```sql
-------SQL Case Study---------


---1. List all customers--

select * from dbo.Customer

---****************************************---

---2. List the first name, last name, and city of all customers

select CONCAT(FirstName,' ',LastName) as [Name of Cust], City from dbo.Customer

---****************************************---

---3. List the customers in Sweden.

select * from dbo.Customer
where Country='Sweden'

---****************************************---

---4. Create a copy of Supplier table. Update the city to Sydney for supplier starting with letter P

--creating new table
select * into [dbo].[Supplier_2] from [dbo].[Supplier]

--Updating new table
Update [dbo].[Supplier_2]
Set City='Sydney'
where CompanyName like 'P%'

--Checking new table
select * from [dbo].[Supplier_2]
where CompanyName like 'P%'

---****************************************---


---5. Create a copy of Products table and Delete all products with unit price higher than $50.

--creating new table
select * into [dbo].[Product_2] from [dbo].[Product]

Select * from dbo.Product_2
where UnitPrice>50
```

```sql
--deleting--
delete from dbo.Product_2
where UnitPrice>50

---****************************************---

---6. List the number of customers in each country--
select Country,count(*) as [Count of cust] from dbo.Customer
group by Country
order by count(*) desc

---****************************************---

---7. List the number of customers in each city sorted high to low
select Country,City,count(*) as [Count of cust] from dbo.Customer
group by Country,City
order by count(*) desc

---****************************************---

---8. List the total amount for items ordered by each customer
select CONCAT(t1.FirstName,' ',t1.LastName) as [Name of Cust],count(t2.OrderNumber) as [Number of Orders],
sum(t2.TotalAmount) as [Total Amount]from dbo.Customer as t1
join dbo.Orders as t2 on t1.Id=t2.CustomerId
group by CONCAT(t1.FirstName,' ',t1.LastName)
order by sum(t2.TotalAmount) desc

---****************************************---

---9. List the number of customers in each country. Only include countries with more than 10 customers
select Country from dbo.Customer
group by Country
having count(*) >10

---****************************************---

---10. List the number of customers in each country, except the USA, sorted high to low. Only include countries with 9 or more customers.
select Country,count(*) as [Count of cust] from dbo.Customer
where Country != 'USA'
group by Country
having count(*) >= 9
order by count(*) desc

---****************************************---

---11. List all customers whose first name or last name contains "ill".
```

```
Select CONCAT(t1.FirstName,' ',t1.LastName) as [Cust Name] from dbo.Customer as t1
where FirstName like '%ill%' or LastName like '%ill%'

---****************************************---

---12. List all customers whose average of their total order amount is between $1000
  and $1200.Limit your output to 5 results.
select CONCAT(t1.FirstName,' ',t1.LastName) as [Name of Cust],count(t2.OrderNumber) as
   [Number of Orders],
avg(t2.TotalAmount) as [Avg Amount]from dbo.Customer as t1
join dbo.Orders as t2 on t1.Id=t2.CustomerId
group by CONCAT(t1.FirstName,' ',t1.LastName)
having avg(t2.TotalAmount) between 1000 and 1200
order by avg(t2.TotalAmount) desc

---****************************************---


---13. List all suppliers in the 'USA', 'Japan', and 'Germany', ordered by country
  from A-Z, and then by company name in reverse order.
select Country,CompanyName,ContactName from dbo.Supplier
where Country in ('USA', 'Japan','Germany')
order by Country,CompanyName desc

---****************************************---

---14. Show summary of orders, sorted by total amount (the largest amount first),
  within each year.

select Year(convert(date,convert(varchar,OrderDate))) as [Year],
       count(Id)[No. of Orders],sum(TotalAmount)[Total Amt],
       AVG(TotalAmount)[Avg Amt] from dbo.Orders
group by Year(convert(date,convert(varchar,OrderDate)))
order by sum(TotalAmount) desc

---****************************************---

/*15.Products with UnitPrice greater than 50 are not selling despite promotions. You
  are asked to
discontinue products over $25. Write a query to relfelct this. Do this in the copy of
  the Product
table. DO NOT perform the update operation in the Product table.*/

delete from dbo.Product_2
where UnitPrice > 25

select * from dbo.Product_2

---****************************************---
```

```
---16. List top 10 most expensive products
select top 10 ProductName,UnitPrice from  dbo.Product
order by UnitPrice desc

---****************************************---

---17. Get the 10th to 15th most expensive products sorted by price
select * from
(select top 15 ProductName,UnitPrice from  dbo.Product
order by UnitPrice desc) as t
order by UnitPrice desc
offset 9 rows

--or
select * from
(select ProductName,UnitPrice, RANK() over(order by UnitPrice desc )
                                     as [rank] from dbo.Product) as t
where t.rank>=10 and t.rank<=15

---****************************************---

---18. Write a query to get the number of supplier countries. Do not count duplicate
  values.
select distinct count(Country) [No. of Countries] from dbo.Supplier

---****************************************---

---19. Find the total sales cost in each month of the year 2013.

select Month(convert(date,convert(varchar,OrderDate))) as [Month],
       count(Id)[No. of Orders],sum(TotalAmount)[Total Amt],
       AVG(TotalAmount)[Avg Amt] from dbo.Orders
where Year(convert(date,convert(varchar,OrderDate)))='2013'
group by Month(convert(date,convert(varchar,OrderDate)))

---****************************************---

---20. List all products with names that start with 'Ca'
select ProductName from dbo.Product
where ProductName like 'Ca%'

---****************************************---


---21. List all products that start with 'Cha' or 'Chan' and have one more character
select ProductName from dbo.Product
where ProductName like 'Cha%'
```

```sql
---****************************************---

/*22.Your manager notices there are some suppliers without fax numbers. He seeks your help to
get a list of suppliers with remark as "No fax number" for suppliers who do not have fax
numbers (fax numbers might be null or blank).Also, Fax number should be displayed for
customer with fax numbers.*/
select ContactName,case when Fax is null then 'No fax number' else Fax end as Fax_new
from dbo.Supplier

---****************************************---


---23. List all orders, their orderDates with product names, quantities, and prices.
select * from dbo.Orders
select * from dbo.OrderItem
select * from [dbo].[Product]

select t1.Id,t1.OrderDate,t3.ProductName,t2.Quantity as Qty,t2.UnitPrice*t2.Quantity as
              [Amount of the item], t1.TotalAmount [Total amount of the order]
              from dbo.Orders as t1
join dbo.OrderItem as t2 on t1.Id=t2.OrderId
join [dbo].[Product] as t3 on t3.id=t2.productid

---****************************************---

---24. List all customers who have not placed any Orders.
select * from [dbo].[Customer]

Select Id,CONCAT(t1.FirstName,' ',t1.LastName) as [Cust Name] from dbo.Customer as t1
where t1.Id not in (select distinct CustomerId from dbo.Orders)

---****************************************---

/*25. List suppliers that have no customers in their country, and customers that have no suppliers
in their country, and customers and suppliers that are from the same country. */

select t1.FirstName,t1.LastName,t1.Country [CustomerCountry],t2.Country [SupplierCountry],
                              t2.ContactName from [dbo].[Customer] t1
full join [dbo].[Supplier] t2 on t1.Country=t2.Country
order by CustomerCountry

---****************************************---
```

```sql
/*26. Match customers that are from the same city and country. That is you are asked to give a list
of customers that are from same country and city. Display firstname, lastname, city and
coutntry of such customers.*/

with cte as (select * from [dbo].[Customer])
select c1.FirstName,c1.LastName,c2.FirstName,c2.LastName,c1.City,c1.Country from cte c1
join cte c2 on c2.City=c1.City and c2.Country=c1.Country and c1.Id != c2.Id
order by c1.Country

---****************************************---



/*27.List all Suppliers and Customers. Give a Label in a separate column as 'Suppliers' if he is a
supplier and 'Customer' if he is a customer accordingly. Also, do not display firstname and
lastname as twoi fields; Display Full name of customer or supplier.*/

select t2.id, t2.ContactName,
    case when t2.Id>0 then 'Supplier' else null end Type from [dbo].[Supplier] as t2
union
select t1.Id,CONCAT(t1.FirstName,' ',t1.LastName) as [Name of Cust],
           case when t1.Id>0 then 'Customer' else null end Type from [dbo].[Customer] as t1

---****************************************---

/*28. Create a copy of orders table. In this copy table, now add a column city of type varchar (40).
Update this city column using the city info in customers table.*/

select * into [dbo].[Orders_2] from [dbo].[Orders]

select * from [dbo].[Orders_2]

Alter table [dbo].[Orders_2] add city varchar(40)

Update [dbo].[Orders_2]
set city = (select city from [dbo].[Customer] where [dbo].[Customer].Id=[dbo].[Orders_2].CustomerId)

---checking
select * from [dbo].[Customer] where Id in (85,79,34)

select * from [dbo].[Orders_2] where CustomerId in (85,79,34)
```

```sql
---****************************************---


/*29. Suppose you would like to see the last OrderID and the OrderDate for this last order that
was shipped to 'Paris'. Along with that information, say you would also like to see the
OrderDate for the last order shipped regardless of the Shipping City. In addition to this, you
would also like to calculate the difference in days between these two OrderDates that you get.
Write a single query which performs this.
(Hint: make use of max (columnname) function to get the last order date and the output is a
single row output.)*/

with cte_paris as (select max(t2.Id) as [ID],max(convert(date,convert(varchar,OrderDate)))
                              as [LastParisOrder]
                              from [dbo].[Customer] as t1
                              join [dbo].[Orders] as t2 on t1.Id=t2.CustomerId
                              where t1.City='Paris'
                              ),
     cte_max as (select max(convert(date,convert(varchar,OrderDate))) as [LastOrder]
                              from [dbo].[Orders])
select *,DATEDIFF(day,[LastParisOrder],[LastOrder]) as [DifferenceIn Days] from cte_paris,cte_max



---****************************************---


/*30. Find those customer countries who do not have suppliers. This might help you provide
better delivery time to customers by adding suppliers to these countires. Use SubQueries.*/

select distinct Country from [dbo].[Customer]
where [dbo].[Customer].Country not in (select distinct [dbo].[Supplier].Country from [dbo].[Supplier])

---****************************************---


/*31. Suppose a company would like to do some targeted marketing where it would contact
customers in the country with the fewest number of orders. It is hoped that this
```

```sql
  targeted
marketing will increase the overall sales in the targeted country. You are asked to write a query
to get all details of such customers from top 5 countries with fewest numbers of orders. Use
Subqueries.*/

with cte_orders as (select t1.Country,count(t2.Id) as [TotalCountOfOrders] from [dbo].[Customer] t1
                   join [dbo].[Orders] t2 on t1.Id=t2.CustomerId
                   group by t1.Country
                   )
Select * from  [dbo].[Customer] where Country in
(select Country from (select *,RANK() over(order by TotalCountOfOrders) as [Rankk] from cte_orders) as t2
where t2.Rankk<=5)
order by Country

---****************************************---


/*32. Let's say you want report of all distinct "OrderIDs" where the customer did not purchase
more than 10% of the average quantity sold for a given product. This way you could review
these orders, and possibly contact the customers, to help determine if there was a reason for
the low quantity order. Write a query to report such orderIDs.*/

select * from [dbo].[OrderItem]

select ProductId,avg(Quantity) as Avg_Qty,avg(Quantity)*0.10 as Avg_Qty_Req,
sum(Quantity) as Total_Qty,count(Quantity) as Count_Qty from [dbo].[OrderItem]
group by ProductId

select * from [dbo].[OrderItem] t1
where t1.Quantity<(select avg(t2.Quantity)*0.10 from [dbo].[OrderItem] t2 where t1.ProductId=t2.ProductId)


---****************************************---

/*33. Find Customers whose total orderitem amount is greater than 7500$ for the year 2013. The
total order item amount for 1 order for a customer is calculated using the formula UnitPrice *
Quantity * (1 - Discount). DO NOT consider the total amount column from 'Order' table to
calculate the total orderItem for a customer.*/
```

```sql
select t1.CustomerId,sum(t2.UnitPrice*t2.Quantity*(1-t2.Discount)) as Amount from
  [dbo].[Orders] as t1
join [dbo].[OrderItem] as t2 on t1.Id=t2.OrderId
where year(convert(date,convert(varchar,OrderDate)))=2013
group by t1.CustomerId
having sum(t2.UnitPrice*t2.Quantity*(1-t2.Discount))>7500

---****************************************---


/*34. Display the top two customers, based on the total dollar amount associated with
  their
orders, per country. The dollar amount is calculated as OI.unitprice * OI.Quantity *
  (1 -
OI.Discount). You might want to perform a query like this so you can reward these
  customers,
since they buy the most per country.
Please note: if you receive the error message for this question "This type of
  correlated subquery
pattern is not supported yet", that is totally fine.*/

select * from (select t1.CustomerId,t3.Country,sum(t2.UnitPrice*t2.Quantity*(1-
  t2.Discount)) as Amount,
    Rank() over(partition by Country order by sum(t2.UnitPrice*t2.Quantity*(1-
      t2.Discount)) desc) as Rank
    from [dbo].[Orders] as t1
join [dbo].[OrderItem] as t2 on t1.Id=t2.OrderId
join [dbo].[Customer] as t3 on t3.Id=t1.CustomerId
group by t3.Country,t1.CustomerId
) as t5
where t5.Rank<3

---****************************************---


---35. Create a View of Products whose unit price is above average Price.

Create view [Productss] as
with cte as (select distinct ProductId, UnitPrice from [dbo].[OrderItem])
select distinct ProductId from cte c1 where c1.UnitPrice> (select AVG(c2.UnitPrice)
  from cte c2)

select * from Productss

---****************************************---

/**36. Write a store procedure that performs the following action:
Check if Product_copy table (this is a copy of Product table) is present. If table
```

```
  exists, the
procedure should drop this table first and recreated.
Add a column Supplier_name in this copy table. Update this column with that of
'CompanyName' column from Supplier tab***/

Create procedure updates
as

drop table if exists Product_2;

select * into Product_2 from Product;

alter table Product_2 add Supplier_name varchar(200);

update Product_2
set Supplier_name=
(select CompanyName from [dbo].[Supplier] where [dbo].[Supplier].Id=[dbo].
  [Product_2].SupplierId)
Go

Exec updates

---*****************************************---
```