

Migen Data Visualization Using Oracle(Ver 1.0)

Zheng Zhu
zheng@dcs.bbk.ac.uk

User Manual
October 7, 2011

Abstract

The following document describes that how to use tools to visualize the data in Migen Project. Data visualization facilitates the understanding of user interaction within Migen framework and may help teachers to develop more proper tasks based on users' performance.

1 Introduction

The MiGen project is tackling some thorny problems that confront all teachers of mathematics: What is algebra for? How is it useful for expressing generalisations? And more fundamental still: What does it mean to generalise in mathematics? The development team has finished the design, implement and test stages and delivered the product to the schools. MiGen Product has been delivered to the end users, however, we also work on data visualization part to help teachers to better understand students' performance through those tools.

We store the user data in the online database which is convenient to insert, update the record. However, it is not good at handle complex queries. To solve that issue, we need make use of data warehouse to organize the data in such way that is easy to access the data, process the results in summarized level. To get the data out is useful for decision making.

Analytic Workspace Manager for Oracle OLAP is the tool we used for data visualization. We will describe how to map from data warehouse attributes to the dimensions of cubes, and maintain and view the cubes.

2 Methodology

We use Oracle data warehouse to store the historical data and we analyze data on the top of data warehouse. Oracle OLAP cubes deliver sophisticate calculations using simple SQL queries. Cube is used to view the results. The tool we adopted is Analytic Workspace Manager 11g, it is available on <http://www.oracle.com/technetwork/database/options/olap/olap-downloads-098860.html>. If you wan to explore the data set, you need to download it first.

2.1 Basic Concepts

A cube is a useful concept for explaining multidimensionality.

A single cube is often loaded from a fact table and the associated dimension tables. The cube is presented as a star schema. End users can interact with the data in an easy-to-understand, intuitive fashion. There are some technical concepts related to the cube:

- Dimensions
- Hierarchies
- Members

A dimension is a collection of items that share some attributes or characteristics. The concept of a hierarchy is intuitive. For example, we organize the date in a hierarchy.

- Year
 1. Month
 - (a) Date

The relationships between members within a dimension define a dimension hierarchy. For each dimension, we can have multiple hierarchies. For example, we have status hierarchy and category hierarchy for indicator dimension. Another widely used hierarchy in business model is location based hierarchy. For example, a sales for a company are broken down by country and then provinces for China, and then city in each province.

Dimensions (such as indicator type, Date) form the edges of the cube. Members from each dimension create intersections within the cube, each of which can potentially hold a data value, the measure. One example of OLAP Cube is shown in Figure 1.

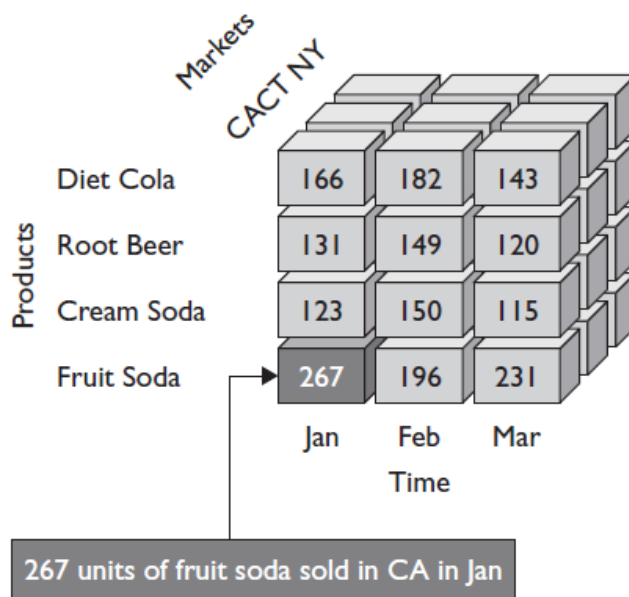


Figure 1: Dimensions represented in an Cube

An inherent value of an OLAP system is the ability for business users to analyze data in a way that make sense to them. One way to provide alternate views of the data is through user-defined attributes or groupings. An attribute is a tag or property assigned to a member. Therefore an attribute is typically text information like “SimpleEventIndicator”. An OLAP cube aggregates the data within its hierarchies, using the parent-child relationships revealed in the hierarchy views. The OLAP engine **does not** calculate aggregates over dimension attribute values¹.

Measures represent business data that is important for analysis, such as sales, in our case, the quantity of indicator. Measure values are the data that fills the intersections in a cube. To be meaningful, a value must be defined in terms of all dimensions in the cube as shown in Figure 1. A Measure supports aggregation operation, i.e., sum or average.

OLAP systems leverage the concept of hierarchies for calculation purposes, providing default calculations simply by aggregating the values of the members up the hierarchy.

We use Analytic Workspace Manager 11g (AWM 11g) to visualize the data. AWM 11g is a tool for creating, developing, and managing multidimensional data in an Oracle 11g data warehouse. With this easy-to-use GUI tool, you create the container for OLAP data, an analytic workspace (AW), and then add OLAP dimensions and cubes.

¹http://download.oracle.com/docs/cd/B28359_01/olap.111/b28124/query.htm#CBAIEHBA

2.2 Procedures of Setup Data Visualization

To launch AWM, we need click awm.bat under awm112020A_Standalone fold on Windows-based system. If it is the first time to open AWM, we are prompted to define a database connection. We need enter a description to identify the server and specify connection information. After we create the connection, we need connect it by providing user name and password.

To create a cube, we first need to create and populate all dimensions for a cube. The process consists of the following steps:

- Create the dimensions.
- Optionally, define summary levels (not required for value-based hierarchies).
- Organize dimension levels into hierarchies.
- Map to a data source.
- Load the dimensions.

For example, assume there are four dimensions: they are date dimension, indicator dimension, task dimension and session dimension. Date dimension contains three levels, they are year, month and dates. Indicator dimension has two hierarchies, status hierarchy contains all status level, indicator status level and indicator level and category hierarchy contains all categories level, category level and indicator level. Session dimension has all session, sessions and model level. Task dimension consists of all task, task and models levels. All the levels information form a hierarchy for each dimension. In addition to levels and hierarchies, there are attributes information. They are used to tag the members of a dimension. The last important component in dimension is mapping dimensions which specifies how to map from source data to the dimension levels. There are three basic steps to mapping dimensions:

- Choose the mapping type.
- Choose the data source.
- Map the source columns to the dimension members and attributes.

An example of mapping is shown in Figure 2. The mapping type is star schema. We select the source data in the

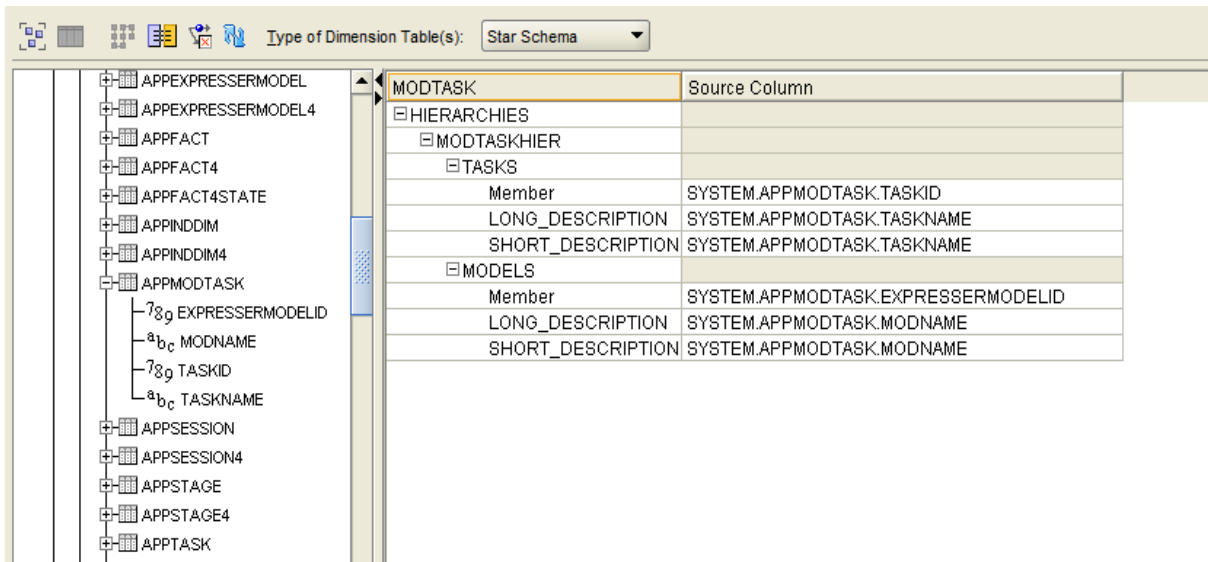


Figure 2: Mapping from Source Column to the Dimension Members and Attributes

schema list on the left panel, and drag it over to the source column of the mapping panel.

We then create two cubes, one for quantity measure and another for duration measure. Each of the cube has six dimension so far. We can make use of all dimensions we generated before, which provide the data visualization from different perspectives. After a cube container is generated, we will first create a measure under the the cube.

The mapping of cubes is similar to the mapping of dimensions. It typically involves creating joins that map data from the fact table for a summary level above the level contained in the fact table. The mapping of the quantity cube is shown in Figure 3:

INDICATOR0815	Source Column
MEASURES	
QUANTITY	SYSTEM.APPFACT.QUANTITY
DIMENSIONS	
DATES	
YEAR	
MONTH	
DATES $\Sigma \uparrow$	SYSTEM.APPDATE.DATEID
Join Condition	SYSTEM.APPDATE.DATEID = SYSTEM.APPFACT.DATEID
INDICATOR	
ALLSTATUS	
STATUS	
ALLCATEGORY	
CATEGORY	
INDICATOR $\Sigma \uparrow$	SYSTEM.APPINDDIM.INDICORTYPEID
Join Condition	SYSTEM.APPINDDIM.INDICORTYPEID = SYSTEM.APPFACT.INDICORTYPEID
MODTASK	
TASKS	
MODELS $\Sigma \uparrow$	SYSTEM.APPMODTASK.EXPRESSERMODELID
Join Condition	SYSTEM.APPMODTASK.EXPRESSERMODELID = SYSTEM.APPFACT.EXPRESSERMODELID
SESSIONS	
ALLSESSION	
SESSIONS $\Sigma \uparrow$	SYSTEM.APPSESSION.SESSION_ID
Join Condition	SYSTEM.APPSESSION.SESSION_ID = SYSTEM.APPFACT.SESSION_ID
USERS	
ALLUSER	
USERS $\Sigma \uparrow$	SYSTEM.APPUSER.USERID
Join Condition	SYSTEM.APPUSER.USERID = SYSTEM.APPFACT.USERID
Filter	

Figure 3: Cube Mapping

For detailed information about building and querying cube, please refer to <http://www.oracle.com/technetwork/database/options/olap/index.html> and http://download.oracle.com/docs/cd/B28359_01/olap.111/b28124/cubes.htm.

2.3 Procedure of Data Visualization

Often the set up stage is already done. The end user will only need to operate the AWM 11g to visualize the results.

After we run the AWM 11g, we first need to create a new Database connection by right click on the Databases. We can set Description as *migen_server*, but you can give other name as well. Connection information should be *rigel.dcs.bbk.ac.uk:1521:migen*, it is connection information and can not be changed. Figure 4 shows the details.

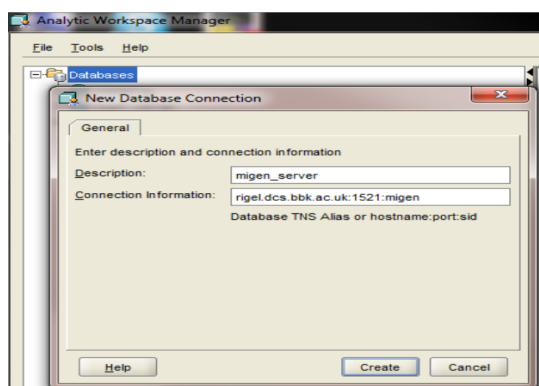


Figure 4: Database Connection Creation

We click the create button to create a database connection.

After the connection is set up, we need to connect the database through it. Click the + located before the connection name, one window will pop up as Figure 5 shows.

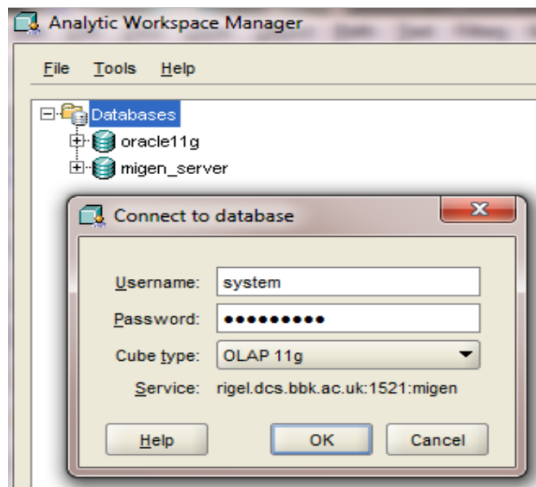


Figure 5: Connection to Database

We will input the user name and password which you can get from Migen group.

Expand the migen analytic workspaces and cubes, we can see there are two cubes under the Cubes fold: INDICATOR_CUBE AND INDICATOR_DURATION which is shown in Figure 6.

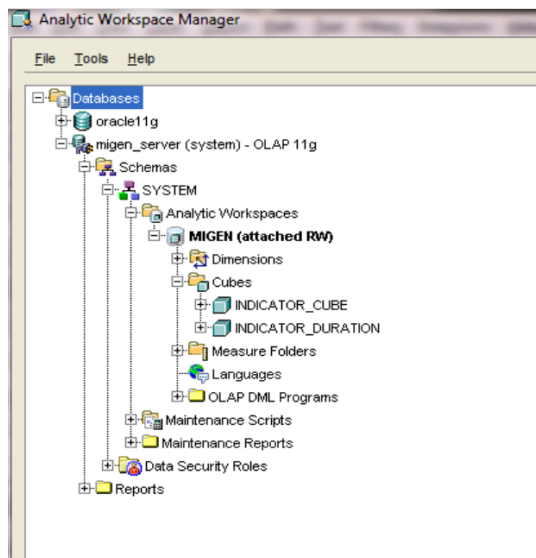


Figure 6: Interface

Right click one cube, we can either maintain cube (load the data into the cube) or view data. Here we click view data indicator_cube as shown in Figure 7.

We will see the Figure 8.

We can specify layout, dimensions and other functions by clicking Query Builder tool which is highlighted on upper left corner of Figure 8. The chart on the bottom of Figure 8 is the plot of the corresponding results displayed on the upper part on the same figure. Here we can see the total number of quantity for 2010 and 2011 from this plot.

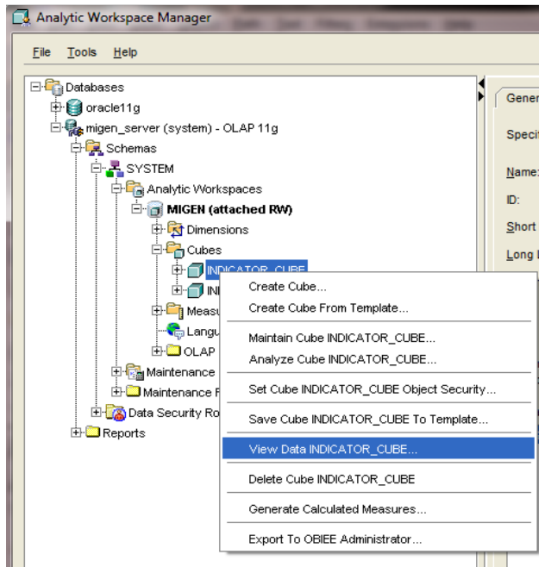


Figure 7: View Data

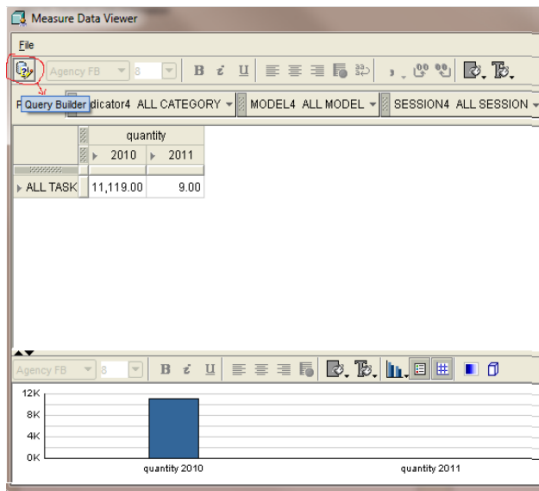


Figure 8: Data Visualization

After we click the Query Builder tool, we will see a window like Figure 9.

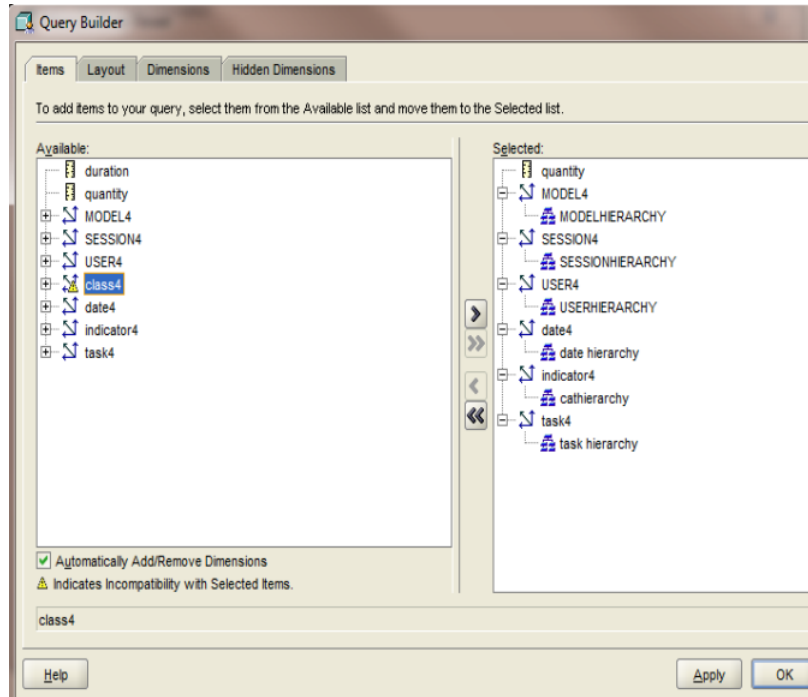


Figure 9: Query Builder Tool

There are four tabs on the top. Under items tab, we can add items to query, select them from the available list and move them to the selected list by making use of ADD tool (>) and Remove tool (<). >> means add all and << means remove all.

Under Layout tab, we can change the layout of the results. Figure 10 shows how to arrange the dimension in order to represent the results more clearly.

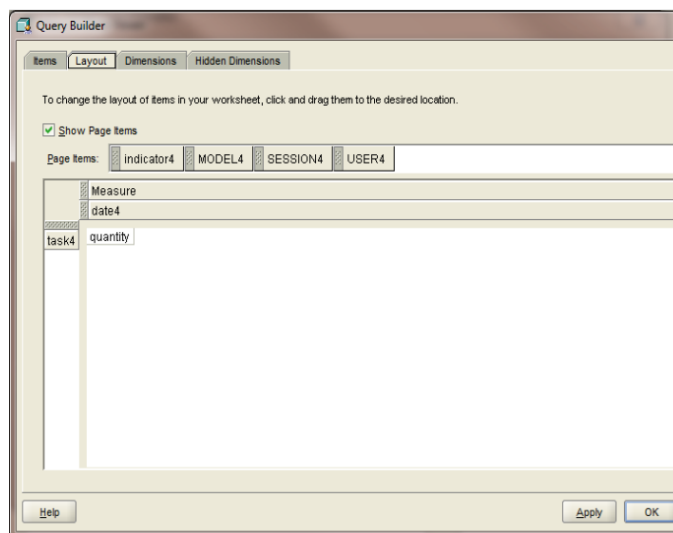


Figure 10: Query Builder Tool - Layout

We can drag the appropriate dimension tiles to the proper axis. You may decide how you want to see the data depending on your need. For example, we can drag the indicator dimension to Y-axis as top figure of Figure 11.

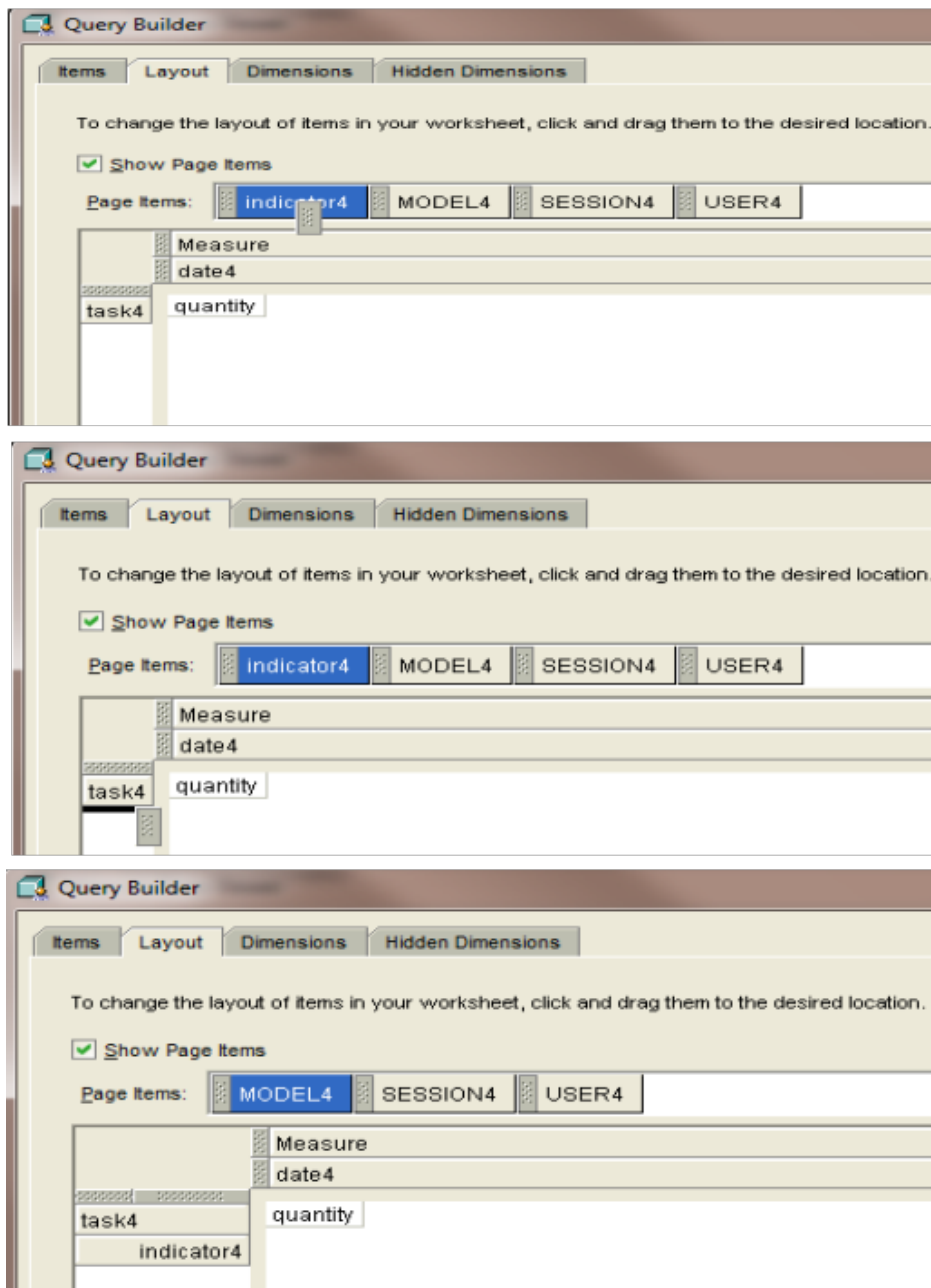


Figure 11: Query Builder Tool - Layout

We pull it to the Y-axis in the middle figure. Bottom figure shows that we manage to rearrange the layout by releasing the left click button and leaving the indicator dimension under the task dimension. Click OK and the result is shown in Figure 12.

From Figure 12, we can see the indicator dimension is underneath the task dimension. Remember the indicator dimension has category hierarchy. We can summarize how many indicators belong any category of indicator explicitly. We also can drill down or up to the different levels of the dimension hierarchy simply and efficiently. This is the power of OLAP.

Measure Data Viewer

File

Dialog 12 B z U

Page Items MODEL4 ALL MODEL SESSION4 ALL SESSION USER4 ALL USER

		quantity	
		2010	2011
ALL TASK	ALL CATEGORY	11,119.00	9.00
	DetectionStateIndicator	209.00	
	FeedbackEventIndicator	1,729.00	
	GoalEventIndicator	244.00	
	SimpleEventIndicator	7,179.00	
	VerificationStateIndicator	1,758.00	9.00
TASK 1	ALL CATEGORY	11,119.00	9.00
	DetectionStateIndicator	209.00	
	FeedbackEventIndicator	1,729.00	
	GoalEventIndicator	244.00	
	SimpleEventIndicator	7,179.00	
	VerificationStateIndicator	1,758.00	9.00
model 100	ALL CATEGORY	31.00	
	DetectionStateIndicator		
	FeedbackEventIndicator	11.00	
	GoalEventIndicator		
	SimpleEventIndicator	11.00	
	VerificationStateIndicator	9.00	

Figure 12: The Layout of the Results

The third tab is dimension tab which is shown in Figure 13.

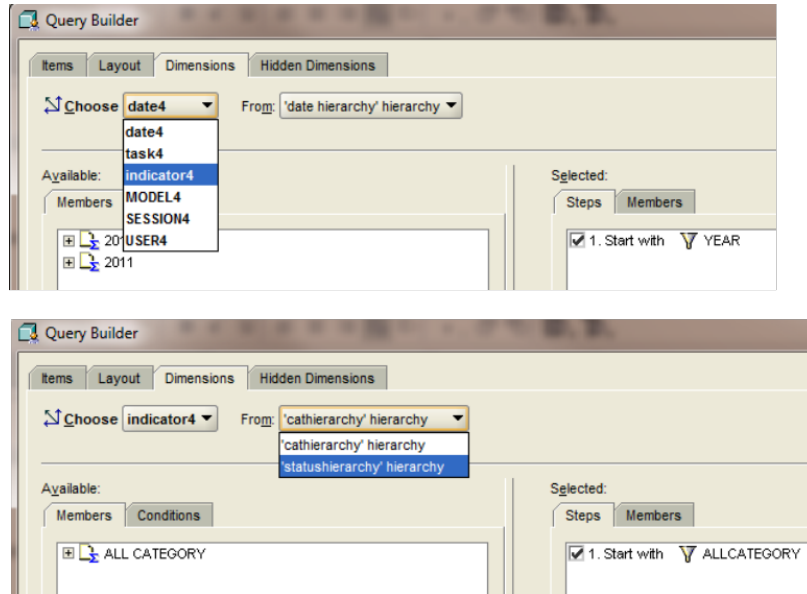


Figure 13: The Dimension Tab

We can specify the dimension and choose corresponding hierarchy. For example, here we choose indicator dimension and status hierarchy.

Furthermore, we need choose the items to display in Figure 14. Click the Remove all items tool(<<) to clear the

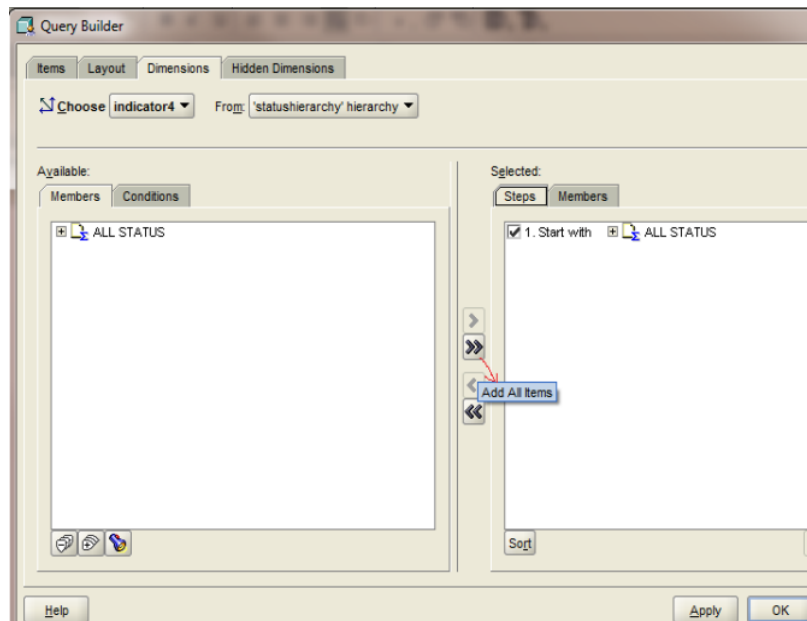


Figure 14: The Add/Remove tool in the Dimension Tab

selected list first. Then we add all status using add all items tool(>>). We can do the same on other dimensions.

The result is shown in Figure 15.

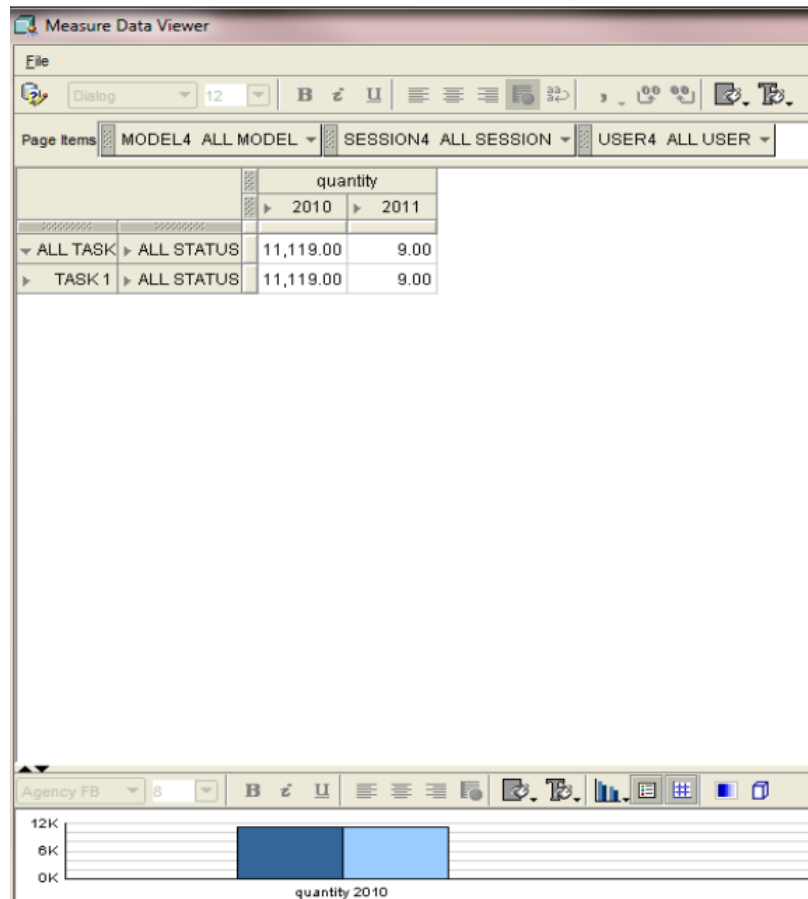


Figure 15: The Result by Modifying the Dimension Tab 1

We can expand the status by clicking on all status. And we can further expand -1 (negative) status by clicking on it, which displays all information belonging to negative indicators as Figure 16 shows.

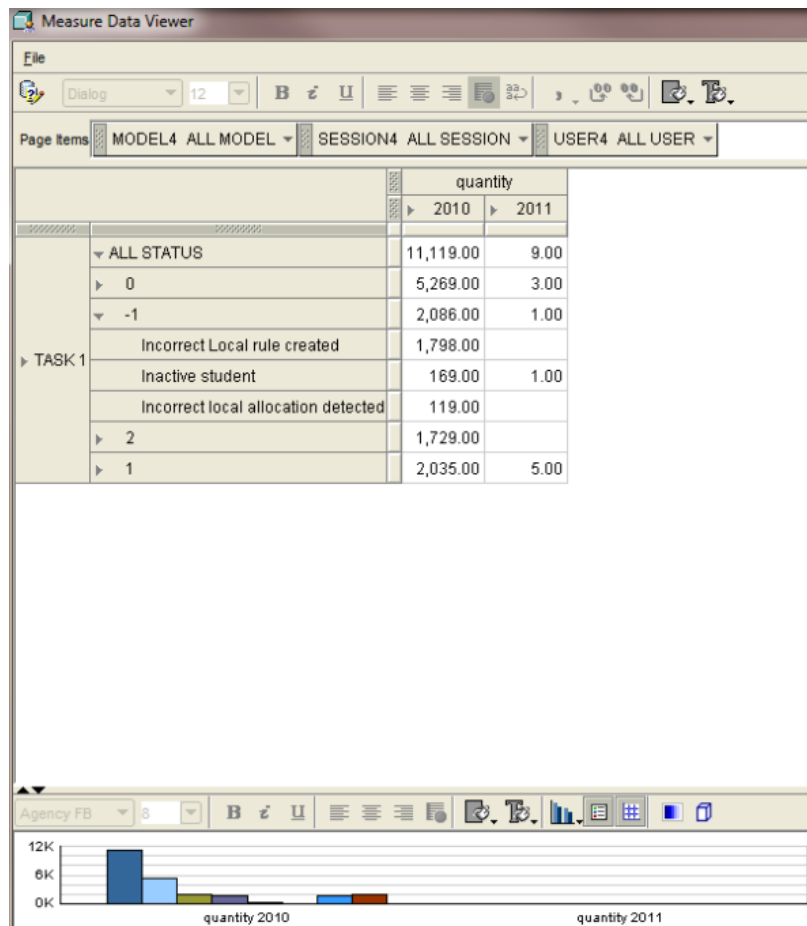


Figure 16: The Result by Modifying the Dimension Tab 2

The dimension tab also provide filter function. For example, given model dimension, we can specify a few conditions. This is very convenient to select desired subset, in particularly for the high volume data.

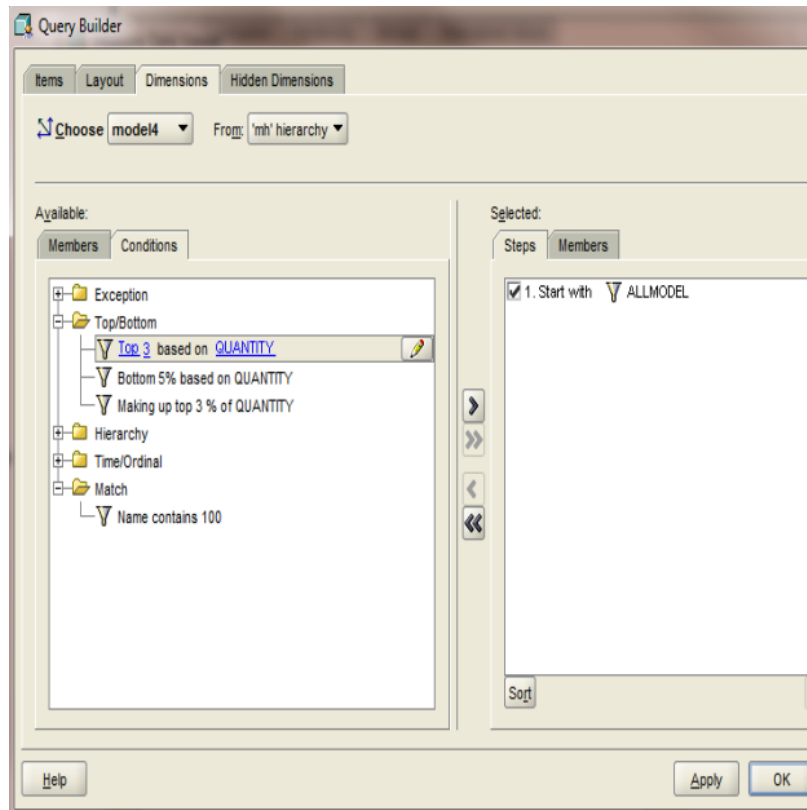


Figure 17: Conditions on the Dimension Tab

In Figure 17, we choose top 3 models based on quantity in top/bottom fold and select model 100 in match fold. After we specify these conditions, we should use Add items (>) to add them to the selected list. Similar, we can select one particular user in user dimension by specifying his username in match fold. Remember here we specify the condition on model dimension, we should drag the model dimension to x axis in Figure 18.

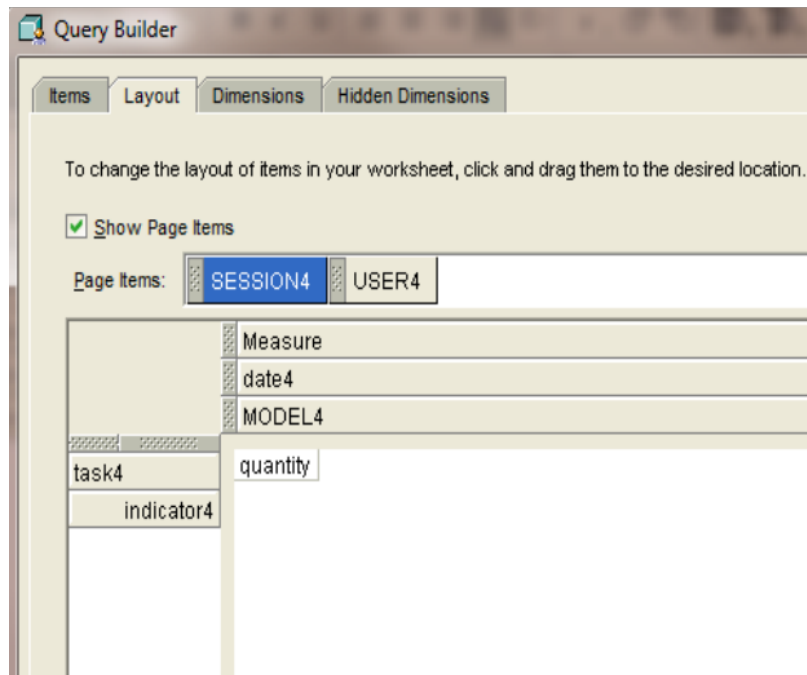


Figure 18: Update Layout to Assess the Conditions

After we click OK, we will get the results shown in Figure 19.

		quantity							
		2010				2011			
		ALL MODEL	model 100	model 159	model 161	model 141	ALL MODEL	model 100	model 159
	ALL STATUS	11,119.00	31.00	492.00	433.00	426.00	9.00		
	0	5,269.00	3.00	259.00	187.00	242.00	3.00		
▶ TASK 1	-1	2,086.00	1.00	104.00	50.00	82.00	1.00		
	2	1,729.00	11.00	72.00	93.00	60.00			
	1	2,035.00	16.00	57.00	103.00	42.00	5.00		

Figure 19: The Results Based on Conditions

We can see the model 100 is select together with another top 3 models.

We mentioned before the plot information shown on the bottom part automatically bring into correspondence with the table. We could change the graph type, i.e., from Bar graph to Pie graph shown in Figure 20.

The resulting figure is Figure 21.

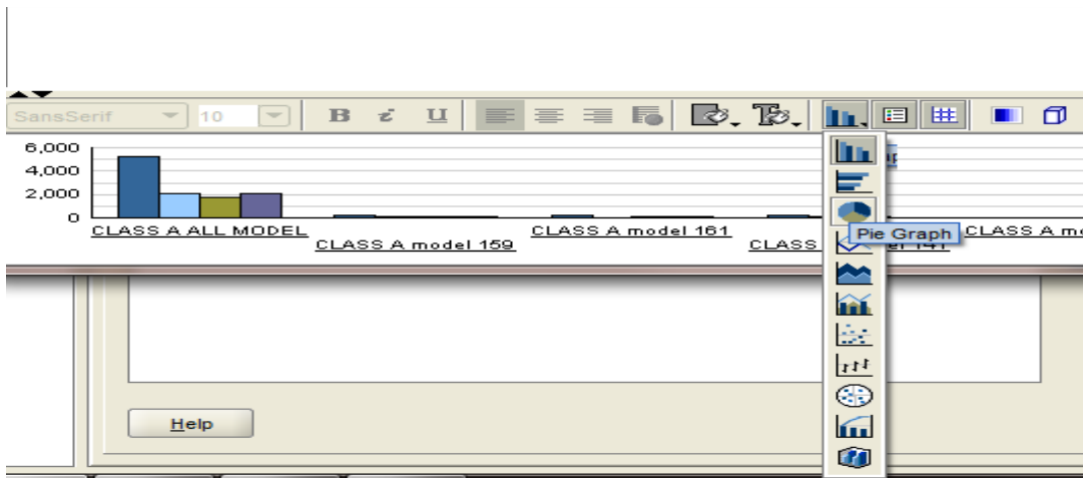


Figure 20: Change Bar Graph

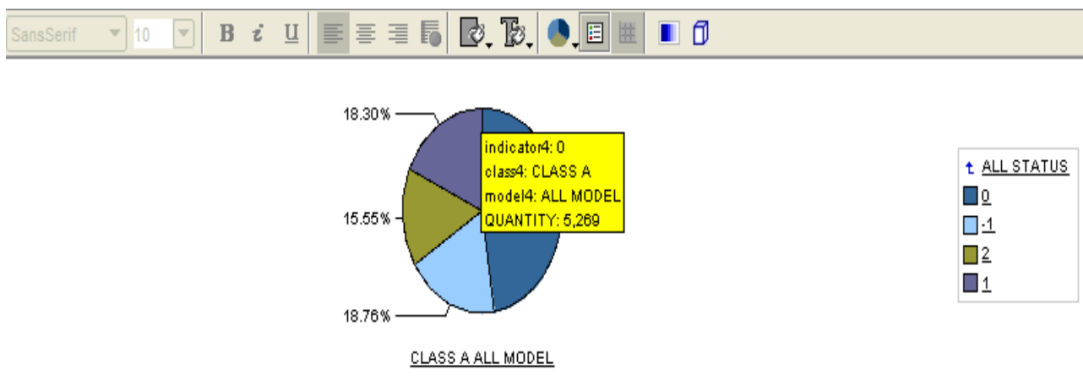


Figure 21: Pie Graph