CaseStudy

```
create database customers;


use customers;


SELECT * FROM fact;
select * from Product;
select * from Location;


--Display the number of states present in the LocationTable.
select count(Distinct state) NoOfStates from Location;


--How many products are of regular type?
select count(Product) RegularType from Product
where type = 'Regular';


--How much spending has been done on marketing of product ID 1?
select sum(Marketing) SpendingOFPID1 from Fact
where ProductId = 1;


--What is the minimum sales of a product?
select min(Sales) MinimumSales from fact;


--Display the max Cost of Good Sold (COGS).
select max(COGS) MaximumCOGS from Fact;


--Display the details of the product where product type is coffee
select * from Product
where Product_Type = 'Coffee';
```

```
--Display the details where total expenses are greater than 40.
select * from Fact
where Total_Expenses > 40;

--What is the average sales in area code 719?
select avg(Sales) AvgSalesOfAreaCode719 from Fact
where area_code = 719;

--Find out the total profit generated by Colorado state.
select sum(Profit) from Fact f join Location l
on l.area_code = f.area_code
where l.State = 'Colorado';

--Display the average inventory for each product ID.
select ProductId, avg(inventory) AverageInventory from Fact
group by ProductId;

--Display state in a sequential order in a Location Table.
select state from Location
order by state;

/*Display the average budget of the Product where the average budget
margin should be greater than 100.*/
select ProductId, avg(Budget_Sales) Average_Budget_Of_The_Product from fact
where Budget_Margin > 100
group by ProductId;

--What is the total sales done on date 2010-01-01?
```

```
select Date, count(Sales) 'Total_Sales_2010-01-01' from fact
where Date = '2010-01-01'
group by Date;

--Display the average total expense of each product ID on an individual date.
select Date, ProductId, avg(Total_Expenses)
'Total_Expenses_Of_individual_ProductID&Date'
from fact
group by Date, ProductId;

/*Display the table with the following attributes such as date,
productID, product_type, product, sales, profit, state, area_code.*/
Select f.Date, p.ProductId, p.Product_Type,
p.Product, f.Sales, f.Profit, l.State, f.Area_Code
from fact f join Product p
on f.ProductId = p.ProductId
join Location l
on f.Area_Code = l.Area_Code;

--Display the rank without any gap to show the sales wise rank.
select Date, ProductId, Sales, RANK() over (order by Sales) SalesWiseRank
from fact;

--Find the state wise profit and sales
select l.State, sum(f.Profit) Profit, sum(f.Sales) Sales
from fact f join Location l
on f.Area_Code = l.Area_Code
group by State;
```

```
--Find the state wise profit and sales along with the productname.
select l.State, p.Product, sum(f.Profit) Profit, sum(f.Sales) Sales
from fact f join Location l
on f.Area_Code = l.Area_Code
join Product p
on f.ProductId = p.ProductId
group by l.State, p.Product;

--If there is an increase in sales of 5%, calculate the increasedsales.
select ProductId, Sales, Sales*1.05 IncreasedSales from fact;

--Find the maximum profit along with the product ID and producttype.
select f.ProductId, p.Product, max(f.Profit) Maximum_Profit
from fact f join Product p
on f.ProductId = p.ProductId
group by f.ProductId, p.Product
order by f.ProductId;

/*Create a stored procedure to fetch the result
according to the product type from Product Table.*/
create procedure ProductsByType @ptype varchar(30)
as
select * from Product where Product_Type = @ptype;
go

exec ProductsByType @ptype = 'Coffee';

/*Write a query by creating a condition in which if the
total expenses is less than 60 then it is a profit or else loss.*/
```

```
select Date, ProductId,
        case
                when Total_Expenses < 60 then 'Profit'
                else 'Loss'
        end as ProfitOrLoss
from fact;

/*Give the total weekly sales value with the date and product
ID details. Useroll-up to pull the data in hierarchical order*/
select Date, ProductId, sum(sales)
as 'Weekly Sales' from fact
group by rollup(Date, ProductId);

/*Apply union and intersection operator on the tables which consist of
attribute area code.*/
select Area_Code from fact
union
select Area_Code from Location;

select Area_Code from fact
intersect
select Area_Code from Location;

/*Create a user-defined function for the product table to fetch a particular
product type based upon the user's preference.*/
create function productType(@product_type varchar(50))
returns table
as
return(
```

```
        select * from Product
        where Product_Type = @product_type
);

select * from productType('Coffee');

--Change the product type from coffee to tea where product ID is 1 and undo it.
declare @trans varchar(30)
begin transaction @trans
update Product
set Product_Type = 'Tea'
where ProductId = 1

declare @trans varchar(30)
rollback transaction @trans;
select * from Product;

/*Display the date, product ID and sales where total expenses are
between 100 to 200.*/
select Date, ProductId, Sales
from fact where Total_Expenses between 100 and 200;



select * from Product;

--Delete the records in the Product Table for regular type.
delete from Product
where Type = 'Regular';
select * from Product
```

```
--Display the ASCII value of the fifth character from the columnProduct.
select ascii(substring(Product, 5, 1)) as ASCIIValue
from Product;
```