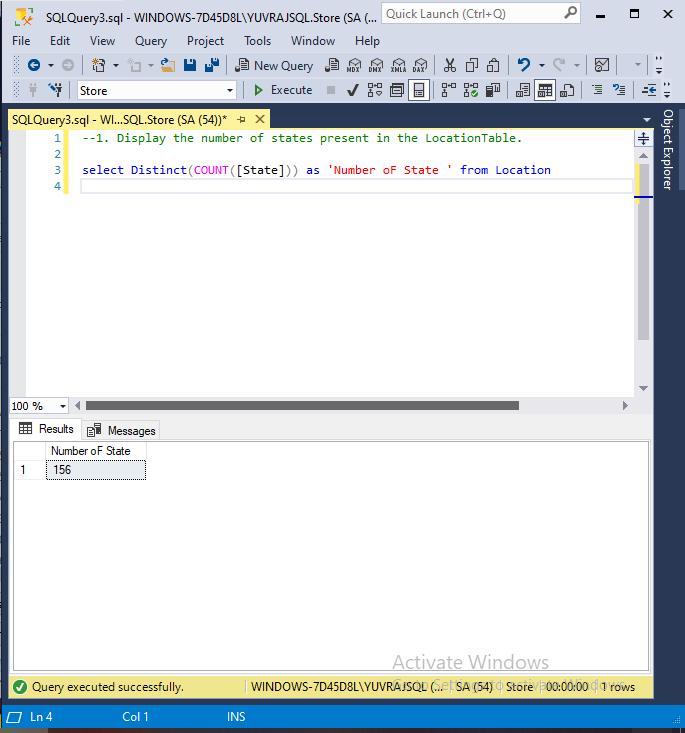
**Problem Statement:**

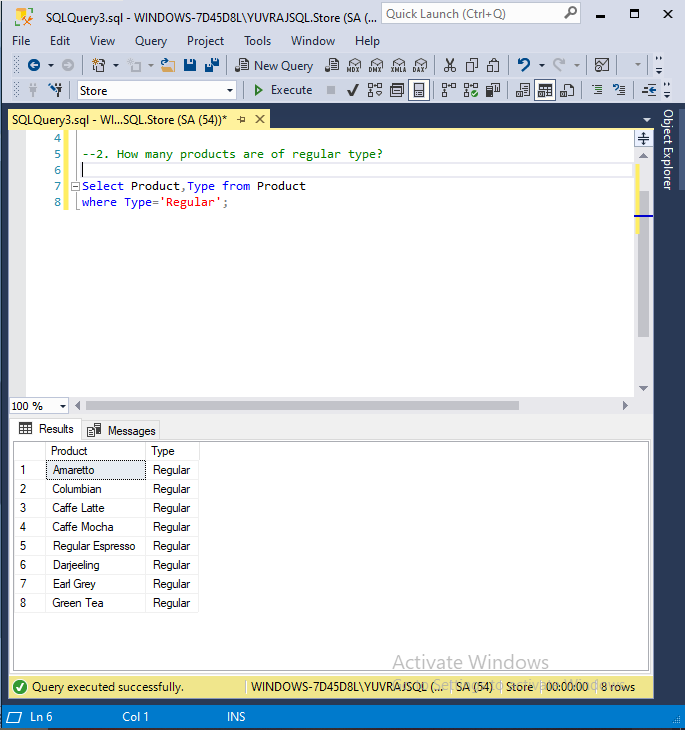
You are a database administrator. You want to use the data to answer a few questions about your customers, especially about the sales and profit coming from different states, money spent in marketing and various other factors such as COGS (Cost of Goods Sold), budget profit etc. You plan on using these insights to help find out which items are being sold the most. You have been provided with the sample of the overall customer data due to privacy issues. But you hope that these samples are enough for you to write fully functioning SQL queries to help answer the questions

**Tasks to be performed:**

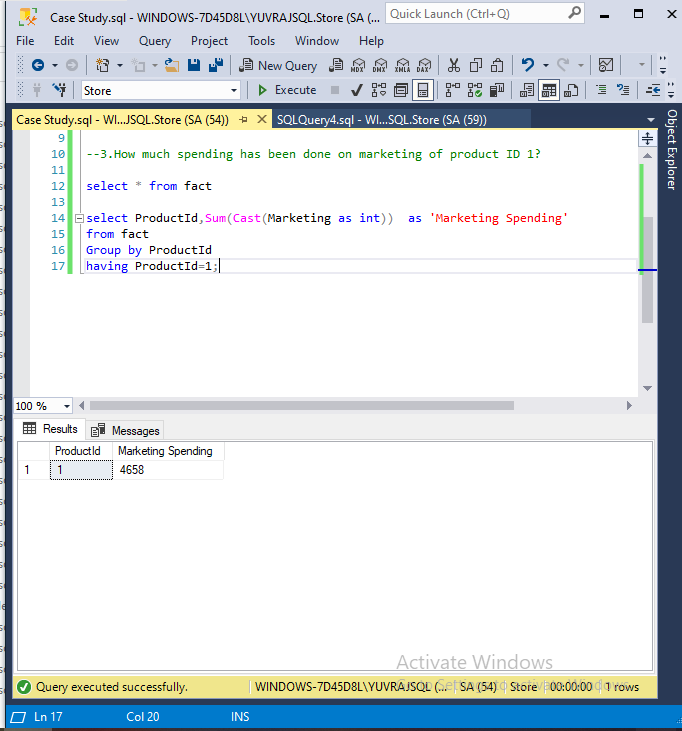
1. Display the number of states present in the Location Table.



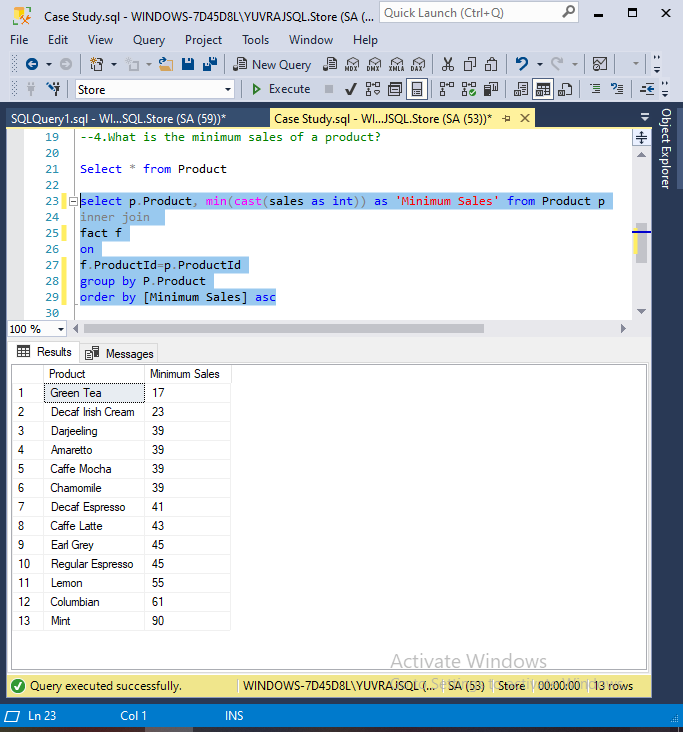
1. How many products are of regular type?



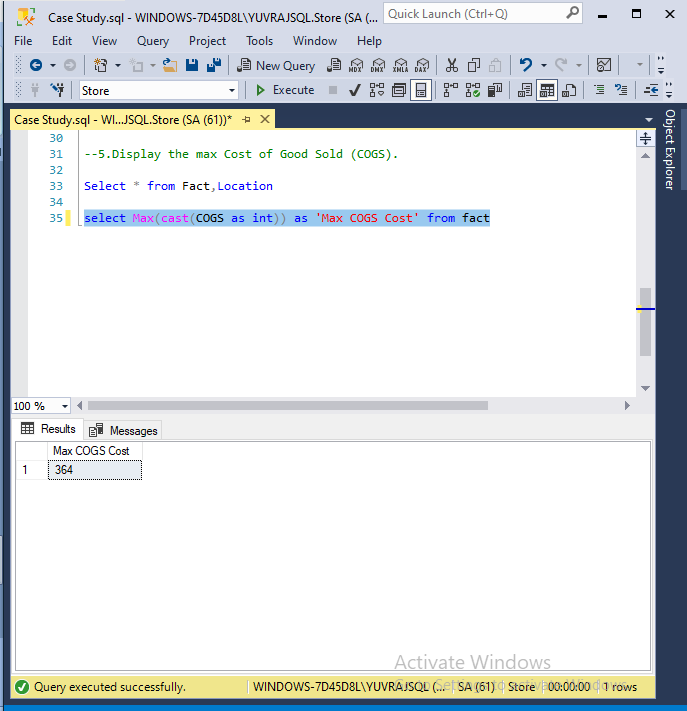
1. How much spending has been done on marketing of product ID 1?



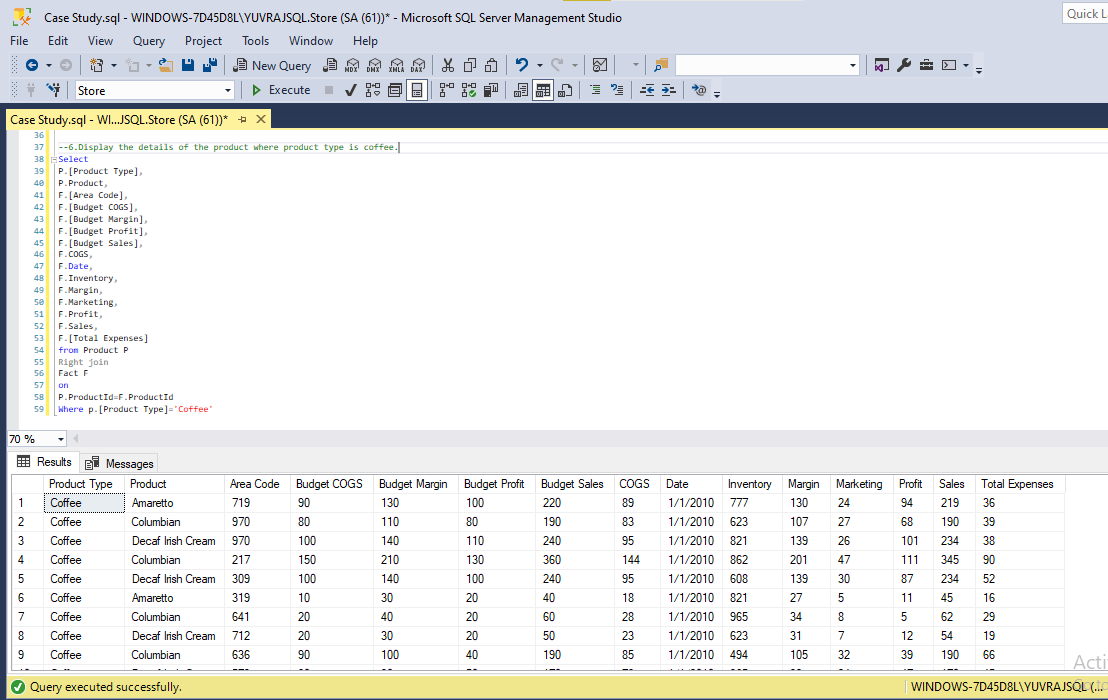
1. What is the minimum sales of a product?



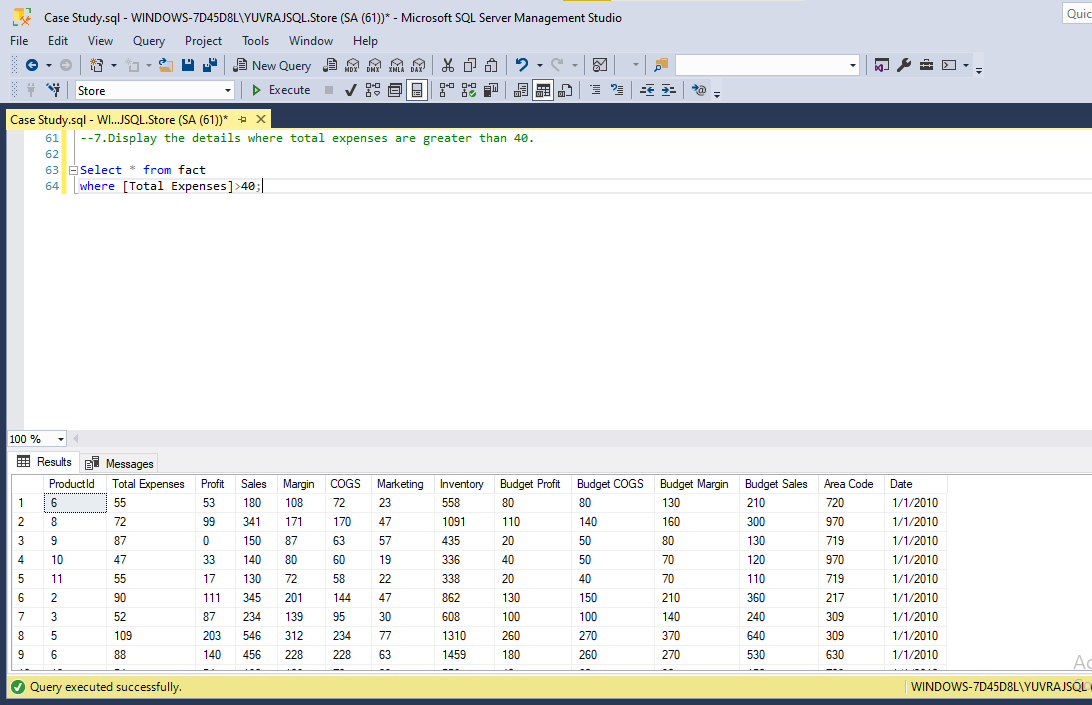
1. Display the max Cost of Good Sold (COGS).



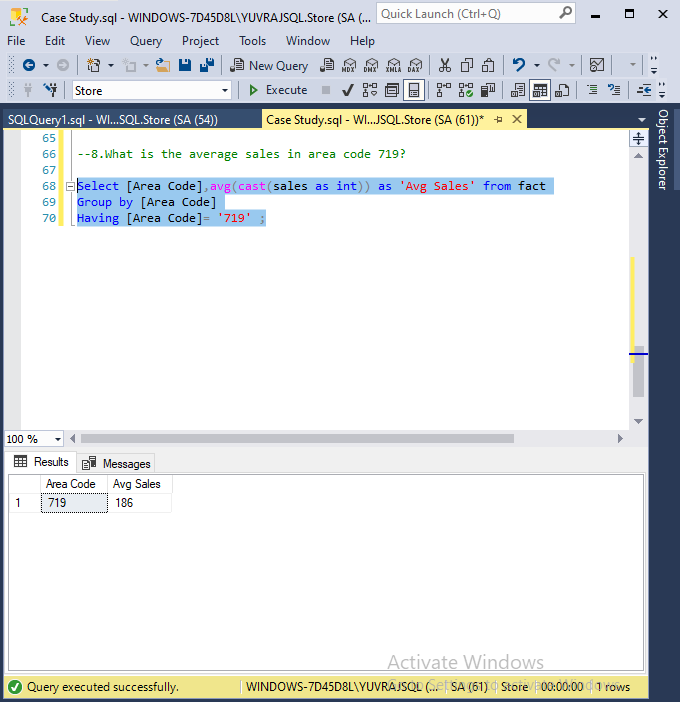
1. Display the details of the product where product type is coffee.



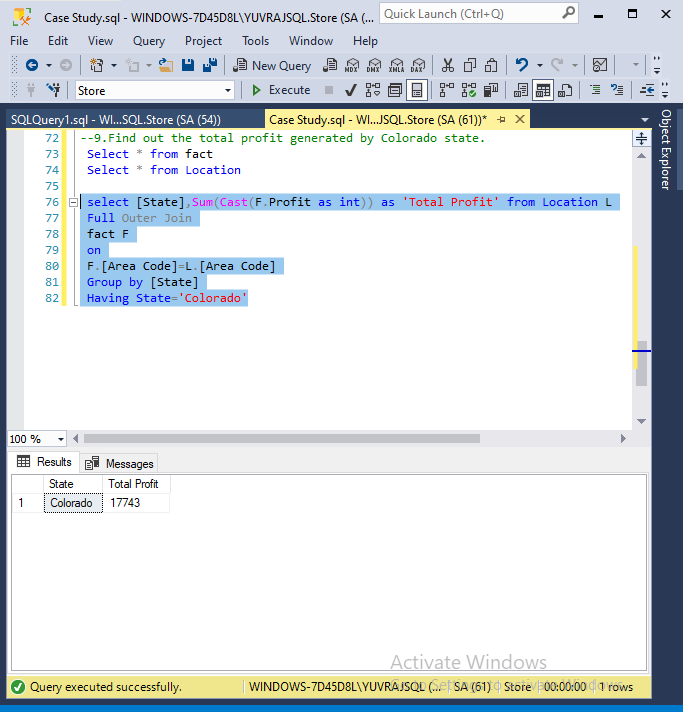
1. Display the details where total expenses are greater than 40.



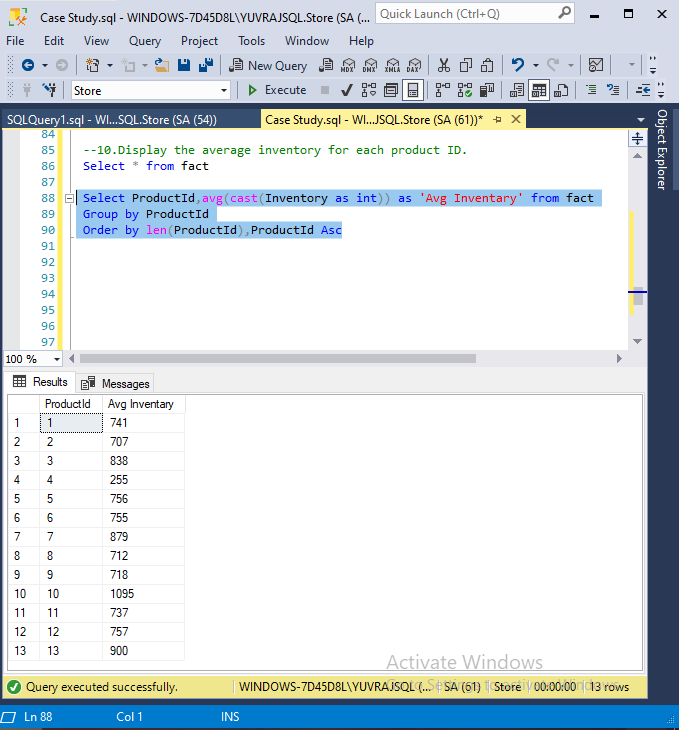
1. What is the average sales in area code 719?



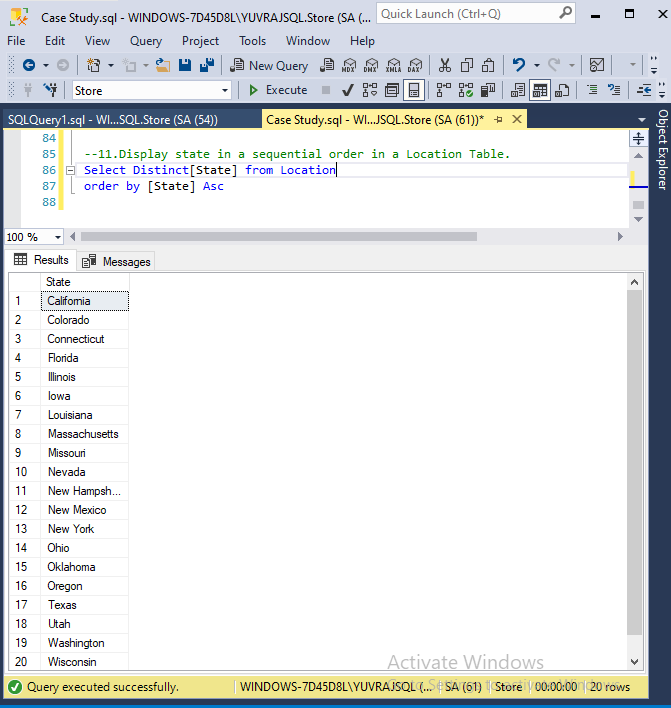
1. Find out the total profit generated by Colorado state.



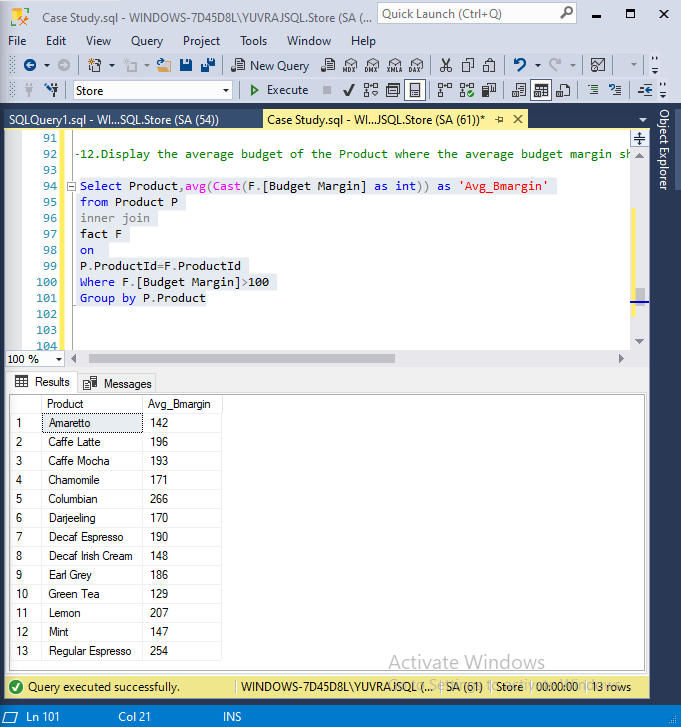
1. Display the average inventory for each product ID.



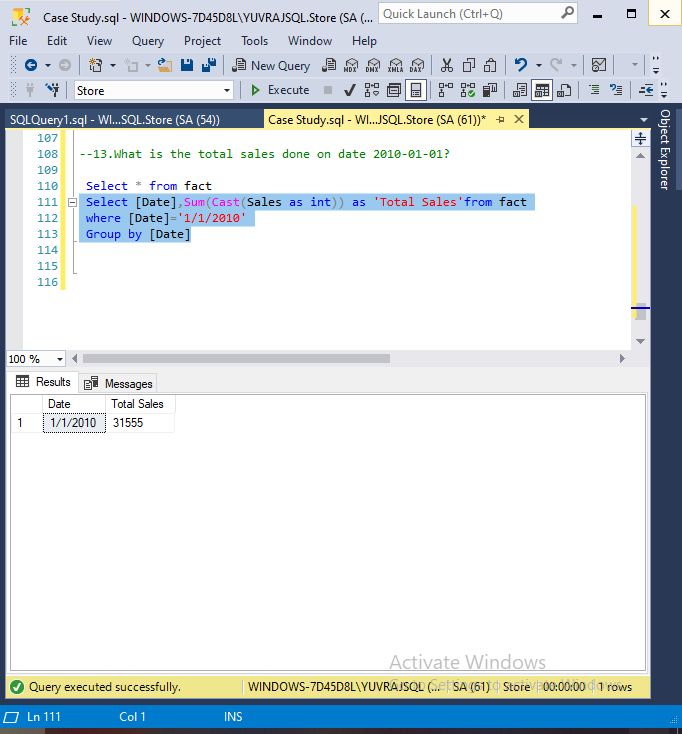
1. Display state in a sequential order in a Location Table.



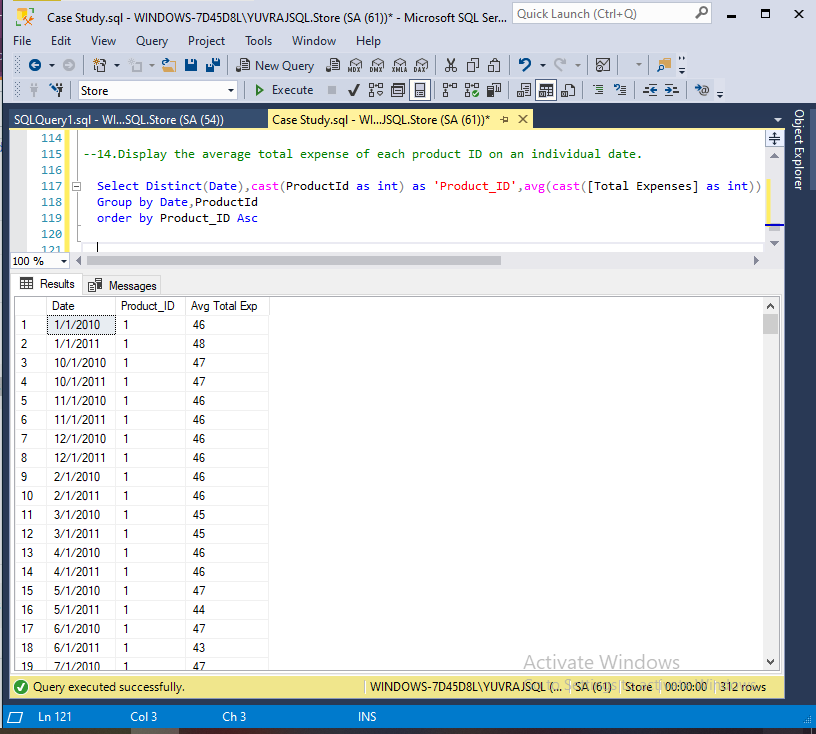
1. Display the average budget of the Product where the average budget margin should be greater than 100.



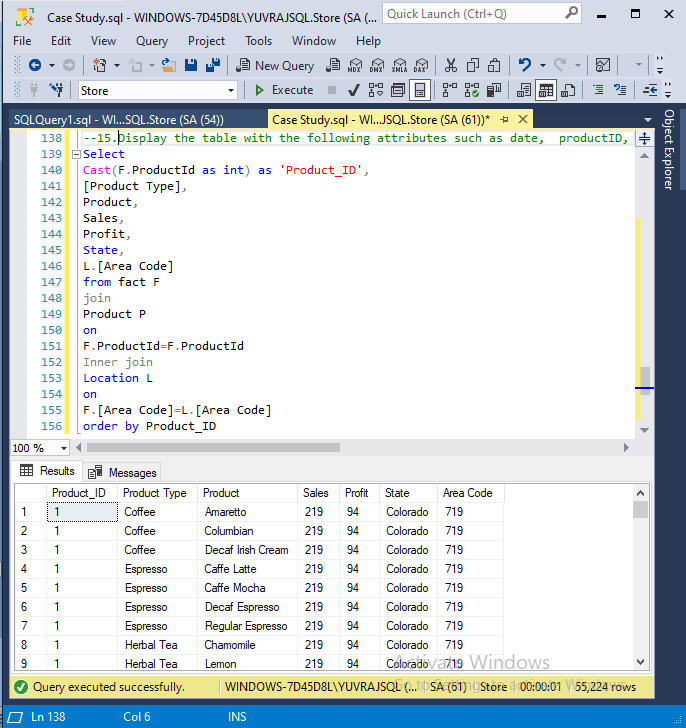
1. What is the total sales done on date 2010-01-01?



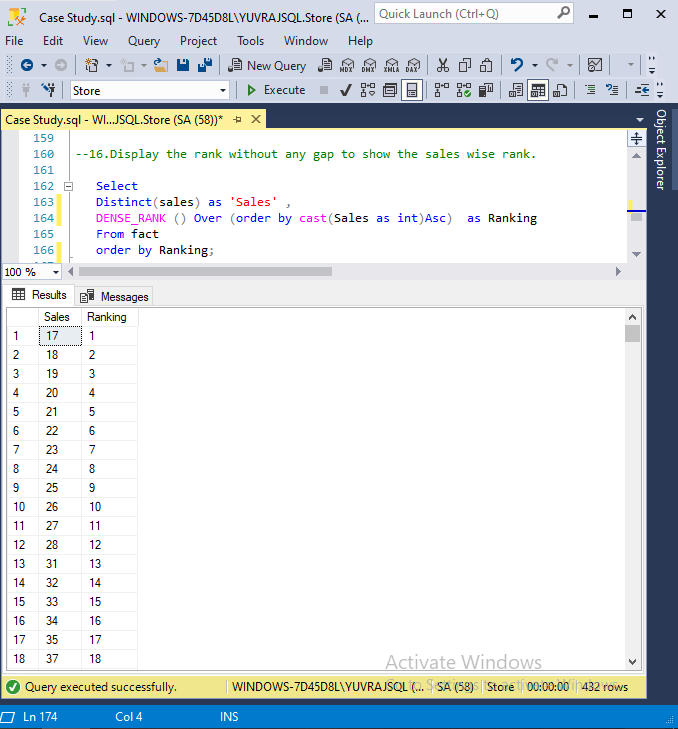
1. Display the average total expense of each product ID on an individual date.



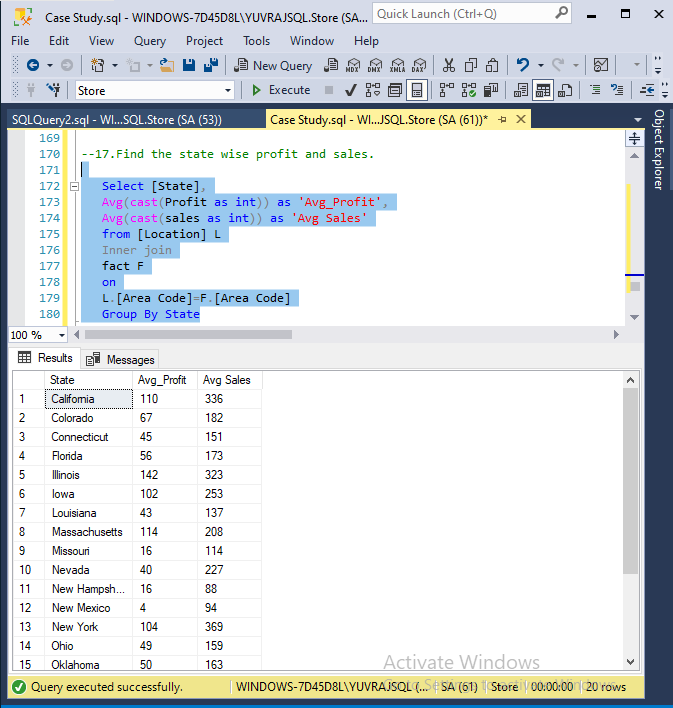
1. Display the table with the following attributes such as date, productID, product\_type, product, sales, profit, state, area\_code.



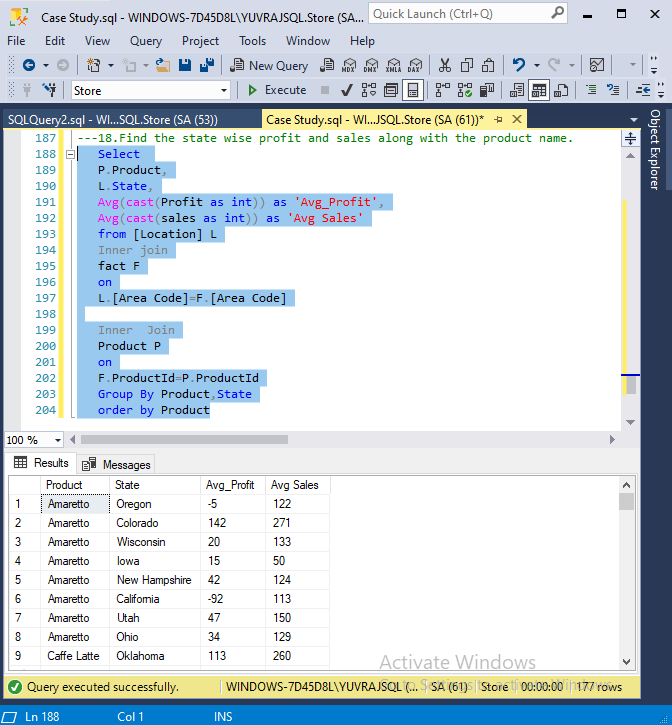
1. Display the rank without any gap to show the sales wise rank.



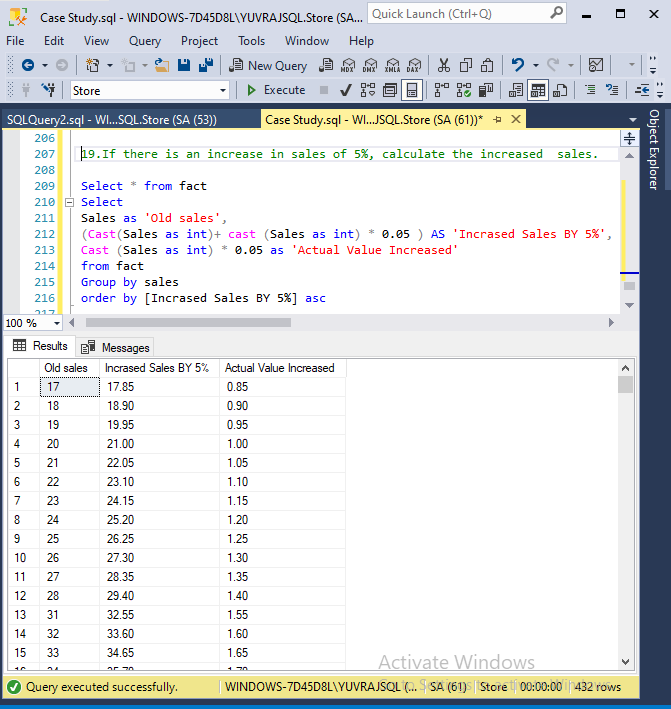
1. Find the state wise profit and sales.



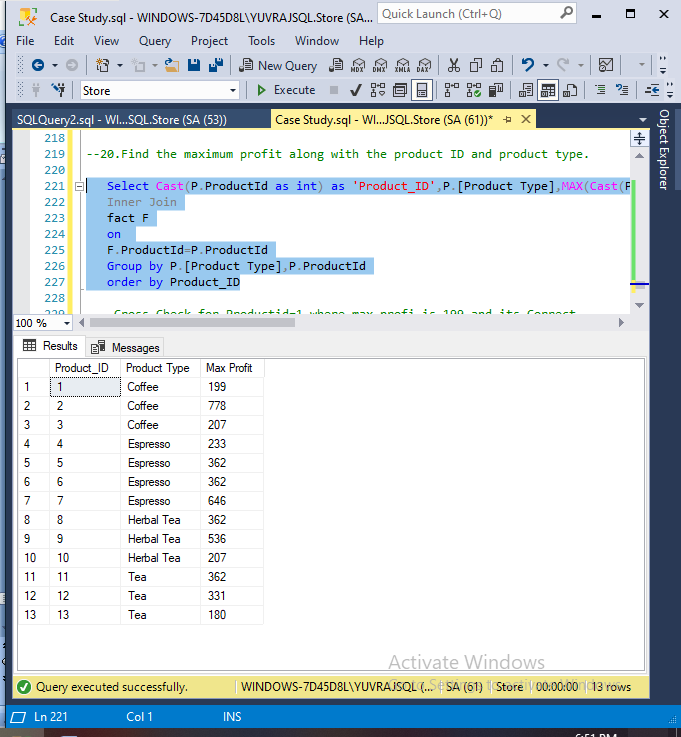
1. Find the state wise profit and sales along with the product name.



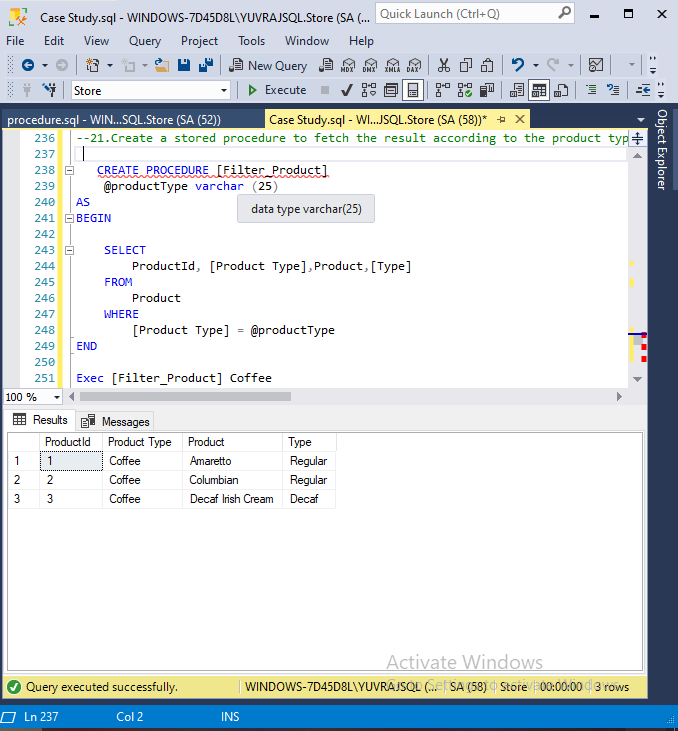
1. If there is an increase in sales of 5%, calculate the increased sales.



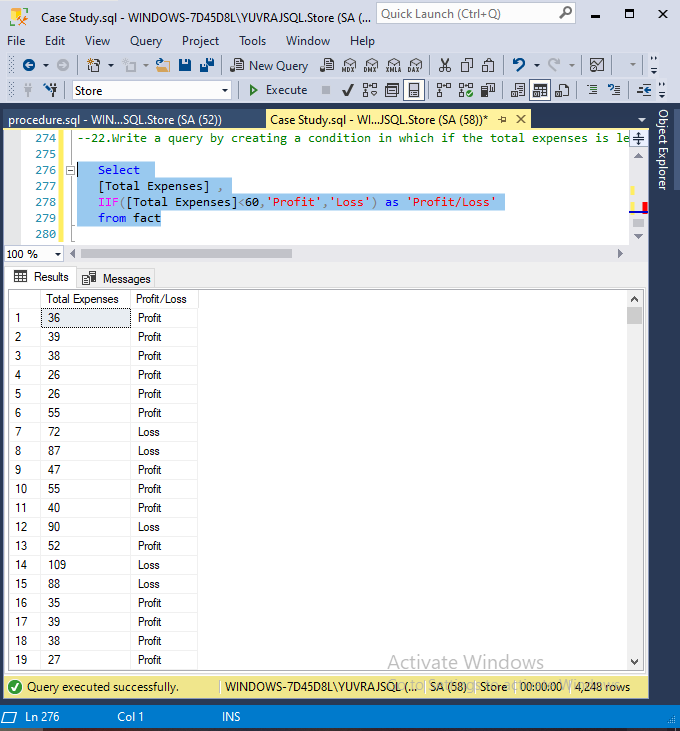
1. Find the maximum profit along with the product ID and product type.



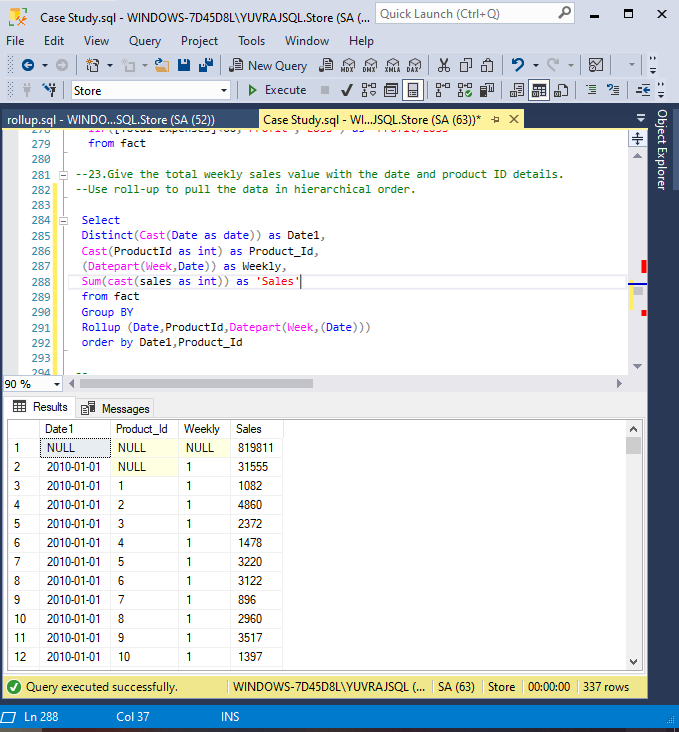
1. Create a stored procedure to fetch the result according to the product type from Product Table.



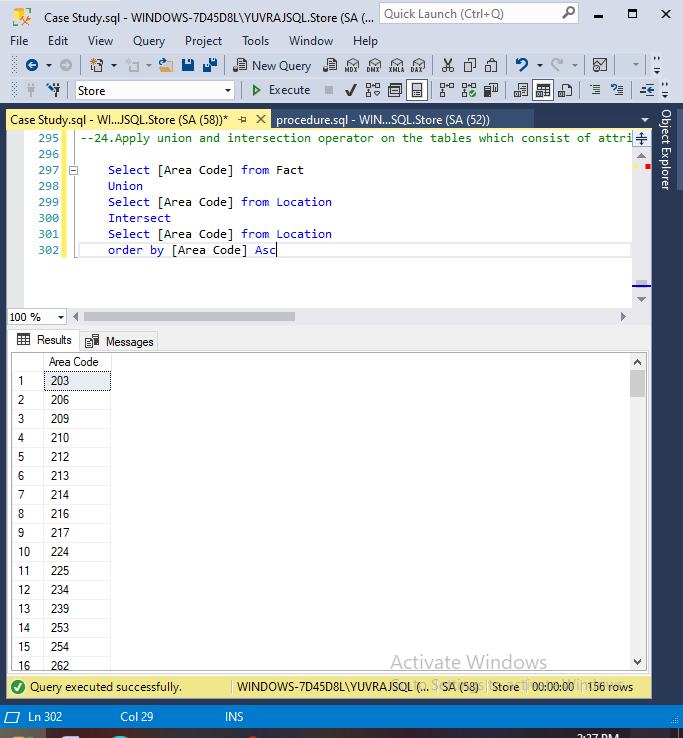
1. Write a query by creating a condition in which if the total expenses is less than 60 then it is a profit or else loss.



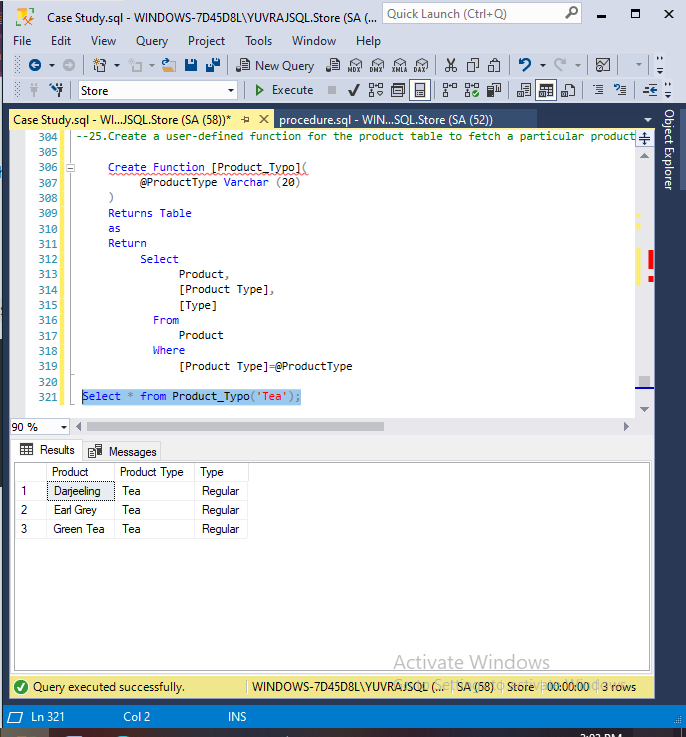
1. Give the total weekly sales value with the date and product ID details. Use roll-up to pull the data in hierarchical order.



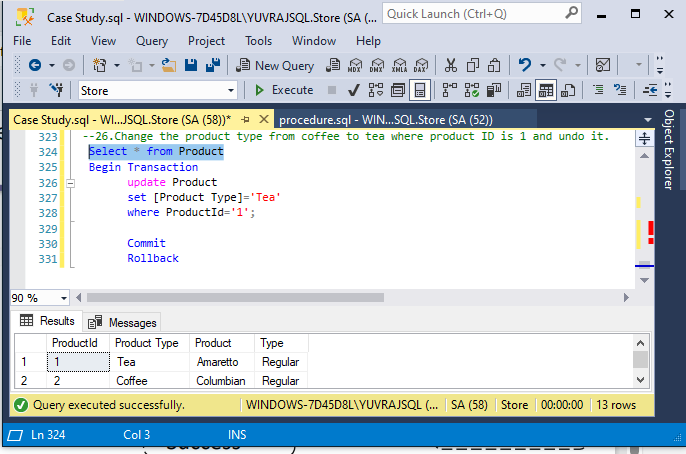
1. Apply union and intersection operator on the tables which consist of attribute area code.

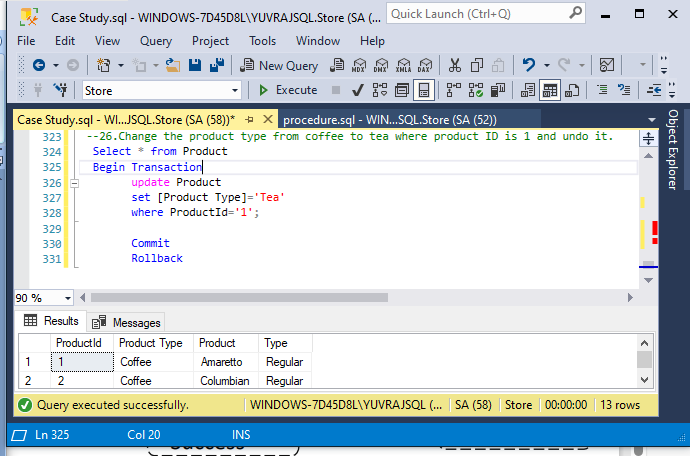


1. Create a user-defined function for the product table to fetch a particular product type based upon the user’s preference.

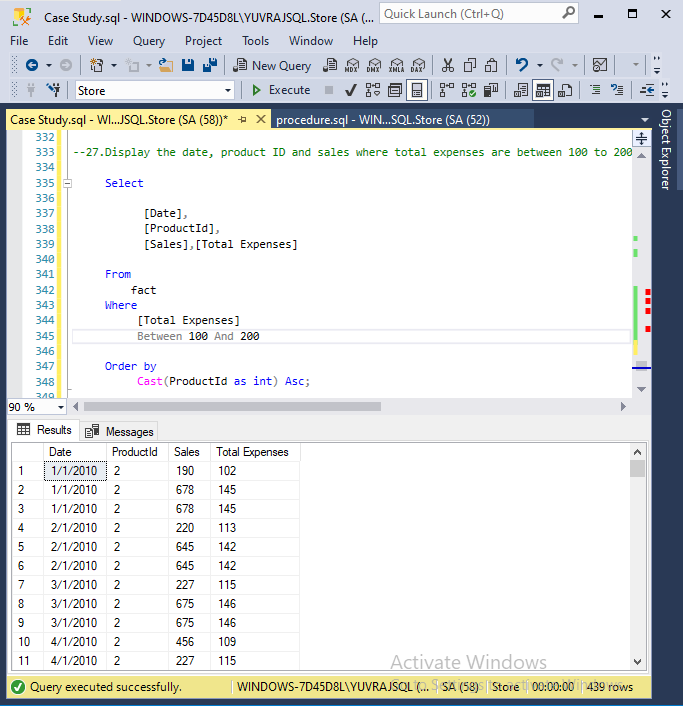


1. Change the product type from coffee to tea where product ID is 1 and undo it.

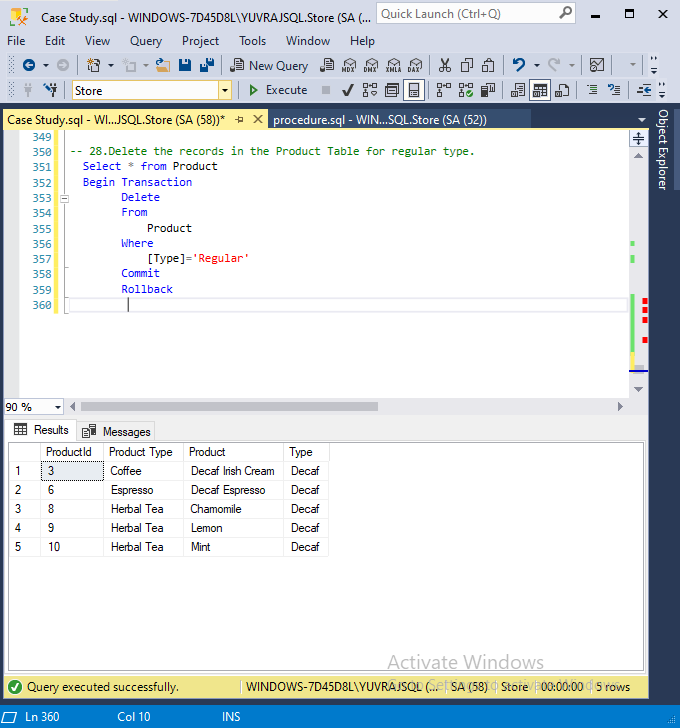




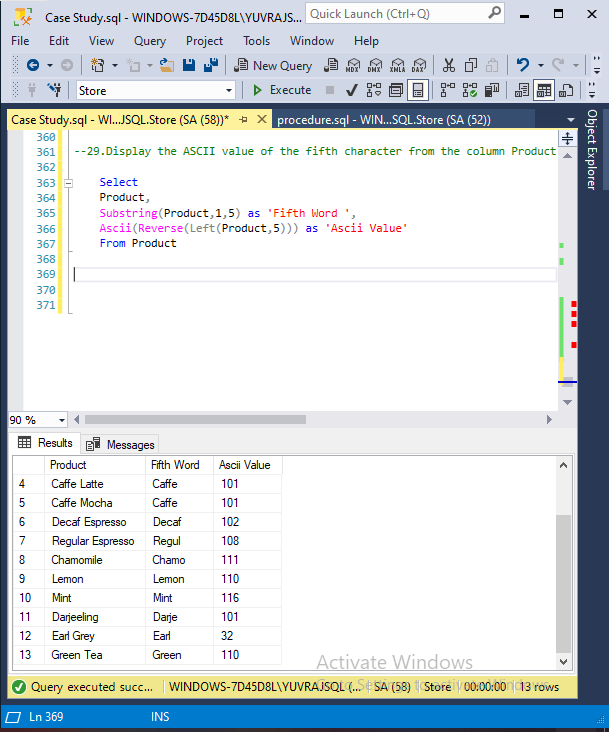
1. Display the date, product ID and sales where total expenses are between 100 to 200.



1. Delete the records in the Product Table for regular type.



1. Display the ASCII value of the fifth character from the column Product.



**CODING SCRIPT**

--1.Display the number of states present in the LocationTable.

select Distinct(COUNT([State])) as 'Number oF State ' from Location

--2.How many products are of regular type?

Select Product,[Type] from Product

where Type='Regular';

--3.How much spending has been done on marketing of product ID 1?

select \* from fact

select ProductId,Sum(Cast(Marketing as int)) as 'Marketing Spending'

from fact

Group by ProductId

having ProductId=1;

--4.What is the minimum sales of a product?

select p.Product, min(cast(sales as int)) as 'Minimum Sales' from Product p

inner join

fact f

on

f.ProductId=p.ProductId

group by P.Product

order by [Minimum Sales] asc

--5.Display the max Cost of Good Sold (COGS).

Select \* from Fact,Location

select Max(cast(COGS as int)) as 'Max COGS Cost' from fact

--6.Display the details of the product where product type is coffee.

Select

P.[Product Type],

P.Product,

F.[Area Code],

F.[Budget COGS],

F.[Budget Margin],

F.[Budget Profit],

F.[Budget Sales],

F.COGS,

F.Date,

F.Inventory,

F.Margin,

F.Marketing,

F.Profit,

F.Sales,

F.[Total Expenses]

from Product P

Right join

Fact F

on

P.ProductId=F.ProductId

Where p.[Product Type]='Coffee'

--7.Display the details where total expenses are greater than 40.

Select \* from fact

where [Total Expenses]>40;

--8.What is the average sales in area code 719?

Select [Area Code],avg(cast(sales as int)) as 'Avg Sales' from fact

Group by [Area Code]

Having [Area Code]= '719' ;

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

-- 303,719,720,970 (Area code Only for reference for cross check)

Select \* from fact

Select \* from Location

select [State],Sum(Cast(F.Profit as int)) as 'Total Profit' from Location L

Full Outer Join

fact F

on

F.[Area Code]=L.[Area Code]

where State='Colorado'

Group by [State]

--10.Display the average inventory for each product ID.

Select \* from fact

Select ProductId,avg(cast(Inventory as int)) as 'Avg Inventary' from fact

Group by ProductId

Order by len(ProductId),ProductId Asc

--11.Display state in a sequential order in a Location Table.

Select \* from Location

Select Distinct[State] from Location

order by [State] Asc

--12.Display the average budget of the Product where the average budget margin should be greater than 100.

Select Product,avg(Cast(F.[Budget Margin] as int)) as 'Avg\_Bmargin'

from Product P

inner join

fact F

on

P.ProductId=F.ProductId

Where F.[Budget Margin]>100

Group by P.Product

--13.What is the total sales done on date 2010-01-01?

Select \* from fact

Select [Date],Sum(Cast(Sales as int)) as 'Total Sales'from fact

where [Date]='1/1/2010'

Group by [Date]

--14.Display the average total expense of each product ID on an individual date.

Select Distinct(Date),cast(ProductId as int) as 'Product\_ID',avg(cast([Total Expenses] as int)) as 'Avg Total Exp' from fact

Group by Date,ProductId

order by Product\_ID Asc

--15.Display the table with the following attributes such as date, productID, product\_type, product, sales, profit, state, area\_code.

Select

Cast(F.ProductId as int) as 'Product\_ID',

[Product Type],

Product,

Sales,

Profit,

State,

L.[Area Code]

from fact F

join

Product P

on

F.ProductId=F.ProductId

Inner join

Location L

on

F.[Area Code]=L.[Area Code]

order by Product\_ID

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

Select

Distinct(sales) as 'Sales' ,

DENSE\_RANK () Over (order by cast(Sales as int) Asc) as Ranking

From fact

order by Ranking;

--17.Find the state wise profit and sales.

Select [State],

Avg(cast(Profit as int)) as 'Avg\_Profit',

Avg(cast(sales as int)) as 'Avg Sales'

from [Location] L

Inner join

fact F

on

L.[Area Code]=F.[Area Code]

Group By State

---18.Find the state wise profit and sales along with the product name.

Select

P.Product,

L.State,

Avg(cast(Profit as int)) as 'Avg\_Profit',

Avg(cast(sales as int)) as 'Avg Sales'

from [Location] L

Inner join

fact F

on

L.[Area Code]=F.[Area Code]

Inner Join

Product P

on

F.ProductId=P.ProductId

Group By Product,State

order by Product

-- 19.If there is an increase in sales of 5%, calculate the increased sales.

Select \* from fact

Select

Sales as 'Old sales',

(Cast(Sales as int)+ cast (Sales as int) \* 0.05 ) AS 'Incrased Sales BY 5%',

Cast (Sales as int) \* 0.05 as 'Actual Value Increased'

from fact

Group by sales

order by [Incrased Sales BY 5%] asc

--20.Find the maximum profit along with the product ID and product type.

Select Cast(P.ProductId as int) as 'Product\_ID',P.[Product Type],MAX(Cast(Profit as int)) as 'Max Profit' from Product P

Inner Join

fact F

on

F.ProductId=P.ProductId

Group by P.[Product Type],P.ProductId

order by Product\_ID

--Cross Check for Productid=1 where max profi is 199 and its Correct

Select \* from Product

Select ProductId,MAX(Cast(Profit as int)) as Max\_Profit from fact

Where ProductId='1'

Group by ProductId

--21.Create a stored procedure to fetch the result according to the product type from Product Table.

CREATE PROCEDURE [Filter\_Product]

@productType varchar (25)

AS

BEGIN

SELECT

ProductId, [Product Type],Product,[Type]

FROM

Product

WHERE

[Product Type] = @productType

END

Exec [Filter\_Product] Coffee

---Extra Query With Join

alter Procedure Product\_Type

@ProductType as Varchar(30)

as

Select

[Product Type],

Product,

[Type],

Sales,

Margin,

COGS,

[Total Expenses]

From Product P

Inner Join

fact F

on

F.ProductId=P.ProductId

where [Product Type]=@ProductType

Exec [Product\_Type] Coffee

--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 'Profit/Loss'

from fact

--23.Give the total weekly sales value with the date and product ID details.

--Use roll-up to pull the data in hierarchical order.

Select

Distinct(Cast(Date as date)) as Date1,

Cast(ProductId as int) as Product\_Id,

(Datepart(Week,Date)) as Weekly,

Sum(cast(sales as int)) as 'Sales'

from fact

Group BY

Rollup (Date,ProductId,Datepart(Week,(Date)))

order by Date1,Product\_Id

--Extra NO use

Select \* from fact

Select

T1.D1,

T1.WeeklySales,

T1.Product\_ID

from

(Select

[Date] as D1,

Sum(cast([Sales] as int)) over(Partition by Datepart(Wk,Date)) as WeeklySales,

Sum(cast([Sales] as int)) over(Partition by (Productid)) as Product\_ID

from fact )as T1

Group By T1.Product\_ID,

Rollup(T1.D1,T1.WeeklySales)

---- Extra No use

Select ProductId,Sales as 'Weekly',Date from fact

Where Date

Between DATEADD(d,-7,Convert(varchar(10),GetDate(),101))

And CONVERT(varchar(10),GetDate(),101)

Group By ProductId

----Extra NO use

Select \* from Fact

Select

DATEPART(Week,[Date]) as Weekly,

COUNT(ProductId) as Product\_ID,

Sales

From Fact

where '20100101' <= Date and '20120112' < Date

Group BY DATEPART (Week,Date),Sales

Order By DATEPART (Week,Date);

--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

Intersect

Select [Area Code] from Location

order by [Area Code] Asc

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

Create Function [Product\_Typo](

@ProductType Varchar (20)

)

Returns Table

as

Return

Select

Product,

[Product Type],

[Type]

From

Product

Where

[Product Type]=@ProductType

Select \* from Product\_Typo('Tea');

--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';

Commit

Rollback

--27.Display the date, product ID and sales where total expenses are between 100 to 200.

Select

[Date],

[ProductId],

[Sales],[Total Expenses]

From

fact

Where

[Total Expenses]

Between 100 And 200

Order by

Cast(ProductId as int) Asc;

-- 28.Delete the records in the Product Table for regular type.

Select \* from Product

Begin Transaction

Delete

From

Product

Where

[Type]='Regular'

Commit

Rollback

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

Select

Product,

Substring(Product,1,5) as 'Fifth Word ',

Ascii(Reverse(Left(Product,5))) as 'Ascii Value'

From Product