Lab 3: Querying Multiple Tables with Joins

Generating Invoice Reports

AdventureWorks Cycles sells directly to retailers, who must be invoiced for their orders. You have been tasked with writing a query to generate a list of invoices to be sent to customers.

Instructions

Write a query that returns the company name from the `SalesLT.Customer` table, the sales order ID and total due from the `SalesLT.SalesOrderHeader` table. Make sure to use the aliases provided, and default column names elsewhere.

```
SELECT c.CompanyName, oh.SalesOrderID, oh.TotalDue

FROM SalesLT.Customer AS c

JOIN SalesLT.SalesOrderHeader AS oh

ON c.CustomerID = oh.CustomerID;
```

Generating Invoice Reports (2)

In order to send out invoices to the customers, you'll need their addresses.

Instructions

Extend your customer orders query to include the main office address for each customer, including the full street address, city, state or province, postal code, and country or region. Make sure to use the aliases provided, and default column names elsewhere.

```
SELECT c.CompanyName, a.AddressLine1, ISNULL(a.AddressLine2, '') AS AddressLine2, a.City,
a.StateProvince, a.PostalCode, a.CountryRegion, oh.SalesOrderID, oh.TotalDue

FROM SalesLT.Customer AS c

JOIN SalesLT.SalesOrderHeader AS oh

ON oh.CustomerID = c.CustomerID

JOIN SalesLT.CustomerAddress AS ca

ON c.CustomerID = ca.CustomerID AND AddressType = 'Main Office'

JOIN SalesLT.Address AS a

ON ca.AddressID = a.AddressID;
```

Retrieving Sales Data

The sales manager wants a list of all customer companies and their contacts (first name and last name), showing the sales order ID and total due for each order they have placed.

Instructions

Customers who have not placed any orders should be included at the bottom of the list with `NULL` values for the order ID and total due. Make sure to use the aliases provided, and default column names elsewhere.

```
SELECT c.FirstName, c.LastName, oh.SalesOrderID, oh.TotalDue, c.CompanyName

FROM SalesLT.Customer AS c

LEFT JOIN SalesLT.SalesOrderHeader AS oh

ON c.CustomerID = oh.CustomerID

ORDER BY oh.SalesOrderID DESC;
```

Retrieving Sales Data (2)

A sales employee has noticed that AdventureWorks does not have address information for all customers.

Instructions

Write a query that returns a list of customer IDs, company names, contact names (first name and last name), and phone numbers for customers with no address stored in the database. Make sure to use the aliases provided, and default column names elsewhere.

```
SELECT c.CompanyName, c.FirstName, c.LastName, c.Phone

FROM SalesLT.Customer AS c

LEFT JOIN SalesLT.CustomerAddress AS ca

ON c.CustomerID = ca.CustomerID

WHERE ca.CustomerID IS NULL;
```

Retrieving Sales Data (3)

Some customers have never placed orders, and some products have never been ordered.

Instructions

- Write a query that returns a column of customer IDs for customers who have never placed an order, and a column of product IDs for products that have never been ordered.
- Each row with a customer ID should have a `NULL` product ID (because the customer has never ordered a product) and each row with a product ID should have a `NULL` customer ID (because the product has never been ordered by a customer).
- Make sure to use the aliases provided, and default column names elsewhere.

```
SELECT c.CustomerID, p.ProductID

FROM SalesLT.Customer AS c

FULL JOIN SalesLT.SalesOrderHeader AS oh

ON c.CustomerID = oh.CustomerID

FULL JOIN SalesLT.SalesOrderDetail AS od

ON od.SalesOrderID = oh.SalesOrderID

FULL JOIN SalesLT.Product AS p

ON p.ProductID = od.ProductID

WHERE oh.SalesOrderID IS NULL

ORDER BY ProductID, CustomerID;
```