

## Assignment Day2 –SQL: Comprehensive practice

### Answer following questions

1. What is a result set?

A result set is a table that includes a set of records. The records are fetched by query statements from the database that is being used. A result set could include 0 or multiple records.

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

Union removes duplicate records and returns distinct records, which takes extra time compared to Union All. Union All does not remove duplicate records and returns all the records from all queries.

And Union cannot be inside a recursive cte(common table expression) but Union All can be.

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

Intersect: It returns the records that are selected by all queries.

Except: It returns the records selected by the query on the left side that are not represented in the query on the right side of "Except".

4. What is the difference between Union and Join?

Union combines rows from multiple result sets vertically and removes duplicate ones. Number of columns should be same. Join combines the result sets of matching records from multiple tables horizontally.

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

Inner join returns matching rows and eliminates non-matching rows. Full join returns all rows from both the left and right tables whether matchable or not.

6. What is difference between left join and outer join

Left join is a part of outer join, which returns matching rows and only the remaining rows from the left table. Outer join consists of left join, right join and full join.

7. What is cross join?

It is a type of join that returns the Cartesian product of rows from the tables in the join. It combines each from the left table with each row from the right table.

8. What is the difference between WHERE clause and HAVING clause?  
Where clause is used to filter records before groups are made. Having clause is used to filter records after they are grouped.
9. Can there be multiple group by columns?  
Yes, group by clause can take one or more fields.

### Write queries for following scenarios

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

```
select count(1)
from Production.Product p
Answer: 504
```

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(1)
from Production.Product p
where p.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 p.ProductSubcategoryID, count(p.ProductSubcategoryID) CountedProducts
from Production.Product p
where p.ProductSubcategoryID is not null
group by p.ProductSubcategoryID
```

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

```
select count(1)
from Production.Product p
where p.ProductSubcategoryID is null
```

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

```
select p.ProductID, sum(p.Quantity) TheSum
from Production.ProductInventory p
group by p.ProductID
```

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 p.ProductID, SUM(p.Quantity) TheSum
from Production.ProductInventory p
where p.LocationID = 40
group by p.ProductID
having SUM(p.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 p.Shelf, p.ProductID, SUM(p.Quantity) TheSum
from Production.ProductInventory p
where p.LocationID = 40
group by p.ProductID, p.Shelf
having SUM(p.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(p.Quantity)
from Production.ProductInventory p
where p.LocationID = 10
```

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

ProductID Shelf TheAvg

-----

```
select p.ProductID, p.Shelf, AVG(p.Quantity) TheAvg
from Production.ProductInventory p
group by rollup(p.Shelf, p.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 p.ProductID, p.Shelf, AVG(p.Quantity) TheAvg
from Production.ProductInventory p
where p.Shelf != 'N/A'
group by rollup(p.Shelf, p.ProductID)

```

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 p.Color, p.Class, count(1) TheCount, AVG(p.ListPrice) AvgPrice
from Production.Product p
where p.Class is not null and p.Color is not null
group by grouping sets (p.Color, p.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 c.Name Country, s.Name Province
from Person.CountryRegion c join Person.StateProvince s
on c.CountryRegionCode = s.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 c.Name Country, s.Name Province
from Person.CountryRegion c join Person.StateProvince s
on c.CountryRegionCode = s.CountryRegionCode
where c.name in ('Germany', 'Canada')
order by c.name

```

### 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 dbo.Products p
join dbo.[Order Details] od
on p.ProductID = od.ProductID
join dbo.Orders o
on od.OrderID = o.OrderID
where o.OrderDate between '1996-09-16' and '2021-09-16'
```

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

```
select top 5 o.ShipPostalCode
from Orders o
group by o.ShipPostalCode
order by count(o.ShipPostalCode) desc
```

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

```
select top 5 o.ShipPostalCode
from Orders o
where o.OrderDate between '2001-09-16' and '2021-09-16'
group by o.ShipPostalCode
order by count(o.ShipPostalCode) desc
```

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

```
select c.City, count(c.CustomerID) CustomersNumber
from Customers c
group by c.City
```

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

```
select c.City, count(c.CustomerID) CustomersNumber
from Customers c
group by c.City
having count(c.CustomerID) > 10
```

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

```
select c.ContactName, o.OrderDate
from Customers c
join Orders o
on o.CustomerID = c.CustomerID
where o.OrderDate > '1998-01-01'
order by c.ContactName
```

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

```
select c.ContactName, max(o.OrderDate)
from Customers c
left join Orders o
on o.CustomerID = c.CustomerID
group by c.ContactName
order by max(o.OrderDate) desc
```

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

```
select c.ContactName, SUM(od.Quantity)
from Customers c
left join Orders o
on o.CustomerID = c.CustomerID
left join [Order Details] od
on od.OrderID = o.OrderID
group by c.ContactName
```

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

```
select o.CustomerID, SUM(od.Quantity)
from Orders o
left join [Order Details] od
on o.OrderID = od.OrderID
group by o.CustomerID
having SUM(od.Quantity) > 100
order by SUM(od.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 spl.CompanyName 'Supplier Company Name', shp.CompanyName 'Shipping Company Name'
from Suppliers spl cross join Shippers shp
```

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

```
select distinct o.OrderDate, p.ProductName from Orders o
join [Order Details] od on o.OrderID = od.OrderID
join Products p on od.ProductID = p.ProductID
```

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

```
select distinct e1.FirstName, e2.FirstName from Employees e1
join Employees e2 on e1.Title = e2.Title
where e1.EmployeeID < e2.EmployeeID
```

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

```
select emper.FirstName, emper.LastName from Employees empee
join Employees emper on empee.ReportsTo = emper.EmployeeID
where emper.Title like '%Manager%'
group by emper.FirstName, emper.LastName
having count(empee.FirstName) > 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 c.City, c.CompanyName 'Name', c.ContactName 'Contact Name', 'Customer'
'Type' from Customers c
union
select s.City, s.CompanyName 'Name', s.ContactName 'Contact Name', 'Supplier' from
Suppliers s
```

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.

Although deleted by Madhu, I guess there might be a desired solution.

```
select * from T1 table1 join T2 table2 on table1.F1 = table2.F2
```

F1	F2
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 T1 table1 left join T2 table2 on table1.F1 = table2.F2
```

F1	F2
1	null
2	2
3	3



GOOD LUCK.