Assignment 2

Tianle Yuan

04/12/2023

-- Write queries for following scenarios1.

1. How many products can you find in the Production.Product table?

I can find 504 products in the table. Below is the code I wrote to get the result:

```
USE AdventureWorks2019
GO
SELECT COUNT(ProductID)
FROM Production.Product
```

2. Write a query that retrieves the number of products in the Production.Product table that are included in a subcategory. The rows that have NULL in column ProductSubcategoryID are considered to not be a part of any subcategory.

```
SELECT COUNT(ProductID)
FROM Production.Product
WHERE ProductSubcategoryID IS NOT NULL
```

3. How many Products reside in each SubCategory? Write a query to display the results with the following titles.

ProductSubcategoryID CountedProducts

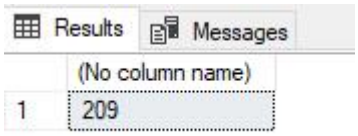
---------- -----------

```
SELECT ProductSubcategoryID, COUNT(ProductID) AS CountedProducts
FROM Production.Product
WHERE ProductSubcategoryID IS NOT NULL
GROUP BY ProductSubcategoryID
```

4. How many products that do not have a product subcategory.

I can find 504 products do not have a product subcategory. Below is the code I wrote to get the result:

```
SELECT COUNT(ProductID)
FROM Production.Product
WHERE ProductSubcategoryID IS NULL
```



5. Write a query to list the sum of products quantity in the Production.ProductInventory table.

```
SELECT SUM(Quantity)
FROM Production.ProductInventory
```

6. Write a query to list the sum of products in the Production.ProductInventory table and LocationID set to 40 and limit the result to include just summarized quantities less than 100.

```
ProductID       TheSum
-----------     ----------
```

```
SELECT ProductID, SUM(Quantity) AS TheSum
FROM Production.ProductInventory
WHERE LocationID = 40
GROUP BY ProductID
HAVING SUM(Quantity) < 100
```

7. Write a query to list the sum of products with the shelf information in the Production.ProductInventory table and LocationID set to 40 and limit the result to include just summarized quantities less than 100

```
Shelf       ProductID       TheSum
----------  -----------     -----------
```

```
SELECT Shelf, ProductID, SUM(Quantity) AS TheSum
FROM Production.ProductInventory
WHERE LocationID = 40
GROUP BY Shelf, ProductID
HAVING SUM(Quantity) < 100
```

8. Write the query to list the average quantity for products where column LocationID has the value of 10 from the table Production.ProductInventory table.

```
SELECT AVG(Quantity)
FROM Production.ProductInventory
WHERE LocationID = 10
```

9. Write query to see the average quantity of products by shelf from the table Production.ProductInventory

```
ProductID   Shelf       TheAvg
----------- ---------- -----------
```

```
SELECT ProductID, Shelf, AVG(Quantity) AS TheAvg
```

```
FROM Production.ProductInventory
GROUP BY ProductID, Shelf
```

10. Write query to see the average quantity of products by shelf excluding rows that has the value of N/A in the column Shelf from the table Production.ProductInventory

ProductID Shelf TheAvg

---------- ---------- ----------

```
SELECT ProductID, Shelf, AVG(Quantity) AS TheAvg
FROM Production.ProductInventory
WHERE Shelf <> 'N/A'
GROUP BY ProductID, Shelf
```

11. List the members (rows) and average list price in the Production.Product table. This should be grouped independently over the Color and the Class column. Exclude the rows where Color or Class are null.

Color Class TheCount AvgPrice

--------------- - ----- ---------- ---------- ---------

```
SELECT Color, Class, COUNT(*) AS TheCount, AVG(ListPrice) AS TheAvg
FROM Production.Product
WHERE Color IS NOT NULL and Class IS NOT NULL
GROUP BY Color, Class
```

Joins:

12. Write a query that lists the country and province names from person. CountryRegion and person. StateProvince tables. Join them and produce a result set similar to the following.

Country Province

--------- --------------------

```
SELECT CR.Name AS Country, SP.Name AS Province
FROM Person.CountryRegion CR INNER JOIN Person.StateProvince SP
ON CR.CountryRegionCode = SP.CountryRegionCode
```

13. Write a query that lists the country and province names from person. CountryRegion and person. StateProvince tables and list the countries filter them by Germany and Canada. Join them and produce a result set similar to the following.

Country Province

--------- --------------------

```
SELECT CR.Name AS Country, SP.Name AS Province
FROM Person.CountryRegion CR INNER JOIN Person.StateProvince SP
ON CR.CountryRegionCode = SP.CountryRegionCode
WHERE CR.Name IN ('Germany', 'Canada')
```

Using Northwnd Database: (Use aliases for all the Joins)

```
USE Northwind
GO
```

14. List all Products that has been sold at least once in last 25 years.

```
SELECT DISTINCT P.ProductID, P.ProductName
FROM dbo.Products P
INNER JOIN dbo.[Order Details] OD ON P.ProductID = OD.ProductID
INNER JOIN dbo.Orders O ON OD.OrderID = O.OrderID
WHERE O.OrderDate >= DATEADD(year, -25, GETDATE())
ORDER BY P.ProductID;
```

15. List top 5 locations (Zip Code) where the products sold most.

```
SELECT TOP(5) O.ShipPostalCode, COUNT(P.ProductID)
FROM dbo.Products P
INNER JOIN dbo.[Order Details] OD ON P.ProductID = OD.ProductID
INNER JOIN dbo.Orders O ON OD.OrderID = O.OrderID
WHERE O.ShipPostalCode IS NOT NULL
GROUP BY O.ShipPostalCode
ORDER BY COUNT(P.ProductID) DESC;
```

16. List top 5 locations (Zip Code) where the products sold most in last 25 years.

```
SELECT TOP(5) O.ShipPostalCode, COUNT(P.ProductID)
FROM dbo.Products P
INNER JOIN dbo.[Order Details] OD ON P.ProductID = OD.ProductID
INNER JOIN dbo.Orders O ON OD.OrderID = O.OrderID
WHERE O.ShipPostalCode IS NOT NULL AND O.OrderDate >= DATEADD(YEAR, -25, GETDATE())
GROUP BY O.ShipPostalCode
ORDER BY COUNT(P.ProductID) DESC;
```

17. List all city names and number of customers in that city.

```
SELECT City, COUNT(*) AS CustomerCount
FROM Customers
GROUP BY City
ORDER BY CustomerCount DESC;
```

18. List city names which have more than 2 customers, and number of customers in that city

```
SELECT City, COUNT(CustomerID) AS CustomerCount
FROM Customers
GROUP BY City
```

```
HAVING COUNT(CustomerID) > 2
ORDER BY CustomerCount DESC;
```

19. List the names of customers who placed orders after 1/1/98 with order date.

```
SELECT C.ContactName, O.OrderDate
FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
WHERE O.OrderDate >= '1/1/98'
```

20. List the names of all customers with most recent order dates

```
SELECT C.ContactName, O.OrderDate
FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
WHERE O.OrderDate = (
    SELECT MAX(OrderDate)
    FROM Orders
)
```

21. Display the names of all customers along with the count of products they bought

```
SELECT C.ContactName, COUNT(O.OrderID) AS ProductCount
FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
GROUP BY C.ContactName
```

22. Display the customer ids who bought more than 100 Products with count of products.

```
SELECT C.CustomerID, COUNT(O.OrderID) AS ProductCount
FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID
HAVING COUNT(O.OrderID) > 100
```

23. List all of the possible ways that suppliers can ship their products. Display the results as below

Supplier Company Name **Shipping Company Name**

```
SELECT DISTINCT SU.CompanyName, SH.CompanyName
FROM Suppliers AS SU
INNER JOIN Products AS P ON SU.SupplierID= P.SupplierID
INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
INNER JOIN Shippers AS SH ON O.ShipVia = SH.ShipperID
```

24. Display the products order each day. Show Order date and Product Name.

```
SELECT P.ProductName, O.OrderDate
FROM Products AS P
```

```
INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
INNER JOIN Orders AS O ON OD.OrderID = O.OrderID
ORDER BY O.OrderDate
```

25. Displays pairs of employees who have the same job title.

```
SELECT A.FirstName + ' ' + A.LastName AS EmployeeA, B.FirstName + ' ' + B.LastName  AS
EmployeeB
FROM Employees A INNER JOIN Employees B ON A.Title = B.Title
```

26. Display all the Managers who have more than 2 employees reporting to them.

```
SELECT M.EmployeeID, M.FirstName + ' ' + M.LastName AS MName, COUNT(*) AS NumEmployees
FROM Employees AS M INNER JOIN Employees AS E ON M.EmployeeID = E.ReportsTo
GROUP BY M.EmployeeID, M.FirstName + ' ' + M.LastName
HAVING COUNT(*) > 2
```

27. Display the customers and suppliers by city. The results should have the following columns

City

Name

Contact Name,

Type (Customer or Supplier)

```
SELECT City, CompanyName, ContactName, 'Customer' AS Type
FROM Customers
UNION
SELECT City, CompanyName, ContactName, 'Supplier' AS Type
FROM Suppliers
ORDER BY City, CompanyName
```