

SQL Case Study – 1

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.

Dataset:

The 3 key datasets for this case study:

a. **FactTable:** The Fact Table has 14 columns mentioned below and 4200 rows. Date, ProductID, Profit, Sales, Margin, COGS, Total Expenses, Marketing, Inventory, Budget Profit, Budget COGS, Budget Margin, Budget Sales, and Area Code

Note: COGS stands for Cost of Goods Sold

b. **ProductTable:** The ProductTable has four columns named Product Type, Product, ProductID, and Type. It has 13 rows which can be broken down into further details to retrieve the information mentioned in theFactTable.

c. **LocationTable:** Finally, the LocationTable has 156 rows and follows a similar approach to ProductTable. It has four columns named Area Code, State, Market, and Market Size.

Tasks to be performed:

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

- `Select Distinct(State) From Location Order By State`

| | State |
|----|----------------|
| 1 | California |
| 2 | Colorado |
| 3 | Connecticut |
| 4 | Florida |
| 5 | Illinois |
| 6 | Iowa |
| 7 | Louisiana |
| 8 | Massachusetts |
| 9 | Missouri |
| 10 | Nevada |
| 11 | New Hampshi... |
| 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?

- `Select Product From Product Where Type = 'Regular'`

| | Product |
|---|------------------|
| 1 | Amaretto |
| 2 | Columbian |
| 3 | Caffe Latte |
| 4 | Caffe Mocha |
| 5 | Regular Espresso |
| 6 | Darjeeling |
| 7 | Earl Grey |
| 8 | Green Tea |

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

- `Select sum(Marketing) As Marketing_Expenses From fact Where ProductId = 1`

| | Marketing_Expenses |
|---|--------------------|
| 1 | 4658 |

4. What is the minimum sales of a product?

- `Select Min(Sales) As Minimum_Sales From fact`

| | Minimum_Sales |
|---|---------------|
| 1 | 17 |

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

- `Select Max(COGS) As Max_Cost_of_Good_Sold From fact`

| | Max_Cost_of_Good_Sold |
|---|-----------------------|
| 1 | 364 |

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

- `Select * From Fact as F`
`Join Location as L`
`On F.Area_Code = L.Area_Code`
`Join Product as P`
`On P.ProductId = F.ProductId`
`Where P.Product_Type = 'Coffee'`

| | Budget_COGS | Budget_Margin | Budget_Sales | Area_Code | Area_Code | State | Market | Market_Size | ProductId | Product_Type |
|----|-------------|---------------|--------------|-----------|-----------|-----------|---------|--------------|-----------|--------------|
| 1 | 90 | 130 | 220 | 719 | 719 | Colorado | Central | Major Market | 1 | Coffee |
| 2 | 80 | 110 | 190 | 970 | 970 | Colorado | Central | Major Market | 2 | Coffee |
| 3 | 100 | 140 | 240 | 970 | 970 | Colorado | Central | Major Market | 3 | Coffee |
| 4 | 150 | 210 | 360 | 217 | 217 | Illinois | Central | Major Market | 2 | Coffee |
| 5 | 100 | 140 | 240 | 309 | 309 | Illinois | Central | Major Market | 3 | Coffee |
| 6 | 10 | 30 | 40 | 319 | 319 | Iowa | Central | Small Market | 1 | Coffee |
| 7 | 20 | 40 | 60 | 641 | 641 | Iowa | Central | Small Market | 2 | Coffee |
| 8 | 20 | 30 | 50 | 712 | 712 | Iowa | Central | Small Market | 3 | Coffee |
| 9 | 90 | 100 | 190 | 636 | 636 | Missouri | Central | Small Market | 2 | Coffee |
| 10 | 80 | 90 | 170 | 573 | 573 | Missouri | Central | Small Market | 3 | Coffee |
| 11 | 60 | 80 | 140 | 740 | 740 | Ohio | Central | Major Market | 1 | Coffee |
| 12 | 60 | 80 | 140 | 614 | 614 | Ohio | Central | Major Market | 2 | Coffee |
| 13 | 60 | 90 | 150 | 614 | 614 | Ohio | Central | Major Market | 3 | Coffee |
| 14 | 50 | 70 | 120 | 262 | 262 | Wiscon... | Central | Small Market | 1 | Coffee |
| 15 | 80 | 110 | 190 | 262 | 262 | Wiscon... | Central | Small Market | 2 | Coffee |
| 16 | 80 | 100 | 180 | 414 | 414 | Wiscon... | Central | Small Market | 3 | Coffee |
| 17 | 110 | 180 | 290 | 475 | 475 | Conne... | East | Small Market | 2 | Coffee |
| 18 | 80 | 110 | 190 | 754 | 754 | Florida | East | Major Market | 3 | Coffee |
| 19 | 80 | 110 | 190 | 954 | 954 | Florida | East | Major Market | 2 | Coffee |
| 20 | 60 | 400 | 460 | 413 | 413 | Massa... | East | Major Market | 2 | Coffee |
| 21 | 20 | 50 | 80 | 602 | 602 | New H... | East | Small Market | 1 | Coffee |

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

- `Select * From Fact as F`
`Join Location as L`
`On F.Area_Code = L.Area_Code`
`Join Product as P`
`On P.ProductId = F.ProductId`
`Where F.Total_Expenses > 40`

| | Date | ProductId | Profit | Sales | Margin | COGS | Total_Expenses | Marketing | Inventory | Budget_Profit | Budget_COGS | Budget_ |
|----|------------|-----------|--------|-------|--------|------|----------------|-----------|-----------|---------------|-------------|---------|
| 1 | 2010-01-01 | 6 | 53 | 180 | 108 | 72 | 55 | 23 | 558 | 80 | 80 | 130 |
| 2 | 2010-01-01 | 8 | 99 | 341 | 171 | 170 | 72 | 47 | 1091 | 110 | 140 | 160 |
| 3 | 2010-01-01 | 9 | 0 | 150 | 87 | 63 | 87 | 57 | 435 | 20 | 50 | 80 |
| 4 | 2010-01-01 | 10 | 33 | 140 | 80 | 60 | 47 | 19 | 336 | 40 | 50 | 70 |
| 5 | 2010-01-01 | 11 | 17 | 130 | 72 | 58 | 55 | 22 | 338 | 20 | 40 | 70 |
| 6 | 2010-01-01 | 2 | 111 | 345 | 201 | 144 | 90 | 47 | 862 | 130 | 150 | 210 |
| 7 | 2010-01-01 | 3 | 87 | 234 | 139 | 95 | 52 | 30 | 608 | 100 | 100 | 140 |
| 8 | 2010-01-01 | 5 | 203 | 546 | 312 | 234 | 109 | 77 | 1310 | 260 | 270 | 370 |
| 9 | 2010-01-01 | 6 | 140 | 456 | 228 | 228 | 88 | 63 | 1459 | 180 | 260 | 270 |
| 10 | 2010-01-01 | 12 | 54 | 180 | 108 | 72 | 54 | 23 | 558 | 40 | 60 | 90 |
| 11 | 2010-01-01 | 8 | 202 | 546 | 312 | 234 | 110 | 77 | 1310 | 200 | 200 | 280 |
| 12 | 2010-01-01 | 9 | 86 | 234 | 139 | 95 | 53 | 30 | 608 | 90 | 80 | 120 |
| 13 | 2010-01-01 | 11 | 141 | 456 | 228 | 228 | 87 | 63 | 1459 | 140 | 190 | 210 |
| 14 | 2010-01-01 | 12 | 175 | 546 | 301 | 245 | 126 | 93 | 1419 | 160 | 210 | 270 |
| 15 | 2010-01-01 | 2 | 39 | 190 | 105 | 85 | 66 | 32 | 494 | 40 | 90 | 100 |
| 16 | 2010-01-01 | 3 | 47 | 170 | 92 | 78 | 45 | 24 | 965 | 50 | 80 | 90 |
| 17 | 2010-01-01 | 13 | -4 | 76 | 42 | 34 | 46 | 12 | 197 | 0 | 20 | 40 |
| 18 | 2010-01-01 | 8 | -11 | 90 | 53 | 37 | 64 | 34 | 261 | 0 | 30 | 40 |
| 19 | 2010-01-01 | 9 | -39 | 99 | 11 | 88 | 50 | 27 | 525 | -20 | 70 | 10 |
| 20 | 2010-01-01 | 12 | -9 | 65 | 36 | 29 | 45 | 11 | 169 | -10 | 20 | 30 |
| 21 | 2010-01-01 | 1 | 34 | 140 | 80 | 60 | 46 | 19 | 336 | 50 | 60 | 80 |
| 22 | 2010-01-01 | 2 | 42 | 140 | 83 | 57 | 41 | 18 | 364 | 50 | 60 | 80 |

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

- `Select Avg(Sales) As Average_Sales_in_area_code_719 From Fact Where Area_Code = 719`

| | Average_Sales_in_area_code_719 |
|---|--------------------------------|
| 1 | 186 |

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

- `Select Sum(Profit) As Colorado_State_Profit From Fact Where Area_Code in (Select Area_Code From Location Where State = 'Colorado')`

| | Average_Sales_in_area_code_719 |
|---|--------------------------------|
| 1 | 186 |

10. Display the average inventory for each product ID.

- `Select ProductId, AVG(Inventory) As Average_Inventory From fact Group By ProductId`

| Results Messages | | |
|------------------|-----------|-------------------|
| | ProductId | Average_Inventory |
| 1 | 9 | 718 |
| 2 | 3 | 838 |
| 3 | 12 | 757 |
| 4 | 6 | 755 |
| 5 | 7 | 879 |
| 6 | 1 | 741 |
| 7 | 10 | 1095 |
| 8 | 4 | 696 |
| 9 | 13 | 900 |
| 10 | 5 | 756 |
| 11 | 2 | 816 |
| 12 | 11 | 737 |
| 13 | 8 | 712 |

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

- `Select Distinct(State) From Location Order By State Asc`

| | 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 |

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

- `Select ProductId, Avg(Budget_Sales) As Avg_BS, Avg(Budget_Margin) As Avg_BM From Fact
Group By ProductId
Having Avg(Budget_Margin) > 100`

| | ProductId | Avg_BS | Avg_BM |
|---|-----------|--------|--------|
| 1 | 12 | 176 | 103 |
| 2 | 6 | 185 | 107 |
| 3 | 7 | 314 | 182 |
| 4 | 2 | 279 | 173 |

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

- `Select Sum(Sales) As Total_Sales_On_2010_01_01 From Fact
Where Date = '2010-01-01'`

| | Total_Sales_On_2010_01_01 |
|---|---------------------------|
| 1 | 31555 |

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

- `Select ProductId, Date, AVG(Total_Expenses) as Average_Total_Expense
From Fact
Group By Date, ProductId
Order by Date, ProductId`

| | ProductId | Date | Average_Total_Expense |
|----|-----------|------------|-----------------------|
| 1 | 1 | 2010-01-01 | 46 |
| 2 | 2 | 2010-01-01 | 63 |
| 3 | 3 | 2010-01-01 | 50 |
| 4 | 4 | 2010-01-01 | 50 |
| 5 | 5 | 2010-01-01 | 60 |
| 6 | 6 | 2010-01-01 | 46 |
| 7 | 7 | 2010-01-01 | 67 |
| 8 | 8 | 2010-01-01 | 50 |
| 9 | 9 | 2010-01-01 | 56 |
| 10 | 10 | 2010-01-01 | 50 |
| 11 | 11 | 2010-01-01 | 43 |
| 12 | 12 | 2010-01-01 | 58 |
| 13 | 13 | 2010-01-01 | 45 |
| 14 | 1 | 2010-01-02 | 46 |
| 15 | 2 | 2010-01-02 | 63 |
| 16 | 3 | 2010-01-02 | 50 |
| 17 | 4 | 2010-01-02 | 49 |
| 18 | 5 | 2010-01-02 | 62 |
| 19 | 6 | 2010-01-02 | 46 |
| 20 | 7 | 2010-01-02 | 67 |
| 21 | 8 | 2010-01-02 | 52 |
| 22 | 9 | 2010-01-02 | 56 |
| 23 | 10 | 2010-01-02 | 50 |
| 24 | 11 | 2010-01-02 | 44 |

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

- Select F.Date, F.ProductId, P.Product_Type, P.Product, F.Sales, F.Profit,
 L.State, F.Area_Code
 From Fact As F
 Join Location As L
 On L.Area_Code = F.Area_Code
 Join Product As P
 On P.ProductId = F.ProductId

| | Date | ProductId | Product_Type | Product | Sales | Profit | State | Area_Code |
|----|------------|-----------|--------------|-------------------|-------|--------|----------|-----------|
| 1 | 2010-01-01 | 1 | Coffee | Amaretto | 219 | 94 | Colorado | 719 |
| 2 | 2010-01-01 | 2 | Coffee | Columbian | 190 | 68 | Colorado | 970 |
| 3 | 2010-01-01 | 3 | Coffee | Decaf Irish Cream | 234 | 101 | Colorado | 970 |
| 4 | 2010-01-01 | 13 | Tea | Green Tea | 100 | 30 | Colorado | 303 |
| 5 | 2010-01-01 | 5 | Espresso | Caffe Mocha | 134 | 54 | Colorado | 303 |
| 6 | 2010-01-01 | 6 | Espresso | Decaf Espresso | 180 | 53 | Colorado | 720 |
| 7 | 2010-01-01 | 8 | Herbal Tea | Chamomile | 341 | 99 | Colorado | 970 |
| 8 | 2010-01-01 | 9 | Herbal Tea | Lemon | 150 | 0 | Colorado | 719 |
| 9 | 2010-01-01 | 10 | Herbal Tea | Mint | 140 | 33 | Colorado | 970 |
| 10 | 2010-01-01 | 11 | Tea | Darjeeling | 130 | 17 | Colorado | 719 |
| 11 | 2010-01-01 | 12 | Tea | Earl Grey | 140 | 36 | Colorado | 303 |
| 12 | 2010-01-01 | 2 | Coffee | Columbian | 345 | 111 | Illinois | 217 |
| 13 | 2010-01-01 | 3 | Coffee | Decaf Irish Cream | 234 | 87 | Illinois | 309 |
| 14 | 2010-01-01 | 5 | Espresso | Caffe Mocha | 546 | 203 | Illinois | 309 |
| 15 | 2010-01-01 | 6 | Espresso | Decaf Espresso | 456 | 140 | Illinois | 630 |
| 16 | 2010-01-01 | 8 | Herbal Tea | Chamomile | 219 | 95 | Illinois | 312 |
| 17 | 2010-01-01 | 9 | Herbal Tea | Lemon | 190 | 68 | Illinois | 630 |
| 18 | 2010-01-01 | 10 | Herbal Tea | Mint | 234 | 101 | Illinois | 773 |
| 19 | 2010-01-01 | 11 | Tea | Darjeeling | 134 | 53 | Illinois | 217 |
| 20 | 2010-01-01 | 12 | Tea | Earl Grey | 180 | 54 | Illinois | 708 |
| 21 | 2010-01-01 | 1 | Coffee | Amaretto | 45 | 11 | Iowa | 319 |
| 22 | 2010-01-01 | 2 | Coffee | Columbian | 62 | 5 | Iowa | 641 |
| 23 | 2010-01-01 | 3 | Coffee | Decaf Irish Cream | 54 | 12 | Iowa | 712 |
| 24 | 2010-01-01 | 5 | Espresso | Caffe Mocha | 43 | 11 | Iowa | 563 |

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

- Select ProductId, Sum(Sales) As Sales,
 ROW_NUMBER()over(Order By Sum(Sales) Desc) as Rank
 From fact
 Group By ProductId

| | ProductId | Sales | Rank |
|----|-----------|--------|------|
| 1 | 2 | 128311 | 1 |
| 2 | 9 | 95926 | 2 |
| 3 | 5 | 84904 | 3 |
| 4 | 6 | 78162 | 4 |
| 5 | 8 | 75578 | 5 |
| 6 | 11 | 73151 | 6 |
| 7 | 12 | 66772 | 7 |
| 8 | 3 | 62248 | 8 |
| 9 | 4 | 35899 | 9 |
| 10 | 10 | 35710 | 10 |
| 11 | 13 | 32850 | 11 |
| 12 | 1 | 26269 | 12 |
| 13 | 7 | 24031 | 13 |

17. Find the state wise profit and sales.

- `Select L.State,Sum(F.Sales) As Sales,Sum(F.Profit) As Profit From Fact As F`
`Join Location As L`
`On L.Area_Code = F.Area_Code`
`Group by L.State`
`Order By L.State`

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

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

- `Select L.State,P.Product,Sum(F.Sales) As Sales,Sum(F.Profit) As Profit From Fact As F`
`Join Location As L`
`On L.Area_Code = F.Area_Code`
`Join Product As P`
`On P.ProductId = F.ProductId`
`Group by L.State,P.Product`
`Order By L.State`

| | State | Product | Sales | Profit |
|----|------------|-------------------|--------|--------|
| 1 | California | Caffe Mocha | 7691 | 886 |
| 2 | California | Decaf Irish Cream | 3739 | -3891 |
| 3 | California | Green Tea | 4027 | 1355 |
| 4 | California | Mint | 3807 | 1555 |
| 5 | California | Lemon | 126... | 5450 |
| 6 | California | Darjeeling | 6507 | 3418 |
| 7 | California | Decaf Espresso | 146... | 6580 |
| 8 | California | Caffe Latte | 120... | 4497 |
| 9 | California | Earl Grey | 4640 | 2334 |
| 10 | California | Columbian | 182... | 8566 |
| 11 | California | Chamomile | 6233 | 3252 |
| 12 | California | Amaretto | 2714 | -2217 |
| 13 | Colorado | Chamomile | 7798 | 2678 |
| 14 | Colorado | Lemon | 2935 | -141 |
| 15 | Colorado | Decaf Espresso | 4027 | 1362 |
| 16 | Colorado | Columbian | 3807 | 1566 |
| 17 | Colorado | Amaretto | 6507 | 3410 |
| 18 | Colorado | Caffe Mocha | 4640 | 2339 |
| 19 | Colorado | Darjeeling | 2902 | 366 |
| 20 | Colorado | Mint | 3102 | 815 |
| 21 | Colorado | Decaf Irish Cream | 6233 | 3250 |

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

- **Select** ProductId,Sales **as** Actual_Sales, Sales*1.05 **as** Increased_Sales
From Fact

| | ProductId | Actual_Sales | Increased_Sales |
|----|-----------|--------------|-----------------|
| 1 | 1 | 219 | 229.95 |
| 2 | 2 | 190 | 199.50 |
| 3 | 3 | 234 | 245.70 |
| 4 | 13 | 100 | 105.00 |
| 5 | 5 | 134 | 140.70 |
| 6 | 6 | 180 | 189.00 |
| 7 | 8 | 341 | 358.05 |
| 8 | 9 | 150 | 157.50 |
| 9 | 10 | 140 | 147.00 |
| 10 | 11 | 130 | 136.50 |
| 11 | 12 | 140 | 147.00 |
| 12 | 2 | 345 | 362.25 |
| 13 | 3 | 234 | 245.70 |
| 14 | 5 | 546 | 573.30 |
| 15 | 6 | 456 | 478.80 |
| 16 | 8 | 219 | 229.95 |
| 17 | 9 | 190 | 199.50 |
| 18 | 10 | 234 | 245.70 |
| 19 | 11 | 134 | 140.70 |
| 20 | 12 | 180 | 189.00 |
| 21 | 1 | 45 | 47.25 |
| 22 | 2 | 62 | 65.10 |
| 23 | 3 | 54 | 56.70 |
| 24 | 5 | 43 | 45.15 |

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

- `Select P.Product_Type, F.ProductId, Max(F.Profit) As Max_Profit From Fact As F
Join Product As P
On F.ProductId = P.ProductId
Group By P.Product_Type,F.ProductId`

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

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

- `Create Proc sp_ProductDetails(@Product_Type varchar(50))
As Begin
Select * From Fact As F
Join Location As L
On L.Area_Code = F.Area_Code
Join Product As P
On P.ProductId = F.ProductId
Where P.Product_Type = @Product_Type
End

Exec sp_ProductDetails 'Coffee'`

| | Budget_COGS | Budget_Margin | Budget_Sales | Area_Code | Area_Code | State | Market | Market_Size | ProductId | Product_Type |
|----|-------------|---------------|--------------|-----------|-----------|-----------|---------|--------------|-----------|--------------|
| 1 | 90 | 130 | 220 | 719 | 719 | Colorado | Central | Major Market | 1 | Coffee |
| 2 | 80 | 110 | 190 | 970 | 970 | Colorado | Central | Major Market | 2 | Coffee |
| 3 | 100 | 140 | 240 | 970 | 970 | Colorado | Central | Major Market | 3 | Coffee |
| 4 | 150 | 210 | 360 | 217 | 217 | Illinois | Central | Major Market | 2 | Coffee |
| 5 | 100 | 140 | 240 | 309 | 309 | Illinois | Central | Major Market | 3 | Coffee |
| 6 | 10 | 30 | 40 | 319 | 319 | Iowa | Central | Small Market | 1 | Coffee |
| 7 | 20 | 40 | 60 | 641 | 641 | Iowa | Central | Small Market | 2 | Coffee |
| 8 | 20 | 30 | 50 | 712 | 712 | Iowa | Central | Small Market | 3 | Coffee |
| 9 | 90 | 100 | 190 | 636 | 636 | Missouri | Central | Small Market | 2 | Coffee |
| 10 | 80 | 90 | 170 | 573 | 573 | Missouri | Central | Small Market | 3 | Coffee |
| 11 | 60 | 80 | 140 | 740 | 740 | Ohio | Central | Major Market | 1 | Coffee |
| 12 | 60 | 80 | 140 | 614 | 614 | Ohio | Central | Major Market | 2 | Coffee |
| 13 | 60 | 90 | 150 | 614 | 614 | Ohio | Central | Major Market | 3 | Coffee |
| 14 | 50 | 70 | 120 | 262 | 262 | Wiscon... | Central | Small Market | 1 | Coffee |
| 15 | 80 | 110 | 190 | 262 | 262 | Wiscon... | Central | Small Market | 2 | Coffee |
| 16 | 80 | 100 | 180 | 414 | 414 | Wiscon... | Central | Small Market | 3 | Coffee |
| 17 | 110 | 180 | 290 | 475 | 475 | Conne... | East | Small Market | 2 | Coffee |
| 18 | 80 | 110 | 190 | 754 | 754 | Florida | East | Major Market | 3 | Coffee |
| 19 | 80 | 110 | 190 | 954 | 954 | Florida | East | Major Market | 2 | Coffee |
| 20 | 60 | 400 | 460 | 413 | 413 | Massa... | East | Major Market | 2 | Coffee |
| 21 | 20 | 50 | 70 | 622 | 622 | Massa... | East | Small Market | 1 | 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** ProductId,Total_Expenses,
IF(Total_Expenses<60,'Profit','Loss') **As** [Profit / Loss]
From Fact

| | ProductId | Total_Expenses | Profit / Loss |
|----|-----------|----------------|---------------|
| 1 | 1 | 36 | Profit |
| 2 | 2 | 39 | Profit |
| 3 | 3 | 38 | Profit |
| 4 | 13 | 26 | Profit |
| 5 | 5 | 26 | Profit |
| 6 | 6 | 55 | Profit |
| 7 | 8 | 72 | Loss |
| 8 | 9 | 87 | Loss |
| 9 | 10 | 47 | Profit |
| 10 | 11 | 55 | Profit |
| 11 | 12 | 40 | Profit |
| 12 | 2 | 90 | Loss |
| 13 | 3 | 52 | Profit |
| 14 | 5 | 109 | Loss |
| 15 | 6 | 88 | Loss |
| 16 | 8 | 35 | Profit |
| 17 | 9 | 39 | Profit |
| 18 | 10 | 38 | Profit |
| 19 | 11 | 27 | Profit |
| 20 | 12 | 54 | Profit |
| 21 | 1 | 16 | Profit |
| 22 | 2 | 29 | 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.

- `Select Date,ProductId,DATEPART(WEEK,Date) As Weeks,
sum(Sales) As [Sum Of Sales] From fact
Group By
Rollup(Date,ProductId)`

| | Date | ProductId | Weeks | Sum Of Sales |
|----|------------|-----------|-------|--------------|
| 1 | 2010-01-01 | 1 | 1 | 1082 |
| 2 | 2010-01-01 | 2 | 1 | 4860 |
| 3 | 2010-01-01 | 3 | 1 | 2372 |
| 4 | 2010-01-01 | 4 | 1 | 1478 |
| 5 | 2010-01-01 | 5 | 1 | 3220 |
| 6 | 2010-01-01 | 6 | 1 | 3122 |
| 7 | 2010-01-01 | 7 | 1 | 896 |
| 8 | 2010-01-01 | 8 | 1 | 2960 |
| 9 | 2010-01-01 | 9 | 1 | 3517 |
| 10 | 2010-01-01 | 10 | 1 | 1397 |
| 11 | 2010-01-01 | 11 | 1 | 2839 |
| 12 | 2010-01-01 | 12 | 1 | 2562 |
| 13 | 2010-01-01 | 13 | 1 | 1250 |
| 14 | 2010-01-01 | NULL | 1 | 31555 |
| 15 | 2010-01-02 | 1 | 1 | 1073 |
| 16 | 2010-01-02 | 2 | 1 | 4821 |
| 17 | 2010-01-02 | 3 | 1 | 2433 |
| 18 | 2010-01-02 | 4 | 1 | 1463 |
| 19 | 2010-01-02 | 5 | 1 | 3348 |
| 20 | 2010-01-02 | 6 | 1 | 3161 |
| 21 | 2010-01-02 | 7 | 1 | 988 |
| 22 | 2010-01-02 | 8 | 1 | 3016 |

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 Count(Area_Code) As Count From
(Select Area_Code From Fact
Union
Select Area_Code From Location) As [Combine Data]`

| | Count |
|---|-------|
| 1 | 156 |

- `Select Area_Code From Fact
Union All
Select Area_Code From Location`
 - `Select Count(Area_Code) As Count From
(Select Area_Code From Fact
Union All
Select Area_Code From Location)`

Select Area_Code From Location) As [Combine Data]

| | Count |
|---|-------|
| 1 | 4404 |

- Select Area_Code From Fact
Intersect
Select Area_Code From Location
 - Select Count(Area_Code) As Count From
(Select Area_Code From Fact
Intersect
Select Area_Code From Location) As [Combine Data]

| | Count |
|---|-------|
| 1 | 156 |

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

- Create Function fn_Product(@Type Varchar(20))
Returns Table
As
Return (Select * From Product Where Type = @Type)

Select * From fn_Product('Regular')

| | ProductId | Product_Type | Product | Type |
|---|-----------|--------------|------------------|---------|
| 1 | 1 | Coffee | Amaretto | Regular |
| 2 | 2 | Coffee | Columbian | Regular |
| 3 | 4 | Espresso | Caffe Latte | Regular |
| 4 | 5 | Espresso | Caffe Mocha | Regular |
| 5 | 7 | Espresso | Regular Espresso | Regular |
| 6 | 11 | Tea | Darjeeling | Regular |
| 7 | 12 | Tea | Earl Grey | Regular |
| 8 | 13 | Tea | Green Tea | Regular |

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

- Begin Transaction

Update Product Set Product_Type = 'Tea'
Where Product_Type = 'Coffee'
Select * From Product

Rollback

| | ProductId | Product_Type | Product | Type |
|----|-----------|--------------|-------------------|---------|
| 1 | 1 | Tea | Amaretto | Regular |
| 2 | 2 | Tea | Columbian | Regular |
| 3 | 3 | Tea | 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.

- **Select** Date,ProductId,Sales **From** Fact
Where Total_Expenses **Between** 100 **And** 200

| | Date | ProductId | Sales |
|----|------------|-----------|-------|
| 1 | 2010-01-01 | 5 | 546 |
| 2 | 2010-01-01 | 8 | 546 |
| 3 | 2010-01-01 | 12 | 546 |
| 4 | 2010-01-01 | 2 | 190 |
| 5 | 2010-01-01 | 5 | 190 |
| 6 | 2010-01-01 | 5 | 61 |
| 7 | 2010-01-01 | 2 | 678 |
| 8 | 2010-01-01 | 9 | 483 |
| 9 | 2010-01-01 | 9 | 190 |
| 10 | 2010-01-01 | 5 | 250 |
| 11 | 2010-01-01 | 6 | 546 |
| 12 | 2010-01-01 | 2 | 678 |
| 13 | 2010-01-01 | 8 | 250 |
| 14 | 2010-01-01 | 11 | 546 |
| 15 | 2010-01-01 | 13 | 17 |
| 16 | 2010-01-01 | 3 | 190 |
| 17 | 2010-01-02 | 5 | 545 |
| 18 | 2010-01-02 | 8 | 545 |
| 19 | 2010-01-02 | 12 | 534 |
| 20 | 2010-01-02 | 2 | 220 |
| 21 | 2010-01-02 | 5 | 220 |
| 22 | 2010-01-02 | 5 | 61 |
| 23 | 2010-01-02 | 2 | 645 |
| 24 | 2010-01-02 | 9 | 495 |

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

- [Begin Transaction](#)

Delete Product Where Type = 'Regular'
Select * From Product

[Rollback](#)

| | 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 columnProduct.

- [Select ASCII\(Substring\(Product,5,1\)\) \[ASCII Value\] From Product](#)

| | ASCII Value |
|----|-------------|
| 1 | 101 |
| 2 | 109 |
| 3 | 102 |
| 4 | 101 |
| 5 | 101 |
| 6 | 102 |
| 7 | 108 |
| 8 | 111 |
| 9 | 110 |
| 10 | NULL |
| 11 | 101 |
| 12 | 32 |
| 13 | 110 |

SQL Case Study – 2

Create the following table:

CREATE TABLE LOCATION (

Location_ID INT PRIMARY KEY,

City VARCHAR(50)

);

INSERT INTO LOCATION (Location_ID, City)

VALUES (122, 'New York'),

(123, 'Dallas'),

(124, 'Chicago'),

(167, 'Boston');

CREATE TABLE DEPARTMENT (

Department_Id INT PRIMARY KEY,

Name VARCHAR(50),

Location_Id INT,

FOREIGN KEY (Location_Id) REFERENCES LOCATION(Location_ID)

);

INSERT INTO DEPARTMENT (Department_Id, Name, Location_Id)

VALUES (10, 'Accounting', 122),

(20, 'Sales', 124),

(30, 'Research', 123),

(40, 'Operations', 167);

Select * From DEPARTMENT

```
CREATE TABLE JOB
(JOB_ID INT PRIMARY KEY,
DESIGNATION VARCHAR(20))
```

```
INSERT INTO JOB VALUES
(667, 'CLERK'),
(668, 'STAFF'),
(669, 'ANALYST'),
(670, 'SALES_PERSON'),
(671, 'MANAGER'),
(672, 'PRESIDENT')
```

```
CREATE TABLE EMPLOYEE
(EMPLOYEE_ID INT,
LAST_NAME VARCHAR(20),
FIRST_NAME VARCHAR(20),
MIDDLE_NAME CHAR(1),
JOB_ID INT FOREIGN KEY
REFERENCES JOB(JOB_ID),
MANAGER_ID INT,
HIRE_DATE DATE,
SALARY INT,
COMM INT,
DEPARTMENT_ID INT FOREIGN KEY
REFERENCES DEPARTMENT(DEPARTMENT_ID))
```

```
INSERT INTO EMPLOYEE VALUES
(7369, 'SMITH', 'JOHN', 'Q', 667, 7902, '17-DEC-84', 800, NULL, 20),
(7499, 'ALLEN', 'KEVIN', 'J', 670, 7698, '20-FEB-84', 1600, 300, 30),
(7505, 'DOYLE', 'JEAN', 'K', 671, 7839, '04-APR-85', 2850, NULL, 30),
(7506, 'DENNIS', 'LYNN', 'S', 671, 7839, '15-MAY-85', 2750, NULL, 30),
(7507, 'BAKER', 'LESLIE', 'D', 671, 7839, '10-JUN-85', 2200, NULL, 40),
(7521, 'WARK', 'CYNTHIA', 'D', 670, 7698, '22-FEB-85', 1250, 500, 30)
```

Simple Queries:

1. List all the employee details.

- `Select * From EMPLOYEE`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |
| 2 | 7499 | ALLEN | KEVIN | J | 670 | 7698 | 1984-02-20 | 1600 | 300 | 30 |
| 3 | 7505 | DOYLE | JEAN | K | 671 | 7839 | 1985-04-04 | 2850 | NULL | 30 |
| 4 | 7506 | DENNIS | LYNN | S | 671 | 7839 | 1985-05-15 | 2750 | NULL | 30 |
| 5 | 7507 | BAKER | LESLIE | D | 671 | 7839 | 1985-06-10 | 2200 | NULL | 40 |
| 6 | 7521 | WARK | CYNTHIA | D | 670 | 7698 | 1985-02-22 | 1250 | 500 | 30 |

2. List all the department details.

- `Select * From DEPARTMENT`

| | Department_Id | Name | Location_Id |
|---|---------------|------------|-------------|
| 1 | 10 | Accounting | 122 |
| 2 | 20 | Sales | 124 |
| 3 | 30 | Research | 123 |
| 4 | 40 | Operations | 167 |

3. List all job details.

- `Select * From JOB`

| | JOB_ID | DESIGNATION |
|---|--------|--------------|
| 1 | 667 | CLERK |
| 2 | 668 | STAFF |
| 3 | 669 | ANALYST |
| 4 | 670 | SALES_PERSON |
| 5 | 671 | MANAGER |
| 6 | 672 | PRESIDENT |

4. List all the locations.

- `Select * From LOCATION`

| | Location_ID | City |
|---|-------------|----------|
| 1 | 122 | New York |
| 2 | 123 | Dallas |
| 3 | 124 | Chicago |
| 4 | 167 | Boston |

5. List out the First Name, Last Name, Salary, Commission for all Employees.

- `Select FIRST_NAME, LAST_NAME, SALARY, COMM From EMPLOYEE`

| | FIRST_NAME | LAST_NAME | SALARY | COMM |
|---|------------|-----------|--------|------|
| 1 | JOHN | SMITH | 800 | NULL |
| 2 | KEVIN | ALLEN | 1600 | 300 |
| 3 | JEAN | DOYLE | 2850 | NULL |
| 4 | LYNN | DENNIS | 2750 | NULL |
| 5 | LESLIE | BAKER | 2200 | NULL |
| 6 | CYNTHIA | WARK | 1250 | 500 |

6. List out the Employee ID, Last Name, Department ID for all employees and alias

Employee ID as "ID of the Employee", Last Name as "Name of the Employee", Department ID as "Dep_id".

- `Select EMPLOYEE_ID As [ID of the Employee], LAST_NAME As [Name of the Employee], DEPARTMENT_ID As Dep_id From EMPLOYEE`

| | ID of the Employee | Name of the Employee | Dep_id |
|---|--------------------|----------------------|--------|
| 1 | 7369 | SMITH | 20 |
| 2 | 7499 | ALLEN | 30 |
| 3 | 7505 | DOYLE | 30 |
| 4 | 7506 | DENNIS | 30 |
| 5 | 7507 | BAKER | 40 |
| 6 | 7521 | WARK | 30 |

7. List out the annual salary of the employees with their names only.

- `Select LAST_NAME+' '+FIRST_NAME As [Employees Name], SALARY * 12 As [Annual Salary] From EMPLOYEE`

| | Employees Name | Annual Salary |
|---|----------------|---------------|
| 1 | SMITH JOHN | 9600 |
| 2 | ALLEN KEVIN | 19200 |
| 3 | DOYLE JEAN | 34200 |
| 4 | DENNIS LYNN | 33000 |
| 5 | BAKER LESLIE | 26400 |
| 6 | WARK CYNTHIA | 15000 |

WHERE Condition:

1. List the details about "Smith".

- `Select * From EMPLOYEE Where LAST_NAME = 'Smith'`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |

2. List out the employees who are working in department 20.

- `Select * From EMPLOYEE`
`Where DEPARTMENT_ID = 20`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |

3. List out the employees who are earning salaries between 3000 and 4500.

- `Select * From EMPLOYEE`
`Where SALARY Between 3000 And 4500`

4. List out the employees who are working in department 10 or 20.

- `Select * From EMPLOYEE`
`Where DEPARTMENT_ID = 10 Or DEPARTMENT_ID = 20`
- `Select * From EMPLOYEE`
`Where DEPARTMENT_ID In (10,20)`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |

5. Find out the employees who are not working in department 10 or 30.

- `Select * From EMPLOYEE`
`Where DEPARTMENT_ID Not In (10,30)`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |
| 2 | 7507 | BAKER | LESLIE | D | 671 | 7839 | 1985-06-10 | 2200 | NULL | 40 |

6. List out the employees whose name starts with 'S'.

- `Select * From EMPLOYEE`
`Where FIRST_NAME Like 'S%'`
- `Select * From EMPLOYEE`
`Where LAST_NAME Like 'S%'`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |

7. List out the employees whose name starts with 'S' and ends with 'H'.

- `Select * From EMPLOYEE`
`Where FIRST_NAME Like 'S%' And LAST_NAME Like '%H'`
- `Select * From EMPLOYEE`
`Where LAST_NAME Like 'S%' And LAST_NAME Like '%H'`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |

8. List out the employees whose name length is 4 and start with 'S'.

- `Select * From EMPLOYEE`
`len(FIRST_NAME) = 4 And FIRST_NAME Like 'S%'`

9. List out employees who are working in department 10 and draw salaries more than 3500.

- `Select * From EMPLOYEE`
`Where DEPARTMENT_ID = 10 And SALARY > 3500`

10. List out the employees who are not receiving commission.

- `Select * From EMPLOYEE`
`Where COMM Is Null`

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |
| 2 | 7505 | DOYLE | JEAN | K | 671 | 7839 | 1985-04-04 | 2850 | NULL | 30 |
| 3 | 7506 | DENNIS | LYNN | S | 671 | 7839 | 1985-05-15 | 2750 | NULL | 30 |
| 4 | 7507 | BAKER | LESLIE | D | 671 | 7839 | 1985-06-10 | 2200 | NULL | 40 |

ORDER BY Clause:

1. List out the Employee ID and Last Name in ascending order based on the Employee ID.

- `Select EMPLOYEE_ID, LAST_NAME`
`From EMPLOYEE`
`Order By EMPLOYEE_ID Asc`

| | EMPLOYEE_ID | LAST_NAME |
|---|-------------|-----------|
| 1 | 7369 | SMITH |
| 2 | 7499 | ALLEN |
| 3 | 7505 | DOYLE |
| 4 | 7506 | DENNIS |
| 5 | 7507 | BAKER |
| 6 | 7521 | WARK |

2. List out the Employee ID and Name in descending order based on salary.

- **Select** EMPLOYEE_ID, FIRST_NAME
From EMPLOYEE
Order By SALARY

| | EMPLOYEE_ID | FIRST_NAME |
|---|-------------|------------|
| 1 | 7369 | JOHN |
| 2 | 7521 | CYNTHIA |
| 3 | 7499 | KEVIN |
| 4 | 7507 | LESLIE |
| 5 | 7506 | LYNN |
| 6 | 7505 | JEAN |

3. List out the employee details according to their Last Name in ascending-order.

- **Select** * **From** EMPLOYEE
Order By LAST_NAME Asc

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7499 | ALLEN | KEVIN | J | 670 | 7698 | 1984-02-20 | 1600 | 300 | 30 |
| 2 | 7507 | BAKER | LESLIE | D | 671 | 7839 | 1985-06-10 | 2200 | NULL | 40 |
| 3 | 7506 | DENNIS | LYNN | S | 671 | 7839 | 1985-05-15 | 2750 | NULL | 30 |
| 4 | 7505 | DOYLE | JEAN | K | 671 | 7839 | 1985-04-04 | 2850 | NULL | 30 |
| 5 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |
| 6 | 7521 | WARK | CYNTHIA | D | 670 | 7698 | 1985-02-22 | 1250 | 500 | 30 |

4. List out the employee details according to their Last Name in ascending order and then Department ID in descending order.

- **Select** * **From** EMPLOYEE
Order By LAST_NAME Asc, DEPARTMENT_ID Desc

| | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7499 | ALLEN | KEVIN | J | 670 | 7698 | 1984-02-20 | 1600 | 300 | 30 |
| 2 | 7507 | BAKER | LESLIE | D | 671 | 7839 | 1985-06-10 | 2200 | NULL | 40 |
| 3 | 7506 | DENNIS | LYNN | S | 671 | 7839 | 1985-05-15 | 2750 | NULL | 30 |
| 4 | 7505 | DOYLE | JEAN | K | 671 | 7839 | 1985-04-04 | 2850 | NULL | 30 |
| 5 | 7369 | SMITH | JOHN | Q | 667 | 7902 | 1984-12-17 | 800 | NULL | 20 |
| 6 | 7521 | WARK | CYNTHIA | D | 670 | 7698 | 1985-02-22 | 1250 | 500 | 30 |

GROUP BY and HAVING Clause:

1. How many employees are in different departments in the organization?

- **Select** DEPARTMENT_ID, **Count**(EMPLOYEE_ID) [Count Of Employees]
From EMPLOYEE
Group by DEPARTMENT_ID

| | DEPARTMENT_ID | Count Of Employees |
|---|---------------|--------------------|
| 1 | 20 | 1 |
| 2 | 30 | 4 |
| 3 | 40 | 1 |

2. List out the department wise maximum salary, minimum salary and average salary of the employees.

- Select DEPARTMENT_ID, Max(SALARY) As [Maximum Salary],
 Min(SALARY) As [Minimum Salary], Avg(SALARY) As [Average Salary]
 From EMPLOYEE
 Group By DEPARTMENT_ID

| | DEPARTMENT_ID | Maximum Salary | Minimum Salary | Average Salary |
|---|---------------|----------------|----------------|----------------|
| 1 | 20 | 800 | 800 | 800 |
| 2 | 30 | 2850 | 1250 | 2112 |
| 3 | 40 | 2200 | 2200 | 2200 |

3. List out the job wise maximum salary, minimum salary and average salary of the employees.

- Select JOB_ID, Max(SALARY) As [Maximum Salary],
 Min(SALARY) As [Minimum Salary], Avg(SALARY) As [Average Salary]
 From EMPLOYEE
 Group By JOB_ID

| | JOB_ID | Maximum Salary | Minimum Salary | Average Salary |
|---|--------|----------------|----------------|----------------|
| 1 | 667 | 800 | 800 | 800 |
| 2 | 670 | 1600 | 1250 | 1425 |
| 3 | 671 | 2850 | 2200 | 2600 |

4. List out the number of employees who joined each month in ascending order.

- Select Month(HIRE_DATE) As Month, COUNT(EMPLOYEE_ID) As [Count Of Employee]
 From EMPLOYEE
 Group By MONTH(HIRE_DATE)
 Order By MONTH(HIRE_DATE) Asc

| | Month | Count Of Employee |
|---|-------|-------------------|
| 1 | 2 | 2 |
| 2 | 4 | 1 |
| 3 | 5 | 1 |
| 4 | 6 | 1 |
| 5 | 12 | 1 |

5. List out the number of employees for each month and year in ascending order based on the year and month.

- `Select Month(HIRE_DATE) As Month, Year(HIRE_DATE) As Year,
COUNT(EMPLOYEE_ID) As [Count Of Employee]
From EMPLOYEE
Group By MONTH(HIRE_DATE), YEAR(HIRE_DATE)
Order By YEAR(HIRE_DATE), MONTH(HIRE_DATE) Asc`

| | Month | Year | Count Of Employee |
|---|-------|------|-------------------|
| 1 | 2 | 1984 | 1 |
| 2 | 12 | 1984 | 1 |
| 3 | 2 | 1985 | 1 |
| 4 | 4 | 1985 | 1 |
| 5 | 5 | 1985 | 1 |
| 6 | 6 | 1985 | 1 |

6. List out the Department ID having at least four employees.

- `Select DEPARTMENT_ID
From EMPLOYEE
Group By DEPARTMENT_ID
Having Count(DEPARTMENT_ID) >= 4`

| | DEPARTMENT_ID |
|---|---------------|
| 1 | 30 |

7. How many employees joined in the month of January?

- `Select COUNT(DEPARTMENT_ID) As Count
From EMPLOYEE
Where Month(HIRE_DATE) = 1`

| | Count |
|---|-------|
| 1 | 0 |

8. How many employees joined in the month of January or September?

- `Select COUNT(DEPARTMENT_ID) As Count
From EMPLOYEE
Where Month(HIRE_DATE) = 1 Or DATENAME(MONTH, HIRE_DATE) = 'September'`

9. How many employees joined in 1985?

- `Select Count(*) As Count From EMPLOYEE
Where Year(HIRE_DATE) = 1985`

| | Count |
|---|-------|
| 1 | 4 |

10. How many employees joined each month in 1985?

- `Select DATENAME(MONTH,HIRE_DATE) Month ,Count(*) As Count From EMPLOYEE
Where Year(HIRE_DATE) = 1985
Group By DATENAME(MONTH,HIRE_DATE)`

| | Month | Count |
|---|----------|-------|
| 1 | April | 1 |
| 2 | February | 1 |
| 3 | June | 1 |
| 4 | May | 1 |

11. How many employees joined in March 1985?

- `Select DATENAME(MONTH,HIRE_DATE) Month ,Count(*) As Count From EMPLOYEE
Where Year(HIRE_DATE) = 1985 And DATENAME(MONTH,HIRE_DATE) = 'March'
Group By DATENAME(MONTH,HIRE_DATE)`

12. Which is the Department ID having greater than or equal to 3 employees joining in April 1985?

- `Select DEPARTMENT_ID From EMPLOYEE
Where DATENAME(Month,HIRE_DATE) = 'April' And Year(HIRE_DATE) = 1985
Group By DEPARTMENT_ID
Having Count(*) >=3`

Joins:

1. List out employees with their department names.

- `Select E.Employee_ID, E.First_Name+' '+E.Last_Name As [Employee Name],
D.Name As Department From EMPLOYEE As E
Join DEPARTMENT As D
On E.DEPARTMENT_ID = D.Department_Id`

| | Employee_ID | Employee Name | Department |
|---|-------------|---------------|------------|
| 1 | 7369 | JOHN SMITH | Sales |
| 2 | 7499 | KEVIN ALLEN | Research |
| 3 | 7505 | JEAN DOYLE | Research |
| 4 | 7506 | LYNN DENNIS | Research |
| 5 | 7507 | LESLIE BAKER | Operations |
| 6 | 7521 | CYNTHIA WARK | Research |

2. Display employees with their designations.

- **Select** E.Employee_ID, E.First_Name+' '+E.Last_Name **As** [Employee Name],
J.Designation **From** EMPLOYEE **As** E
Join Job **As** J
On E.JOB_ID = J.JOB_ID

| | Employee_ID | Employee Name | Designation |
|---|-------------|---------------|--------------|
| 1 | 7369 | JOHN SMITH | CLERK |
| 2 | 7499 | KEVIN ALLEN | SALES_PERSON |
| 3 | 7505 | JEAN DOYLE | MANAGER |
| 4 | 7506 | LYNN DENNIS | MANAGER |
| 5 | 7507 | LESLIE BAKER | MANAGER |
| 6 | 7521 | CYNTHIA WARK | SALES_PERSON |

3. Display the employees with their department names and regional groups.

- **Select** E.Employee_ID, E.First_Name+' '+E.Last_Name **As** [Employee Name],
D.Name **As** Department, L.City **From** EMPLOYEE **As** E
Join DEPARTMENT **As** D
On E.DEPARTMENT_ID = D.Department_Id
Join LOCATION **As** L
On D.Location_Id = L.Location_ID

| | Employee_ID | Employee Name | Department | City |
|---|-------------|---------------|------------|---------|
| 1 | 7369 | JOHN SMITH | Sales | Chicago |
| 2 | 7499 | KEVIN ALLEN | Research | Dallas |
| 3 | 7505 | JEAN DOYLE | Research | Dallas |
| 4 | 7506 | LYNN DENNIS | Research | Dallas |
| 5 | 7507 | LESLIE BAKER | Operations | Boston |
| 6 | 7521 | CYNTHIA WARK | Research | Dallas |

4. How many employees are working in different departments? Display with department names.

- **Select** count(E.Employee_ID) [Employees Count],
D.Name **As** Department **From** EMPLOYEE **As** E
Join DEPARTMENT **As** D
On E.DEPARTMENT_ID = D.Department_Id
Group By D.Name

| | Employees Count | Department |
|---|-----------------|------------|
| 1 | 1 | Operations |
| 2 | 4 | Research |
| 3 | 1 | Sales |

5. How many employees are working in the sales department?

- `Select count(E.Employee_Id) [Employees Count],
D.Name As Department From EMPLOYEE As E
Join DEPARTMENT As D
On E.DEPARTMENT_ID = D.Department_Id
Where D.Name = 'Sales'
Group By D.Name`

| | Employees Count | Department |
|---|-----------------|------------|
| 1 | 1 | Sales |

6. Which is the department having greater than or equal to 5 employees? Display the department names in ascending order.

- `Select D.Name As Department From EMPLOYEE As E
Join DEPARTMENT As D
On E.DEPARTMENT_ID = D.Department_Id
Group By D.Name
Having count(E.Employee_Id) >= 5
Order By D.Name Asc`

7. How many jobs are there in the organization? Display with designations.

- `Select Count(*) As [Count Of Employees], DESIGNATION From Job
Group By DESIGNATION`

| | Count Of Employees | DESIGNATION |
|---|--------------------|--------------|
| 1 | 1 | ANALYST |
| 2 | 1 | CLERK |
| 3 | 1 | MANAGER |
| 4 | 1 | PRESIDENT |
| 5 | 1 | SALES_PERSON |
| 6 | 1 | STAFF |

8. How many employees are working in "New York"?

- `Select Count(E.Employee_Id) As [Count Of Employees] From EMPLOYEE As E
Join DEPARTMENT As D
On E.DEPARTMENT_ID = D.Department_Id
Join LOCATION As L
On L.Location_Id = D.Location_Id
Where L.City = 'New York'`

9. Display the employee details with salary grades. Use conditional statement to create a grade column.

- ```

Select EMPLOYEE_ID,FIRST_NAME+' '+LAST_NAME As [Employee Name],SALARY,
Case When SALARY > 2000 Then 'High'
When SALARY > 1000 Then 'Medium'
Else 'Low'
End As [Salary Grade]
From EMPLOYEE
Order By SALARY

```

|   | EMPLOYEE_ID | Employee Name | SALARY | Salary Grade |
|---|-------------|---------------|--------|--------------|
| 1 | 7369        | JOHN SMITH    | 800    | Low          |
| 2 | 7521        | CYNTHIA WARK  | 1250   | Medium       |
| 3 | 7499        | KEVIN ALLEN   | 1600   | Medium       |
| 4 | 7507        | LESLIE BAKER  | 2200   | High         |
| 5 | 7506        | LYNN DENNIS   | 2750   | High         |
| 6 | 7505        | JEAN DOYLE    | 2850   | High         |

**10. List out the number of employees grade wise. Use conditional statement to create a grade column.**

- ```

Select D.Name As Department,Count(E.Employee_Id)[Employees Count],
Case When Count(E.EMPLOYEE_ID)>=4 Then 'Max'
Else 'Vacent' End As Grade
From EMPLOYEE As E
Join DEPARTMENT As D
On D.Department_Id = E.DEPARTMENT_ID
Group By D.Name
Order By Grade Asc

```

| | Department | Employees Count | Grade |
|---|------------|-----------------|--------|
| 1 | Research | 4 | Max |
| 2 | Sales | 1 | Vacant |
| 3 | Operations | 1 | Vacant |

11. Display the employee salary grades and the number of employees between 2000 to 5000 range of salary.

- ```

Select EMPLOYEE_ID,FIRST_NAME+' '+LAST_NAME As [Employee Name],SALARY,
Case When SALARY > 2000 Then 'High'
When SALARY > 1000 Then 'Medium'
Else 'Low'
End As [Salary Grade]
From EMPLOYEE
Where SALARY Between 2000 And 5000
Order By SALARY

```

|   | EMPLOYEE_ID | Employee Name | SALARY | Salary Grade |
|---|-------------|---------------|--------|--------------|
| 1 | 7507        | LESLIE BAKER  | 2200   | High         |
| 2 | 7506        | LYNN DENNIS   | 2750   | High         |
| 3 | 7505        | JEAN DOYLE    | 2850   | High         |

## 12. Display all employees in sales or operation departments.

- `Select E.EMPLOYEE_ID,E.FIRST_NAME+' '+E.LAST_NAME As [Employee Name]`  
`From EMPLOYEE As E`  
`Join DEPARTMENT As D`  
`On D.Department_Id = E.DEPARTMENT_ID`  
`Where D.Name In ('Sales','Operations')`

|   | EMPLOYEE_ID | Employee Name |
|---|-------------|---------------|
| 1 | 7369        | JOHN SMITH    |
| 2 | 7507        | LESLIE BAKER  |

## SET Operators:

### 1. List out the distinct jobs in sales and accounting departments.

- `Select J.Designation From Job As J`  
`Join EMPLOYEE As E`  
`On E.JOB_ID = J.JOB_ID`  
`Join DEPARTMENT As D`  
`On D.Department_Id = E.DEPARTMENT_ID`  
`Where D.Name = 'Sales'`

`Union`

```
Select J.Designation From Job As J
Join EMPLOYEE As E
On E.JOB_ID = J.JOB_ID
Join DEPARTMENT As D
On D.Department_Id = E.DEPARTMENT_ID
Where D.Name = 'Accounting'
```

### 2. List out all the jobs in sales and accounting departments.

- `Select J.Designation From Job As J`  
`Join EMPLOYEE As E`  
`On E.JOB_ID = J.JOB_ID`  
`Join DEPARTMENT As D`  
`On D.Department_Id = E.DEPARTMENT_ID`  
`Where D.Name = 'Sales'`

`Union All`

```
Select J.Designation From Job As J
Join EMPLOYEE As E
On E.JOB_ID = J.JOB_ID
Join DEPARTMENT As D
On D.Department_Id = E.DEPARTMENT_ID
Where D.Name = 'Accounting'
```

### 3. List out the common jobs in research and accounting departments in ascending order.

- With First As (  
Select J.Designation From Job As J  
Join EMPLOYEE As E  
On E.JOB\_ID = J.JOB\_ID  
Join DEPARTMENT As D  
On D.Department\_Id = E.DEPARTMENT\_ID  
Where D.Name = 'Research'  
Union  
Select J.Designation From Job As J  
Join EMPLOYEE As E  
On E.JOB\_ID = J.JOB\_ID  
Join DEPARTMENT As D  
On D.Department\_Id = E.DEPARTMENT\_ID  
Where D.Name = 'Accounting' )  
  
Select \* From First Order By Designation Asc

|   | Designation  |
|---|--------------|
| 1 | MANAGER      |
| 2 | SALES_PERSON |

## Subqueries:

### 1. Display the employees list who got the maximum salary.

- Select EMPLOYEE\_ID,First\_Name From EMPLOYEE  
Where SALARY in (Select MAX(SALARY) From EMPLOYEE)

|   | EMPLOYEE_ID | First_Name |
|---|-------------|------------|
| 1 | 7505        | JEAN       |

### 2. Display the employees who are working in the sales department.

- Select E.EMPLOYEE\_ID,E.First\_Name From EMPLOYEE As E  
Where DEPARTMENT\_ID in  
(Select D.DEPARTMENT\_ID From DEPARTMENT As D  
Where D.Department\_Id = E.DEPARTMENT\_ID And D.Name = 'Sales')

|   | EMPLOYEE_ID | First_Name |
|---|-------------|------------|
| 1 | 7369        | JOHN       |

### 3. Display the employees who are working as 'Clerk'.

- `Select E.EMPLOYEE_ID,E.First_Name From EMPLOYEE As E  
Where JOB_ID in  
(Select J.Job_Id From Job As J  
Where J.JOB_ID = E.JOB_ID And J.DESIGNATION = 'Clerk')`

|   | EMPLOYEE_ID | First_Name |
|---|-------------|------------|
| 1 | 7369        | JOHN       |

### 4. Display the list of employees who are living in "New York".

- `Select E.EMPLOYEE_ID,E.First_Name From EMPLOYEE As E  
Where DEPARTMENT_ID in  
(Select D.DEPARTMENT_ID From DEPARTMENT As D  
Where D.Department_Id = E.DEPARTMENT_ID And D.Location_Id in  
(Select Location_Id From LOCATION As L  
Where L.Location_Id = D.Location_Id And City = 'New York'))`

### 5. Find out the number of employees working in the sales department.

- `Select Count(*) As Count From EMPLOYEE Where DEPARTMENT_ID in  
(Select Department_Id From DEPARTMENT Where Name = 'Sales')`

|   | Count |
|---|-------|
| 1 | 1     |

### 6. Update the salaries of employees who are working as clerks on the basis of 10%.

- `Begin Transaction`

`Update E Set E.Salary = E.Salary * 1.1 From EMPLOYEE As E  
Where E.JOB_ID In  
(Select J.Job_Id from Job As J  
Where J.JOB_ID = E.JOB_ID And J.DESIGNATION = 'Clerk')`

`Select * From EMPLOYEE`

`Rollback`

|   | EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE  | SALARY | COMM | DEPARTMENT |
|---|-------------|-----------|------------|-------------|--------|------------|------------|--------|------|------------|
| 1 | 7369        | SMITH     | JOHN       | Q           | 667    | 7902       | 1984-12-17 | 880    | NULL | 20         |
| 2 | 7499        | ALLEN     | KEVIN      | J           | 670    | 7698       | 1984-02-20 | 1600   | 300  | 30         |
| 3 | 7505        | DOYLE     | JEAN       | K           | 671    | 7839       | 1985-04-04 | 2850   | NULL | 30         |
| 4 | 7506        | DENNIS    | LYNN       | S           | 671    | 7839       | 1985-05-15 | 2750   | NULL | 30         |
| 5 | 7507        | BAKER     | LESLIE     | D           | 671    | 7839       | 1985-06-10 | 2200   | NULL | 40         |
| 6 | 7521        | WARK      | CYNTHIA    | D           | 670    | 7698       | 1985-02-22 | 1250   | 500  | 30         |



## 7. Delete the employees who are working in the accounting department.

- Begin Transaction

```
Delete From E From EMPLOYEE As E
Where E.Department_Id in
(Select Department_Id From DEPARTMENT Where Name = 'Accounting')
```

Rollback

## 8. Display the second highest salary drawing employee details.

- Select Salary From  
(Select Salary,DENSE\_RANK()over(Order By Salary Desc) As Sub From EMPLOYEE) As Sub2  
Where Sub = 2

|   | Salary |
|---|--------|
| 1 | 2750   |

## 9. Display the nth highest salary drawing employee details.

- Select \*,Salary From  
(Select \*,DENSE\_RANK()over(Order By Salary Desc) As Sub From EMPLOYEE) As Sub2  
Where Sub = 5

## 10. List out the employees who earn more than every employee in department 30.

- Select EMPLOYEE\_ID,FIRST\_NAME From EMPLOYEE  
Where SALARY >  
(Select Max(SALARY) From EMPLOYEE Where DEPARTMENT\_ID = 30)

## 11. List out the employees who earn more than the lowest salary in department. Find out whose department has no employees.

- Select EMPLOYEE\_ID,FIRST\_NAME From EMPLOYEE  
Where SALARY >  
(Select Min(SALARY) From EMPLOYEE)
- Select Name From DEPARTMENT Where Department\_Id Not In  
(Select Department\_Id From EMPLOYEE)

|   | Name       |
|---|------------|
| 1 | Accounting |

## 12. Find out which department has no employees.

- Select Name From DEPARTMENT Where Department\_Id Not In  
(Select Department\_Id From EMPLOYEE)

**13. Find out the employees who earn greater than the average salary for their department.**

- `Select DEPARTMENT_ID,Max(Salary) As [Maximum Salary] From EMPLOYEE  
Where SALARY > (Select Avg(SALARY) From EMPLOYEE)  
Group By DEPARTMENT_ID`

|   | DEPARTMENT_ID | Maximum Salary |
|---|---------------|----------------|
| 1 | 30            | 2850           |
| 2 | 40            | 2200           |

# SQL Case Study – 3

## Problem Statement:

You are the database developer of an international bank. You are responsible for managing the bank's database. You want to use the data to answer a few questions about your customers regarding withdrawal, deposit and so on, especially about the transaction amount on a particular date across various regions of the world. Perform SQL queries to get the key insights of a customer.

## Dataset:

The 3 key datasets for this case study are:

- a. **Continent:** The Continent table has two attributes i.e., region\_id and region\_name, where region\_name consists of different continents such as Asia, Europe, Africa etc., assigned with the unique region id.
- b. **Customers:** The Customers table has four attributes named customer\_id, region\_id, start\_date and end\_date which consists of 3500 records.
- c. **Transaction:** Finally, the Transaction table contains around 5850 records and has four attributes named customer\_id, txn\_date, txn\_type and txn\_amount.

1. Display the count of customers in each region who have done the transaction in the year 2020.

- ```
Select R.region_name,Count(Distinct(C.Customer_Id)) As [Count Of Customers] From Customers As C
Join Continent As R
On R.region_id = C.region_id
Join [Transaction] As T
On T.customer_id = C.customer_id
Where Year(T.txn_date) = 2020
Group By R.region_name
```

| | region_name | Count Of Customers |
|---|-------------|--------------------|
| 1 | America | 105 |
| 2 | Asia | 95 |
| 3 | Africa | 102 |
| 4 | Australia | 110 |
| 5 | Europe | 88 |

2. Display the maximum and minimum transaction amount of each transaction type.

- `Select txn_type,Max(txn_amount) As Max,Min(txn_amount) As Min From [Transaction]
Group By txn_type`

| | txn_type | Max | Min |
|---|------------|------|-----|
| 1 | purchase | 999 | 1 |
| 2 | withdrawal | 999 | 1 |
| 3 | deposit | 1000 | 0 |

3. Display the customer id, region name and transaction amount where transaction type is deposit and transaction amount > 2000.

- `Select C.customer_id ,R.region_name,T.txn_amount From Customers As C
Join Continent As R
On R.region_id = C.region_id
Join [Transaction] As T
On T.customer_id = C.customer_id
Where txn_type = 'deposit' And T.txn_amount > 2000`

4. Find duplicate records in the Customer table.

- `Select Customer_Id From
(Select Customer_Id,
ROW_NUMBER()over(Partition By Customer_Id Order By Customer_Id) As Sub
From Customers)As Sub2
Where Sub > 1`

5. Display the customer id, region name, transaction type and transaction amount for the minimum transaction amount in deposit.

- `Select Top 1 C.Customer_Id,R.Region_name,T.txn_type,T.txn_amount
From Customers As C
Join Continent As R On R.region_id = C.region_id
Join [Transaction] As T On T.customer_id = C.customer_id
Where T.txn_type = 'Deposit'
Order By T.txn_amount Asc`

| | Customer_Id | Region_name | txn_type | txn_amount |
|---|-------------|-------------|----------|------------|
| 1 | 32 | Asia | deposit | 0 |

6. Create a stored procedure to display details of customers in the Transaction table where the transaction date is greater than Jun 2020.

- `Create proc sp_customerdetails
As Begin
Select * From [Transaction] Where datepart(month,txn_date)>6
And Year(txn_date) = 2020
End

Exec sp_customerdetails`

7. Create a stored procedure to insert a record in the Continent table.

- **Create proc** sp_insert_continent
(@region_id int,@region_name varchar(20))
As Begin
if not exists(**Select** 1 **From** Continent **Where** region_id = @region_id)
Begin
Insert into Continent(region_id,region_name) **Values**(@region_id,@region_name)
End
End

Exec sp_insert_continent 6,'Antartica'

8. Create a stored procedure to display the details of transactions that happened on a specific day.

- **Create proc** sp_txn_details
(@date date)
As Begin
Select * **From** [Transaction] **Where** txn_date = @date
End

Exec sp_txn_details '2020-01-21'

| | customer_id | txn_date | txn_type | txn_amount |
|----|-------------|------------|----------|------------|
| 1 | 429 | 2020-01-21 | deposit | 82 |
| 2 | 366 | 2020-01-21 | deposit | 965 |
| 3 | 230 | 2020-01-21 | deposit | 675 |
| 4 | 346 | 2020-01-21 | deposit | 916 |
| 5 | 352 | 2020-01-21 | deposit | 416 |
| 6 | 61 | 2020-01-21 | deposit | 319 |
| 7 | 408 | 2020-01-21 | deposit | 514 |
| 8 | 371 | 2020-01-21 | deposit | 528 |
| 9 | 9 | 2020-01-21 | deposit | 669 |
| 10 | 23 | 2020-01-21 | deposit | 334 |
| 11 | 126 | 2020-01-21 | deposit | 120 |
| 12 | 323 | 2020-01-21 | deposit | 603 |
| 13 | 40 | 2020-01-21 | deposit | 857 |
| 14 | 113 | 2020-01-21 | deposit | 14 |
| 15 | 300 | 2020-01-21 | deposit | 672 |
| 16 | 220 | 2020-01-21 | deposit | 307 |
| 17 | 355 | 2020-01-21 | deposit | 367 |
| 18 | 463 | 2020-01-21 | deposit | 881 |
| 19 | 38 | 2020-01-21 | deposit | 367 |
| 20 | 224 | 2020-01-21 | deposit | 487 |
| 21 | 376 | 2020-01-21 | deposit | 518 |

9. Create a user defined function to add 10% of the transaction amount in a table.

- **Create Function** fn_hike
(@txn_amount float)
Returns Float
As Begin
Declare @res float
Set @res = @txn_amount*1.1

```
Return @res
End
```

```
Select txn_amount,[Increased 10%] = dbo.fn_hike(txn_amount) From [Transaction]
```

| | txn_amount | Increased 10% |
|---|------------|---------------|
| 1 | 82 | 90.2 |
| 2 | 712 | 783.2 |
| 3 | 196 | 215.6 |
| 4 | 563 | 619.3 |
| 5 | 626 | 688.6 |

10. Create a user defined function to find the total transaction amount for a given transaction type.

- Create Function fn_sum_txn_amount
 (@txn_type varchar(30))
 Returns float
 As Begin
 Declare @res float
 Select @res = Sum(txn_amount) From [Transaction]
 Where txn_type = @txn_type
 Return @res
 End

```
Select dbo.fn_sum_txn_amount('deposit') as [Sum Of Txn_amount]
```

11. Create a table value function which comprises the columns customer_id, region_id,txn_date , txn_type , txn_amount which will retrieve data from the above table.

- Create Function fn_comprises()
 Returns Table
 As
 Return
 (Select C.customer_id,C.region_id,t.txn_date,t.txn_type,txn_amount
 From Customers As C
 Join [Transaction] As T
 On C.customer_id = T.customer_id)
 Select * From fn_comprises()

12. Create a TRY...CATCH block to print a region id and region name in a single column.

- Begin Try
 Select region_id + region_name From Continent
 End Try
 Begin Catch
 Select ERROR_MESSAGE() As [Error_Message]
 End Catch

| | Error_Message |
|---|--|
| 1 | Conversion failed when converting the nvarchar value 'Australia' to data type tinyint. |

13. Create a TRY...CATCH block to insert a value in the Continent table.

- `Begin Try`
`Insert into Continent values('Ayush',4)`
`End Try`
`Begin Catch`
`Select ERROR_MESSAGE() As [Error_Message]`
`End Catch`

| | Error_Message |
|---|--|
| 1 | Conversion failed when converting the nvarchar value 'Australia' to data type tinyint. |

14. Create a trigger to prevent deleting a table in a database.

- `Create Trigger tr_ddl_delete On Continent`
`For Delete`
`As Begin`
`Print 'Deleting From This Table Prevented By Master'`
`Rollback`
`End`

`Delete From Continent Where region_id = 6`

| Messages |
|--|
| Deleting From This Table Prevented By Master Msg 3609, Level 16, State 1, Line 540 The transaction ended in the trigger. The batch has been aborted. Completion time: 2024-01-29T20:57:13.1171355+05:30 |

15. Create a trigger to audit the data in a table.

- `Create Table Trigger_Message(Id int identity(1,1),Tr_Message Varchar(50))`

```
Create Trigger tr_DML On Continent
For Insert,Delete
As Begin
Declare @In_Id int,@Dl_Id int
Select @In_Id = region_id From inserted

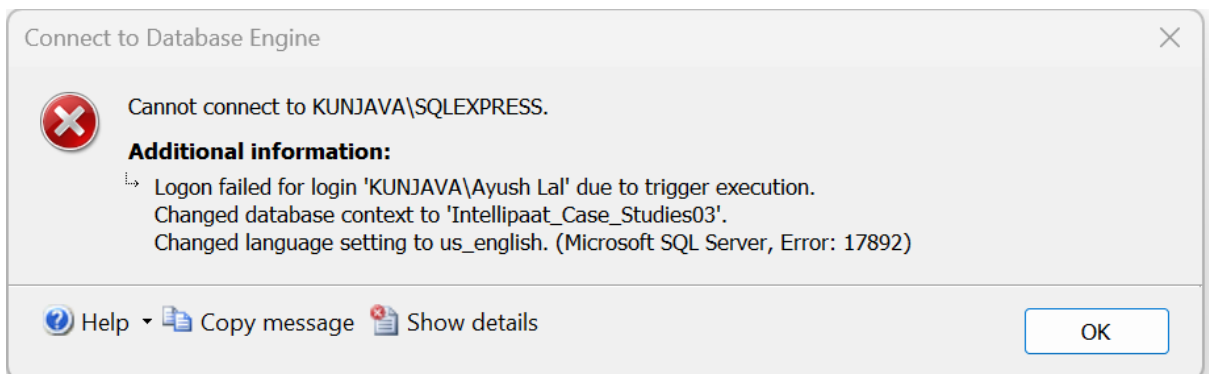
If @In_Id Is Not Null
Begin
Insert Into Trigger_Message(Tr_Message) values('New Id '+cast(@In_Id as varchar(20))
+' Added At '+cast(GETDATE() as varchar(20)));
End
Select @Dl_Id = region_id From deleted
If @Dl_Id Is Not Null
Begin
Insert Into Trigger_Message(Tr_Message) values('New Id '+cast(@Dl_Id as varchar(20))
+' Deleted At '+cast(GETDATE() as varchar(20)));
End
End;
```

```
Insert into Continent values(7,'India')
Delete From Continent Where region_id = 7
Select * From Trigger_Message
```

| | Id | Tr_Message |
|---|----|--|
| 1 | 1 | New Id 7 Added At Jan 29 2024 9:15PM |
| 2 | 2 | New Id 7 Deleted At Jan 29 2024 9:15PM |

16. Create a trigger to prevent login of the same user id in multiple pages.

- **Create Trigger** Tr_pr_mul_login **On** All **Server**
For Logon
As Begin
Declare @name **nvarchar**(50)
Set @name = **ORIGINAL_LOGIN**()
If (**Select** **Count**(*) **From** sys.dm_exec_sessions
Where is_user_process = 1 **And** original_login_name = @name) > 1
Print 'Access To New Query Window Blocked By Head'
Rollback
End



17. Display top n customers on the basis of transaction type.

- **Create Function** fn_top_n_txn
(@Id **int**,@txn_type **varchar**(20))
Returns Table
As Return
(**Select** Top (@Id) customer_id **From** Customers **Where** customer_id in
(**Select** customer_id **From** [Transaction] **Where** txn_type = @txn_type))

Select * **From** fn_top_n_txn(10,'Deposit')

18. Create a pivot table to display the total purchase, withdrawal and deposit for all the customers.

```
Select * From
(Select customer_id,txn_type,txn_amount From [Transaction]) As Source_Table
Pivot(
Sum(txn_amount) For txn_type in ([Deposit],[Purchase],[Withdrawal])
)As Pivot_Table
```


| | customer_id | Deposit | Purchase | Withdrawal |
|---|-------------|---------|----------|------------|
| 1 | 261 | 2045 | 778 | 1298 |
| 2 | 238 | 1642 | 914 | 1207 |
| 3 | 23 | 1168 | 916 | 930 |
| 4 | 355 | 2049 | 2271 | 630 |
| 5 | 378 | 3586 | 876 | 1120 |
| 6 | 46 | 3149 | 2504 | 541 |
| 7 | 215 | 2770 | 2073 | 283 |
| 8 | 69 | 3976 | 4531 | 2530 |
| 9 | 192 | 6793 | 3309 | 2345 |