Lab 6 – Using Subqueries and APPLY

OVERVIEW

In this lab, you will use subqueries and the APPLY operator to retrieve data from the **AdventureWorksLT** database.

Before starting this lab, you should view **Module 6 – Using Subqueries and APPLY** in the Course *Querying with Transact-SQL*. Then, if you have not already done so, follow the instructions in the **Getting Started** document for this course to set up the lab environment.

If you find some of the challenges difficult, don't worry – you can find suggested solutions for all of the challenges in the **Lab Solution** folder for this module.

WHAT YOU'LL NEED

- An Azure SQL Database instance with the **AdventureWorksLT** sample database. Review the **Getting Started** document for information about how to provision this.

CHALLENGE 1: RETRIEVE PRODUCT PRICE INFORMATION

Adventure Works products each have a standard cost price that indicates the cost of manufacturing the product, and a list price that indicates the recommended selling price for the product. This data is stored in the **SalesLT.Product** table. Whenever a product is ordered, the actual unit price at which it was sold is also recorded in the **SalesLT.SalesOrderDetail** table. You must use subqueries to compare the cost and list prices for each product with the unit prices charged in each sale.

Tip: Review the documentation for subqueries in Subquery Fundamentals.

1. Retrieve products whose list price is higher than the average unit price

Retrieve the product ID, name, and list price for each product where the list price is higher than the average unit price for all products that have been sold.

2. Retrieve Products with a list price of $100 or more that have been sold for less than $100

Retrieve the product ID, name, and list price for each product where the list price is $100 or more, and the product has been sold for less than $100.

3. Retrieve the cost, list price, and average selling price for each product

Retrieve the product ID, name, cost, and list price for each product along with the average unit price for which that product has been sold.

4. Retrieve products that have an average selling price that is lower than the cost

Filter your previous query to include only products where the cost price is higher than the average selling price.

CHALLENGE 2: RETRIEVE CUSTOMER INFORMATION

The **AdventureWorksLT** database includes a table-valued user-defined function named **dbo.ufnGetCustomerInformation**. You must use this function to retrieve details of customers based on customer ID values retrieved from tables in the database.

Tip: Review the documentation for the APPLY operator in Using APPLY.

1. Retrieve customer information for all sales orders

Retrieve the sales order ID, customer ID, first name, last name, and total due for all sales orders from the **SalesLT.SalesOrderHeader** table and the **dbo.ufnGetCustomerInformation** function.

2. Retrieve customer address information

Retrieve the customer ID, first name, last name, address line 1 and city for all customers from the **SalesLT.Address** and **SalesLT.CustomerAddress** tables, and the **dbo.ufnGetCustomerInformation** function.

NEXT STEPS

Well done! You've completed the lab, and you're ready to learn how to use table expressions in **Module 7 – Using Table Expressions** in the Course *Querying with Transact-SQL*.