Lab 1: Introduction to Transact-SQL

Retrieving Customer Data

AdventureWorks Cycles is a company that sells directly to retailers, who then sell products to consumers. Each retailer that is an AdventureWorks customer has provided a named contact for all communication from AdventureWorks.

The sales manager at AdventureWorks has asked you to generate some reports containing details of the company's customers to support a direct sales campaign. Let's start with some basic exploration.

Instructions

Familiarize yourself with the `Customer` table by writing a Transact-SQL query that retrieves all columns for all customers.

```
SELECT *

FROM SalesLT.Customer;
```

Create List of Customer Contacts

As a next step, it would be good to have a structured view of the names of your customer contacts.

Instructions

Create a table that lists all customer contact names. The table should include the `Title`, `FirstName`, `MiddleName`, `LastName` and `Suffix` of all customers.

```
SELECT Title, FirstName, MiddleName, LastName , Suffix

FROM SalesLT.Customer;
```

Create List of Customer Contacts (2)

Each customer has an assigned salesperson. Can you finish the query to include the salesperson and a nicely structured display of the customers' names?

Instructions

- Complete the query to list the following elements for all customers:
    - The salesperson
    - A column named `CustomerName` that displays how the customer contact should be greeted (e.g. "Mr Smith").
    - The customer's phone number (`Phone`)

- Don't forget to space out the contents of your `CustomerName` column with `+ ' ' +` and use the alias provided.

SELECT Salesperson , Title + ' ' + LastName AS CustomerName, Phone

FROM SalesLT.Customer;

Retrieving Customer and Sales Data

As a reminder, if you want to build a string with an integer (e.g. `id`) you can use:

```
CAST(id AS VARCHAR)
```

Put it to the test; as you continue to work with the AdventureWorks customer data, you must create queries for reports that have been requested by the sales team.

Instructions

Provide a list of all customer companies in the format `<Customer ID>: <Company Name>` (e.g. `78: Preferred Bikes`). You'll need to use `VARCHAR` in your solution. Don't forget to use the alias provided.

SELECT CAST(CustomerID AS VARCHAR(10)) + ': ' + CompanyName AS CustomerCompany

FROM SalesLT.Customer;

Retrieving Customer and Sales Data (2)

The `SalesLT.SalesOrderHeader` table contains records of sales orders. You have been asked to retrieve data for a report that shows:

- The sales order number and revision number in the format `<Order Number> (<Revision>)` (e.g. `SO71774 (2)`).
- The order date converted to ANSI standard format `yyyy.mm.dd` (e.g. `2015.01.31`).

Instructions

Complete the query on the right to create the 2-column table that's specified above.

SELECT SalesOrderNumber + ' (' + STR(RevisionNumber, 1) + ')' AS OrderRevision,

CONVERT(nvarchar(30), OrderDate, 102) AS OrderDate

FROM SalesLT.SalesOrderHeader;

Retrieving Customer Contact Names

In this exercise, you'll write a query that returns a list of customer names. The list must consist of a single field in the format `<first name> <last name>` (e.g. Keith Harris) if the middle name is unknown, or `<first name> <middle name> <last name>` (e.g. Jane M. Gates) if a middle name is stored in the database.

Instructions

Retrieve customer contact names including middle names when they're known.

SELECT FirstName + ' ' + ISNULL(MiddleName + ' ', '') + LastName

AS CustomerName

FROM SalesLT.Customer;

Retrieving Primary Contact Details

Customers may provide AdventureWorks with an email address, a phone number, or both. If an email address is available, then it should be used as the primary contact method; if not, then the phone number should be used. Here, you will write a query that returns a list of customer IDs in one column, and a second column named `PrimaryContact` that contains the email address if known, and otherwise the phone number.

Note: In the sample data provided in `AdventureWorksLT`, there are no customer records without an email address. Therefore, to verify that your query works as expected, we have run the following `UPDATE` statement to remove some existing email addresses before you write your query. (Don't worry, you'll learn about `UPDATE`statements later in the course!)

```
UPDATE SalesLT.Customer
SET EmailAddress = NULL
WHERE CustomerID % 7 = 1;
```

Instructions

Write a query that returns a list of customer IDs in one column, and a second column called `PrimaryContact` that contains the email address if known, and otherwise the phone number.

SELECT CustomerID, COALESCE(EmailAddress,Phone) AS PrimaryContact

FROM SalesLT.Customer;

Retrieving Shipping Status

You have been asked to create a query that returns a list of sales order IDs and order dates with a column named `ShippingStatus` that contains the text "Shipped" for orders with a known ship date, and "Awaiting Shipment" for orders with no ship date.

Note: In the sample data provided in `AdventureWorksLT`, there are no sales order header records without a ship date. Therefore, to verify that your query works as expected, we have run the following `UPDATE` statement to remove some existing ship dates before you write your query. (Again, don't worry, you'll learn about `UPDATE` statements later in the course!)

```
UPDATE SalesLT.SalesOrderHeader
```

```
SET ShipDate = NULL
WHERE SalesOrderID > 71899;
```

Instructions

Write a query to list sales order IDs and order dates with a column named `ShippingStatus` that contains the text 'Shipped' for orders with a known ship date, and 'Awaiting Shipment' for orders with no ship date.

```
SELECT SalesOrderID, OrderDate,
  CASE
    WHEN ShipDate IS NULL THEN 'Awaiting Shipment'
    ELSE 'Shipped'
  END AS ShippingStatus
FROM SalesLT.SalesOrderHeader;
```