Advanced Aggregation: Rollup, Cube, Grouping Sets

Exercise Handout

QA

Contents

Overview

The main purpose of this activity is to familiarise yourself with how to use the advanced grouping clauses ROLLUP, CUBE and GROUPING SETS within TSQL.

Objectives

At the end of this activity, you will be able to use advanced grouping techniques: ROLLUP, CUBE and GROUPING SETS.

Setup: Launch SQL Server Management Studio

1. On the Start menu, click All Programs, and then click SQL Server Management Studio.
2. The Microsoft SQL Server Management Studio window opens, and then the Connect to Server dialog box will appear.
3. In the Connect to Server dialog box, click Connect to accept the default settings.
4. Open the provided script file QATSQLPLUS Setup.sql and run the script to create database QATSQLPLUS.

Exercise 1: Custom Subtotals

In this exercise, you will produce a summary of number of delegates with custom subtotals.

The main tasks for this exercise are as follows:

1. Create a query that returns list of events with a subtotal for VendorName, CourseName and StartDate.
2. Create a query that returns list of events with a subtotal for only VendorName, and CourseName.

Task 1: Using ROLLUP AND CUBE

1. Write a query to return all columns from the VendorCourseDateDelegateCount view. The columns returned should be:
    - VendorName
    - CourseName
    - StartDate
    - NumberDelegates
2. Test the query. The query should return 9 rows.
3. Change the query so that the NumberDelegates is aggregated using the SUM function, keeping the other columns in the SELECT list. Add a GROUP BY clause as appropriate.
4. Test the query. The query should still return 9 rows.
5. Change the query to add the WITH ROLLUP clause. Unlike most clauses in TSQL, the order of columns listed in the GROUP BY does matter when using ROLLUP.
6. Test the query. The query should return 20 rows. If the number of rows is not 20 then recheck the order of the columns in the GROUP BY clause. Extra rows will be added to summarise the combinations of:
    - VendorName, CourseName, StartDate
    - VendorName, CourseName
    - VendorName
    - Total
7. Modify the query by changing WITH ROLLUP to WITH CUBE.
8. Test the query. The query should now return 52 rows. The extra rows summarise all the combinations of VendorName, CourseName, StartDate and Total.
9. Keep the query window open for the following tasks.

Task 2: Use GROUPING SETS

1. Write a query to return all columns from the VendorCourseDateDelegateCount view. The columns returned should be:
    - VendorName
    - CourseName
    - StartDate
    - NumberDelegates
2. Test the query. This should return 9 rows.
3. Change the query so that the NumberDelegates is aggregated using the SUM function, keeping the other columns in the SELECT list. Add a GROUP BY clause as appropriate.
4. Test the query. This should still return 9 rows.
5. Change the query to add additional rows for the subtotals for Vendor only and VendorName, CourseName and StartDate. Hint: Grouping Sets.
6. Test the query. This should return 12 rows. An additional row should be added for each VendorName (with a vendor total).
7. The query window can be closed. Save the query if you wish.

Learn. To Change.

QA.com