



BUSINESS INTELLIGENCE **FOR A PASSIONATE COMMUNITY**

[Implementing Aggregate Awareness in SAP BusinessObjects 3.1

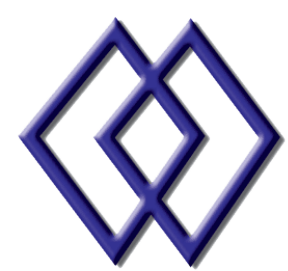
Michael Ward

Creative Technology and Training Solutions, LLC



Presentation Objectives/Agenda

- ◇ The importance of Aggregate Awareness, which takes advantage of database summary tables, speeding up report queries
- ◇ Summary tables and adding them to the universe
- ◇ The @Aggregate Aware Function, how it works, and how query efficiency is improved through its use
- ◇ Redefining objects: How to apply @Aggregate Aware to the Select statement SQL
- ◇ Detecting Incompatibility and setting objects to be compatible with the Summary tables
- ◇ Testing Aggregate Awareness with the report tool



Aggregate Awareness

- ◆ What Is Aggregate Awareness?
- ◆ Summary Tables
- ◆ Use @Aggregate_Aware Function
- ◆ Detect Incompatibility
- ◆ Aggregate Awareness Testing



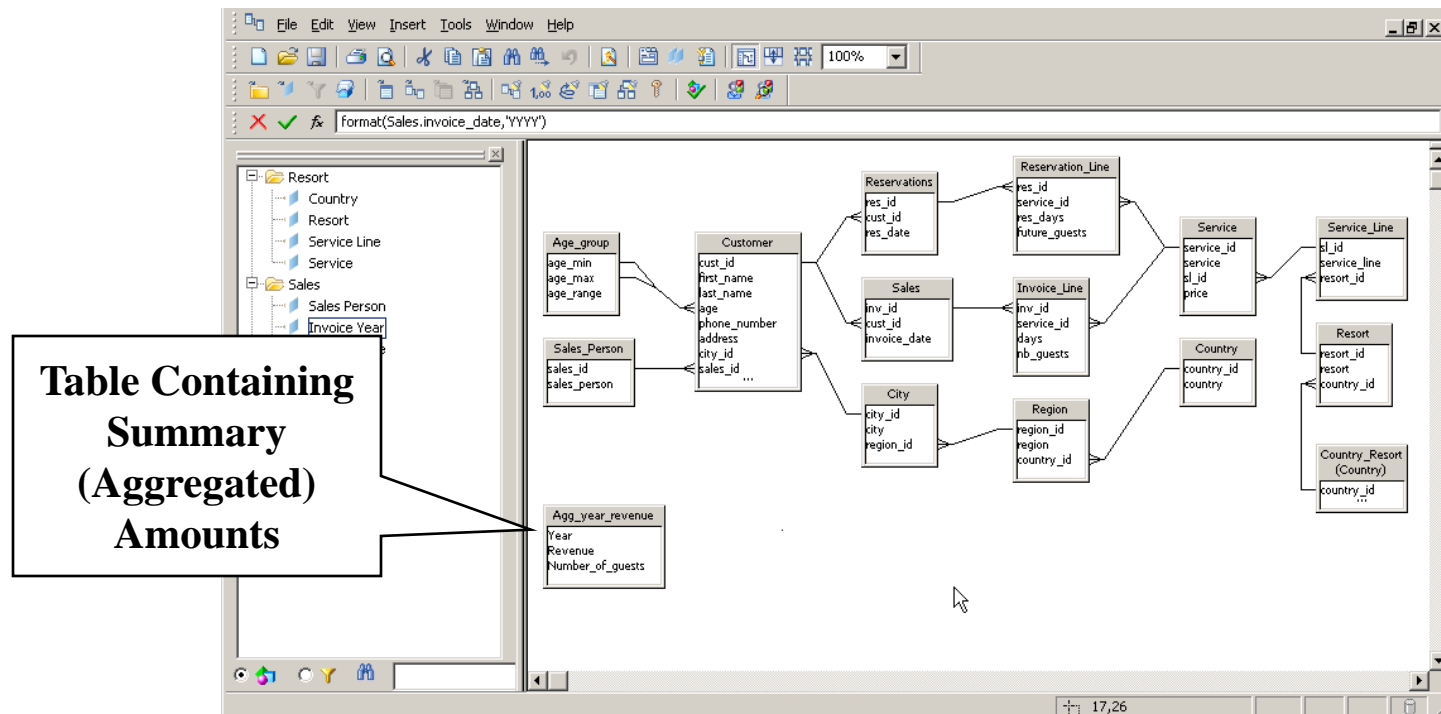
Aggregate Awareness

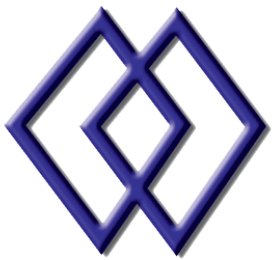
- ◆ Process That Incorporates Higher Level Aggregation (Summary Tables) into Universe
- ◆ Speeds up Queries That Contain Statistics Reported at Higher Levels (i.e. Year)

Aggregate Awareness

◆ Summary Tables

◇ May or May Not Actually Be Joined to Detail Tables in Universe





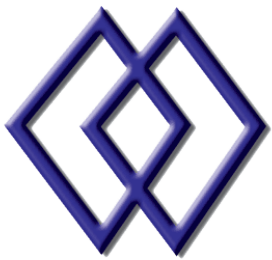
Summary Tables

- ◆ Store Data at Higher Level Than Detail (Fact) Tables
- ◆ Increase Processing Efficiency
 - ◇ Fewer Rows Retrieved
 - ◇ Few If Any Calculations Required
 - ◇ Few or No Joins



Apply Aggregate Awareness

- ◆ Insert Summary Tables into Universe
- ◆ Add Joins, Set Cardinality (If Needed)
- ◆ Redefine or Create Objects Using @Aggregate_Aware
- ◆ Define Compatible/Incompatible Objects Using Aggregate Navigation



@Aggregate_Aware

- ◆ Can Be Used with Aggregates (Measures) and Dimensions
- ◆ Selects Information from Most Aggregated to Least Aggregated (Summary to Detailed)

```
@Aggregate_Aware(sum(aggregate table1),...,sum(aggregate tableN)) Defines a measure object using precalculated aggregate tables.
```

Apply Aggregate Awareness

◆ Redefine Objects

4. Click *OK*

1. Edit Object to Redefine

Last Definition for @Aggregate_Aware Function Must Be Valid for All Queries

2. Redefine Select Statement Using @Aggregate_Aware Function, Summary Column First, Detail Column Second

3. *Parse* to Check Syntax

5. Click *OK*

Apply Aggregate Awareness

◆ Defining Compatible Objects

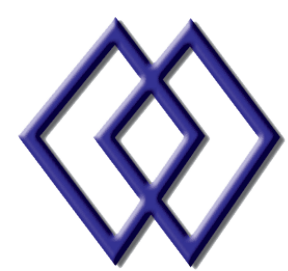
The screenshot shows the 'Aggregate Navigation' dialog box with the following components:

- Universe Tables:** A list of tables including Age_group, Agg_year_revenue, City, Country, Country_Resort, Customer, Invoice_Line, Region, Reservation_Line, Reservations, and Resort.
- Associated Incompatible Objects:** A list of objects including Resort, Sales, Customer, Reservations, Measures, Revenue, Future Guests, Future Revenue, and Number of Guests.
- Buttons:** Detect Incompatibility, OK, Cancel, and Help.
- Options:** Checkboxes for 'With incompatibles only' and 'Incompatibles only'.

Callout boxes provide the following instructions:

1. Select Tools → Aggregate Navigation
2. Select Summary Table
3. Click *Detect Incompatibility (Only Once)*
4. Incompatible Objects Are Displayed With Check Mark Next To Them
5. Uncheck Objects if Necessary
6. Click OK, Save Universe

Note: Summary Table Not Joined, All Other Tables Incompatible



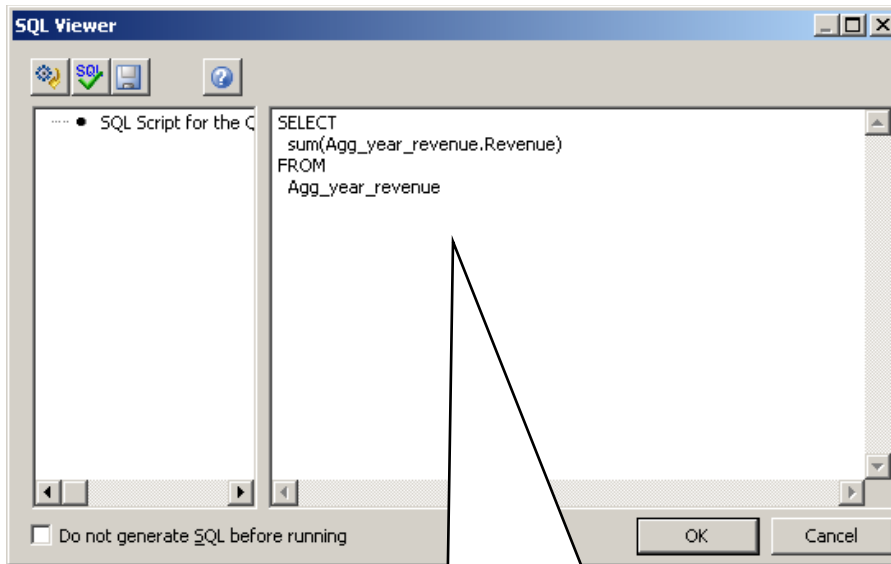
Detect Incompatibility

- ◆ Checks SQL Statement for Columns from Selected Table
- ◆ Check Incompatible Settings at least once
- ◆ Set On or Off Manually as Necessary
- ◆ Check All Necessary Tables

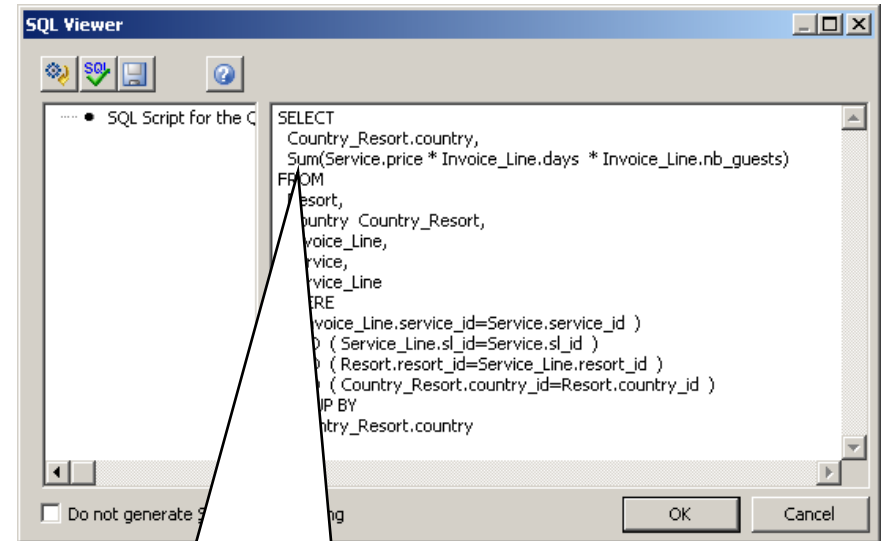


Aggregate Awareness

◆ Testing Using Desktop Intelligence™



**Only Revenue Selected,
Business Objects™ Uses
Summary Table**



**Resort and Revenue Selected,
Business Objects™ Uses
Detail Tables**

Aggregate Awareness

◆ Derived Tables

◇ SQL Used to Create View Like Table

The screenshot shows a database management tool interface. On the left, there is a 'Structure' window displaying a tree view of database objects including 'Resort', 'Sales', 'Customer', 'Reservations', and 'Measures'. A context menu is open over the 'Derived Tables...' option. The menu items include 'Edit Derived Tables...', 'Join...', 'Context...', 'Number of Rows in Table...', 'Change Table Display Ctrl+T', 'Gridlines', 'Page Breaks', and 'Options...'. A callout box with a pointer to the 'Derived Tables...' option contains the text: '1. Right Mouse Click In Structure Window and Select Derived Tables'.

Aggregate Awareness

◆ Derived Tables

The screenshot shows a software interface for creating a derived table. The main window is titled "Derived Tables" and contains the following elements:

- Derived Table:** A text box containing the name "AGG_YEAR_QTR_MONTH".
- Enter SQL Expression:** A text area containing the following SQL query:


```
SELECT
format(Sales.invoice_date,'YYYY') AS Year,
'Q' + format(Sales.invoice_date,'Q') as Quarter,
format(Sales.invoice_date,'MM') as Month,
Sum(Service.price * Invoice_Line.days * Invoice_Line.nb_guests) as Revenue,
Sum(Invoice_Line.nb_guests) AS Number_of_Guest
```
- SQL Expression Editor:** A panel with three columns:
 - Tables and Columns:** Lists "C:\BDC\Class(ClassData)\club" and its sub-items: "Age_group", "Agg_Year_Qtr_Revenue", "Agg_year_revenue", "City", and "Country".
 - Operators:** Lists logical operators: "<<", "<=", ">=", ">=", "AND", and "BETWEEN AND".
 - Functions:** Lists "Number", "Charact", "Date", and "@Funct".
- Description:** A text area for describing the table.
- Buttons:** "Check Syntax", "OK", "Cancel", and "Help".

Four numbered callouts are overlaid on the interface:

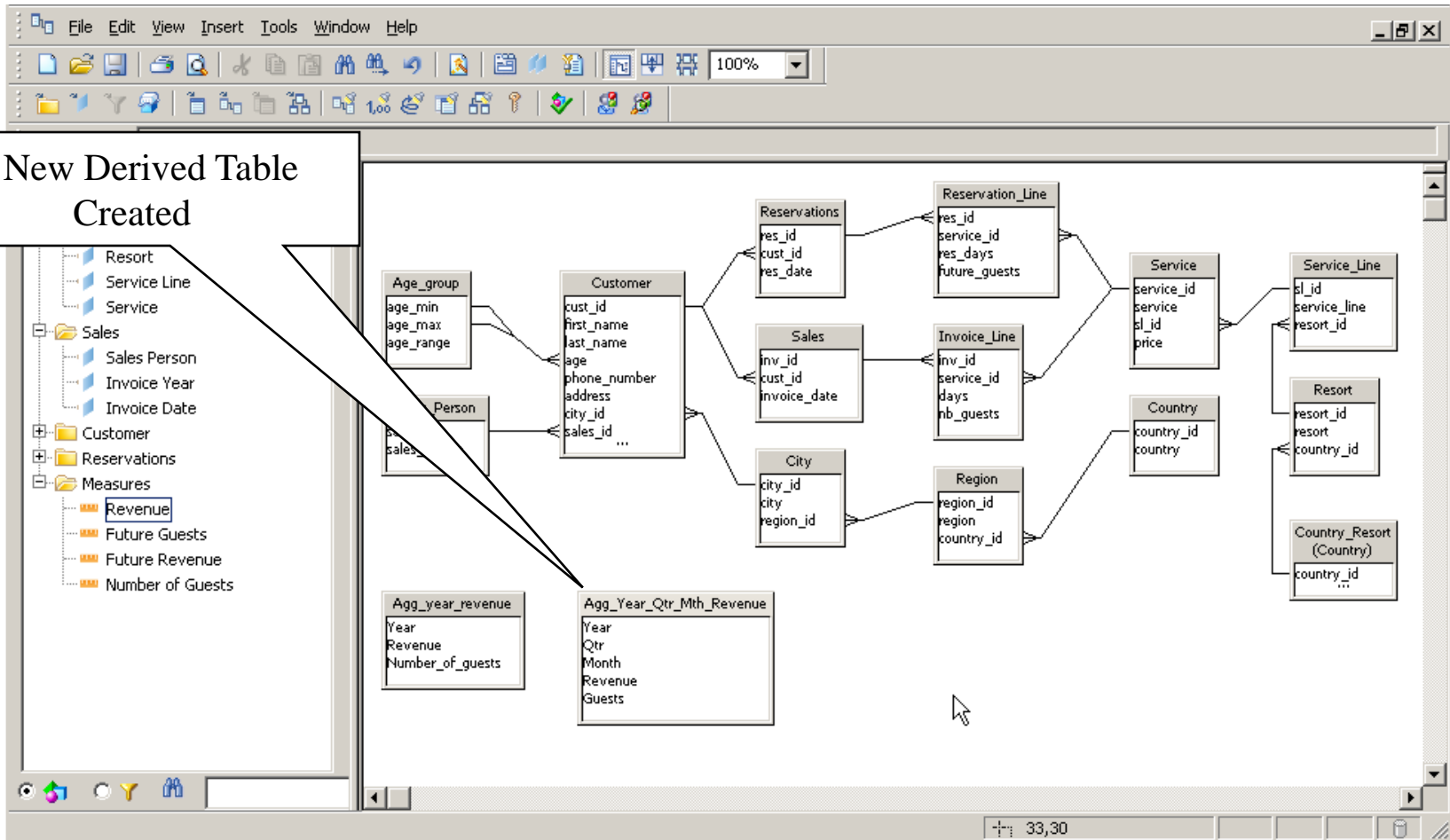
- 2. Name the Table:** Points to the "Derived Table" text box.
- 3. Enter the SQL or Copy SQL From File Using Copy/Paste:** Points to the "Enter SQL Expression" text area.
- 4. Check Syntax:** Points to the "Check Syntax" button.
- 5. Click OK:** Points to the "OK" button.

The background shows a data model with tables like "Resort", "Country", "Sales_Person", "Age_group", "Service_Line", and "Country_Resort". A "Measures" list on the left includes "Revenue", "Future Guests", "Future Revenue", and "Number of Guests".

Aggregate Awareness

◆ Adding to Aggregate Awareness

6. New Derived Table Created



Aggregate Awareness

◆ Adding to Aggregate Awareness

7. New Column from New Table added to @Aggregate Aware Objects

The screenshot displays the 'Edit Properties of Revenue' dialog box. The 'Definition' tab is active, showing the following fields:

- Name:** Revenue
- Type:** Number
- Description:** Total Revenue In Dollars
- Select:** @Aggregate_Aware(sum(Agg_year_revenue.Revenue),Sum(Agg_Year_Qtr_Mth_Revenue.Revenue),Sum(Service.price * Invoice_Line.days * Invoice_Line.nb_guests))
- Where:** (Empty)

The background shows a data model with the following tables and fields:

- Sales_Person:** sales_id, sales_person
- Customer:** cust_id, first_name, last_name, address, city_id, sales_id
- Agg_year_revenue:** Year, Revenue, Number_of_guests
- Agg_Year_Qtr_Mth_Revenue:** Year, Qtr, Month, Revenue, Guests

Aggregate Awareness

◆ Setting Aggregate Navigation

8. Select Tools / Aggregate Navigation

9. Select New Derived Table

10. Click *Detect Incompatibility (Only Once)*

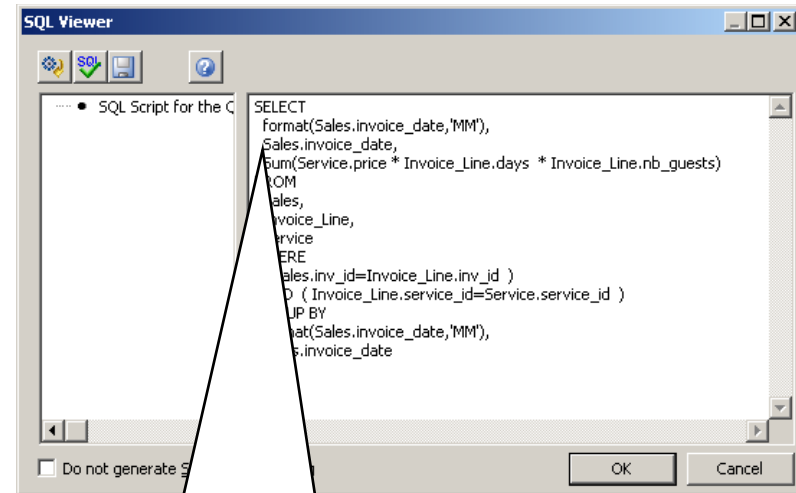
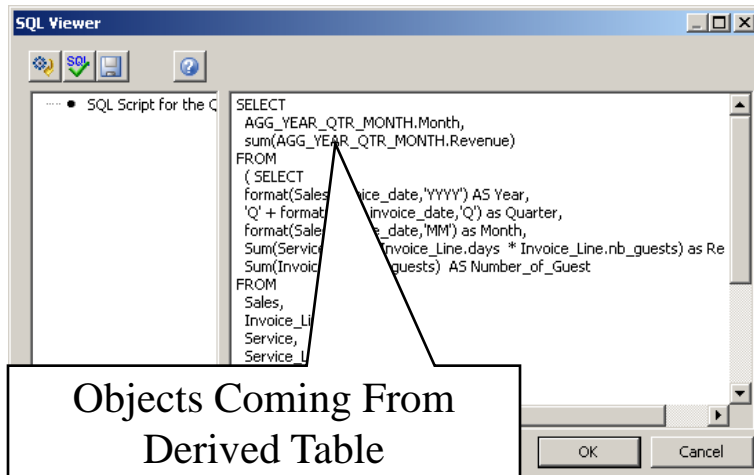
11. Uncheck Columns if Necessary

12. Click OK



Aggregate Awareness

◆ Test Using Desktop Intelligence



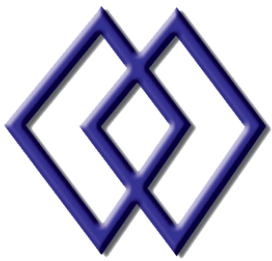
Note: Test Done
After Invoice Month
had been Made
Aggregate Aware
Using the New
Derived Table

Addition of Invoice Date
Causes Aggregate Aware
Objects to Use Fact Based
Columns



Presentation Objectives/Agenda

- ◇ The importance of Aggregate Awareness, which takes advantage of database summary tables, speeding up report queries
- ◇ Summary tables and adding them to the universe
- ◇ The @Aggregate Aware Function, how it works, and how query efficiency is improved through its use
- ◇ Redefining objects: How to apply @Aggregate Aware to the Select statement SQL
- ◇ Detecting Incompatibility and setting objects to be compatible with the Summary tables
- ◇ Testing Aggregate Awareness with the report tool



Contact Information:

Michael Ward
Creative Technology and Training Solutions, LLC
56132 Parkview
Shelby Twp., MI 48316

(586) 677-8300 Office
(586) 677-8301 Fax
(586) 484-5523 Cell

www.cttsbi.com

Email: MichaelWard@cttsbi.com

Thank you for participating.

Please remember to complete and return your evaluation form following this session.

For ongoing education on this area of focus, visit the Year-Round Community page at www.asug.com/yrcc

**SESSION CODE:
5004**