Assignment Day2 –SQL: Comprehensive practice

Amber Han

Answer following questions

1. What is a result set?

   An SQL result set is a set of rows from a database, as well as metadata about the query such as the column names, and the types and sizes of each column.

2. What is the difference between Union and Union All?

   UNION performs a deduplication step before returning the final results, UNION ALL retains all duplicates and returns the full, concatenated results.

3. What are the other Set Operators SQL Server has?

   Intersect, Except.

4. What is the difference between Union and Join?

   a join is used to combine columns from different tables, the union is used to combine rows.

5. What is the difference between INNER JOIN and FULL JOIN?

   A full join will return rows from both tables even if there are no matching data. Inner join will return only matched rows.

6. What is difference between left join and outer join

   Left join is part of outer join. Outer join has left join, right join, and full join.

7. What is cross join?

   Cross join returns the Cartesian product of the sets of records from the two joined tables.

8. What is the difference between WHERE clause and HAVING clause?

   A WHERE clause is used to filter records from results. The filter occurs before any groupings are made.

A HAVING clause is used to filter values from a group.

9. Can there be multiple group by columns?
   A GROUP BY clause can contain 2 or more columns, or in other words, a group can consists of 2 or more columns.

Write queries for following scenarios

1. How many products can you find in the Production.Product table?
   504
   ```
   SELECT COUNT(*) 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(*) 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(ProductSubcategoryID) AS CountedProducts
FROM Production.Product
WHERE ProductSubcategoryID IS NOT NULL
GROUP BY ProductSubcategoryID
```

4. How many products that do not have a product subcategory.
   ```
   SELECT COUNT(*)
   FROM Production.Product
   WHERE ProductSubcategoryID IS NULL
   ```

5. Write a query to list the summary of products quantity in the Production.ProductInventory table.
   ```
   SELECT * FROM Production.ProductInventory
   ```

6. Write a query to list the summary 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 summary 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 ProductID, Shelf
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) AS TheAvg
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 ROLLUP(Shelf, ProductID)
```

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 ROLLUP(Shelf, ProductID)
ORDER BY 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

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

Joins:

```
SELECT Color, Class, COUNT(*) AS TheCount, AVG(ListPrice) AS AvgPrice
FROM Production.Product
WHERE Class IS NOT NULL AND Color IS NOT NULL
GROUP BY GROUPING SETS ((Color), (Class))
```

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 DISTINCT c.Name AS Country, s.Name AS Province
FROM Person.StateProvince s
INNER JOIN Person.CountryRegion c
ON s.CountryRegionCode = c.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 DISTINCT c.Name AS Country, s.Name AS Province
FROM Person.StateProvince s
INNER JOIN Person.CountryRegion c
ON s.CountryRegionCode = c.CountryRegionCode
WHERE c.Name = 'Germany' OR c.Name = 'Canada'
```

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

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

```
SELECT DISTINCT p.ProductName
FROM Products p INNER JOIN [Order Details] o
ON p.ProductID = o.ProductID
INNER JOIN Orders r
```

```
ON r.OrderID = o.OrderID
Where r.OrderDate BETWEEN '1995-10-21' AND ' 2020-10-21'
```

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

```
SELECT TOP 5 ShipPostalCode FROM Orders
GROUP BY ShipPostalCode
ORDER BY COUNT(ShipPostalCode) DESC
```

16. List top 5 locations (Zip Code) where the products sold most in last 20 years.

```
SELECT TOP 5 ShipPostalCode FROM Orders
Where OrderDate BETWEEN '2000-10-21' AND ' 2020-10-21'
GROUP BY ShipPostalCode
ORDER BY COUNT(ShipPostalCode) DESC
```

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

```
SELECT City, COUNT(ContactName) AS 'Number Of Customers'
FROM Customers
GROUP BY City
```

18. List city names which have more than 10 customers, and number of customers in that city

```
SELECT City, COUNT(ContactName) AS 'Number Of Customers'
FROM Customers
GROUP BY City
HAVING COUNT(ContactName) > 10
```

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

```
SELECT DISTINCT c.ContactName FROM Orders o INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
WHERE OrderDate BETWEEN '1998-01-01' AND '2020-10-21'
```

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

```
SELECT CustomerID, OrderDate
FROM (SELECT DISTINCT CustomerID, OrderDate, dense_rank() OVER (PARTITION BY
CustomerID ORDER BY orderDate DESC) rnk FROM Orders)dt
WHERE dt.rnk = 1
```

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

```
SELECT  c.ContactName, COUNT(c.ContactName) AS 'Count'
FROM Orders o INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
GROUP BY c.ContactName
ORDER BY COUNT(c.ContactName) DESC
```

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

```
SELECT  c.ContactName, SUM(r.Quantity) AS 'SUM'
FROM Orders o INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
INNER JOIN [Order Details] r
ON r.OrderID = o.OrderID
```

```
GROUP BY c.ContactName
HAVING SUM(r.Quantity) > 100
ORDER BY SUM(r.Quantity) DESC
```

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 u.CompanyName AS 'Supplier Company Name', s.CompanyName AS 'Shipping Company Name'
FROM Shippers s CROSS JOIN Suppliers u
```

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

```
SELECT DISTINCT r.OrderDate, p.ProductName
FROM Products p INNER JOIN [Order Details] o
ON p.ProductID = o.ProductID
INNER JOIN Orders r
ON r.OrderID = o.OrderID
```

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

```
SELECT * FROM Employees e INNER JOIN Employees m
ON e.Title = m.Title
```

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

```
SELECT e.EmployeeID, e.LastName, e.FirstName, e.Title FROM Employees e INNER JOIN
Employees m
ON e.EmployeeID = m.ReportsTo
WHERE e.Title LIKE '%manager%'
GROUP BY e.EmployeeID, e.LastName, e.FirstName, e.Title
HAVING COUNT(m.ReportsTo) > 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, ContactName, 'Customer' AS TYPE FROM Customers
UNION
SELECT city, ContactName, 'Supplier' AS TYPE FROM Suppliers
```

28. Have two tables T1 and T2

F1.T1	F2.T2
1	2

2	3
3	4

Please write a query to inner join these two tables and write down the result of this query.

```
SELECT * FROM F1 INNER JOIN F2 ON F1.T1 = F2.T2
F1.T1           F2.T2
2                 2
3                 3
```

29. Based on above two table, Please write a query to left outer join these two tables and write down the result of this query.

SELECT * FROM F1 LEFT JOIN F2 ON F1.T1 = F2.T2

```
F1.T1           F2.T2
1                 null
2                 2
3                 3
```

GOOD LUCK.