

Business Objects XI Designer

Lesson 3: Advanced Universe
Development

Lesson Objectives

- In this lesson, you will learn about:
 - What is a Loop?
 - Automated routines to identify the loops
 - Resolve loops using Alias and Context
 - Reusability of Objects in a Universe

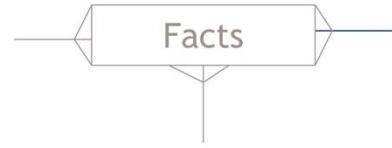


3.1: Lookup and Fact tables

Overview

- **Lookup tables:**
 - A look up table or dimension table contains information associated with a particular subject or entity.
- **Fact tables:**
 - A fact table contains statistical information about transactions. A fact table is characterized by the following join cardinality structure.





 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 3

Lookup and Fact tables:**Look up tables:**

A lookup table can hold information on Employees from a geographical perspective such as their names, telephone numbers as well as the cities and countries in which they reside.

Fact tables:

It may contain numerical figures such as Salary or Sales revenue. In a BusinessObjects Universe, most of the measures are defined from fact tables.

3.2: Join Path Problems

Overview

- A Join Path is a series of joins that a query can use to access data in the tables linked by the joins.
- Join Path problems can arise from the way that lookup and fact tables are related in a relational database.
- There are three types of Join Path problems:
 - Loops
 - Chasm traps
 - Fan traps
- In this course, we will discuss about Loops only.

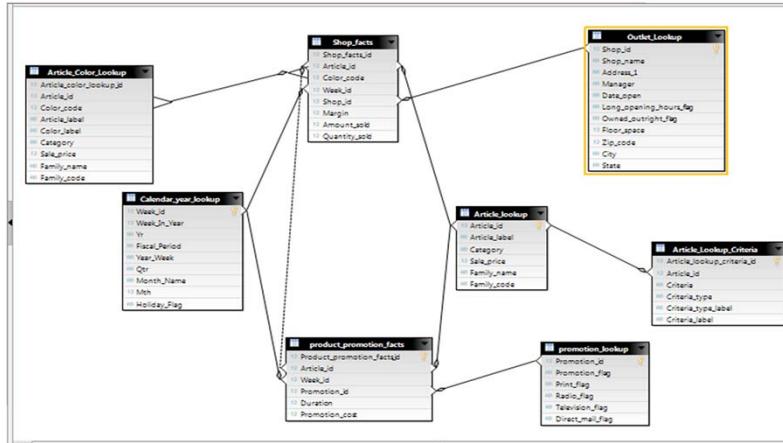


Copyright © Capgemini 2015. All Rights Reserved 4

3.3: What is a Loop?

Overview

- A Loop is caused by a circular set of Joins which define a closed path through a set of tables.



Universe
Structure
forming
a Loop

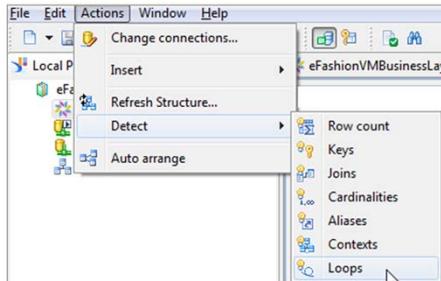
What is a Loop?

- A loop occurs in a Universe Structure, when there is continuous set of joins between a group of tables. Each individual join is necessary to satisfy specific User requirements. The combination of joins, however, forms a closed path which will cause problems when the SQL for a Query is generated.
- In a database, multiple paths between tables may be valid and implemented to meet specific user requirements. When each path is included individually in a query, it returns a distinct set of results.
- However, if a query includes more than one path, then the information returned can be incorrect. The rows that are returned are an intersection of the results for each path. Hence fewer rows are returned than expected. It is also often difficult to determine the problem when you examine the results. To help you detect potential loops in a Universe, Designer provides an automatic **loop detection** function.

3.3: What is a Loop?

Detecting Loops

- A Loop can be detected by selecting the data Foundation and Actions->Detect->Loops.



- A Loop can be detected by Visual analysis of schema, as well.

Capgemini CONSULTING TECHNOLOGY OUTSOURCING Copyright © Capgemini 2015. All Rights Reserved 6

Detecting Loops:

- Loops detect those problems for Designers for which Business Objects is designed to detect automatically. Business Objects then suggests appropriate structural resolutions that are required to guarantee correct query results for the User every time.
- Business Objects has different routines built into it. It uses these routines at various stages of the Loop detection / Resolution process. These are as follows:
 - > Detect Loops
 - > Check Integrity
 - > Detect Aliases
 - > Detect Contexts
- Loops can be identified with Visual analysis of schema, as well.

3.4: Resolving Loops by using Alias

Overview

- An Alias is a temporary name, which is assigned to a table, within an SQL Statement.
- An Alias breaks a Loop by using the same table twice in the same query for a different purpose.
- An Alias is identical to the base table but has a different name.
- The process of Alias detection is automated by Business Objects.



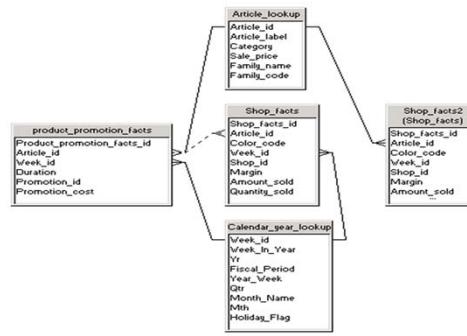
Copyright © Capgemini 2015. All Rights Reserved 7

Resolving Loops by using Alias:

- In SQL, an Alias is an alternative name for a table. The purpose of Aliases is to resolve structural issues in a database arising from SQL limitations.
- **For example:** One of the rules of SQL is that no table can be referenced more than once in the same SQL statement when each table is used for a different purpose.
- In Designer, an Alias is just a pointer to another table. A designer places one or more Aliases in the Structure pane so that Business Objects and Web Intelligence can generate the appropriate SQL statements for certain types of queries.

3.4: Resolving Loops by using Alias
Defining Objects from Aliases

- When Aliases are used in the Universe Structure window, it is essential that Objects are defined from the appropriate Alias.



Copyright © Capgemini 2015. All Rights Reserved 8

Defining Objects from Aliases:

- In a Business Objects Universe, Aliases are simply pointers to the original table.
- Aliases are put in place purely for the purpose of generating the appropriate SQL statements when certain query requests are made. However, in order to make them work, the Universe's Objects must be generated from the appropriate Aliases.

3.5: Resolving Loops by using Context

Resolving Loops: Using Contexts

- Aliases are not the best option to resolve the Loops.
- Certain database structures will require “Contexts” in the Universe Structures to resolve the loops.
 - Contexts are a collection of Joins which provide a valid query path to generate SQL.
 - The most common use of Contexts is to separate two query paths.
 - Contexts are used to direct Join Paths in a schema which contains multiple fact tables.



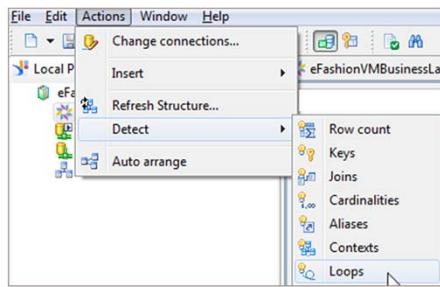
Copyright © Capgemini 2015. All Rights Reserved 9

Resolving Loops: Using Contexts:

- A loop situation is caused in a database structure if multiple fact tables in the schema share common “dimension” or “lookup tables”.
- A “context” is a rule that determines which of the two paths can be chosen when more than one path is possible in the database.
- With certain database structures, you may need to use contexts rather than Aliases to resolve loops. A situation where this commonly occurs is a transactional database with multiple fact tables (“multiple stars”) that share lookup tables.
- When a user runs a query from a Universe containing contexts, the **Desktop Intelligence** or **Web Intelligence** prompts the user to indicate the correct perspective for the query.

3.5: Resolving Loops by using Context Creating Context

- Designer can automatically detect Contexts, or a developer can manually create Contexts.
- You can use the auto detection of context while resolving a loop.



Copyright © Capgemini 2015. All Rights Reserved 10

Creating Context:

- Once the loop is identified, a developer can detect the possible set of contexts by choosing an Automated detection method.
- To start the context detection, select **Tools → Automated Detection → Detect Contexts**.

Demo on Aliases and Contexts

- Demo using Aliases and Contexts



3.6: Resolving Loops: Using Contexts

@Select

- @Select function enables you to re-use the SELECT statement of an existing object.
- @Select() cuts down development and maintenance periods.

The screenshot shows the Business Layer navigation pane on the left with objects like Time Period, Store, Product, Lines, Category, SKU Number, Label, SKU desc, Color, Unit Price MSRP, Extended Price, and Sold at (unit price). The main panel displays the 'Measure Properties' dialog for the 'Sold at (unit price)' measure. The 'Name' is set to 'Sold at (unit price)' and 'Active'. The 'Description' is 'This is the actual unit price per SKU obtained at sale time (i.e. Revenue/Quantity)'. The 'Data type' is 'Numeric' and the 'Aggregation function' is 'Sum'. The 'SQL definition' field contains the complex logic provided in the list item. Below it, there are tabs for 'Advanced', 'Source information', and 'Custom Properties'. The 'Where:' section is empty. The Capgemini logo is at the bottom left, and the page number 12 is at the bottom right.

@Select function:

- The @Select function allows a designer to re-use the SELECT statement of an existing Object. Suppose a highly complex statement needs to be re-used elsewhere in the Universe, with a different name attached to the Objects or with slight extensions to the SELECT clause. Then by virtue of the @Select function, the Designer does not have to create the object right from the scratch.
- @Select provides the benefit of inheritance. If several objects are based on some object through @Select, then any modification to the original object's SELECT clause leads to the modification in the derived object. This provides the designer with one central point to modify many objects in the Universe.

Syntax:

```
@Select(Classname\Objectname)
```

3.6: Resolving Loops: Using Contexts

@Where

- This function lets you re-use the WHERE clause of an existing object.
- @Where() can be used to re-use WHERE clauses from “dummy” Objects.
- Syntax:

```
@Where(Classname\ObjectName)
```



Copyright © Capgemini 2015. All Rights Reserved 13

@Where function:

- The @Where function allows a designer to re-use the WHERE statement of an existing object.
- A single Object can be defined that contains the requisite WHERE clause. Subsequently, any other objects that require it can be derived from this Object.
- This again centralizes maintenance of the WHERE clause for the whole Universe around a single original Object.

3.6: Resolving Loops: Using Contexts

Demo on Object Oriented Techniques

- Demo using Object-Oriented Techniques



Copyright © Capgemini 2015. All Rights Reserved 14

Summary

- In this lesson, you have learnt about:
 - The concept of a Loop
 - Automated routines to identify the loops
 - Methods to resolve loops by using Alias and Context
 - The concept of Reusability of objects in a Universe



Review Question

- Question 1: A circular set of join forms a loop.
 - True / False
- Question 2: ___ tables contain statistical information about transaction, while ___ tables contain information associated with a particular entity or subject.
- Question 3: When we create a table alias, a duplicate copy of the table is created.
 - True / False

