



DATA WAREHOUSING CONCEPTS BUSINESS OBJECTS [UNIVERSE & REPORTING]

By
Ganesh
Jegannathan

Data Warehousing Concepts



BY Ganesh Jegannathan, MphasiS an *HP* Company

Before we Begin

Need to understand

- Data
- Database
- DBMS → RDBMS / NDBMS / HDBMS
- RDBMS
- RDBMS – Relationships
- Data Model

Data:

Data can be defined as, any information which helps to run your business. In an HR application data is the employee, his salary information, employee name. In RDBMS , DATA is stored in the form of rows and tables



Database:

Software which is used to maintain all the data an organization has. We can have multiple tables which contains many rows. Databases organize the data in such a way it is very easy to maintain.

RDBMS:

RDBMS stands for Relational Database Management system; it works based on the set theory, entities and its relationship. An entity is a real world thing, when you define a business, you deal with customer, products, sales etc, these are called entities. All the entities have its own properties which we call as attributes / columns in the database. In employee entity (Table) we store all the information about employees (Name, Address, salary, manager, etc).

RDBMS – Relationships

One to One

• One to Many / Many to One

• Many to Many

→

Person and PAN Card

→

Person and Credit Card

→

Person and Doctor

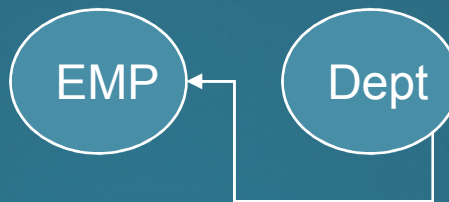


Data Model

It's a pictorial representation of the entities, relationships between entities and the attributes which reflects how the data is stored to run the business. In software development life cycle, after the requirement gathering phase, designing the data model is done. As a part of the project, we develop conceptual model, logical model and physical model.

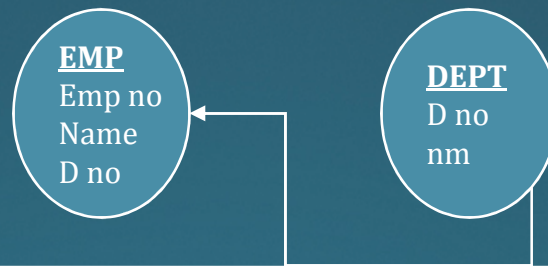
Conceptual Model:

It is a very high level view (bird's view) of how the business runs. We cross verify the model with the business to make sure we are on the right track. In this phase, we identify most of the entities which are required by business. You will see lot of many to many relationships between the identified entities.



Logical Model:

In this model we actually complete the model with all the details of entities including the attributes, domains and its relationships. We resolve all the many to many relationships using associative tables in this phase. You can completely visualize the data movement within the business using this model. Every project must have a logical model even if the project is very small.



Physical Model:

We create the tables in their respective databases. Here DBA will take a call on identifying certain changes to logical model to show better performance while running the application. These DBA s will be good in the respective databases and its utilities so that they can take a call on how best they can deploy the data model. At the logical level we call entities and attributes where as at physical level we call it as tables and columns.

Physically creating tables and joins



Normalization VS De-Normalization

De-Normalized form

Emp no	Ename	D Name
1	TIM	Sales
2	JOHN	Sales

Normalized form

Dept no	D Name
100	Sales

Emp no	Ename	Dept no
1	TIM	100
2	JOHN	100

Normalization helps to remove redundant data resulting in using less storage space which in turn makes our transaction to run faster.....

.....
This way we could have speed up the transaction we do. So, by splitting the tables to smaller, the inserts, updates and deletes run faster where as select statements run slower as we join many tables (this is the disadvantage).

De-Normalization:

→ Only to speed up the Query



Normalization

Its about breaking entities into their most granular form

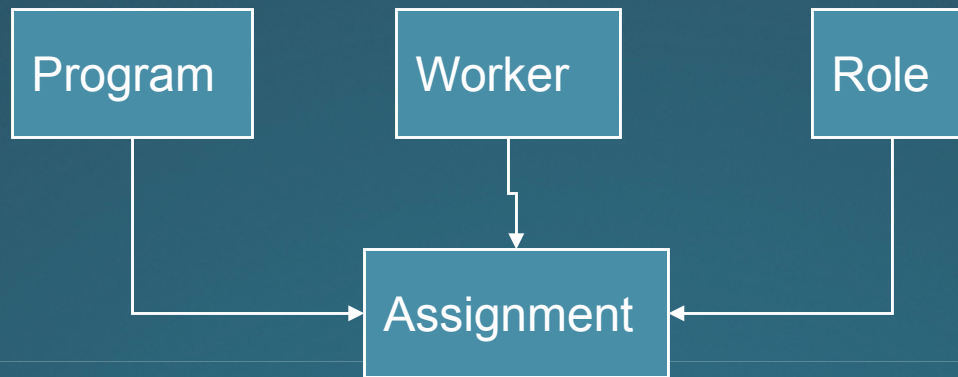
- 1st Normal Form (1NF)
 - Every attribute must be atomic
 - Repeating attributes moved to a separate entity

- 2nd Normal form (2NF)
 - No attributes in the child entity are really facts about the parent
 - Its usually applies to dependant entities
 - All the attributes should be functionally / partially dependent on the primary key

- 3rd Normal form (3NF)
 - All the attributes should be primarily dependent on the primary key
 - If attributes and key of another entity appear as attributes of an entity, move them back to other entity or to a new entity (often a type entity)



Example



This is OK if.....
Any combination is
valid

BUT.....

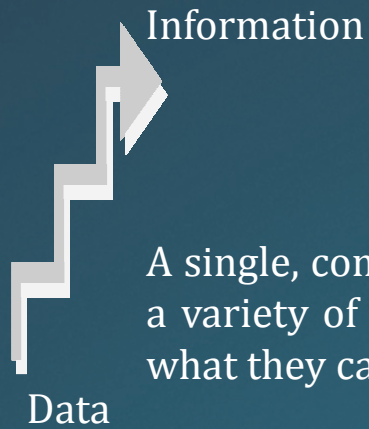
Workers have defined role

Programs require certain roles

Workers are assigned programs independent of roles

Then we will end up writing lots and lots of coding to make sure to implement these business rules. If we normalize it some of the rules are automatically taken care of





Data Warehouse

A single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context.

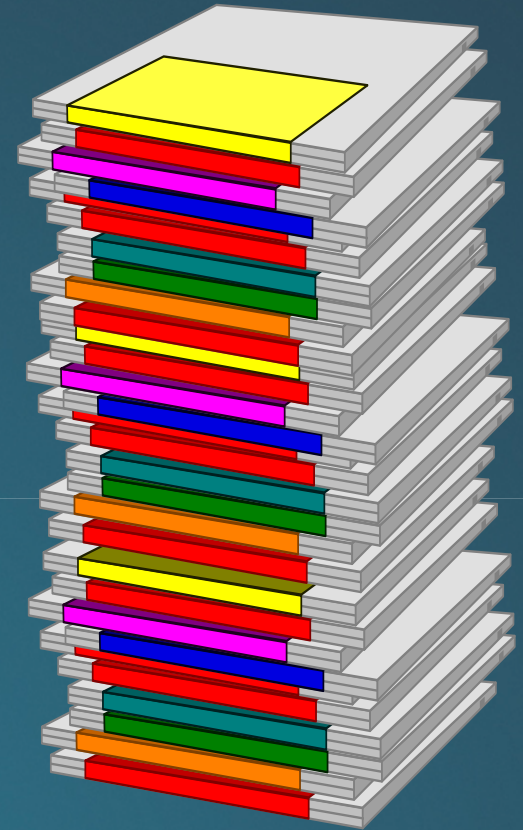
[Barry Devlin]

Subject Oriented, Integrated, Time varying, Non Volatile collection of data that is used primarily in organizational decision making

[Bill Inmon]

A process of transforming data into information & making available to users in timely enough manner to make difference

[Forrester Research, April 1996]



Subject Oriented

- Data is arranged by the common subject area of a company, such as sales, marketing, finance.

Integrated

- Data warehouse integrates data derived from various functional systems in the organizations and provides a unified and consistent view of the overall organizational situation

Time Variant

- Warehouse data represent the flow of data through time, and queries are based on time range.

Non-Volatile

- Data are always added to the warehouse for maintaining the company's entire history.



Why Data Warehousing?

Need of Intelligent Information in Competitive Market

Who are the potential Customers ?
Which Products are sold the most ?

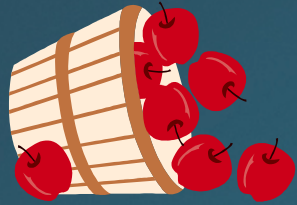
What are the region-wise preferences ?
What are the competitor products ?

What will be the impact on revenue ?
Results of promotion schemes introduced ?

What are the projected sales ?
What if you sale more quantity of a particular product ?



OLTP vs. Data Warehouse

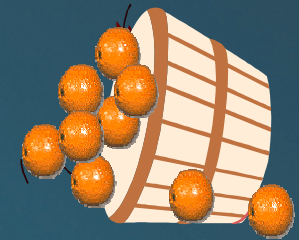


OLTP

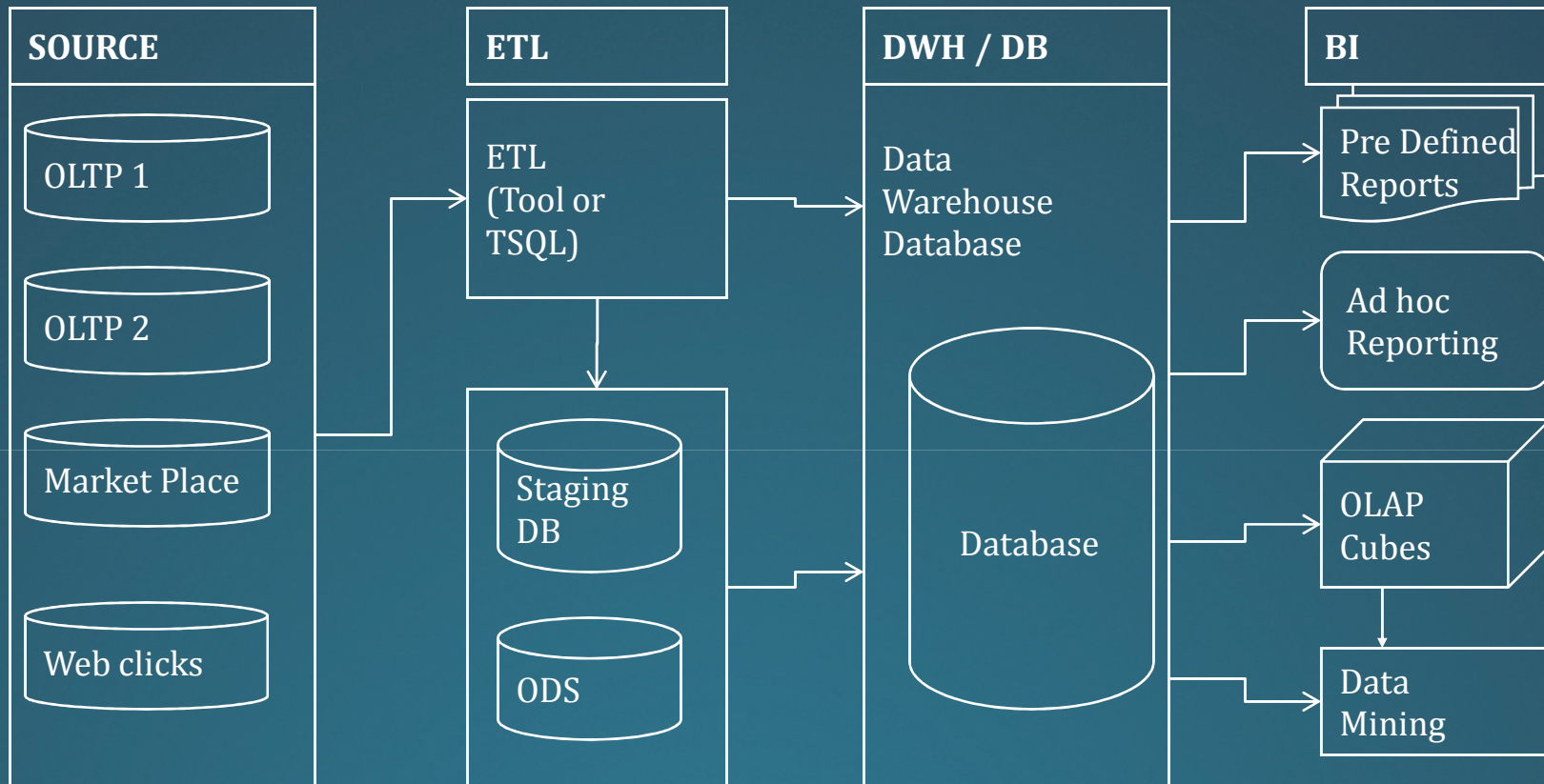
- Application Oriented
- Used to run business
- Detailed data
- Current up to date
- Isolated Data
- Repetitive access
- Clerical User
- Performance Sensitive
- Few Records accessed at a time (tens)
- Read/Update Access
- No data redundancy , De-Normalized
- Database Size 100MB -100 GB
- Transaction throughput is the performance metric
- Thousands of users
- Managed in entirety

WAREHOUSE (DSS)

- Subject Oriented
- Used to analyze business
- Summarized and refined
- Snapshot data
- Integrated Data
- Ad-hoc access
- Knowledge User (Manager)
- Performance relaxed
- Large volumes accessed at a time(millions)
- Mostly Read (Batch Update)
- Redundancy present , Normalized
- Database Size 100 GB - few terabytes
- Query throughput is the performance metric
- Hundreds of users
- Managed by subsets



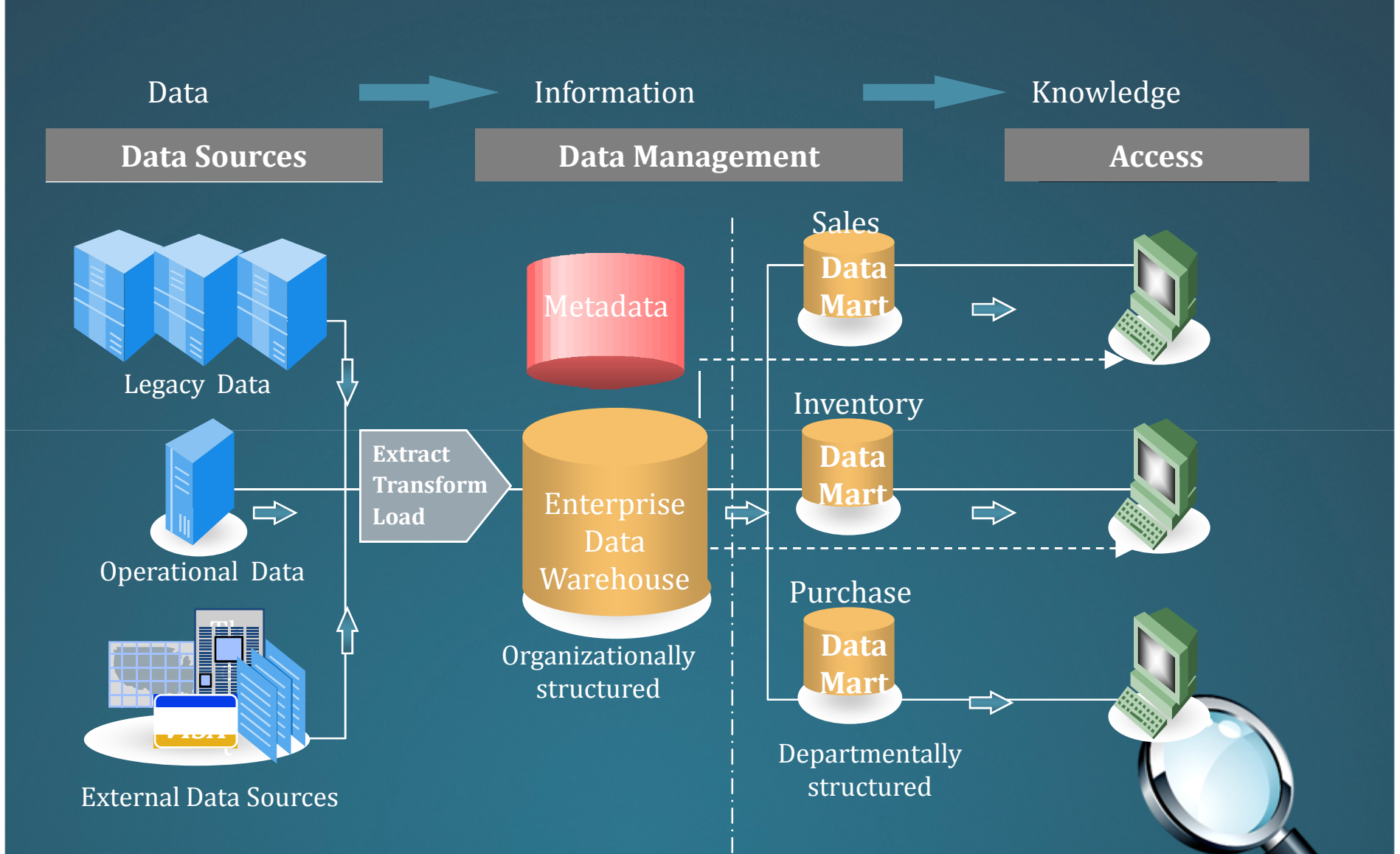
General DSS Architecture



Close the loop (write back to OLTP about the findings in DSS)

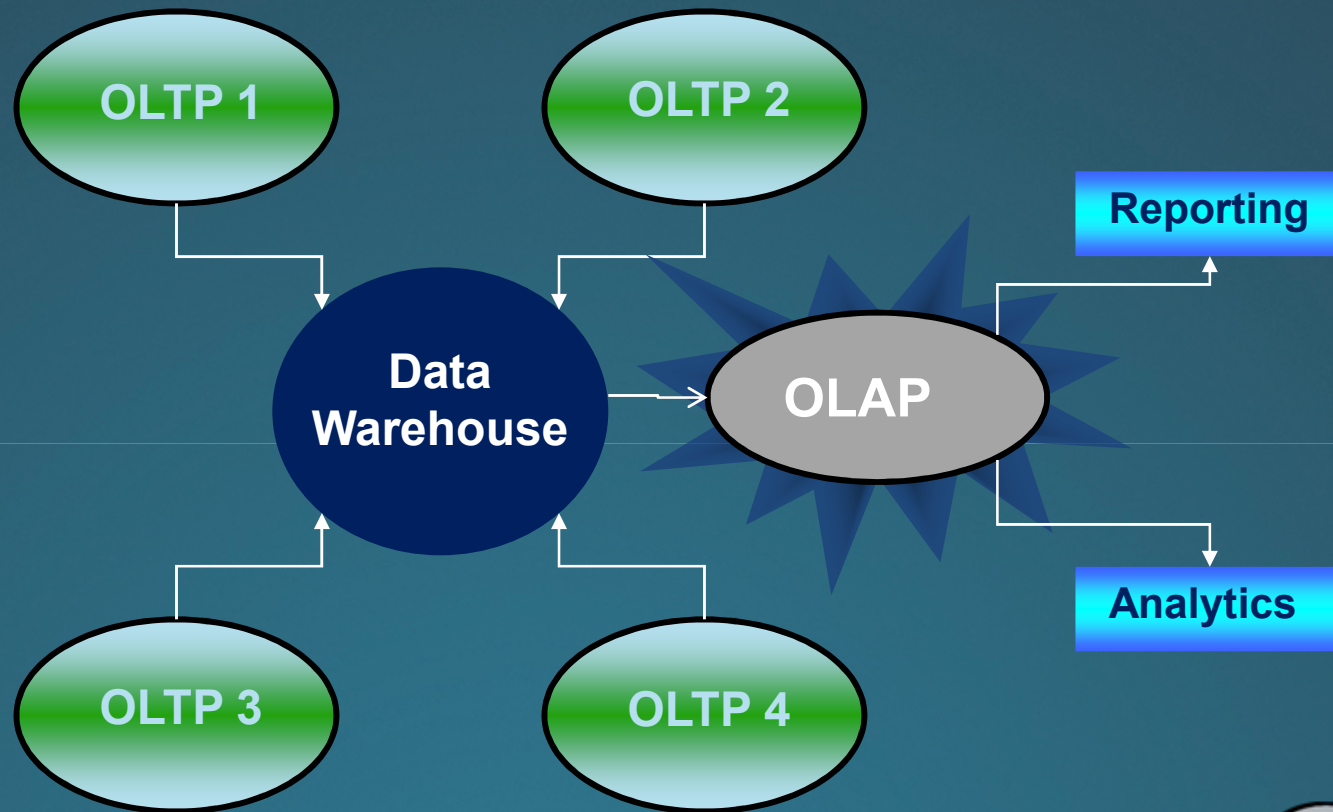


Complete Warehouse Solution Architecture



BY Ganesh Jegannathan, MphasiS an HP Company

Example for a DSS



To summarize ...

OLTP Systems are
used to “run” a business

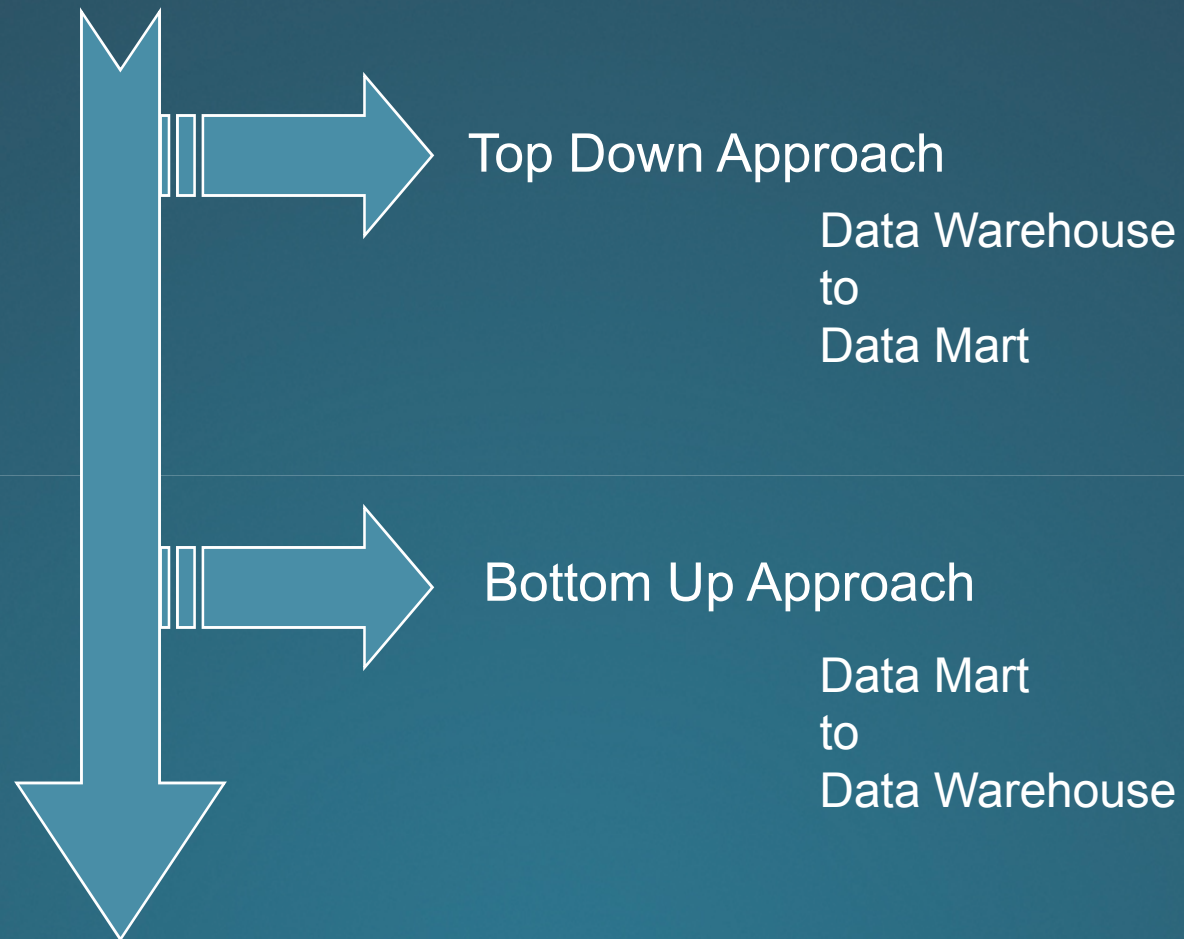


The Data Warehouse helps to
“optimize” the business



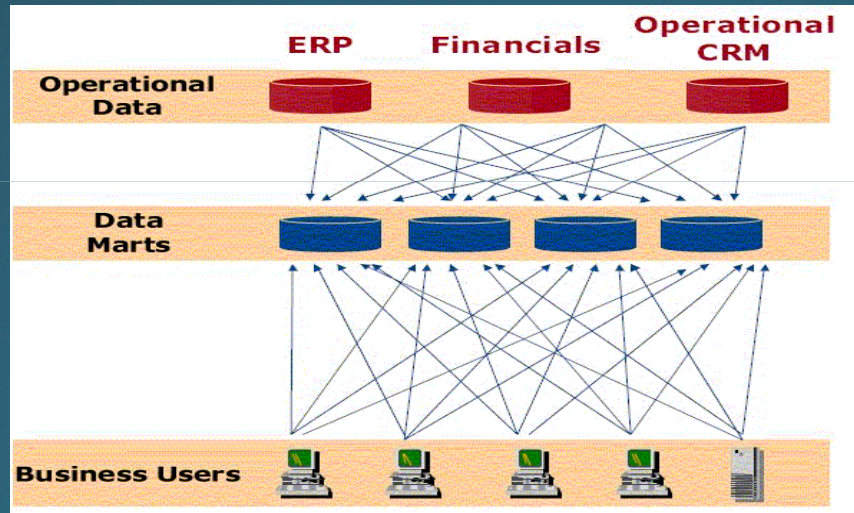
BY Ganesh Jegannathan, MphasiS an HP Company

Approaches to build Data Warehouse



Distributed Approach

- Various departments can start creating different data marts. Each can start working independently and see the ROI in a short span.
- In the long run integrating these data adds the complexity and Cost will be higher as there are more systems to maintain.



Gives only part of the answer

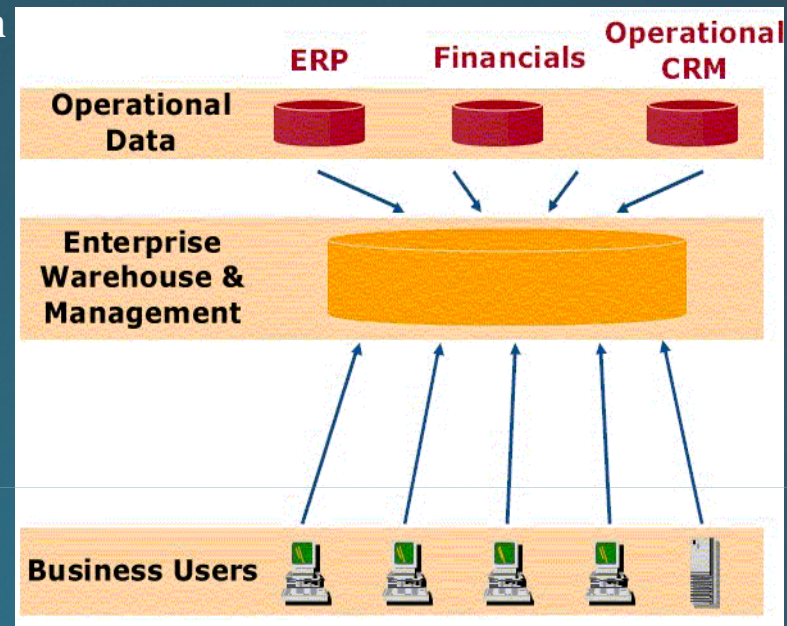
Requires time and effort to put the pieces together

No guarantee it's the right answer



Centralized Approach...

Centralized data warehouse contains the data in one place, easy to answer Any business question. In the long run this has the cost advantage over the Non centralized data warehouse. Not very easy to implement as it needs more time and resources. ROI won't be seen until the implementation is completed. So recommended approach is to implement the centralized data warehouse is, start with one subject area and keep adding one subject area at a time, this way organization will get the see the ROI at various stages



Delivers **one version of the truth** for increased confidence and speed in decision-making

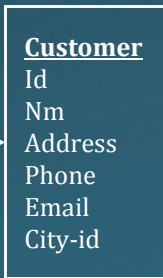
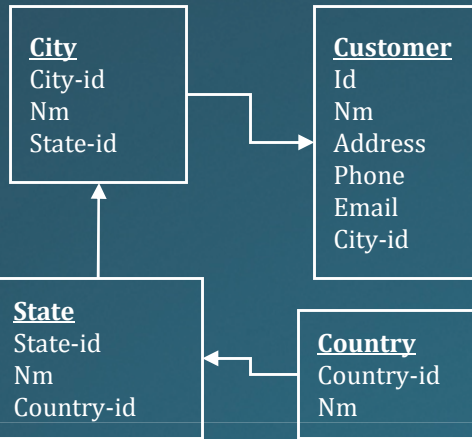


Data Marts

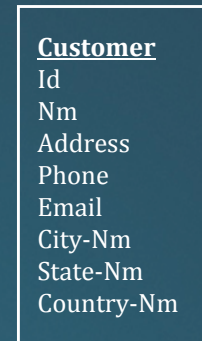
- From the Data Warehouse , atomic data flows to various departments for their customized needs. If this data is periodically extracted from data warehouse and loaded into a local database, it becomes a data mart.
- The data in Data Mart has a different level of granularity than that of Data Warehouse.
- Since the data in Data Marts is highly customized and lightly summarized , the departments can do whatever they want without worrying about resource utilization.
- Also the departments can use the analytical software they find convenient.
- The cost of processing becomes very low.



Data Modeling VS Dimensional Modeling

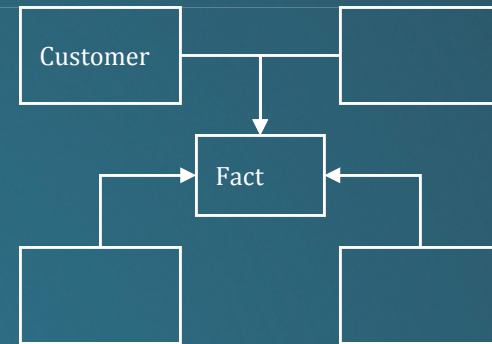
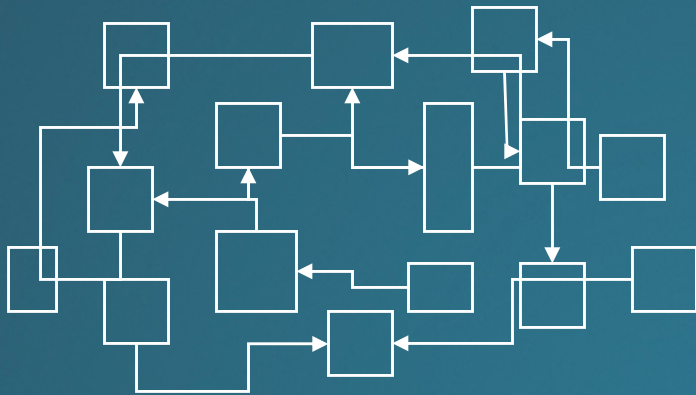


Dimensional Modeling is to implement DWH and Data Modeling/ER Modeling is to live with OLTP system



Entity & Transaction Entity

Dimensions & Facts



Data Warehousing SCHEMAS & OBJECTS

- A schema is a collection of database objects, including tables, views, indexes, and synonyms.
- There is a variety of ways of arranging schema objects in the schema models designed for data warehousing. The are:
 - Star Schema
 - Snowflake Schema
 - Galaxy Schema

Star Schema:

- It Consists of a fact table connected to a set of dimensional tables
- Data in Dimension tables is De-Normalized

Snowflake Schema:

- It is refinement of star schema where some dimensional hierarchy is normalized in to a set of dimensional tables

Galaxy Schema:

- Multiple fact tables share dimension tables viewed as a collection of stars, therefore called galaxy schema



Dimensional Modeling

Star Schema

VS

Snow Flack Schema

Complete De-Normalized form

Normalized Star Schema

Sales Analysis Example

Actual sales goes into fact and customer details, date, sales etc go to dimensions

Now you can visualize a snowflake does not have redundant data in the dimension tables

The visual of the model looks like star, so we call it as star schema.

We will end up having lot of look up tables for the dimension table

In case of star schema all the information about customer are stored in the same table

As the no of tables become more the query performance goes down

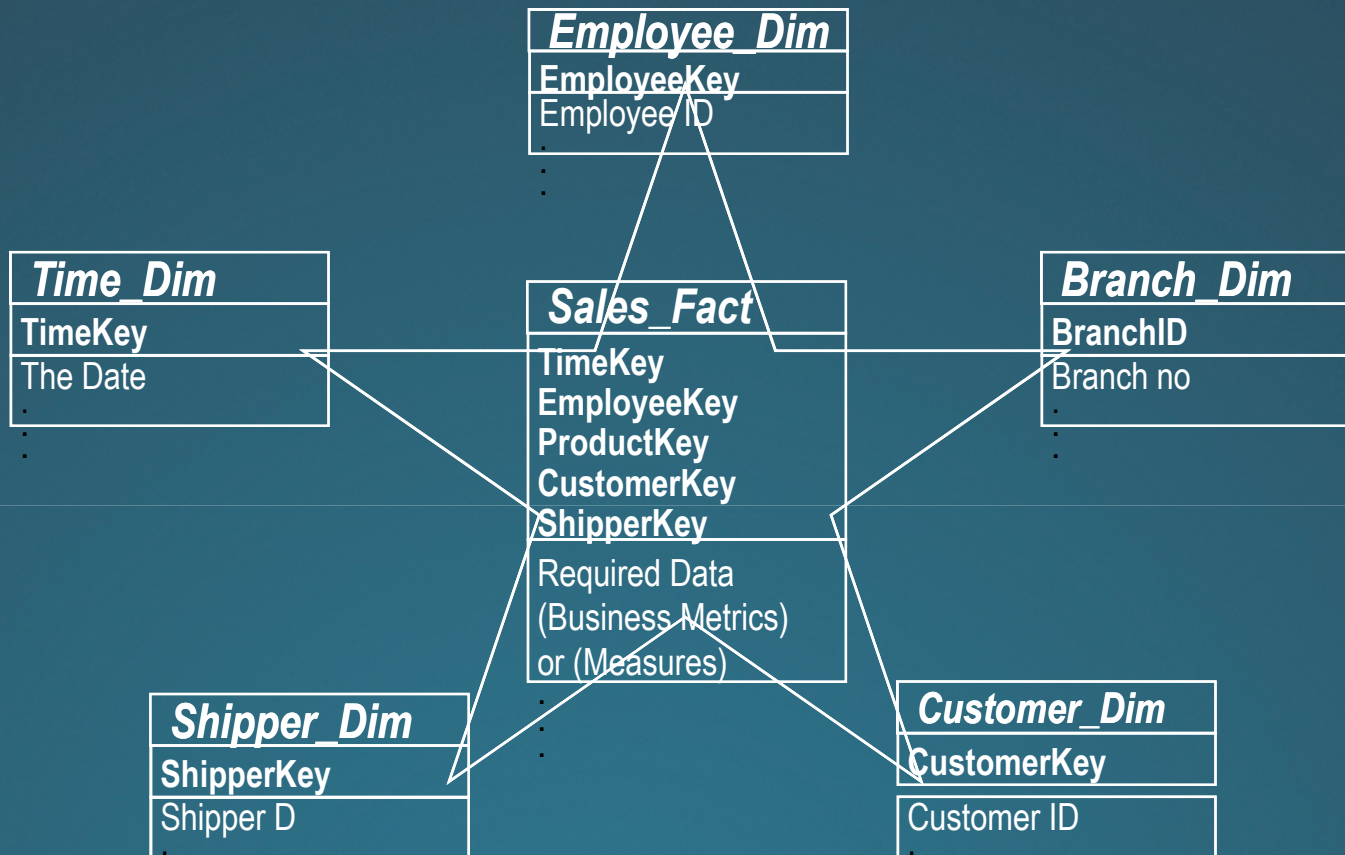
So usually the dimensions in star schema are wider in nature meaning lot of columns

So you have to take a decision on whether to go for star or snow flake depending on data volume and the user performance requirements etc

So your dimension will have lot of redundant information.



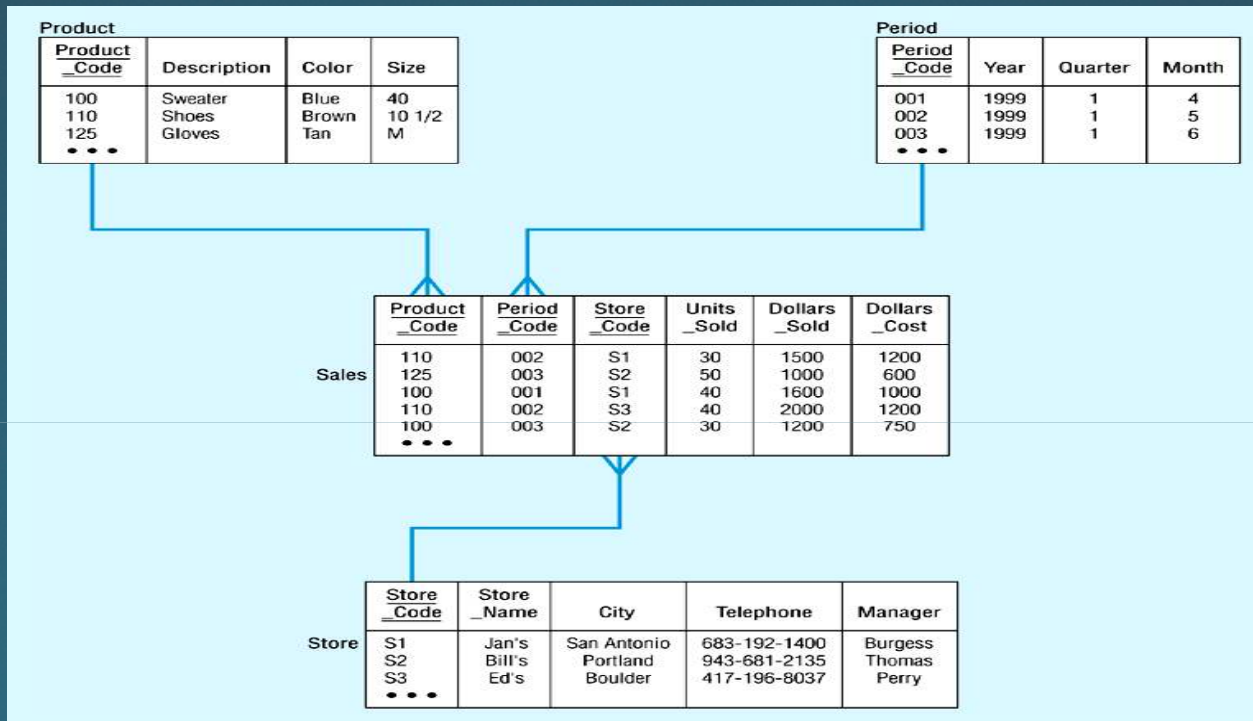
Star Schema



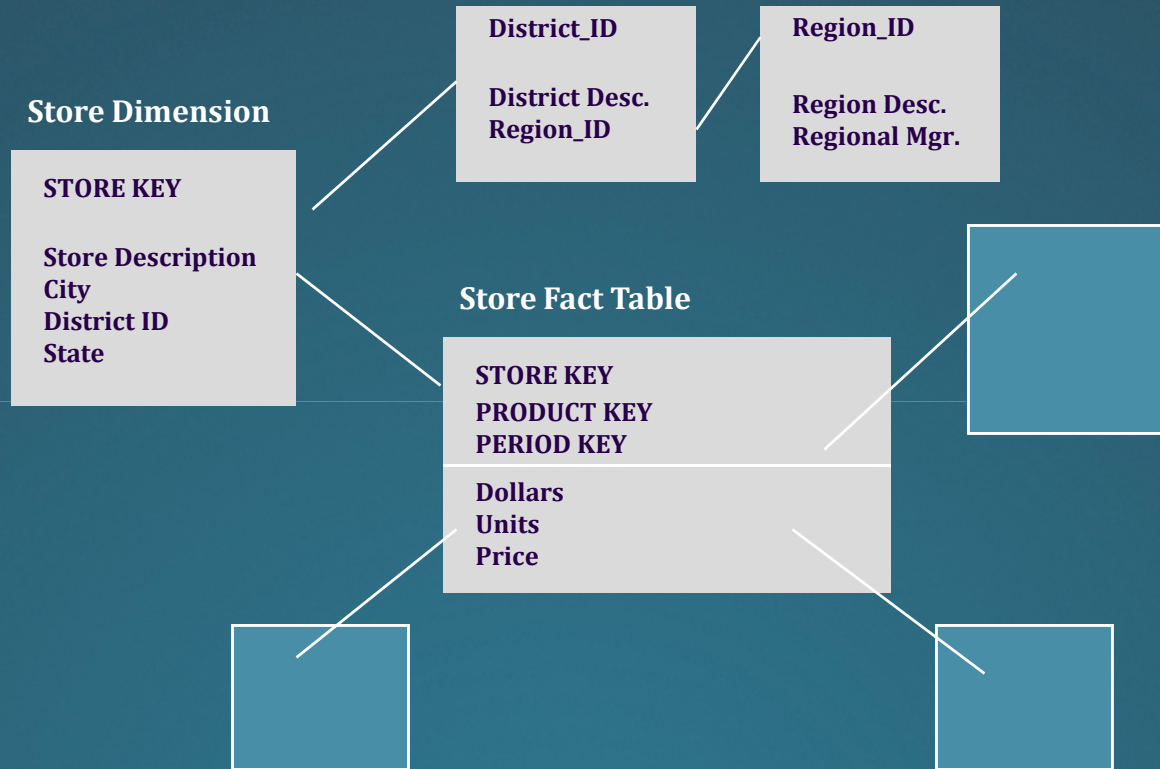
A star schema is a highly De-Normalized, query-centric model where information is broken into two groups: facts and dimensions.



Star Schema with Sample Data - Example

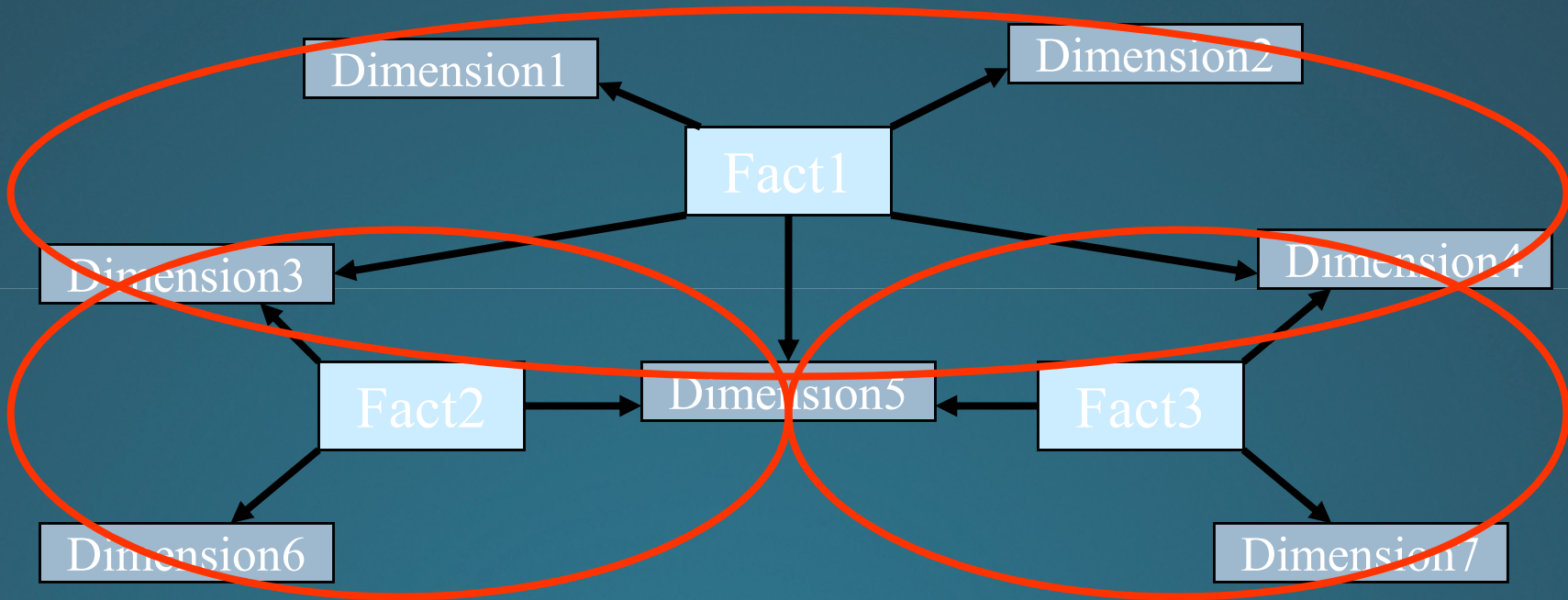


Snowflake Schema



Galaxy Schema

Multiple Groups of Facts links by few common dimensions



Data Warehousing Objects

All the three types of Schemas are described in the Data Modeling section

Various Objects used in Data Warehousing are:

- **Fact Tables**
- **Dimension Tables**
- **Hierarchies**
- **Unique Identifiers**
- **Relationships**



Fact Types

Additive facts:

- Additive facts are facts that can be summed up through all of the dimensions in the fact table

Semi-Additive facts:

- Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table

Non-additive facts:

- Non-additive facts are facts that cannot be summed up for any of the dimensions Present in the fact table



Types of Fact tables :

Based on the classification there are two types of fact tables

- **Cumulative**
- **Snapshot**

Cumulative

This type of fact table describes what has happened over a period of time
For example this fact table may describe the total sales by product by store by day
The facts for this type of fact tables are mostly additive. The first example is a
Cumulative fact table.

Snapshot

This type of fact table describes the state of things in a particular instance
Of time, and usually includes more semi additive and non-additive facts.
The second example presented is a snapshot fact table



Data Warehousing Objects Contd....

Dimension Tables:

- **Dimension tables**
 - Define business in terms already familiar to users
 - Wide rows with lots of descriptive text
 - Small tables (about a million rows)
 - Joined to fact table by a foreign key
 - heavily indexed
 - typical dimensions
time periods, geographic region (markets, cities),
products, customers, salesperson, etc.



Dimension tables Types

- **Slowly Changing dimensions**
- **Junk Dimensions**
- **Confirmed Dimensions**
- **Degenerated Dimensions**



Dimensions and SCD

Histories are maintained over time period

There are three different ways we capture the history

Slowly Changing Dimension 1 (SCD1 / TYPE1)

Slowly Changing Dimension 2 (SCD2 / TYPE2)

Slowly Changing Dimension 3 (SCD3 / TYPE3)

TYPE1 / SCD1:

We use this type then we are not storing any history. Only the current information is stored in this type of dimension. So if your business says, we don't need history, then we can opt for TYPE1 / SCD1.

OLTP Table

Source

DW Dimension Table

Target

Empno	Ename	Job	Sal	Loc
1	Tim	Mgr	20000	Blr

Empno	Ename	Job	Sal	Loc
1	Tim	Mgr	20000	Blr



TYPE2 / SCD2:

We use this method to store the complete history.

We use a id called surrogate ID in TYPE II because the actual id can exists more than once

we can do better analysis like while this was selling at 'X'\$ we use to sell 100 qty where as when are selling at 'Y'\$ we are selling 150 qty etc. If we don't store the history you cannot get this kind of information from

Source

Empno	Ename	Job	Sal	Loc
1	Tim	Mgr	20000	Blr

Target

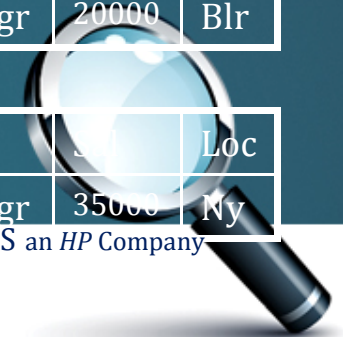
Sid	Empno	Ename	Job	Sal	Loc
1	1	Tim	Mgr	20000	Blr
2	1	Tim	D-Mgr	20000	Blr
3	1	Tim	D-Mgr	35000	Ny

Type 1 Target

Empno	Ename	Job	Sal	Loc
1	Tim	Mgr	20000	Blr

Empno	Ename	Job	Sal	Loc
1	Tim	D-Mgr	20000	Blr

Empno	Ename	Job	Sal	Loc
1	Tim	D-Mgr	35000	Ny



TYPE III /

SCD3:

In this method we store current and previous values of the dimension data

Source

Empno	Ename	Job	Sal	Loc
1	Tim	Mgr	20000	Blr

Target

1st Run

Empno	Ename	Job	Sal	Loc	Pre_Job	Pre_Sal	Pre_Loc
1	Tim	Mgr	20000	Blr	Null	Null	Null

On Updating the source from Mgr to D-Mgr

1	Tim	D-Mgr	20000	Blr	Mgr	Null	Null
---	-----	-------	-------	-----	-----	------	------



SCD 1-2-3 Pros and Cons

Advantages:

This is the easiest way to handle the Slowly Changing dimension, Since there is no need to keep track of the old information.

Disadvantages:

All the history is lost. By applying this methodology, it is not possible to track back in history.

Eg, In above case, company would not be able to know Christina lived in Illinois before.

Advantages:

This allows us to accurately keep all historical information

Disadvantages:

This will cause the size of the table to grow fast where the number of rows for the table is very high to start with, storage and performance can become a concern

Advantages:

This does not increase the size of the table, since new information is updated. This allows us to keep some part of history

Disadvantages:

Type 3 will not be able to keep all history where an attribute is changed more than Once. **For eg,** if Christina later moves from to Texas on December 15,2003 the California information is lost



Degenerated Dimension

Degenerated dimension is a dimension which is derived from the fact table and doesn't have its own dimension table.

Degenerate dimensions are often used when a fact table's grain represents transactional level data and one wishes to maintain system specific identifiers such as order numbers, invoice numbers and the like without forcing their inclusion in their own dimension.

Confirmed Dimensions

Dimension which is fixed and reusable. It is also called as fixed dimension. It is a dimension which doesn't effect with respect to time.

Ex : if name of the city is changed from Bombay to Mumbai, name will not change from time to time, once the change is done ,The change is permanent. This type of dimensions are called confirmed or fixed dimensions.

Junk Dimensions

A dimension where one can store random transactional codes, flags and text attributes that are not related to other dimensions and which provides a simple way for users to easily find those unrelated attributes.

➤Ex: Martial Status : (Yes or No)
Gender : (M or F) etc



OLAP

- Online Analytical Processing. The way analyst slice and dice the DW data. Viewing the fact information through multiple dimensions is called slice and dice. Example: if we see the revenue by product and year then we call the report as multi dimensional reporting. In the industry we have three types of OLAP.

ROLAP

MOLAP

HOLAP

Usually when we implement the DW, we create of static reports, some times we call it as canned reports as well. As users become good in understanding the data, to analyze why such slip or peak happened in specific scenario we will open up for the Ad hoc query. Using Adhoc query facility end users can pull in the data what they need in the way they want to see. Any tool, which exposes the DW model through meta data and helps to create the report then we call it as BI tool.



What is OLAP?

- What is OLAP?
 - Online Analytical Processing. Viewing data in a multi dimensional way.
- Why OLAP?
 - “Slice and dice” for data warehouse.
 - RDBMS is a 2 dimensional way of storing / viewing the data
 - OLAP is a multi dimensional way of storing / viewing the data



BUSINESS OBJECTS

- Universe Designer



BY Ganesh Jegannathan, MphasiS an *HP* Company

OLAP

- What is OLAP?
 - Online Analytical Processing. Viewing data in a multi dimensional way.
- Why OLAP?
 - “Slice and dice” for data warehouse.
 - RDBMS is a 2 dimensional way of storing / viewing the data
 - OLAP is a multi dimensional way of storing / viewing the data

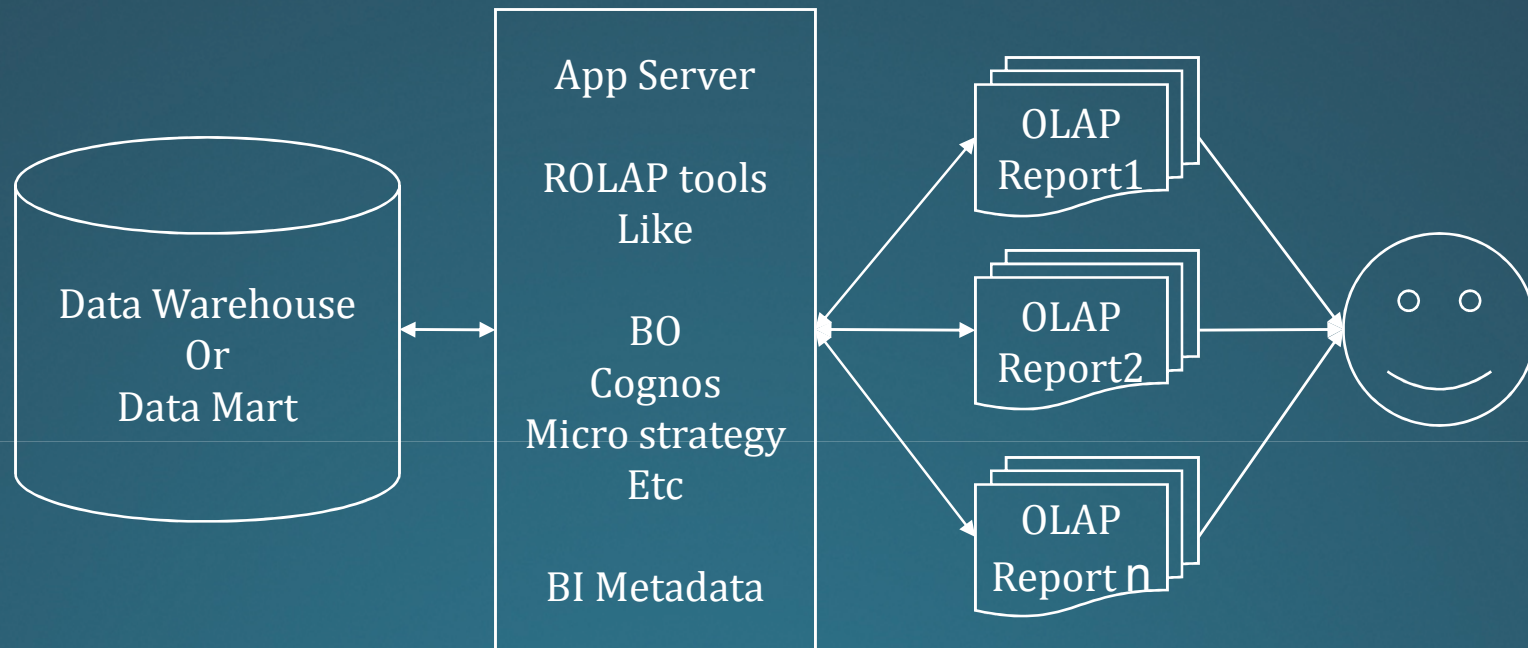
OLAP - Types

Three types of OLAP in the industry.

1. MOLAP – Multi dimensional OLAP (Ex MSOLAP, Essbase, Cognos).
2. ROLAP – Relational OLAP (Ex Business Objects, Micro strategy).
3. HOLAP – Hybrid OLAP



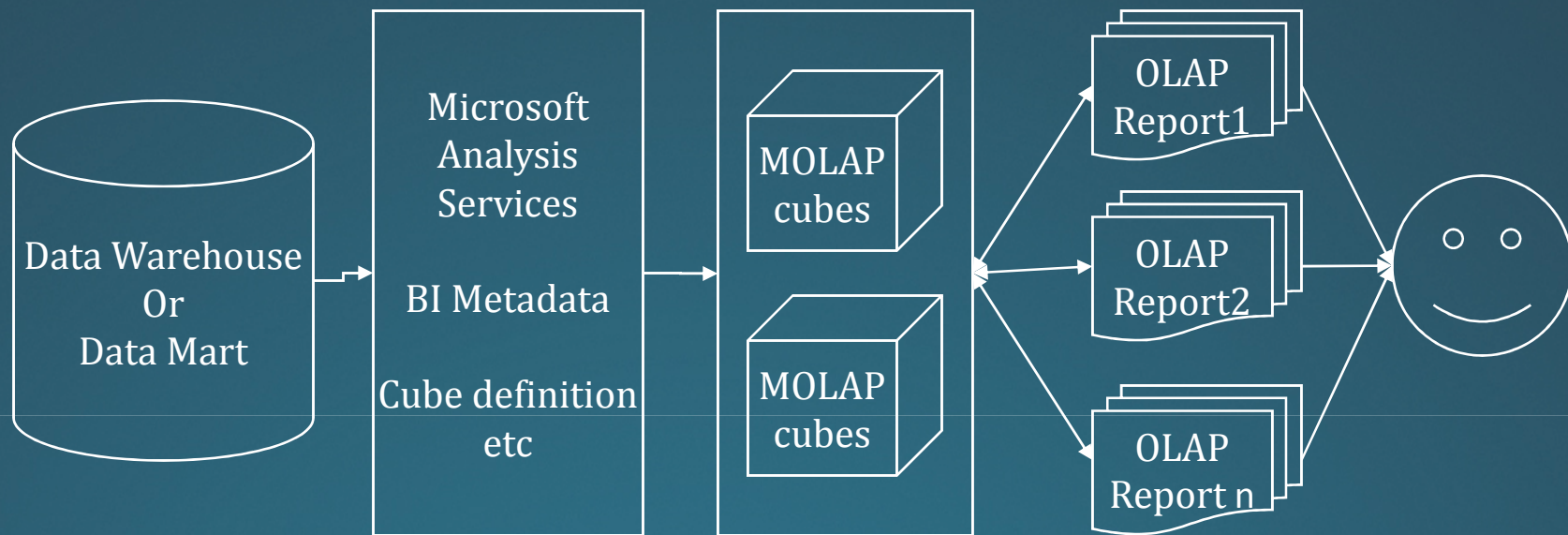
Architecture diagram of ROLAP



When a report is executed by end user the actual SQL is issued to RDBMS to get the data. Some BI tools can even store the results set in the application server and periodically refresh that report based on the data refreshes which happen in DW.



Architecture diagram of MOLAP



When a report is executed by end user the actual data is retrieved from the MOLAP cubes. The way it retrieves by using MDX queries based on the report. MDX stands for Multidimensional expression. SQL is used to get the data RDBMS, MDX is used to get the data from MOLAP. The MOLAP cubes are refreshed periodically based on the data refreshes which happen in DW.



BI Tools VS OLTP Reporting tools

- Why not we can use Crystal Reports
- What is Metadata?
- Evaluation of Data Warehousing reporting
 - Batch Reporting
 - Ad hoc
- In order to enable our business users to explore the data in the way they want, we should expose the database in a way business can understand. In BI tools we store that meta data (mapping between the tool and database) information. In BO, we create this meta data using Designer.



Business Intelligence [BI?] - Introduction

- Business Intelligence is a technology based on profit oriented models that reduces operating costs and provide increased profitability by improving productivity, sales, service and helps to make decision making capabilities at no time.
- Business Intelligence Models are based on multi dimensional analysis and key performance indicators (KPI) at an enterprise.
- Key Performance Indicators are quantifiable measurements, agreed to beforehand, that reflect the critical success factors of an organization. They will differ depending on the organization.
 - ex-
 - A business may have as one of its Key Performance Indicators the percentage of its income that comes from return customers.
 - A school may focus its Key Performance Indicators on graduation rates of its students.

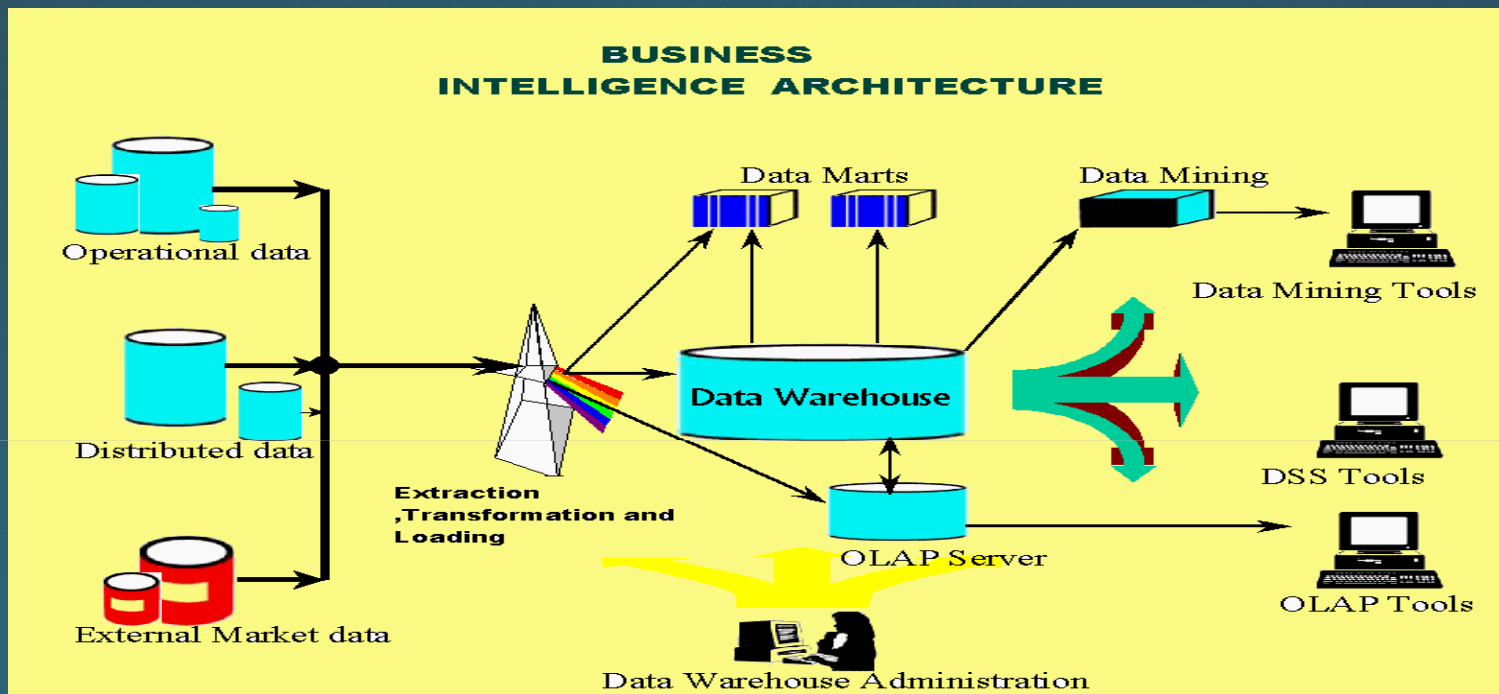


Why BI?

- BI technologies help bring decision-makers the data in a form they can quickly digest and apply to their decision making.
- BI turns data into information for managers and executives and in general, people making decisions in a company.
- Companies want to use technology tactically to make their operations more effective and more efficient - Business intelligence can be the catalyst for that efficiency and effectiveness.



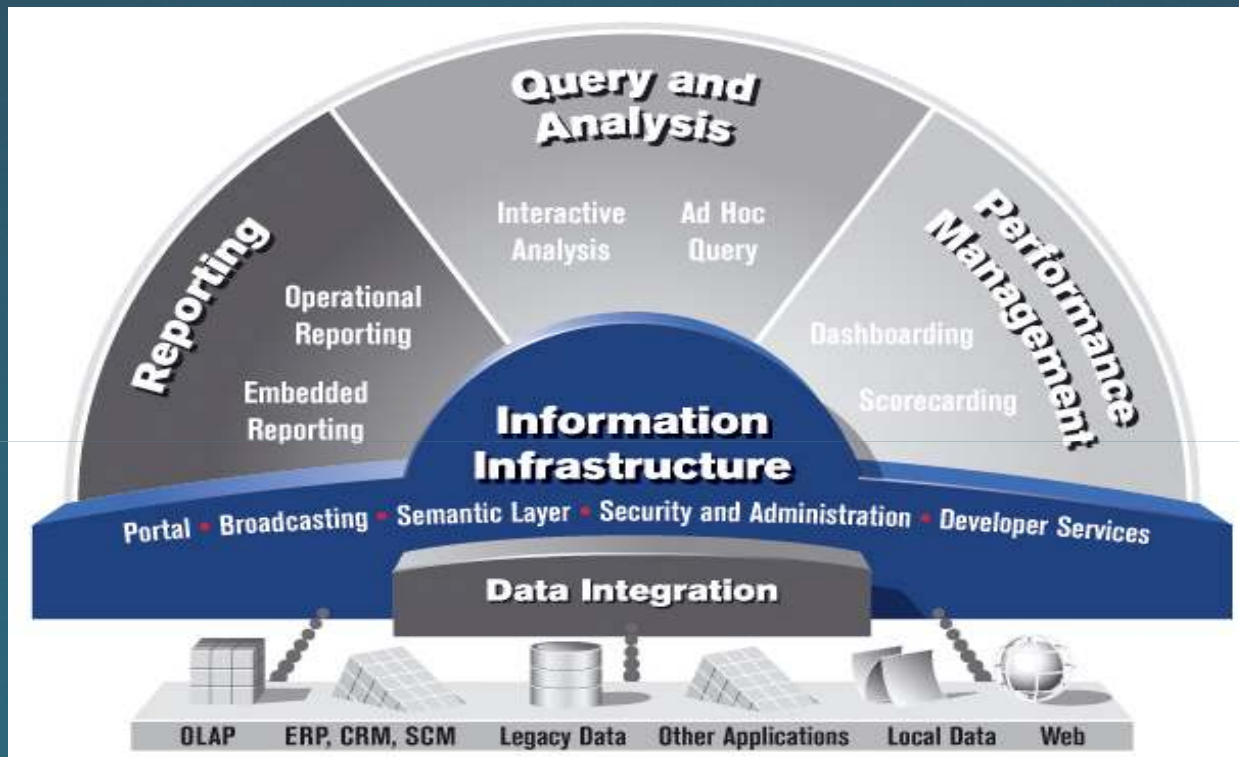
Business Intelligence Architecture



BY Ganesh Jegannathan, Mphasis an HP Company

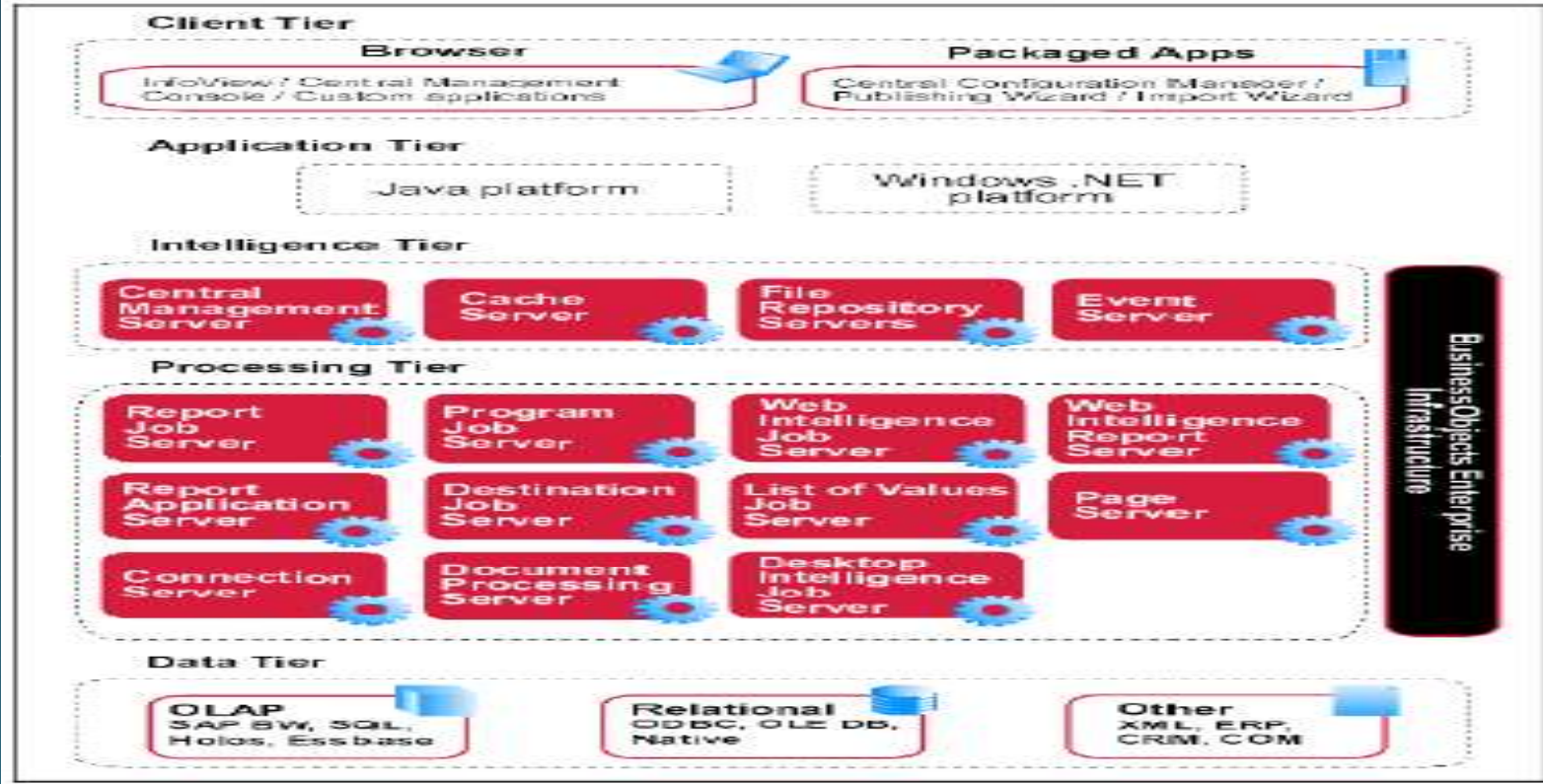


Business Objects Components



BY Ganesh Jegannathan, MphasiS an HP Company

Business Objects Architecture

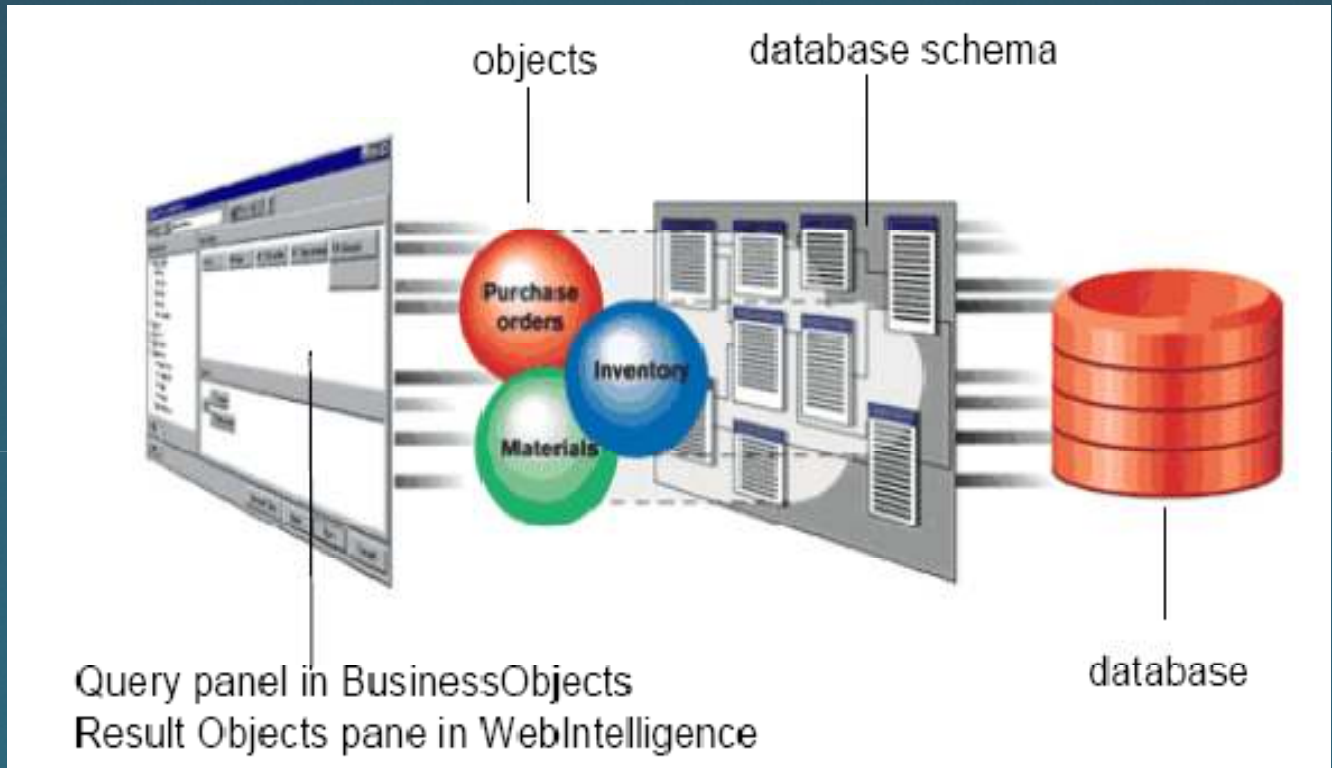


Business Objects

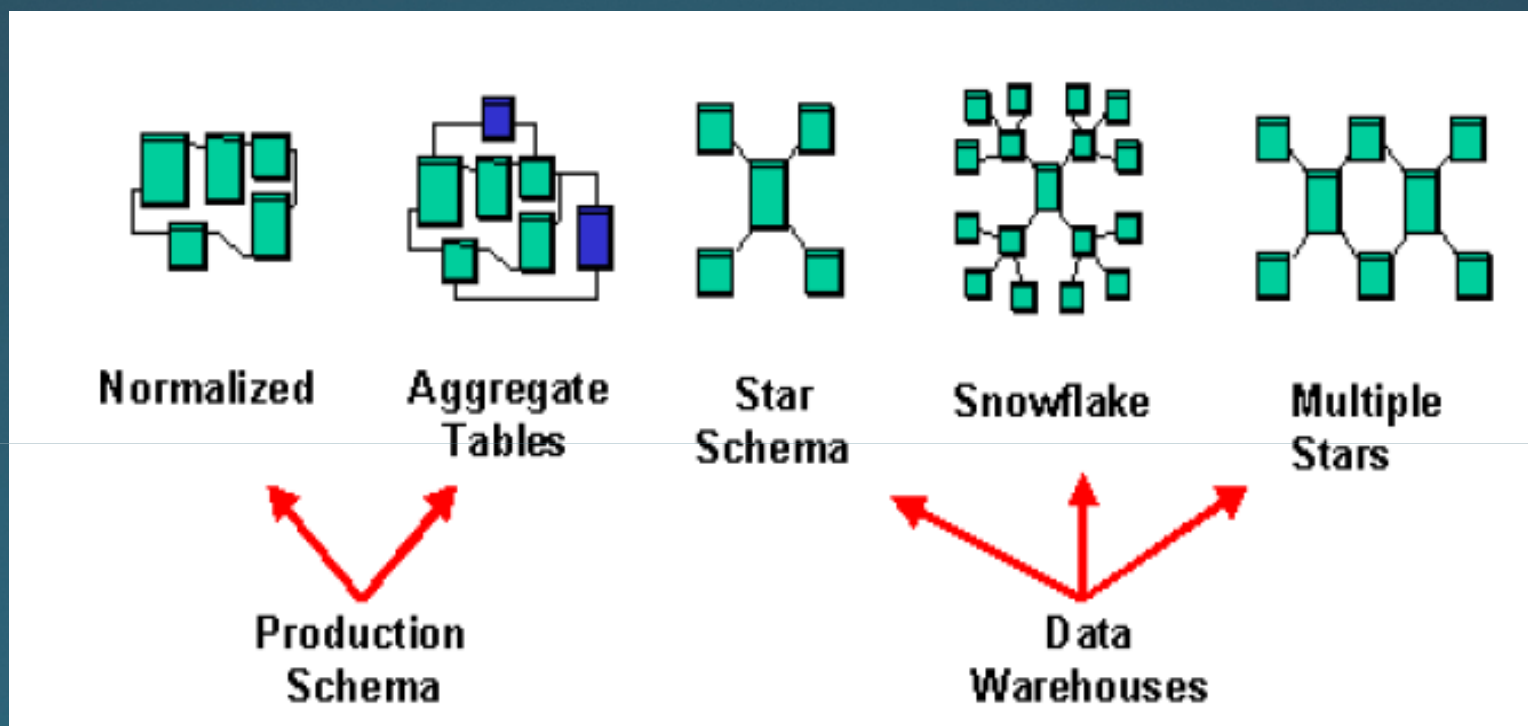
- Business Objects Enterprise is a business intelligence (BI) platform that powers the management and secure deployment of specialized end-user tools for reporting, query and analysis, performance management, and analytic applications on a proven and scalable.
- Frequently Used Components of business object.
 - CMS (Central Management Server)
 - Designer (Universe creation)
 - Desktop Intelligence (Desk-I) (Report Development)
 - Info view
 - Web Intelligence (Web-I) (Report Development)
 - Crystal Reports (CR) (Report Development)
 - Report Import / Export wizard



Database to Reporting using BO



Types of Schemas supported by BO



Designer component of BO

- Designer is the component used by the universe designer to create and maintain the universe.
- Universe is the semantic layer that isolates end users from the technical issues of the database structure.
- Designer is the place where the business object designers will create the universe to interact with databases or any files.
- As designer, you are responsible for creating universes, the metadata layer of BUSINESS OBJECTS. This metadata layer is based on the paradigm of universes, classes, and objects.
- The resource who uses the designer should be good in the data modeling concepts and also should have the domain knowledge which helps the resource to see the data in the business view.



Universe in Business Objects

- The Universe is the metadata for the BUSINESS OBJECT.
- Universe is a business oriented mapping of the data structure found in the database: tables, columns, joins, etc.
- A universe can represent any specific application, system, or group of users. For example, a universe can relate to a department in a company such as marketing or accounting.
- A universe can also pertain to a section within a department or just about any set of organized procedures such as a payroll or inventory system. A universe consists of a set of objects, grouped by classes.



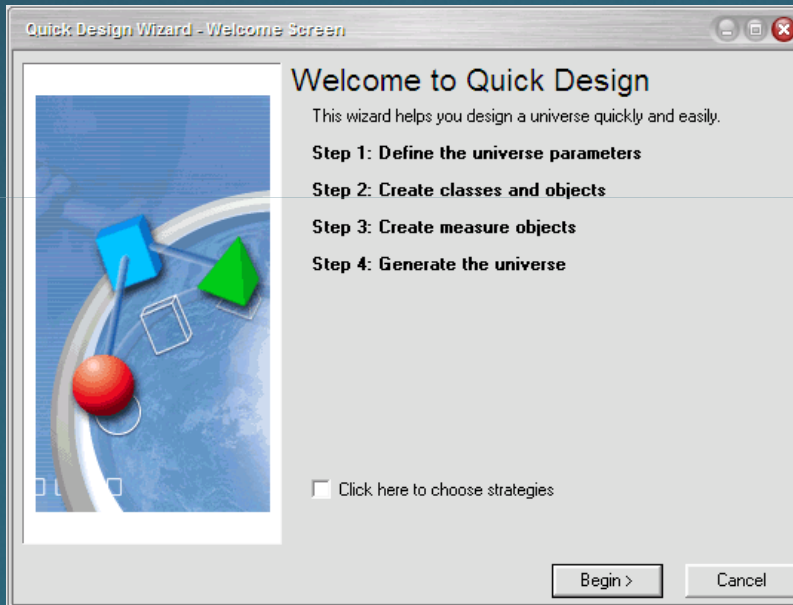
Definitions - Classes and Objects

- Classes
 - A class is a container of objects.
 - A class is the equivalent of a folder in the Windows environment or you can consider a table as the class.
 - You create classes to house objects that have a common purpose in the universe.
 - Classes is the logical grouping objects within a universe.
 - In general, the name of the class reflects a business concepts that conveys the category or type of objects.
 - Ex(Emp, Dept, Sales, Marketing, etc.,)
- Objects
 - Object is the field or column from the table.
 - An object is the most redefined component in a universe. It can be any data or any derivation of the data in the database.
 - The name of the object suggest a concept drawn from the terminology of a business or discipline.
 - Ex (empno, deptno, amount_sold, actual_price, margin, etc.,).



Designing and Working with Universe

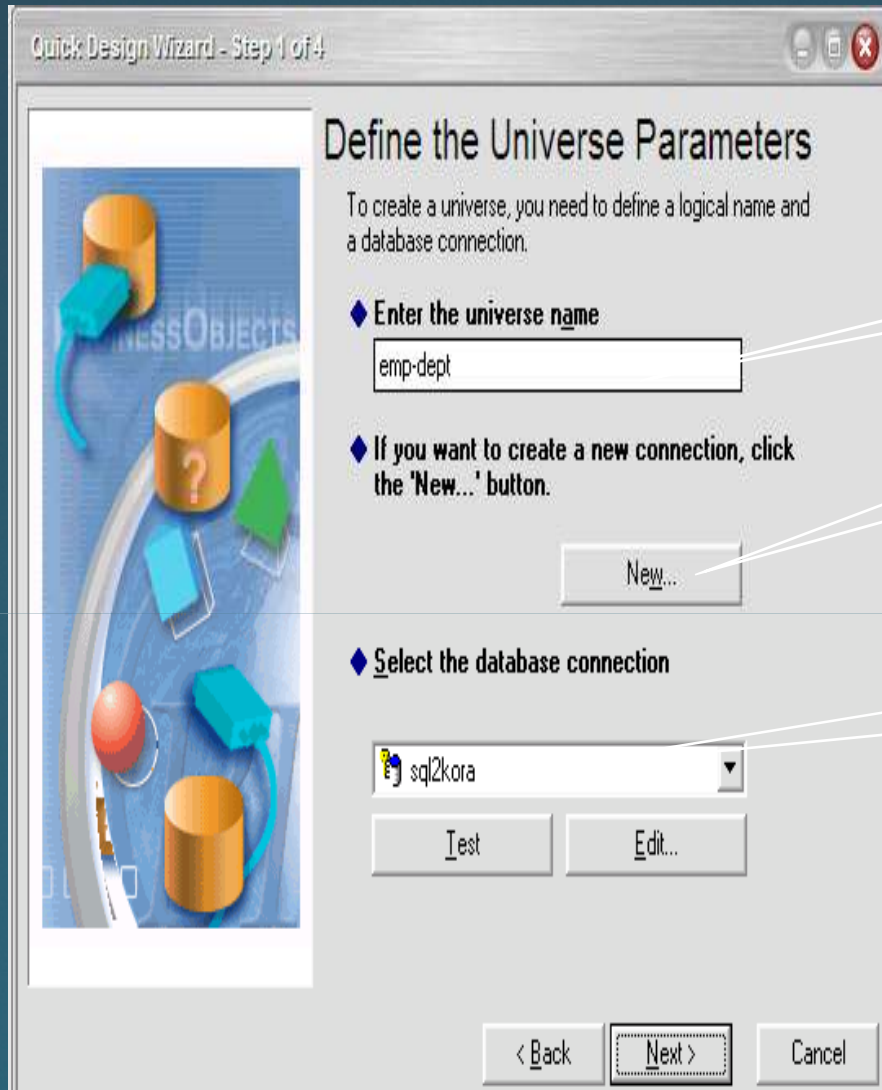
- Creating a universe-
 - Open Business Object Designer.
 - Click Quick design wizard.



Quick design wizard

Click begin to start to create a universe.





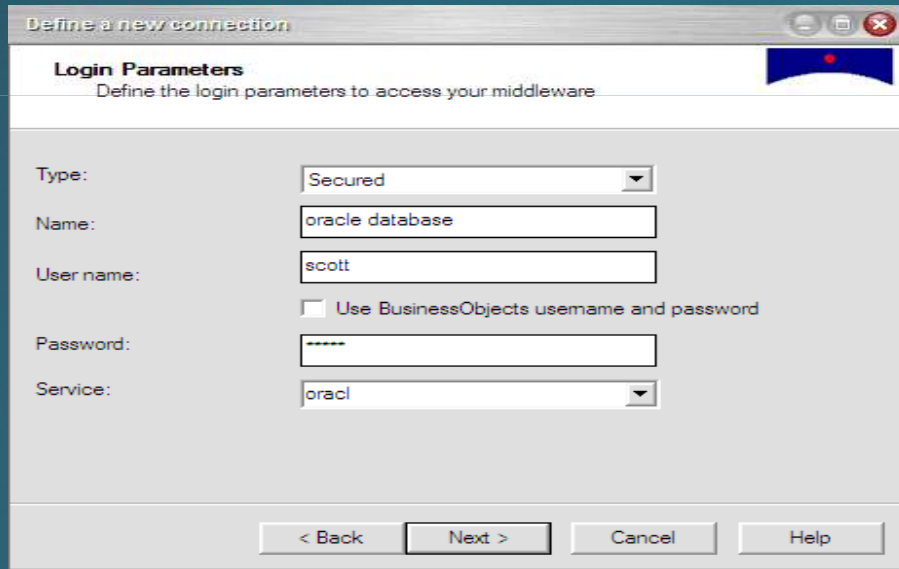
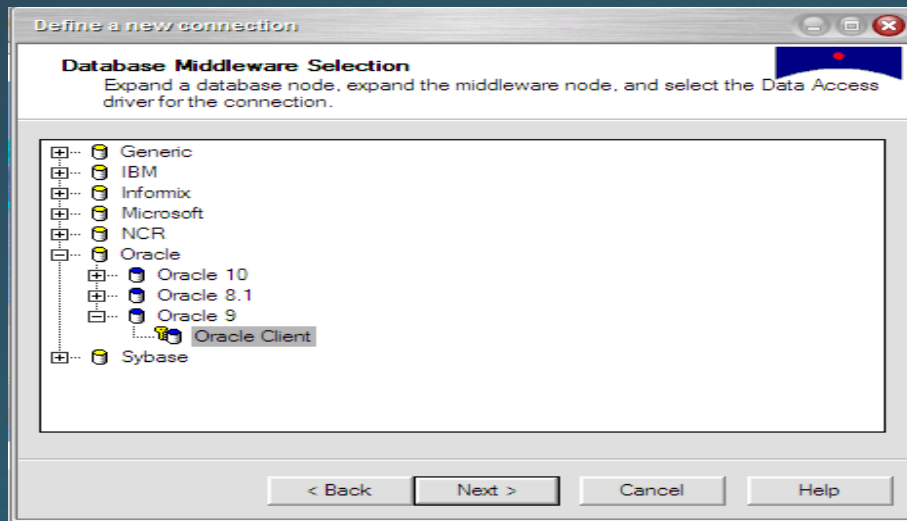
Enter name of the universe.

To create new ODBC Connection

Select the database connection string (ODBC).

Then click next....





If the user don't have any prior connection and would like to create new connection, click new from above the menu

To connect to the oracle database, select the oracle client series from the given options.

If your database is not available in the selection list, choose the Generic and connect through the general ODBC connection.

In connection type we have three modes namely secured, personal and shared.

Once the mode is set, the designer cant edit the mode.



Three Modes of Connections

Shared connections

- Allows access to data for all Business Objects users.
- Shared connections can be useful in a universe testing environment

Personal connections

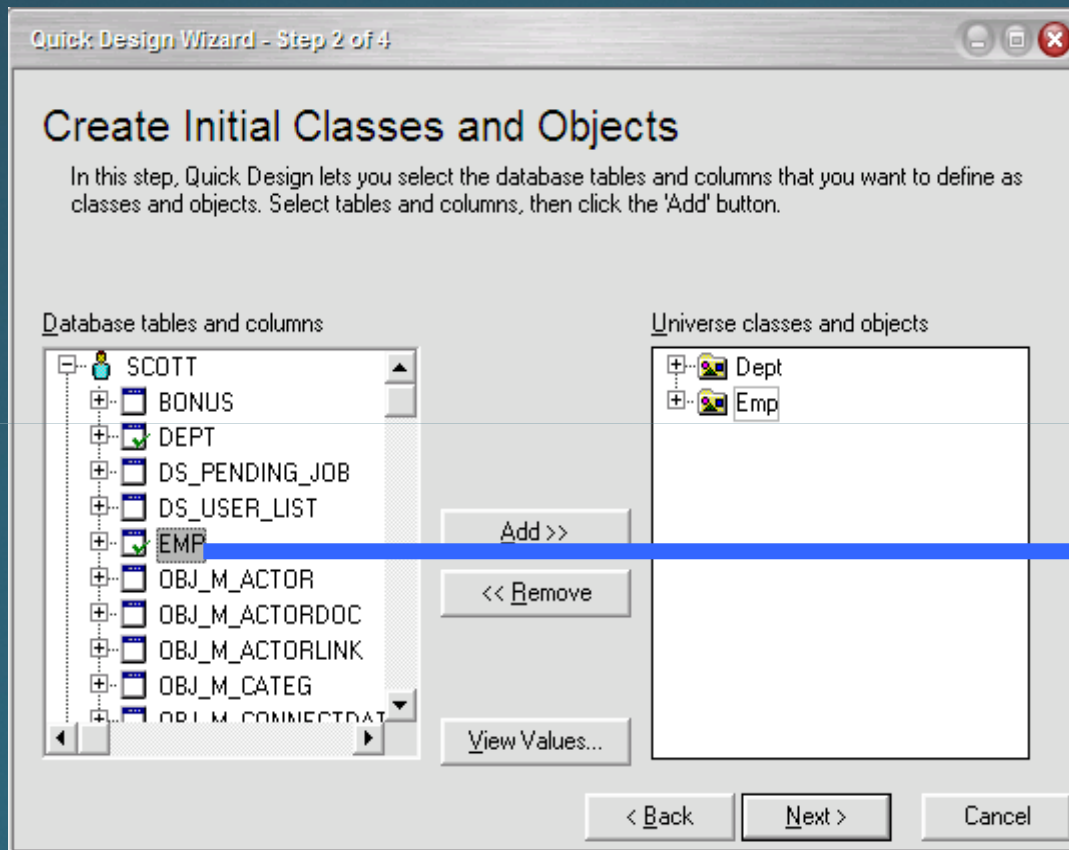
- Restricts access to data to universe creator and the computer on which it was created.
- You do not use personal connections to distribute universes.
- You could use personal connections in the following situations:
 - To access personal data on a local machine
 - To access specific database accounts to test an SQL sample through the Free-hand SQL option in Business Objects

Secured connections

- Centralizes and controls access to data. It is the safest type of connection, and should be used to protect access to sensitive data.
- You can create secured connections with Designer or Supervisor .
- Connections are stored in the security domain of the repository. These can be shared with designers and supervisors with the appropriate privileges.
- You must use secured connections if you want to distribute universes through the Business Objects repository
- If we want to distribute reports over the net using repository etc then the connection should be secured connection



Classes and Objects

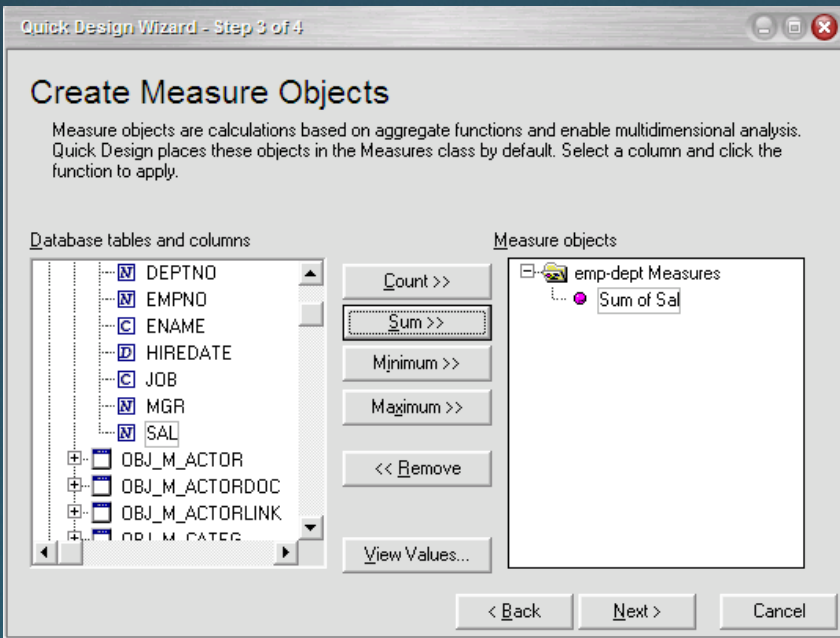


Select the appropriate tables from the database.

Click add to add the tables into universe.

Then click next.....





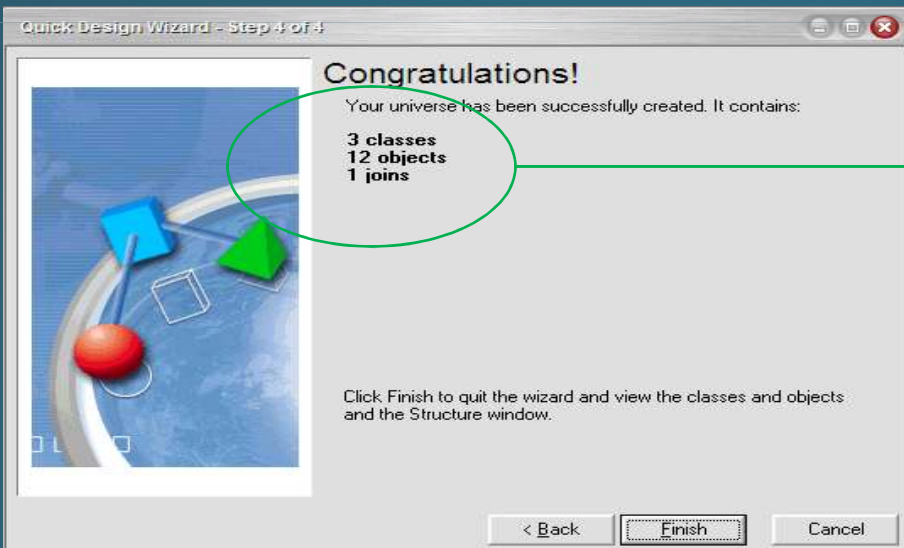
Select the measure element from the tables.
Ex-
select sal from EMP and click sum to get the measure as sum (sal)

3 classes are the scott.Dept, scott.Emp, and try measures.

When we don't specify any measures, universe create its own empty classes.

12 objects includes the fields and measures.

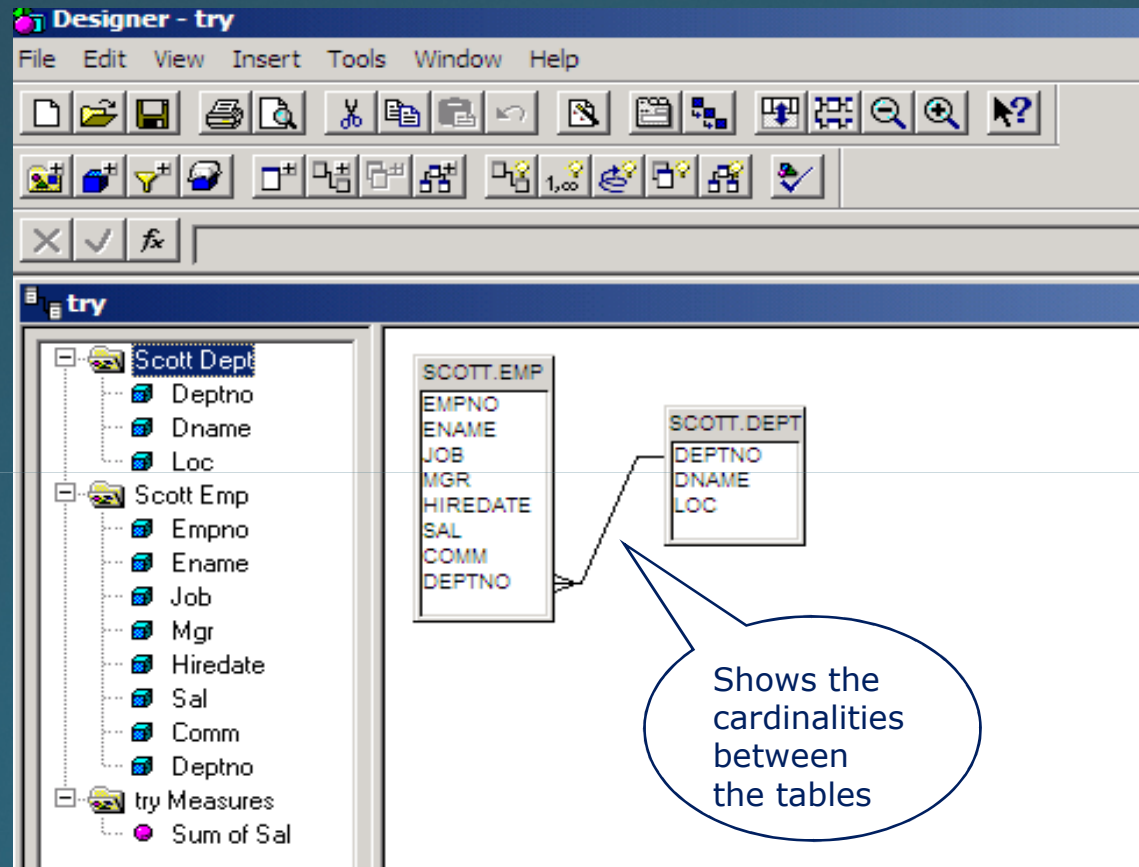
1 join shows the relation between the Emp and Dept tables



Click finish



Designer Window



Here we can find all the selected tables and derived values as classes, objects and measures.

Ex-
Classes- Scott Dept, Scott Emp.

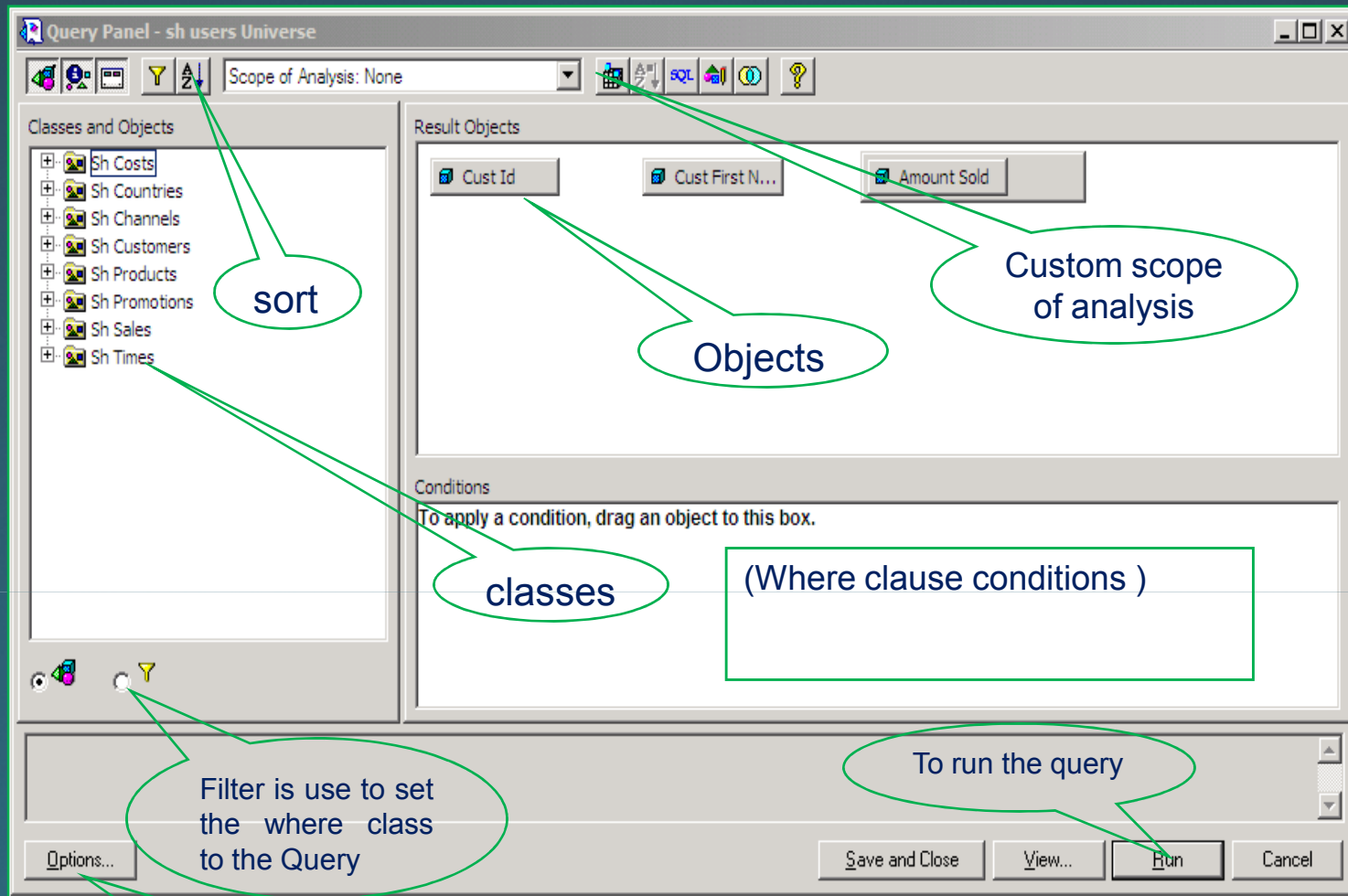
Objects- deptno, dname, loc, empno, ename, etc..

Measures- Sum of Sal.

Note-

In the designer we can also create our own classes and objects.





To restrict returning the duplicate rows
To restrict the number of rows to display.

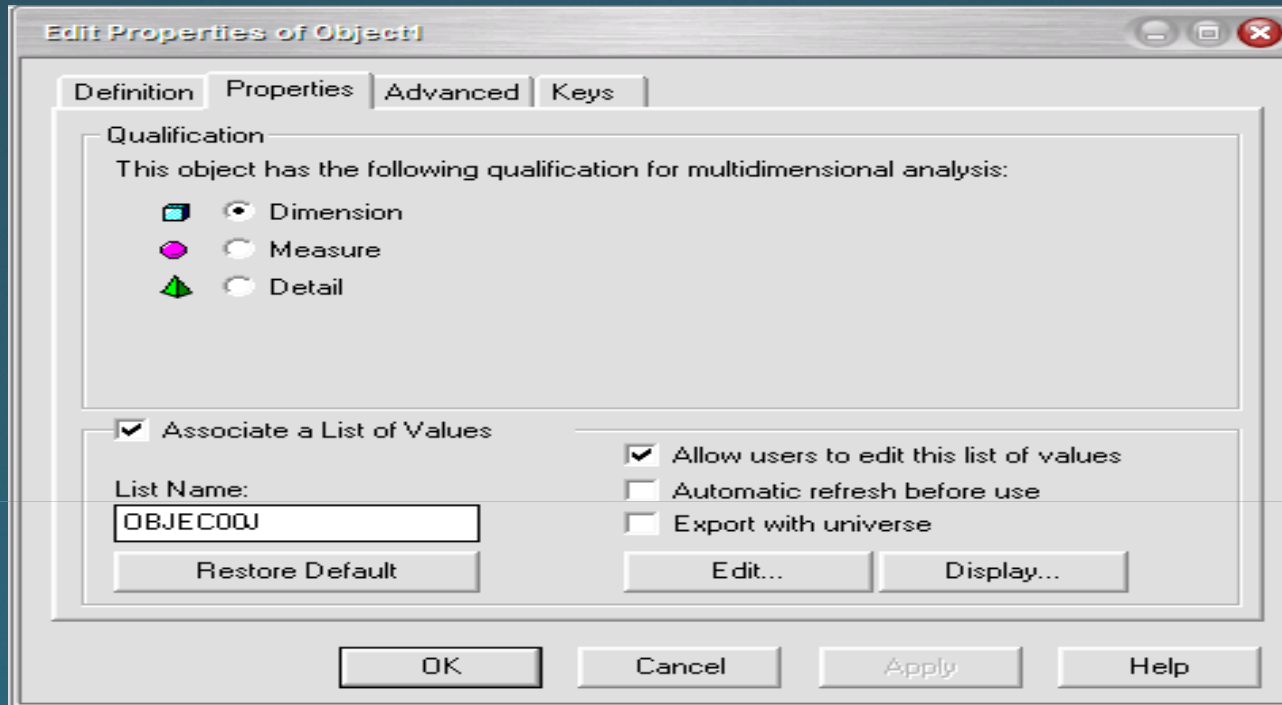


Inserting Classes and Objects




- Inserting Classes
 - To insert classes in designer, click classes in the insert menu.
 - Name the classes and click ok.
 - The classes are like creating another folder in a universe according to the business requirement if required
- Inserting Objects
 - To insert object, click objects in the insert menu.
 - Name the object, specify the data type and in the Select column click the arrow and select the column from the tables.
 - The object can be any columns or derived columns.



Object types



In properties the user have to select the object qualification for multidimensional analysis as any one of the following

- Dimension 
- Measure 
- Detail 



Dimension, Measure and Detail Objects

Dimension - Focus of analysis in a query. A dimension maps to one or more columns or functions in the database that are key to a query. Dimension is the default qualification t object creation.

Ex- Resort, Country, employee

Measure - Contains aggregate functions that map to statistics in the database. Measures are very flexible objects as they are dynamic. The returned values for a measure object vary depending on the dimension and detail objects used with it in the query. It always return numeric information.

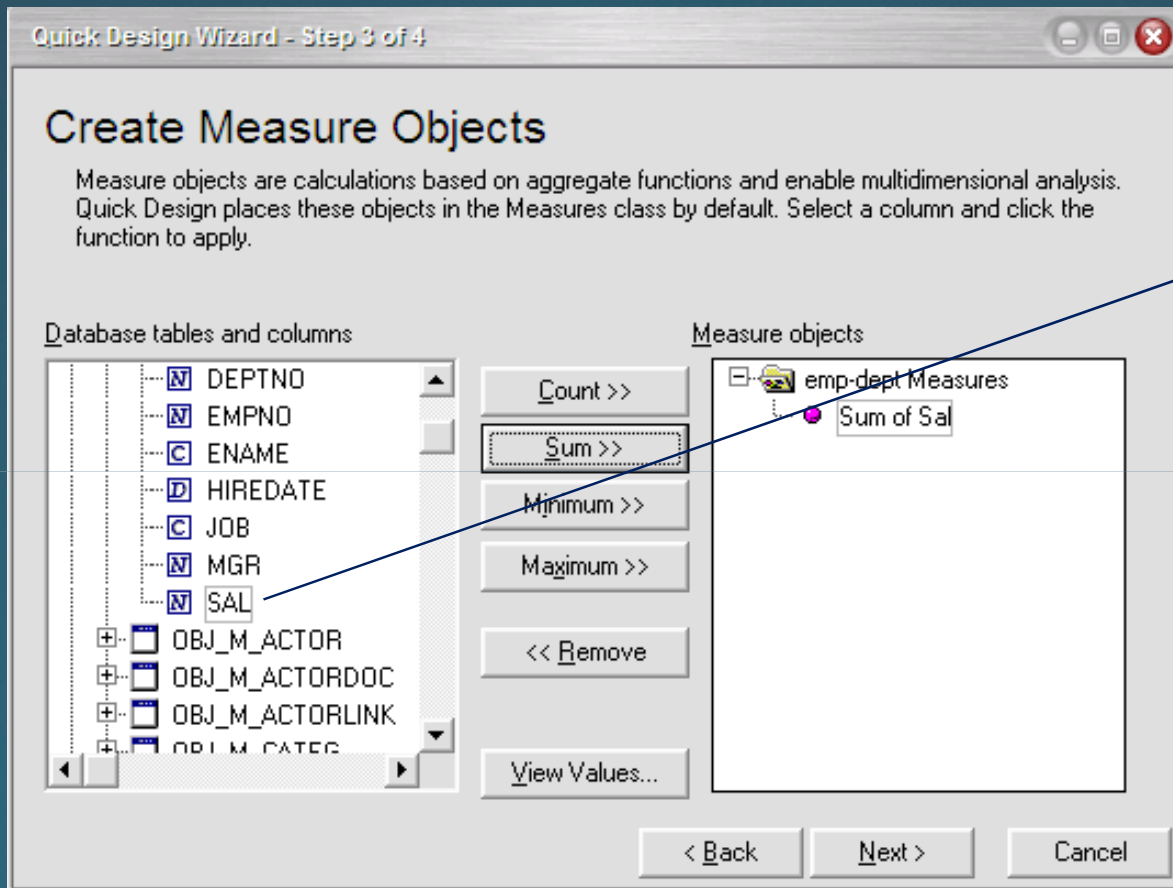
Ex- Revenue, number of guests, margin

Details - Provides descriptive data about a dimension. A detail is always attached to a dimension. It maps to one or more columns or functions in the database that provide detailed information related to a dimension.

Ex- Age, Phone number, address



Creating Measures



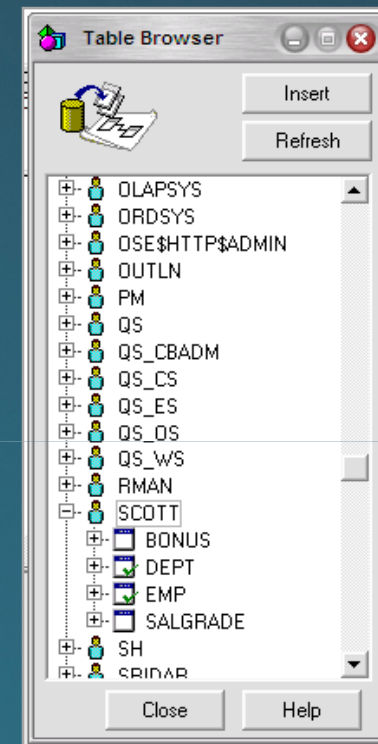
Select the measure element from the tables.

Ex-
select sal from EMP
and click sum to get
the measure as sum
(sal)

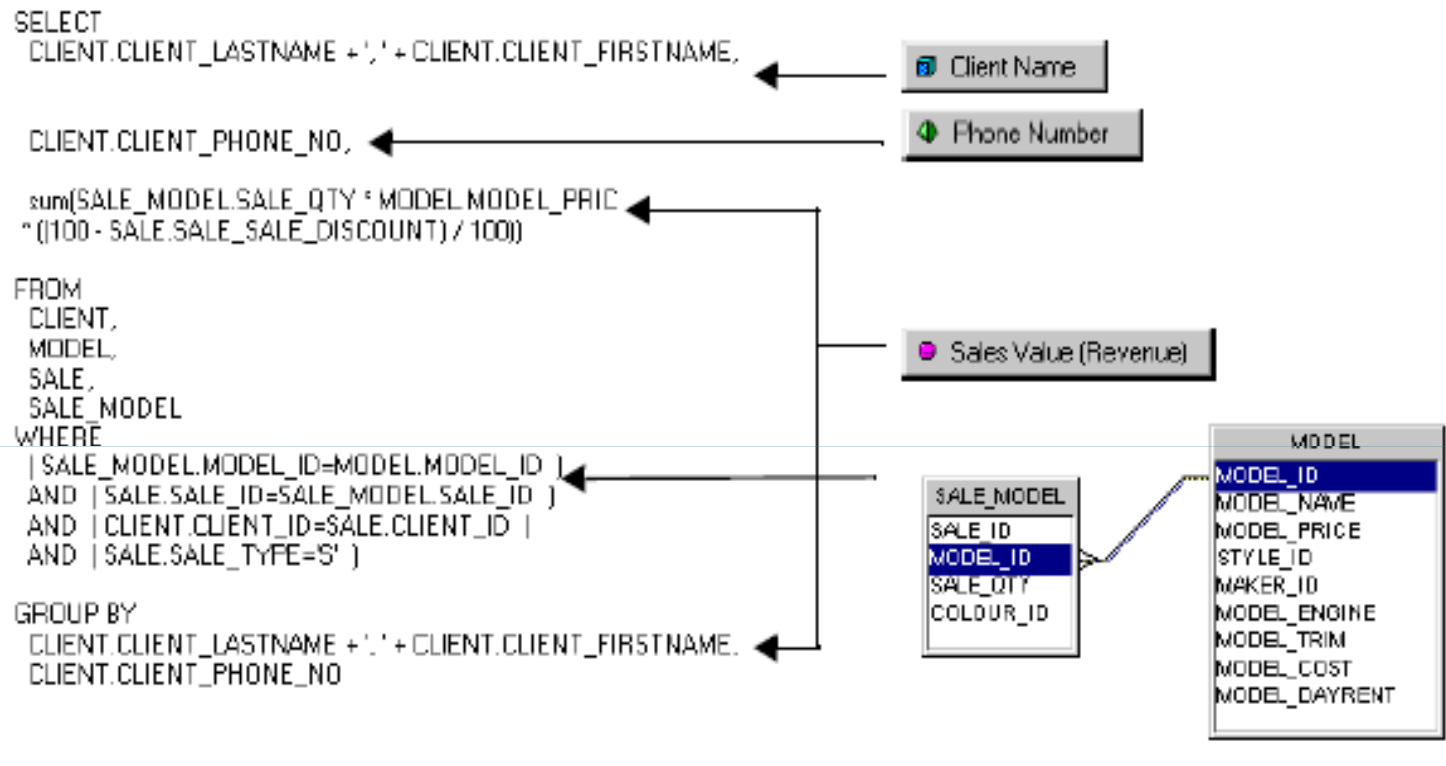


Inserting Tables, Derived tables, Classes & Objects in the Universe

- **Tables**
 - Tables can only retrieve from the database.
 - To insert a table in designer, click the tab insert in the menu.
 - Select the tables from the list and click insert.
- The table browser shows the tables list in the database.
- The user can insert any tables from the list.
- We need to create a join for the inserted table to link with the existing tables.



How BO generates the SQL



Derived Tables in BO

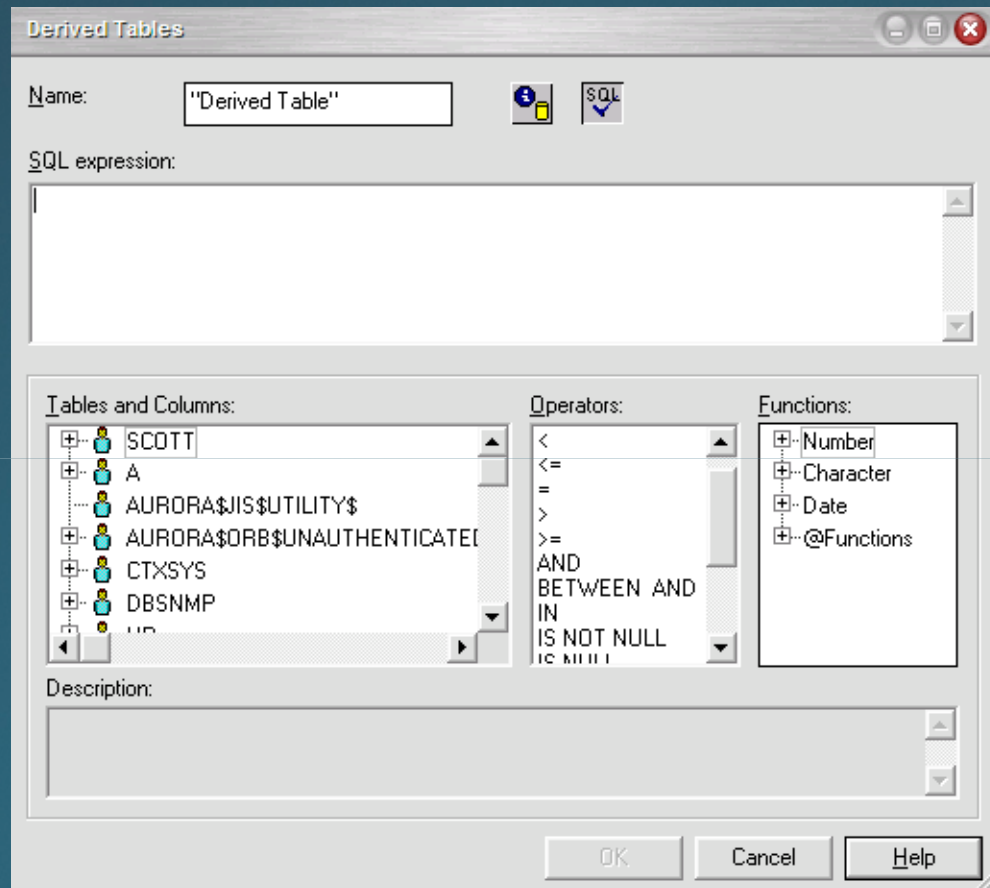
- A derived table is defined by an SQL query at the universe level that can be used as a logical table in Designer.

Derived tables have the following advantages:

- You can include complex calculations and functions in a derived table. These operations are performed before the result set is returned to a document, which saves time and reduces the need for complex analysis of large amounts of data at the report level.
- Derived tables can, in some cases, replace statistical tables that hold results for complex calculations that are incorporated into the universe using aggregate awareness. These aggregate tables are costly to maintain and refresh frequently. Derived tables can return the same data and provide real time data analysis.
- Derived tables are similar to database views, with the advantage that the SQL for a derived table can include Business Objects prompts.



Creating Derived tables



To user can either type the query or use the tables and columns, operators and functions.

The derived tables appears in the schema with the physical database tables.



Scenario where you will use Derived table

- Think you have customer table (cust_id, name, city) and supplier table (supp_id, name, city)
- End user wants to do analysis of customer and supplier based on the city
- In this case you cannot join the supplier and customer table to get the information.
- If you join the customer and supplier table with the city column then it becomes many to many which inflates the results
- So, in this case you can create a derived table with the following select
 - Select city from customer union select city from supplier
- Now you do join the derived table with customer and supplier tables based on the city and do the analysis.



Security at Object Level

- The advance option will give the security option to the object.

Security level access.

- Defines the security access level of the object. You can select a security level which restricts use of the object to end users with the appropriate security level. Security access levels are assigned to end user profiles in Business Objects Supervisor by an administrator.
- You can assign the following security access levels: Public -- Confidential -
- Private
- If you assign Public then all users can see and use the object. If you assign Restricted, then only users with the user profile of Restricted or higher can see and use the object.

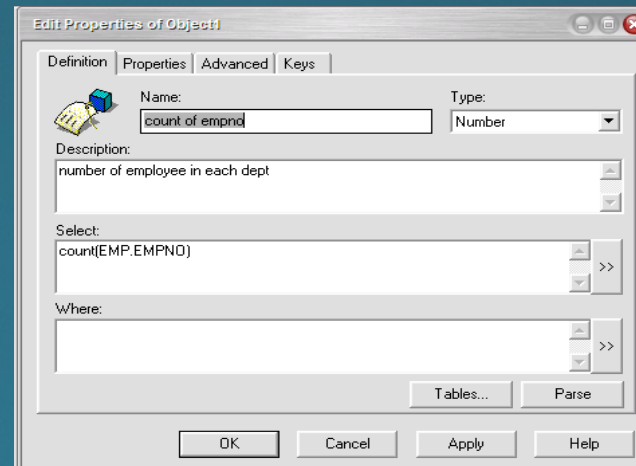


Conditions

- Used to create the pre defined filters which we can use to get the data required
- Its use full for end users to get data based on certain criteria like get all returned customers, grade A, grade B, grade C customers etc
- These conditions are used as report filters by the end users while developing reports.
- The usage makes the reusability to a great extent in the environment.
- We recommend to use as many conditions required to make your end user very effective while creating reports.

Defining your own objects (derived)

To create a object, the user need to select the data type of the object. In the select box, click the arrow then you will get the list of tables and columns, classes and objects, operators and functions. In the where box, specify the condition



The screenshot shows a dialog box titled "Edit Properties of Object". It has four tabs: "Definition", "Properties", "Advanced", and "Keys". The "Definition" tab is selected. The "Name" field contains "count of empno" and the "Type" dropdown is set to "Number". The "Description" field contains "number of employee in each dept". The "Select" field contains "count(EMP.EMPNO)". The "Where" field is empty. There are "Tables..." and "Parse" buttons at the bottom right of the dialog, and "OK", "Cancel", "Apply", and "Help" buttons at the bottom.



Keys in BO

- The Keys tab allows you to define index awareness for an object. Index awareness is the ability to take advantage of the indexes on key columns to speed data retrieval.
- The objects that you create in Designer are based on database columns that are meaningful to an end user. For example, a Customer object retrieves the field that contains the customer name. In this situation the customer table typically has a primary key (for example an integer) that is not meaningful to the end user, but which is very important for database performance. When you set up index awareness in Designer you tell Designer which database columns are primary and foreign keys. This can have a dramatic effect on query performance in the following ways:
- Designer can take advantage of the indexes on key columns to speed data retrieval.



Joins in BO

Joins?

- Once you have inserted more than one table in the schema, you need to create joins between related tables.
- A join is a condition that links the data in separate but related tables. The tables usually have a parent-child relationship. If a query does not contain a join, the database returns a result set that contains all possible combinations of the rows in the query tables. Such a result set is known as a Cartesian product and is rarely useful.

Why use joins in a schema?

- You use joins to ensure that queries returning data from multiple tables do not return incorrect results. A join between two tables defines how data is returned when both tables are included in a query.
- Each table in a schema contains data in one or more columns that correspond to user requirements. Linking all tables in the schema with joins ensures that you restrict the number of ways that data from columns in different tables can be combined in a query.
- Joins limit column combinations between tables to matching or common columns. This prevents result data being returned that contains information from columns that have no sense being matched.
- Manual join defined in the Where clause for an object, are created at run time, so are not considered by Designer for integrity checks and context detection.



Types of join

EQUI-Joins

Link tables based on the equality between the values in the column of one table and the values in the column of another. Because the same column is present in both tables, the join synchronizes the two tables.

Theta-Joins

Link tables based on relationship other than equality between two columns. A theta join could use any operator other than "equal" operator. Ex- to find customers age between the minimum & maximum (customer.age between age_group.age_min and age_group.age_max)

Outer-Joins

Link two tables, one of which has rows do not match those in common column of other table. You define outer join by specifying which table is outer table in original equi-join. The outer table contains column for which you want to return all values, even if they are unmatched.

Shortcut Joins

Join providing alternative path between two tables, bypassing intermediate tables, leading to same result, regardless of direction. Optimized query time by cutting long join paths as short as possible



Working with Joins

To insert join, click join in the insert menu.

The screenshot shows the 'Edit Join' dialog box with the following details:

- Table1:** SCOTT.DEPT (Selected)
- Table2:** SCOTT.EMP (Selected)
- Join Condition:** =
- Join Type:** Inner join (Selected)
- Outer join:** Outer join
- Cardinality:** Cardinality. 1 N. Detect. 1 N.
- Shortcut join:** Shortcut join
- Expression:** SCOTT.DEPT.DEPTNO=SCOTT.EMP.DEPTNO
- Buttons:** Edit..., Parse

In the join editor, we have to select the respective table from the list boxes.

-The combo box in between the table1 and table2 is to specify the join condition either it is inner join, outer join, etc.,

- The cardinality use to detect the relationship between the two tables.

- In the expression, we can view the joined key columns.

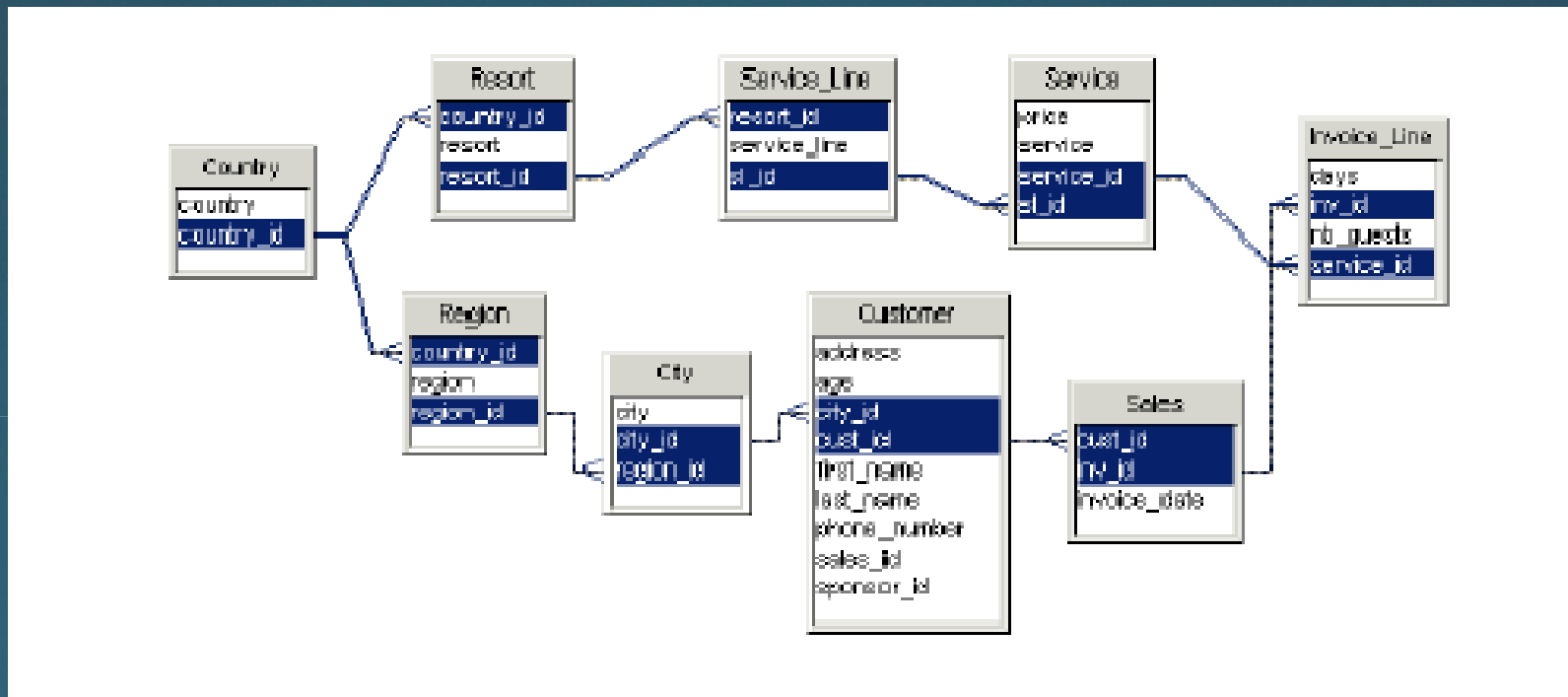


Join Problems

Type of Join Path	Returns	Description
Loop	Too few rows	Joins form multiple paths between lookup tables.
Converging many to one joins	Too many rows	Many to one joins from two fact tables converge on a single lookup table. This type of join convergence can lead to a join path problem called a chasm trap.
Serial many to one joins	Too many rows	A one to many join links a table which is in turn linked by a one to many join. This type of fanning out of one to many joins can lead to a join path problem called a fan trap.



Loop (Example)



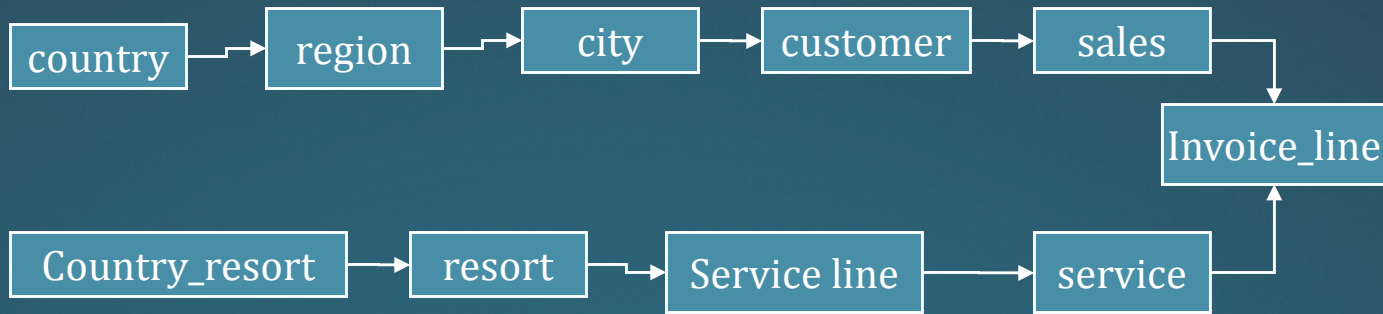
Loops

It also happens in the following model.

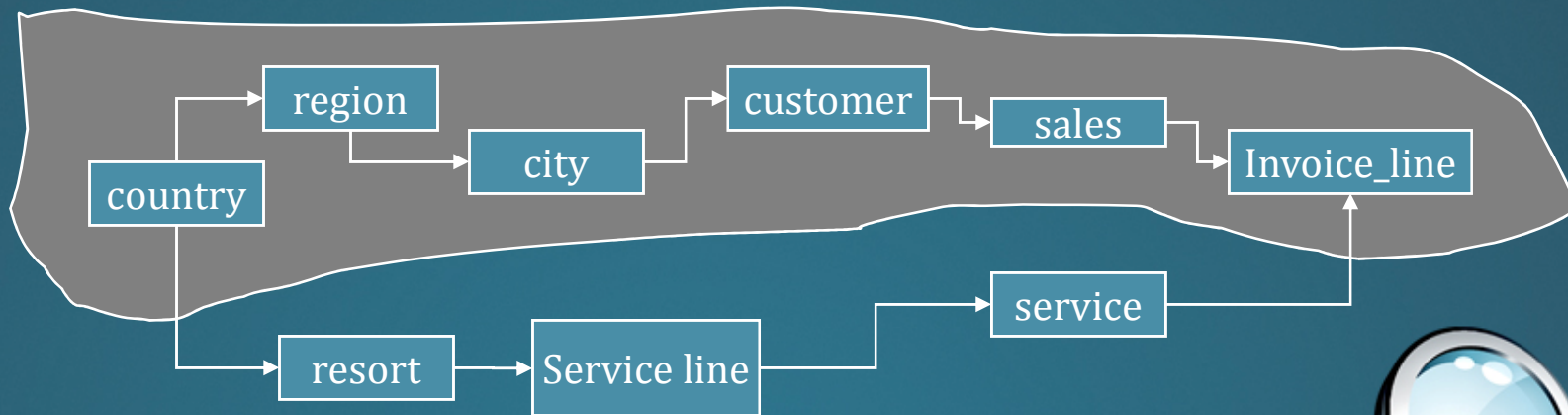
- Say we have country table which referred by suppliers and customers table
- Here when we design our schema, there will be one to many relationship between country and supplier and there will be one to many relationship between the country and customer.
- In this case when we join supplier, customer and country we may loose some records if we don't have the supplier and customer in the same country.
- In this case we loose records. So when we you are not getting the expected number of rows, then you can check on the loops.



Loop solving by alias



Loop solving by context



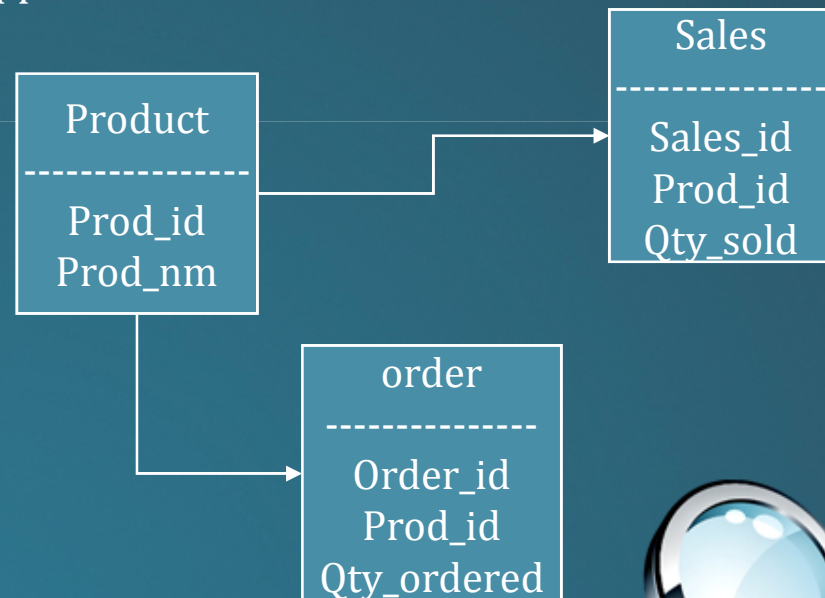
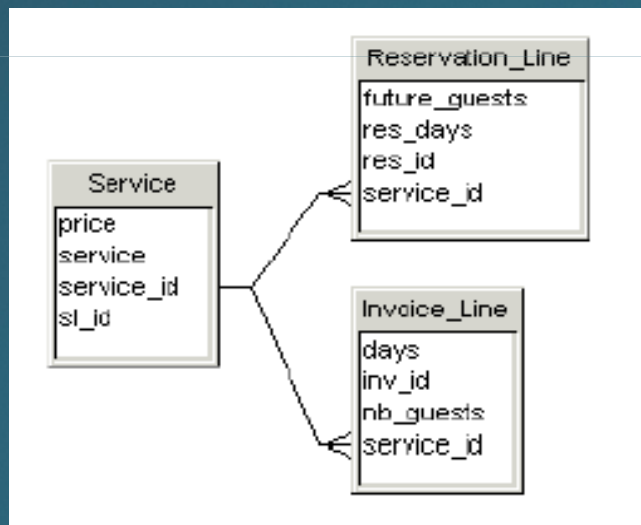
Aliasing and Contexts for join problems

- **Alias**
 - Aliases are references to existing tables in a schema. An Alias is a table that is an exact duplicate of the original table (base table), with a different name. The data in the table is exactly the same as the original table, but the different name "tricks" the SQL of a query to accept that you are using two different tables.
- **Context**
 - Contexts are a collection of joins which provide a valid query path for BusinessObjects and Web Intelligence to generate SQL.

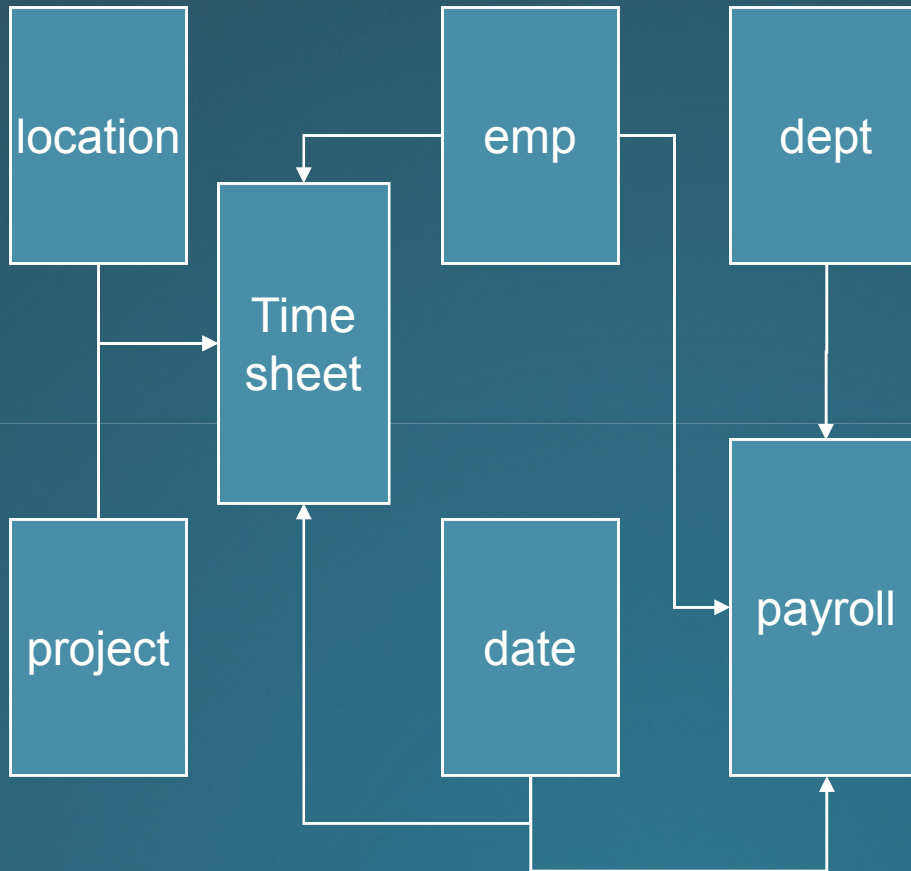


Chasm Trap

- It happens in the D/W environment where we have multiple star schemas
- When you have a confirmed dimension where we share the same dimension key to more than one fact, we get more records than we expected.
- Consider we have product_dim, sales_fact (sales_perspective) and order_fact (supplier perspective), we get this chasm trap
- We create contexts in order to solve the issue. Context being product sales in terms of customer, product orders in terms of Supplier.



Chasm trap and Loop - Example2



Location.loc_id = timesheet.loc_id
Project.proj_id = timesheet.proj_id
Emp.emp_id = timesheet.emp_id
Date.date_id = timesheet.date_id
Emp.emp_id = payroll.emp_id
Date.date_id = payroll.date_id
Dept.dept_id = payroll.dept_id

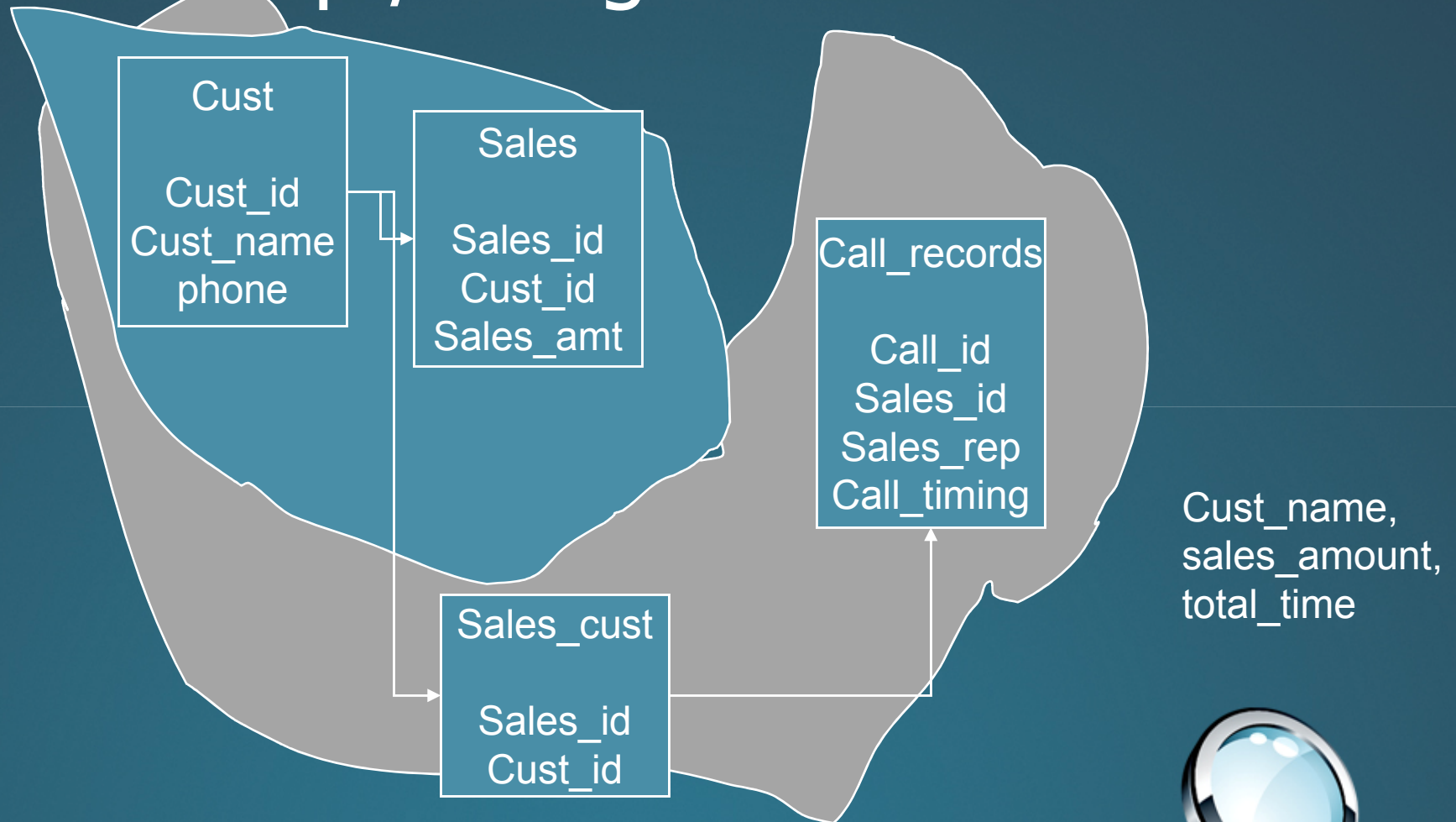
Timesheet

payroll

BY Ganesh Jegannathan, MphasiS an HP Company



Fan Trap / Usage of derived table



@ Functions

These functions are used in BO to generate SQL.
This provides flexibility to change the select / where clause in the SQL.

@aggregate_aware -- used only when aggregate tables exists in the Datawarehouse or data mart. (This is nothing but Query rewrite mechanism of Oracle database)

@Prompt

```
@Prompt('message','type',[lov],[MONO|MULTI],[FREE|CONSTRAINED])
```

In the where clause you type the following function

```
country = @Prompt('Enter Country','A','Customers\Country','MONO','FREE')
```

@Where

@Select

@Variable

You can use the variable in the where clause. Usage - @variable ('BROWSER')

@Script

This is involving through VBA script. If you are dealing with generation of complex where clause you see the SCRIPT variable. If I find any code I will pass it on to you. VBA macros can run only in Windows environment.



@Function	Description	Usually used in object
@Aggregate_Aware	Incorporates columns containing aggregated and dimension data into objects.	Select statement
@Prompt	Prompts user to enter a value for a restriction each time the object using the @Prompt function is included in a query.	<ul style="list-style-type: none"> Select statement Where clause
@Script	Runs a script each time the object using the @Script function is included in a query.	Where clause
@Select	Allows you to use the Select statement of another object.	Select statement
@Variable	Calls the value of a variable stored in memory, for example in a referenced text file.	Where clause
@Where	Allows you to use the Where clause of another object.	Where clause



Aggregate Awareness

- @Aggregate_awareness
 - This is equivalent to Query re write option in Oracle database.
 - Three steps
 - Identify the summary tables
 - Higher aggregate tables (YYYY wise)
 - Medium aggregate table (QQ wise)
 - Low level aggregate table (Month wise)
 - Detail table (Day wise)
 - Use Aggregate awareness functions in all the objects.
 - Set the properties in Aggregate Navigation.



List Of Values

- It's the data which is associated to an object
 - We can associate data either from RDBMS or from a flat file
 - By default every object gets associated to an LOV, when you choose to display it runs Select DISTINCT query and populates the results.
 - List of values based on the external file is fixed.
- Don't associate the LOV for the measures and Detail objects as this wont be usually used as where clause in the analytical queries.
- LOV values can be associated from a file as well.

Aggregate Awareness

- This is equivalent to Query re write option in Oracle database.
- Three steps
 - Identify the summary tables
 - Higher aggregate tables (YYYY wise)
 - Medium aggregate table (QQ wise)
 - Low level aggregate table (Month wise)
 - Detail table (Day wise)
 - Use Aggregate awareness functions in all the objects.
 - Set the properties in Aggregate Navigation.



Hierarchies

- Are used to perform multidimensional analysis.
- By default the class objects become the default hierarchy for the analysis.
- If you want to create customized hierarchies from different classes then create the custom hierarchy.
- While we use the drill or slice and dice in BO or Webi you can see these hierarchies so that you can use it for analysis.

Linking Universes

- In order to reuse the existing universes, to replicate the existing classes and objects from existing universe we can link the universes.
- For example, say we have HR universe got created and used in the company. We need to create a SMG (Sales and Marketing) universe, but want to reuse the existing Employee class and Period class as in HR universe then while creating the SMG universe, link the HR universe.
- Once you link the HR universe, the system inherits all the tables and objects into SMG.
- In SMG universe, you can hide the objects we don't want to show, and add the required tables and objects.
- Either you can link or include the universe while creating the new one.
 - Include copies the entire universe to new universe, thus it allows any changes in the universe.
 - Link refers the objects in the new universe, in this we cannot change the existing objects as it gets inherited

BY Ganesh Jegannathan, MphasiS an HP Company



Linking Universes

- In order to reuse the existing universes, to replicate the existing classes and objects from existing universe we can link the universes.
- For example, say we have HR universe got created and used in the company. We need to create a SMG (Sales and Marketing) universe, but want to reuse the existing Employee class and Period class as in HR universe then while creating the SMG universe, link the HR universe.
- Once you link the HR universe, the system inherits all the tables and objects into SMG.
- In SMG universe, you can hide the objects we don't want to show, and add the required tables and objects.
- Either you can link or include the universe while creating the new one.
 - Include copies the entire universe to new universe, thus it allows any changes in the universe.
 - Link refers the objects in the new universe, in this we cannot change the existing objects as it gets inherited.



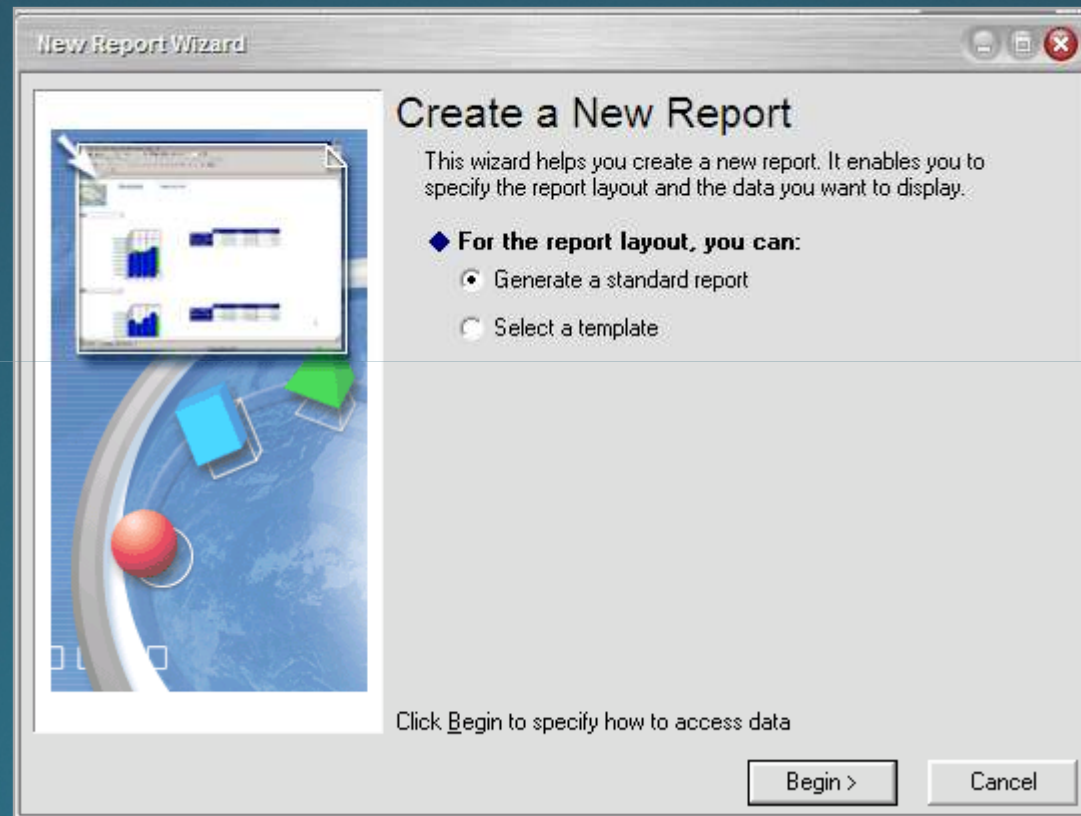
REPORTING



BY Ganesh Jegannathan, MphasiS an *HP* Company

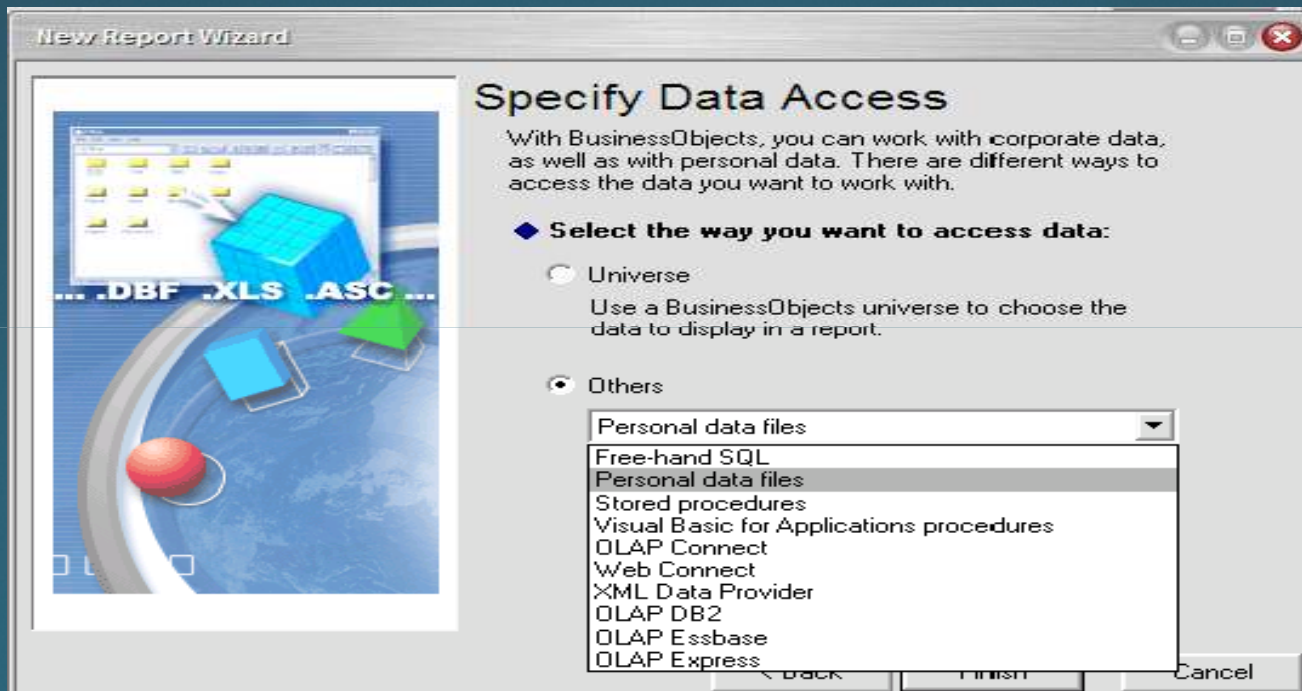
Workflow for accessing data

- Building a data provider when you create a new document.



Specify the data access

- Select universe, to query with universe as data provider.
- To work with other data providers, select others.



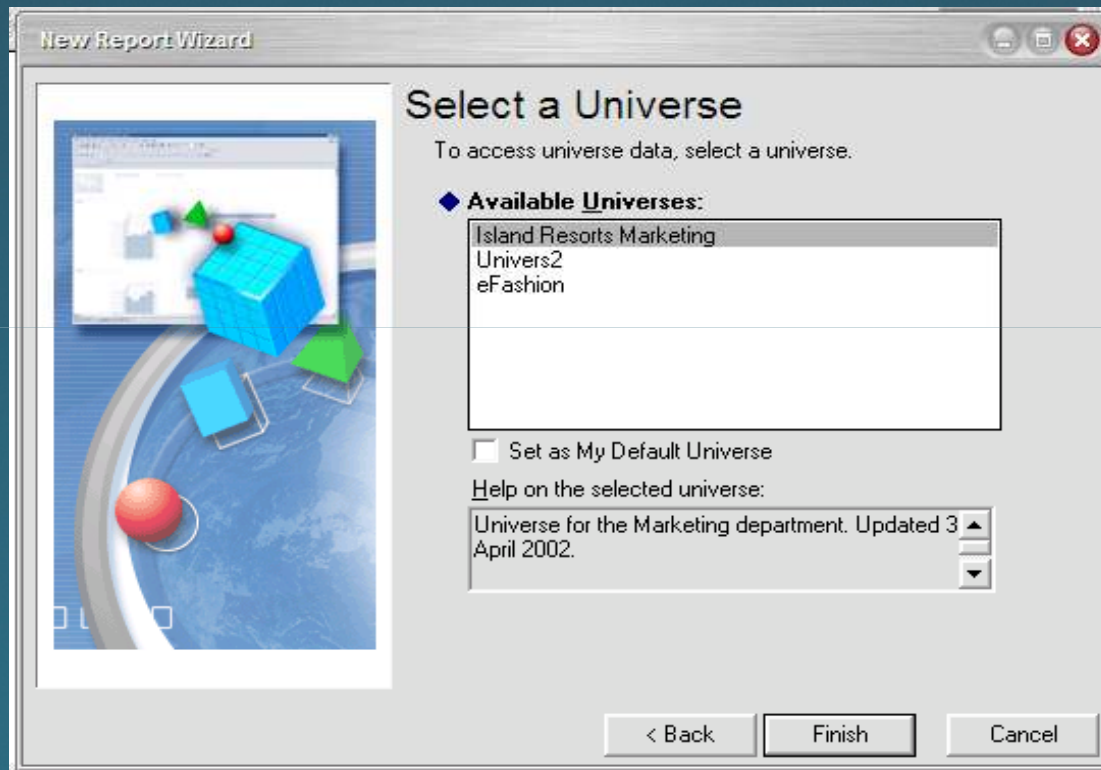
Data provider and their set up

Data provider	Who sets it up?
Queries on universes	The universe designer sets up the connection in the universe, so the connection is completely hidden from you when you build or edit queries. Note: The supervisor may modify the existing connection or assign a new connection to the universe
Stored procedures	The supervisor creates the connection to access the stored procedure.
Free-hand SQL	In free-hand SQL, you can create your own connection to the database. Once you have created the connection, you can make it available to other users.
Personal data files and XML files	When you access data in a personal data file or XML file, you select the file and in doing so, you “connect” to it. This is not a technical task, it’s just a question of selecting the right file
VBA procedures	A VBA procedure runs a VBA macro that retrieves data for your BusinessObjects report. The person who creates the macro defines the connection to the data source in the macro code.
OLAP Servers and SAP	The database administrator at your site sets up the servers so that you can access them from BusinessObjects.
World Wide Web pages	You access WWW pages using the Web Connect data provider.



If Universe

- Select the universe from the list and click finish and it will take to the query panel



If Others

- In other data providers we will discuss about Personal Data Files and Free-Hand SQL
- In Personal Data files, the following are the some of the format used as file.
 - Text files
 - Microsoft Excel
 - dBase
 - XML
- The main benefits of using personal data files are as follows:
 - You can display corporate data next to personal data in the same report.



Browse the file

Access Personal Data

You use this box to specify the file that contains the data you want.

File

Name: C:\Documents and Settings\Sridar K\Deskt Browse...

Format: Text Files (*.asc; *.prn; *.txt; *.csv)

Text File

Delimiter:

- Tabulation
- Space
- Character: |

First row contains column names.

Build Hierarchies and Start in Drill Mode

View... Run Cancel Help

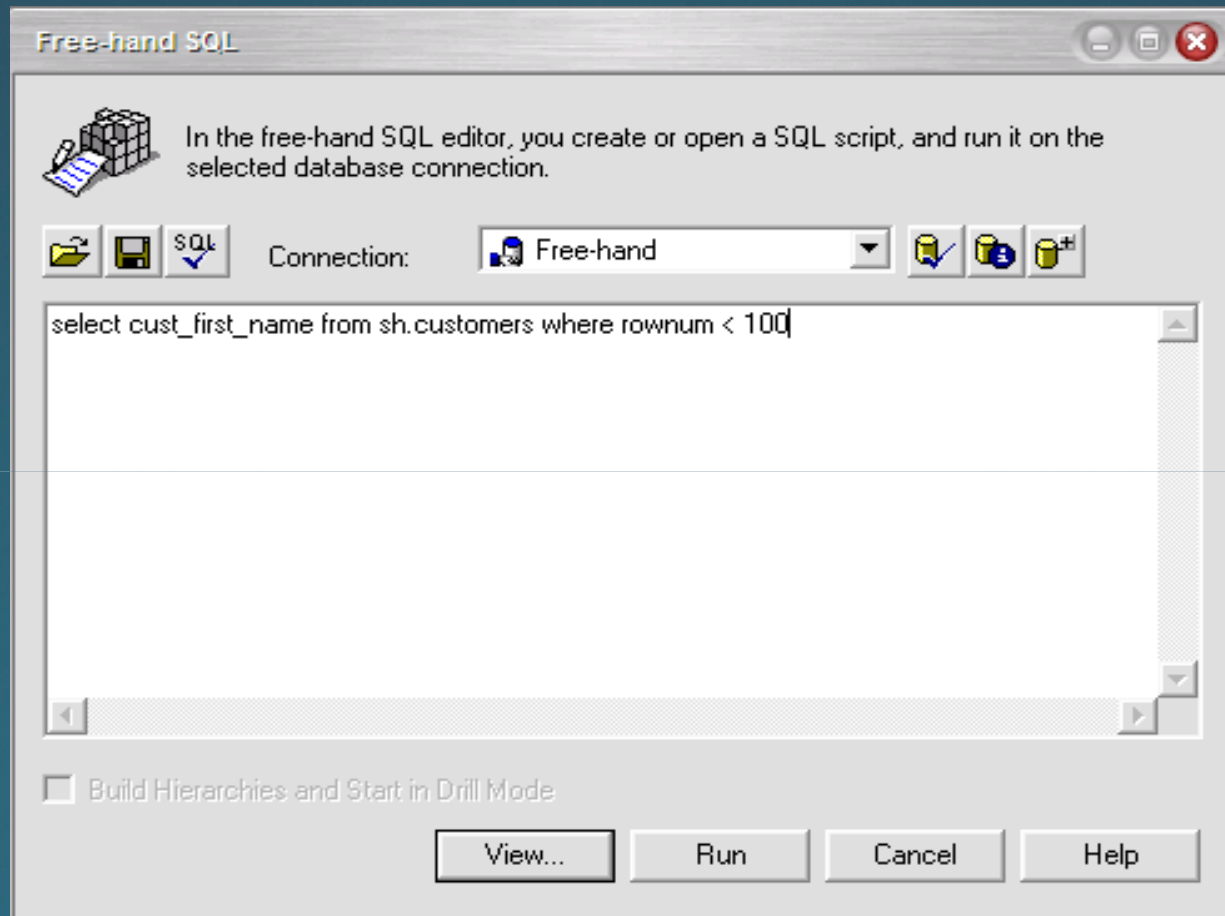
Name- Browse the data file from appropriate folder with the proper format.

Delimi1ter- It shows how the data are separated in the data file. Select *Tabulations*, *Space* or *Character*. (If you select *Character* you need to enter the character that delimits the data.)

To get the column name in the report, click in the check box
Click run, it will take to the Query panel.



Working with Free-Hand SQL



Reporting Environment

The document name displays in the title bar

Menu bar

Find in Report

Toolbars

Report Manager window

Report window

Report Manager border

Vertical scroll bar

The name of the report displays on the report tab.

Status Bar

Horizontal scroll bar

Product Line	Category	Revenue
Accessories	Shirts, jackets, sweaters	\$754,881
	Other accessories	\$147,043
	Shirts, jackets, sweaters	\$20,200
	Jeans	\$99,250
	Leisure wear	\$147,043
	Shirts	\$549,200
Accessories		\$1,980,316

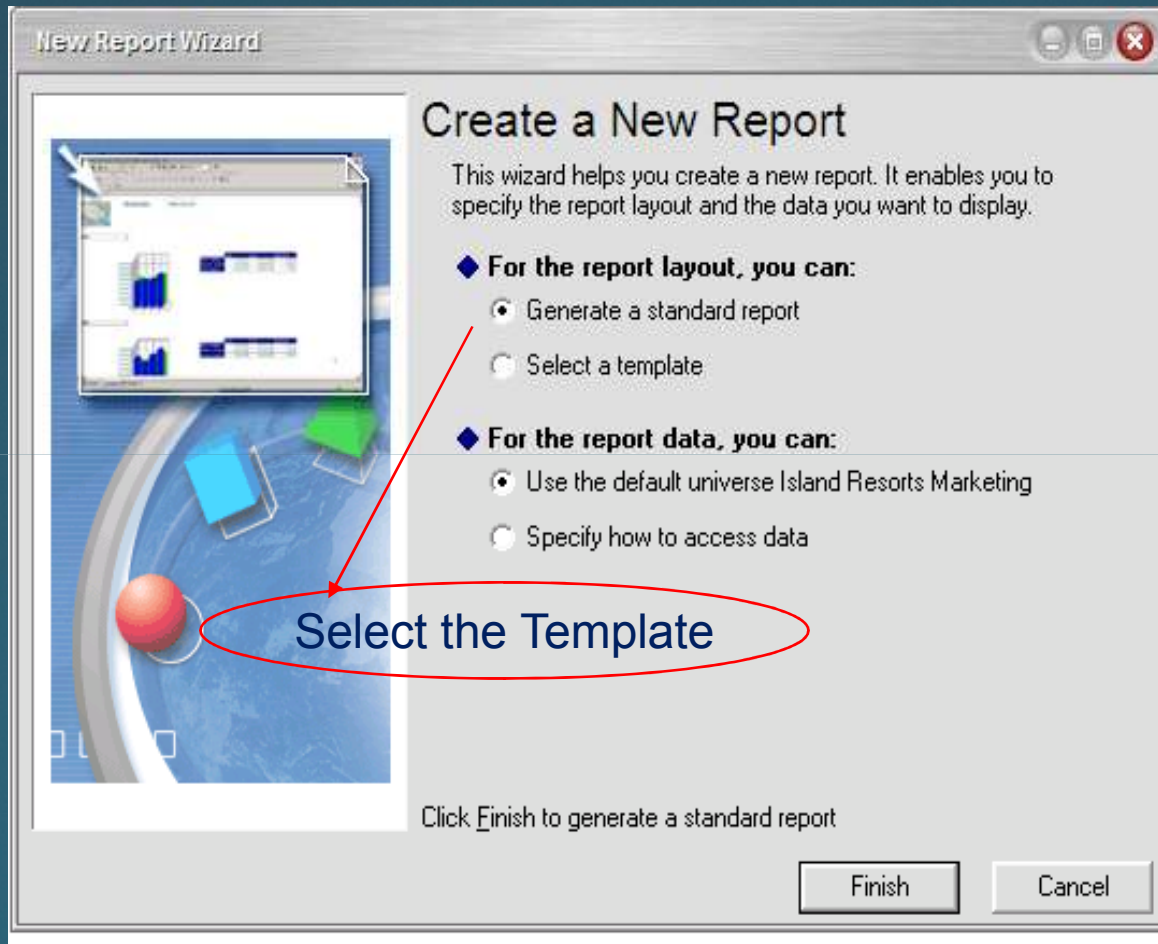
Line revenue pie chart data:

- Shirts: 40%
- Jeans: 34%
- Leisure wear: 10%
- Shirts, jackets, sweaters: 2%
- Other accessories: 1%

BY Ganesh Jegannathan, Mphasis an HP Company



Setting up Templates and Standard Report styles



When you create a new document in Business Objects, you choose in what type of report layout your data will be displayed.

You can either generate a standard report or generate a report from a set of templates.



Types of Reports

- List / Tabular report
- Master/Details Reports.
- Cross Tab reports
- Reports by Charts.
- Drill Down Reports.



Working with Master/Details Reports

- The basic report format is master and details reports.
- Master/detail reports display data in sections. Each section contains a “master” or parent piece of data, for example a resort, or a year. The rest of the data in the section relates to the master.
- A report can have more than one master.
- The details can be a table or chart or cross tab.

Basic Report

Quarter	Resort	Revenue
Q1	Bahamas Beach	224,881.00
Q1	French Riviera	208,565.00
Q1	Hawaiian Club	357,170.00
Q2	Bahamas Beach	237,872.00
Q2	French Riviera	242,165.00
Q2	Hawaiian Club	341,780.00
Q3	Bahamas Beach	263,422.00
Q3	French Riviera	226,125.00
Q3	Hawaiian Club	395,150.00
Q4	Bahamas Beach	245,269.00
Q4	French Riviera	158,565.00
Q4	Hawaiian Club	385,560.00

Master/Details Reports

Q1	
Resort	Revenue
Bahamas Beach	224,881.00
French Riviera	208,565.00
Hawaiian Club	357,170.00
Q2	
Resort	Revenue
Bahamas Beach	237,872.00
French Riviera	242,165.00
Hawaiian Club	341,780.00
Q3	
Resort	Revenue
Bahamas Beach	263,422.00
French Riviera	226,125.00
Hawaiian Club	395,150.00

Select the column Quarter, drag and put it in the top of the table. Then the Column values will become the master values and the rest will be the detail in records.



Cross Tab Reports / Inserting a cross tab

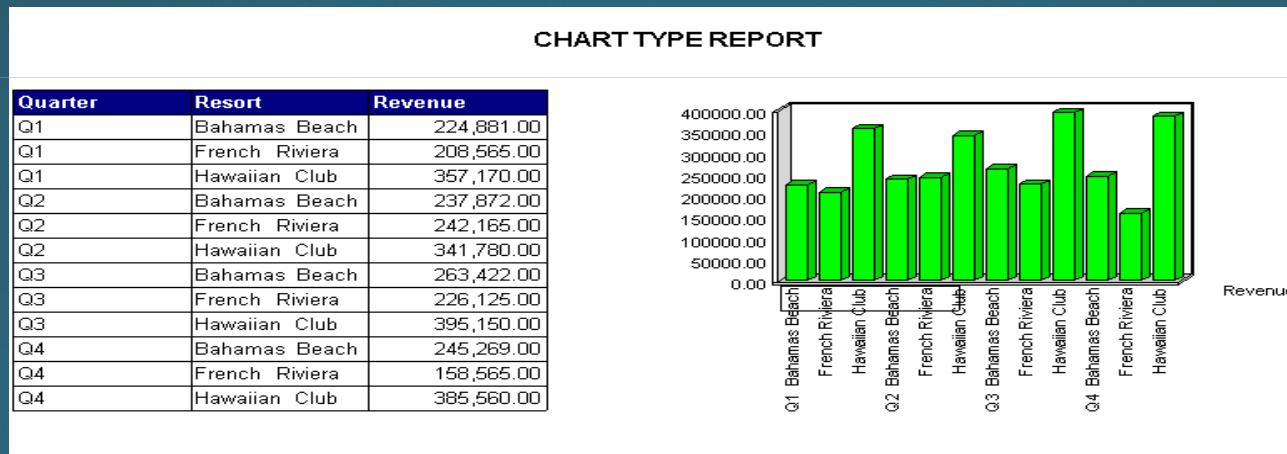
- You can create a crosstab by dragging variables from the Report Manager to the report to create a table and then re-organize the table into a cross tab.
- To create a crosstab, you need at least two dimensions/details and one measure object.
- Either select a column and drag it to the upper right corner of the table or, Drag a variable from the report manager window to the upper right corner of the table
- When the cursor changes to the Turn to cross tab cursor and the status bar displays the message *Drop to turn to cross tab*, release the mouse.
- The table becomes a cross tab

	Q1	Q2	Q3	Q4
Bahamas Beach	224,881.00	237,872.00	263,422.00	245,269.00
French Riviera	208,565.00	242,165.00	226,125.00	158,565.00
Hawaiian Club	357,170.00	341,780.00	395,150.00	385,560.00

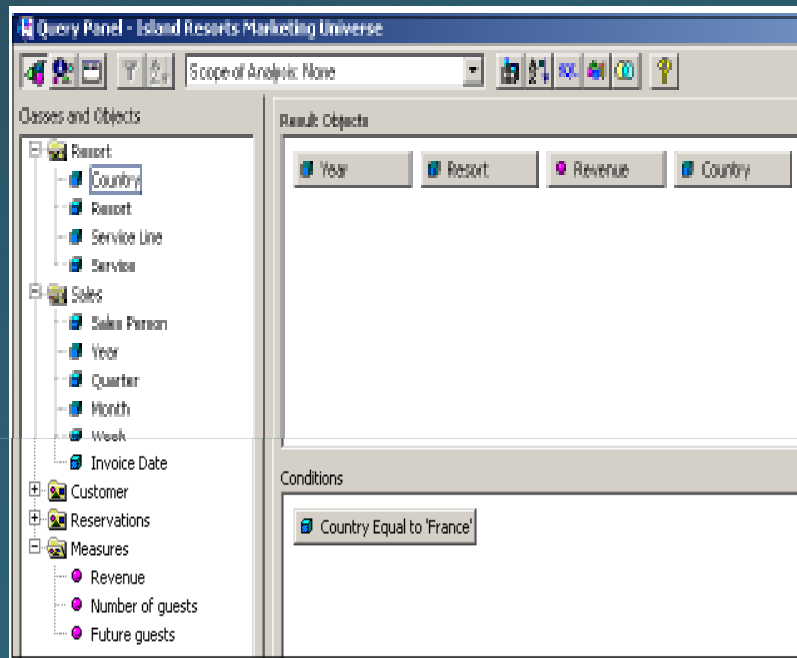


CHART TYPE REPORT

- Business Objects has several basic chart types, some with two-dimensional and some with three-dimensional views.
- There are several ways to create a chart. You can use the Insert Chart wizard
- turn an existing table or crosstab into a chart
- copy and paste an existing table or crosstab and then turn it into chart
- To create a chart you need to use at least one measure object and one dimension or detail object



Single Value Conditioning



SINGLE VALUE CONDITIONING

Year	Resort	Country	Quarter	Revenue
FY1998	French Riviera	France	Q1	76,270.00
FY1998	French Riviera	France	Q2	85,560.00
FY1998	French Riviera	France	Q3	81,085.00
FY1998	French Riviera	France	Q4	53,025.00
FY1999	French Riviera	France	Q1	70,400.00
FY1999	French Riviera	France	Q2	80,050.00
FY1999	French Riviera	France	Q3	74,960.00
FY1999	French Riviera	France	Q4	54,900.00
FY2000	French Riviera	France	Q1	61,895.00
FY2000	French Riviera	France	Q2	76,555.00
FY2000	French Riviera	France	Q3	70,080.00
FY2000	French Riviera	France	Q4	50,640.00

Inserting single condition in query panel



Multi Value Conditioning

The screenshot shows a 'Query Panel - Island Resorts Marketing Universe' window. On the left, a tree view lists 'Classes and Objects' including Resort, Sales, Customer, Reservations, and Measures. The 'Result Objects' section shows 'Country', 'Year', 'Quarter', 'Resort', and 'Revenue'. The 'Conditions' section contains a logical expression: 'Country Equal to 'France'' OR 'Country Equal to 'US'' AND 'Year Equal to 'FY1998''. The 'Scope of Analysis' is set to 'None'.

Year	Resort	Country	Quarter	Revenue
FY1998	Bahamas Beach	US	Q1	67,085.00
FY1998	Bahamas Beach	US	Q2	75,600.00
FY1998	Bahamas Beach	US	Q3	78,788.00
FY1998	Bahamas Beach	US	Q4	66,456.00
FY1998	French Riviera	France	Q1	76,270.00
FY1998	French Riviera	France	Q2	85,560.00
FY1998	French Riviera	France	Q3	81,085.00
FY1998	French Riviera	France	Q4	53,025.00
FY1998	Hawaiian Club	US	Q1	113,505.00
FY1998	Hawaiian Club	US	Q2	111,330.00
FY1998	Hawaiian Club	US	Q3	129,120.00
FY1998	Hawaiian Club	US	Q4	125,730.00

More than one condition to retrieve and filter results



Prompts

Choose Edit Data Provider

Year	Revenue
FY1998	295,940.00
FY1999	280,310.00
FY2000	259,170.00

Operators

- Equal to
- Different from
- Greater than
- Greater than or equal to
- Less than
- Less than or equal to
- Between
- Not between
- In list
- Not in list
- Is null
- Is not null
- Matches pattern
- Different from pattern
- Both
- Except

Result Objects

Year Revenue

Conditions

Country <Select an operator>

In Query panel select Conditioning Object



Equal to Operator

The screenshot shows a query builder interface with two main sections: 'Operands' and 'Conditions'. In the 'Operands' section, a list of options is shown, with 'Type a new prompt' highlighted. In the 'Conditions' section, a condition is defined as 'Country Equal to Prompt ("WHICH COUNTRY")'. The 'Result Objects' section shows 'Year' and 'Revenue' selected.

Operands	Result Objects
<ul style="list-style-type: none">Type a new constantShow list of valuesType a new promptShow list of promptsSelect an objectCreate a subquery (ANY)CalculationSelect Query Results (ANY)	<ul style="list-style-type: none">YearRevenue

Conditions

Conditions
Country Equal to Prompt ("WHICH COUNTRY")

The screenshot shows a query builder interface with two main sections: 'Operators' and 'Conditions'. In the 'Operators' section, a list of operators is shown, with 'Equal to' highlighted. In the 'Conditions' section, a condition is defined as 'Country Equal to <Select an operand type>'. The 'Result Objects' section shows 'Year' and 'Revenue' selected.

Operators	Result Objects
<ul style="list-style-type: none">Equal toDifferent fromGreater thanGreater than or equal toLess thanLess than or equal toBetweenNot betweenIn listNot in listIs nullIs not nullMatches patternDifferent from patternBothExcept	<ul style="list-style-type: none">YearRevenue

Conditions

Conditions
Country Equal to <Select an operand type>

Choose Type a new Prompt
in Operands



Enter or Select Values

WHICH COUNTRY

US

OK
Cancel
Help
Values...

Choose the value from List of Values

List of Values of Country

Tabular View Hierarchical View

Country

Australia
France
Germany
Holland
Japan
UK
US

Show Selected Only

Refresh OK Cancel Help

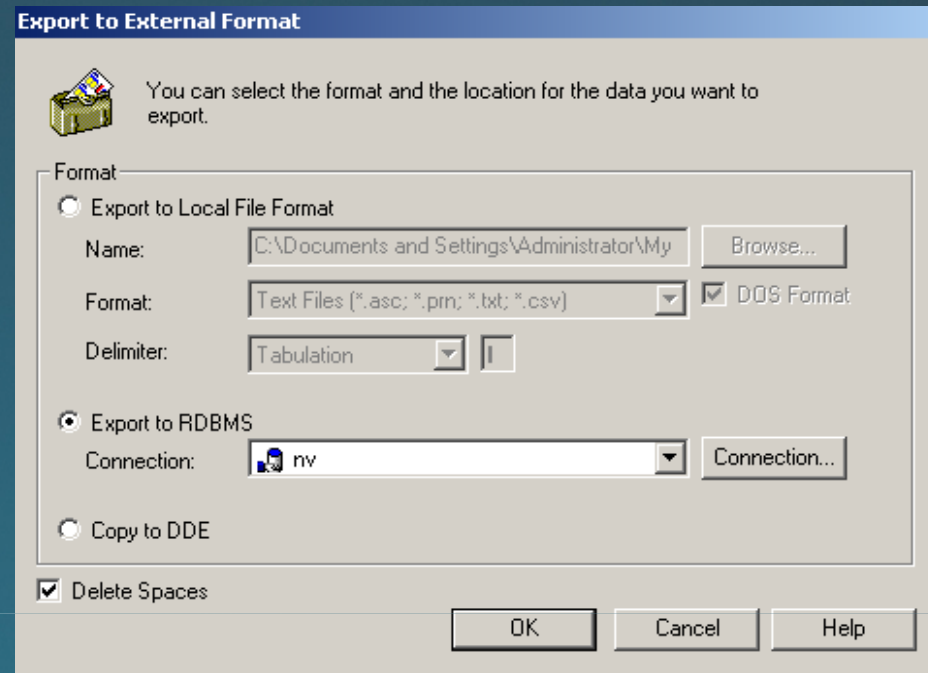
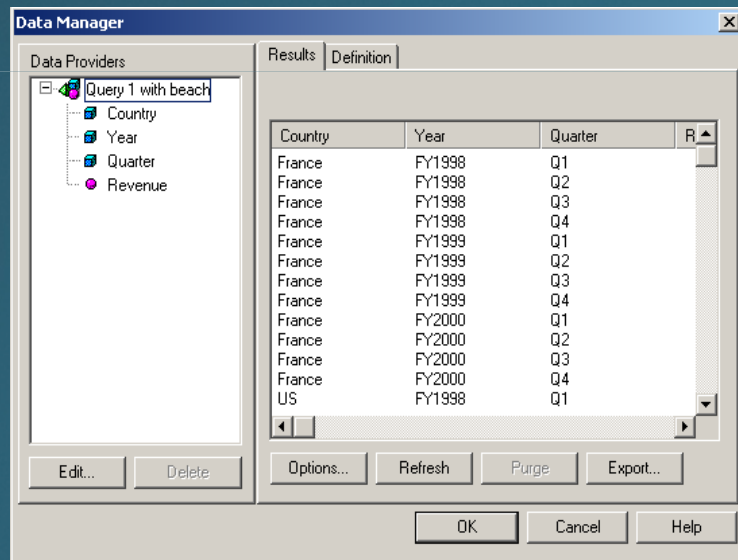
Then it will give the result as below

Year	Country	Revenue
FY1998	US	767,614.00
FY1999	US	826,930.00
FY2000	US	856,560.00



BY Ganesh Jegannathan, MphasiS an HP Company

Exporting



We can export the Report Data to Excel, Text file, RDBMS etc.



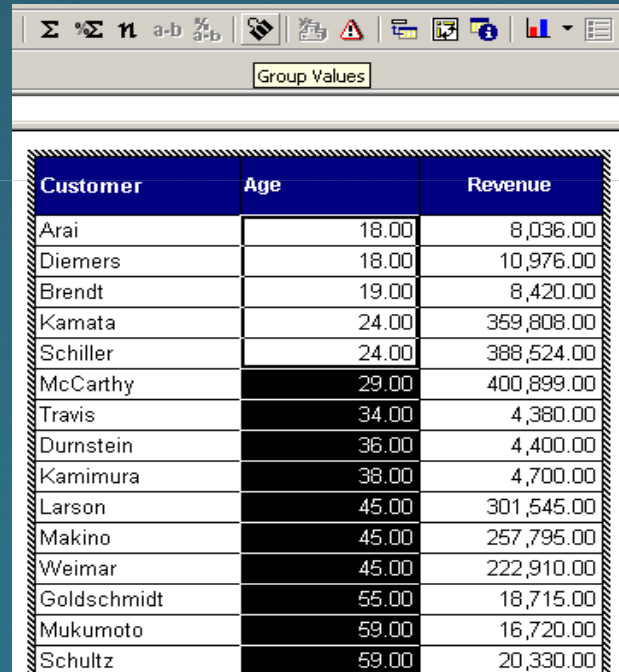
Creating Local Variables by Grouping Values

You can also create new variables by grouping the values of existing variables which enables you to create dynamic groups for analysis purposes. Grouping values prior to analyzing your data in drill mode is useful, for example if you have variable that has many values then it allows you to create an intermediate level of detail in your analysis

EX :

In the given table we can group customers according to their age group

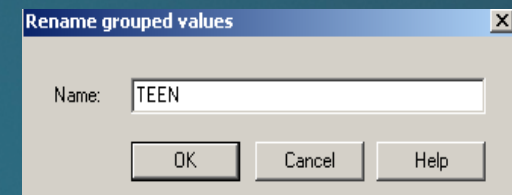
Customer	Age	Revenue
Arai	18.00	8,036.00
Baker	64.00	441,594.00
Brendt	19.00	8,420.00
Diemers	18.00	10,976.00
Durmstein	36.00	4,400.00
Goldschmidt	55.00	18,715.00
Kamata	24.00	359,808.00
Kamimura	38.00	4,700.00
Larson	45.00	301,545.00
Makino	45.00	257,795.00
McCarthy	29.00	400,899.00
Mukumoto	59.00	16,720.00
Okumura	74.00	11,872.00
Oneda	68.00	387,088.00
Reinman	72.00	12,112.00
Schiller	24.00	388,524.00
Schultz	59.00	20,330.00
Swenson	74.00	10,704.00
Titzman	63.00	394,996.00
Travis	34.00	4,380.00
Weimar	45.00	222,910.00



Customer	Age	Revenue
Arai	18.00	8,036.00
Diemers	18.00	10,976.00
Brendt	19.00	8,420.00
Kamata	24.00	359,808.00
Schiller	24.00	388,524.00
McCarthy	29.00	400,899.00
Travis	34.00	4,380.00
Durmstein	36.00	4,400.00
Kamimura	38.00	4,700.00
Larson	45.00	301,545.00
Makino	45.00	257,795.00
Weimar	45.00	222,910.00
Goldschmidt	55.00	18,715.00
Mukumoto	59.00	16,720.00
Schultz	59.00	20,330.00

Select the ages you want to group together and click group values

After selecting group values we can rename the group



Rename grouped values

Name:

OK Cancel Help



After Forming groups the table will look like this

age factor	Customer	Revenue
old	Baker	441,594.00
old	Goldschmidt	18,715.00
old	Mukumoto	16,720.00
old	Okumura	11,872.00
old	Oneda	387,088.00
old	Reinman	12,112.00
old	Schultz	20,330.00
old	Swenson	10,704.00
old	Titzman	394,996.00
middleaged	Durnstein	4,400.00
middleaged	Kamimura	4,700.00
middleaged	Larson	301,545.00
middleaged	Makino	257,795.00
middleaged	McCarthy	400,899.00
middleaged	Travis	4,380.00
middleaged	Weimar	222,910.00
teens	Arai	8,036.00
teens	Brendt	8,420.00
teens	Diemers	10,976.00
teens	Kamata	359,808.00
teens	Schiller	388,524.00

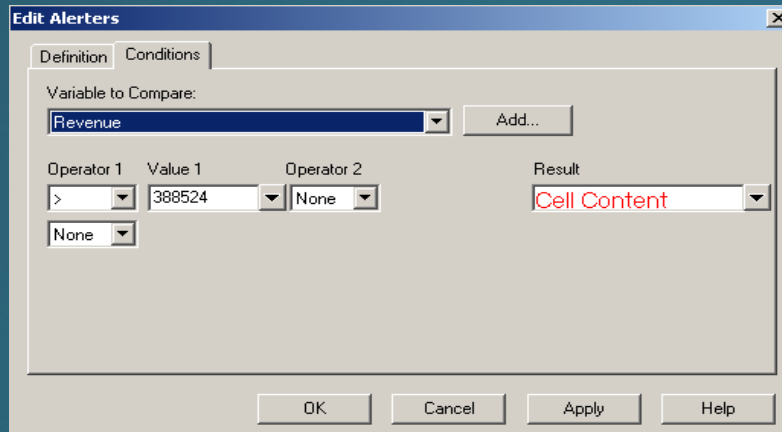
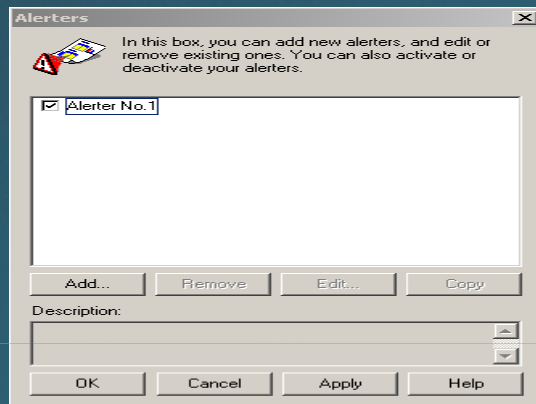
age factor	Customer	Revenue
old	Baker	441,594.00
	Goldschmidt	18,715.00
	Mukumoto	16,720.00
	Okumura	11,872.00
	Oneda	387,088.00
	Reinman	12,112.00
	Schultz	20,330.00
	Swenson	10,704.00
	Titzman	394,996.00
middleaged	Durnstein	4,400.00
	Kamimura	4,700.00
	Larson	301,545.00
	Makino	257,795.00
	McCarthy	400,899.00
	Travis	4,380.00
teens	Weimar	222,910.00
	Arai	8,036.00
	Brendt	8,420.00
	Diemers	10,976.00
	Kamata	359,808.00
Schiller	388,524.00	

Apply center across break to group table like this



Alerter

In the previous table given we need to differ customers having revenue greater than 388524. Select alerter and place condition there



age factor	Customer	Revenue
old	Baker	441,594.00
	Goldschmidt	18,715.00
	Mukumoto	16,720.00
	Okumura	11,872.00
	Oneda	387,088.00
	Reinman	12,112.00
	Schultz	20,330.00
	Swenson	10,704.00
middleaged	Titzman	394,996.00
	Durnstein	4,400.00
	Kamimura	4,700.00
	Larson	301,545.00
	Makino	257,795.00
	McCarthy	400,899.00
teens	Travis	4,380.00
	Weimar	222,910.00
	Arai	8,036.00
	Brendt	8,420.00
	Diemers	10,976.00
	Kamata	359,808.00
	Schiller	388,524.00



Variance Percentage

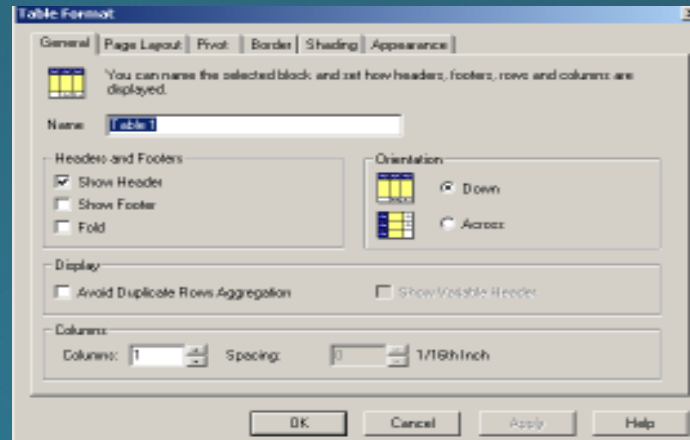
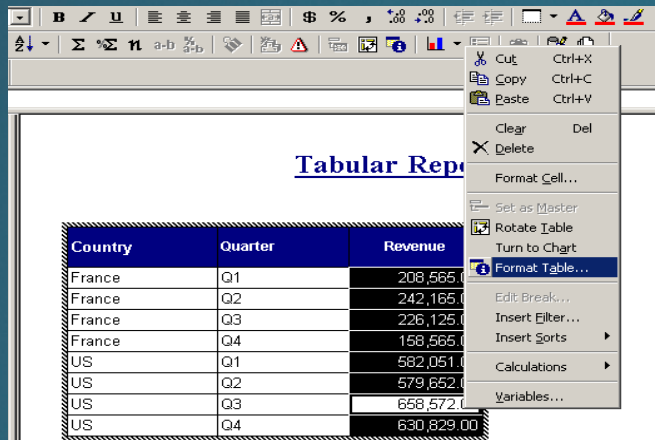
Calculating the difference in revenue between two quarters

In this example you have a cross tab that displays revenue per resort per quarter and total revenue per quarter. You want to show in the table the difference in revenue between two quarters

a-b $\frac{a-b}{a-b}$

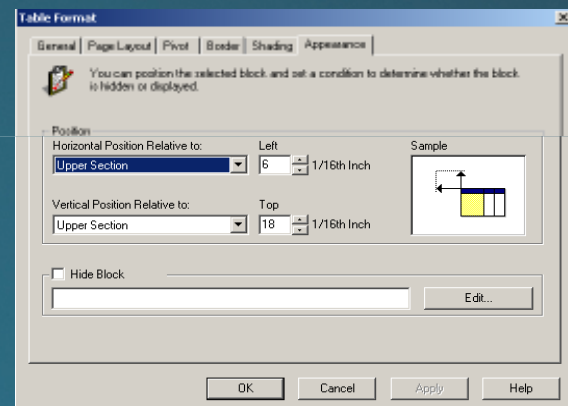
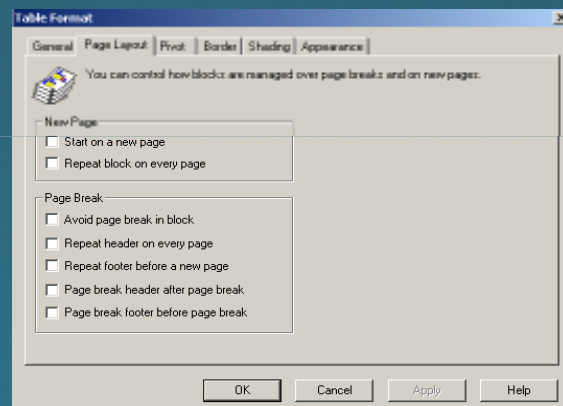
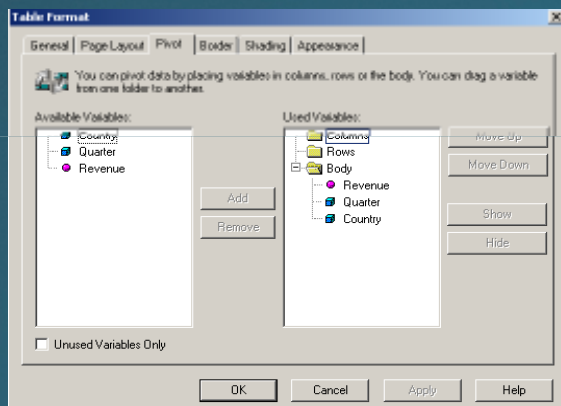
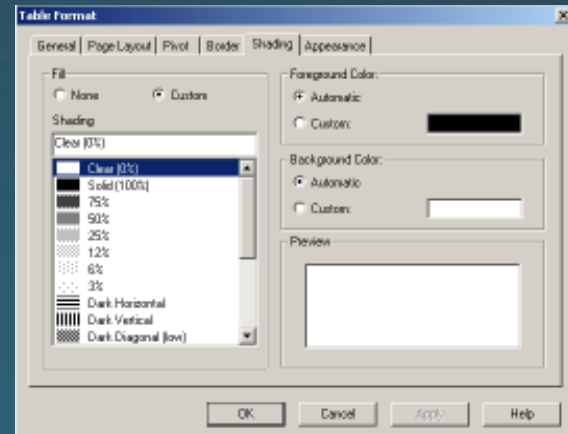
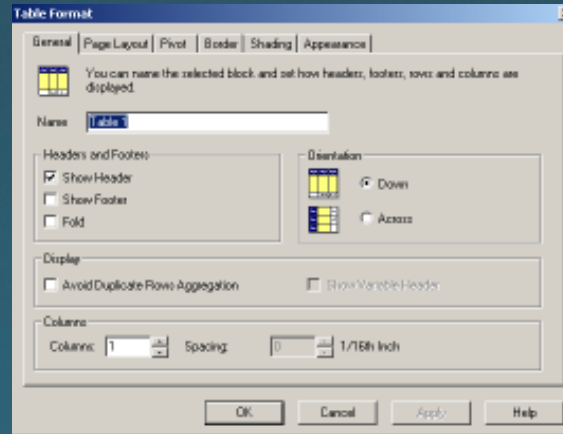
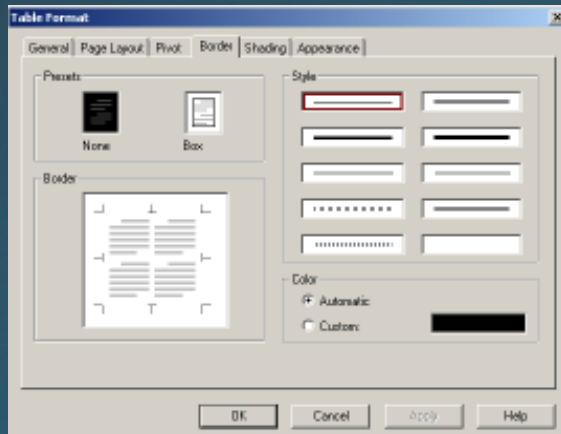
	Q1	Q2	Q2-Q1	Q3	Q4	Q4-Q3
France	208,565.00	242,165.00	16.11 %	226,125.00	158,565.00	-29.88 %
US	582,051.00	579,652.00	-0.41 %	658,572.00	630,829.00	-4.21 %

Formatting



Right click
Format Table gives this window





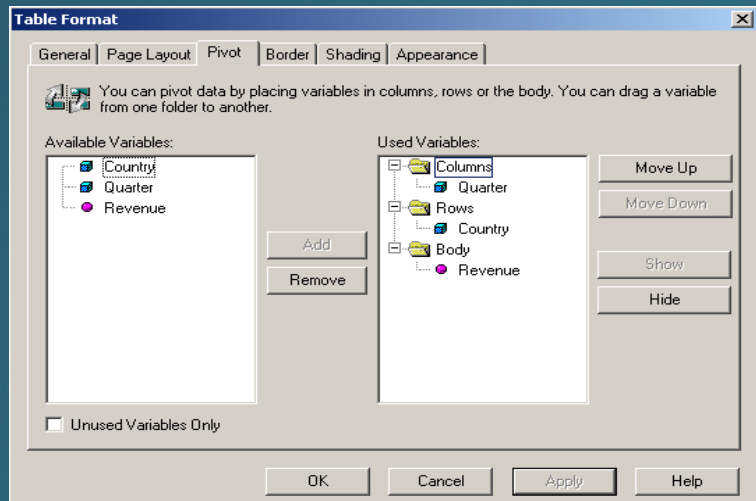
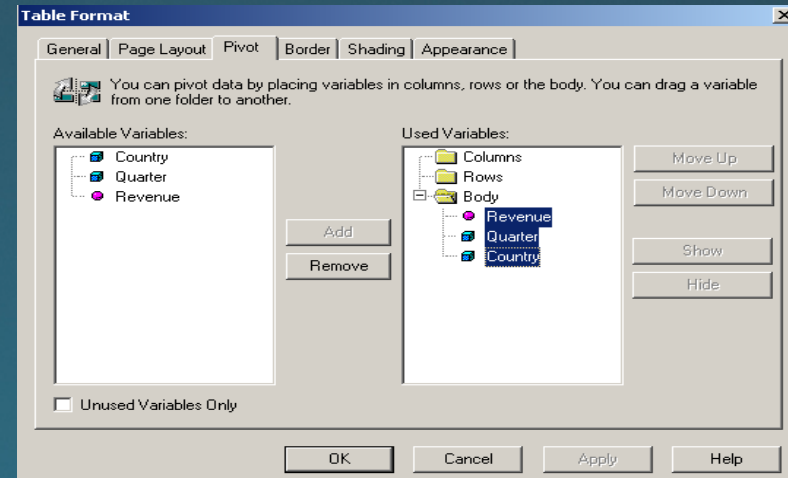
All these tabs are used to Format the table as required



BY Ganesh Jegannathan, Mphasis an HP Company

Cross Tab

Revenue	Quarter	Country
208,565.00	Q1	France
582,051.00	Q1	US
242,165.00	Q2	France
579,652.00	Q2	US
226,125.00	Q3	France
658,572.00	Q3	US
158,565.00	Q4	France
630,829.00	Q4	US



A table like this can be changed into Cross Tab by changing pivot and assigning values to row column and body

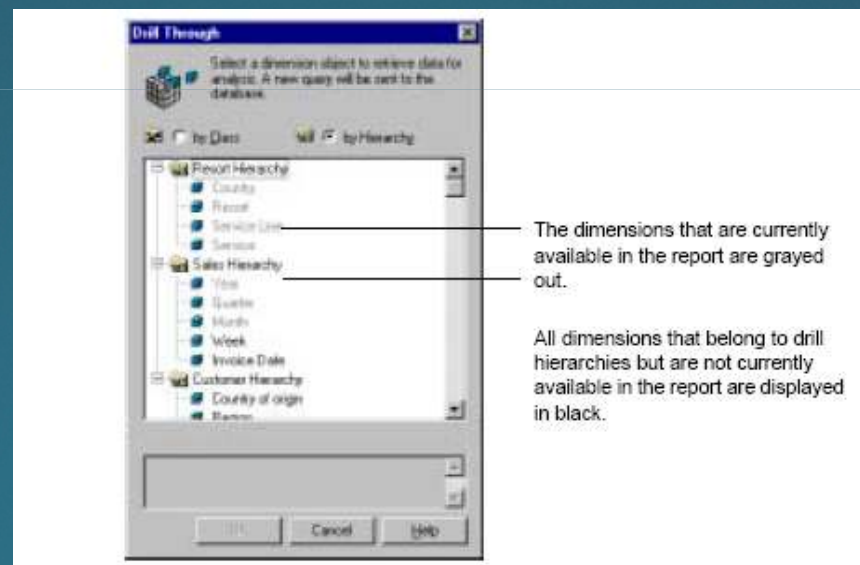
	France	US
Q1	208,565.00	582,051.00
Q2	242,165.00	579,652.00
Q3	226,125.00	658,572.00
Q4	158,565.00	630,829.00



Drilling through to the database to bring in new data

If the lowest level of detail you need is not currently available in the report, you can drill through to the database directly from drill mode and get the data you need

If the lowest level of detail you need is not currently available in the report, you can drill through to the database directly from drill mode and get the data you need.

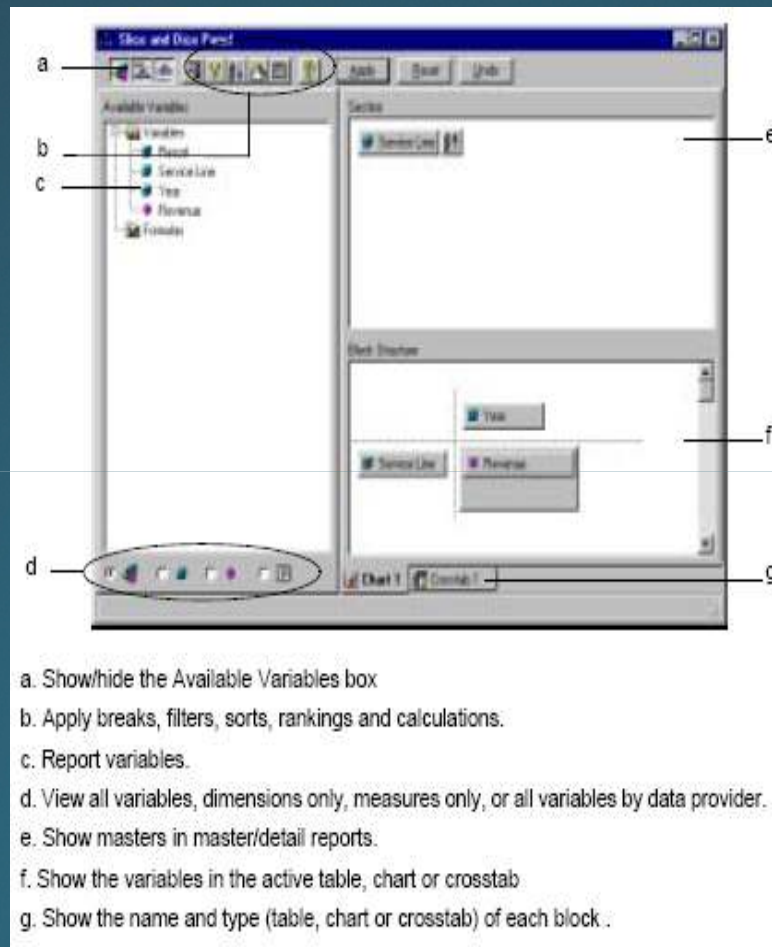


Drilling using custom hierarchies

- You may find that the default order in which dimension objects are arranged in hierarchies is not optimal for your analysis needs, or that you need to drill on a hierarchy that has objects from different classes
- You can edit a hierarchy by changing the order of the dimensions it contains, by adding dimensions to it, and removing dimensions from it. You can also rename a hierarchy, and even delete it.



SLICE AND DICE



Slice and Dice

Slice-and-dice mode enables you to switch the position of data in a report, for example by moving columns to rows to create a crosstab.

You can also use slice-and-dice mode to:

- work with master/detail reports
- display and remove data
- rename, reset and delete blocks
- turn tables and crosstabs into charts, and vice versa
- apply, edit and delete breaks, filters, sorts, rankings and calculations



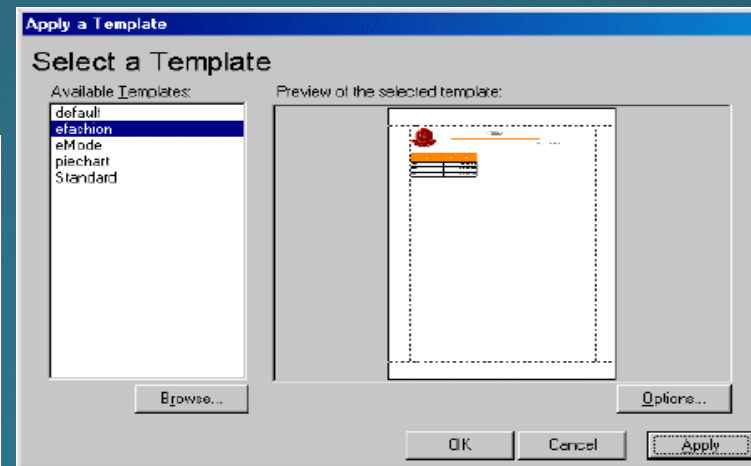
Templates

A template is a special Business Objects document that contains pre-defined styles and a structure that you use as a foundation to create reports. Business Objects installs several templates for you to use and you can also create your own. A standard report does not contain information on page setup or include custom elements such as graphics

Saving a Template

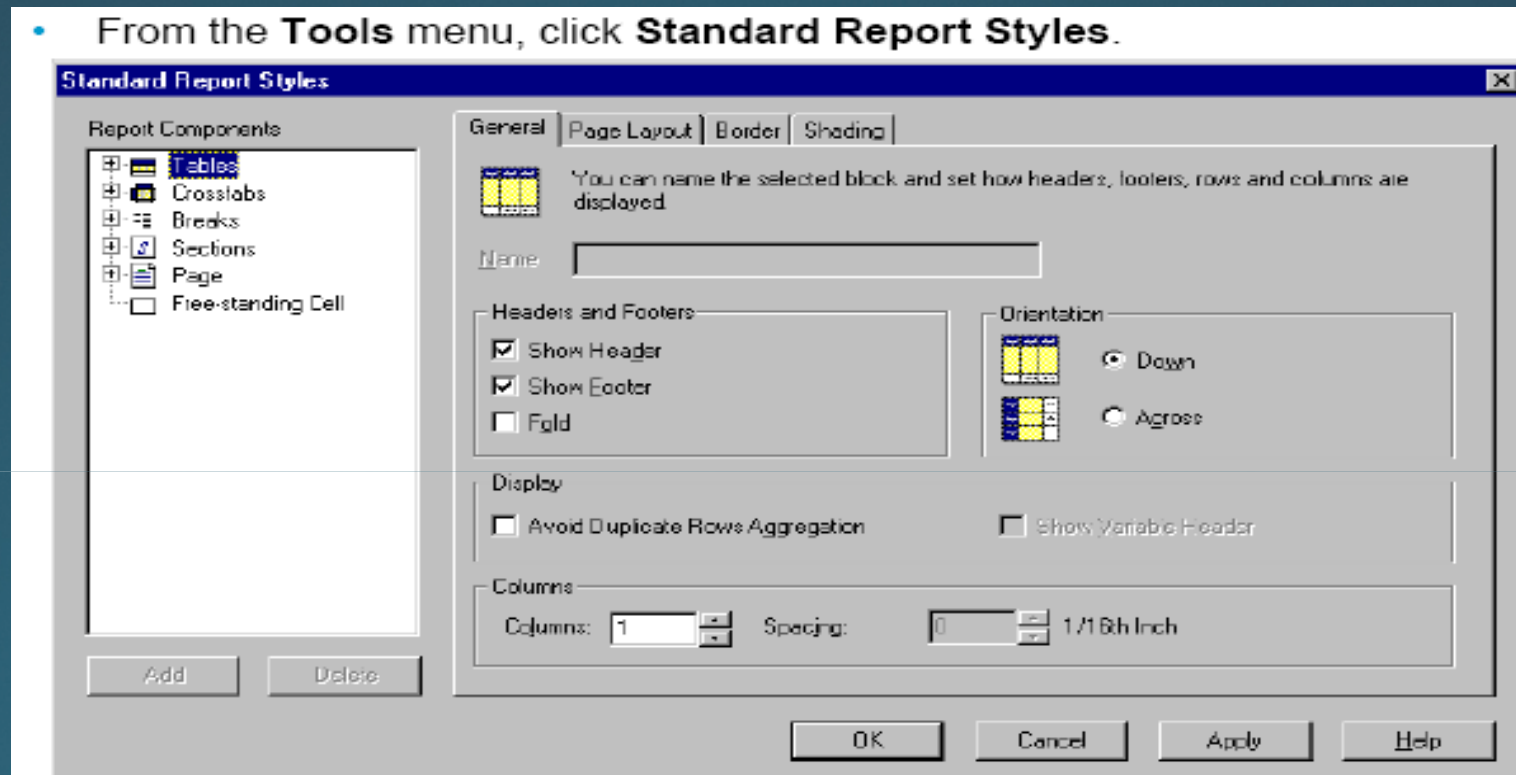
When you create a new document in Business Objects, you choose in what type of report layout your data will be displayed. You can either generate a standard report or generate a report from a set of templates.

Applying a template



Customizing standard report styles

- From the **Tools** menu, click **Standard Report Styles**.



Editing settings and applying to an open report

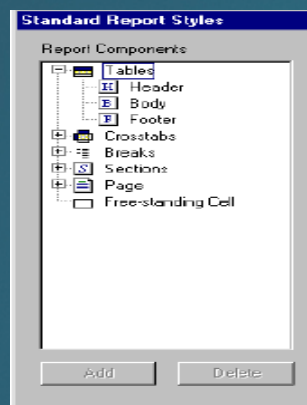
Tables : You can edit the default formatting and page layout options for tables.

Cross tabs : You can edit the default formatting and page layout options for cross tabs.

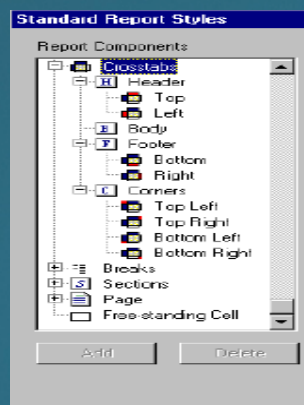
Breaks : You can edit the settings for breaks you insert in tables and cross tabs. You can define up to nine different break levels with different attributes set for each level.

Sections : You can set different attributes for up to nine different sections in a report. You can set attributes for the cells that are displayed at the top of each section and that contain the master value and for the background shading of the section.

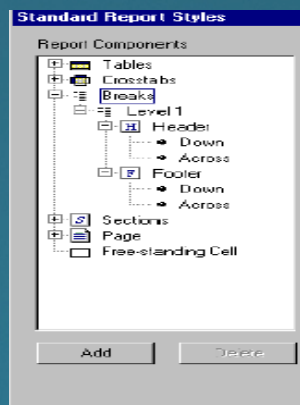
Tables



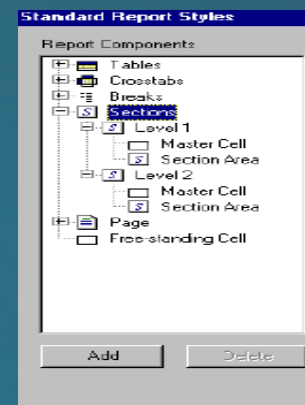
Cross tabs



Breaks



Sections

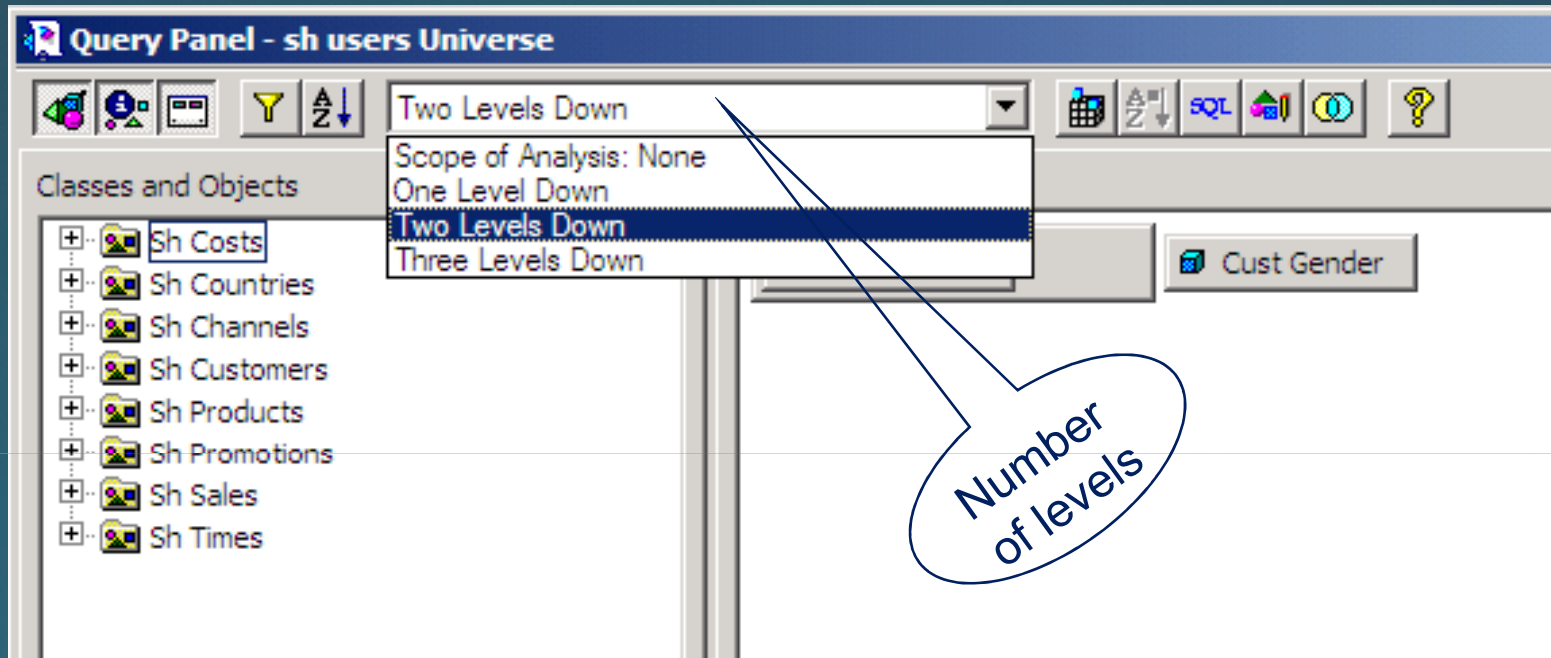


Defining the scope of analysis

- Analysis means looking at data from different viewpoints and on different levels of detail. In reports, you can slice and dice data, which means restructuring a report to analyze the data from a new perspective. You can also work in drill mode, which enables you to display data in progressively greater detail.
- Scope of analysis means a subset of data, returned by a query, that you will use for analysis in your report. The data for your scope of analysis does not appear in the report until you decide that you want to use it in analysis.
- The scope of analysis you can define depends on hierarchies in the universe. A hierarchy, which the designer sets up when creating the universe, consists of dimension objects ranked from less detailed to more detailed. For example, the Sales hierarchy in the demonstration universe consists of the following dimension objects: Year, Quarter, Month, Week, Invoice Date. You select dimension objects that belong to hierarchies to define scope of analysis.



Scope of analysis



The data for the objects in your scope of analysis does not appear in the report, but it is available for use in analysis depends on the level the user opted.



Custom's scope of Analysis

The custom's scope of analysis shows the hierarchies and dimensions that you can use in the query.

To include all the dimensions in a hierarchy in your scope of analysis, click the box that appears next to the hierarchy's name, then click OK.

To include only some dimensions from different hierarchies in your scope of analysis, click the box next to each dimension you want to include, then click OK.

If you are ready to run the query, click Run.



Setting where clause

The screenshot shows the 'Query Panel - sh users Universe' interface. At the top, there is a toolbar with various icons and a 'Scope of Analysis: None' dropdown. The main area is divided into three sections:

- Operators:** A list of 18 operators with radio buttons. 'Equal to' is selected and highlighted in blue. The operators are: Equal to, Different from, Greater than, Greater than or equal to, Less than, Less than or equal to, Between, Not between, In list, Not in list, Is null, Is not null, Matches pattern, Different from pattern, Both, and Except.
- Result Objects:** Three buttons are visible: 'Cust First N...', 'Cust Gender', and 'Country Name'.
- Conditions:** A single condition is shown: 'Country Id Equal to <Select an operand type >'. The 'Equal to' operator is highlighted in blue.

A blue oval callout points to the 'Operators' list with the following text:

In the left side, the user can view the available conditions. The selected condition will be applied to the condition column. Then select the operand type.



Query Panel - sh users Universe

Scope of Analysis: None

Operands

- Type a new constant
- Show list of values
- Type a new prompt
- Show list of prompts
- Select an object
- Create a subquery (ANY)
- Calculation

Result Objects

- Cust First N...
- Country Name

Conditions

- Country Id Equal to 'AU'

List of Values of Country Id

Tabular View Hierarchical View

Country Id
AR
AU
BR
DE
DK
ES
FR
IE
IN
JP
MY
NL
NZ
PL
SA
TR
UK

Show Selected Only

In the left, the user select the show list of values.

Then select the values from the list of values box.

In the condition box we can see the selected values as where clause.



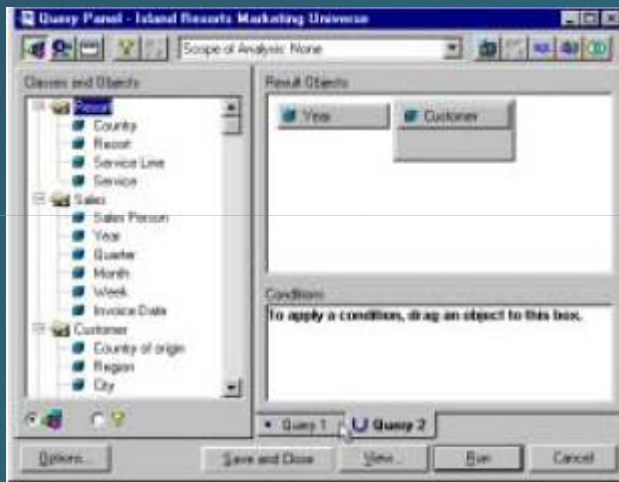
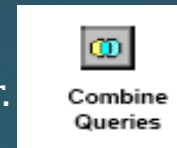
Writing Combined Queries

- To build two or more queries in the query panel .
- To build a combined query, query panel used the following set operators.
 - Union
 - Intersect
 - Minus
- Queries that you combine must contain the same number of objects in order to return the same number of columns of data.
- When you build a combined query to return data from more than one object in one column, you must use objects of the same type (character, date or number).



COMBINED QUERY

In the Query Panel, build a query.
Click the Combine Queries button on the Query Panel toolbar.
The existing query appears in the Query 1 tab.
A second tab, Query 2, also appears and is now active:



If you want to use a different operator, click the Query 2 tab with your right mouse button, then click the operator you want on the shortcut menu

UNION:

- Combine the data from two objects in a single column in the report.
- UNION is especially useful for working with incompatible objects.



INTERSECT:

Obtain data common to two sets of results

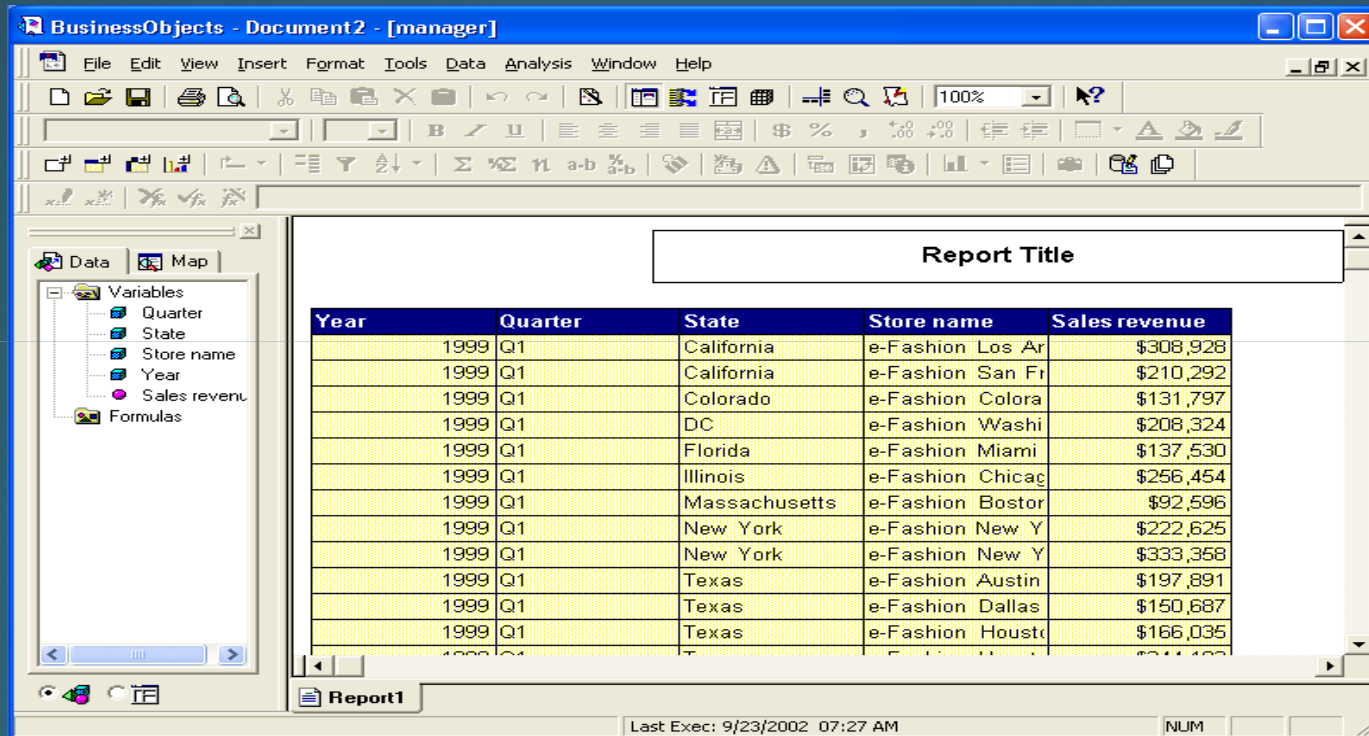
MINUS:

Exclude the results of one query from the main query result.



Creating Multiple Queries in one Document

Multiple Queries from one Universe, Assuming that we have the following report



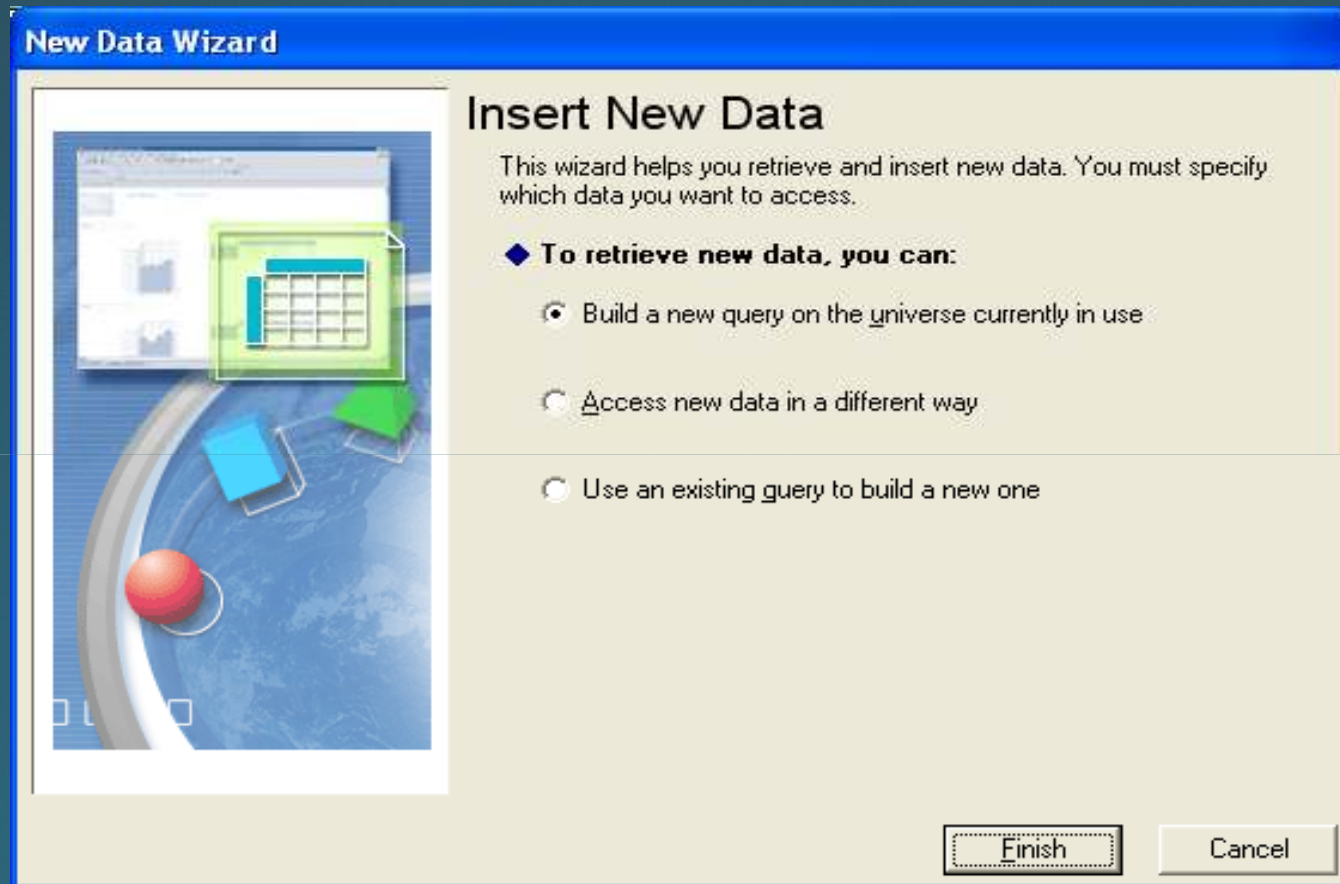
The screenshot shows the BusinessObjects interface with a report titled 'Report1'. The report contains a table with the following data:

Year	Quarter	State	Store name	Sales revenue
1999	Q1	California	e-Fashion Los Ar	\$308,928
1999	Q1	California	e-Fashion San Fr	\$210,292
1999	Q1	Colorado	e-Fashion Colora	\$131,797
1999	Q1	DC	e-Fashion Washi	\$208,324
1999	Q1	Florida	e-Fashion Miami	\$137,530
1999	Q1	Illinois	e-Fashion Chicag	\$256,454
1999	Q1	Massachusetts	e-Fashion Bostor	\$92,596
1999	Q1	New York	e-Fashion New Y	\$222,625
1999	Q1	New York	e-Fashion New Y	\$333,358
1999	Q1	Texas	e-Fashion Austin	\$197,891
1999	Q1	Texas	e-Fashion Dallas	\$150,687
1999	Q1	Texas	e-Fashion Houstr	\$166,035

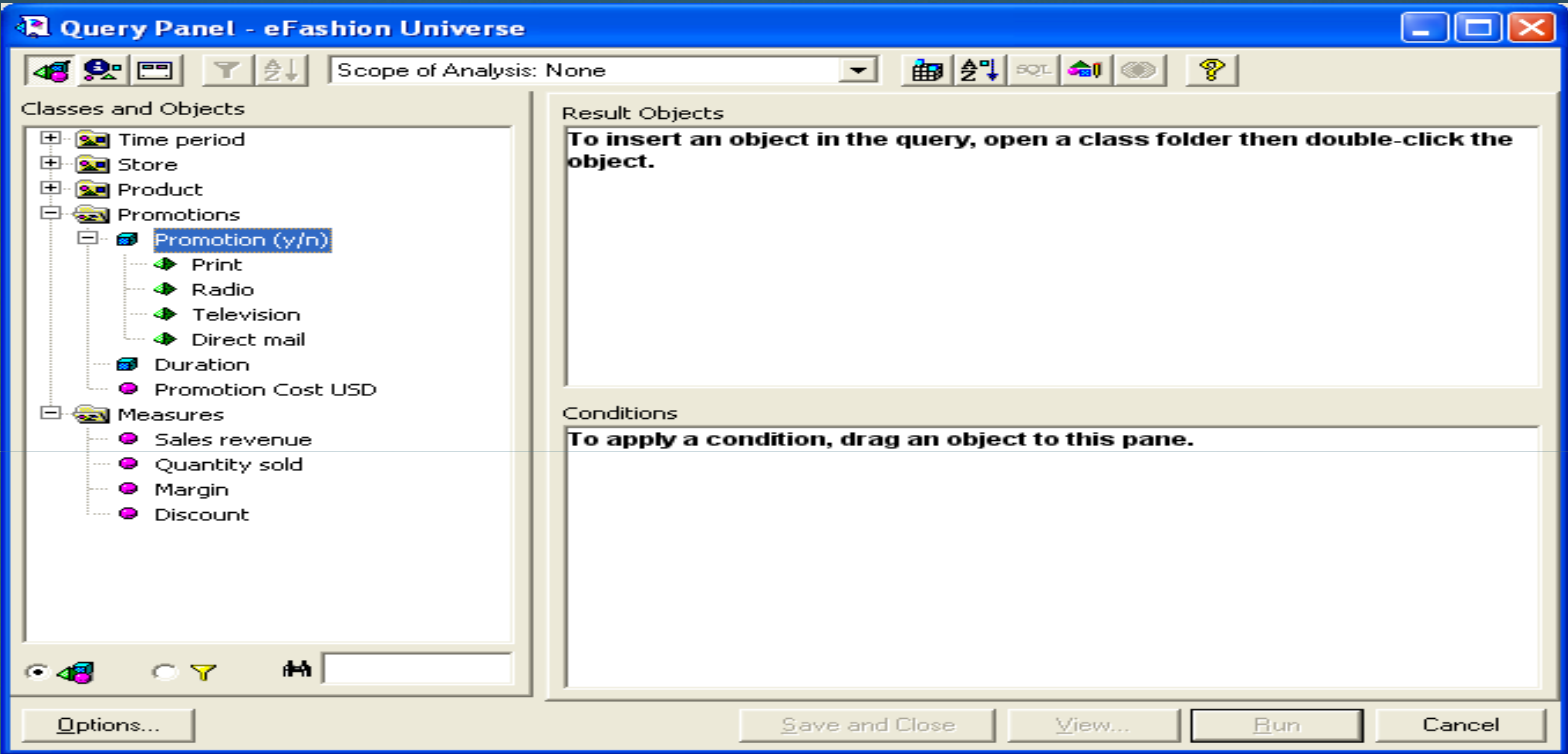


BY Ganesh Jegannathan, Mphasis an HP Company

We want to have a query on the existing universe we need to go to the menu Data and click on New Data Provider we get the following options

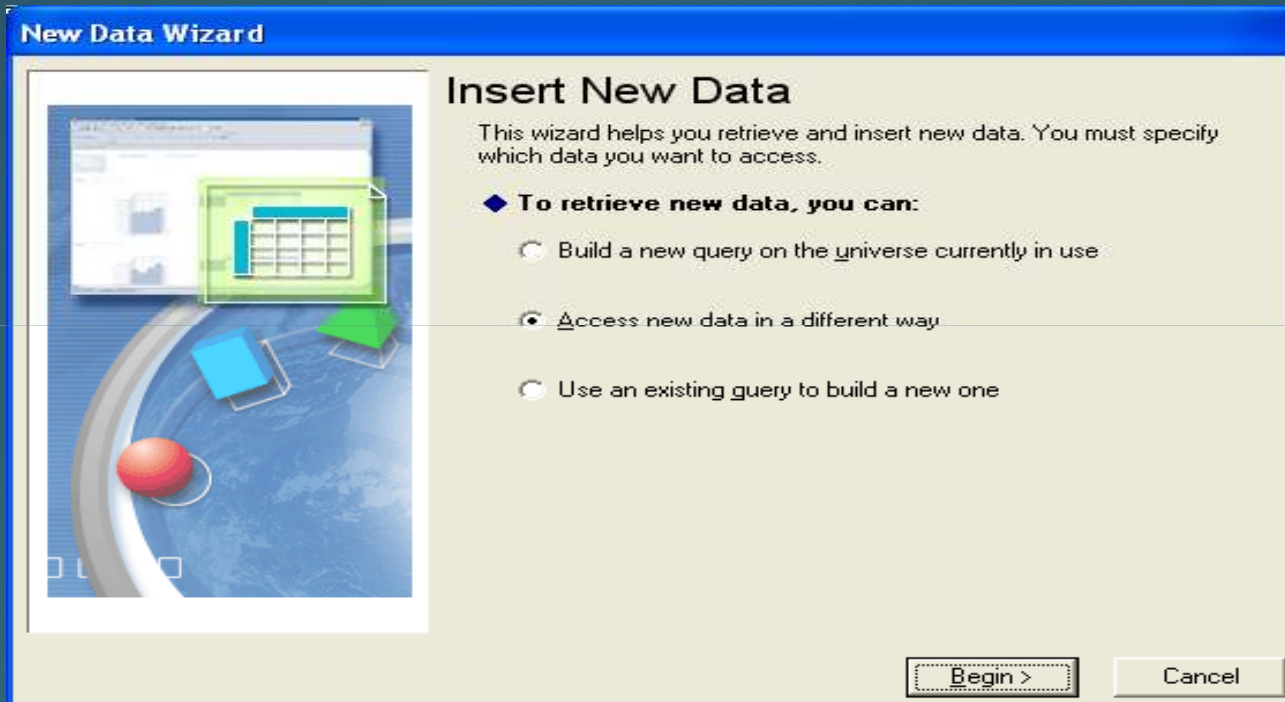


By choosing “Build a new query on the universe currently in use.” we get the following option and create a new on the existing universe

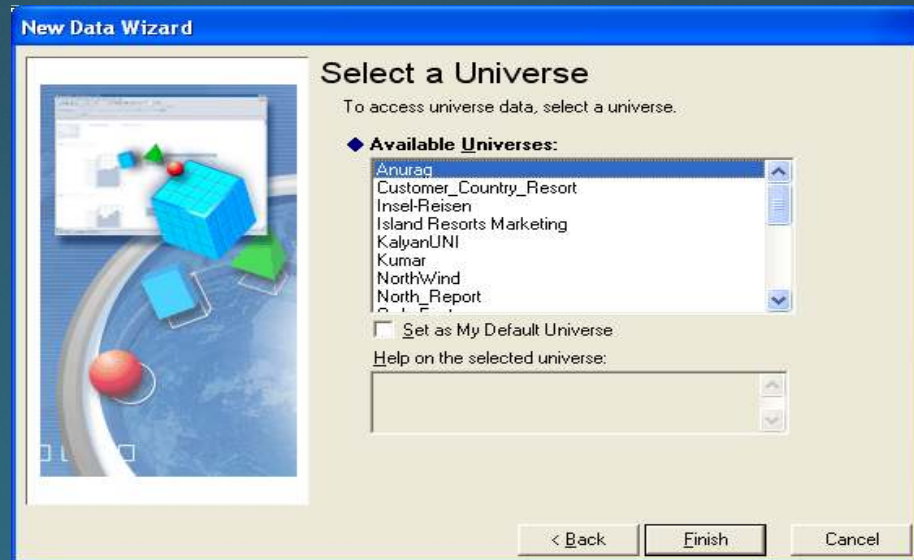


Queries using multiple universes

By choosing Access new data in a different way we can access data from different Universe



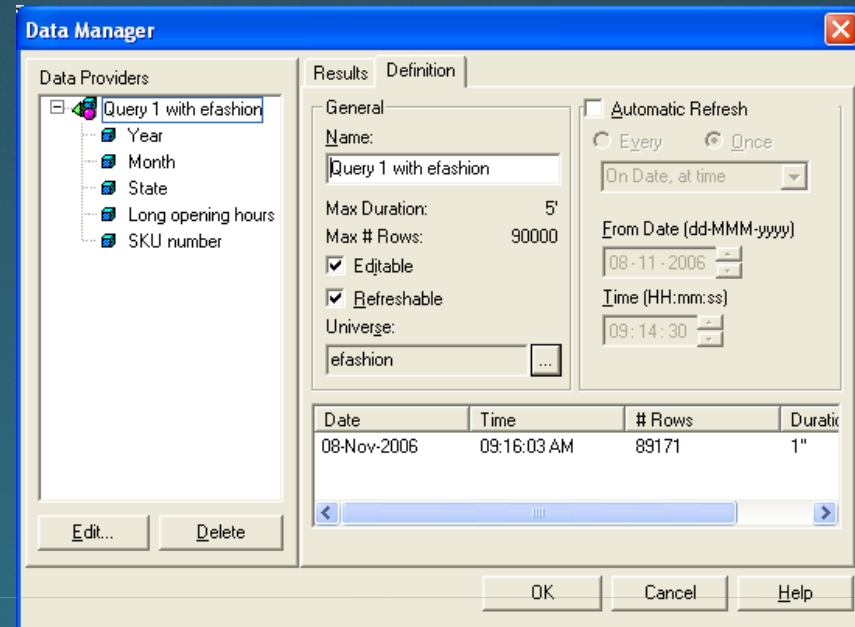
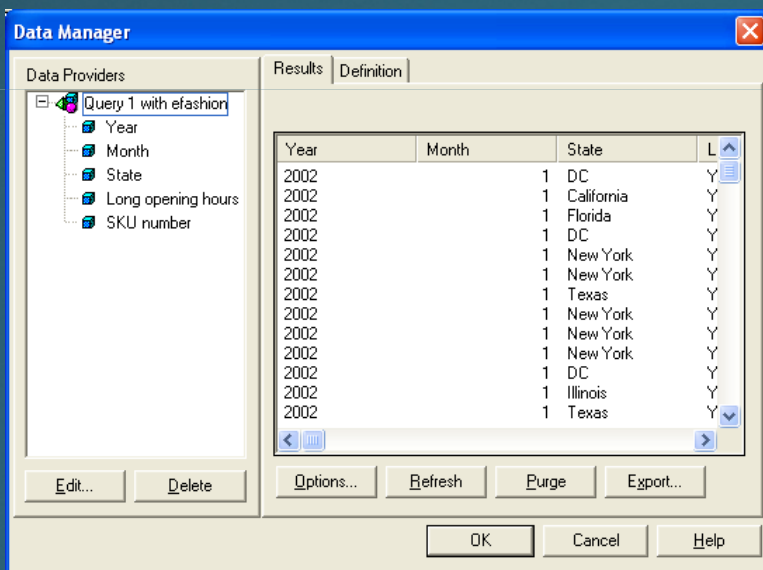
Next



By clicking on next we can select new universe from Available Universes



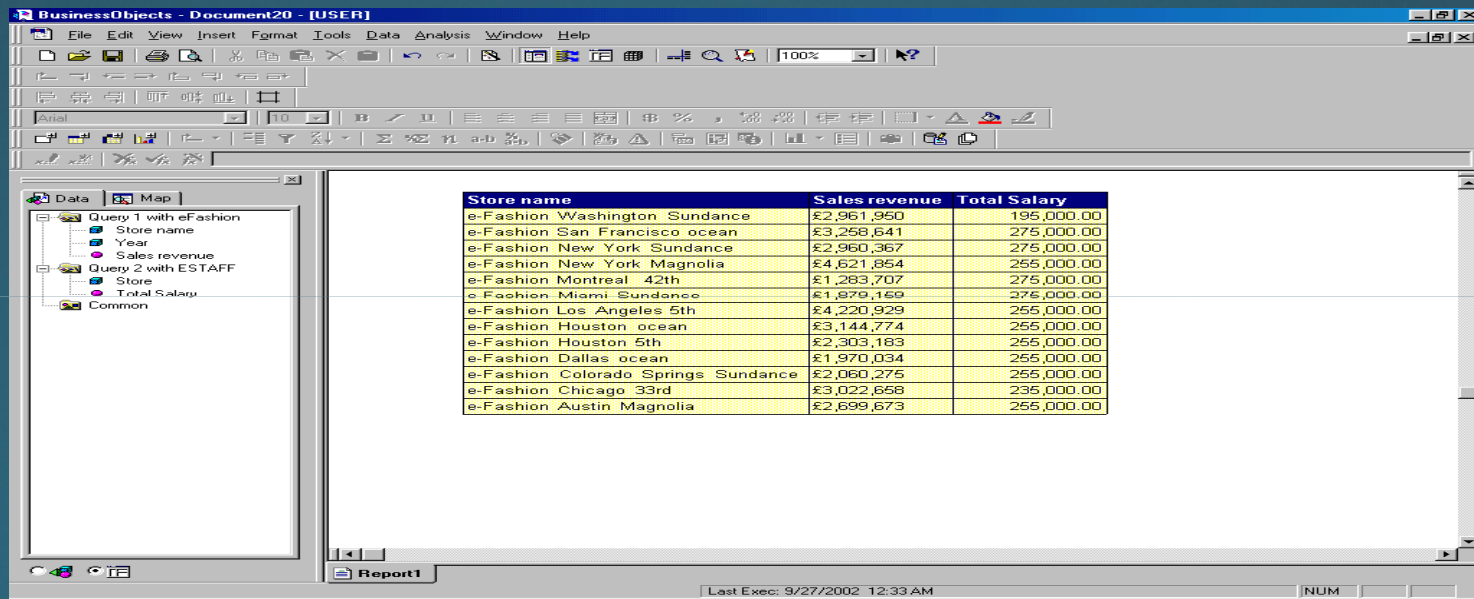
Naming a Data Provider: Go To Data Menu and click on View Data. You can see the following screen



Go to Definition Tab and give a name to Data Provider in General



- Linking is performed in the Data Manager
- Highlight one of the objects to be linked and press the link button
- Select the other object to be linked in the new window
- Close the Data Manager



Store name	Sales revenue	Total Salary
e-Fashion Washington Sundance	£2,961,950	195,000.00
e-Fashion San Francisco ocean	£3,258,641	275,000.00
e-Fashion New York Sundance	£2,960,367	275,000.00
e-Fashion New York Magnolia	£4,621,864	255,000.00
e-Fashion Montreal 42th	£1,263,707	275,000.00
e-Fashion Miami Sundance	£1,879,169	275,000.00
e-Fashion Los Angeles 5th	£4,220,929	255,000.00
e-Fashion Houston ocean	£3,144,774	255,000.00
e-Fashion Houston 5th	£2,303,183	255,000.00
e-Fashion Dallas ocean	£1,870,034	255,000.00
e-Fashion Colorado Springs Sundance	£2,060,275	255,000.00
e-Fashion Chicago 33rd	£3,022,658	235,000.00
e-Fashion Austin Magnolia	£2,699,673	255,000.00



Applying conditions

A condition is a way of limiting the data that a query returns

In Business Objects, you can set three types of conditions on a query:

Predefined conditions

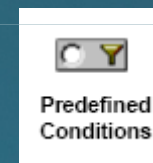
When universe designers build universes, they can create predefined conditions to use.

Simple conditions

Enable you to limit data returned by a result object.

Complex conditions

Enable you to limit the query results by any object in the universe.



To apply a predefined condition

1. Click **Predefined Conditions** below the Classes and Objects box in the Query Panel. The Predefined Conditions box replaces the Classes and Objects box.
2. Double-click the predefined condition you want to apply. The condition appears in the Conditions box. When you run the query, only the data corresponding to the predefined condition appears in the report.



To apply your own simple condition

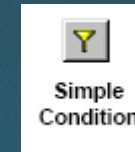
Before you can apply a simple condition on an object, you must include the object in the query.

Then:

1. Click the object icon in the Result Objects box.
2. Click the **Simple Condition** button on the toolbar.

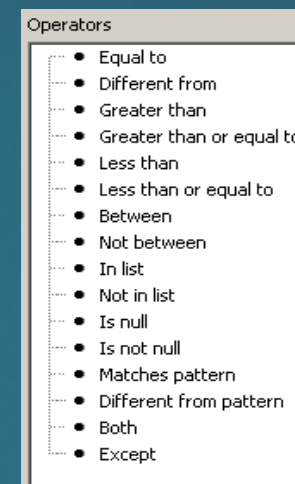
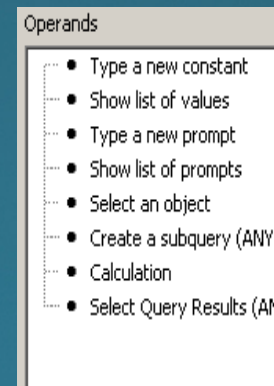
The list of values for the object is retrieved from the database, and appears in the List of Values dialog box:

3. Hold down the Ctrl key on your keyboard, click the values you want the object to retrieve, then click **OK**. The condition appears in the Conditions box. When you run the query, only the data corresponding to the value's you selected will appear in the report.



Applying complex conditions on queries

You can limit the data that queries return by applying conditions. You apply complex conditions by combining an object with an operator (for example greater than), and an operand (for example, values that you type, or another object).



User objects in Reports

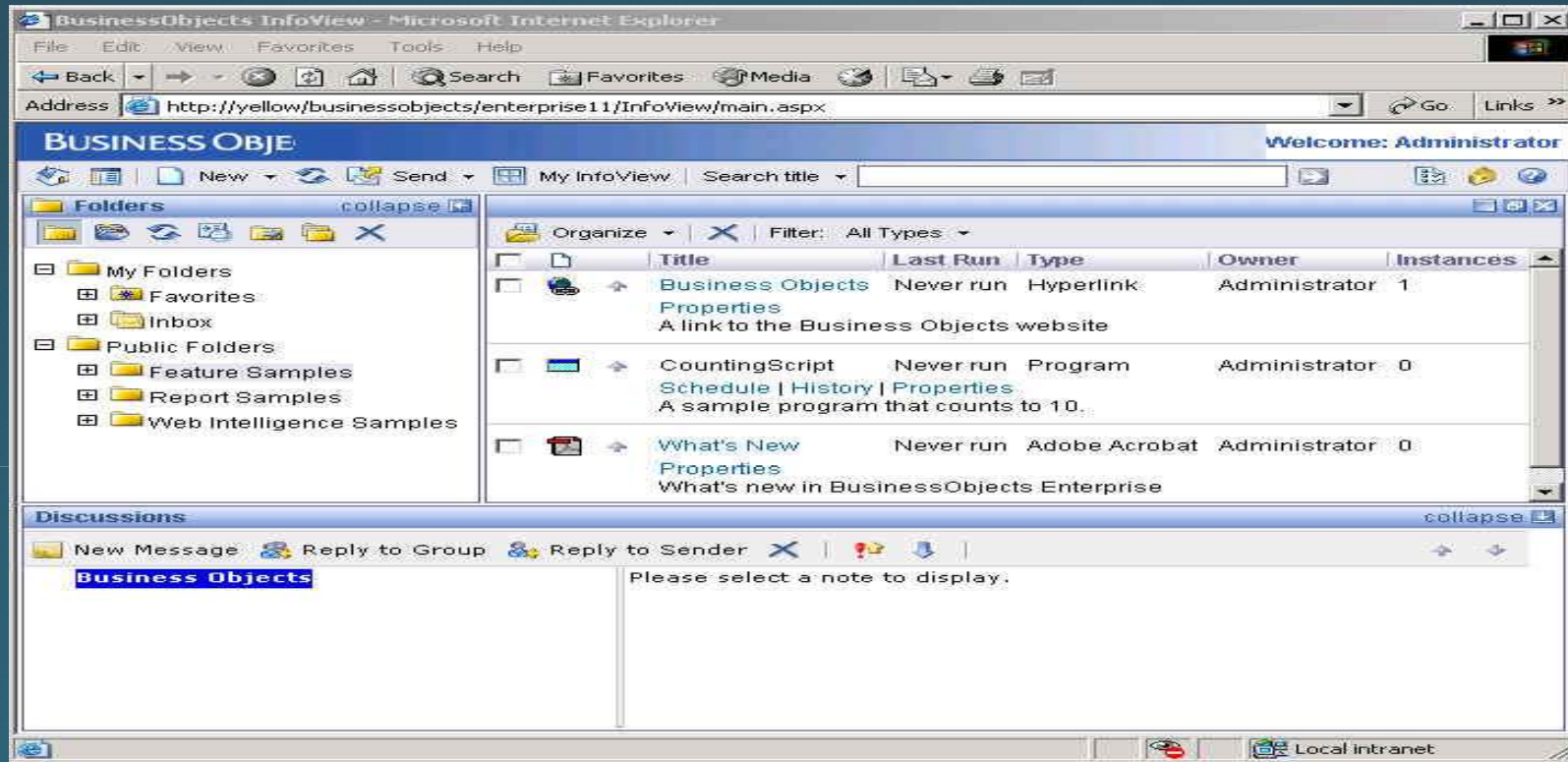
- A universe consists primarily of classes and objects created by the universe designer.
- If the objects in a universe do not meet your needs, you can create your own additional user objects.
- user objects enable you to:
 - Make additional calculations beyond those provided by the base universe objects.
 - Apply functions to text, for example to capitalize data.
 - Group data.

User object consist of?

- A user object has a name, a type (character, date or numeric), a qualification
- (dimension, measure or detail) and a formula. The formula contains a combination of functions, objects, user objects, operators, and text.

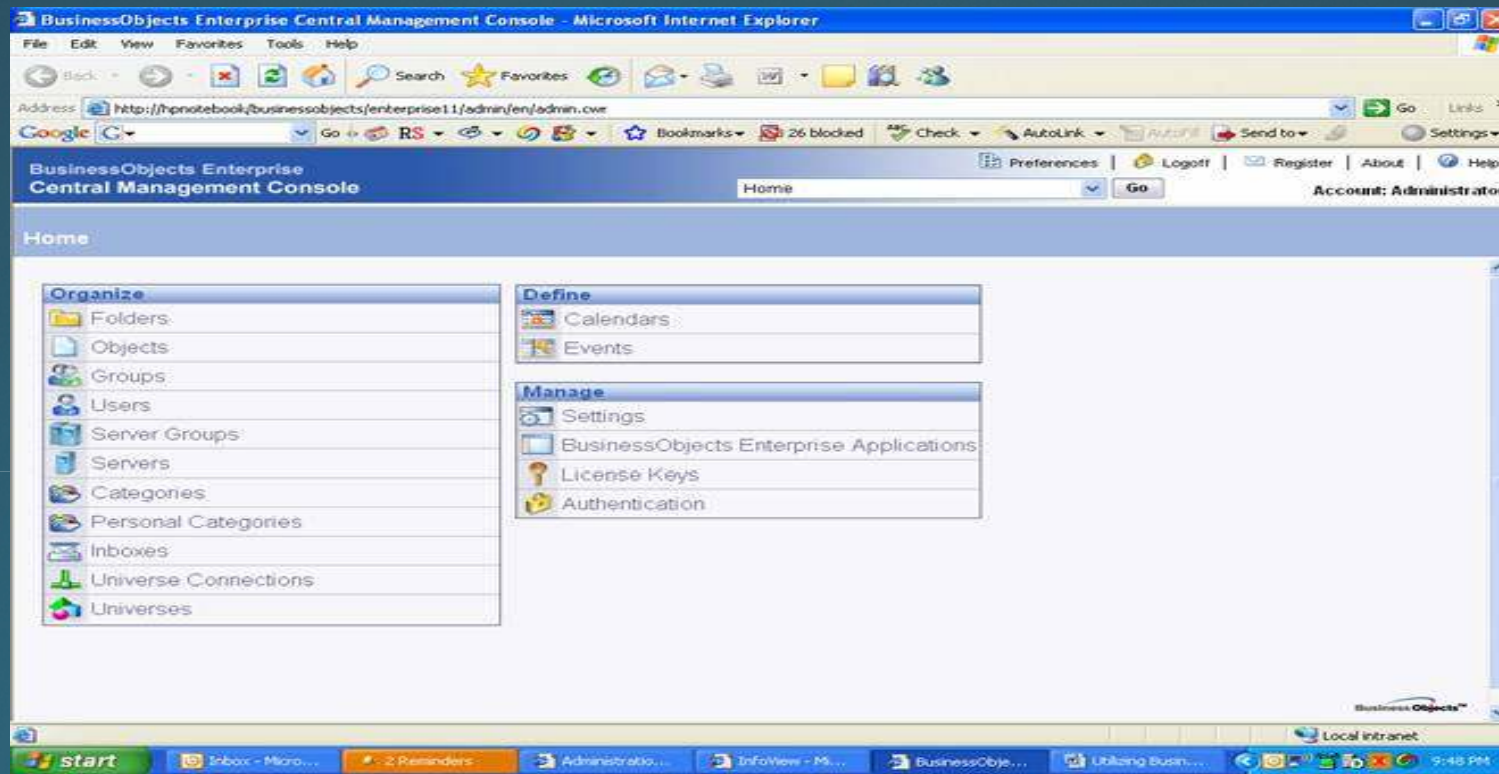


Infoview



BY Ganesh Jegannathan, Mphasis an HP Company

Central Management Console



BY Ganesh Jegannathan, Mphasis an HP Company

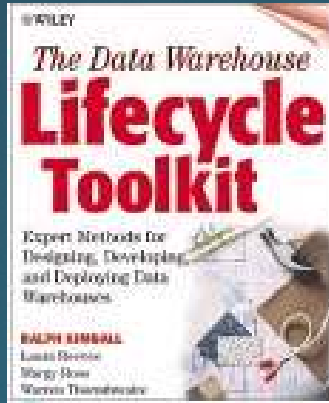


Central Configuration Manager

1. Web Application Server
2. Web component Adapter
3. Central Management server
4. Event Server
5. Destination Job server
6. IFRS
7. OFRS
8. Crystal Report Cache Server
9. Desktop Intelligence Cache Server
10. Program job Server
11. Crystal Report job Server
12. Web Intelligence job Server
13. Web intelligence report Server
14. LOV job server
15. Desktop intelligence job server
16. Desktop intelligence report server
17. Connection Server
18. Crystal report page server
19. Web intelligence report server
20. Report Application server



Reference Material



The Data Warehouse Lifecycle toolkit : Expert Methods for Designing, Developing and Deploying Data Warehouse.

Ralph Kimball, Laura Reeves, Margy Ross, Warren Thomthwaite



Building the Data warehouse, 2nd edition

William h, Inmon

