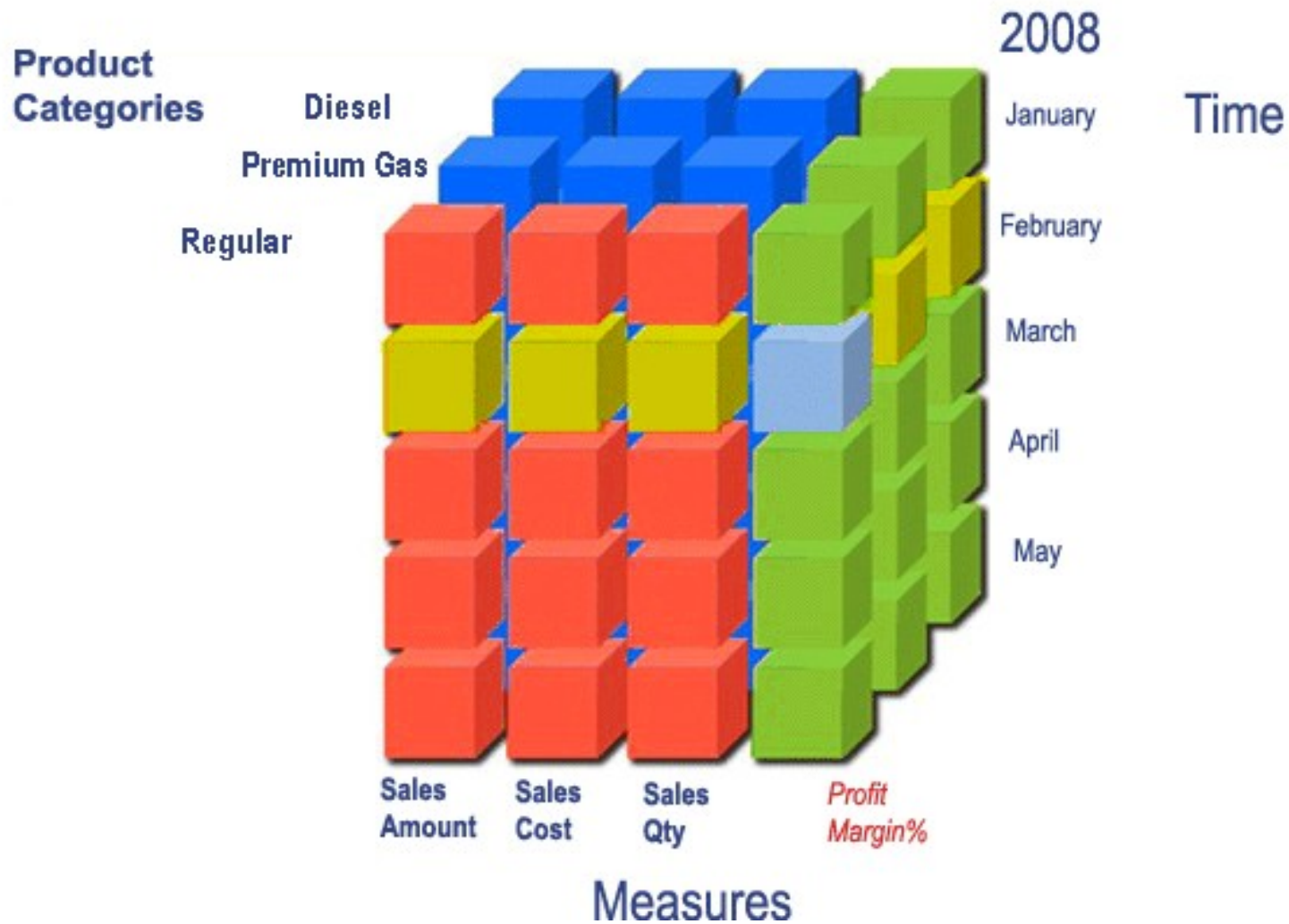
edited by Vladan Mijatovic
vladan.mijatovic@univr.it

Cube structure

Fact table consists of the measurements, metrics or facts of a business process. It is often located at the centre of a star schema or a snowflake schema, surrounded by dimension tables.

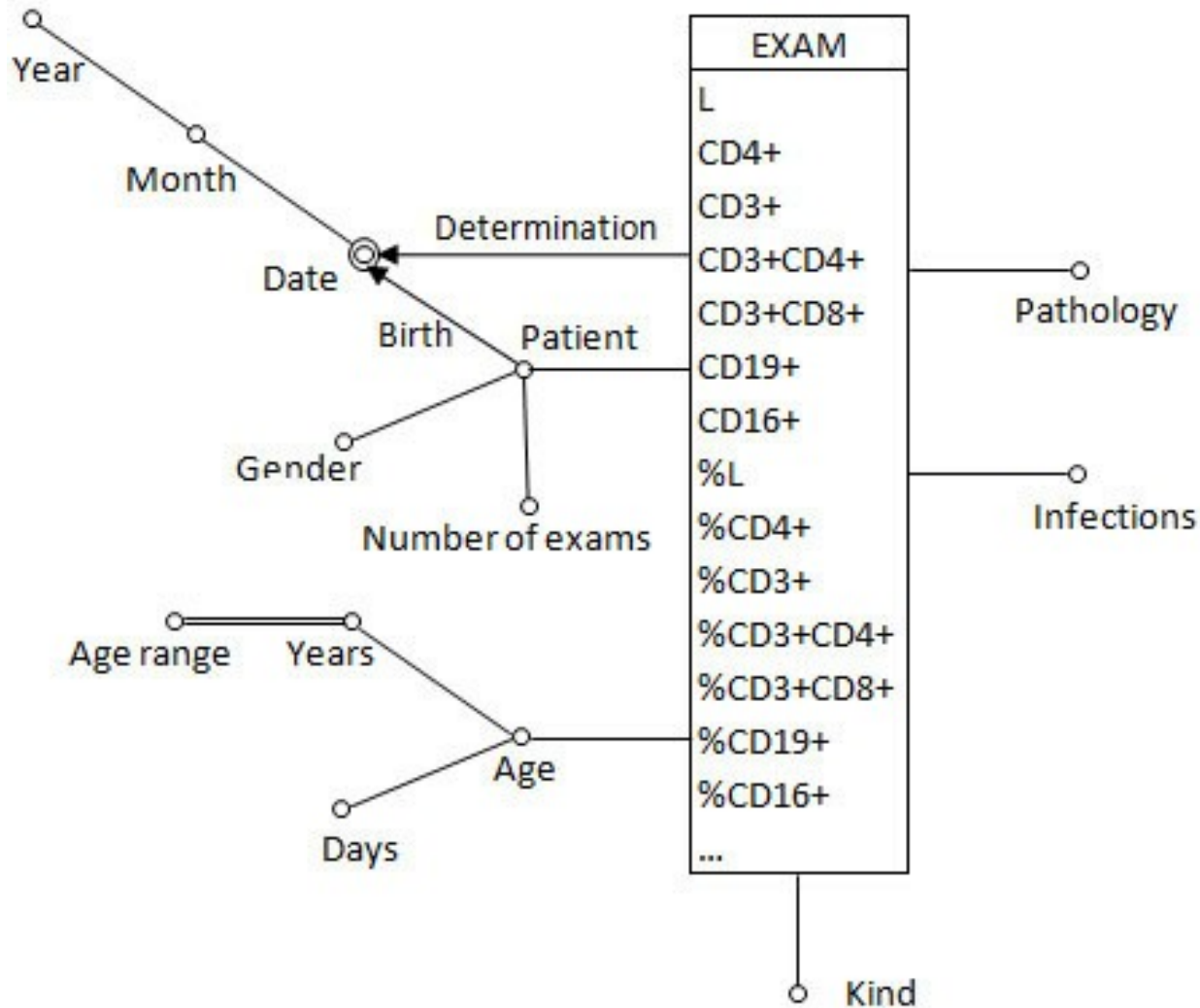
Dimension: is a data element that categorizes each item in a data set into non-overlapping regions. Each dimension in a data warehouse may have one or more hierarchies applied to it.

Multidimensional Model



Conceptual Model

Dimensional Fact Model (DFM) is an ad-hoc formalism specifically devised to support the conceptual modeling phase in a DW project.



How to Design a Mondrian Schema

- A schema defines a multi-dimensional database. It contains a logical model, consisting of cubes, hierarchies, and members, and a mapping of this model onto a physical model.
- The logical model consists of the constructs used to write queries in MDX language: cubes, dimensions, hierarchies, levels, and members
- The physical model is the source of the data which is presented through the logical model. It is typically a star schema, which is a set of tables in a relational database; later, we shall see examples of other kinds of mappings.

Fact table - cube

- A cube is a named collection of measures and dimensions. The one thing the measures and dimensions have in common is the fact table, here "exam". As we shall see, the fact table holds the columns from which measures are calculated, and contains references to the tables which hold the dimensions.

```
<Cube name="exam" cache="true" enabled="true">
```

```
<Table name="rilevazioni_ft" schema="public">
```

```
</Table>
```

```
...
```

```
</Cube>
```

- The fact table is defined using the <Table> element. If the fact table is not in the default schema, you can provide an explicit schema using the "schema" attribute, for example

```
<Table schema="custom_schema" name="rilevazioni_ft"/>
```

Dimensions

- We'll also have various dimensions:

```
<Dimension name="Paziente">
```

```
  <Hierarchy hasAll="true">
```

```
    <Level name="Paziente" column="id" type="String"  
    uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
```

```
  </Level>
```

```
  </Hierarchy>
```

```
</Dimension>
```

- It comes very handy to use “dimension usage”

```
<DimensionUsage source="Paziente" name="Paziente" foreignKey="id">
```

```
</DimensionUsage>
```

Measures

- The exam cube defines several measures, including "n" and "Numero"

```
<Measure name="n" column="n" formatString="###.###" aggregator="avg">  
</Measure>
```

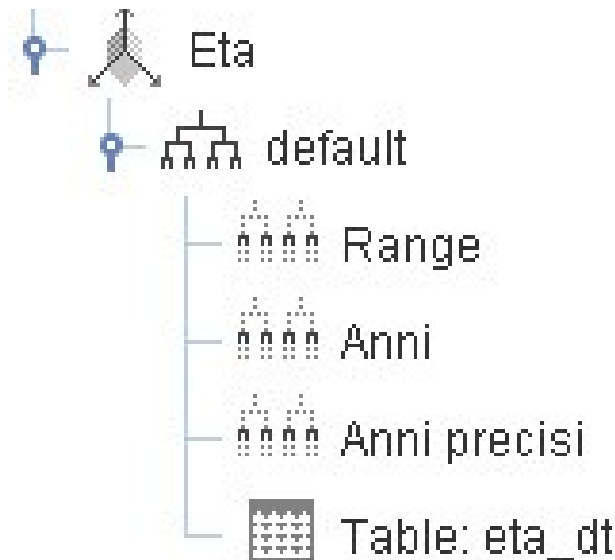

...

```
<Measure name="Numero" column="id" formatString="###" aggregator="distinct-  
count">  
</Measure>
```
- Each measure has a name, a column in the fact table, and an aggregator.
- The aggregator is usually "sum", but "count", "min", "max", "avg", and "distinct count" are also allowed; "distinct count" has some limitations if your cube contains a parent-child hierarchy.

Members, Hierarchies, Levels

- A member is a point within a dimension determined by a particular set of attribute values. The 'gender' dimension has the two members 'M' and 'F'. 'AU', 'DW' and 'HH' are all members of the 'patologia'
- A hierarchy is a set of members organized into a structure for convenient analysis. For example, the 'data' hierarchy consists of the year, month and day. The hierarchy allows you form intermediate sub-totals or group count: the sub-total of patients examined in certain period and their clinical features.
- A level is a collection of members of the hierarchy which have the same distance from the root of the hierarchy

Hierarchy - example

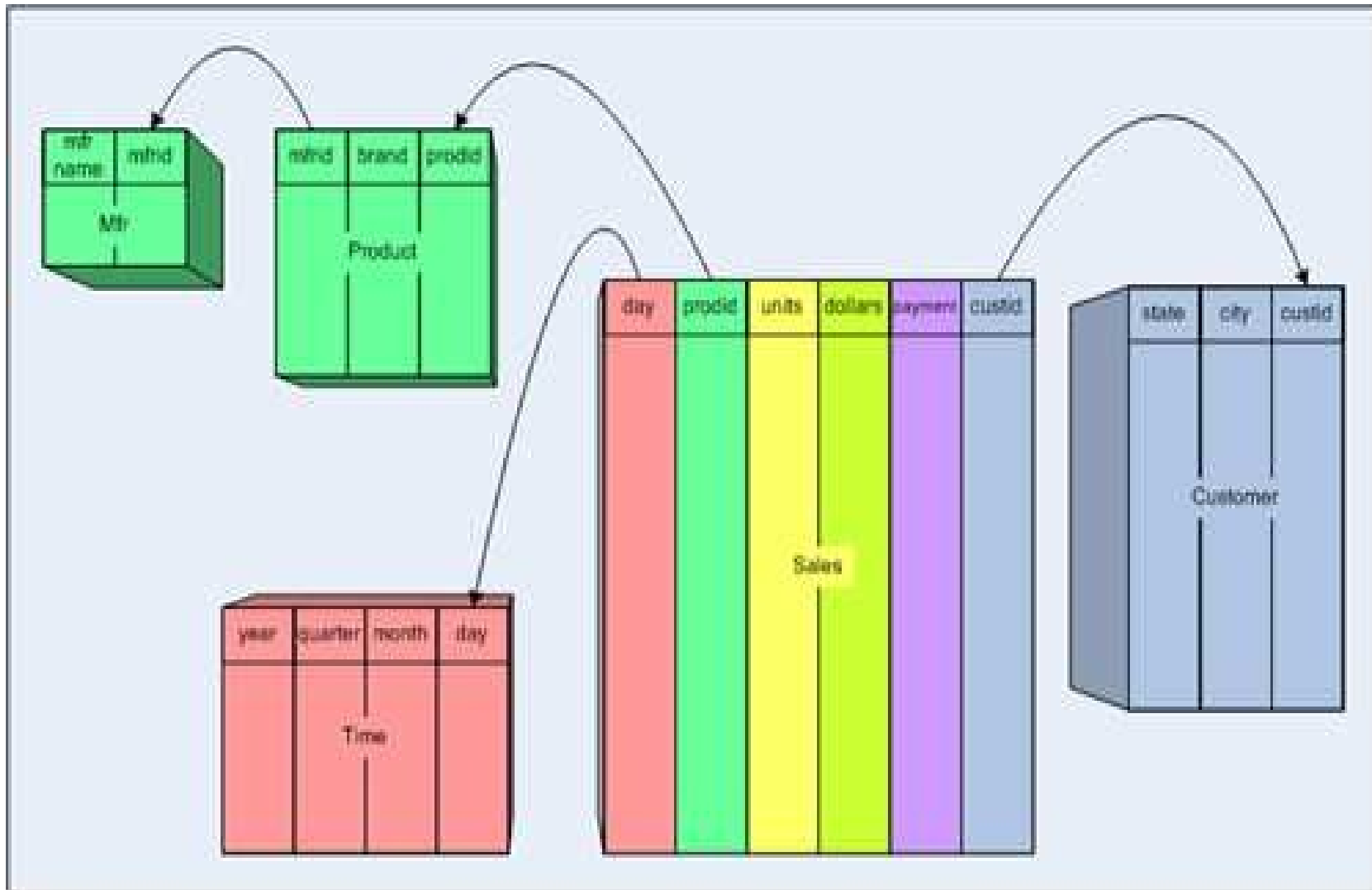


```
<Dimension name="Eta">
  <Hierarchy hasAll="true" allMemberName="Tutte le eta&#39;" primaryKey="eta">
    <Table name="eta_dt" schema="public">
      </Table>
      <Level name="Range" column="range" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
      <Level name="Anni" column="etaint" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
      <Level name="Anni precisi" column="eta" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
```

Aggregation and storage strategies

- Database store fact data in multidimensional format, but if there are more than few, this data will be sparse, and the multidimensional format doesn't perform well
- Pre-computed aggregates are necessary for large data sets, otherwise certain queries could not be answered without reading the entire contents of the fact table
- The final component of the aggregation strategy is the cache. The cache holds pre-computed aggregations in memory so subsequent queries can access cell values without going to disk. If the cache holds the required data set at a lower level of aggregation, it can compute the required data by rolling up
- The cache is arguably the most important part of the aggregation strategy because it is adaptive

Aggregate tables - Example



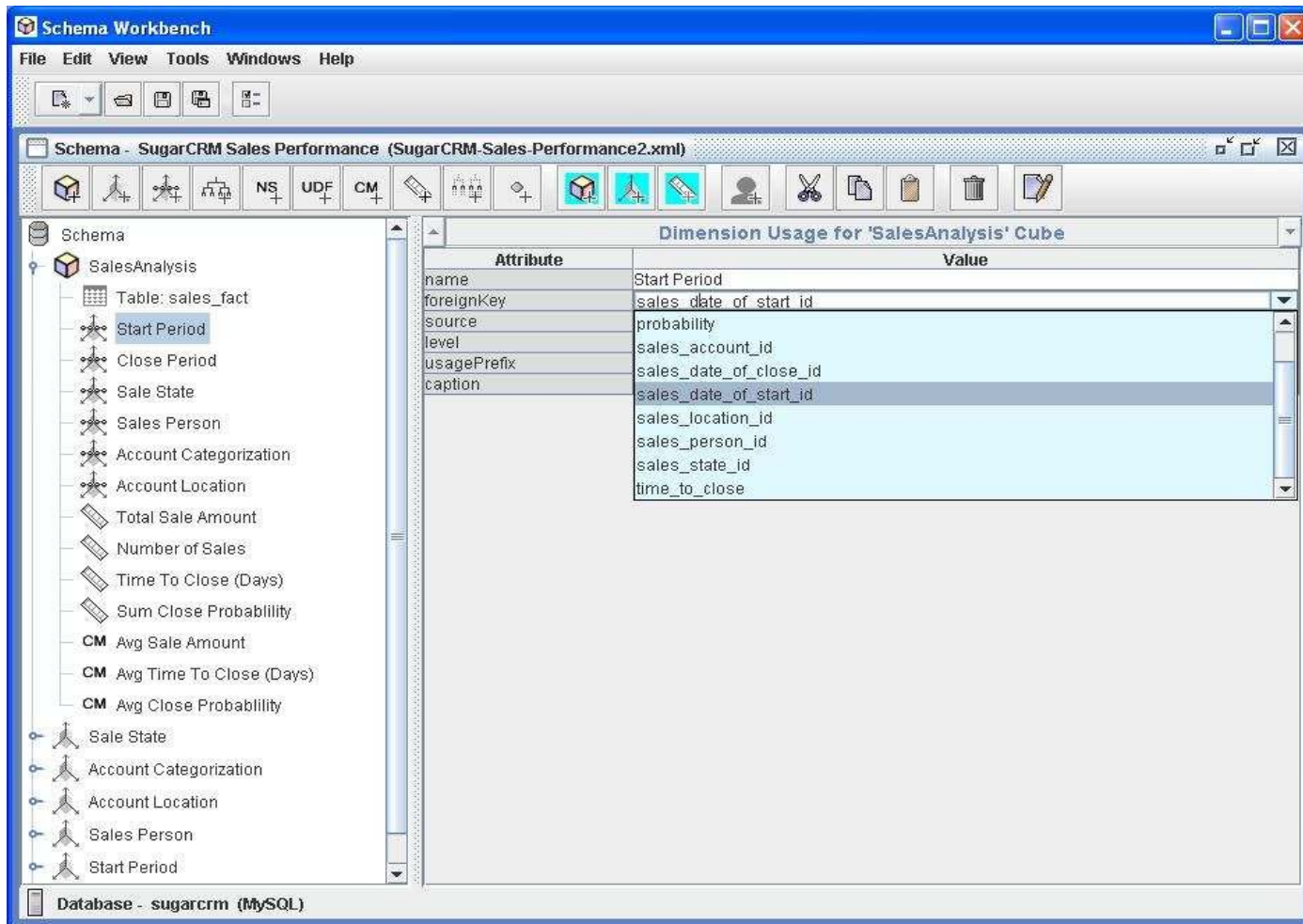
Aggregate tables – cont.

```
<Cube name="Sales">
  <Table name="sales">
    <AggName name="agg_1">
      <AggFactCount column="row count"/>
      <AggMeasure name="[Measures].[Unit Sales]" column="sum units"/>
      <AggMeasure name="[Measures].[Min Units]" column="min units"/>
      <AggMeasure name="[Measures].[Max Units]" column="max units"/>
      <AggMeasure name="[Measures].[Dollar Sales]" column="sum
dollars"/>
      <AggLevel name="[Time].[Year]" column="year"/>
      <AggLevel name="[Time].[Quarter]" column="quarter"/>
      <AggLevel name="[Product].[Mfrid]" column="mfrid"/>
      <AggLevel name="[Product].[Brand]" column="brand"/>
      <AggLevel name="[Product].[Prodid]" column="prodid"/>
    </AggName>
  </Table>

  <!-- Rest of the cube definition -->
</Cube>
```

Schema Workbench – example

Create or edit elements in the schema. The Workbench validates your changes against the cube database tables and column names.



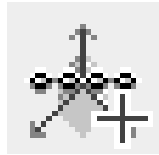
Main components



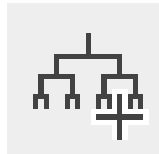
Add Cube



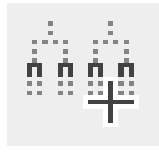
Add Dimension



Add Dimension Usage



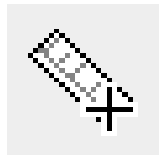
Add Hierarchy



Add Level



Add Calculated Member



Add Measure

Workshop II – building the cube

The goal of this second workshop is to build a cube starting from the database from the previous workshop

So the cube will have:

- 3 Dimensions: Region, Department, Positions
- 3 Measures: Actual, Budget, Variance
- 1 calculated measure of your choice (optional)

Database connection

Database Connection

General
Advanced
Options
Pooling
Clustering

Connection Name:
labsia

Connection Type:
MaxDB (SAP DB)
MonetDB
MySQL
Neoview
Netezza
Oracle
Oracle RDB
Palo MOLAP Server
PostgreSQL
Remedy Action Request System
SAP ERP System
SQLite
Sybase
SybaseIQ
Teradata
UniVerse database
Vertica

Access:
Native (JDBC)
ODBC
JNDI

Settings

Host Name:
157.27.243.188

Database Name:
labsia00

Port Number:
5432

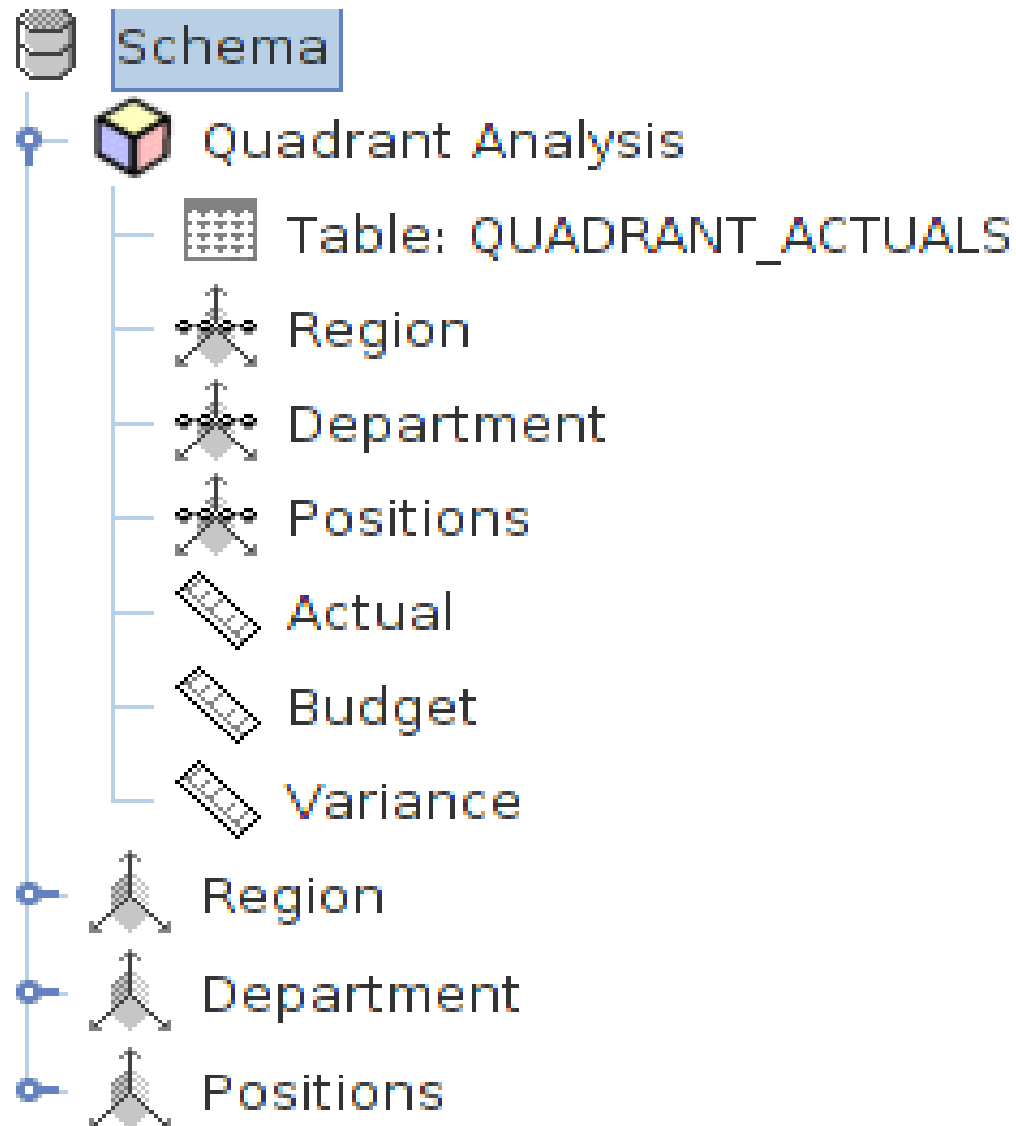
User Name:
userlab00

Password:
.....

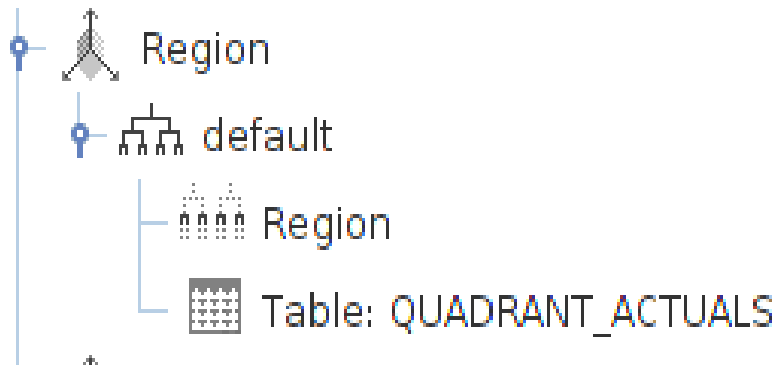
Test

OK Cancel

Workshop - schema structure



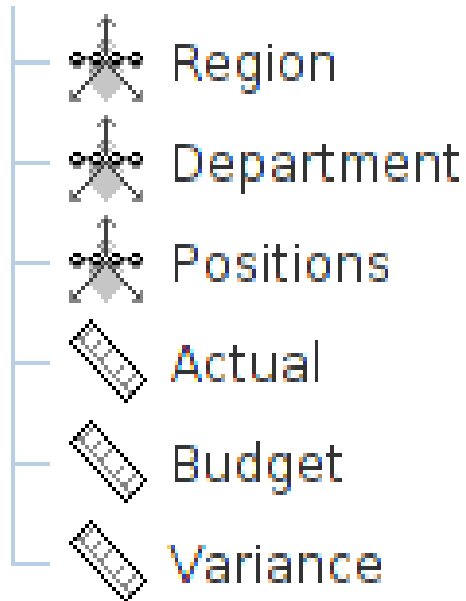
Dimensions



- Set schema and table name in “table”

```
<Dimension highCardinality="false" name="Region">  
  <Hierarchy hasAll="true" allMemberName="All Regions">  
    <Table name="QUADRANT_ACTUALS">  
    </Table>  
    <Level name="Region" column="REGION" type="String" uniqueMembers="true" levelType="Regular" hideMemberIf="Never">  
    </Level>  
  </Hierarchy>  
</Dimension>
```

Dimension usage and measure



- Select the source in the “usage dimension”
- Select the column and aggregator in “measures”

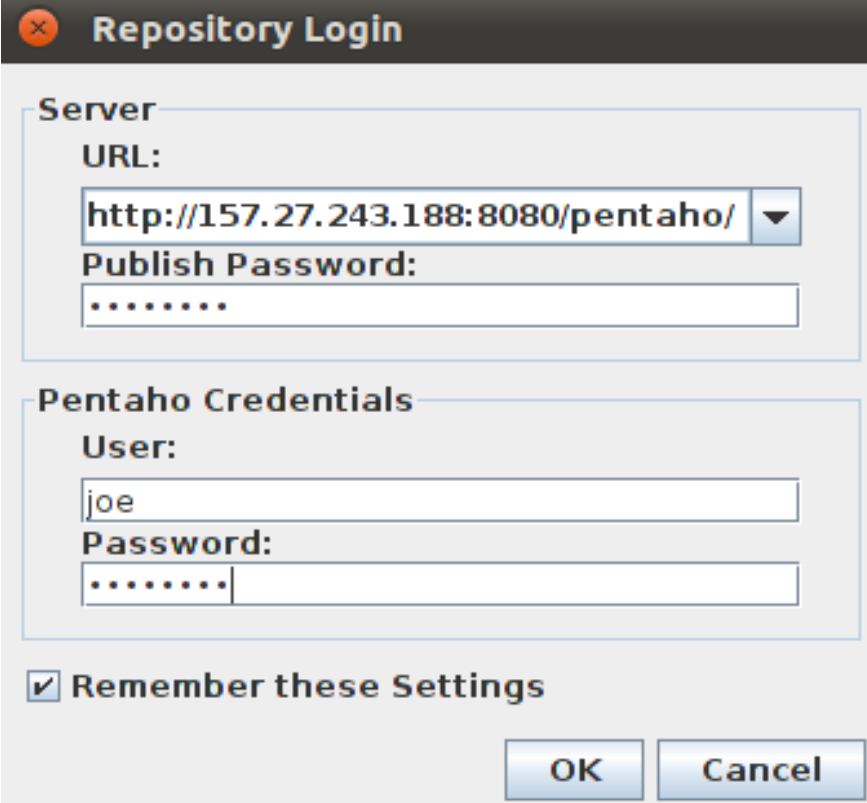
```
<DimensionUsage source="Region" name="Region" highCardinality="false">  
</DimensionUsage>
```

```
<Measure name="Actual" column="ACTUAL" formatString="#,###.00" aggregator="sum">  
</Measure>
```

Publishing the schema

Note: remember to start the bi-server (`./start-pentaho.sh`)

- Edit the server's publish password, located in `publisher_config.xml` in `pentaho-solutions/system`
- Go select Publish.. from “File” menu
- Set the data as described below:



The screenshot shows a dialog box titled "Repository Login" with a close button (red X) in the top-left corner. The dialog is divided into two main sections: "Server" and "Pentaho Credentials".

Server Section:

- URL:** A text box containing `http://157.27.243.188:8080/pentaho/` with a dropdown arrow on the right.
- Publish Password:** A text box containing seven dots, indicating a masked password.

Pentaho Credentials Section:

- User:** A text box containing the text "joe".
- Password:** A text box containing seven dots, indicating a masked password.

At the bottom of the dialog, there is a checked checkbox labeled "Remember these Settings". Below the checkbox are two buttons: "OK" and "Cancel".