

SQL CASE Study 1

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

Query: Select Count (Distinct State) as TotalStates from Location

Select distinct State from Location;

The screenshot shows the SQL Server Management Studio interface. At the top, there's a title bar for 'SQLQuery3.sql - PR...THAKUR\pritu (60)*'. Below it, a query editor window contains two queries:

```
CREATE DATABASE SQL_StudyCase_1
Select Count(Distinct State) as TotalStates from Location
Select distinct State from Location
```

Below the queries, the 'Results' tab is selected, showing a table with one row:

TotalStates
1 20

Below this, another table shows the 20 distinct states:

	State
1	California
2	Colorado
3	Connecticut
4	Florida
5	Illinois
6	Iowa
7	Louisiana
8	Massachusetts
9	Missouri
10	Nevada
11	New Hampshire
12	New Mexico
13	New York
14	Ohio
15	Oklahoma
16	Oregon
17	Texas
18	Utah
19	Washington
20	Wisconsin

2. How many products are of regular type?

Query: Select Count (*) As Regular_Products

FROM Product

WHERE Product.Type = 'Regular';

Select Product_Type, Type from Product where Type = 'Regular'

The screenshot shows the SQL Server Management Studio interface. The title bar reads "SQLQuery3.sql - PR...THAKUR\pritu (60)*". The query pane displays a query tree with the following structure:

```
└─ Select * from Product
    └─ Select Count (*) As Regular_Products
        └─ FROM Product
            └─ WHERE Product.Type = 'Regular';
    └─ Select Product_Type, Type from Product where Type = 'Regular'
```

The "Results" tab is selected, showing a table with one row:

	Regular_Products
1	8

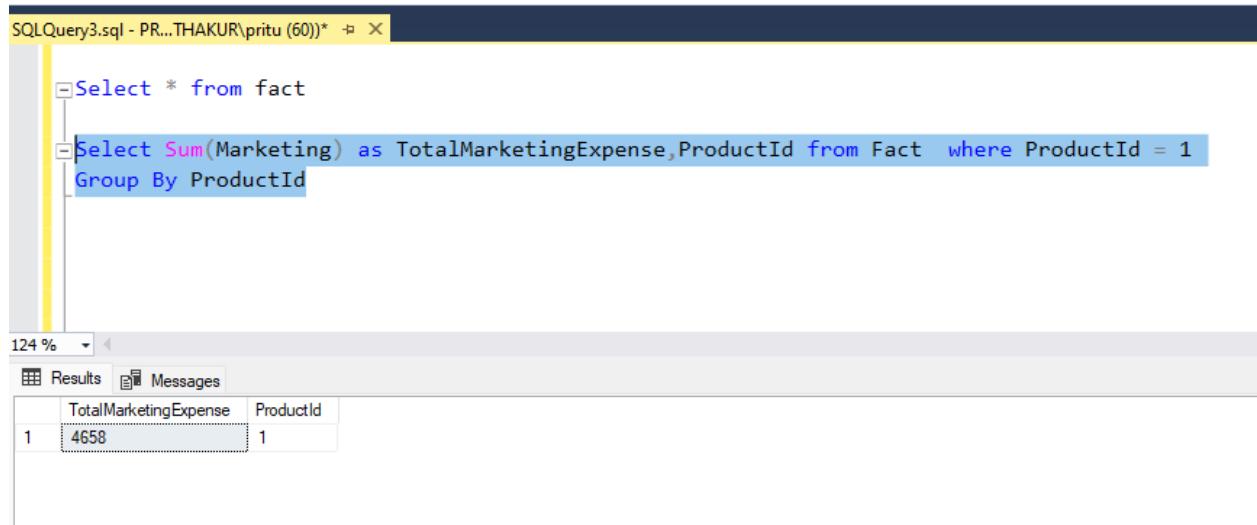
Below the results, another table is shown:

	Product_Type	Type
1	Coffee	Regular
2	Coffee	Regular
3	Espresso	Regular
4	Espresso	Regular
5	Espresso	Regular
6	Tea	Regular
7	Tea	Regular
8	Tea	Regular

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

Query: Select Sum (Marketing) as TotalMarketingExpense,ProductId from Fact where ProductId = 1

Group By ProductId



The screenshot shows a SQL Server Management Studio window titled "SQLQuery3.sql - PR...THAKUR\pritu (60)*". The query pane contains the following code:

```
Select * from fact
Select Sum(Marketing) as TotalMarketingExpense,ProductId from Fact where ProductId = 1
Group By ProductId
```

The results pane shows a single row of data:

	TotalMarketingExpense	ProductId
1	4658	1

4. What is the minimum sales of a product?

Query: Select Min (Sales) as MinimumSales,ProductId from fact group by ProductId order by ProductId asc

A screenshot of a SQL Server Management Studio window. The query pane contains the following T-SQL code:

```
Select Min(Sales) as MinimumSales, ProductId from fact  
group by ProductId  
order by ProductId asc
```

The results pane shows a table with two columns: MinimumSales and ProductId. The data is as follows:

	MinimumSales	ProductId
1	39	1
2	61	2
3	23	3
4	43	4
5	39	5
6	41	6
7	45	7
8	39	8
9	55	9
10	90	10
11	39	11
12	45	12
13	17	13

5. Display the max Cost of Goods Sold (COGS).

Query: 1). Select Max (COGS) from Fact

2). Select * from Fact where COGS = (Select Max(COGS) from Fact)

SQLQuery2.sql - PR...THAKUR\pritu (51)*

```
1 Select Max(COGS) from Fact
2
3 Select * from Fact where COGS = (Select Max(COGS) from Fact)
4
5 Select * from fact
```

123 %

Results Messages

	(No column name)
1	364

	Date	ProductId	Profit	Sales	Margin	COGS	Total_Expenses	Marketing	Inventory	Budget_Profit	Budget_COGS	Budget_Margin	Budget_Sales	Area_Code
1	2010-01-07	2	370	912	548	364	178	127	2654	370	340	520	860	845
2	2010-01-07	2	370	912	548	364	178	127	2654	530	450	690	1140	707
3	2010-01-08	2	368	910	546	364	178	127	2617	370	340	520	860	845
4	2010-01-08	2	368	910	546	364	178	127	2617	520	450	680	1130	650
5	2011-01-07	2	511	904	548	364	166	127	2654	370	340	520	860	716
6	2011-01-07	2	511	904	548	364	166	127	2654	530	450	690	1140	650
7	2011-01-08	2	508	902	546	364	166	127	2617	370	340	520	860	914
8	2011-01-08	2	508	902	546	364	166	127	2617	520	450	680	1130	213

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

Query: Select * from Product where Product_Type='Coffee'

```
Select * from Product where Product_Type='Coffee'
```

123 %

Results Messages

	ProductId	Product_Type	Product	Type
1	1	Coffee	Amaretto	Regular
2	2	Coffee	Columbian	Regular
3	3	Coffee	Decaf Irish Cream	Decaf

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

Query: Select * from Fact where Total_Expenses > 40

	Date	ProductId	Profit	Sales	Margin	COGS	Total_Expenses	Marketing	Inventory	Budget_Profit	Budget_COGS	Budget_Margin	Budget_Sales	Area_Code
112	2010-01-02	6	59	189	114	75	55	24	549	90	80	140	220	303
113	2010-01-02	8	99	340	170	170	71	47	1073	100	140	150	290	970
114	2010-01-02	9	-3	129	75	54	78	49	427	10	40	70	110	970
115	2010-01-02	10	33	140	80	60	47	19	329	40	50	70	120	303
116	2010-01-02	11	18	132	73	59	55	22	331	10	50	60	110	719
117	2010-01-02	2	115	354	206	148	91	48	844	140	150	220	370	312
118	2010-01-02	3	95	254	150	104	55	33	596	100	110	150	260	847
119	2010-01-02	5	201	545	311	234	110	77	1282	260	270	370	640	708
120	2010-01-02	6	142	457	229	228	87	63	1436	180	260	270	530	312
121	2010-01-02	12	59	189	114	75	55	24	549	50	60	100	160	630
122	2010-01-02	8	201	545	311	234	110	77	1282	190	200	270	470	563
123	2010-01-02	9	95	254	150	104	55	33	596	110	80	140	220	563
124	2010-01-02	11	141	457	229	228	88	63	1436	140	190	210	400	563
125	2010-01-02	12	169	534	294	240	125	91	1391	150	200	260	460	515
126	2010-01-02	2	42	193	107	86	65	32	484	50	90	110	200	314
127	2010-01-02	3	46	165	90	75	44	23	956	60	70	100	170	816
128	2010-01-02	13	-4	74	41	33	45	12	193	0	20	40	60	573
129	2010-01-02	8	-8	104	61	43	69	39	256	10	20	60	90	636

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

Query: 1) Select Avg (Sales) as AverageSale,Area_Code from fact group By Area_Code having Area_Code = 719

Or

2) Select Avg (Sales) as AverageSale from fact where Area_Code = 719

SQLQuery2.sql - PR...THAKUR\pritu (51)* X

```
Select Avg(Sales) as AverageSale,Area_Code from fact  
group By Area_Code  
having Area_Code = 719
```

```
Select Avg(Sales) as AverageSale from fact where Area_Code = 719
```

123 %

Results Messages

	AverageSale	Area_Code
1	186	719

	AverageSale
1	186

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

Query: Select Sum(F.Profit) as TotalProfit from Fact F

inner join Location L

on F.Area_Code = L.Area_Code

where L.State= 'Colorado'

The screenshot shows a SQL query being run in a database environment. The query consists of several statements:

```
>Select * from Product  
Select * from Fact  
  
Select Sum(F.Profit) as TotalProfit from Fact F  
inner join Location L  
on F.Area_Code = L.Area_Code  
where L.State= 'Colorado'  
  
Select Area_Code from Location where State= 'Colorado'  
Select Sum(Profit) as TotalProfit from Fact where Area_Code in( 303,719,720,970)
```

The results pane shows a single row of data:

TotalProfit
17743

10. Display the average inventory for each product ID.

Query: Select ProductId,Avg(Inventory) as AverageInventory from Fact
group by ProductId order by ProductId asc

```
Select * from Product
Select * from Fact

Select ProductId,Avg(Inventory) as AverageInventory from Fact
group by ProductId order by ProductId asc
```

123 %

Results Messages

	ProductId	AverageInventory
1	1	741
2	2	707
3	3	838
4	4	255
5	5	756
6	6	755
7	7	879
8	8	712
9	9	718
10	10	1095
11	11	737
12	12	757
13	13	900

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

Query: Select Row_Number() Over(Order By State asc) as RowNumber, State From Location

```
SQLQuery2.sql - FRAUDDETETION (51) | D | X  
Select Row_Number() Over(Order By State asc) as RowNumber, State  
From Location;
```

123 %

Results Messages

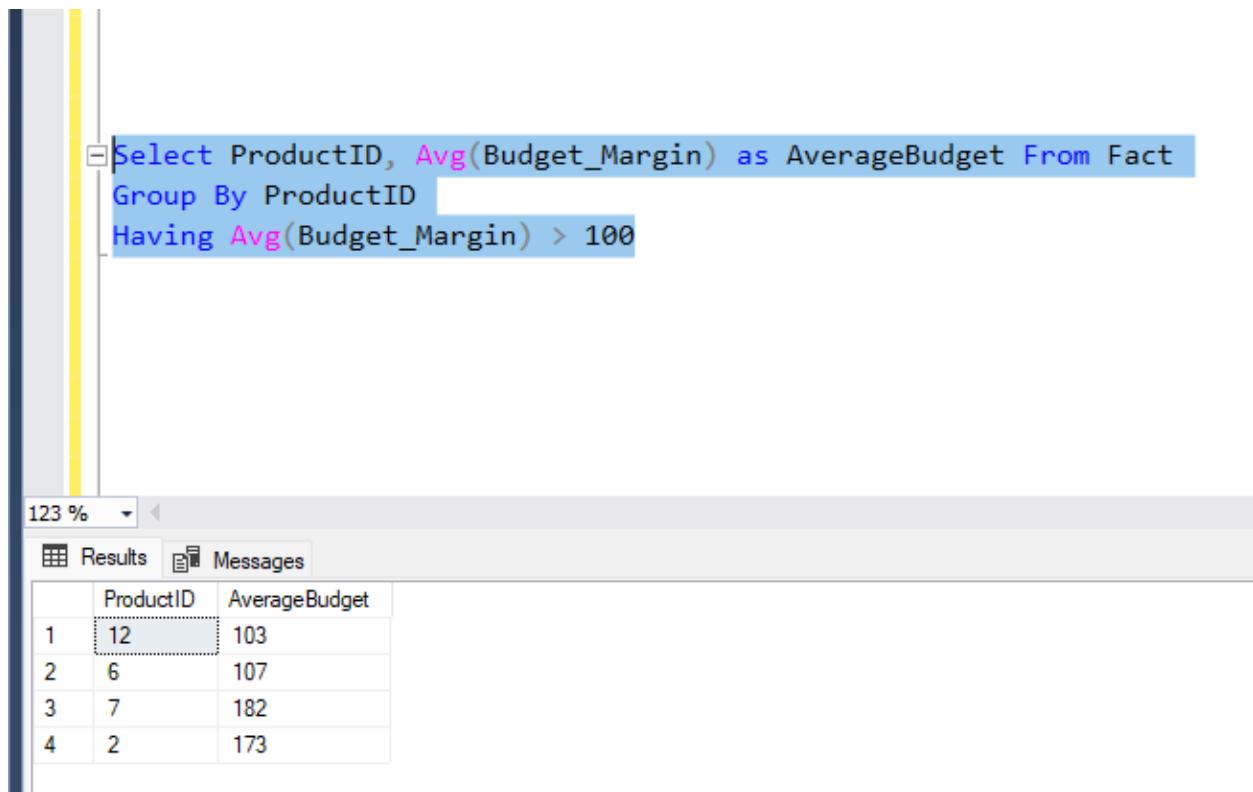
	RowNumber	State
1	1	Califomia
2	2	Califomia
3	3	Califomia
4	4	Califomia
5	5	Califomia
6	6	Califomia
7	7	Califomia
8	8	Califomia
9	9	Califomia
10	10	Califomia
11	11	Califomia
12	12	Califomia
13	13	Califomia
14	14	Califomia

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

Query:

```
Select ProductID, Avg(Budget_Margin) as AverageBudget From Fact  
Group By ProductID
```

```
Having Avg (Budget_Margin) > 100
```



The screenshot shows a SQL query window in SSMS. The query is:

```
Select ProductID, Avg(Budget_Margin) as AverageBudget From Fact  
Group By ProductID  
Having Avg(Budget_Margin) > 100
```

The results grid displays the following data:

	ProductID	AverageBudget
1	12	103
2	6	107
3	7	182
4	2	173

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

Query:

```
Select Sum (Sales) as TotalSales from Fact where Date = '2010-01-01'
```

The screenshot shows a SQL Server Management Studio window titled "SQLQuery2.sql - PR...THAKUR\pritu (51)*". The query pane contains the following code:

```
Select * from Location  
Select * from Product  
Select * from Fact  
  
Select Sum(Sales) as TotalSales from Fact where Date = '2010-01-01'
```

The results pane shows a single row of data:

	TotalSales
1	31555

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

Query:

```
Select Avg(Total_Expenses) as AverageExpense,ProductId,Date from  
Fact  
Group by ProductId,date  
Order By ProductId, Date asc;
```

SQLQuery2.sql - PR...THAKUR\pritu (51)* ✎ ×

```
Select Avg(Total_Expenses) as AverageExpense,ProductId,Date from Fact  
Group by ProductId,date  
Order By ProductId, Date asc
```

123 % ⌂

Results Messages

	AverageExpense	ProductId	Date
1	46	1	2010-01-01
2	46	1	2010-01-02
3	45	1	2010-01-03
4	46	1	2010-01-04
5	47	1	2010-01-05
6	47	1	2010-01-06
7	47	1	2010-01-07
8	47	1	2010-01-08
9	47	1	2010-01-09
10	47	1	2010-01-10
11	46	1	2010-01-11
12	46	1	2010-01-12
13	48	1	2011-01-01

15. Display the table with the following attributes such as date, productID, product_type, product, sales, profit, state, area_code.

Query: Select F.Date, F.ProductID, P.Product_type,
P.Product, F.sales, F.Profit, L.State, L.area_code
FROM FACT F
left join Product P
on P.ProductId=F.ProductId
join Location L
on L.Area_Code=F.Area_Code

SQLQuery2.sql - PR...THAKUR\pritu (51)* X

```
Select F.Date, F.ProductID, P.Product_type,
P.Product, F.sales, F.Profit, L.State, L.area_code
FROM FACT F
left join Product P
on P.ProductId=F.ProductId
join Location L
on L.Area_Code=F.Area_Code
```

123 %

Results Messages

	Date	ProductID	Product_type	Product	sales	Profit	State	area_code
1	2010-01-02	6	Espresso	Decaf Espresso	130	29	Connecticut	203
2	2010-01-03	9	Herbal Tea	Lemon	189	39	Connecticut	203
3	2010-01-03	10	Herbal Tea	Mint	163	46	Connecticut	203
4	2010-01-04	11	Tea	Darjeeling	115	43	Connecticut	203
5	2010-01-04	2	Coffee	Columbian	292	107	Connecticut	203
6	2010-01-05	6	Espresso	Decaf Espresso	113	22	Connecticut	203
7	2010-01-06	13	Tea	Green Tea	51	15	Connecticut	203
8	2010-01-06	5	Espresso	Caffe Mocha	128	-2	Connecticut	203
9	2010-01-06	9	Herbal Tea	Lemon	169	32	Connecticut	203
10	2010-01-07	11	Tea	Darjeeling	87	32	Connecticut	203
11	2010-01-07	6	Espresso	Decaf Espresso	100	16	Connecticut	203
12	2010-01-07	2	Coffee	Columbian	225	75	Connecticut	203
13	2010-01-08	11	Tea	Darjeeling	83	28	Connecticut	203
14	2010-01-08	6	Espresso	Decaf Espresso	98	16	Connecticut	203
15	2010-01-09	11	Tea	Darjeeling	108	41	Connecticut	203
16	2010-01-10	5	Espresso	Caffe Mocha	144	1	Connecticut	203
17	2010-01-11	13	Tea	Green Tea	90	33	Connecticut	203
18	2010-01-11	2	Coffee	Columbian	325	122	Connecticut	203

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

Query: Select ProductId, Sales, Dense_Rank() Over (Order by Sales desc) As Rank From Fact;

SQLQuery1.sql - PR...THAKUR\pritu (72)* X

```
Select ProductId, Sales, Dense_Rank() Over (Order by Sales desc) As Rank
FROM Fact;

SELECT * FROM fact
```

123 %

Results Messages

	ProductId	Sales	Rank
1	2	912	1
2	2	912	1
3	2	910	2
4	2	910	2
5	2	904	3
6	2	904	3
7	2	902	4
8	2	902	4
9	2	890	5
10	2	890	5
11	2	882	6
12	2	882	6
13	2	842	7
14	2	842	7
15	2	815	8

17. Find the state wise profit and sales.

Query:

```
Select L.State, Sum(F.Profit) as Total_Profit, Sum(F.Sales) as Total_Sales
From Fact F JOIN Location L
```

```
On F.Area_Code = L.Area_Code
```

```
Group By L.State
```

```
Order By Total_Profit desc;
```

SQLQuery1.sql - PR...THAKUR\pritu (72)* X

```
SELECT * FROM fact

Select L.State, Sum(F.Profit) as Total_Profit, Sum(F.Sales) as Total_Sales
From Fact F JOIN Location L
On F.Area_Code = L.Area_Code
Group By L.State
Order By Total_Profit desc;
```

123 %

Results Messages

	State	Total_Profit	Total_Sales
1	California	31785	96892
2	Illinois	30821	69883
3	Iowa	22212	54750
4	New York	20096	70852
5	Colorado	17743	48179
6	Massachusetts	16442	29965
7	Texas	15766	37410
8	Oregon	12439	40899
9	Florida	12310	37443
10	Washington	11405	38930
11	Ohio	10773	34517
12	Nevada	10616	60159
13	Wisconsin	8702	33069
14	Oklahoma	8558	27463
15	Utah	7751	35384
16	Connecticut	7621	25429
17	Louisiana	7355	23161
18	Missouri	3601	24647
19	New Hampshire	2748	14887
20	New Mexico	799	15892

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

Query:

```
Select L.State,P.Product, Sum(F.Profit) as Total_Profit, Sum(F.Sales) as
Total_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

Order By L.State,Total_Profit desc;

```
SQLQuery1.sql - PR...THAKUR\pritu (72)* ↵ X
SELECT * FROM Product

Select L.State,P.Product, Sum(F.Profit) as Total_Profit, Sum(F.Sales) as Total_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
Order By L.State,Total_Profit desc;

123 % ▾
Results Messages


|    | State       | Product           | Total_Profit | Total_Sales |
|----|-------------|-------------------|--------------|-------------|
| 1  | California  | Columbian         | 8566         | 18245       |
| 2  | California  | Decaf Espresso    | 6580         | 14607       |
| 3  | California  | Lemon             | 5450         | 12681       |
| 4  | California  | Caffe Latte       | 4497         | 12001       |
| 5  | California  | Darjeeling        | 3418         | 6507        |
| 6  | California  | Chamomile         | 3252         | 6233        |
| 7  | California  | Earl Grey         | 2334         | 4640        |
| 8  | California  | Mint              | 1555         | 3807        |
| 9  | California  | Green Tea         | 1355         | 4027        |
| 10 | California  | Caffe Mocha       | 886          | 7691        |
| 11 | California  | Amaretto          | -2217        | 2714        |
| 12 | California  | Decaf Irish Cream | -3891        | 3739        |
| 13 | Colorado    | Amaretto          | 3410         | 6507        |
| 14 | Colorado    | Decaf Irish Cream | 3250         | 6233        |
| 15 | Colorado    | Chamomile         | 2678         | 7798        |
| 16 | Colorado    | Caffe Mocha       | 2339         | 4640        |
| 17 | Colorado    | Columbian         | 1566         | 3807        |
| 18 | Colorado    | Decaf Espresso    | 1362         | 4027        |
| 19 | Colorado    | Green Tea         | 1272         | 3237        |
| 20 | Colorado    | Earl Grey         | 826          | 2991        |
| 21 | Colorado    | Mint              | 815          | 3102        |
| 22 | Colorado    | Darjeeling        | 366          | 2902        |
| 23 | Colorado    | Lemon             | -141         | 2935        |
| 24 | Connecticut | Columbian         | 2999         | 6923        |


```

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

Query:

-----Increased Sales with Product Name and state wise-----

```
Select L.State,P.Product,Sum(F.Sales) as Original_Sales,Sum(F.Sales) *  
1.05 as Increased_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
```

```
Order By L.State, Original_Sales desc;
```

-----Increased Sale from fact Table-----

```
Select ProductId,Sales as Original_Sales,Sales * 1.05 as Increased_Sales  
From Fact;
```

-----Increased sale for Total sales-----

```
Select ProductId,Sum (Sales) as Total_Original_Sales, Sum (Sales) * 1.05  
AS Total_Increased_Sales
```

```
From Fact
```

```
Group By ProductId
```

```
Order By ProductId;
```

```

SQLQuery1.sql - PR...THAKUR\pritu (72)* ↗ X
-----Increased Sales with Product Name and state wise-----
Select L.State,P.Product,Sum(F.Sales) as Original_Sales,Sum(F.Sales) * 1.05 as Increased_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
Order By L.State, Original_Sales desc;
-----Increased Sale from fact Table-----
Select ProductId,Sales as Original_Sales,Sales * 1.05 as Increased_Sales
From Fact;
-----Increased sale for Total sales-----
Select ProductId,Sum (Sales) as Total_Original_Sales, Sum (Sales) * 1.05 AS Total_Increased_Sales
From Fact
Group By ProductId
Order By ProductId;

```

123 %

	Results	Messages																																				
1	<table border="1"> <thead> <tr> <th>State</th> <th>Product</th> <th>Original_Sales</th> <th>Increased_Sales</th> </tr> </thead> <tbody> <tr><td>1 California</td><td>Columbian</td><td>18245</td><td>19157.25</td></tr> <tr><td>2 California</td><td>Decaf Espresso</td><td>14607</td><td>15337.35</td></tr> <tr><td>3 California</td><td>Lemon</td><td>12681</td><td>13315.05</td></tr> <tr><td>4 California</td><td>Caffe Latte</td><td>12001</td><td>12601.05</td></tr> <tr><td>5 California</td><td>Caffe Mocha</td><td>7691</td><td>8075.55</td></tr> <tr><td>6 California</td><td>Darjeeling</td><td>6507</td><td>6832.35</td></tr> <tr><td>7 California</td><td>Chamomile</td><td>6233</td><td>6544.65</td></tr> <tr><td>8 California</td><td>Earl Grey</td><td>4640</td><td>4872.00</td></tr> </tbody> </table>	State	Product	Original_Sales	Increased_Sales	1 California	Columbian	18245	19157.25	2 California	Decaf Espresso	14607	15337.35	3 California	Lemon	12681	13315.05	4 California	Caffe Latte	12001	12601.05	5 California	Caffe Mocha	7691	8075.55	6 California	Darjeeling	6507	6832.35	7 California	Chamomile	6233	6544.65	8 California	Earl Grey	4640	4872.00	
State	Product	Original_Sales	Increased_Sales																																			
1 California	Columbian	18245	19157.25																																			
2 California	Decaf Espresso	14607	15337.35																																			
3 California	Lemon	12681	13315.05																																			
4 California	Caffe Latte	12001	12601.05																																			
5 California	Caffe Mocha	7691	8075.55																																			
6 California	Darjeeling	6507	6832.35																																			
7 California	Chamomile	6233	6544.65																																			
8 California	Earl Grey	4640	4872.00																																			
2	<table border="1"> <thead> <tr> <th>ProductId</th> <th>Original_Sales</th> <th>Increased_Sales</th> </tr> </thead> <tbody> <tr><td>1</td><td>219</td><td>229.95</td></tr> <tr><td>2</td><td>190</td><td>199.50</td></tr> <tr><td>3</td><td>234</td><td>245.70</td></tr> <tr><td>4</td><td>100</td><td>105.00</td></tr> <tr><td>5</td><td>134</td><td>140.70</td></tr> <tr><td>6</td><td>180</td><td>189.00</td></tr> <tr><td>7</td><td>341</td><td>358.05</td></tr> <tr><td>8</td><td>150</td><td>157.50</td></tr> </tbody> </table>	ProductId	Original_Sales	Increased_Sales	1	219	229.95	2	190	199.50	3	234	245.70	4	100	105.00	5	134	140.70	6	180	189.00	7	341	358.05	8	150	157.50										
ProductId	Original_Sales	Increased_Sales																																				
1	219	229.95																																				
2	190	199.50																																				
3	234	245.70																																				
4	100	105.00																																				
5	134	140.70																																				
6	180	189.00																																				
7	341	358.05																																				
8	150	157.50																																				
3	<table border="1"> <thead> <tr> <th>ProductId</th> <th>Total_Original_Sales</th> <th>Total_Increased_Sales</th> </tr> </thead> <tbody> <tr><td>1</td><td>26269</td><td>27582.45</td></tr> <tr><td>2</td><td>128311</td><td>134726.55</td></tr> <tr><td>3</td><td>62248</td><td>65360.40</td></tr> <tr><td>4</td><td>35899</td><td>37693.95</td></tr> <tr><td>5</td><td>84904</td><td>89149.20</td></tr> <tr><td>6</td><td>78162</td><td>82070.10</td></tr> <tr><td>7</td><td>24031</td><td>25232.55</td></tr> </tbody> </table>	ProductId	Total_Original_Sales	Total_Increased_Sales	1	26269	27582.45	2	128311	134726.55	3	62248	65360.40	4	35899	37693.95	5	84904	89149.20	6	78162	82070.10	7	24031	25232.55													
ProductId	Total_Original_Sales	Total_Increased_Sales																																				
1	26269	27582.45																																				
2	128311	134726.55																																				
3	62248	65360.40																																				
4	35899	37693.95																																				
5	84904	89149.20																																				
6	78162	82070.10																																				
7	24031	25232.55																																				

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

Query:

```

SELECT F.ProductID,P.Product_Type,MAX(F.Profit) as Maximum_Profit
FROM Fact F
JOIN Product P

```

```

ON F.ProductID = P.ProductID
GROUP BY F.ProductID, P.Product_Type
ORDER BY ProductId,Maximum_Profit DESC;

```

The screenshot shows a SQL query window titled "SQLQuery1.sql - PR...THAKUR\pritu (55)*" containing the following T-SQL code:

```

SELECT F.ProductID, P.Product_Type, MAX(F.Profit) as Maximum_Profit
FROM Fact F
JOIN Product P
ON F.ProductID = P.ProductID
GROUP BY F.ProductID, P.Product_Type
ORDER BY ProductId, Maximum_Profit DESC

Select * from Product

```

Below the query window is a results grid titled "Results". The data is presented in a table with columns: ProductID, Product_Type, and Maximum_Profit. The data rows are:

	ProductID	Product_Type	Maximum_Profit
1	1	Coffee	199
2	2	Coffee	778
3	3	Coffee	207
4	4	Espresso	233
5	5	Espresso	362
6	6	Espresso	362
7	7	Espresso	646
8	8	Herbal Tea	362
9	9	Herbal Tea	536
10	10	Herbal Tea	207
11	11	Tea	362
12	12	Tea	331
13	13	Tea	180

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

Query: Create Procedure DetailsByProduct

```

@InputProduct Nvarchar(100)

As

Begin
    Select *
        From Product
    Where Product_Type = @InputProduct;

End;

```

Exec DetailsByProduct @InputProduct='Coffee'

The screenshot shows a SQL Server Management Studio window with two panes. The top pane displays the creation of a stored procedure named 'DetailsByProduct' with the following script:

```

Create Procedure DetailsByProduct
@InputProduct Nvarchar(100)
As
Begin
    Select *
        From Product
    Where Product_Type = @InputProduct;
End;

```

The bottom pane shows the execution of the stored procedure with the parameter '@InputProduct='Coffee'' and the resulting data from the 'Product' table:

	ProductId	Product_Type	Product	Type
1	1	Coffee	Amaretto	Regular
2	2	Coffee	Columbian	Regular
3	3	Coffee	Decaf Irish Cream	Decaf

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.

Query:

Select Date,ProductId,Total_Expenses,

Case when Total_Expenses < 60 Then 'Profit'

Else 'Loss'

End as ProfitOrLoss

from Fact;

The screenshot shows a SQL Server Management Studio interface. The top bar has tabs for 'SQLQuery4.sql - PR...' and 'SQLQuery3.sql - PR...'. The main area contains a query script:

```
Select Date, ProductId, Total_Expenses,  
Case when Total_Expenses < 60 Then 'Profit'  
Else 'Loss'  
End as ProfitOrLoss  
from Fact;
```

Below the script is a results grid with two tabs: 'Results' and 'Messages'. The 'Results' tab displays the following data:

	Date	ProductId	Total_Expenses	ProfitOrLoss
1	2010-01-01	1	36	Profit
2	2010-01-01	2	39	Profit
3	2010-01-01	3	38	Profit
4	2010-01-01	13	26	Profit
5	2010-01-01	5	26	Profit
6	2010-01-01	6	55	Profit
7	2010-01-01	8	72	Loss
8	2010-01-01	9	87	Loss
9	2010-01-01	10	47	Profit
10	2010-01-01	11	55	Profit
11	2010-01-01	12	40	Profit
12	2010-01-01	2	90	Loss
13	2010-01-01	3	52	Profit
14	2010-01-01	5	109	Loss
15	2010-01-01	6	88	Loss
16	2010-01-01	8	35	Profit
17	2010-01-01	9	39	Profit
18	2010-01-01	10	38	Profit

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

Query: SELECT

```
    CONCAT(
        FORMAT(Date, 'yyyy-MM'), ' - Week ',
        DATEPART(WEEK, Date) - DATEPART(WEEK, DATEADD(MONTH,
DATEDIFF(MONTH, 0, Date), 0)) + 1
    ) AS WeekOfMonthYear,
--    YEAR(Date) AS YearNum,
--    MONTH(Date) AS MonthNum,
    ProductID,
    SUM(Sales) AS TotalSales
FROM Fact
GROUP BY ROLLUP (
    CONCAT(
        FORMAT(Date, 'yyyy-MM'), ' - Week ',
        DATEPART(WEEK, Date) - DATEPART(WEEK, DATEADD(MONTH,
DATEDIFF(MONTH, 0, Date), 0)) + 1
    ),
    ProductID)
ORDER BY TotalSales;
```

```

SELECT
    CONCAT(
        FORMAT(Date, 'yyyy-MM'), ' - Week ',
        DATEPART(WEEK, Date) - DATEPART(WEEK, DATEADD(MONTH, DATEDIFF(MONTH, 0, Date), 0)) + 1
    ) AS WeekOfMonthYear,
    -- YEAR(Date) AS YearNum,
    -- MONTH(Date) AS MonthNum,
    ProductID,
    SUM(Sales) AS TotalSales
FROM Fact
GROUP BY ROLLUP (
    CONCAT(
        FORMAT(Date, 'yyyy-MM'), ' - Week ',
        DATEPART(WEEK, Date) - DATEPART(WEEK, DATEADD(MONTH, DATEDIFF(MONTH, 0, Date), 0)) + 1
    ),
    ProductID)
ORDER BY TotalSales;

```

123 %

	WeekOfMonthYear	ProductID	TotalSales
1	2011-01 - Week 1	7	1003
2	2011-01 - Week 1	1	1210
3	2011-01 - Week 1	13	1399
4	2011-01 - Week 1	10	1564
5	2011-01 - Week 1	4	1653
6	2010-01 - Week 1	7	1884
7	2010-01 - Week 1	1	2155
8	2010-01 - Week 1	13	2534
9	2011-01 - Week 1	3	2656
10	2010-01 - Week 1	10	2814
11	2011-01 - Week 1	12	2867
12	2010-01 - Week 1	4	2941
13	2011-01 - Week 1	11	3178
14	2010-01 - Week 3	7	3265
15	2010-01 - Week 3	1	3283

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

Query:

SELECT Area_Code FROM Fact

UNION

SELECT Area_Code FROM Location

Order By Area_Code asc;

```
SELECT Area_Code FROM Fact
```

```
INTERSECT
```

```
SELECT Area_Code FROM Location
```

```
ORDER BY AREA_CODE ASC;
```

SQLQuery4.sql - PR...THAKUR\pritu (58)* ✎ × SQLQuery3.

```
select * from fact
Select * from Location

SELECT Area_Code FROM Fact
UNION
SELECT Area_Code FROM Location
Order By Area_Code asc;
```

123 %

Results Messages

	Area_Code
1	203
2	206
3	209
4	210
5	212
6	213
7	214
8	216
9	217
10	224
11	225
12	234
13	239
14	253
15	254
16	262
17	281
18	303
19	305
20	309
21	310
22	312

SQLQuery4.sql - PR...THAKUR\pritu (58)* ➔ X SQLQuery3

```
SELECT Area_Code FROM Fact
INTERSECT
SELECT Area_Code FROM Location
ORDER BY AREA_CODE ASC;
```

123 %

Results Messages

	Area_Code
1	203
2	206
3	209
4	210
5	212
6	213
7	214
8	216
9	217
10	224
11	225
12	234
13	239
14	253
15	254
16	262
17	281
18	303
19	305
20	309
21	310
22	312
23	314
24	315
25	318
26	319
27	321
28	323

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

Query: CREATE FUNCTION DetailsByProductType

(@InputProduct VARCHAR(100))

RETURNS TABLE

AS

RETURN

(

SELECT ProductId, Product, Product_Type

FROM Product

WHERE Product_Type = @InputProduct

);

Select * from DetailsByProductType('Tea')

The screenshot shows a SQL Server Management Studio interface with two tabs: 'SQLQuery4.sql - PR...THAKUR\pritu (58)*' and 'SQLQuery3.sql - PR...THAKUR\pritu (55)*'. The code in the window is as follows:

```
select * from Product
Select * from Location

CREATE FUNCTION DetailsByProductType
(@InputProduct VARCHAR(100))
RETURNS TABLE
AS
RETURN
(
    SELECT ProductId, Product, Product_Type
    FROM Product
    WHERE Product_Type = @InputProduct
);

Select * from DetailsByProductType('Tea')
```

The results grid shows the following data:

	ProductId	Product	Product_Type
1	11	Darjeeling	Tea
2	12	Earl Grey	Tea
3	13	Green Tea	Tea

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

Query:

Begin Tran

Update Product

Set Product_Type ='Tea' where ProductId = 1

Rollback

The screenshot shows a SQL query window titled "SQLQuery1.sql - PR...THAKUR\pritu (55)*". The query is:

```
Select * from Product  
Begin Tran  
Update Product  
Set Product_Type = 'Tea' where ProductId = 1
```

The results tab displays a table with 13 rows of product data. The first row, which has ProductId 1 and Product_Type 'Tea', is highlighted with a green border.

	ProductId	Product_Type	Product	Type
1	1	Tea	Amaretto	Regular
2	2	Coffee	Columbian	Regular
3	3	Coffee	Decaf Irish Cream	Decaf
4	4	Espresso	Caffe Latte	Regular
5	5	Espresso	Caffe Mocha	Regular
6	6	Espresso	Decaf Espresso	Decaf
7	7	Espresso	Regular Espresso	Regular
8	8	Herbal Tea	Chamomile	Decaf
9	9	Herbal Tea	Lemon	Decaf
10	10	Herbal Tea	Mint	Decaf
11	11	Tea	Darjeeling	Regular
12	12	Tea	Earl Grey	Regular
13	13	Tea	Green Tea	Regular

Rollbacked

The screenshot shows a SQL query window titled "SQLQuery1.sql - PR...THAKUR\pritu (55)*". The query is:

```
Select * from Product  
Begin Tran  
Update Product  
Set Product_Type = 'Tea' where ProductId = 1  
Rollback
```

The results tab displays the following table:

	ProductId	Product_Type	Product	Type
1	1	Coffee	Amaretto	Regular
2	2	Coffee	Columbian	Regular
3	3	Coffee	Decaf Irish Cream	Decaf
4	4	Espresso	Caffe Latte	Regular
5	5	Espresso	Caffe Mocha	Regular
6	6	Espresso	Decaf Espresso	Decaf
7	7	Espresso	Regular Espresso	Regular
8	8	Herbal Tea	Chamomile	Decaf
9	9	Herbal Tea	Lemon	Decaf
10	10	Herbal Tea	Mint	Decaf
11	11	Tea	Darjeeling	Regular
12	12	Tea	Earl Grey	Regular
13	13	Tea	Green Tea	Regular

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

Query:

```
Select Date,ProductID,Sales,Total_Expenses from Fact where  
Total_Expenses between 100 and 200.
```

SQLQuery1.sql - PR...THAKUR\pritu (55)*

```
Select Date, ProductID, Sales, Total_Expenses from Fact  
where Total_Expenses between 100 and 200
```

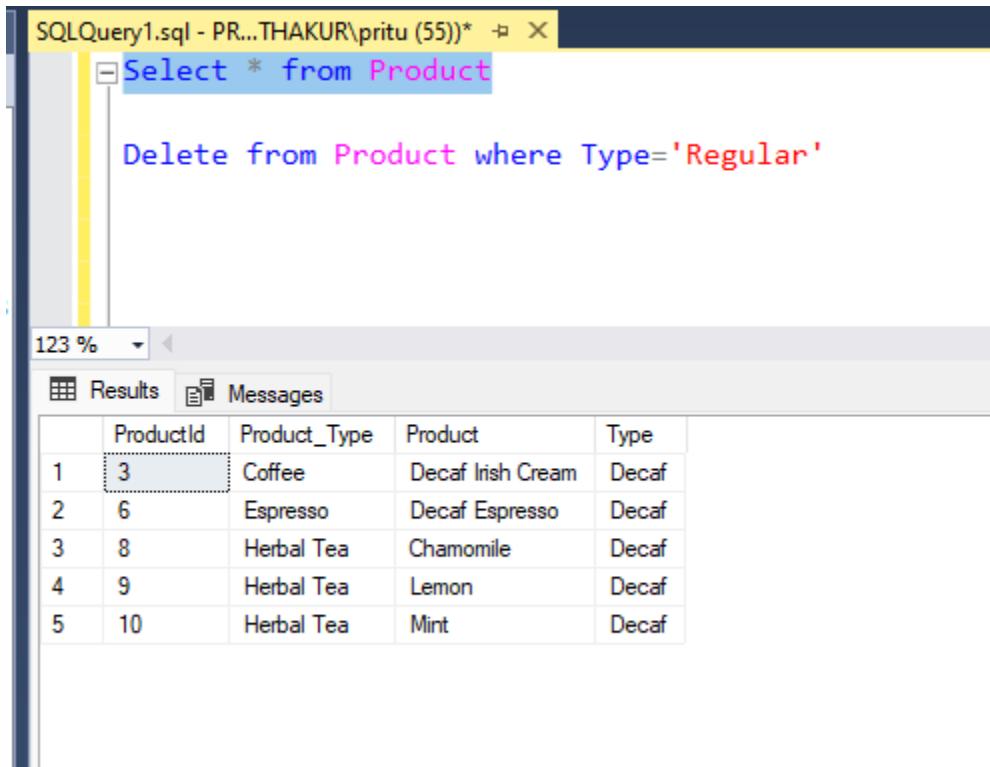
123 %

Results Messages

	Date	ProductID	Sales	Total_Expenses
1	2010-01-01	5	546	109
2	2010-01-01	8	546	110
3	2010-01-01	12	546	126
4	2010-01-01	2	190	102
5	2010-01-01	5	190	102
6	2010-01-01	5	61	128
7	2010-01-01	2	678	145
8	2010-01-01	9	483	127
9	2010-01-01	9	190	101
10	2010-01-01	5	250	128
11	2010-01-01	6	546	109
12	2010-01-01	2	678	145
13	2010-01-01	8	250	129
14	2010-01-01	11	546	109
15	2010-01-01	13	17	126
16	2010-01-01	3	190	102
17	2010-01-02	5	545	110
18	2010-01-02	8	545	110
19	2010-01-02	12	534	125
20	2010-01-02	2	220	113
21	2010-01-02	5	220	112
22	2010-01-02	5	61	143
23	2010-01-02	2	645	142
24	2010-01-02	9	495	124
25	2010-01-02	9	220	113
26	2010-01-02	5	290	143
27	2010-01-02	6	545	109
28	2010-01-02	2	645	142
29	2010-01-02	8	290	143
30	2010-01-02	11	545	109
31	2010-01-02	13	23	125
32	2010-01-02	3	220	113
33	2010-01-03	5	567	113
34	2010-01-03	8	567	112
35	2010-01-03	12	546	126

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

Query: Delete from Product where Type='Regular'



```
SQLQuery1.sql - PR...THAKUR\pritu (55)* X
Select * from Product

Delete from Product where Type='Regular'

123 % < >
Results Messages
```

	ProductId	Product_Type	Product	Type
1	3	Coffee	Decaf Irish Cream	Decaf
2	6	Espresso	Decaf Espresso	Decaf
3	8	Herbal Tea	Chamomile	Decaf
4	9	Herbal Tea	Lemon	Decaf
5	10	Herbal Tea	Mint	Decaf

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

Query: SELECT Product,

```
    ASCII(SUBSTRING(Product, 5, 1)) AS FifthCharacterASCII
```

```
FROM Product;
```

SQLQuery4.sql - PR...THAKUR\pritu (58)* → X SQLQuery3.sql - PR...THAKUR\pritu (55)*

```
SELECT Product,
       ASCII(SUBSTRING(Product, 5, 1)) AS FifthCharacterASCII
  FROM Product;
```

123 %

Results Messages

	Product	FifthCharacterASCII
1	Amaretto	101
2	Columbian	109
3	Decaf Irish Cream	102
4	Caffe Latte	101
5	Caffe Mocha	101
6	Decaf Espresso	102
7	Regular Espresso	108
8	Chamomile	111
9	Lemon	110
10	Mint	NULL
11	Darjeeling	101
12	Earl Grey	32
13	Green Tea	110