Omkar Mankame

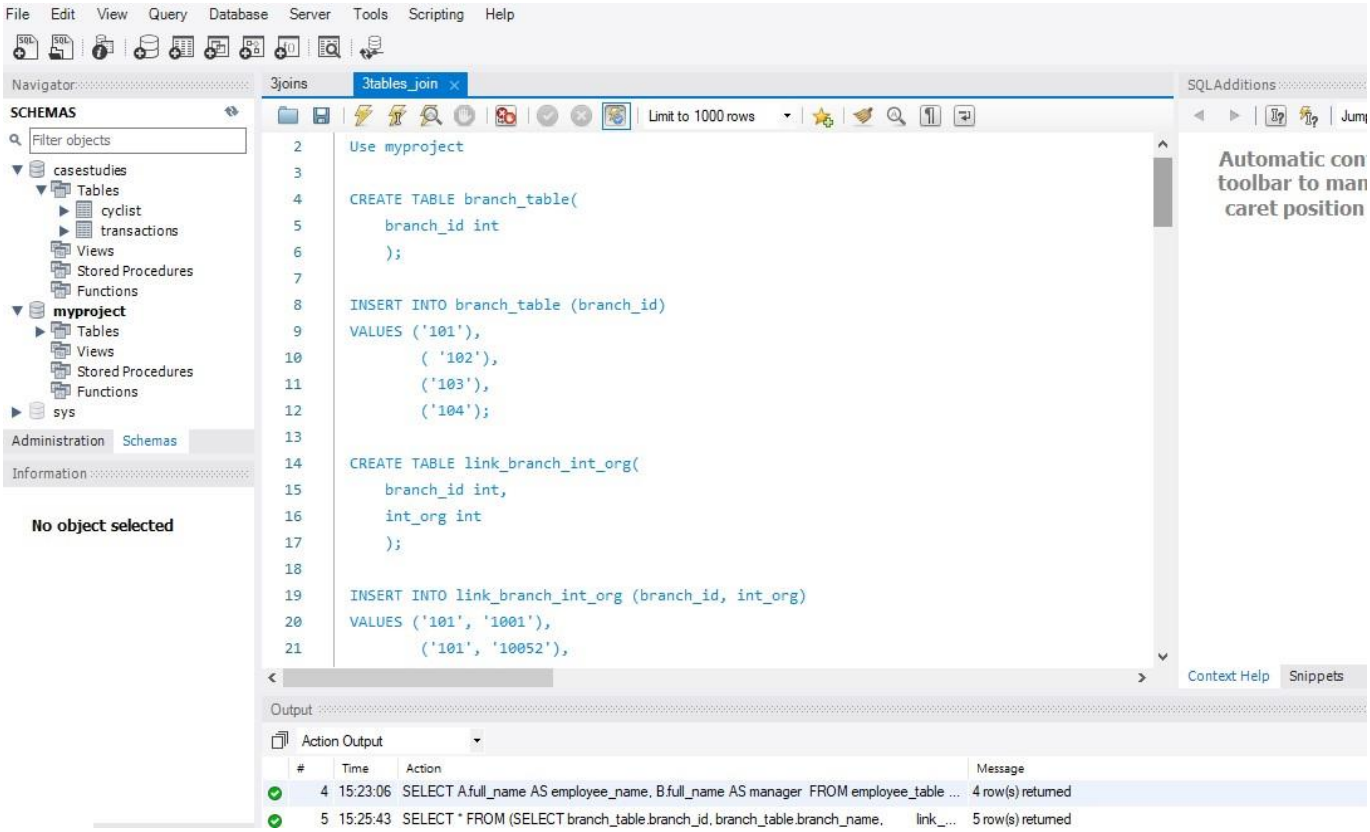
Few SQL projects

4 Sept 2024

I have done a few projects using SQL for data analysis. I have shared three complex and concise projects in this document. Others have been shared in the GitHub repository. The code is also mentioned in the GitHub codespace.

Project 1 – Joining 3 tables

1. Created a database and 3 tables.
2. Inserted values in the tables.



3. The Tables are as below.

```
SELECT *
FROM branch_table;
```

Result Grid | Filter Rows: | Edit:

branch_id	branch_name
101	Toronto
102	Guelph
103	Mississauga
104	Oakville
NULL	NULL

```
SELECT *
FROM internal_organization_table;
```

Result Grid | Filter Rows: | Export:

int_org_id	org_name	code_	party_id
10052	org Guelph	3	101
12	NULL	4	101
10001	org Toronto	3	102
7	NULL	4	103

```
SELECT *
FROM link_branch_int_org;
```

Result Grid | Filter Rows: | Export:

branch_id	int_org
101	1001
101	10052
102	1002
103	10052
104	NULL

4. Joined two tables each to get desired elements.

Left join is used to get all the values from int_org and the matching values from link_branch_int_org.

```
SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org AS group_code,
       int_org.org_name AS group_name,
       int_org.code_ AS region_code
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org;

/*
```

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

branch_id	group_code	group_name	region_code
101	1001	Toronto	3
101	10052	null	4
102	1002	Guelph	3
103	10052	null	4
104	NULL	NULL	NULL

Inner join is used to get all the values from both the tables including the ones that don't match.

```
SELECT branch_table.branch_id,
       branch_table.branch_name,
       link_branch_int_org.int_org
FROM branch_table
INNER JOIN link_branch_int_org
    ON link_branch_int_org.branch_id = branch_table.branch_id;
```

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

branch_id	branch_name	int_org
101	Toronto	1001
101	Toronto	10052
102	Guelph	1002
103	Mississauga	10052
104	Oakville	NULL

Result 27

5. Created 5 joins to get the desired result.

1st - Left Join

```
SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org AS group_code,
       int_org.org_name AS group_name,
       int_org.code_ AS region_code
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org;
```

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

branch_id	group_code	group_name	region_code
101	1001	Toronto	3
101	10052	null	4
102	1002	Guelph	3
103	10052	null	4
104	NULL	NULL	NULL

Result 3

2nd – Inner Join

3joins | 3tables_join* | employee_manager | Credit Card Limit

Limit to 1000 rows

```
WHERE branch_id = '103';

UPDATE branch_table
SET branch_name ='Oakville'
WHERE branch_id = '104';

SELECT branch_table.branch_id,
       branch_table.branch_name,
       link_branch_int_org.int_org
FROM branch_table
INNER JOIN link_branch_int_org
    ON link_branch_int_org.branch_id = branch_table.branch_id;
```

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

branch_id	branch_name	int_org
101	Toronto	1001
101	Toronto	10052
102	Guelph	1002
103	Mississauga	10052
104	Oakville	NULL

Result 4

3rd – Left Join

```
        ON link_branch_int_org.branch_id = branch_table.branch_id

SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
       int_org.org_name,
       int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org;

SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
       int_org.org_name,
```

branch_id	int_org	org_name	code_
101	1001	Toronto	3
101	10052	null	4
102	1002	Guelph	3
103	10052	null	4
104	NULL	NULL	NULL

4th – Left Join

```
        ON link_branch_int_org.int_org = int_org.int_org;

SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
       int_org.org_name,
       int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org
    WHERE int_org.code_ = '03';

SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
```

branch_id	int_org	org_name	code_
101	1001	Toronto	3
102	1002	Guelph	3

```
SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
       int_org.org_name,
       int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org
    WHERE int_org.code_ = '04';

SELECT *
FROM (SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
```

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

branch_id	int_org	org_name	code_
101	10052	null	4
103	10052	null	4

Limit to 1000 rows

```
SELECT *
FROM (SELECT branch_table.branch_id,
       branch_table.branch_name,
       link_branch_int_org.int_org
FROM branch_table
INNER JOIN link_branch_int_org
    ON link_branch_int_org.branch_id = branch_table.branch_id) AS A
LEFT JOIN (SELECT link_branch_int_org.branch_id,
       link_branch_int_org.int_org,
       int_org.org_name,
       int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
    ON link_branch_int_org.int_org = int_org.int_org
    WHERE int_org.code_ = '03') AS B
    ON A.int_org = B.int_org;
```

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

branch_id	branch_name	int_org	branch_id	int_org	org_name	code_
101	Toronto	1001	101	1001	Toronto	3
101	Toronto	10052	NULL	NULL	NULL	NULL
102	Guelph	1002	102	1002	Guelph	3
103	Mississauga	10052	NULL	NULL	NULL	NULL
104	Oakville	NULL	NULL	NULL	NULL	NULL

6. The complex query above created a join, another join in the subquery made it possible to get the desired result as shown in the image.

SQL code (Also available in GitHub Codespace) –

```
Use myproject;

/* CREATE TABLE branch_table(
        branch_id int
  );

INSERT INTO branch_table (branch_id)
VALUES ('101'),
                ( '102'),
    ('103'),
    ('104');

CREATE TABLE link_branch_int_org(
        branch_id int,
  int_org int
  );

INSERT INTO link_branch_int_org (branch_id, int_org)
VALUES ('101', '1001'),
                ('101', '10052'),
    ('102', '1002'),
    ('103', '10052'),
```

```
        ('104', null);

CREATE TABLE int_org(
            int_org int,
    org_name varchar(100),
    code_ int
    );

INSERT INTO int_org (int_org, org_name, code_)
VALUES ('1001', 'Toronto', '03'),
                        ('1002', 'Guelph', '03'),
        ('10052', 'null', '04');

UPDATE internal_organization_table
SET org_name = 'org Guelph'
WHERE int_org_id = '10052';

UPDATE internal_organization_table
SET org_name = 'org Toronto'
WHERE int_org_id = '10001';
*/

SELECT link_branch_int_org.branch_id,
                        link_branch_int_org.int_org AS group_code,
        int_org.org_name AS group_name,
        int_org.code_ AS region_code
FROM link_branch_int_org
LEFT JOIN int_org
            ON link_branch_int_org.int_org = int_org.int_org;
```

```
/*
ALTER TABLE branch_table
ADD branch_name varchar(100);

ALTER TABLE branch_table
ADD PRIMARY KEY (branch_id);

UPDATE branch_table
SET branch_name ='Toronto'
WHERE branch_id = '101';

UPDATE branch_table
SET branch_name ='Guelph'
WHERE branch_id = '102';

UPDATE branch_table
SET branch_name ='Mississauga'
WHERE branch_id = '103';

UPDATE branch_table
SET branch_name ='Oakville'
WHERE branch_id = '104';
*/

SELECT branch_table.branch_id,
                    branch_table.branch_name,
    link_branch_int_org.int_org
FROM branch_table
```

```
INNER JOIN link_branch_int_org
        ON link_branch_int_org.branch_id = branch_table.branch_id;

SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
    int_org.org_name,
    int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
        ON link_branch_int_org.int_org = int_org.int_org;

SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
    int_org.org_name,
    int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
        ON link_branch_int_org.int_org = int_org.int_org
  WHERE int_org.code_ = '03';

SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
    int_org.org_name,
    int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
        ON link_branch_int_org.int_org = int_org.int_org
  WHERE int_org.code_ = '04';
```

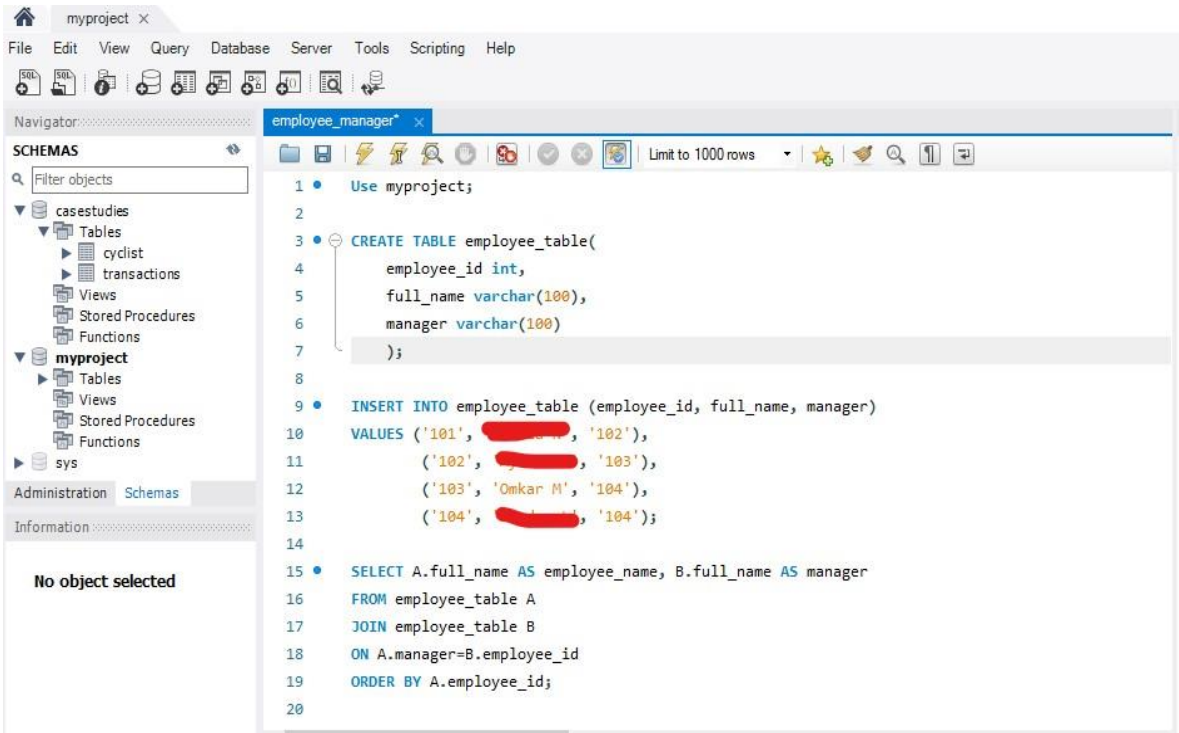
```
SELECT *
FROM (SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
      int_org.org_name,
      int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
          ON link_branch_int_org.int_org = int_org.int_org
   WHERE int_org.code_ = '03') AS A
LEFT JOIN (SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
      int_org.org_name,
      int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
          ON link_branch_int_org.int_org = int_org.int_org
   WHERE int_org.code_ = '04') AS B
          ON A.int_org = B.int_org;

SELECT *
FROM (SELECT branch_table.branch_id,
                branch_table.branch_name,
      link_branch_int_org.int_org
FROM branch_table
INNER JOIN link_branch_int_org
          ON link_branch_int_org.branch_id = branch_table.branch_id) AS A
LEFT JOIN (SELECT link_branch_int_org.branch_id,
                link_branch_int_org.int_org,
      int_org.org_name,
```

```
    int_org.code_
FROM link_branch_int_org
LEFT JOIN int_org
        ON link_branch_int_org.int_org = int_org.int_org
  WHERE int_org.code_ = '03') AS B
        ON A.int_org = B.int_org;

SELECT *
FROM branch_table;

SELECT *
FROM int_org;

SELECT *
FROM link_branch_int_org;

SELECT *
FROM internal_organization_table;
```

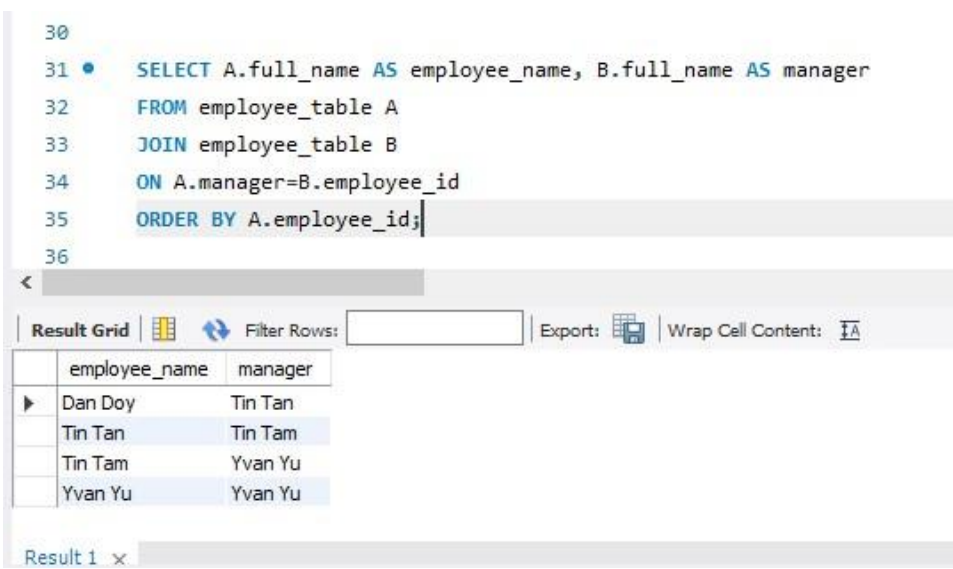
Project 2 - Employee_Manager Table – Self Join

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

The values have been updated in the result table. The below image is for reference and understanding only.



3. Used Self Join to produce result as seen below.



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;
```

Project 3 – Credit Limit of Credit Card

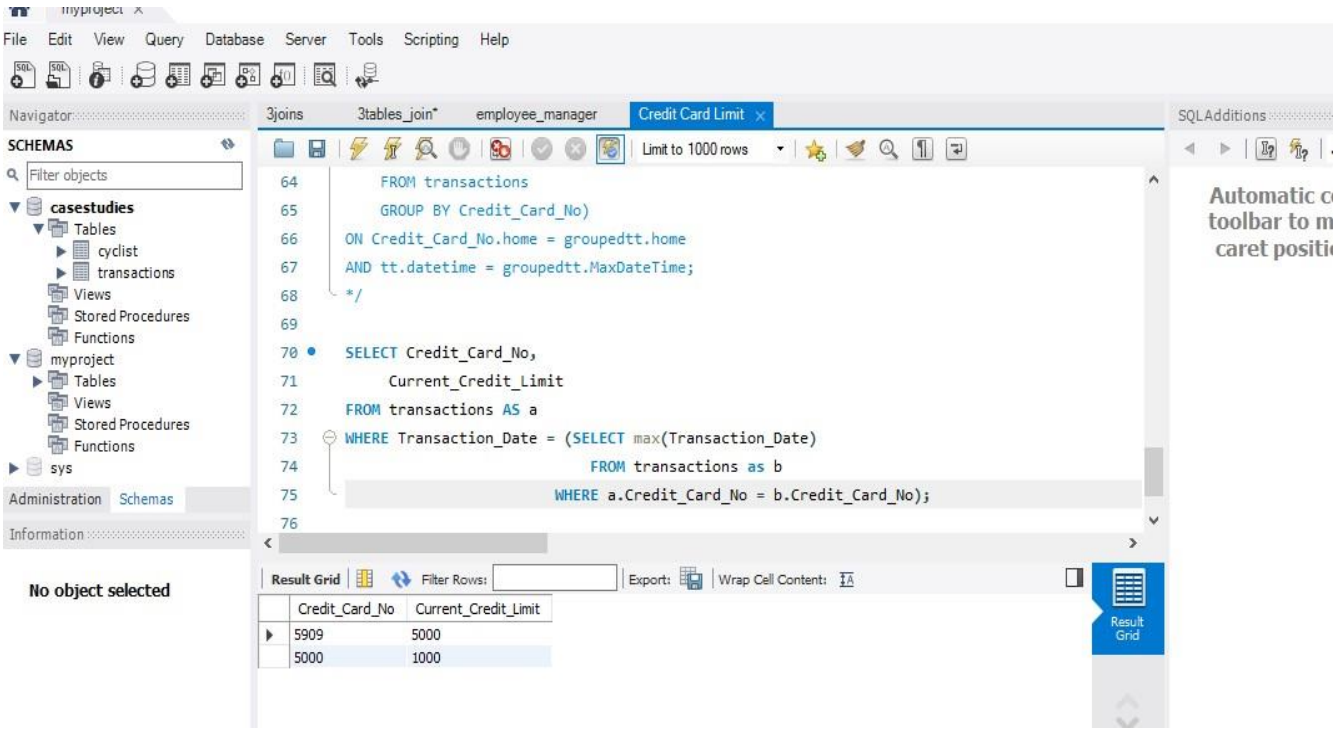
1. Created database and Table with transaction id, credit card number, credit card limit and transaction date.
2. Inserted values in the table.

```sql
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_
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,
```

3. Understood data and framed results using joins.

```sql

SELECT Credit_Card_No,
     Current_Credit_Limit,
     max(Transaction_Date)
FROM transactions
GROUP BY Credit_Card_No, Current_Credit_Limit;

SELECT Credit_Card_No,
     Current_Credit_Limit
FROM transactions
WHERE Transaction_Date = (SELECT max(Transaction_Date)
                          FROM transactions);

/*
SELECT Credit_Card_No,
     Current_Credit_Limit
FROM transactions
WHERE Transaction_Date = (SELECT max(Transaction_Date)
                          FROM transactions
                          GROUP BY Credit_Card_No)
```

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



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,
    max(Transaction_Date)
FROM transactions
GROUP BY Credit_Card_No, Current_Credit_Limit;

SELECT Credit_Card_No,
            Current_Credit_Limit
FROM transactions
WHERE Transaction_Date = (SELECT max(Transaction_Date)
                                                    FROM transactions);

/*
SELECT Credit_Card_No,
            Current_Credit_Limit
FROM transactions
WHERE Transaction_Date = (SELECT max(Transaction_Date)
                                                FROM transactions
                                                GROUP BY Credit_Card_No)
                    GROUP BY Transaction_Date
      ORDER BY Transaction_Date;
```

```
SELECT Credit_Card_No,
            Current_Credit_Limit
FROM transactions AS s1
JOIN (SELECT
  Credit_Card_No,
            Current_Credit_Limit,
  MAX(`Transaction_Date`)
FROM transactions AS dt
GROUP BY Credit_Card_No) AS s2;

  ON s1.Credit_Card_No = s2.Credit_Card_No
ORDER BY `datetime`;

SELECT Credit_Card_No,
            Current_Credit_Limit
FROM transactions
 Transcation_Date;

SELECT Credit_Card_No,
            Current_Credit_Limit
FROM transactions
INNER JOIN
   (SELECT Credit_Card_No, MAX(Transaction_Date)
   FROM transactions
   GROUP BY Credit_Card_No)
ON Credit_Card_No.home = groupedtt.home
AND tt.datetime = groupedtt.MaxDateTime;
*/
```

```
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);


SELECT max(Transaction_Date)
                          FROM transactions;
```