QUESTION #1

```sql
--Retrieve all columns from the Person.Address and Person.StateProvince tables
where the state is 'California'.
select *
from Person.StateProvince as sp
inner join Person.Address a
on a.StateProvinceID = sp.StateProvinceID
where Name = 'California'
```

QUESTION #2

```sql
--Retrieve the product names and their corresponding categories from the
Production.Product and Production.ProductCategory tables.
select p.Name as ProductName, pc.Name as Category
from Production.Product p
join Production.ProductSubcategory psc
on p.ProductSubcategoryID = psc.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = psc.ProductCategoryID
```

QUESTION #3

```sql
--List the sales order numbers, order dates, and customer names for all orders
from the Sales.SalesOrderHeader and Sales.Customer tables.
select soh.SalesOrderNumber, soh.OrderDate, c.PersonID
from Sales.Customer c
join Sales.SalesOrderHeader soh
on c.TerritoryID = soh.TerritoryID
```

QUESTION #4

```sql
select distinct CustomerID
from Sales.Customer
union
select BusinessEntityID
from HumanResources.Employee
```

QUESTION #5

```sql
select Name
from Sales.Currency
union
select name
from Sales.SalesReason
```

QUESTION #6

```sql
select name
from Production.Product
union
select Name
from Production.Product
order by Name
```

QUESTION #7

```sql
--Find the names of products with a quantity greater than 10 in the
SalesOrderDetail table.
select ProductID, Name
from Production.Product
```

```sql
    where ProductID in (
    select ProductID
    from Sales.SalesOrderDetail
    where OrderQty > 10
    )

    select ProductID, Name
    from Production.Product
    where ProductID in (
                select ProductID
                from Sales.SalesOrderDetail
                where OrderQty > 10)
    order by Name
```

QUESTION #8
```sql
    --List all customers who have placed orders in the year 2011.
    select CustomerID
    from Sales.Customer
    where CustomerID in (
    select CustomerID
    from Sales.SalesOrderHeader
    where YEAR(OrderDate) = '2011')
```

QUESTION #9
```sql
    --Retrieve the employees who have the same job title as employee 'Ken Sanchez'
    (assuming you know the EmployeeID).
    select BusinessEntityID
    from HumanResources.Employee
    where JobTitle = ( select JobTitle
    from HumanResources.Employee
    where BusinessEntityID = 3 )
```

QUESTION #10
```sql
    --List products with a price higher than the average price of all products.
    select ProductID, Name, ListPrice
    from Production.Product
    where ListPrice >  ( select avg (ListPrice) as AVGListPrice
    from Production.Product)

    select ProductID, Name, ListPrice
    from Production.Product
    where ListPrice > (
    select avg(ListPrice)
    from Production.Product)
```

QUESTION #11
```sql
--List customers who have placed more than 7000 orders:
select c.CustomerID, count(sod.OrderQty) as OrderQty
from Sales.Customer c
join Sales.SalesOrderHeader soh
on c.TerritoryID = soh.TerritoryID
join Sales.SalesOrderDetail sod
on sod.SalesOrderID = soh.SalesOrderID
group by c.CustomerID
```

```sql
having count(sod.OrderQty) > 7000
order by OrderQty desc
```

QUESTION #12
```sql
-Retrieve the average order amount for each salesperson:
select SalesPersonID, avg(TotalDue) as avgAmount
from Sales.SalesOrderHeader
where SalesPersonID is not null
group by SalesPersonID
```

QUESTION #13
```sql
--Find products with prices above the average price:
select Name as ProductName, ListPrice
from Production.Product
where ListPrice > (select avg(ListPrice)
                              from Production.Product)
```

QUESTION #14
```sql
--Retrieve customers who have placed orders in more than one year:
select CustomerID
from Sales.SalesOrderHeader
group by CustomerID
having count(distinct year(OrderDate)) > 1
```

QUESTION #15
```sql
select distinct CustomerID, number_of_customers
from (
        select CustomerID,
        COUNT(year(orderdate)) over (partition by customerID) number_of_customers
        from Sales.SalesOrderHeader) x
where number_of_customers > 1
order by CustomerID
```

QUESTION #16
```sql
--Find the top 5 products with the highest total sales:
select top 5 ProductID, Name, (select sum(LineTotal) from Sales.SalesOrderDetail) as
totalsales
from Production.Product

select x.ProductID, p.Name, x.total_sales
from (select top 10 ProductID ,sum(UnitPrice*OrderQty) as total_sales
             from Sales.SalesOrderDetail
             group by ProductID
             order by sum(UnitPrice*OrderQty) desc) x
join Production.Product p
on p.ProductID = x.ProductID
```

QUESTION #17
```sql
--List customers who have placed more than 7000 orders:
select c.CustomerID, count(sod.OrderQty) as OrderQty
from Sales.Customer c
```

```sql
join Sales.SalesOrderHeader soh
on c.TerritoryID = soh.TerritoryID
join Sales.SalesOrderDetail sod
on sod.SalesOrderID = soh.SalesOrderID
group by c.CustomerID
having count(sod.OrderQty) > 7000
order by OrderQty desc
```

QUESTION #18
```sql
--Retrieve the average order amount for each salesperson:
select SalesPersonID, avg(TotalDue) as avgAmount
from Sales.SalesOrderHeader
where SalesPersonID is not null
group by SalesPersonID
```

QUESTION #19
```sql
--Find products with prices above the average price:
select Name as ProductName, ListPrice
from Production.Product
where ListPrice > (select avg(ListPrice)
                                from Production.Product)
```

QUESTION #20
```sql
--Retrieve customers who have placed orders in more than one year:
select CustomerID
from Sales.SalesOrderHeader
group by CustomerID
having count(distinct year(OrderDate)) > 1


select CustomerID, COUNT(distinct year(OrderDate)) number_of_orders
from Sales.SalesOrderHeader
group by CustomerID
having COUNT(distinct year(OrderDate)) > 1
order by CustomerID

select distinct CustomerID, number_of_customers
from (
        select CustomerID,
        COUNT(year(orderdate)) over (partition by customerID) number_of_customers
        from Sales.SalesOrderHeader) x
where number_of_customers > 1
order by CustomerID
```

QUESTION #21
```sql
--Find the top 5 products with the highest total sales:
select top 5 ProductID, Name, (select sum(LineTotal) from Sales.SalesOrderDetail) as
totalsales
from Production.Product

select x.ProductID, p.Name, x.total_sales
from (select top 10 ProductID ,sum(UnitPrice*OrderQty) as total_sales
            from Sales.SalesOrderDetail
            group by ProductID
```

```
                order by sum(UnitPrice*OrderQty) desc) x
join Production.Product p
on p.ProductID = x.ProductID
```

QUESTION #22
```
--Total Revenue by Year:
select YEAR(OrderDate) as OrderDate, sum(TotalDue) as totalRevenue
from Sales.SalesOrderHeader
group by Year(OrderDate)
```

QUESTION #23
```
--Number of Products in Each Subcategory:
select count(p.ProductID) as numbersOfProduct, ps.Name as SubCategory
from Production.Product p
join Production.ProductSubcategory ps
on p.ProductSubcategoryID = ps.ProductSubcategoryID
group by ps.Name
order by numbersOfProduct desc
```

QUESTION #24
```
--Total Sales by Territory:
select t.Name, t.TerritoryID, sum(soh.TotalDue) as TotalSales
from Sales.SalesOrderHeader soh
join Sales.SalesTerritory t
on t.TerritoryID = soh.TerritoryID
group by t.TerritoryID, t.Name
```

QUESTION #25
```
--Average List Price by Product Model:
select pm.Name, pm.ProductModelID, avg(p.ListPrice) as AVGListPrice
from Production.Product p
join Production.ProductModel pm
on p.ProductModelID = pm.ProductModelID
group by pm.Name, pm.ProductModelID
```

QUESTION #26
```
--Number of Customers in Each City:
select count(c.CustomerID) as NumbersOfCustomers, a.City
from Sales.Customer c
join Person.StateProvince sp
on sp.TerritoryID = c.TerritoryID
join Person.Address a
on a.StateProvinceID = sp.StateProvinceID
group by a.City
order by NumbersOfCustomers desc
```

QUESTION #27
```
--Total Sales by Month:
select Month(OrderDate) as Months, sum(TotalDue) as TotalSales
from Sales.SalesOrderHeader
group by Month(OrderDate)
```

QUESTION #28
```sql
--Total Sales by Product Category:
select pc.Name, sum(soh.TotalDue) as TotalSales
from Production.ProductCategory pc
join Production.ProductSubcategory ps
on ps.ProductCategoryID = pc.ProductCategoryID
join Production.Product p
on p.ProductSubcategoryID = ps.ProductSubcategoryID
join Sales.SalesOrderDetail sod
on p.ProductID = sod.ProductID
JOIN Sales.SalesOrderHeader soh
on soh.SalesOrderID = sod.SalesOrderID
group by pc.Name
```

QUESTION #29
```sql
--Average Order Quantity by Product Subcategory:
select avg(sod.OrderQty) as AVGOrderQuantity, ps.Name
from Sales.SalesOrderDetail sod
join Production.Product p
on p.ProductID = sod.ProductID
join Production.ProductSubcategory ps
on ps.ProductSubcategoryID = p.ProductSubcategoryID
group by ps.Name
```

QUESTION #30
```sql
--Number of Orders by Customer:
select c.CustomerID, count(sod.OrderQty) as NumbersOfOrders
from Sales.SalesOrderDetail sod
join Sales.SalesOrderHeader soh
on soh.SalesOrderID = sod.SalesOrderID
join Sales.Customer c
on c.TerritoryID = soh.TerritoryID
group by c.CustomerID
```

QUESTION #31
```sql
--Retrieve all columns from the "Product" table.
select *
from Production.Product
```

QUESTION #32
```sql
--List the unique product categories from the "ProductCategory" table.
select distinct Name
from Production.ProductCategory
```

QUESTION #33
```sql
--Display the product name, list price, and standard cost for all products with a
list price greater than $500.
select Name, ListPrice, StandardCost
from Production.Product
where ListPrice > 500
```

QUESTION #34
```sql
--Find the total sales amount for each salesperson in the "SalesPerson" table.
select SalesPersonID, sum(TotalDue) as totalSales
```

```sql
from Sales.SalesOrderHeader
group by SalesPersonID
```

QUESTION #35
```sql
--Retrieve the top 5 customers with the highest total sales.
select top 5 SalesPersonID, sum(TotalDue) as totalSales
from Sales.SalesOrderHeader
group by SalesPersonID
order by SalesPersonID desc
```

QUESTION #36
```sql
--Calculate the average days in inventory for each product.
select Name, AVG(DaysToManufacture) as AVGDay
from Production.Product
group by Name
```

QUESTION #37
```sql
--Identify products with no sales in the "SalesOrderDetail" table.
select SalesOrderDetailID
from Sales.SalesOrderDetail
where CarrierTrackingNumber is null
```

QUESTION #38
```sql
--List the top 10 products with the highest quantity on hand.
select top 10 ProductID, sum(Quantity) as highestQuantity
from Production.ProductInventory
group by ProductID
order by highestQuantity desc
```

QUESTION #39
```sql
--Retrieve the first name, last name, and email address of all customers.
select FirstName, LastName
from person.person
```

QUESTION #40
```sql
--Get the product name, standard cost, and list price for all products with a
standard cost greater than 500.
select Name, StandardCost, ListPrice
from Production.Product
where StandardCost > 500
```

QUESTION #41
```sql
--Retrieve all columns from the Person.Person table for the first 10 rows.
select top 10 *
from Person.Person
```

QUESTION #42
```sql
--Find the names and standard costs of the first 5 products in the Production.Product
table.

select top 5 Name, StandardCost
from Production.Product
```

```sql
order by StandardCost desc
```

QUESTION #43
```sql
--List the unique product categories from the Production.ProductCategory table.
SELECT distinct Name
from Production.ProductCategory
```

QUESTION #44
```sql
--Display the sales order number and total due amount for the first 5 orders in the
Sales.SalesOrderHeader table.
select top 5 SalesOrderNumber, TotalDue
from Sales.SalesOrderHeader
```

QUESTION #45
```sql
--Retrieve the login ID and job title for all employees in the
HumanResources.Employee table.
select LoginID, JobTitle
from HumanResources.Employee
```

QUESTION #46
```sql
--Retrieve the first and last names of customers from the Person.Person table where
the email promotion is 1.
select FirstName, LastName
from Person.Person
where EmailPromotion = 1
```

QUESTION #47
```sql
--Find the product names and list prices of products with a list price greater than
$500 in the Production.Product table.
select Name, ListPrice
from Production.Product
where ListPrice > 500
```

QUESTION #48
```sql
--List the distinct product models from the Production.Product table.
select distinct ProductModelID
from Production.Product
```

QUESTION #49
```sql
--Display the sales order number, order date, and total due for orders with a total
due amount greater than $1000 in the Sales.SalesOrderHeader table.
select SalesOrderNumber, OrderDate, TotalDue
from Sales.SalesOrderHeader
where TotalDue > 1000
```

QUESTION #50
```sql
--Retrieve the names and job titles of employees from the HumanResources.Employee
table who were hired after '2020-01-01'.
select JobTitle
from HumanResources.Employee
where HireDate > '2012-01-01'
```

QUESTION #51
```sql
--List the distinct product categories along with their category IDs from the
Production.ProductCategory table.
select distinct ProductCategoryID, Name
from Production.ProductCategory
```

QUESTION #52
```sql
--Find the product names and colors of products with a color specified in the
Production.Product table.
select Name, Color
from Production.Product
```

QUESTION #53
```sql
--Display the distinct sales order statuses from the Sales.SalesOrderHeader table.
select distinct Status
from Sales.SalesOrderHeader
```

QUESTION #54
```sql
--Retrieve the first and last names of customers along with their email addresses
from the Person.Person table.
select FirstName, LastName
from Person.Person
```

QUESTION #55
```sql
--List the product names and sizes of products with a size specified in the
Production.Product table.
select Name, Size
from Production.Product
where size is not null
```

QUESTION #56
```sql
--Display the unique product numbers from the Production.Product table.
select distinct ProductNumber
from Production.Product
```

QUESTION #57
```sql
--Retrieve the sales order number and total due amount for orders with a total due
amount less than $500 in the Sales.SalesOrderHeader table.
select SalesOrderNumber, TotalDue
from Sales.SalesOrderHeader
where TotalDue < 500
```

QUESTION #58
```sql
--Find the distinct product models along with their standard costs from the
Production.Product table.
select distinct ProductModelID, StandardCost
from Production.Product
where ProductModelID is not null
```

QUESTION #59
```sql
--List the first and last names of employees from the HumanResources.Employee table
who have the job title 'Production Technician'.
select *
```

```sql
from HumanResources.Employee
where JobTitle like 'Production Technician - WC45'
```

QUESTION #60
```sql
--Display the product names and colors of products with a color specified in the
Production.Product table, ordered by product name.
select Name, Color
from Production.Product
where Color is not null
```

QUESTION #61
```sql
--Query 1: Easy
--Problem: Retrieve all product names and their prices
select Name, ListPrice from Production.Product
```

QUESTION #62
```sql
--Query 2: Easy
--Problem: Find the total number of employees.
select count( *) from HumanResources.Employee
```

QUESTION #63
```sql
--Query 3: Easy
--Problem: Get the top 5 customers who have made the most purchases.
select top 5 c.CustomerID, count(SalesOrderID) as Most_Purchases from Sales.Customer
c
join Sales.SalesOrderHeader soh
on soh.TerritoryID = c.TerritoryID
group by c.CustomerID
order by Most_Purchases desc
```

QUESTION #64
```sql
--Query 4: Easy
--Problem: Retrieve the names of all products with a price greater than $500.
select ProductID, Name, ListPrice from Production.Product
where ListPrice > '500'
```

QUESTION #65
```sql
--Query 5: Medium
--Problem: Find the average price of products in each product category.
select pc.Name, avg(p.ListPrice) AveragePrice from Production.Product p
join Production.ProductSubcategory psc
on psc.ProductSubcategoryID = p.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = psc.ProductCategoryID
group by pc.Name
```

QUESTION #66
```sql
--Query 6: Medium
--Problem: Identify the employee(s) who have the highest salary.
select top 10 * from HumanResources.Employee
order by
```

QUESTION #67
```sql
--Query 7: Medium
--Problem: Find the top 10 customers with the highest total purchase amount.
select top 10 c.CustomerID, sum(soh.TotalDue) as Total_Purchase from
Sales.SalesOrderHeader soh
join Sales.Customer c
on soh.CustomerID = c.CustomerID
group by c.CustomerID
order by Total_Purchase
```

QUESTION #68
```sql
--Query 8: Medium
--Problem: Retrieve all orders placed in the year 2022.
select * from Sales.SalesOrderHeader
where year(OrderDate) > '2012'
order by OrderDate
```

QUESTION #69
```sql
--Query 9: Hard
--Problem: Calculate the total revenue generated from each product category.
select sum(sod.OrderQty*sod.UnitPrice) as TotalRevenue, pc.Name as Product_Category
from Production.Product p
join Production.ProductSubcategory ps
on ps.ProductSubcategoryID = p.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = ps.ProductCategoryID
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by pc.Name
```

QUESTION #70
```sql
--Query 10: Hard
--Problem: Find the employee with the longest tenure (based on hire date).
select top 1 * from HumanResources.Employee
order by HireDate asc
```

QUESTION #71
```sql
--Query 11: Hard
--Problem: Retrieve the names of customers who have not placed any orders.
select * from Sales.Customer
where CustomerID not in (select distinct CustomerID from Sales.SalesOrderHeader)
```

QUESTION #72
```sql
--Query 12: Hard
--Problem: List the top 3 most expensive products.
select top 3 * from Production.Product
order by ListPrice desc
```

QUESTION #73
```sql
--Query 13: Medium
--Problem: Find the total number of orders placed by each customer in the year 2022.
select c.CustomerID, count(soh.SalesOrderID) TotalNumbersOfOrders from
Sales.SalesOrderHeader soh
```

```sql
join Sales.Customer c
on c.TerritoryID = soh.TerritoryID
where year(soh.OrderDate) like '2012'
group by c.CustomerID
```

QUESTION #74
```sql
--Query 15: Medium
--Problem: Calculate the total sales amount for each year.
select sum(TotalDue) TotalSalesAmount, year(OrderDate) from Sales.SalesOrderHeader
group by year(OrderDate)
```

QUESTION #75
```sql
--Query 16:
--Retrieve all products with their names and list prices:
select Name as ProductName, ListPrice
from Production.Product
```

QUESTION #76
```sql
--Query 17:
--Retrieve the top 10 customers who spent the most:
select top 10 c.CustomerID, sum(soh.TotalDue) TotalSpent from Sales.Customer c
join Sales.SalesOrderHeader soh
on soh.TerritoryID = c.TerritoryID
group by c.CustomerID
order by TotalSpent desc
```

QUESTION #77
```sql
--Query 18:
--Retrieve the average order quantity for each product category:
select pc.Name as [Category Name], avg(sod.OrderQty) as [Average Quantity] from
Production.Product p
join Production.ProductSubcategory ps
on ps.ProductSubcategoryID = p.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = ps.ProductCategoryID
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by pc.Name
```

QUESTION #78
```sql
--Query 19:
--Find the total number of products sold each month in 2023:
select Month(soh.OrderDate) Order_Date,year(soh.OrderDate) DateInYear,
count(sod.ProductID) NumberOfProducts from Sales.SalesOrderDetail sod
join Sales.SalesOrderHeader soh
on soh.SalesOrderID = sod.SalesOrderID
group by Month(soh.OrderDate), year(soh.OrderDate)
having year(soh.OrderDate) like '2012'
order by Order_Date
```

QUESTION #79
```sql
--Query 20:
--Retrieve the top 5 most sold products:
```

```sql
select top 5 p.Name as [Product Name], sum(sod.OrderQty) as [Number of Products] from
Production.Product p
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by p.Name
order by sum(sod.OrderQty) desc
```

QUESTION #80

```sql
--Query 21:
--Find the number of orders placed by each customer:
select c.CustomerID, count(soh.SalesOrderID) [Number of Orders] from
Sales.SalesOrderHeader soh
join Sales.Customer c
on c.TerritoryID = soh.TerritoryID
group by c.CustomerID
```

QUESTION #81

```sql
--Query 22:
--Retrieve the average time it takes for each product to be shipped after an order is
placed:
select sod.ProductID, avg(DATEDIFF(DAY, soh.OrderDate, soh.ShipDate)) [Average Date]
from Sales.SalesOrderHeader soh
join Sales.SalesOrderDetail sod
on sod.SalesOrderID = soh.SalesOrderID
group by sod.ProductID
```

QUESTION #82

```sql
--Query 23:
--Find the top 3 employees with the highest sales amount in 2022:
--select * from HumanResources.Employee e
--join Sales.SalesOrderHeader soh
--on soh.CustomerID = e.BusinessEntityID
```

QUESTION #83

```sql
--Query 24:
--Retrieve the number of orders shipped late (> 3 days after order date) for each
month:
select month(OrderDate) Months, count(DATEDIFF(DAY, OrderDate, ShipDate))
datedifference from Sales.SalesOrderHeader
where DATEDIFF(DAY, OrderDate, ShipDate) > 3
group by month(OrderDate)
order by Months
```

QUESTION #84

```sql
--Query 25:
--Find the total revenue generated by each product category in 2023:
select pc.Name [Category Name], sum(soh.TotalDue) TotalRevenue from
Production.Product p
join Production.ProductSubcategory ps
on ps.ProductSubcategoryID = p.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = ps.ProductCategoryID
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
```

```sql
join Sales.SalesOrderHeader soh
on soh.SalesOrderID = sod.SalesOrderID
group by pc.Name
```

QUESTION #85
```sql
--Query 26:
--Retrieve the top 5 customers with the highest average order amount:
select top 5 c.CustomerID, avg(soh.TotalDue) AverageAmount from Sales.Customer c
join Sales.SalesOrderHeader soh
on soh.TerritoryID = c.TerritoryID
group by c.CustomerID
order by AverageAmount desc
```

QUESTION #86
```sql
--Query 27:
--Find the products that have never been sold:
select * from Production.Product
where ProductID not in (select distinct ProductID from Sales.SalesOrderDetail)
```

QUESTION #87
```sql
--Query 28:
--Retrieve the top 3 sales territories with the highest total sales amount in 2023:
select top 3 st.Name Territories, st.TerritoryID, sum(TotalDue) TotalSales from
Sales.SalesOrderHeader soh
join Sales.SalesTerritory st
on st.TerritoryID = soh.TerritoryID
where year(OrderDate) like '2011'
group by st.TerritoryID, st.Name
order by TotalSales desc
```

QUESTION #88
```sql
--Query 30:
--Retrieve the top 5 best-selling products in each product category:
WITH RankedProducts AS (
    SELECT p.Name AS ProductName, pc.Name AS Category,
          ROW_NUMBER() OVER(PARTITION BY pc.Name ORDER BY SUM(sod.OrderQty) DESC) AS
Rank
    FROM Production.Product AS p
    JOIN Production.ProductSubcategory AS psc ON p.ProductSubcategoryID =
psc.ProductSubcategoryID
    JOIN Production.ProductCategory AS pc ON psc.ProductCategoryID =
pc.ProductCategoryID
    JOIN Sales.SalesOrderDetail AS sod ON p.ProductID = sod.ProductID
    GROUP BY p.Name, pc.Name
)
SELECT ProductName, Category
FROM RankedProducts
WHERE Rank <= 5;
```

QUESTION #89
```sql
--Query 31:
--Find the customers who have placed orders in all product categories:
select * from Sales.Customer c
join Sales.SalesOrderHeader soh
```

```sql
on soh.TerritoryID = c.TerritoryID
join Sales.SalesOrderDetail sod
on sod.SalesOrderID = soh.SalesOrderID
join Production.Product p
on p.ProductID = sod.ProductID
join Production.ProductSubcategory ps
on ps.ProductSubcategoryID = p.ProductSubcategoryID
join Production.ProductCategory pc
on pc.ProductCategoryID = ps.ProductSubcategoryID
```

QUESTION #90
```sql
--Retrieve all columns from the 'Product' table for products with a list price
greater than $500.
select *
from Production.Product
where ListPrice > 500
```

QUESTION #91
```sql
--List the top 5 customers who have made the highest total sales, including their
total sales amount.
select top 5 c.CustomerID, sum(soh.TotalDue) as TotalSalesAmount
from Sales.Customer c
join Sales.SalesOrderHeader soh
on soh.CustomerID = c.CustomerID
group by c.CustomerID
order by TotalSalesAmount desc;
```

QUESTION #92
```sql
--Retrieve the average number of days between order date and ship date for each
order.
select SalesOrderID, avg(datediff(day, OrderDate, ShipDate)) as AvgDays
from Sales.SalesOrderHeader
group by SalesOrderID
```

QUESTION #93
```sql
--Find the total number of products sold in each category. Display the category name
and the total quantity sold.
select pc.ProductCategoryID, pc.Name, sum(sod.OrderQty) as totalOrders
from Production.ProductCategory pc
join Production.ProductSubcategory ps
on ps.ProductCategoryID = pc.ProductCategoryID
join Production.Product p
on p.ProductSubcategoryID = ps.ProductSubcategoryID
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by pc.ProductCategoryID, pc.Name
```

QUESTION #94
```sql
--Retrieve the top 10 best-selling products based on the total quantity sold. Include
the product name and total quantity sold.
select top 10 p.ProductID, p.Name, sum(sod.OrderQty) as TotalQuantity
from Production.Product p
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
```

```sql
group by p.ProductID, p.Name
order by TotalQuantity desc;
```

QUESTION #95
```sql
--Calculate the total sales amount for each year. Display the year and the
corresponding total sales.
select year(OrderDate) ordersInYear, sum(TotalDue) TotalSalesAmount
from Sales.SalesOrderHeader
group by year(OrderDate)
```

QUESTION #96
```sql
--Find the top 3 products with the highest profit margin. Display the product name
and profit margin.
select top 3 p.ProductID, p.Name as ProductName, (sod.LineTotal - p.StandardCost) /
sod.LineTotal * 100 as ProfitMargin
from Production.Product p
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
order by ProfitMargin desc;
```

QUESTION #97
```sql
--Retrieve the average rating for each product. Display the product name and its
average rating.
select p.Name, avg(pr.Rating) as AvgRating
from Production.Product p
join Production.ProductReview pr
on p.ProductID = pr.ProductID
group by p.Name
```

QUESTION #98
```sql
--Retrieve all columns from the 'Product' table for products with a list price
greater than $500.
select *
from Production.Product
where ListPrice > 500
```

QUESTION #99
```sql
--List the top 5 customers who have made the highest total sales, including their
total sales amount.
select top 5 c.CustomerID, sum(soh.TotalDue) as TotalSalesAmount
from Sales.Customer c
join Sales.SalesOrderHeader soh
on soh.CustomerID = c.CustomerID
group by c.CustomerID
order by TotalSalesAmount desc;
```

QUESTION #100
```sql
--Retrieve the average number of days between order date and ship date for each
order.
select SalesOrderID, avg(datediff(day, OrderDate, ShipDate)) as AvgDays
from Sales.SalesOrderHeader
group by SalesOrderID
```

QUESTION #101
```sql
--Find the total number of products sold in each category. Display the category name
and the total quantity sold.
select pc.ProductCategoryID, pc.Name, sum(sod.OrderQty) as totalOrders
from Production.ProductCategory pc
join Production.ProductSubcategory ps
on ps.ProductCategoryID = pc.ProductCategoryID
join Production.Product p
on p.ProductSubcategoryID = ps.ProductSubcategoryID
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by pc.ProductCategoryID, pc.Name
```

QUESTION #102
```sql
--Retrieve the top 10 best-selling products based on the total quantity sold. Include
the product name and total quantity sold.
select top 10 p.ProductID, p.Name, sum(sod.OrderQty) as TotalQuantity
from Production.Product p
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
group by p.ProductID, p.Name
order by TotalQuantity desc;
```

QUESTION #103
```sql
--Calculate the total sales amount for each year. Display the year and the
corresponding total sales.
select year(OrderDate) ordersInYear, sum(TotalDue) TotalSalesAmount
from Sales.SalesOrderHeader
group by year(OrderDate)
```

QUESTION #104
```sql
--Find the top 3 products with the highest profit margin. Display the product name
and profit margin.
select top 3 p.ProductID, p.Name as ProductName, (sod.LineTotal - p.StandardCost) /
sod.LineTotal * 100 as ProfitMargin
from Production.Product p
join Sales.SalesOrderDetail sod
on sod.ProductID = p.ProductID
order by ProfitMargin desc;
```

QUESTION #105
```sql
--Retrieve the average rating for each product. Display the product name and its
average rating.
select p.Name, avg(pr.Rating) as AvgRating
from Production.Product p
join Production.ProductReview pr
on p.ProductID = pr.ProductID
group by p.Name
```

QUESTION #106
```sql
--Find products with a price higher than the average price:
select Name, ListPrice
from Production.Product
where ListPrice > (
select AVG(ListPrice)
```

```sql
from Production.Product
)
```

QUESTION #107
```sql
--Get the total number of products in a specific subcategory:
Select Name, (
select count(*)
from Production.Product
where ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID)
from Production.ProductCategory
```

QUESTION #108
```sql
--List all products with their names and the corresponding product category names:
select Name, (select Name as ProductSubCategory
                from Production.ProductSubcategory
                where Product.ProductSubcategoryID =
ProductSubcategory.ProductSubcategoryID)
from Production.Product
```

QUESTION #109
```sql
--Show details of products with prices greater than the average product price:
select ProductID, Name, ListPrice
from Production.Product
where ListPrice > (select avg(ListPrice)
                    from Production.Product)
```

QUESTION #110
```sql
--List all sales orders with their details and the associated customer information:
select SalesOrderID, OrderDate, TotalDue,
(select concat(FirstName,' ' , LastName) from Person.Person where
Person.BusinessEntityID = SalesOrderHeader.CustomerID)
from Sales.SalesOrderHeader
```

QUESTION #111
```sql
--List products with their names and the number of units sold for each product:
select p.ProductID, p.Name as productName, (select sum(OrderQty) as UnitSold
                                            from Sales.SalesOrderDetail where ProductID =
p.ProductID)
from Production.Product p
```

QUESTION #112
```sql
--Show the details of customers who have not placed any orders:
select CustomerID
from Sales.Customer
where CustomerID not in (select distinct CustomerID
                            from Sales.SalesOrderHeader)
```

QUESTION #113
```sql
--Display the order details for a specific sales order along with the product names:
select so.SalesOrderID, so.OrderQty, so.UnitPrice, (
select Name
from Production.Product
where ProductID = so.ProductID) as ProductName
```

```sql
from Sales.SalesOrderDetail so
```

QUESTION #114
```sql
select ed.DepartmentID, ep.Rate from HumanResources.EmployeePayHistory ep
join HumanResources.EmployeeDepartmentHistory ed
on ed.BusinessEntityID = ep.BusinessEntityID
where ep.Rate = (select Max(eph.Rate) from HumanResources.EmployeePayHistory eph
                        join HumanResources.EmployeeDepartmentHistory edh
                        on edh.BusinessEntityID = eph.BusinessEntityID
                        where edh.DepartmentID = ed.DepartmentID)
```

QUESTION #115
```sql
select ProductID, Name from Production.Product
where ProductID not in (select distinct ProductID from Sales.SalesOrderDetail)
```

QUESTION #116
```sql
select c.CustomerID, (select count(soh.CustomerID) from Sales.SalesOrderHeader soh
where soh.CustomerID = c.CustomerID) from Sales.Customer c
where (select count(soh.CustomerID) from Sales.SalesOrderHeader soh where
soh.CustomerID = c.CustomerID) > 5
```

QUESTION #117
```sql
select * from Production.product
where ListPrice > (select AVG(ListPrice) from Production.Product)
```

QUESTION #118
```sql
select top 3 * from Production.Product
order by ListPrice desc
```

QUESTION #119
```sql
SELECT
    ProductID,
    Name,
    ListPrice
FROM
    Production.Product
WHERE
    ListPrice IN (SELECT TOP 3 ListPrice
                  FROM Production.Product
                  ORDER BY ListPrice DESC);
```

QUESTION #120
```sql
--Concatenate First and Last Name:
select concat(FirstName, ' ', LastName) as FullName
from Person.Person
```

QUESTION #121
```sql
--Convert to Uppercase:
select upper(concat(FirstName,' ', LastName)) as FullName
from Person.Person
```

QUESTION #122
```sql
--Substring Example:
```

```sql
select substring(JobTitle, 1, 10) as substrings
from HumanResources.Employee
```

QUESTION #123
```sql
--Length of Email Addresses:
select EmailAddress, len(EmailAddress) as length
from Person.EmailAddress
```

QUESTION #124
```sql
--Reverse String:
select EmailAddress, reverse(EmailAddress) as reversed_address
from Person.EmailAddress
```

QUESTION #125
```sql
--Left Trim Whitespace:
select ltrim(Name)
from Production.Product
```

QUESTION #126
```sql
--Right Trim Whitespace:
select rtrim(Name)
from Production.Product
```

QUESTION #127
```sql
--Replace Text:
select replace(name, ' ', '--')
from Production.Product
```

QUESTION #128
```sql
--Extract Initials from Full Name:
select FirstName, LastName, left(FirstName, 1) + left(LastName, 1) as initials
from Person.Person
```

QUESTION #129
```sql
--Title Case for Product Names:
select ProductID, UPPER(Left(Name, 1)) + LOWER(substring(Name, 2, len(Name))) as
Title_Case
from Production.Product
```

QUESTION #130
```sql
select left(Name, 1), right(Name,1)
from Production.Product
```

QUESTION #131
```sql
select comments
from Production.ProductReview
```