

Omkar Mankame

SQL Queries

Query 1 - Employee_Manager Table – Self Join

Problem Statement: Display names of employees and their manager names.

Solution:

1. Created database.
2. Created a table with employees, id and manager. Inserted values.

```
3 • CREATE TABLE employee_table(  
4     employee_id int,  
5     full_name varchar(100),  
6     manager varchar(100)  
7 );  
8  
9 • INSERT INTO employee_table (employee_id, full_name, manager)  
10 VALUES ('101', 'Dan Doy', '102'),  
11          ('102', 'Tin Tan', '103'),  
12          ('103', 'Tin Tam', '104'),  
13          ('104', 'Yvan Yu', '104');
```

< Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	employee_id	full_name	manager
▶	101	Dan Doy	102
	102	Tin Tan	103
	103	Tin Tam	104
	104	Yvan Yu	104

Given Data:

Result: Used Self Join to produce result as seen below.

```

30
31 • SELECT A.full_name AS employee_name, B.full_name AS manager
32 FROM employee_table A
33 JOIN employee_table B
34 ON A.manager=B.employee_id
35 ORDER BY A.employee_id;
36

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	employee_name	manager
▶	Dan Doy	Tin Tan
	Tin Tan	Tin Tam
	Tin Tam	Yvan Yu
	Yvan Yu	Yvan Yu

Result 1 x

SQL code (Also available in GitHub Codespace) –

Use myproject;

```

CREATE TABLE employee_table(
  employee_id int,
  full_name varchar(100),
  manager varchar(100)
);

```

```

INSERT INTO employee_table (employee_id, full_name, manager)
VALUES ('101', 'Dan Doy ', '102'),
      ('102', 'Tin Tan', '103'),
      ('103', ' Tin Tam ', '104'),
      ('104', 'Yvan Yu ', '104');

```

```

SELECT A.full_name AS employee_name, B.full_name AS manager
FROM employee_table A
JOIN employee_table B
ON A.manager=B.employee_id
ORDER BY A.employee_id;

```

Query 2 – Credit Limit of Credit Card – Correlated Query

Problem Statement: Find the credit limit of a credit card for the maximum transaction date.

Solution:

1. Created database and Table with transaction id, credit card number, credit card limit and transaction date.
2. Inserted values in the table.

```
1 • USE casestudies;
2
3 • CREATE TABLE Transactions (
4     Transaction_Id int,
5     Credit_Card_No int,
6     Current_Credit_Limit int,
7     Transaction_Date date
8 );
9
10 • INSERT INTO Transactions (Transaction_Id, Credit_Card_No, Current_
11     VALUES ('101', '5909', '5000', '2024-04-04'),
12             ('102', '5909', '3000', '2024-01-01'),
13             ('103', '5909', '2000', '2023-12-12'),
14             ('104', '5000', '1000', '2024-07-07'),
15             ('105', '5000', '5000', '2024-05-05'),
16             ('106', '5000', '3000', '2023-12-12'),
17             ('107', '5000', '3000', '2023-11-11');
18
19 • SELECT Credit_Card_No,
20         Current_Credit_Limit,
```

Given Data:

```
81 • SELECT *
82 FROM transactions;
```

Transaction_Id	Credit_Card_No	Current_Credit_Limit	Transaction_Date
101	5909	5000	2024-04-04
102	5909	3000	2024-01-01
103	5909	2000	2023-12-12
104	5000	1000	2024-07-07
105	5000	5000	2024-05-05

transactions 3 x

Result: Final result was attained using joins, group by and alias.

```
69
70 • SELECT Credit_Card_No,
71       Current_Credit_Limit
72 FROM transactions AS a
73 WHERE Transaction_Date = (SELECT max(Transaction_Date)
74                           FROM transactions as b
75                           WHERE a.Credit_Card_No = b.Credit_Card_No);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Credit_Card_No	Current_Credit_Limit
5909	5000
5000	1000

SQL code (Also available in GitHub Codespace) –

USE casestudies;

```
CREATE TABLE Transactions (
  Transaction_Id int,
  Credit_Card_No int,
  Current_Credit_Limit int,
  Transaction_Date date
);
```

```
INSERT INTO Transactions (Transaction_Id, Credit_Card_No, Current_Credit_Limit, Transaction_Date)
VALUES ('101', '5909', '5000', '2024-04-04'),
      ('102', '5909', '3000', '2024-01-01'),
      ('103', '5909', '2000', '2023-12-12'),
      ('104', '5000', '1000', '2024-07-07'),
      ('105', '5000', '5000', '2024-05-05'),
      ('106', '5000', '3000', '2023-12-12'),
      ('107', '5000', '3000', '2023-11-11');
```

```
SELECT Credit_Card_No,
       Current_Credit_Limit
FROM transactions AS a
WHERE Transaction_Date = (SELECT max(Transaction_Date) FROM transactions as b WHERE a.Credit_Card_No =
b.Credit_Card_No);
```

Query 3 – GROUP BY_ORDER BY

Problem Statement: Find customer count from each country from customer table.

Given Data:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany

Result Table:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID);
```

Number of Records: 21

COUNT(CustomerID)	Country
1	Ireland
1	Norway
1	Poland
2	Belgium
2	Denmark
2	Finland
2	Sweden
2	Switzerland