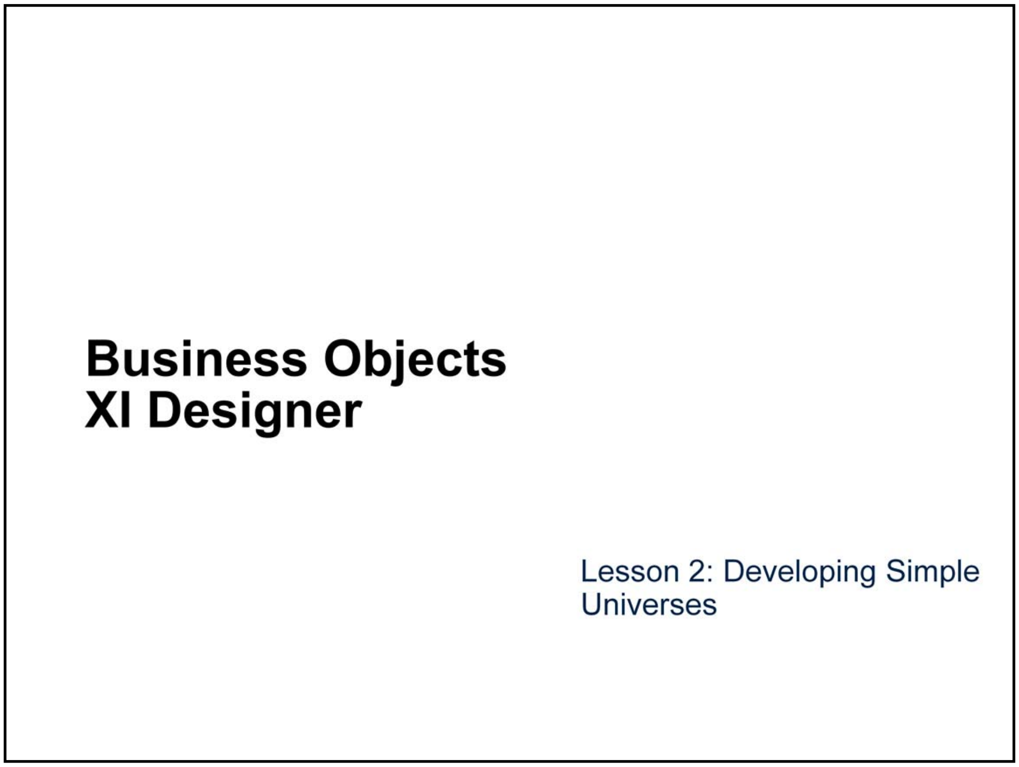
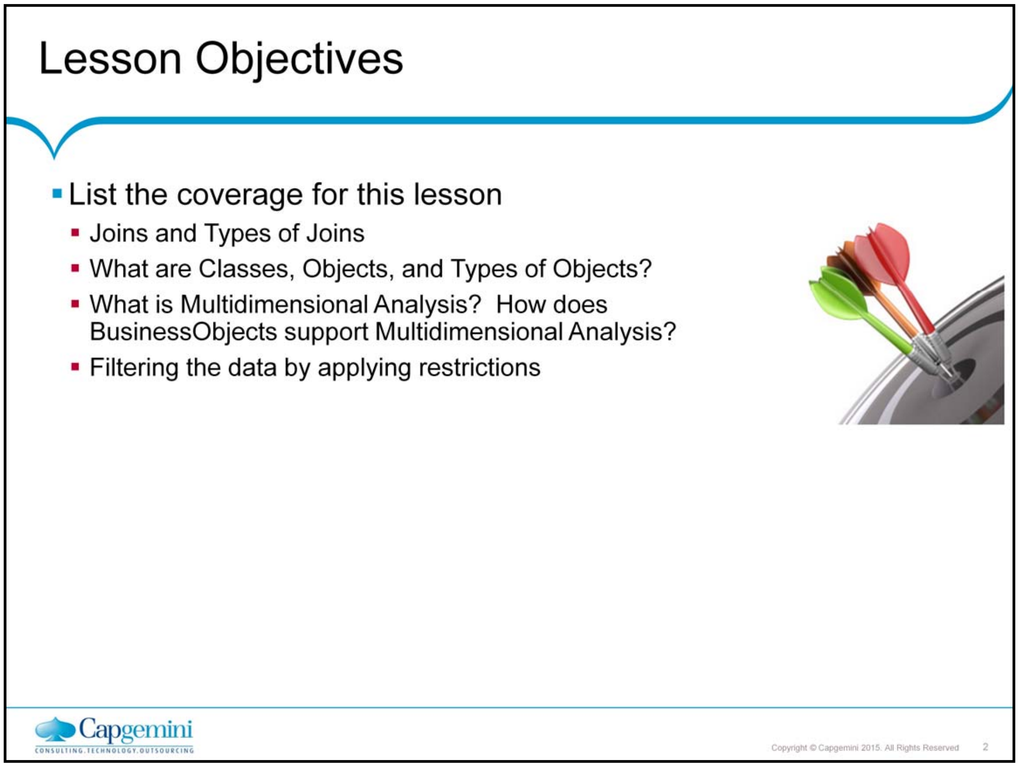


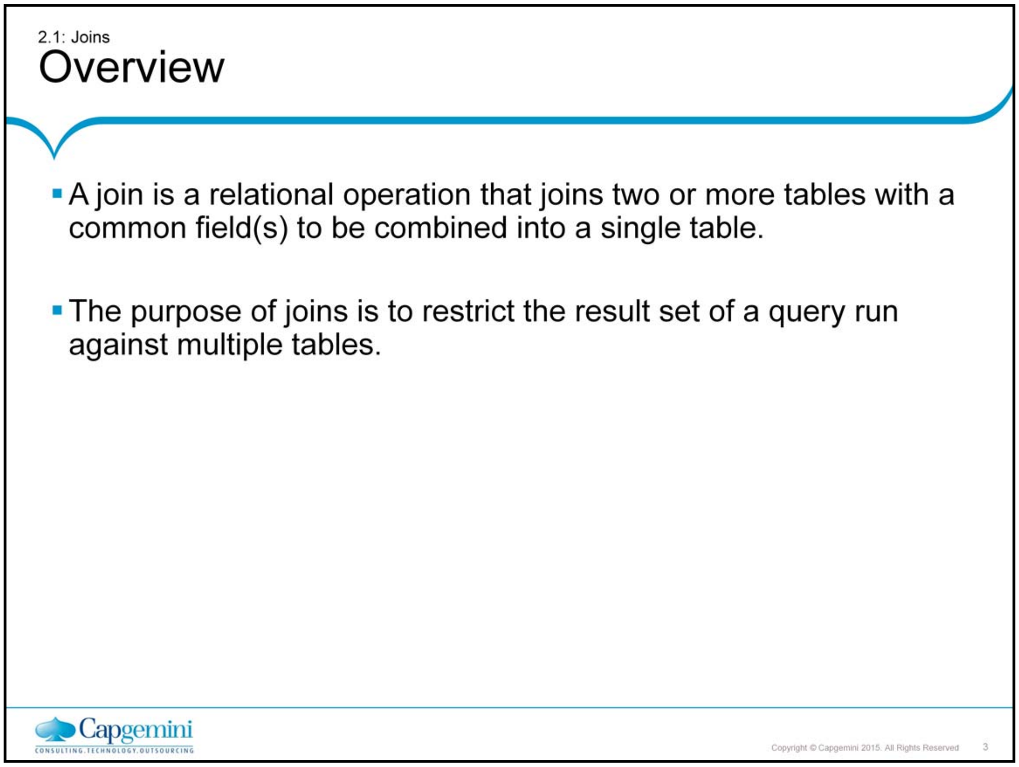
Business Objects XI Designer

Lesson 2: Developing Simple Universes



Lesson Objectives

- List the coverage for this lesson
  - Joins and Types of Joins
  - What are Classes, Objects, and Types of Objects?
  - What is Multidimensional Analysis? How does BusinessObjects support Multidimensional Analysis?
  - Filtering the data by applying restrictions

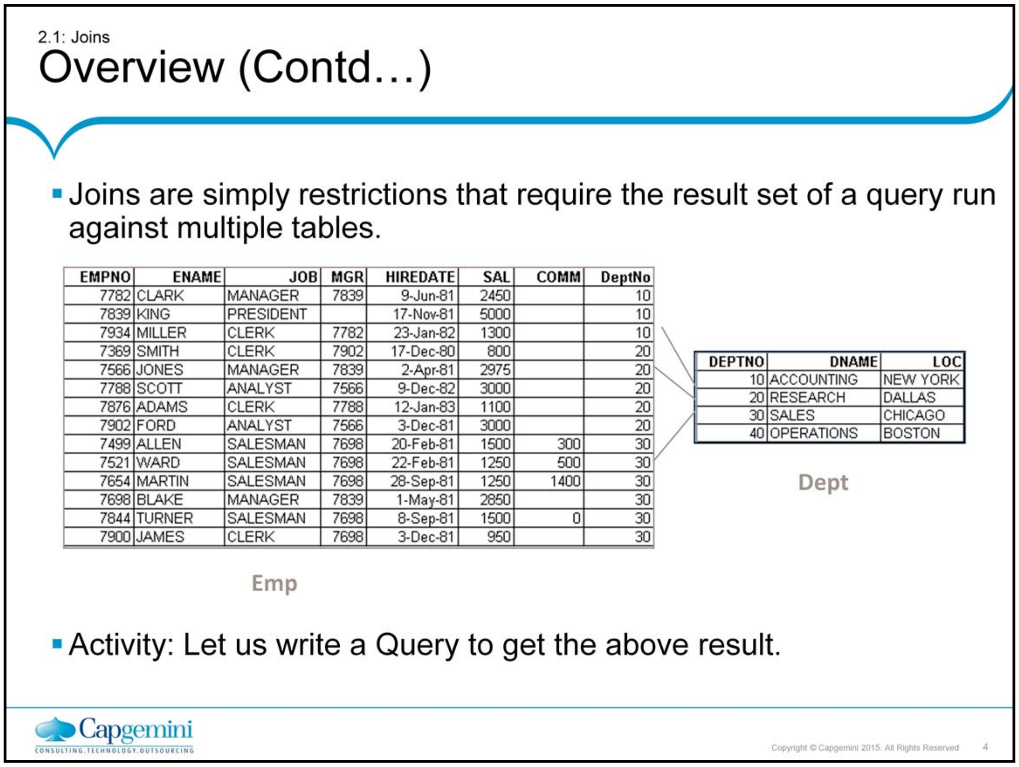


2.1: Joins

Overview

- A join is a relational operation that joins two or more tables with a common field(s) to be combined into a single table.
- The purpose of joins is to restrict the result set of a query run against multiple tables.

3



2.1: Joins

Overview (Contd…)

- Joins are simply restrictions that require the result set of a query run against multiple tables.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DeptNo
7782	CLARK	MANAGER	7839	9-Jun-81	2450		10
7839	KING	PRESIDENT		17-Nov-81	5000		10
7934	MILLER	CLERK	7782	23-Jan-82	1300		10
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7566	JONES	MANAGER	7839	2-Apr-81	2975		20
7788	SCOTT	ANALYST	7566	9-Dec-82	3000		20
7876	ADAMS	CLERK	7788	12-Jan-83	1100		20
7902	FORD	ANALYST	7566	3-Dec-81	3000		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	1500	300	30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7698	BLAKE	MANAGER	7839	1-May-81	2850		30
7844	TURNER	SALESMAN	7698	8-Sep-81	1500	0	30
7900	JAMES	CLERK	7698	3-Dec-81	950		30

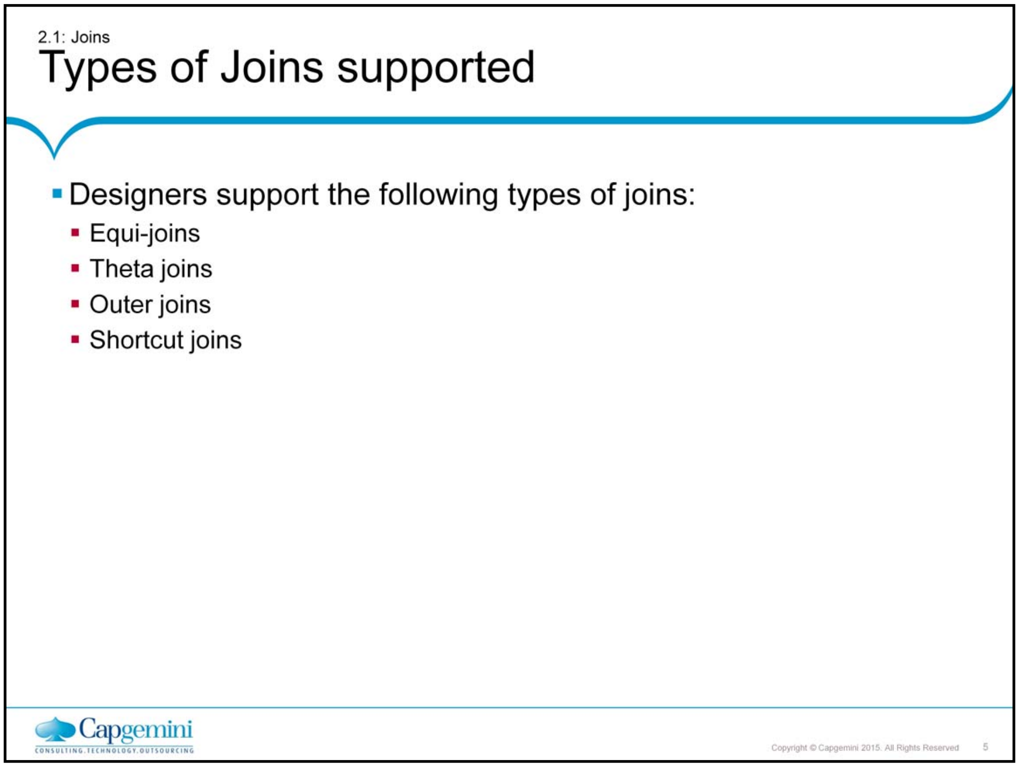
Emp

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Dept

- Activity: Let us write a Query to get the above result.

Add the notes here.



2.1: Joins

Types of Joins supported

- Designers support the following types of joins:
  - Equi-joins
  - Theta joins
  - Outer joins
  - Shortcut joins

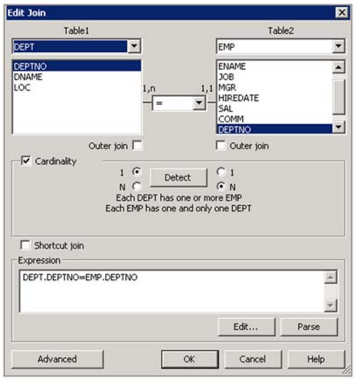
5

Add the notes here.

2.1: Joins

Equi-Join

- An equi-join is used when we want to retrieve matching records only.



Edit Join

Table1: DEPT

Columns: DEPTNO, DNAME, LOC (DEPTNO selected)

Table2: EMP

Columns: ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO (DEPTNO selected)

1,n = 1,1

Outer join □   □ Outer join

☑ Cardinality: 1 ● N ○   ○ 1 ● N

Each DEPT has one or more EMP
Each EMP has one and only one DEPT

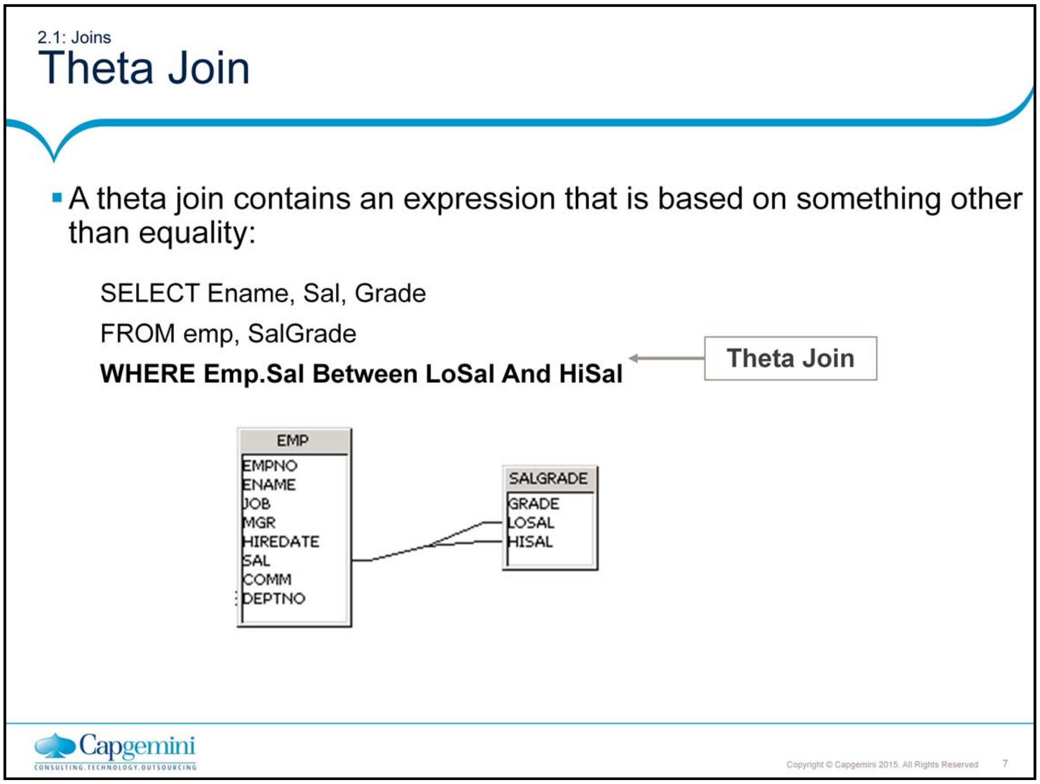
□ Shortcut join

Expression:

```
DEPT.DEPTNO=EMP.DEPTNO
```

- Activity: Let us write a Query to get the above result.

Add the notes here.



2.1: Joins

Theta Join

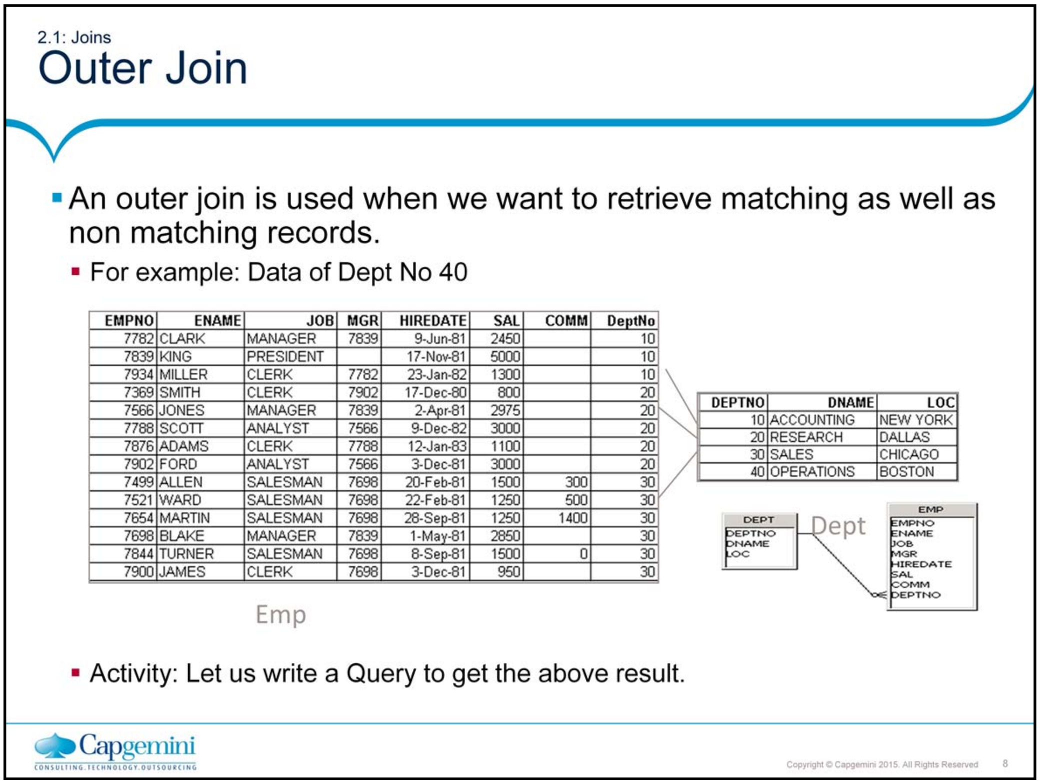
- A theta join contains an expression that is based on something other than equality:

```
SELECT Ename, Sal, Grade
FROM emp, SalGrade
WHERE Emp.Sal Between LoSal And HiSal
```

Theta Join

Theta Join:
A theta join links tables that have a relationship based on something other than equality. For example, a theta join may be used when some form of range relationship needs to be expressed in the join restriction to explain the relationship between data held in one table and that in another table.

For example: All Employees have a Salary associated with them. However, these employees can be categorized by the Salary ranges held in the SalGrade table. The problem arises in how to specify the join information between these two tables in order to make sure that each Employee is associated with their appropriate Salary range. A theta join helps in maintaining such relationships.



2.1: Joins

Outer Join

- An outer join is used when we want to retrieve matching as well as non matching records.
  - For example: Data of Dept No 40

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DeptNo
7782	CLARK	MANAGER	7839	9-Jun-81	2450		10
7839	KING	PRESIDENT		17-Nov-81	5000		10
7934	MILLER	CLERK	7782	23-Jan-82	1300		10
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7566	JONES	MANAGER	7839	2-Apr-81	2975		20
7788	SCOTT	ANALYST	7566	9-Dec-82	3000		20
7876	ADAMS	CLERK	7788	12-Jan-83	1100		20
7902	FORD	ANALYST	7566	3-Dec-81	3000		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	1500	300	30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7698	BLAKE	MANAGER	7839	1-May-81	2850		30
7844	TURNER	SALESMAN	7698	8-Sep-81	1500	0	30
7900	JAMES	CLERK	7698	3-Dec-81	950		30

Emp

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Dept

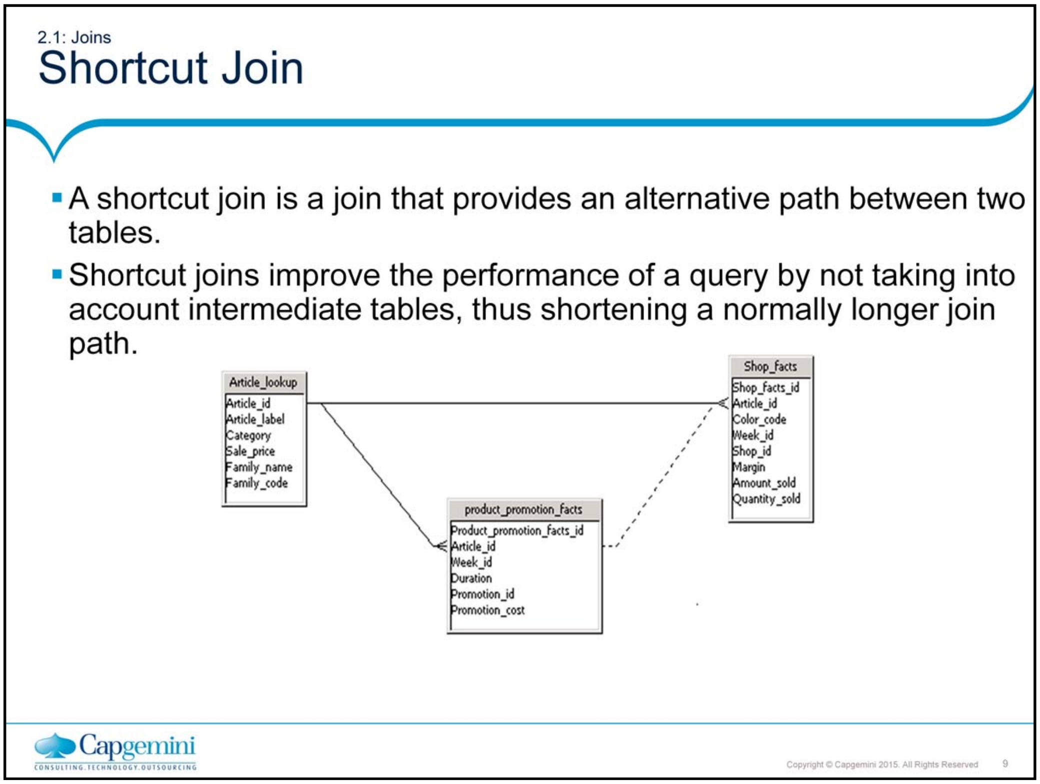
  - Activity: Let us write a Query to get the above result.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

8

<u>Outer Join</u>:

- An outer join is a slight modification of the equi-join. It can be used in situations where the data in one table does not have the corresponding data in the secondary table and a report requirement is to fetch the matching as well as non matching records from one table.
- This situation can be best explained through an example:
  - ➢ Suppose an Analyst is looking at data that comes from two tables, namely Department Name from Dept table and Employee Name from Emp table.
  - ➢ The report requirement is to provide the details of all departments along with the list of employees working under these department.
  - ➢ In this case, to get a clear picture, we need the names of all departments irrespective of whether employees are working under them or not.
  - ➢ Sometimes the requirement can be, “I want to see all those departments where no employee is working (for example: newly formed departments)”.
  - ➢ In both cases we need to have an outer join.



2.1: Joins

Shortcut Join

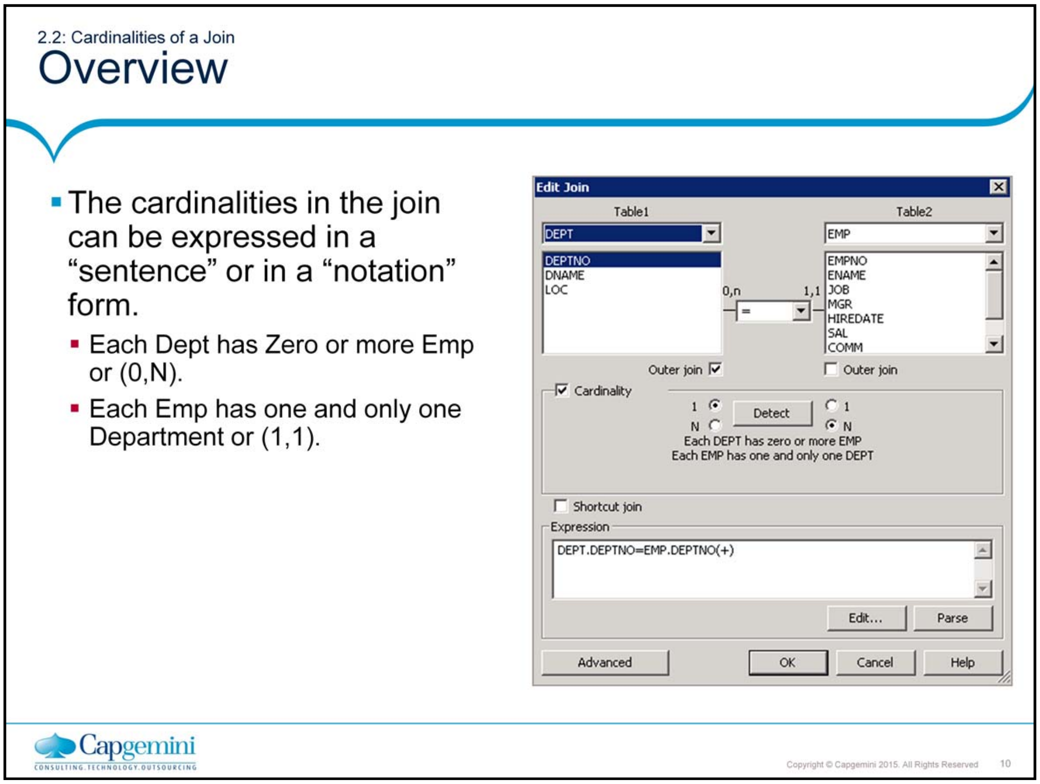
- A shortcut join is a join that provides an alternative path between two tables.
- Shortcut joins improve the performance of a query by not taking into account intermediate tables, thus shortening a normally longer join path.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 9

<u>Shortcut Join</u>:

- A shortcut join is possible only if both the tables are detailed to same level.
  For example: A product_promotion_fact can be joined with Shop_fact via Article lookup table only.
- Since both the fact tables are detailed to the Article level, a shortcut join between both the fact tables is possible to improve the performance.
- **Note:** Shortcut join will appear as a dotted line.



2.2: Cardinalities of a Join

Overview

- The cardinalities in the join can be expressed in a “sentence” or in a “notation” form.
  - Each Dept has Zero or more Emp or (0,N).
  - Each Emp has one and only one Department or (1,1).

Edit Join

Table1: DEPT

Table2: EMP

DEPTNO
DNAME
LOC

0,n = 1,1

EMPNO
ENAME
JOB
MGR
HIREDATE
SAL
COMM

Outer join ☑ □ Outer join

☑ Cardinality

1 ● N ○ Detect ○ 1 ● N

Each DEPT has zero or more EMP
Each EMP has one and only one DEPT

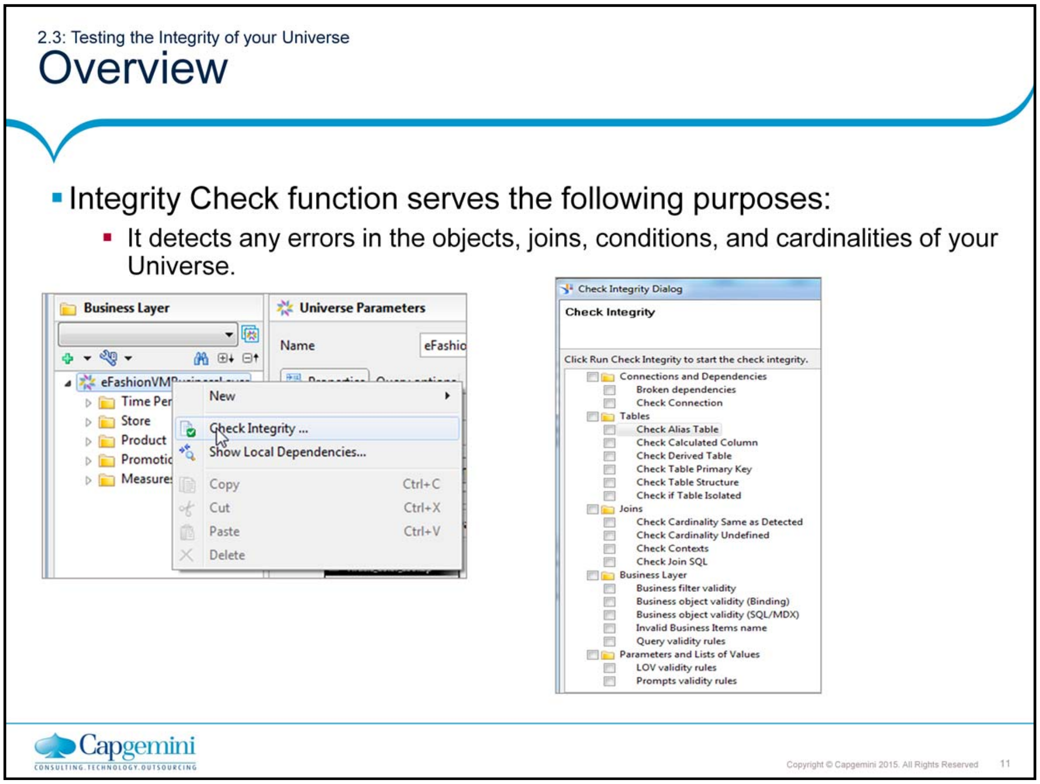
□ Shortcut join

Expression

```
DEPT.DEPTNO=EMP.DEPTNO(+)
```

<u>Cardinalities</u>:
Cardinality expresses the minimum and maximum number of instances of an entity B that can be associated with an instance of an entity A. The minimum and the maximum number of instances can be equal to 0, 1, or N.

Displaying the cardinalities:
The **Graphics** tab of the **Options** dialog box contains a number of options for the display of cardinalities in the **Structure** pane.



2.3: Testing the Integrity of your Universe

Overview

- Integrity Check function serves the following purposes:
  - It detects any errors in the objects, joins, conditions, and cardinalities of your Universe.

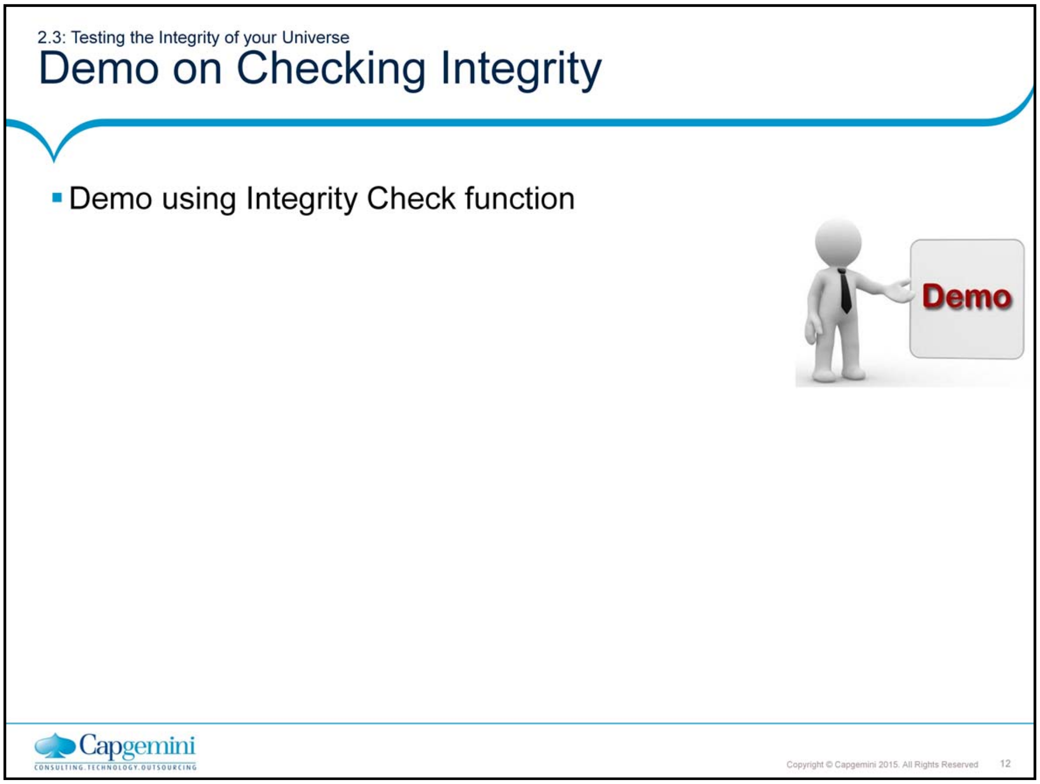
You can run a check integrity at any time for different objects and resources in the information design tool:

• Resources (data foundations, business layers, connections and shortcuts) in the Local Projects View

• Elements in the data foundation and business layer (tables, contexts, business layer objects, queries, parameters, lists of values) in the editor

Following steps are performed to Check Integrity:

1. Right-click the resource or object you wish to run a check integrity for and select Check Integrity.
2. In the left-hand pane of the "Check Integrity" dialog box, select the rules you want to apply.
3. Click Check Integrity.

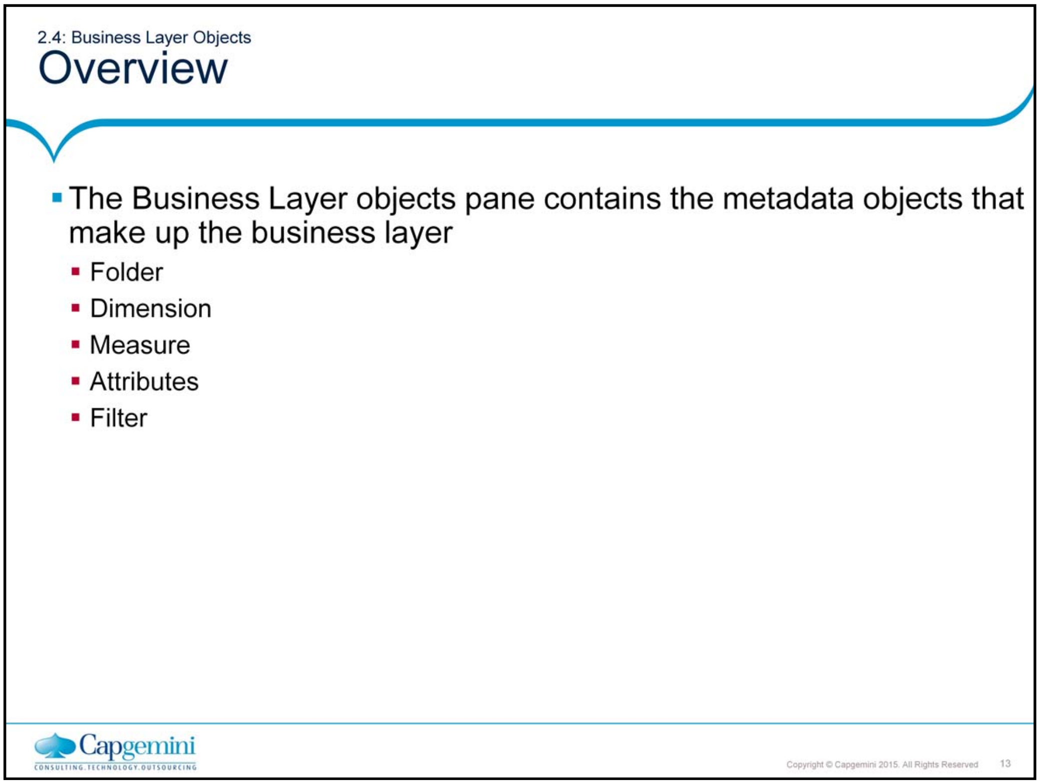


2.3: Testing the Integrity of your Universe

Demo on Checking Integrity

- Demo using Integrity Check function

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



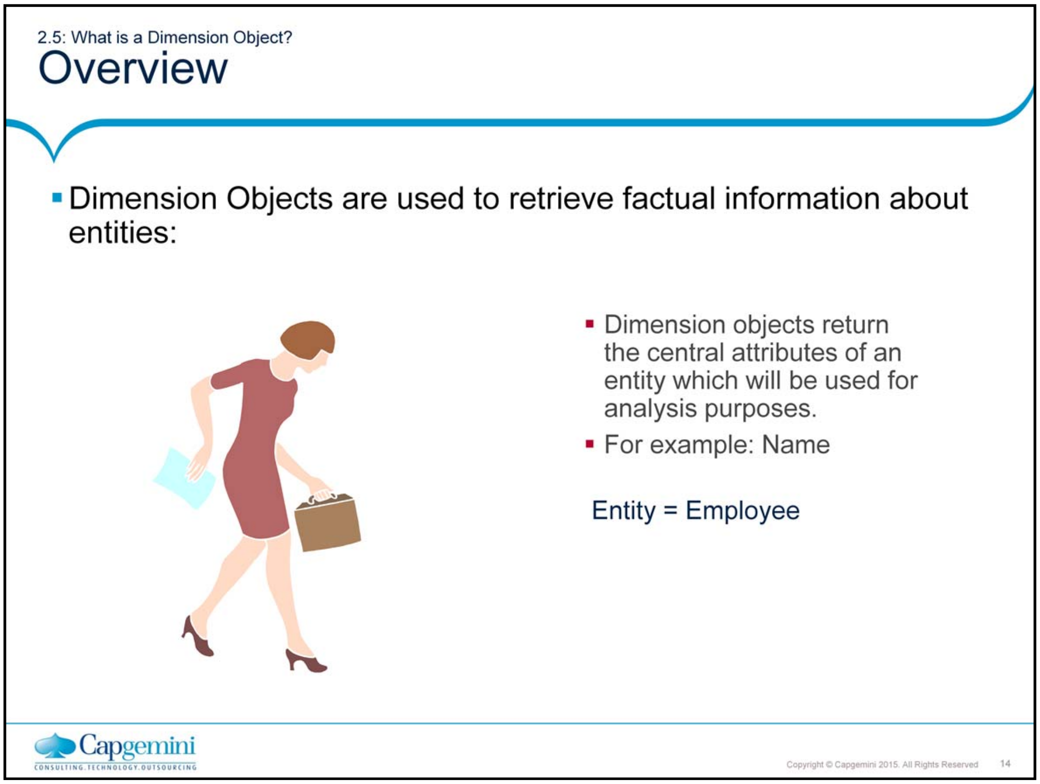
2.4: Business Layer Objects

Overview

- The Business Layer objects pane contains the metadata objects that make up the business layer
  - Folder
  - Dimension
  - Measure
  - Attributes
  - Filter

Copyright © Capgemini 2015. All Rights Reserved 13

- Business Layer has three types of objects:
- Folder : A folder is a container that holds a group of related objects. You create folders to group objects that have a common purpose in the business layer. The folder has no role in a query. It is only used to organize objects.
- Dimension: A dimension is an object that maps to one or more table columns or a function in a database and represents an axis of analysis in a query. For example, Product, Geography, Time, and Employee are common dimensions. Each dimension classifies an aspect of an activity in a business environment.
- Measure: Measures are objects that represent calculations and aggregate functions that map to statistical and analytic data in the database. In a business layer, measures represent the factual information(data)
- Attribute: An attribute is an object attached to a parent object that provides additional descriptive information about the parent. Attributes can be defined for dimensions, hierarchies, and levels
- Filter: A filter is a condition object that limits the data returned in a query. Filters can be inserted into the Query Filters pane in the Query Panel to be applied to the query.



2.5: What is a Dimension Object?

Overview

- Dimension Objects are used to retrieve factual information about entities:
  - Dimension objects return the central attributes of an entity which will be used for analysis purposes.
  - For example: Name

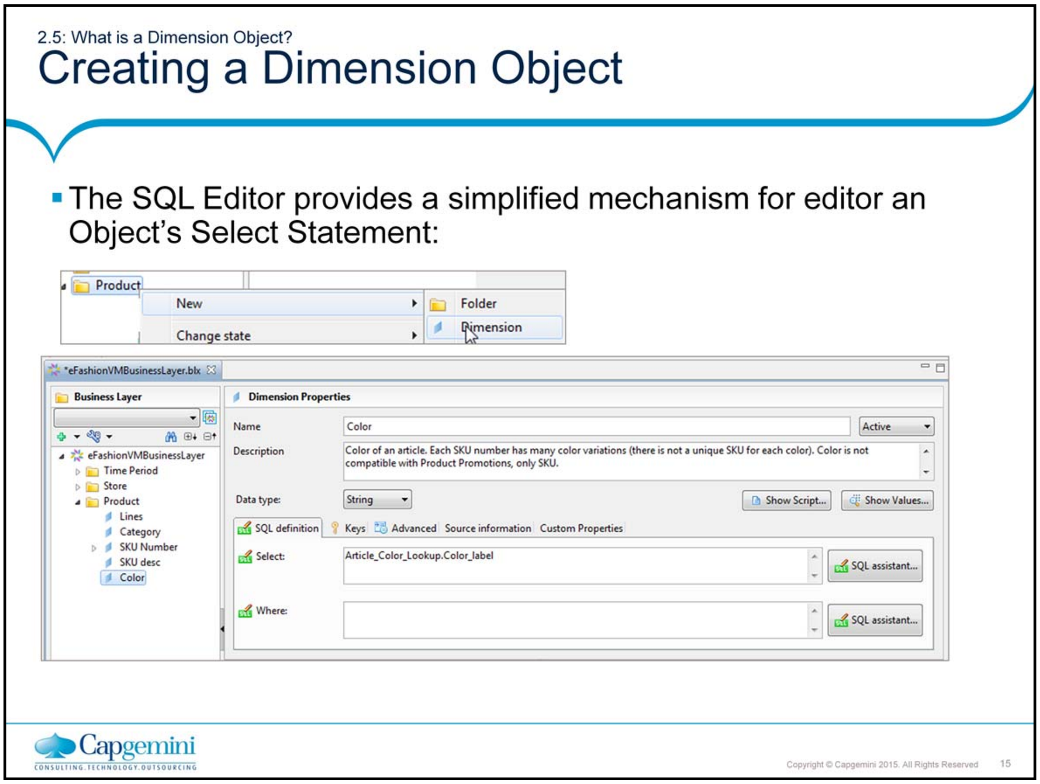
Entity = Employee

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

14

<u>What is Dimension Object?</u>

- The qualification of an object reveals how it can be used in multidimensional analysis. It may be qualified as a **dimension**, **attribute**, or a **measure**. This section deals with **dimension objects** and **attribute objects** and the differences between the two. Measure objects will be dealt with later in the course.
- The **attribute objects** returns information on the additional attributes of the entity in question. The data it returns is purely informational and is never used for analysis purpose.



2.5: What is a Dimension Object?

Creating a Dimension Object

- The SQL Editor provides a simplified mechanism for editor an Object’s Select Statement:

Name: Color — Active

Description: Color of an article. Each SKU number has many color variations (there is not a unique SKU for each color). Color is not compatible with Product Promotions, only SKU.

Data type: String

Select: Article_Color_Lookup.Color_label

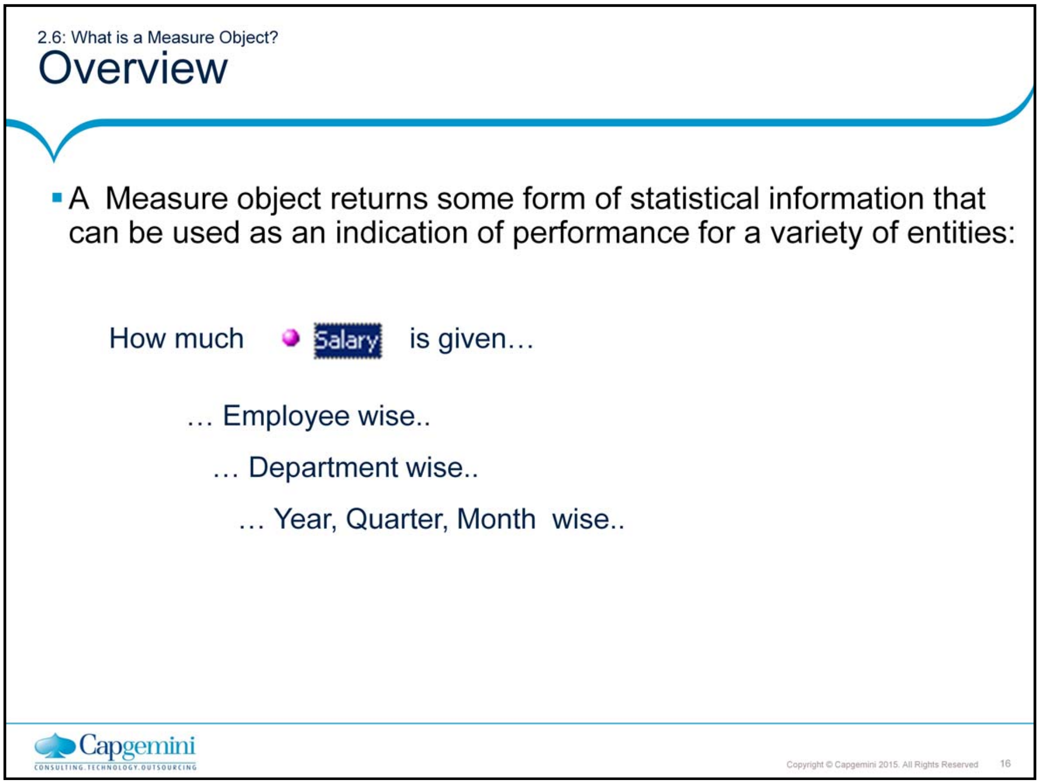
Where:

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

15

<u>Creating a Dimension Object</u>:

- To create a dimension objects, we need to follow the following steps:
    - Open the Folder, select New → Dimension.
    - Give the name of the Dimension, it’s data type.
    - Provide a meaningful description for the object.
    - Select the relevant field name from the **Tables and Columns** pane.



2.6: What is a Measure Object?

Overview

- A Measure object returns some form of statistical information that can be used as an indication of performance for a variety of entities:

How much ● Salary is given…

… Employee wise..

… Department wise..

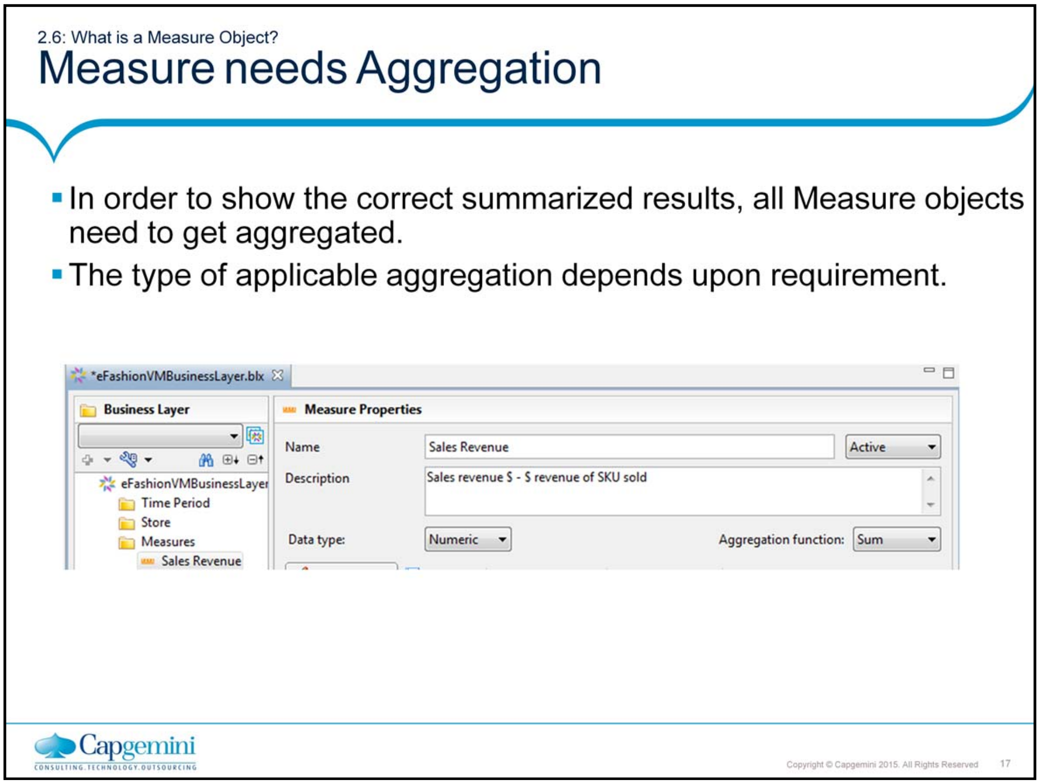
… Year, Quarter, Month wise..

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

16

<u>What is a Measure Object?</u>

- A Measure object is used to return statistical information to an Analyst. It returns the “numeric information” that Dimension objects are compared against to quantify performance.
- Measure objects are very flexible due to the fact that they are semantically dynamic. This means that their meaning in a query will be dependant on the context in which they are actually used, or in other words the other objects against which they are being projected.
- **For example:**
    - ➢ If the **Employee Name Object** and the **Salary Measure Object** are placed in the **Query Panel**, then the query will return the total salary paid to an employee for all the number of years for which we have data in the warehouse.
    - ➢ If the **Year** and **Salary** are combined in the **Query Panel**, then the **Measure** will return the total salary for each Year for all employees.
    - ➢ Finally, if the **Employee Name**, **Year**, and **Salary** are combined together in the **Query Panel**, then the **Salary Measure** will return the total salary paid per year, for each employee.



2.6: What is a Measure Object?

Measure needs Aggregation

- In order to show the correct summarized results, all Measure objects need to get aggregated.
- The type of applicable aggregation depends upon requirement.

*eFashionVMBusinessLayer.blx

Measure Properties

Name: Sales Revenue — Active

Description: Sales revenue $ - $ revenue of SKU sold

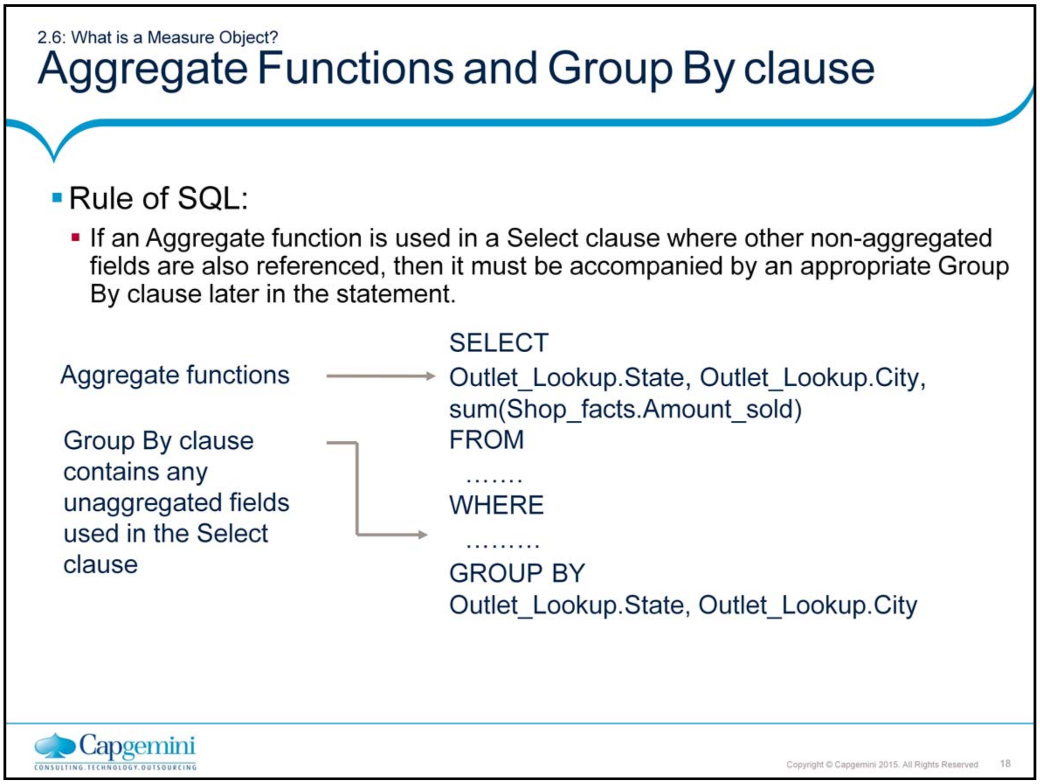
Data type: Numeric

Aggregation function: Sum

17

Measure needs Aggregation:

- While displaying the results, the value of Measure objects needs to get re-calculated according to available dimensions. Therefore it is important to specify the type of aggregation that needs to be applied on these Measure objects to display correct summarized result.
- This helps in building a query with proper **Aggregate** function and **Group By** clause.



2.6: What is a Measure Object?

Aggregate Functions and Group By clause

- Rule of SQL:
  - If an Aggregate function is used in a Select clause where other non-aggregated fields are also referenced, then it must be accompanied by an appropriate Group By clause later in the statement.

Aggregate functions

Group By clause contains any unaggregated fields used in the Select clause

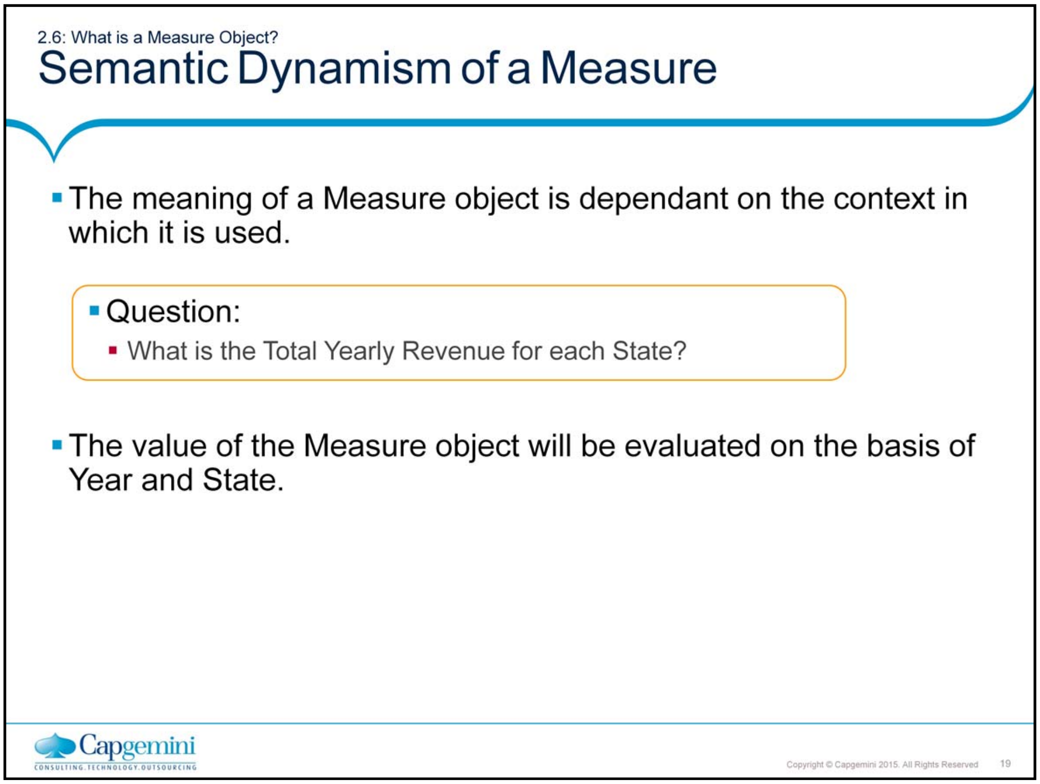
```
SELECT
Outlet_Lookup.State, Outlet_Lookup.City,
sum(Shop_facts.Amount_sold)
FROM
  …….
WHERE
  ………
GROUP BY
Outlet_Lookup.State, Outlet_Lookup.City
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

18

Aggregate Functions and Group By clause:

- If an Aggregate function is used in the **Select** clause of the statement, then a **Group By** clause must be added to the SQL statement. This will help to ensure that the level of aggregation is controlled by the other fields referenced in the **Select** clause. This is a rule of SQL.
- **For Example:** The SQL statement in the example in the above slide takes the data from the State & City and compares it with the Revenue that has been generated by the orders placed in those cities. Even though in the database this data is stored in a format whereby each city has many transactions attributed to it (because each city has many orders), the result set returned has only one row of data for each city comparing the city name with the amount of revenue generated in that city.
- The criterion that controls this behavior is the **Group By** clause at the bottom of the SQL statement. The server will generate a revenue calculation for every invoice that has been placed by each customer for each city. It will then look at the **Group By** clause and apply the **Aggregate** function (in this case **Sum**), surrounding the revenue calculation, to this result set according to the level indicated in the **Group By** clause. The **Group By** clause, tells it to group the Result set by the State and City name.



2.6: What is a Measure Object?

Semantic Dynamism of a Measure

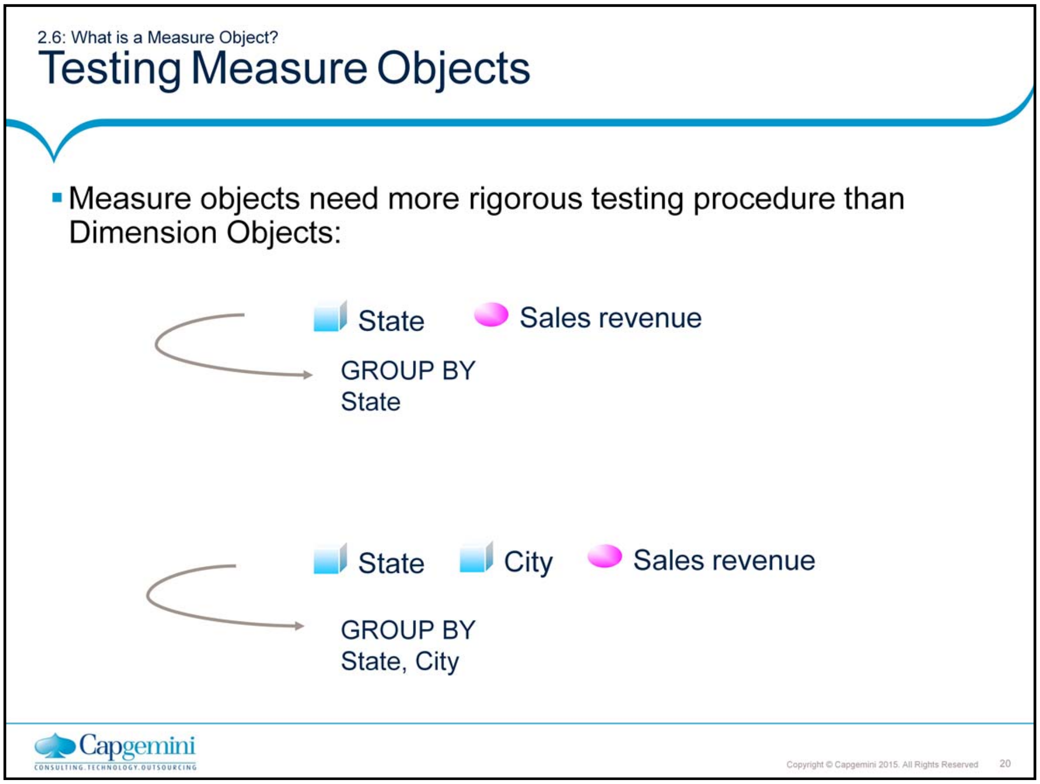
- The meaning of a Measure object is dependant on the context in which it is used.

- Question:
  - What is the Total Yearly Revenue for each State?

- The value of the Measure object will be evaluated on the basis of Year and State.

Semantic Dynamism of a Measure:

- Measure objects in a BusinessObjects Universe are semantically dynamic. The values returned by a Measure Object are dependant on the context in which they are being used. The context is set by the other Objects that are used in the query.
- Business Objects uses the **Group By** rule of SQL in the inference engine to control the semantically dynamic behavior of Measure objects. This guarantees that in whatever combination of Objects that the Measure is used, its return result will always be aggregated to the appropriate level.



2.6: What is a Measure Object?

Testing Measure Objects

- Measure objects need more rigorous testing procedure than Dimension Objects:

State Sales revenue

GROUP BY
State

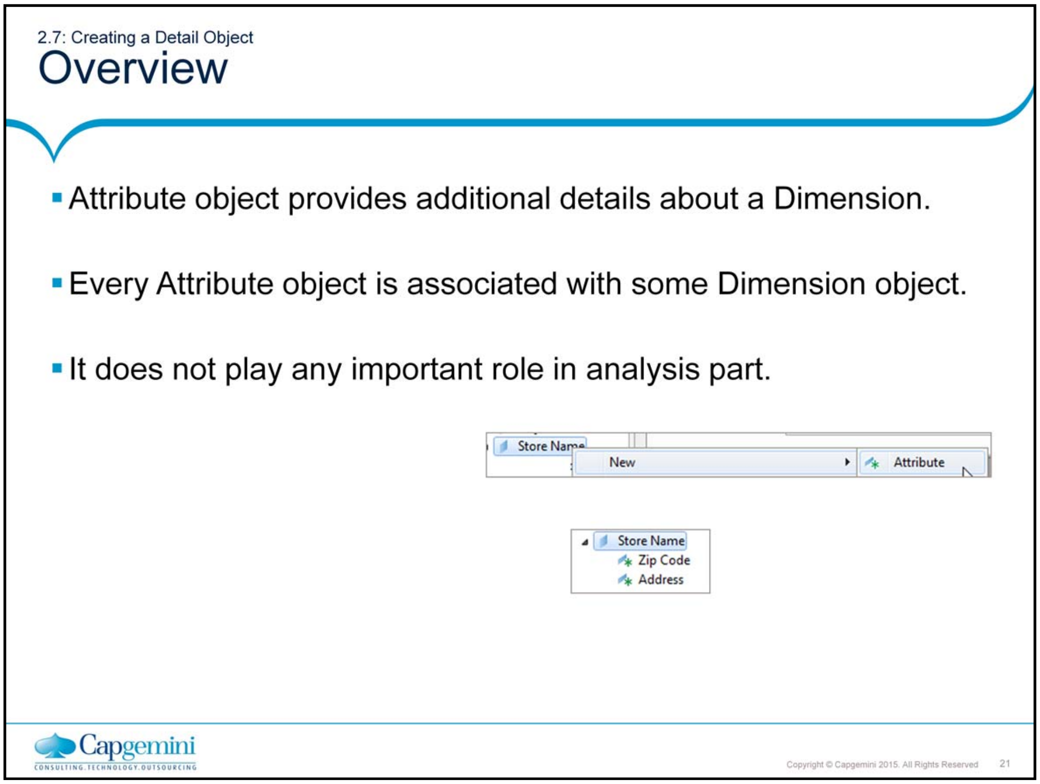
State City Sales revenue

GROUP BY
State, City

20

<u>Testing Measure Objects</u>:

- Measure objects require a slightly more rigorous testing procedure than Dimension objects. By definition, they must return a different set of results for each context into which they are placed. Therefore the designer must ascertain that the appropriate level of aggregation is being achieved (the correct GROUP BY clause generated) for each situation in which the Object may be used.
- In the example in the above slide, Sales Revenue generation can be tested at the level of State and/or City. Each of these situations (and possible combinations) must be tested, and the SQL must be examined to determine that it is correct.



2.7: Creating a Detail Object

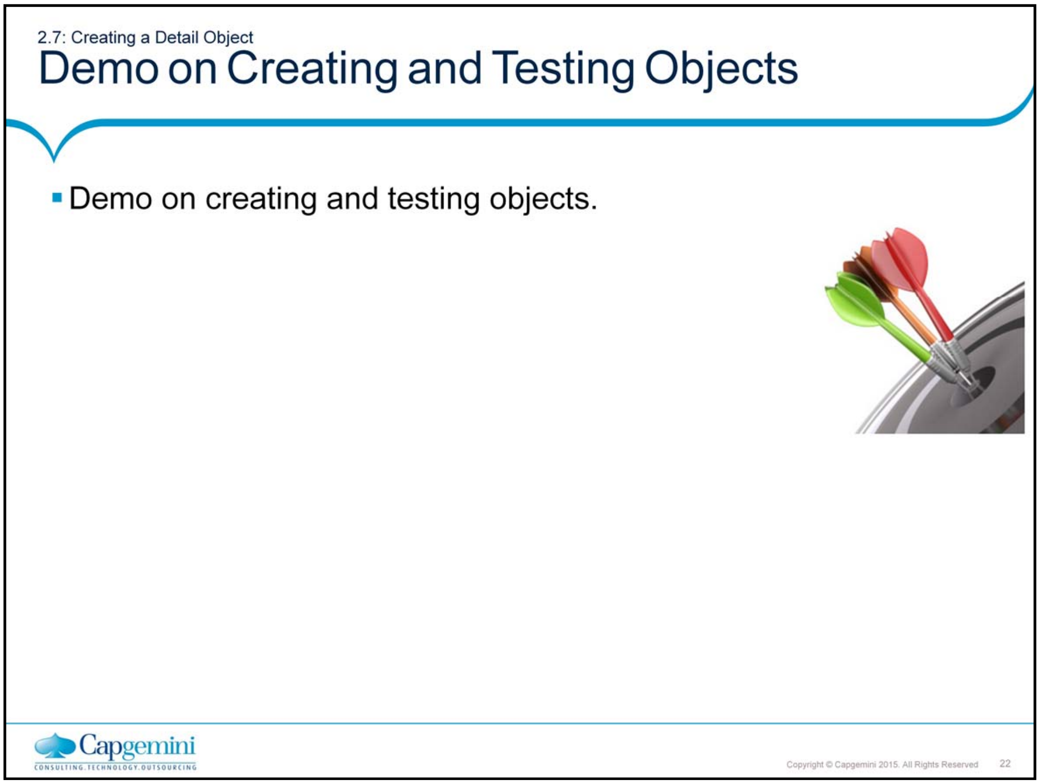
Overview

- Attribute object provides additional details about a Dimension.
- Every Attribute object is associated with some Dimension object.
- It does not play any important role in analysis part.

21

Creating a Attribute Object:

- Detail objects are used by an end user as a source of additional information, on the entity being analyzed, should they require it.
  For example: A detail of the Customer Name object that has just been created can be a Phone Number or an Address Object.
- Attribute objects are less likely to be used in queries, in general, and conditions, in particular. It is less likely that a list of values needs to be associated with the object.
- Attribute objects return information on ancillary attributes of an entity that are not central to Analysis.
- **For example:** Address, Phone Numbers

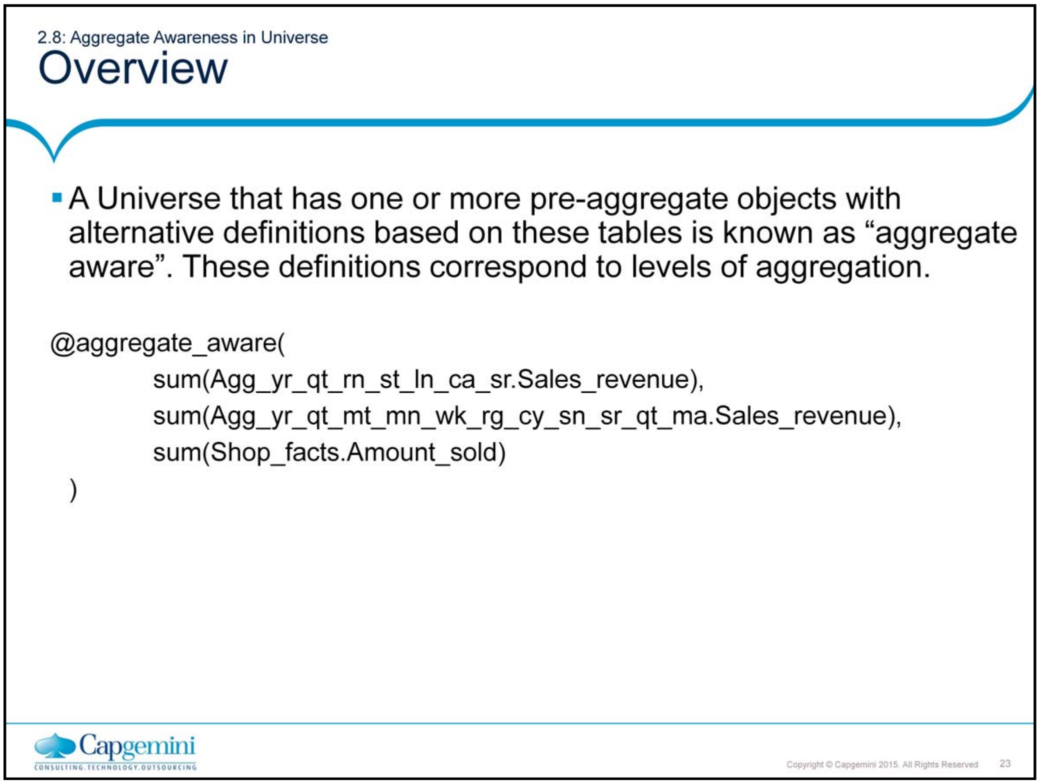


2.7: Creating a Detail Object

Demo on Creating and Testing Objects

- Demo on creating and testing objects.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



2.8: Aggregate Awareness in Universe

Overview

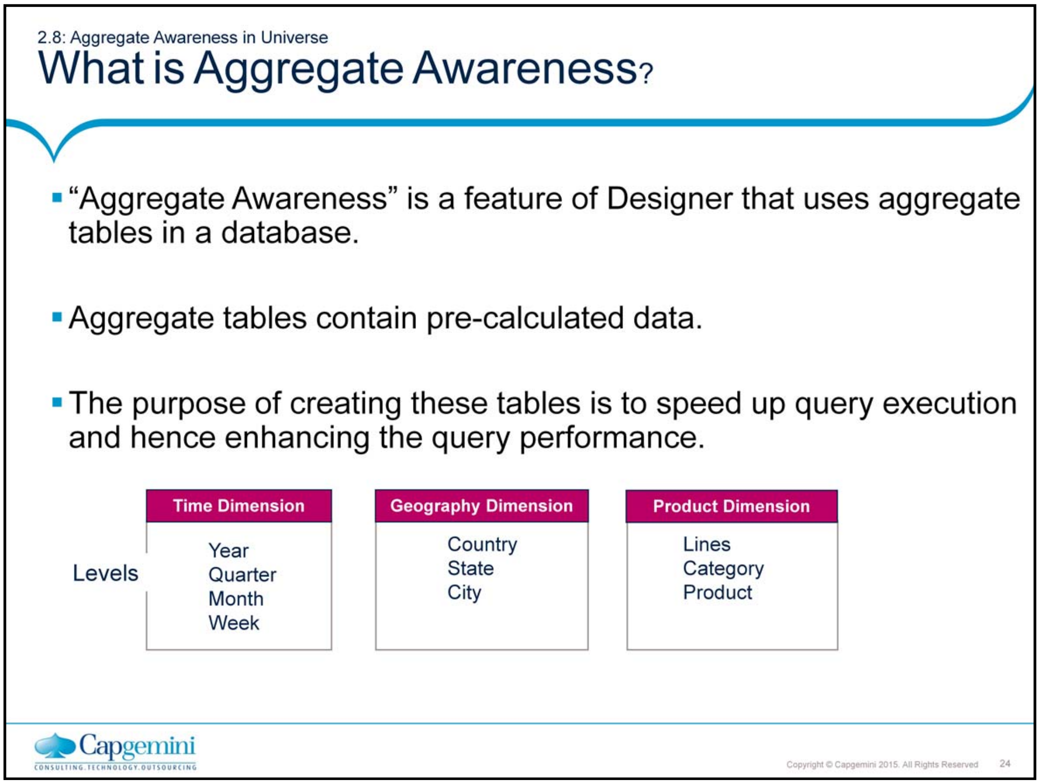
- A Universe that has one or more pre-aggregate objects with alternative definitions based on these tables is known as “aggregate aware”. These definitions correspond to levels of aggregation.

```
@aggregate_aware(
          sum(Agg_yr_qt_rn_st_ln_ca_sr.Sales_revenue),
          sum(Agg_yr_qt_mt_mn_wk_rg_cy_sn_sr_qt_ma.Sales_revenue),
          sum(Shop_facts.Amount_sold)
  )
```

23

<u>Aggregate Awareness in Universe</u>:

- “**Aggregate awareness**” is a feature of Business Objects Designer that makes use of “aggregate tables” in a database to improve the performance of query processing. These are tables that contain pre-calculated data. The reliability of the technique depends on the accuracy of the aggregate tables. In fact, they must be refreshed at the same time as all fact tables.
- **For example:** A Sales Revenue object can be pre-aggregated by month, by quarter, or by year. Queries built from such a Universe return information aggregated to the appropriate level at optimal speed.



2.8: Aggregate Awareness in Universe

What is Aggregate Awareness?

- “Aggregate Awareness” is a feature of Designer that uses aggregate tables in a database.
- Aggregate tables contain pre-calculated data.
- The purpose of creating these tables is to speed up query execution and hence enhancing the query performance.

	Time Dimension	Geography Dimension	Product Dimension
Levels	Year Quarter Month Week	Country State City	Lines Category Product

24

<u>What is Aggregate awareness</u>?

- At its lowest level, the data warehouse can store daily information about products as per geographical location. In other words, there is one row for product purchases. Assume that there are 1000 different products sold everyday. This can be mathematically expressed as 365 days x 1000 products = 365,000 rows per year. If you have data for three years, then there will be more than a million records.
- Thus, if you were to seek information about yearly sales per product, you will cause your database engine to add up a very large number of rows. However, in actual fact, the yearly sales of companies may involve fewer rows. This can be represented as 3 years x 1000 Products = 3000 rows
- Thus, 3000 rows from a table are sufficient to answer the question. Clearly, it will be far more efficient to pre-summarize these rows into aggregate tables.

2.8: Aggregate Awareness in Universe

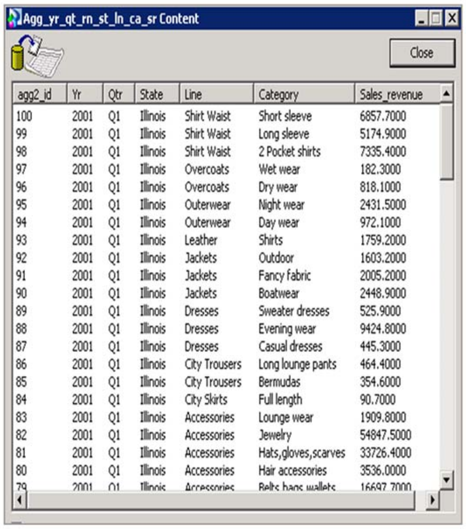
Creating Aggregate Awareness

- The order needs to be from Higher level of aggregation to Lower level of Aggregation.

```
@aggregate_aware
(
sum(Agg_yr_qt_rn_st_ln_ca_sr.
Sales_revenue),

sum(Agg_yr_qt_mt_mn_wk_rg_cy_
sn_sr_qt_ma.Sales_revenue),

sum(Shop_facts.Amount_sold)
)
```

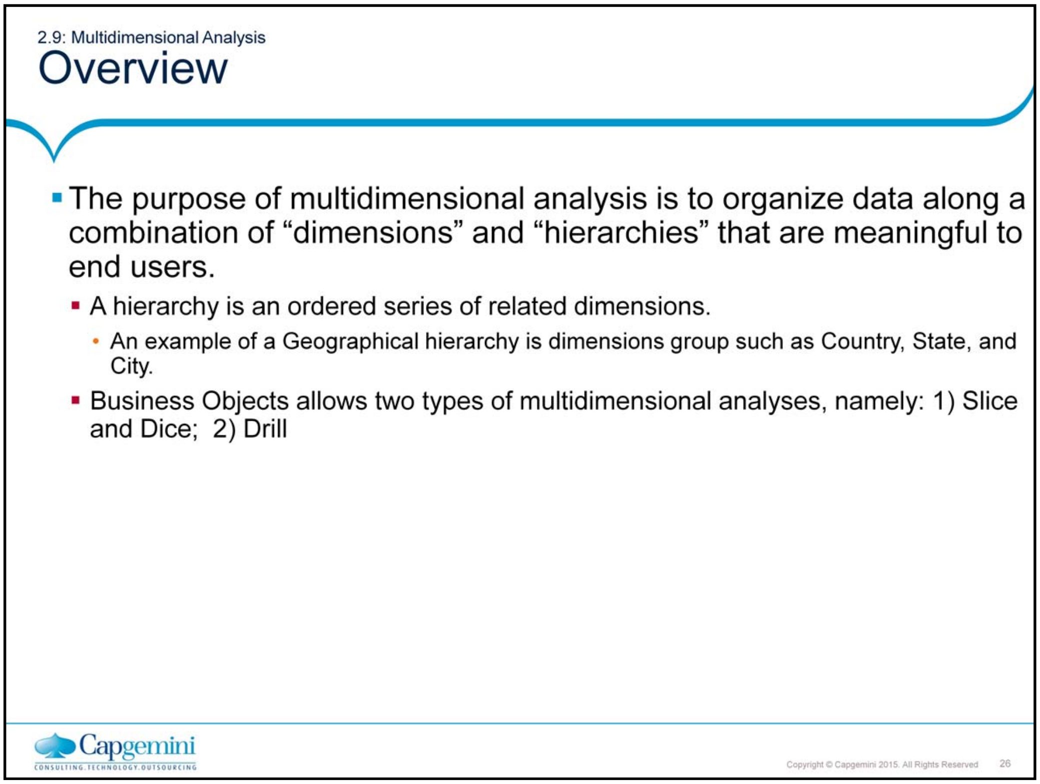


agg2_id	Yr	Qtr	State	Line	Category	Sales_revenue
100	2001	Q1	Illinois	Shirt Waist	Short sleeve	6857.7000
99	2001	Q1	Illinois	Shirt Waist	Long sleeve	5174.9000
98	2001	Q1	Illinois	Shirt Waist	2 Pocket shirts	7335.4000
97	2001	Q1	Illinois	Overcoats	Wet wear	182.3000
96	2001	Q1	Illinois	Overcoats	Dry wear	818.1000
95	2001	Q1	Illinois	Outerwear	Night wear	2431.5000
94	2001	Q1	Illinois	Outerwear	Day wear	972.1000
93	2001	Q1	Illinois	Leather	Shirts	1759.2000
92	2001	Q1	Illinois	Jackets	Outdoor	1603.2000
91	2001	Q1	Illinois	Jackets	Fancy fabric	2005.2000
90	2001	Q1	Illinois	Jackets	Boatwear	2448.9000
89	2001	Q1	Illinois	Dresses	Sweater dresses	525.9000
88	2001	Q1	Illinois	Dresses	Evening wear	9424.8000
87	2001	Q1	Illinois	Dresses	Casual dresses	445.3000
86	2001	Q1	Illinois	City Trousers	Long lounge pants	464.4000
85	2001	Q1	Illinois	City Trousers	Bermudas	354.6000
84	2001	Q1	Illinois	City Skirts	Full length	90.7000
83	2001	Q1	Illinois	Accessories	Lounge wear	1909.8000
82	2001	Q1	Illinois	Accessories	Jewelry	54847.5000
81	2001	Q1	Illinois	Accessories	Hats,gloves,scarves	33726.4000
80	2001	Q1	Illinois	Accessories	Hair accessories	3536.0000
79	2001	Q1	Illinois	Accessories	Belts bags wallets	16697.7000

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

<u>Creating Aggregate Awareness</u>:

- While creating aggregate_aware objects, a developer needs to be aware of various aggregate tables and level of granularity.
- The order of parameters needs to be from Higher level of aggregation to Lower level of aggregation. The last parameter will be from the base table itself.



2.9: Multidimensional Analysis

Overview

- The purpose of multidimensional analysis is to organize data along a combination of “dimensions” and “hierarchies” that are meaningful to end users.
  - A hierarchy is an ordered series of related dimensions.
    - An example of a Geographical hierarchy is dimensions group such as Country, State, and City.
  - Business Objects allows two types of multidimensional analyses, namely: 1) Slice and Dice; 2) Drill

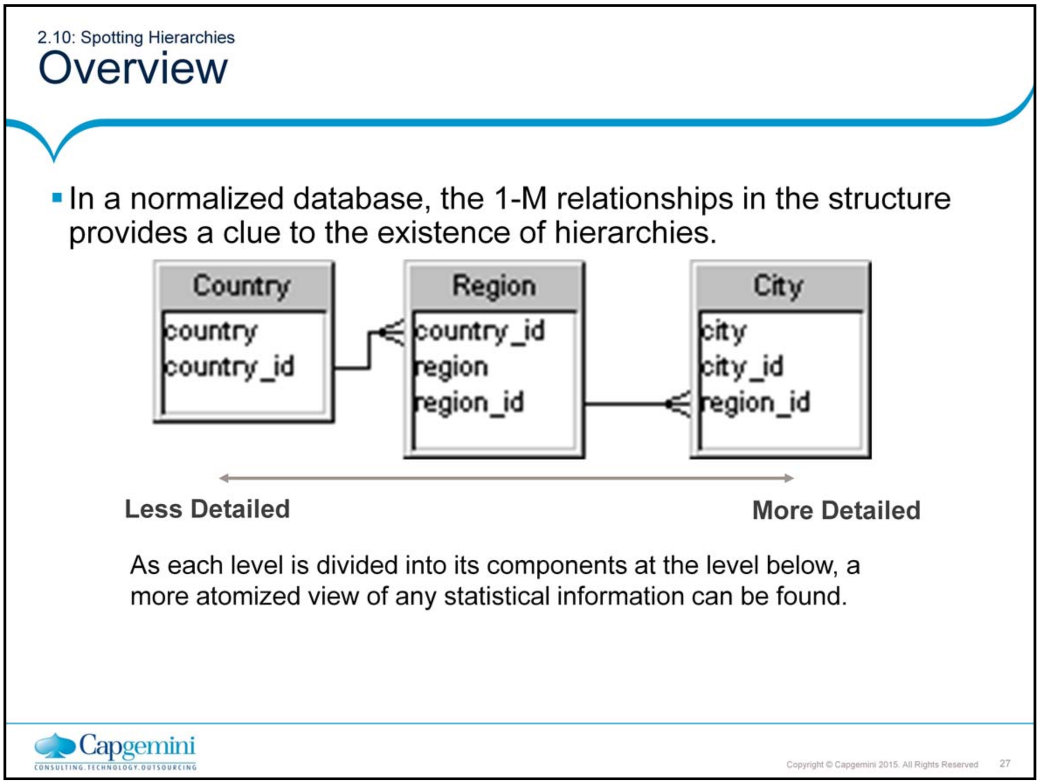
Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 26

Multidimensional Analysis:

- Multidimensional Analysis gives the ability to end users to observe data from various viewpoints. In this way, an end user can spot trends or exceptions in the data. A dimension is simply the object to be tracked. A dimension can be an object such as State, City or Product Lines.
- With slice and dice, an end user can rotate a micro-cube in order to view it from different perspectives.
- **For example:**
  - Let us say that a micro-cube is made up of three hierarchies: Country, State, Product Lines along with Sales revenue as measure.
  - The sales manager may wish to view Revenue by Country. By rotating the micro-cube, the sales manager can also view Revenue by State or by Product Lines.
- Thus, a micro-cube with n dimensions has n x (n - 1) possible views.
- Drill gives an end user the ability to navigate through hierarchical levels of detail. The notion of hierarchy is very important in drill. It provides the framework for “drilling up” or “drilling down”. Drill is the central mechanism through which multidimensional analysis can be conducted.



2.10: Spotting Hierarchies

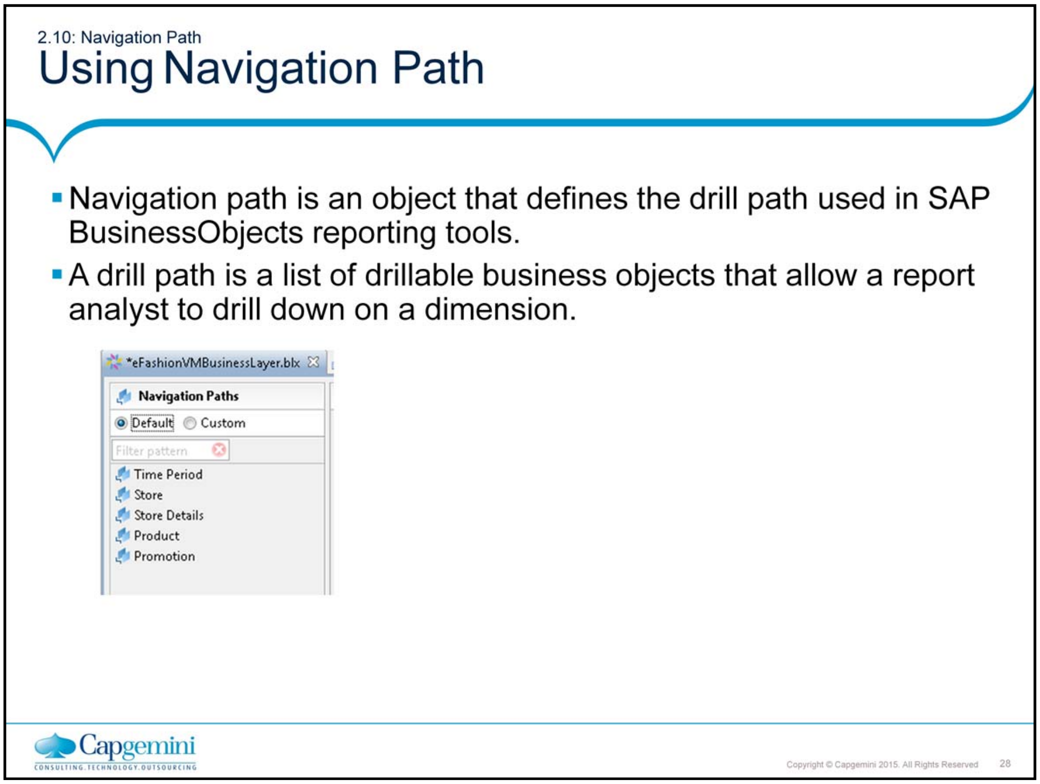
Overview

- In a normalized database, the 1-M relationships in the structure provides a clue to the existence of hierarchies.

As each level is divided into its components at the level below, a more atomized view of any statistical information can be found.

<u>Spotting Hierarchies</u>:

- The hierarchies implicit in the data are dependent on the nature of the data and the way it has been stored in the database.
- Thus you may need to analyze the data very carefully in order to find the hierarchies in your specific system that are best suited to your users’ requirements for analysis.



2.10: Navigation Path

Using Navigation Path

- Navigation path is an object that defines the drill path used in SAP BusinessObjects reporting tools.
- A drill path is a list of drillable business objects that allow a report analyst to drill down on a dimension.

*eFashionVMBusinessLayer.blx

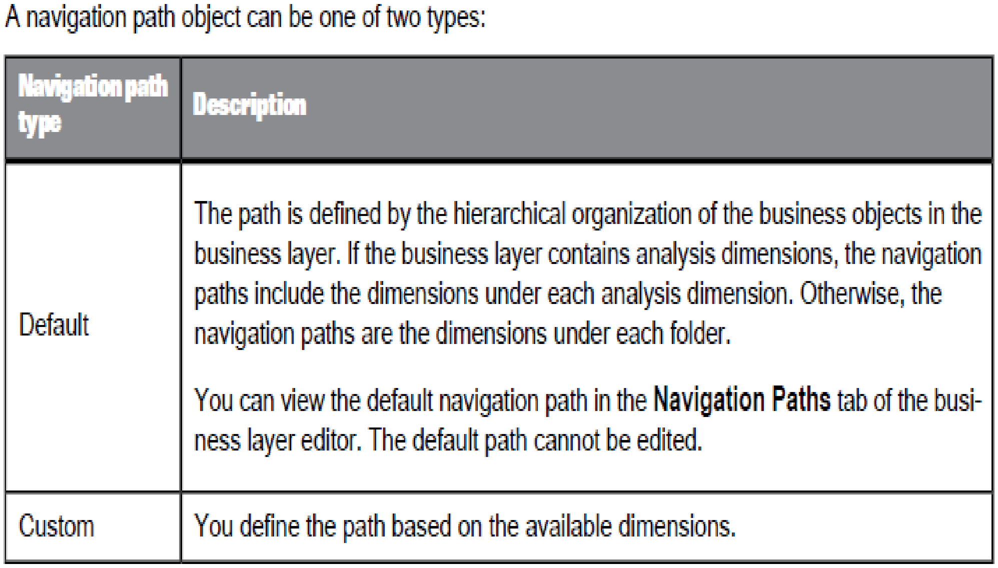
Navigation Paths

● Default ○ Custom

Filter pattern

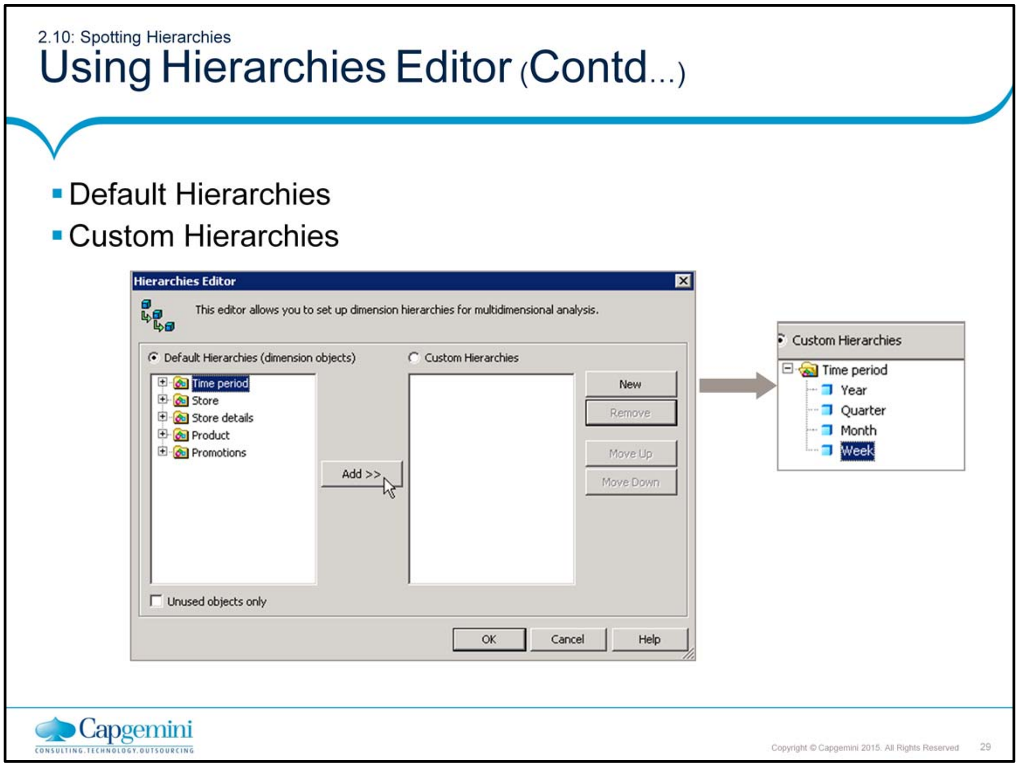
- Time Period
- Store
- Store Details
- Product
- Promotion

28



A navigation path object can be one of two types:

Navigation path type	Description
Default	The path is defined by the hierarchical organization of the business objects in the business layer. If the business layer contains analysis dimensions, the navigation paths include the dimensions under each analysis dimension. Otherwise, the navigation paths are the dimensions under each folder. You can view the default navigation path in the **Navigation Paths** tab of the business layer editor. The default path cannot be edited.
Custom	You define the path based on the available dimensions.



2.10: Spotting Hierarchies

Using Hierarchies Editor (Contd…)

- Default Hierarchies
- Custom Hierarchies

Hierarchies Editor

This editor allows you to set up dimension hierarchies for multidimensional analysis.

● Default Hierarchies (dimension objects)

○ Custom Hierarchies

Inserting a navigation path object into a business layer

Creating a Hierarchy:

1. Open the business layer in the editor by double-clicking the business layer name in the Local Projects View.

2. Click the Navigation Paths tab under the Business Layer pane.

3. Select Custom at the top of the Navigation Paths pane.

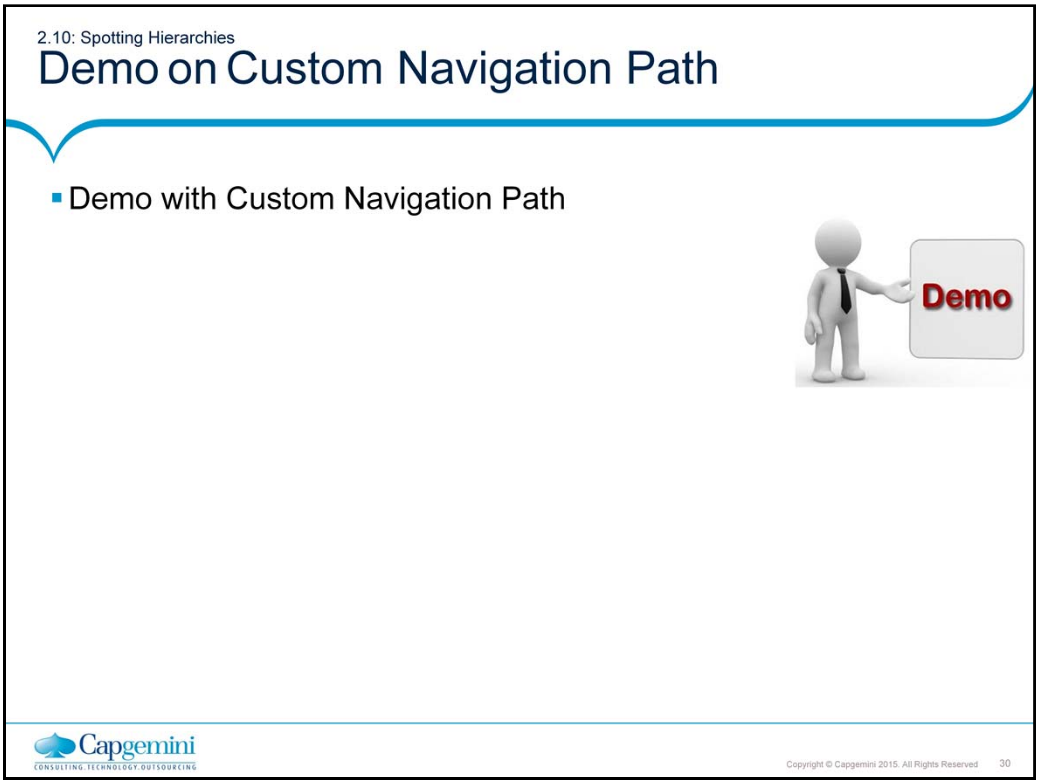
4. Click the Insert Navigation Path icon .

5. Enter a Name and optionally a Description for the path.

The name and description are available to display in the query and reporting tools that use the published universe.

6. Click Add to select dimensions for the path. Use the up and down arrow keys to change the order of dimensions in the list.

7. Save the business layer.



2.10: Spotting Hierarchies

Demo on Custom Navigation Path

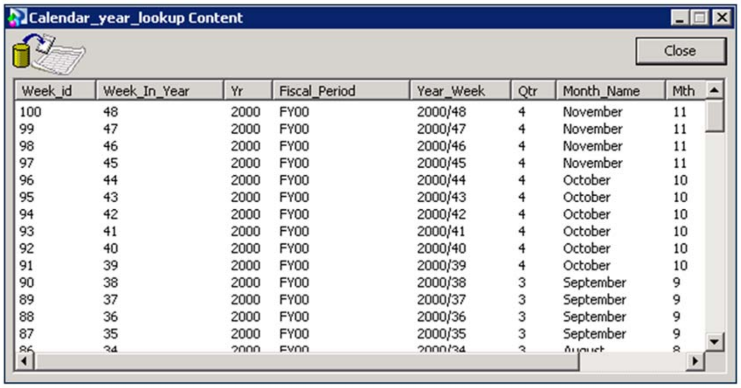
- Demo with Custom Navigation Path

30

2.10: Time Hierarchy

DBA Defined Time Periods

- Altering the database table structure gives the designer the flexibility to provide their own definition of periods.

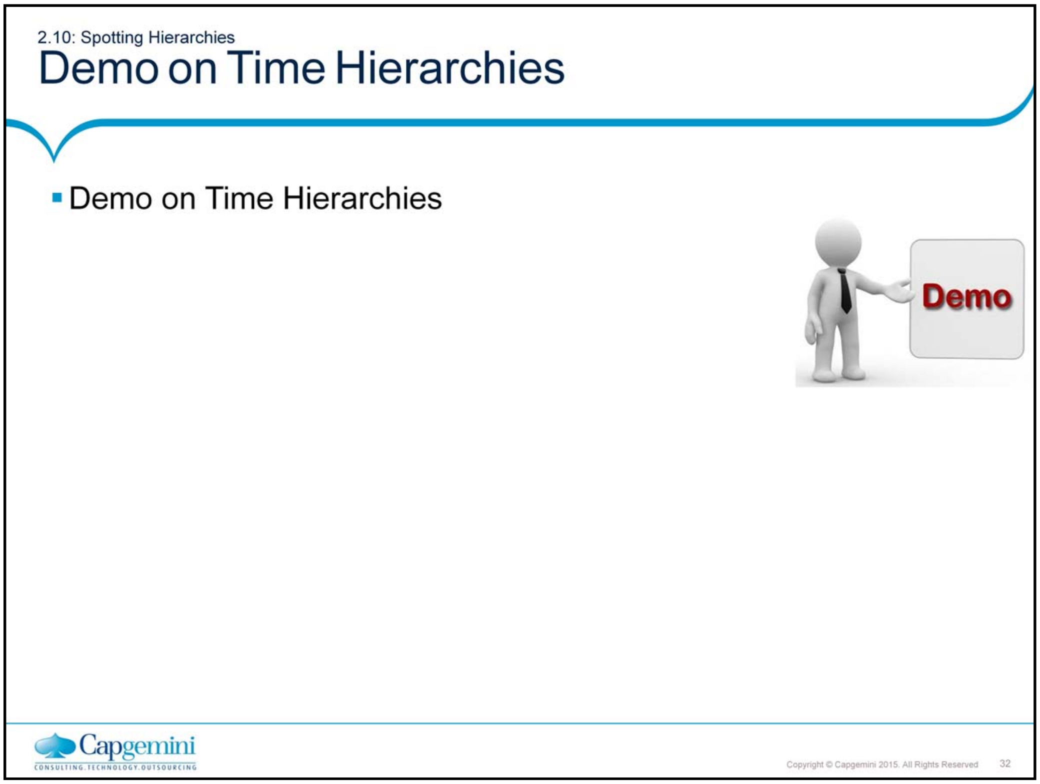


Week_id	Week_In_Year	Yr	Fiscal_Period	Year_Week	Qtr	Month_Name	Mth
100	48	2000	FY00	2000/48	4	November	11
99	47	2000	FY00	2000/47	4	November	11
98	46	2000	FY00	2000/46	4	November	11
97	45	2000	FY00	2000/45	4	November	11
96	44	2000	FY00	2000/44	4	October	10
95	43	2000	FY00	2000/43	4	October	10
94	42	2000	FY00	2000/42	4	October	10
93	41	2000	FY00	2000/41	4	October	10
92	40	2000	FY00	2000/40	4	October	10
91	39	2000	FY00	2000/39	4	October	10
90	38	2000	FY00	2000/38	3	September	9
89	37	2000	FY00	2000/37	3	September	9
88	36	2000	FY00	2000/36	3	September	9
87	35	2000	FY00	2000/35	3	September	9
86	34	2000	FY00	2000/34	3	August	8

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

DBA Defined Time Periods:

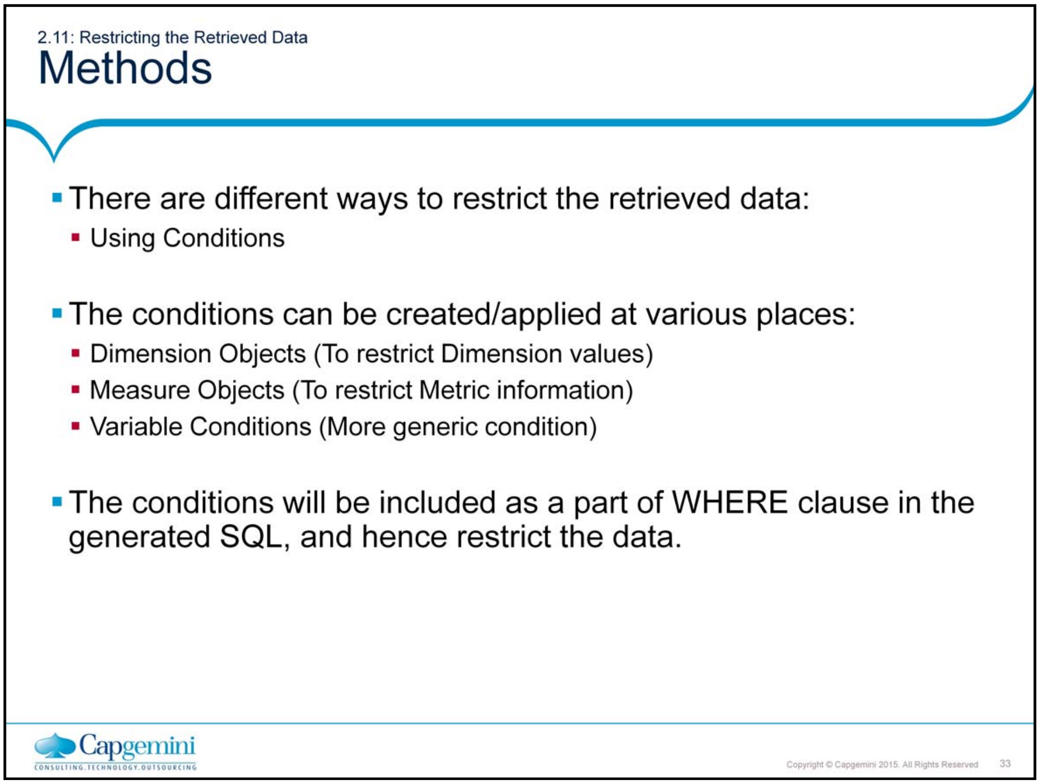
- It is possible for the database structure to be altered to allow the insertion of company or accountancy-specific time period definitions into a time hierarchy.
- **For example:** A table structure(s) can be created that reflects the time period in the financial year. This means that the year definitions will run from 1st April XXXX to 31st March XXXX+1. It also implies that the definition of Quarters will be very different to that of a calendar year function.
  - ➢ **Calendar Year:** 01/01/2003 – 31/12/2003
  - ➢ **Calendar Quarter 1:** 01/01/2003 – 31/03/2003
  - ➢ **Financial Year:** 01/04/2003 – 31/03/2004
  - ➢ **Financial Quarter 1:** 01/04/2003 – 30/06/2003
- This method is extremely flexible. However, it does require that the database be modified. It will require the assistance or permission from the DBA. It is a server-led initiative rather than BusinessObjects-led initiative. It also increases the size of the database, and will slow down any queries due to the presence of a join.



2.10: Spotting Hierarchies

Demo on Time Hierarchies

- Demo on Time Hierarchies



2.11: Restricting the Retrieved Data

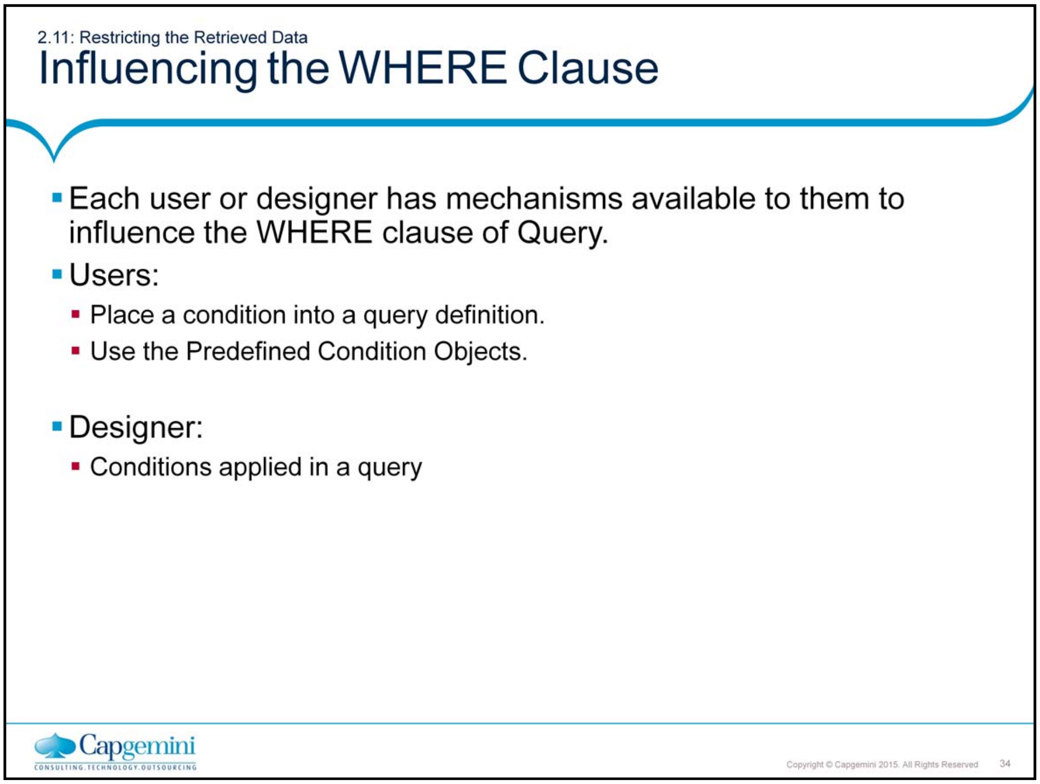
Methods

- There are different ways to restrict the retrieved data:
  - Using Conditions
- The conditions can be created/applied at various places:
  - Dimension Objects (To restrict Dimension values)
  - Measure Objects (To restrict Metric information)
  - Variable Conditions (More generic condition)
- The conditions will be included as a part of WHERE clause in the generated SQL, and hence restrict the data.

33

Restricting the Retrieved Data:

- Conditions are used to cut down or hide the important data from the end user. It reduces amount of network traffic dramatically. Therefore it enhances the flow of information around the organization. It is used to focus the analyst’s attention on specific subject matter, possibly related only to themselves.
- Restrictions are applied by the Administrators to enhance the security of data.
- **Example:** A Sales Person only sees his own Sales figures, Salary, or Bonus. Analysts will no longer be overwhelmed by the volume of data returned.



2.11: Restricting the Retrieved Data

Influencing the WHERE Clause

- Each user or designer has mechanisms available to them to influence the WHERE clause of Query.
- Users:
  - Place a condition into a query definition.
  - Use the Predefined Condition Objects.
- Designer:
  - Conditions applied in a query

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

34

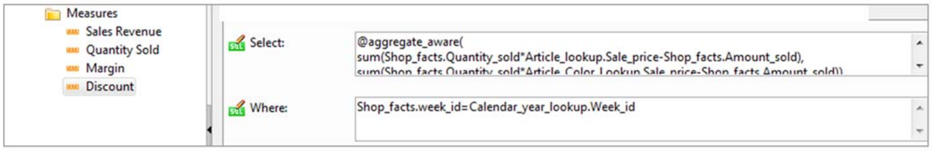
Influencing the WHERE Clause:

- An end user can place a condition into the query:
  - by defining it themselves from scratch, or
  - by using predefined **Condition Object** Objects provided by the Information Designer
- Designer has numerous ways of influencing the WHERE clause of an SQL, either directly or indirectly.
  - A direct influence means that Designer forces a condition to be applied in a Query, dependant on the combination of Objects placed in the Query panel.
  - An indirect influence over the WHERE clause is exerted by giving the User the choice of whether to use a condition or not.

2.11: Restricting the Retrieved Data

Restricting Data using WHERE clause

- The WHERE clause of an Object provides one way to restrict the data returned by an Object.
  - The figure below shows how to restrict the data



Select:

```
@aggregate_aware(
sum(Shop_facts.Quantity_sold*Article_lookup.Sale_price-Shop_facts.Amount_sold),
sum(Shop_facts.Quantity_sold*Article_Color_Lookup.Sale_price-Shop_facts.Amount_sold))
```

Where:

```
Shop_facts.week_id=Calendar_year_lookup.Week_id
```

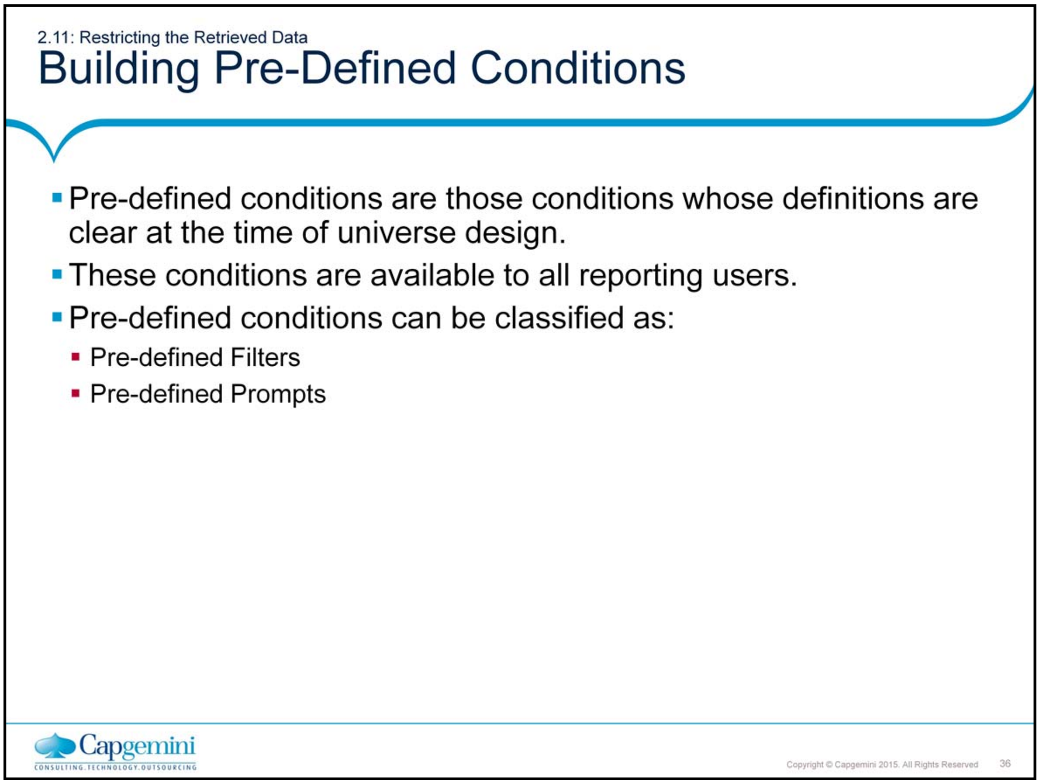
  - Note: Use only if it is must. (Not a Good Practice)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

35

<u>Restricting Data using WHERE clause</u>:

- The WHERE clause in the above mentioned example will ensure that the output will belong to Sales department only. It can be used to show selected data only.
- Use of hard coded values in the WHERE clause of a Dimension Objects definition is not recommended, as it can interrupt the multidimensional nature of the Universe structure.
  For example: A user will not get the information about other departments.



2.11: Restricting the Retrieved Data

Building Pre-Defined Conditions

- Pre-defined conditions are those conditions whose definitions are clear at the time of universe design.
- These conditions are available to all reporting users.
- Pre-defined conditions can be classified as:
  - Pre-defined Filters
  - Pre-defined Prompts

36

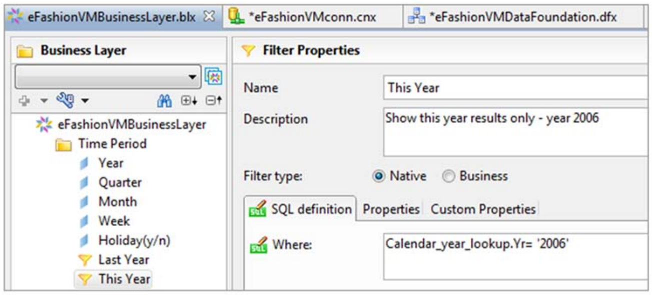
<u>Building Pre-Defined Conditions</u>:

- Pre-defined conditions are defined by the Universe designer to help the end user. It eliminates the need of re-creating the same set of conditions while preparing reports.
- These conditions act as an object, and can be included in the report by using just a drag and drop method.

2.11: Restricting the Retrieved Data

Pre-defined Filters

- Pre-defined filters are condition objects whose criterion is fixed.



Name: This Year

Description: Show this year results only - year 2006

Filter type: ● Native ○ Business

Where: Calendar_year_lookup.Yr= '2006'

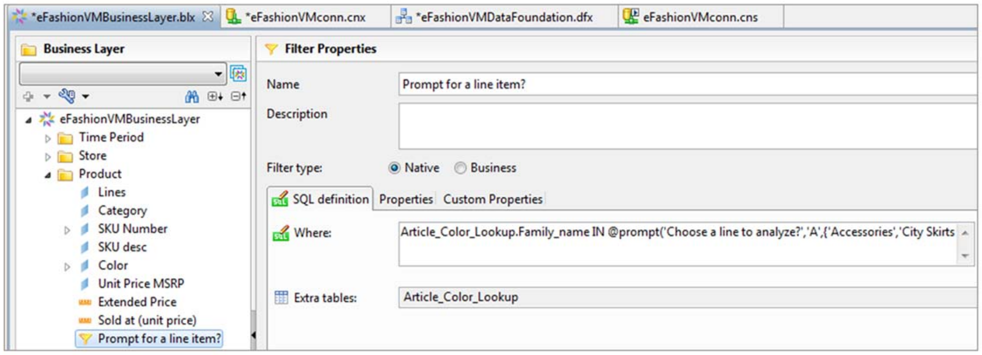
<u>Pre-defined Filters</u>:

- Pre-defined filters are static conditions that do not change. It’s criteria is hard coded during design time, and it can be applied by applying drag and drop method. It is very useful while scheduling documents.
- **Example:** A company wants to generate many reports for current year. In such a case, a report designer either has to create filter every time or use the prompt and specify the value. A pre-defined filter eliminates both the problems and leads to more simplicity.
- **Note:** Since they are static in nature, their value will not change unless they are changed in the universe.

2.11: Restricting the Retrieved Data

Flexible Restrictions: @Prompt

- The @Prompt functions allows a Designer to build a flexible filter condition whose criteria can be changed during run time.



Name: Prompt for a line item?

Description:

Filter type: ● Native ○ Business

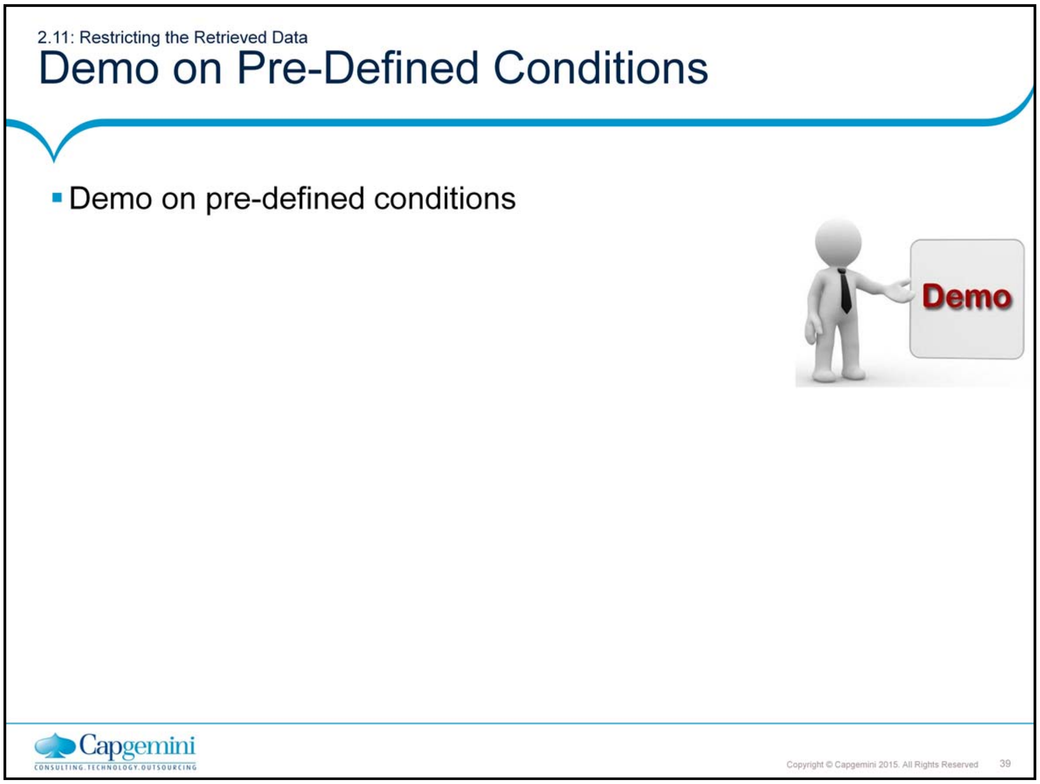
Where: Article_Color_Lookup.Family_name IN @prompt('Choose a line to analyze?','A',{'Accessories','City Skirts

Extra tables: Article_Color_Lookup

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

38

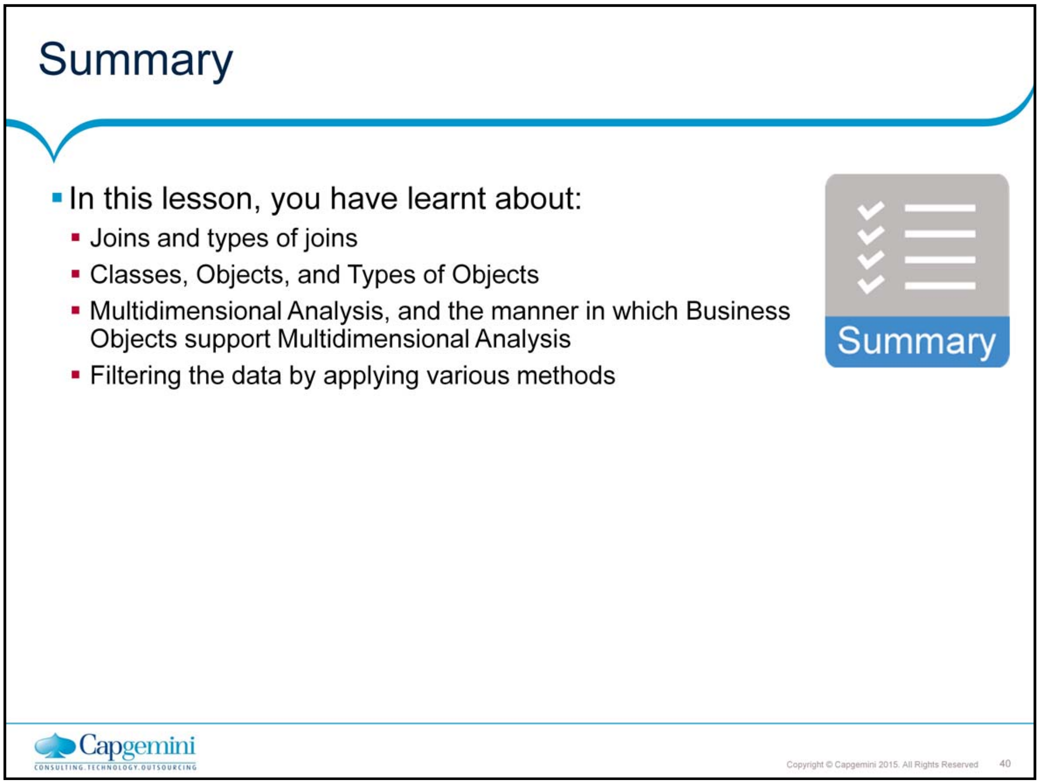
<u>Flexible Restrictions: @Prompt</u>:
Prompts provide a much more flexible approach than using hard coded values in the pre-defined filters. It gives the user the flexibility to adjust the value used in the condition each time the query is run without having to build their own.



2.11: Restricting the Retrieved Data

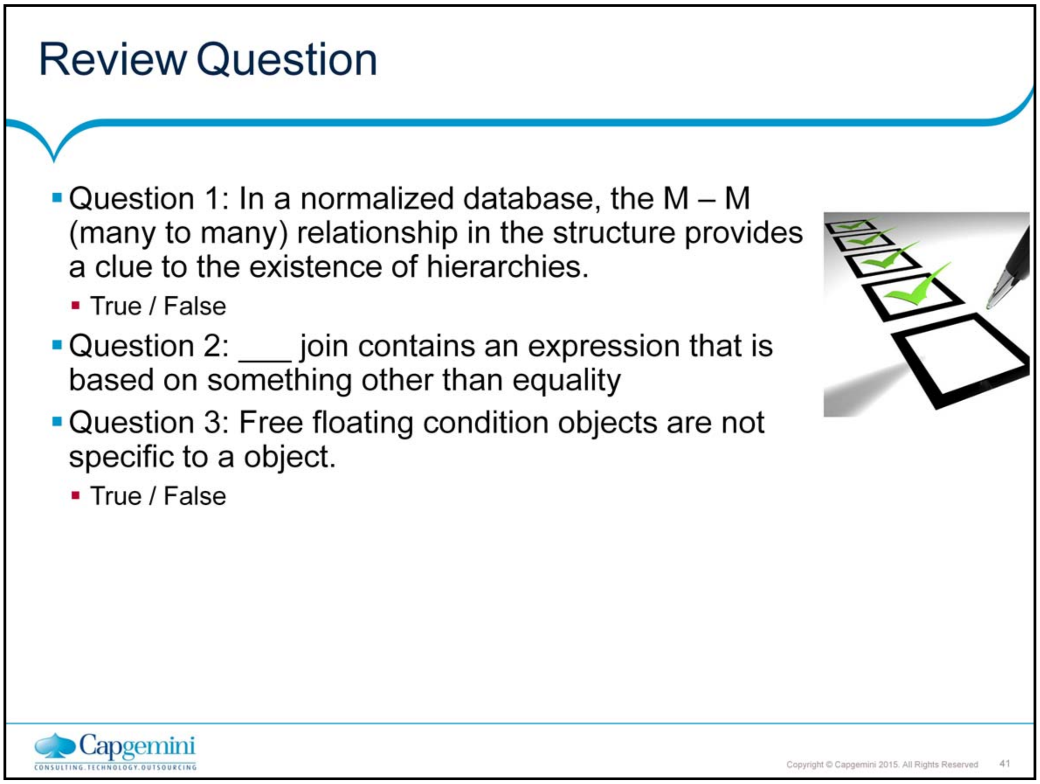
Demo on Pre-Defined Conditions

- Demo on pre-defined conditions



Summary

- In this lesson, you have learnt about:
  - Joins and types of joins
  - Classes, Objects, and Types of Objects
  - Multidimensional Analysis, and the manner in which Business Objects support Multidimensional Analysis
  - Filtering the data by applying various methods



Review Question

- Question 1: In a normalized database, the M – M (many to many) relationship in the structure provides a clue to the existence of hierarchies.
  - True / False
- Question 2: ___ join contains an expression that is based on something other than equality
- Question 3: Free floating condition objects are not specific to a object.
  - True / False