Microsoft

Querying with Transact-SQL

Lab 8 – Grouping Sets and Pivoting Data

Overview

In this lab, you will use grouping sets and the PIVOT operator to summarize data in the **AdventureWorksLT** database.

What You'll Need

- An SQL Server Database instance with the **AdventureWorksLT** sample database.

Challenge 1: Retrieve Regional Sales Totals

Adventure Works sells products to customers in multiple country/regions around the world.

1. Retrieve totals for country/region and state/province

Tip: Review the documentation for GROUP BY in the Transact-SQL Language Reference.

An existing report uses the following query to return total sales revenue grouped by country/region and state/province.

```
SELECT a.CountryRegion, a.StateProvince, SUM(soh.TotalDue) AS Revenue
FROM SalesLT.Address AS a
JOIN SalesLT.CustomerAddress AS ca ON a.AddressID = ca.AddressID
JOIN SalesLT.Customer AS c ON ca.CustomerID = c.CustomerID
JOIN SalesLT.SalesOrderHeader as soh ON c.CustomerID = soh.CustomerID
GROUP BY a.CountryRegion, a.StateProvince
ORDER BY a.CountryRegion, a.StateProvince;
```

You have been asked to modify this query so that the results include a grand total for all sales revenue and a subtotal for each country/region in addition to the state/province subtotals that are already returned.

2. Indicate the grouping level in the results

Tip: Review the documentation for the GROUPING_ID function in the Transact-SQL Language Reference.

Modify your query to include a column named Level that indicates at which level in the total, country/region, and state/province hierarchy the revenue figure in the row is aggregated. For example, the grand total row should contain the value ‘Total’, the row showing the subtotal for United States should contain the value ‘United States Subtotal’, and the row showing the subtotal for California should contain the value ‘California Subtotal’.

3. Add a grouping level for cities

Extend your query to include a grouping for individual cities.

Challenge 2: Retrieve Customer Sales Revenue by Category

Adventure Works products are grouped into categories, which in turn have parent categories (defined in the **SalesLT.vGetAllCategories** view). Adventure Works customers are retail companies, and they may place orders for products of any category. The revenue for each product in an order is recorded as the **LineTotal** value in the **SalesLT.SalesOrderDetail** table.

1. Retrieve customer sales revenue for each parent category

Tip: Review the documentation for the PIVOT operator in the FROM clause syntax in the Transact-SQL language reference.

Retrieve a list of customer company names together with their total revenue for each parent category in *Accessories*, *Bikes*, *Clothing*, and *Components*.