

Building Sales Analysis Dashboard Using Hyperion Interactive Reports

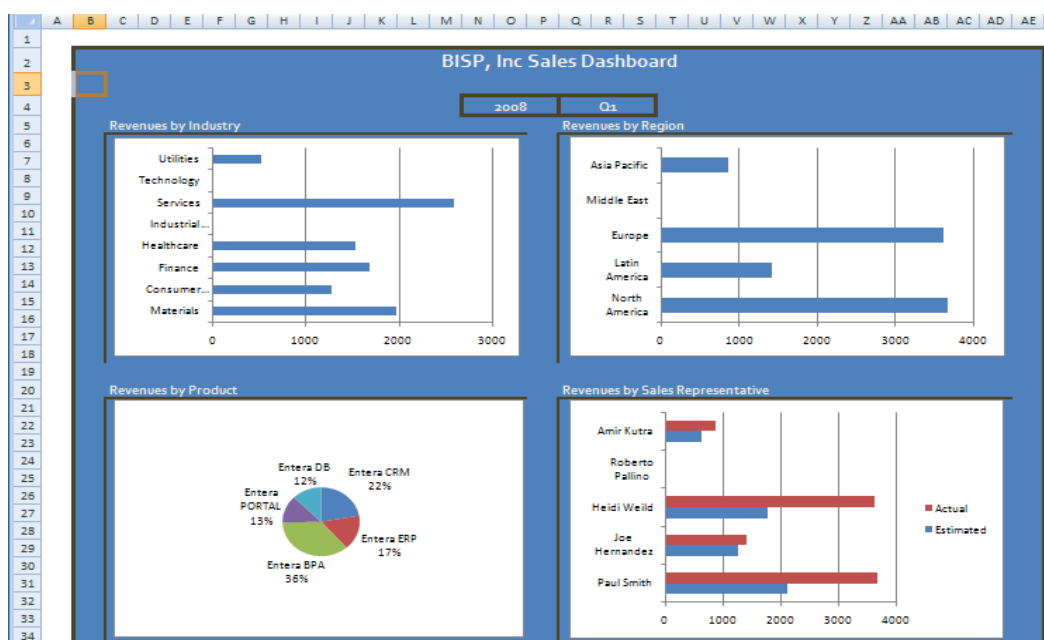
Prepared By
Amit Sharma
BISP
learnhyperion.wordpress.com

Building Sales Analysis Dashboard using Hyperion Interactive Reports

We'll see today how to build Sales Analysis Dashboard using Hyperion Interactive reports. During the session I'll be touching various components of Hyperion interactive reports and how it can be used to build rich formatted reports. We'll see the step by step approach to create.

- 1) Understanding the business requirements.
- 2) Design the structure of the requirements
- 3) Building the database connection.
- 4) Designing the Data Model [Metadata Modeling]
- 5) Creating queries and tabular reports
- 6) Defining the custom column(s)
- 7) Designing and formatting the Dashboard
- 8) Incorporating the custom control.
- 9) Adding scripting to the Dashboard
- 10) Publishing reporting in the Dashboard.

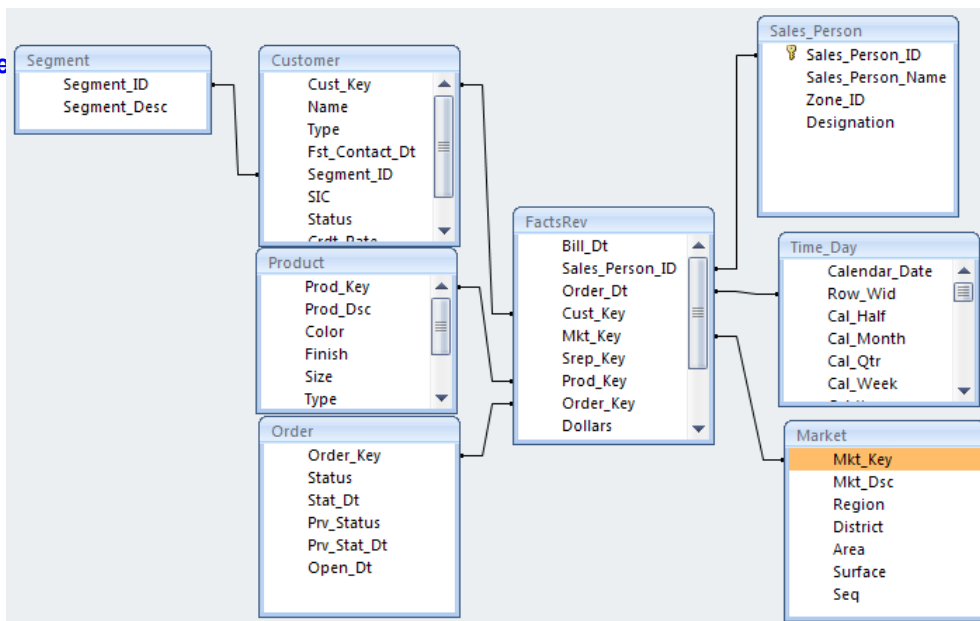
Business Requirements: As a business I was the excel sheet and asked to build the rich web based reports. The excel sheet business users were using for many year and reluctant to move to a new environment despite the current system was full of flaws and high prone to error. There were so structured approached to store and manage the data, people used to pull data from custom based .Net application and do local desktop analysis. The method was so simple just pull data from custom Sales Order Process system and export data into excel for further analysis.



As exiting reports are best means to start up requirement analysis. These reports gives you clear picture about what people like to see. However, when working with such reports you need to pay significant attention to the flaws associated with the current system.

Design the structure of the requirements: It took couple of days to me to pull the relevant table/data structure. The customized application had close to 120 tables which spans to customer, product, region, transaction, order, shipping information. In order to serve the current request I needed to do dimensional modeling and bring dimension and fact concepts from transaction based current system.

Finally I designed the model from transaction system to OLAP model consists of Dimension and facts. The proposed model is as given below.



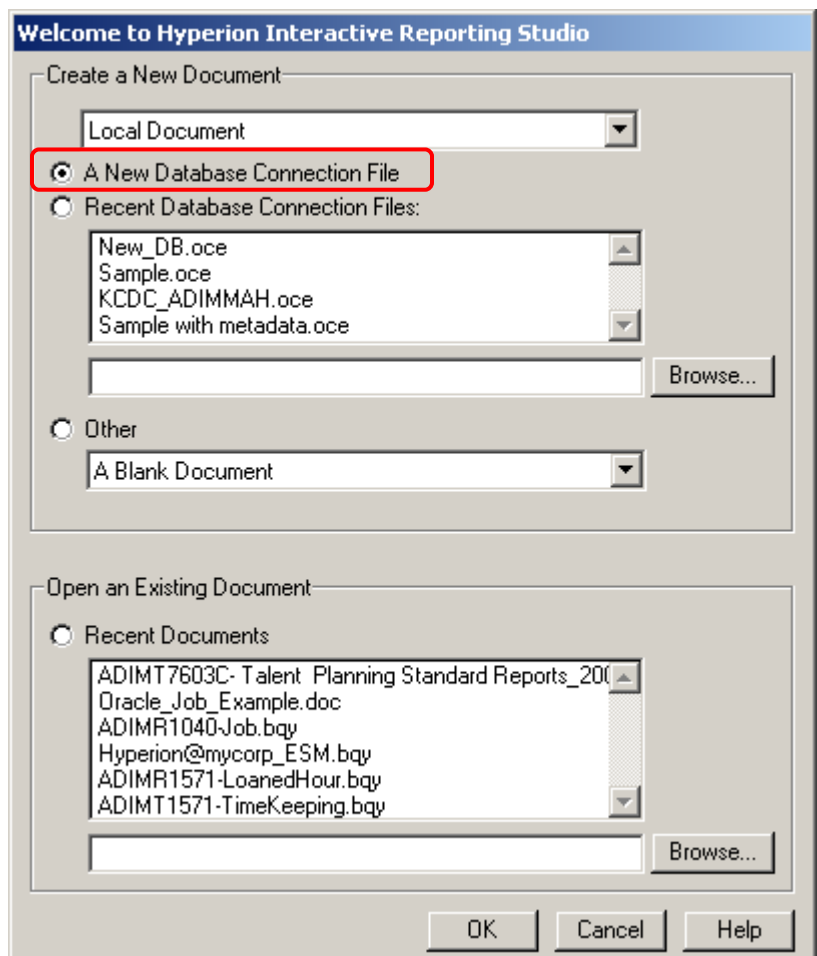
While doing physical modeling we identified the following dimensions and facts.

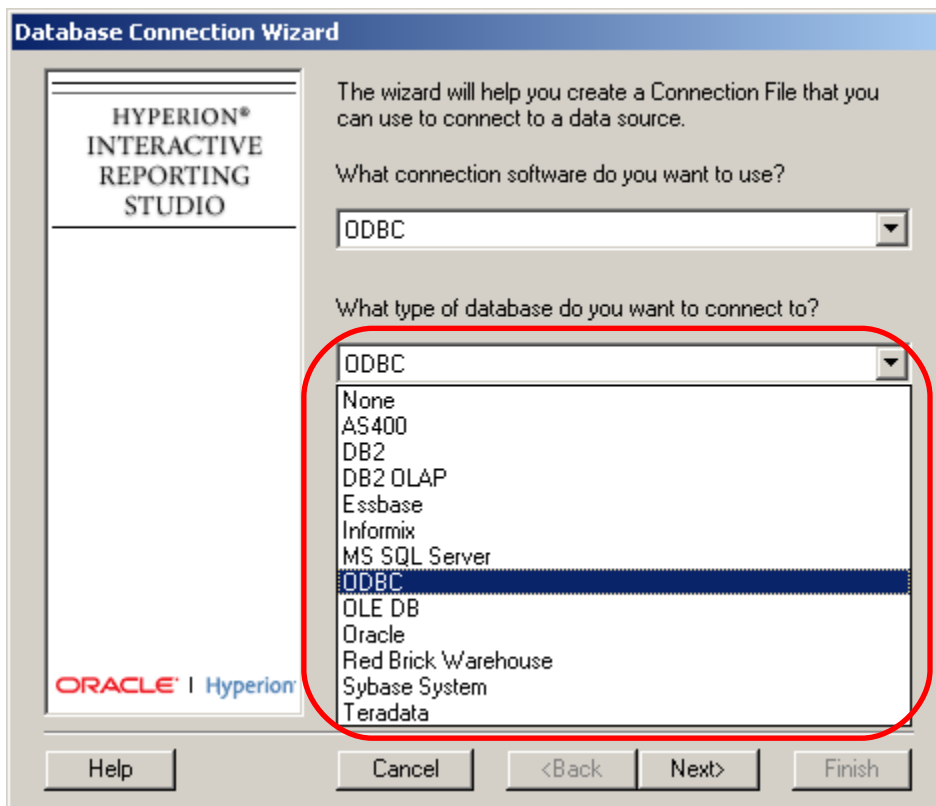
- 1) Customer
- 2) Products
- 3) Orders
- 4) Sales Person
- 5) Time Day
- 6) Market

And the fact table is **factsRev** (Fact Revenue). The above model is snowflake schema. Once we designed the model and ensure we have all to go ahead and build our reports.

Building the database connection :

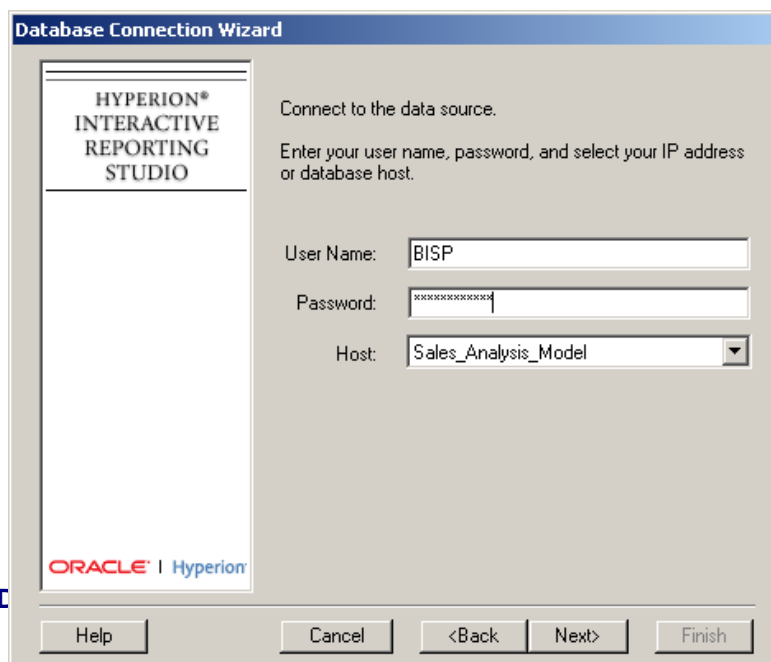
Create a new database connection
File (OCE) open database connectivity



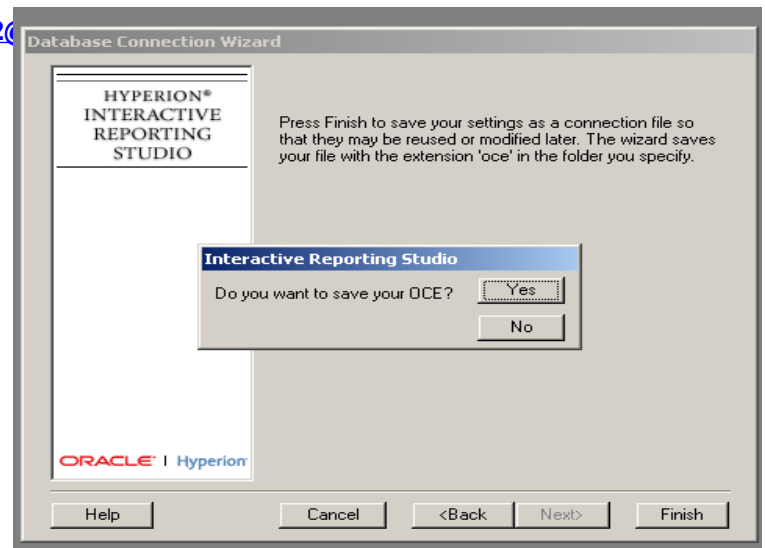


The below type of driver Hyperion IR supports

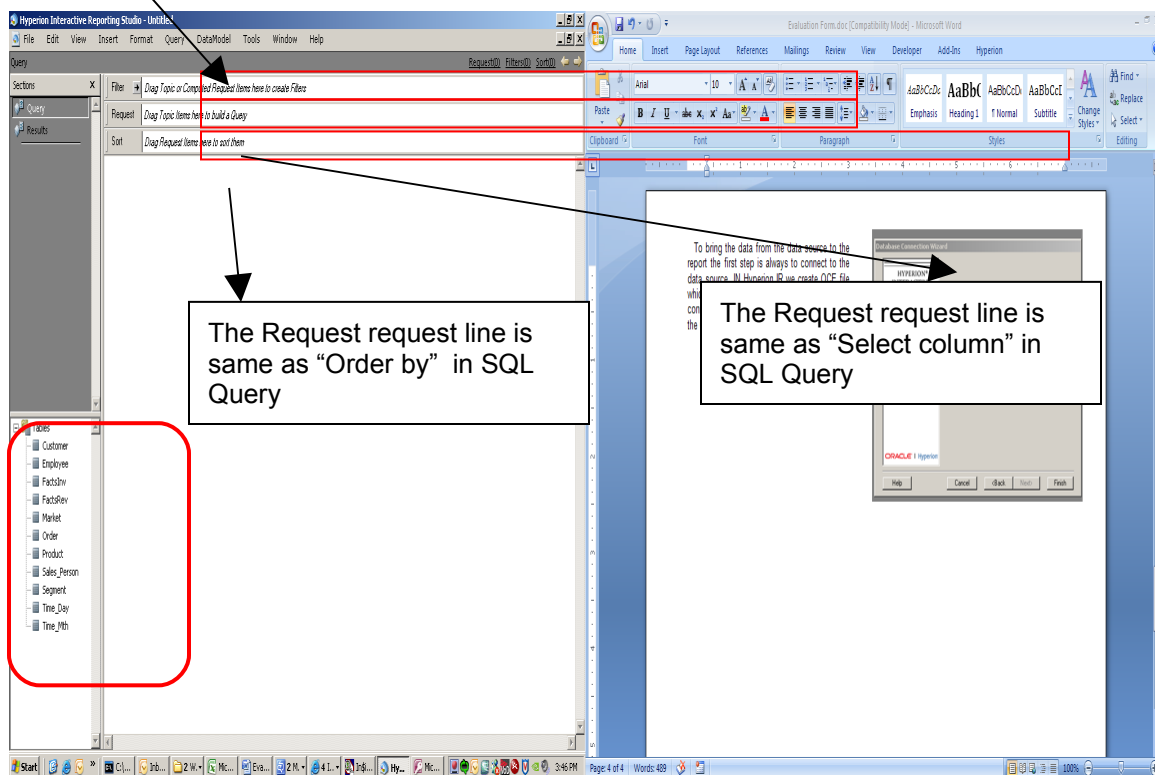
Option	Description
What connection software do you want to use?	Select the connection software to use when connecting to the database. Depending on the connection software, additional fields may be displayed. These fields enable you to customize the Interactive Reporting database connection file; show metadata settings; and select ODBC logon dialogs.
What type of database do you want to connect to?	Select the type of database to which to connect from the pull-down list.



To bring the data from the data source to the report the first step is always to connect to the data source. IN Hyperion IR we create OCE file which is connection file. The OCE file store the connection related all the information. However, the OCE does not store the password.

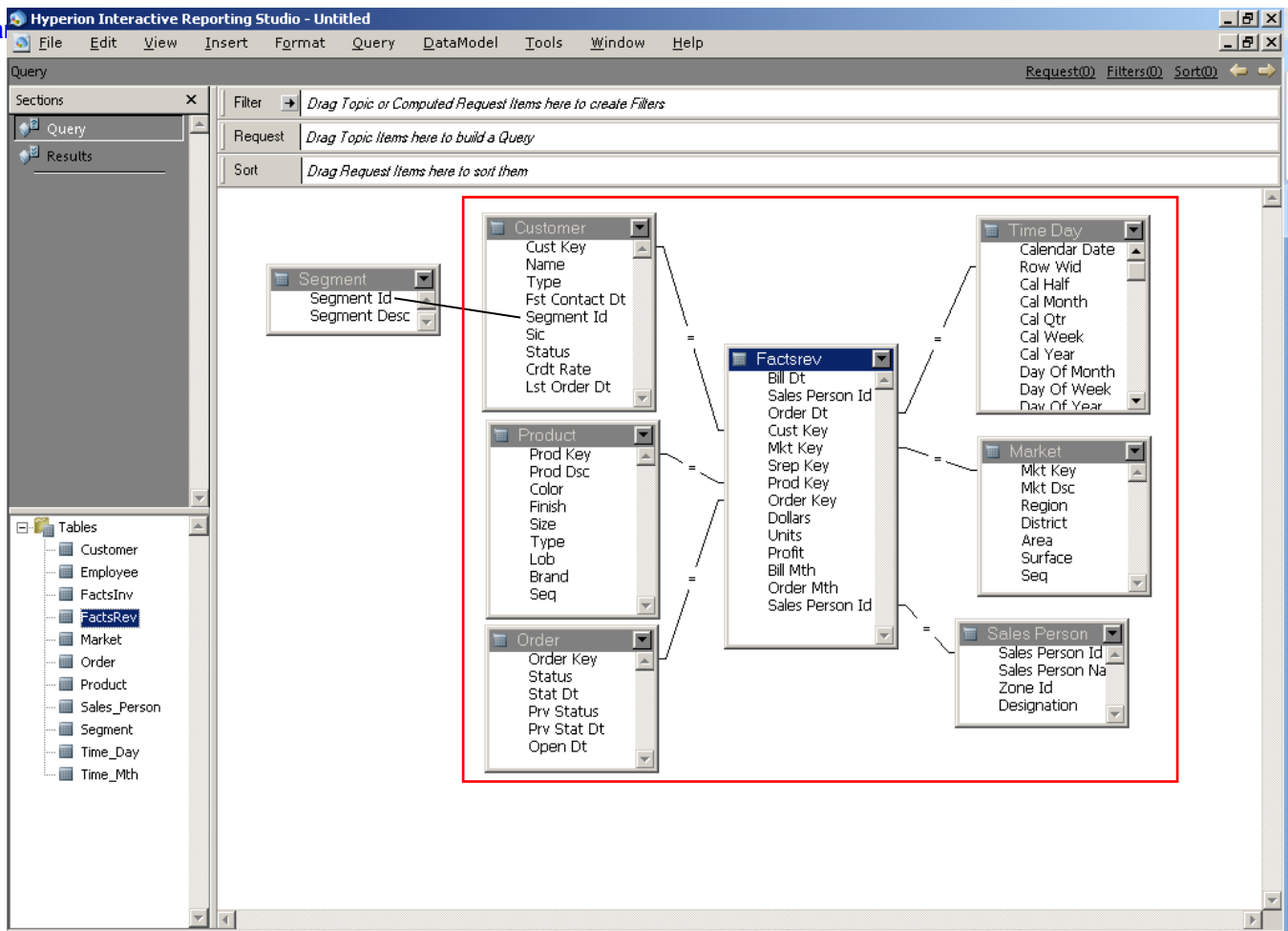


The filter request line is same as "where" clause in SQL Query



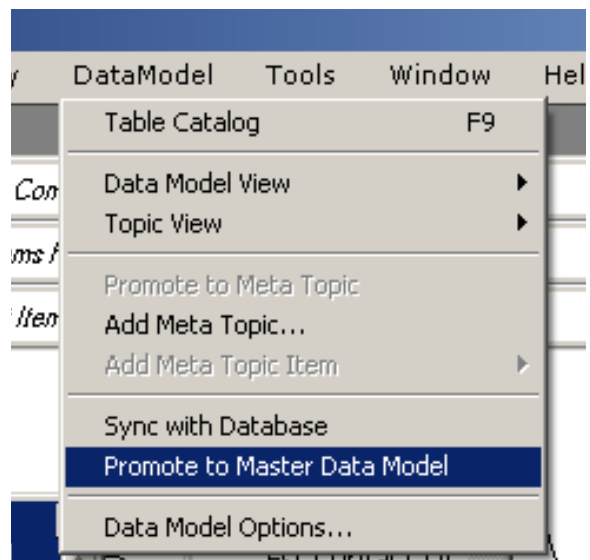
The next step is to create a data model same as it is provided to you by the data modeler. Sometimes we need to make some changes into the logical modeling. For example in the given model we may need to create multiple instance of Customer Dimension to present. It is always a good practice to create a *master data model* above this logical model. This reduces the complexities and gives simple view to the developer. Preferably one should create a logical view upon this model.

So my next step is to create a Master Data Model and hide the complexity and provide business view to the model.



The benefit of converting the current model to Master Data Model is that any changes to the master data model get propagated to all dependent queries that are based on the master data model. Each time a new query is inserted into a Interactive Reporting document file that contains a master data model, you are prompted to link the new query to the master data model. When a query is promoted to a master data model, it is added to the Section pane as a new section. Once you promote a query to a master data model, you cannot undo it.

Now your query model got frozen and it follows all the changes you do in Master Data Model.

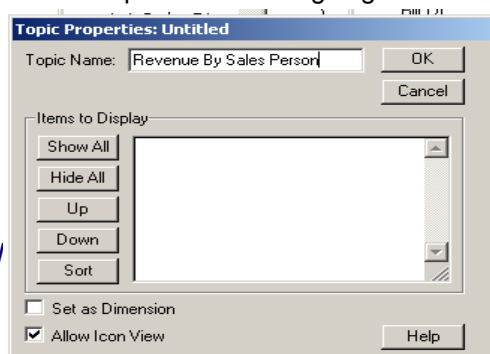


Now let's create a simple view over this complex model. I'm going to add new view [Add Meta Topic]

DataModel → Add Meta Topic

Download Hyperion/

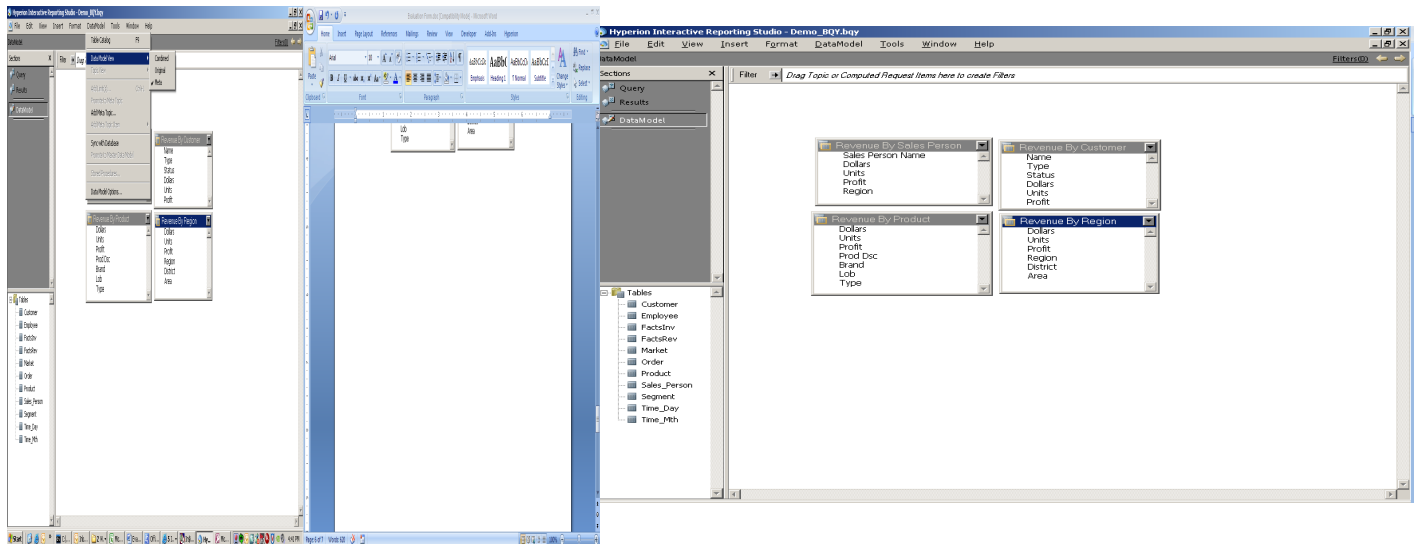
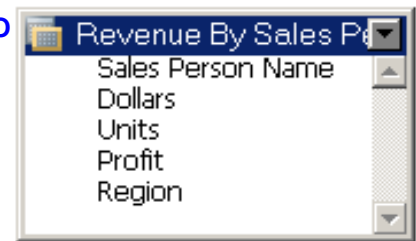
dpress.com



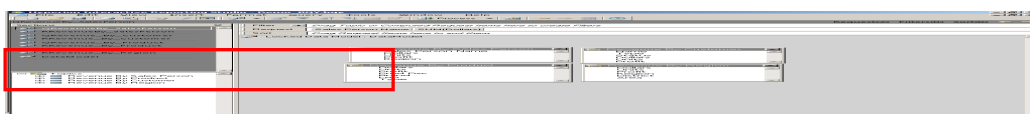
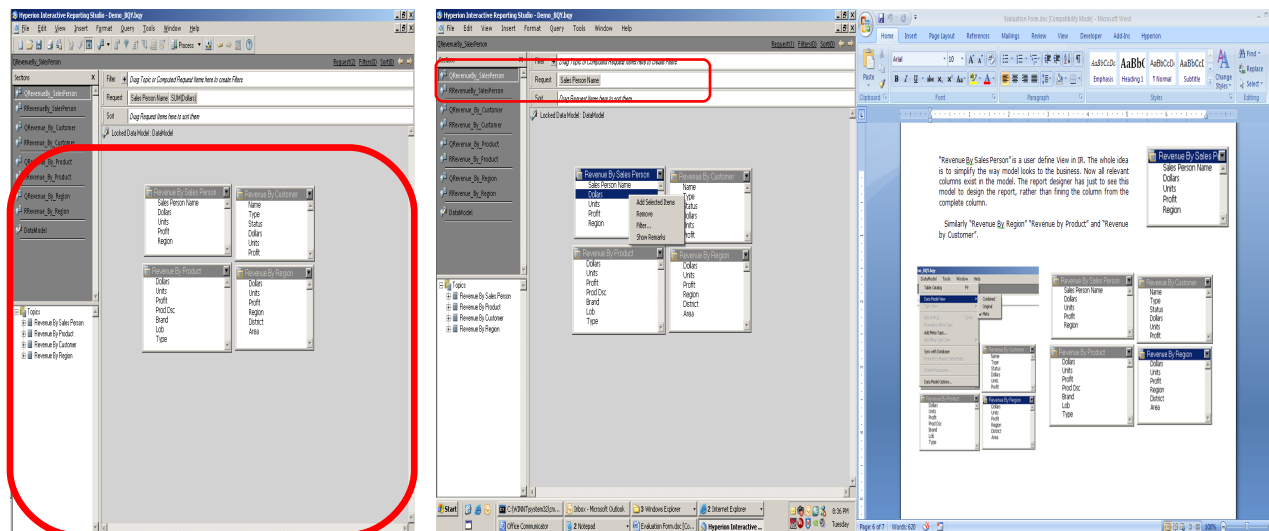
Learn Hyperion/OBIEE Tutorial

"Revenue By Sales Person" is a user define View in IR. The whole idea is to simplify the way model looks to the business. Now all relevant columns exist in the model. The report designer has just to see this model to design the report, rather than fining the column from the complete column.

Similarly "Revenue By Region" "Revenue by Product" and "Revenue by Customer".

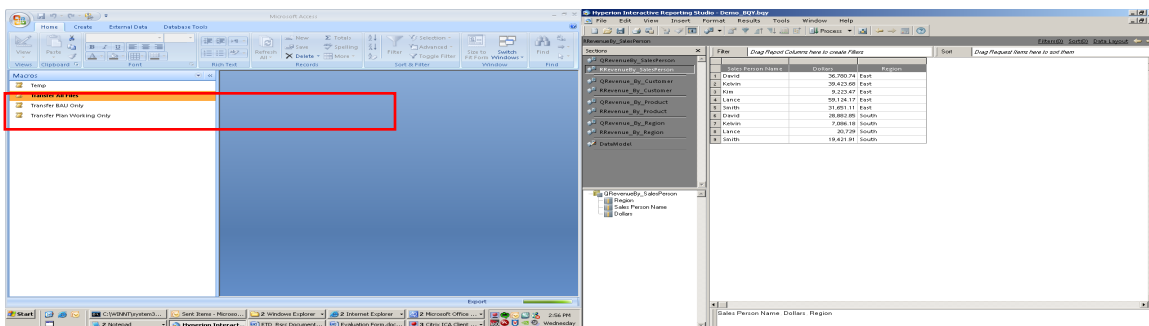


Once we designed the Meta Topic model, its time to design the actual query for four different kind of analysis we need. Finally we'll place all the query result in Dashboard and will give integrated look to all the reports.



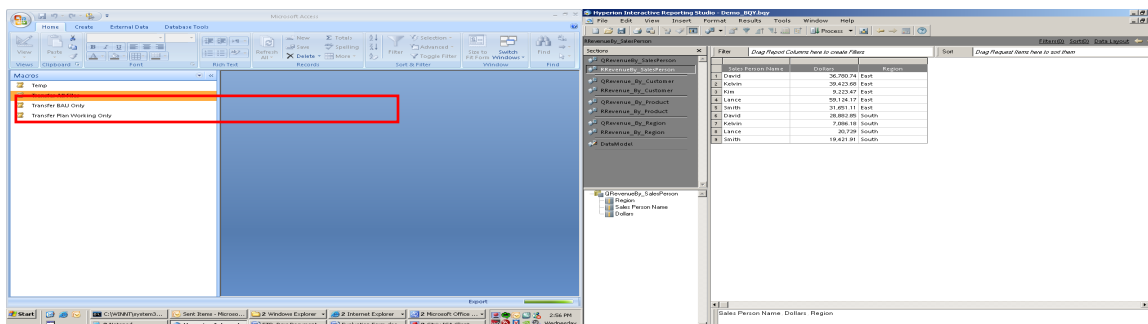
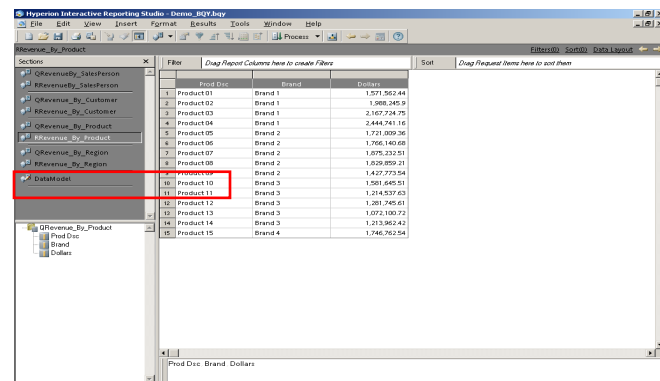
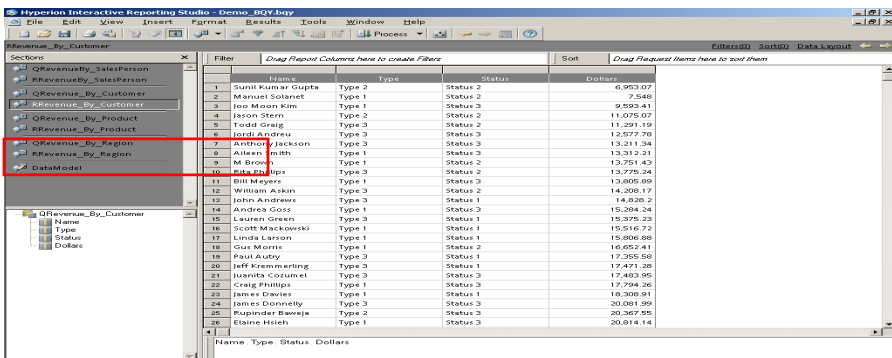
Once the query is designed, the next step is to process the query and capture. The output of the query is given below.

[Download Hyperion/OBIEE tutorials from learnhyperion.wordpress.com](http://learnhyperion.wordpress.com)



Similarly we need to create query for other 3 subject areas.

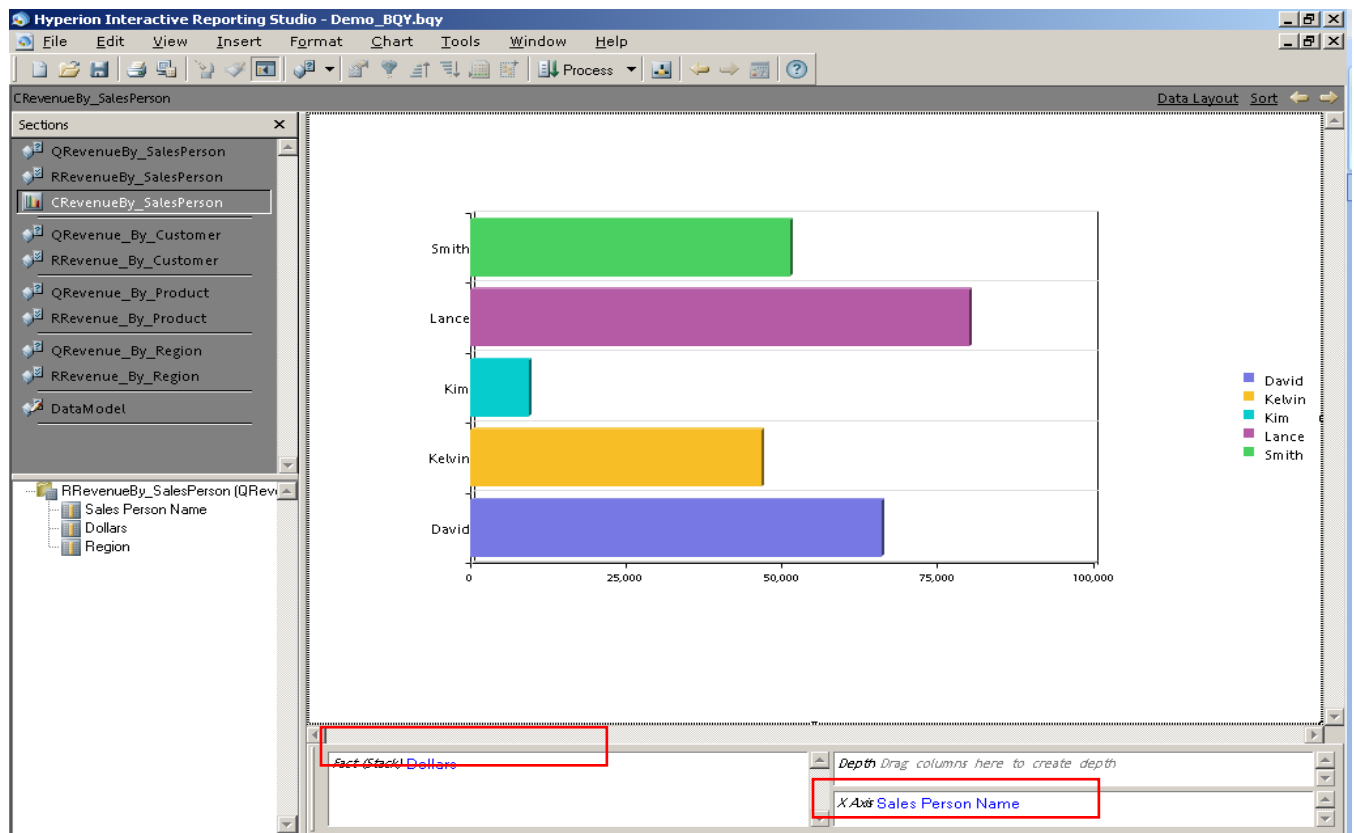
- 1) Revenue by Customer.
- 2) Revenue by Product.
- 3) Revenue by Region.



However the users want output in chart format so we need to create one chart for all the reports and place them all into Dashboard.

Use option Insert→Chart

A Hyperion interactive report support rich formatting in chart and has many kinds of chart to suite your business requirements. One can change the properties based on business needs. In the given business requirement business wants to see the horizontal bar chart.

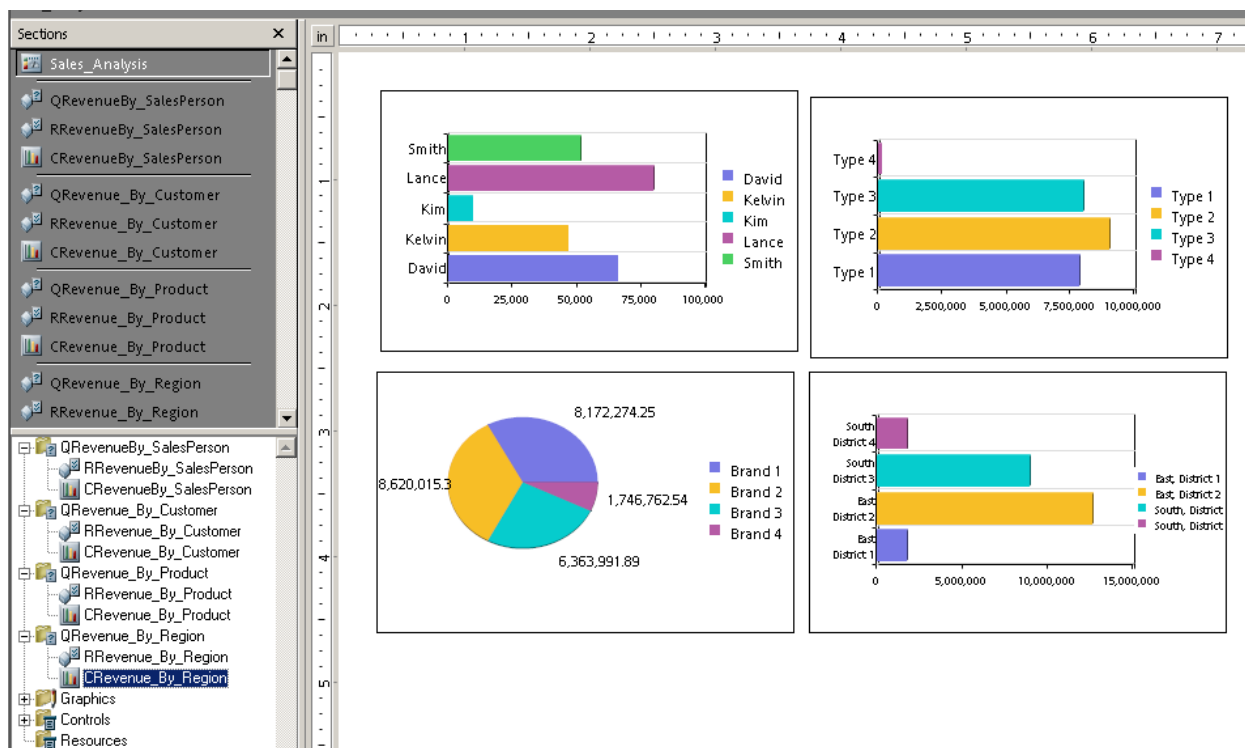


Make sure you select the fact in fact layout and dimension(s) in X or Y Axis. Incorrect selection may leads to hang your system, depends upon the size of fact data. Similarly create chart for all the other 3 reports.

It's time to create a dashboard and incorporate all the Charts into a single dashboard and give users a integrated view to the data.

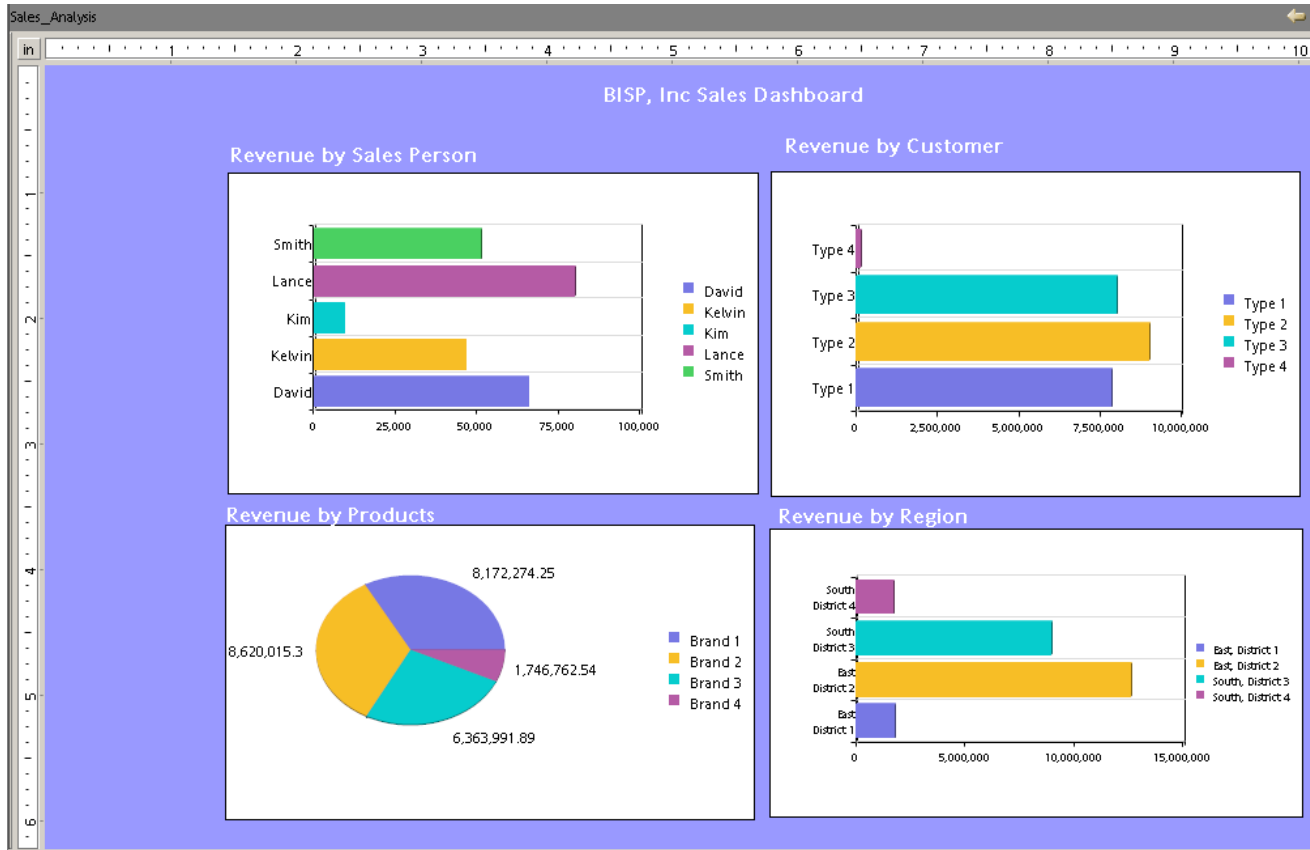
Insert Dashboard.

Insert→New Dashboard



After inserting chart, you can place the chart into dashboard simply by selecting chart and dragging and dropping into dashboard. The way the charts are looking is not good, you suppose to format it and place appropriate heading.

So finally your IR Dashboard look like this way.



In the second of the document we'll talk about incorporating Dashboard Java Script.