1. What is a result set?
   The data set we retrieve from a table by query.
2. What is the difference between Union and Union All?
   Union will eliminate the duplicate rows from all the table.
   Union All will keep the duplicated rows.
3. What are the other Set Operators SQL Server has?
   Intersect, Minus/Except.
4. What is the difference between Union and Join?
   Union combines the result sets from two or more select queries.
   Join combines the result sets of the tables with the join condition.
5. What is the difference between INNER JOIN and FULL JOIN?
   Inner join will only return the records that satisfy the join condition.
   Full join return all the records from the table including non-matching rows.
6. What is difference between left join and outer join
   Left join will fetch all the records from the left table but only those records from the right table which will satisfy the join condition. For the non-matching records, right table will return the null value.
   Outer join will sill return non-matching rows in both tables.
7. What is cross join?
   cross join will return the Cartesian product of rows from the tables.
8. What is the difference between WHERE clause and HAVING clause?
   Where clause is used to filter records from the tables.
   Having clause is used to filter records from the groups.
9. Can there be multiple group by columns?
   Yes

```sql
--1
select count(P.ProductID)
from Production.Product P

--2
select count(P.ProductID), P.ProductSubcategoryID
from Production.Product P
where P.ProductSubcategoryID is not Null
group by P.ProductSubcategoryID

--3
select P.ProductSubcategoryID, count(P.ProductID) as CountedProducts
from Production.Product P
where P.ProductSubcategoryID is not Null
group by P.ProductSubcategoryID

--4
```

```sql
select count(P.ProductID) as CountedNoSubcatProducts
from Production.Product P
where P.ProductSubcategoryID is Null

--5
select sum(P.ProductID) as ProductsQuantity
from Production.Product P

--6
select P.ProductID, P.Quantity as TheSum
from Production.ProductInventory P
where P.LocationID = 40 and P.Quantity < 100

--7
select P.Shelf, ProductID, P.Quantity as TheSum
from Production.ProductInventory P
where P.LocationID = 40 and P.Quantity < 100

--8
select avg(P.Quantity) as TheAvg
from Production.ProductInventory P
where P.LocationID = 10

--9
select P.ProductID, P.Shelf, avg(P.Quantity) as TheAvg
from Production.ProductInventory P
group by P.ProductID, P.Shelf

--10
select P.ProductID, P.Shelf, avg(P.Quantity) as TheAvg
from Production.ProductInventory P
group by P.ProductID, P.Shelf
having P.Shelf != 'N/A'

--11
select P.Color, P.Class, count(P.ProductID) as TheCount, avg(P.ListPrice) as AvgPrice
from Production.Product P
group by P.Color, P.Class
having P.Color is not null and P.Class is not null

--12
select C.Name as Country, S.Name as Province
from Person.CountryRegion C left join Person.StateProvince S
on C.CountryRegionCode = S.CountryRegionCode
group by C.Name, S.Name

--13
select C.Name as Country, S.Name as Province
from Person.CountryRegion C left join Person.StateProvince S
on C.CountryRegionCode = S.CountryRegionCode
group by C.Name, S.Name
having C.Name = 'Germany' or C.Name = 'Canada'

--14
select P.ProductID, P.ProductName
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
```

```sql
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
where YEAR(O.OrderDate) + 25 > YEAR(GETDATE())
group by P.ProductID, P.ProductName
order by P.ProductID

--15
select C.PostalCode, sum(OD.Quantity) as Sold
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
inner join dbo.Customers as C
on C.CustomerID = O.CustomerID
group by C.PostalCode
having C.PostalCode is not null
order by Sold desc
offset 0 rows
fetch next 5 rows only

--16
select C.PostalCode, O.OrderDate, sum(OD.Quantity) as Sold
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
inner join dbo.Customers as C
on C.CustomerID = O.CustomerID
group by C.PostalCode, O.OrderDate
having C.PostalCode is not null and YEAR(O.OrderDate) + 20 > YEAR(GETDATE())
order by Sold desc
offset 0 rows
fetch next 5 rows only

--17
select C.city, count(C.CustomerID) as CustomerNum
from dbo.Customers C
group by C.city

--18
select C.city, count(C.CustomerID) as CustomerNum
from dbo.Customers C
group by C.city
having count(C.CustomerID) > 10

--19
select C.ContactName, O.OrderDate
from dbo.Orders as O
inner join dbo.Customers as C
on C.CustomerID = O.CustomerID
where O.OrderDate > '1998-01-01'

--20
select C.ContactName, O.OrderDate
from dbo.Orders as O
inner join dbo.Customers as C
```

```sql
on C.CustomerID = O.CustomerID
where OrderDate = (select max(OrderDate) from dbo.Orders)

--21
select C.ContactName, count(P.ProductID) as NumProduct
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
inner join dbo.Customers as C
on C.CustomerID = O.CustomerID
group by C.ContactName

--22
select C.CustomerID, count(P.ProductID) as NumProduct
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
inner join dbo.Customers as C
on C.CustomerID = O.CustomerID
group by C.CustomerID
having count(P.ProductID) > 100


--23
select distinct Supply.CompanyName, Ship.CompanyName
from dbo.Suppliers as Supply
inner join dbo.Products as P
on P.SupplierID = Supply.SupplierID
inner join dbo.[Order Details] as OD
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
inner join dbo.Shippers as Ship
on Ship.ShipperID = O.ShipVia


--24
select O.OrderDate, P.ProductName
from dbo.[Order Details] as OD
inner join dbo.Products as P
on OD.ProductID =  P.ProductID
inner join dbo.Orders as O
on O.OrderID = OD.OrderID
order by O.OrderDate

--25
select E1.EmployeeID, E2.EmployeeID
from dbo.Employees as E1, dbo.Employees as E2
where E1.Title = E2.Title and E1.EmployeeID != E2.EmployeeID

--26
select E2.FirstName + ' ' + E2.LastName as Manager, count(E1.EmployeeID) as Num
from dbo.Employees as E1, dbo.Employees as E2
where E1.ReportsTo = E2.EmployeeID
```

```sql
group by E2.FirstName + ' ' + E2.LastName
having count(E1.EmployeeID) > 2

--27
select C.City, C.CompanyName, C.ContactName, 'Customer ' as [Type]
from dbo.Customers as C
union
select S.City, S.CompanyName, S.ContactName, 'Supplier' as [Type]
from dbo.Suppliers as S
order by C.city



--28
Select *
from F1.T1
inner join F2.T2
```

result
2
3

```sql
--29
Select *
from F1.T1
left join F2.T2
```

result
1
2
3