```sql
create database casestudy1
use casestudy1


select * from product
select * from location
select * from fact

-- 1. Display the number of states present in the LocationTable
or
-- 1) How many states are there where products have been sold

select count (distinct state) AS COUNTOFSTATE from location

--2. How many products are of regular type?
select count(product) as countofRegulartype from product where type='regular'

--3. How much spending has been done on marketing of product ID 1? --
select sum(marketing) as marketing from fact where productid=1

--4. What is the minimum sales of a product?
select  min(sales) as min_sales from fact

-- 5) Display max Cost of Good Sold

select max(COGS) as max_COGS from fact

--issue -- 6)Display the Details of the product table  where product type is coffee-
>select * from product where product_type='coffee'

-- 6)Display the Details of the productid where product type is coffee

select ProductId from Product where product_type='coffee'

--7) Display the details where total_expenses is greater than 40.

select * from fact where total_expenses>40

--8) What is the average sales in Area_Code 719 ?

select avg(sales) as avg_sales from fact where area_code=719

-- 9) Find out the total profit generated by Colorado state.

select sum(profit) as total_profit from fact
inner join
location on  fact.Area_Code=location.Area_Code
where State='Colorado'

-- 10)  Display the average inventory for each product id

SELECT productid,AVG(Inventory) AS AVERAGE_Inventory FROM fact group by productid
--or
SELECT productid,AVG(Inventory) AS AVERAGE_Inventory FROM fact group by productid order
by productid


-- 11) Display state in a sequential order in a location table
```

```sql
select * from location order by state
--or
select  * from location order by state desc

-- 12)  Display the average budget margin of the store where average budget margin should
be greater than 100

SELECT productid,AVG(budget_margin) AS AVG_budget_margin FROM fact GROUP BY productid
HAVING AVG(budget_margin)> 100

-- 13)  What is the total sales done on 2010-01-01

select  sum(sales) as total_sales from fact where date='2010-01-01'

-- issue (display the average total expense of individual product id on each date)

-- 14) display the average total expense of each product id on individual date

SELECT  productid, date, avg(total_expenses) as avg_total_expense FROM fact group by
productid ,date
--or
SELECT productid,date,avg(total_expenses) as avg_total_expense FROM fact group by
productid ,date order by date
--or
SELECT productid,date,avg(total_expenses) as avg_total_expense FROM fact group by
productid,date order by date,productid

--15) Display the table with the following attributes such as Date, productid,
product_type, product, Sales, profit,
--   state, area_code

select
fact.Date,fact.ProductId,Product.product_type,Product.product,fact.Sales,fact.profit,Loca
tion.state,Location.Area_Code
from fact inner join location
on Fact.Area_code=Location.area_code
inner join dbo.product
on fact.productid=product.productid

--16) Display the rank without any gap to show the Sales wise rank.

SELECT
  Date,
  productid,
  Sales,
  profit,
  area_code,
  DENSE_RANK() OVER (ORDER BY Sales aSC) AS Sales_Rank
FROM Fact;

--This query will display the rank of each product based on their sales, with no gaps
between the ranks.

--17) Find the State wise Profit and Sales.

select state, sum(sales) as Sales,sum(profit) as Profit from fact inner join location on
fact.Area_Code=location.Area_Code
```

```sql
group by state


--18) Find the State wise Profit and Sales along with the Product Name

select Product , state, sum(sales) as Sales,sum(profit) as Profit from fact inner join
location on fact.Area_Code=location.Area_Code
inner join Product on Fact.ProductId=Product.ProductId
group by  state ,Product


--19) If there is an increase in sales of 5%. Calculate the increased sales.

--increase_in_salaes  -->5% = 5/100=0.05)

SELECT sales, (sales * 0.05) AS increase_in_sales
FROM Fact

select * from fact

--20) Find the maximum profit along with the Product id and Product Type.


Select Fact.ProductId, Product.Product Type, Fact.Profit from Fact
inner join
Product
on Fact.ProductId = Product.ProductId
where Fact.Profit = (Select MAX(Profit) from Fact)

--or

SELECT Product.ProductId, product_type, MAX(profit) AS max_profit
FROM fact inner join product on fact.ProductId=Product.ProductId
group by Product.ProductId,Product.Product_Type



--21) Create a Stored Procedure to fetch the result according to the product type from
Product.


CREATE PROCEDURE PType @Prod_typ varchar(20) as
Select * from Product
Where Product_Type = @Prod_typ

Exec PType @Prod_typ = 'Coffee'
Exec PType @Prod_typ ='Tea'

--22) 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 total_expenses,IIF(total_expenses<60,'profit','loss') as status from fact
or
select total_expenses,
case
when total_expenses<60 then 'profit'
else 'loss'
end from fact
```

```
--23) Give the total weekly sales value with the Date and productid details. Use roll-up
to pull the data in hierarchical order.


SELECT DATEPART(WEEK, [Date]) AS [Weekly Sales], [Date], ProductID, SUM(Sales) AS
Total_Sale
FROM fact
GROUP BY ROLLUP (DATEPART(WEEK, [Date]), [Date], ProductID)

--or

Select productid,
SUM(Sales)as Total_Sale from
fact
group by productid
with Rollup

--24) 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

--25) Create a user-defined function for the product table to fetch a particular product
type based upon the user’s preference.

Create function Producttable(@product_type varchar(50))
returns
table as
return select * from product where Product_Type=@product_type

select * from dbo.Producttable('Coffee')

--26) Change the product type from coffee to tea where product id is 1 and undo it.


select * from product

begin transaction
update product set product_type='Tea' where productid =1

select * from product

rollback transaction

select * from product

--27) Display the Date, productid and sales where total expenses are between 100 to 200.
```

```sql
select date, productid ,sales ,Total_Expenses from fact where Total_Expenses between 100
and 200

--28) Delete the records in the product table for regular type.

select * from product

DELETE FROM product WHERE Type = 'Regular';

select * from product

--29) Display the ASCII value of the fifth character from the column product.

SELECT Product,
(ASCII(SUBSTRING(Product,5,1)))
AS
ASCII_VALUE_OF_fifth_CHARACTER
FROM Product

-- ASCII (American Standard Code for Information Interchange
--it will contain space also for giving value

--for findng ascii value of individual letter

--select ascii('a')
--SELECT ASCII('G') AS ASCII_VALUE;
```