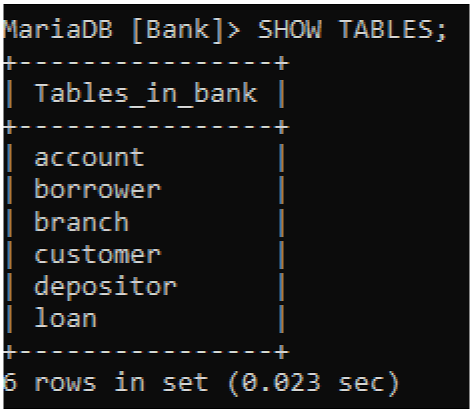
CSE370: Database Systems
LAB Assignment 03

Name: Fathin Ishrak

ID: 20301027

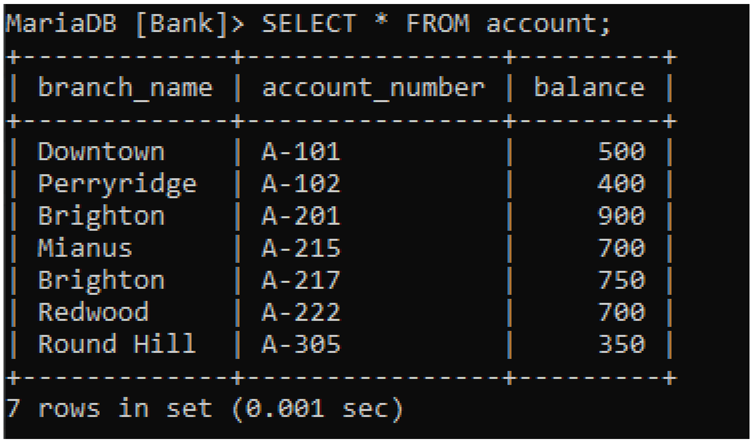
Lab Section: 07

SHOW TABLES;



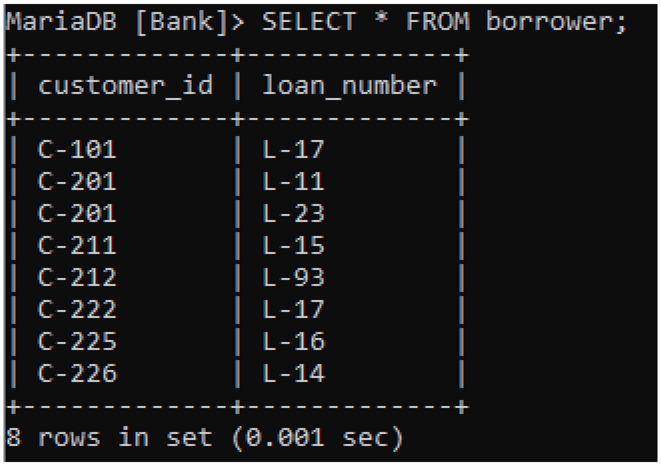
```
MariaDB [Bank]> SHOW TABLES;
+----------------+
| Tables_in_bank |
+----------------+
| account        |
| borrower       |
| branch         |
| customer       |
| depositor      |
| loan           |
+----------------+
6 rows in set (0.023 sec)
```

SELECT * FROM account;



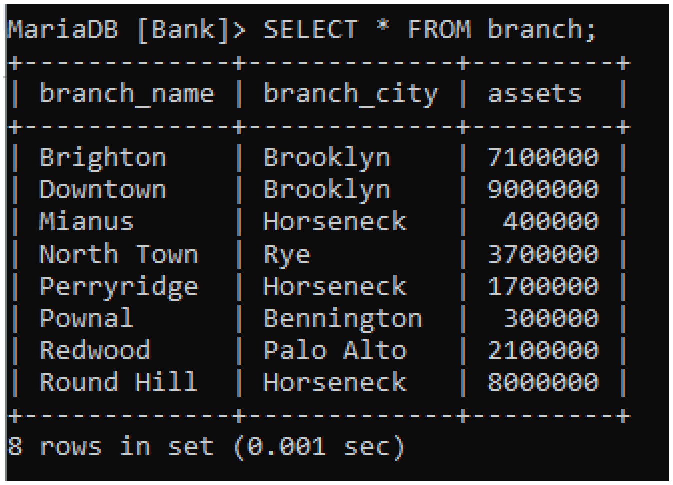
```
MariaDB [Bank]> SELECT * FROM account;
+-------------+----------------+---------+
| branch_name | account_number | balance |
+-------------+----------------+---------+
| Downtown    | A-101          |     500 |
| Perryridge  | A-102          |     400 |
| Brighton    | A-201          |     900 |
| Mianus      | A-215          |     700 |
| Brighton    | A-217          |     750 |
| Redwood     | A-222          |     700 |
| Round Hill  | A-305          |     350 |
+-------------+----------------+---------+
7 rows in set (0.001 sec)
```

SELECT * FROM borrower;



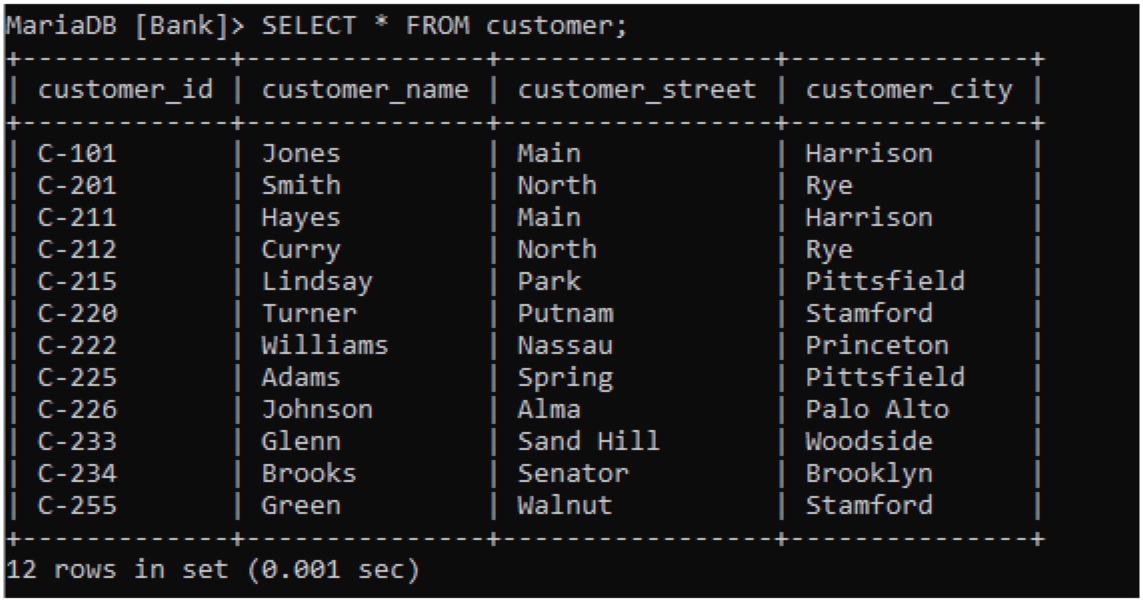
```
MariaDB [Bank]> SELECT * FROM borrower;
+-------------+-------------+
| customer_id | loan_number |
+-------------+-------------+
| C-101       | L-17        |
| C-201       | L-11        |
| C-201       | L-23        |
| C-211       | L-15        |
| C-212       | L-93        |
| C-222       | L-17        |
| C-225       | L-16        |
| C-226       | L-14        |
+-------------+-------------+
8 rows in set (0.001 sec)
```

SELECT * FROM branch;



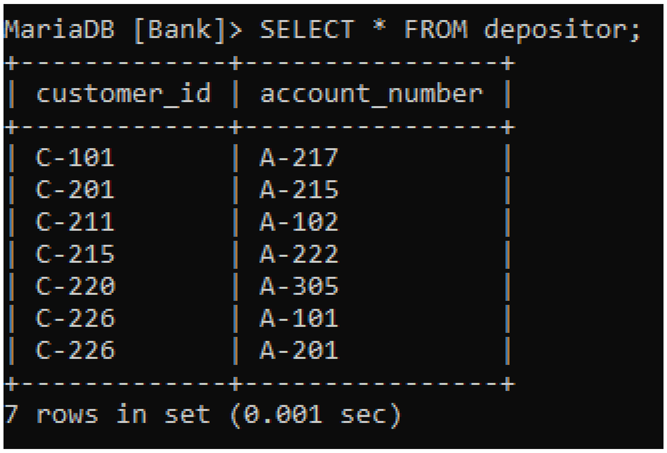
```
MariaDB [Bank]> SELECT * FROM branch;
+-------------+-------------+---------+
| branch_name | branch_city | assets  |
+-------------+-------------+---------+
| Brighton    | Brooklyn    | 7100000 |
| Downtown    | Brooklyn    | 9000000 |
| Mianus      | Horseneck   |  400000 |
| North Town  | Rye         | 3700000 |
| Perryridge  | Horseneck   | 1700000 |
| Pownal      | Bennington  |  300000 |
| Redwood     | Palo Alto   | 2100000 |
| Round Hill  | Horseneck   | 8000000 |
+-------------+-------------+---------+
8 rows in set (0.001 sec)
```

SELECT * FROM customer;



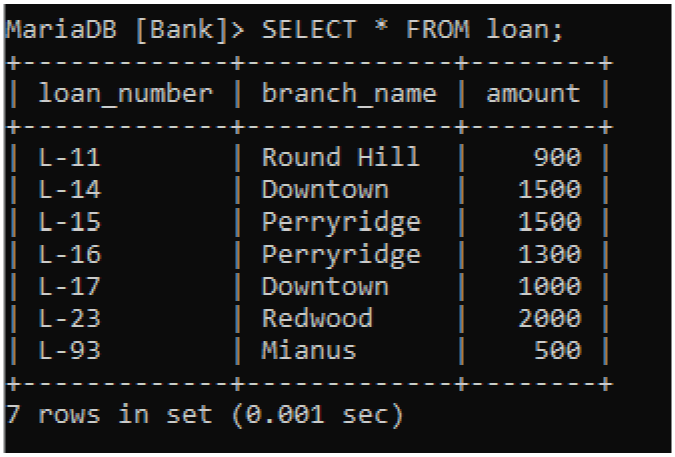
```
MariaDB [Bank]> SELECT * FROM customer;
+-------------+---------------+-----------------+---------------+
| customer_id | customer_name | customer_street | customer_city |
+-------------+---------------+-----------------+---------------+
| C-101       | Jones         | Main            | Harrison      |
| C-201       | Smith         | North           | Rye           |
| C-211       | Hayes         | Main            | Harrison      |
| C-212       | Curry         | North           | Rye           |
| C-215       | Lindsay       | Park            | Pittsfield    |
| C-220       | Turner        | Putnam          | Stamford      |
| C-222       | Williams      | Nassau          | Princeton     |
| C-225       | Adams         | Spring          | Pittsfield    |
| C-226       | Johnson       | Alma            | Palo Alto     |
| C-233       | Glenn         | Sand Hill       | Woodside      |
| C-234       | Brooks        | Senator         | Brooklyn      |
| C-255       | Green         | Walnut          | Stamford      |
+-------------+---------------+-----------------+---------------+
12 rows in set (0.001 sec)
```

SELECT * FROM depositor;



```
MariaDB [Bank]> SELECT * FROM depositor;
+-------------+----------------+
| customer_id | account_number |
+-------------+----------------+
| C-101       | A-217          |
| C-201       | A-215          |
| C-211       | A-102          |
| C-215       | A-222          |
| C-220       | A-305          |
| C-226       | A-101          |
| C-226       | A-201          |
+-------------+----------------+
7 rows in set (0.001 sec)
```

SELECT * FROM loan;



```
MariaDB [Bank]> SELECT * FROM loan;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L-11        | Round Hill  |    900 |
| L-14        | Downtown    |   1500 |
| L-15        | Perryridge  |   1500 |
| L-16        | Perryridge  |   1300 |
| L-17        | Downtown    |   1000 |
| L-23        | Redwood     |   2000 |
| L-93        | Mianus      |    500 |
+-------------+-------------+--------+
7 rows in set (0.001 sec)
```

Assignment Tasks:

1. **Find the name and loan number of all customers having a loan at the Downtown branch.**

   SELECT c.customer_name, l.loan_number

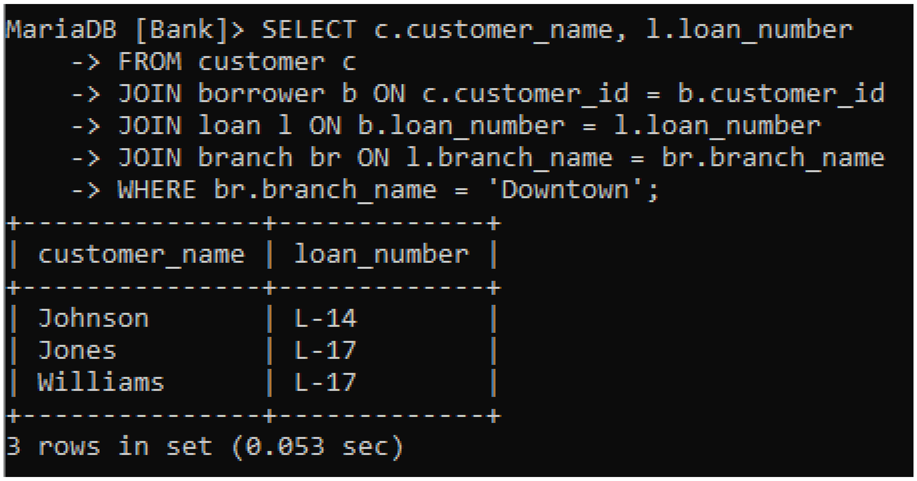
   FROM customer c

   JOIN borrower b ON c.customer_id = b.customer_id

   JOIN loan l ON b.loan_number = l.loan_number

   JOIN branch br ON l.branch_name = br.branch_name

   WHERE br.branch_name = 'Downtown';



```
MariaDB [Bank]> SELECT c.customer_name, l.loan_number
    -> FROM customer c
    -> JOIN borrower b ON c.customer_id = b.customer_id
    -> JOIN loan l ON b.loan_number = l.loan_number
    -> JOIN branch br ON l.branch_name = br.branch_name
    -> WHERE br.branch_name = 'Downtown';
+---------------+-------------+
| customer_name | loan_number |
+---------------+-------------+
| Johnson       | L-14        |
| Jones         | L-17        |
| Williams      | L-17        |
+---------------+-------------+
3 rows in set (0.053 sec)
```

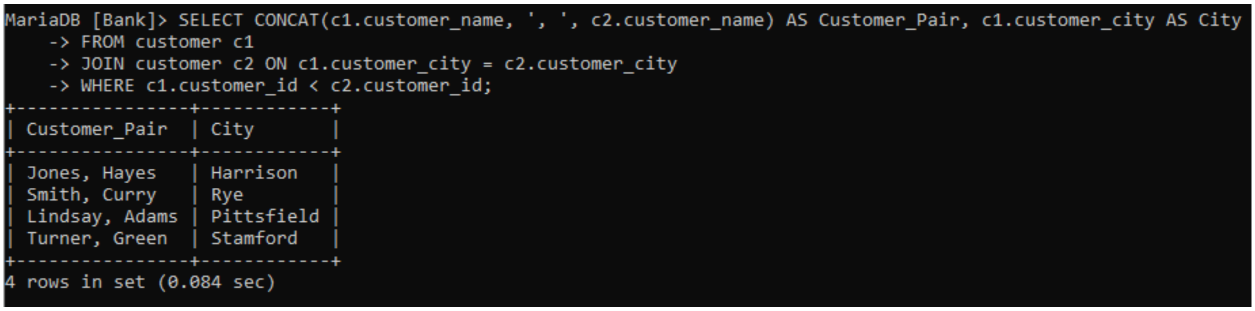
2. **Find all the possible pairs of customers who are from the same city. show in the format Customer1, Customer2, City.**

   SELECT CONCAT(c1.customer_name, ', ', c2.customer_name) AS Customer_Pair, c1.customer_city AS City

   FROM customer c1

JOIN customer c2 ON c1.customer_city = c2.customer_city

WHERE c1.customer_id < c2.customer_id;



```
MariaDB [Bank]> SELECT CONCAT(c1.customer_name, ', ', c2.customer_name) AS Customer_Pair, c1.customer_city AS City
    -> FROM customer c1
    -> JOIN customer c2 ON c1.customer_city = c2.customer_city
    -> WHERE c1.customer_id < c2.customer_id;
+----------------+------------+
| Customer_Pair  | City       |
+----------------+------------+
| Jones, Hayes   | Harrison   |
| Smith, Curry   | Rye        |
| Lindsay, Adams | Pittsfield |
| Turner, Green  | Stamford   |
+----------------+------------+
4 rows in set (0.084 sec)
```

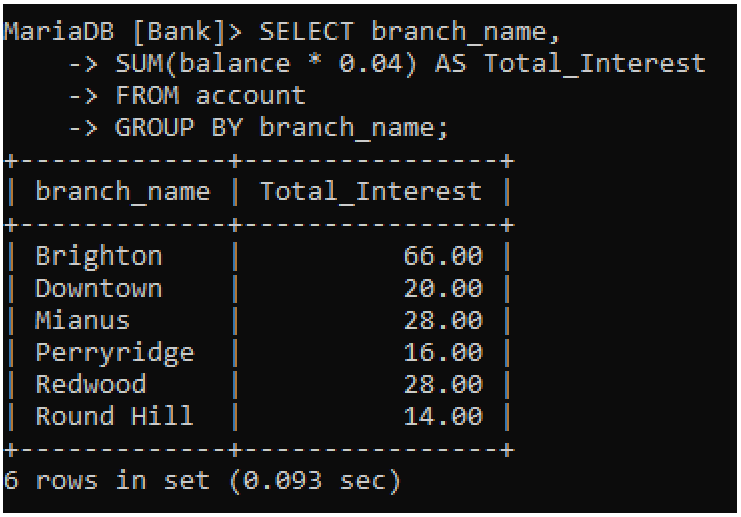
3. **If the bank gives out 4% interest to all accounts, show the total interest across each branch. Print Branch_name, Total_Interest.**

SELECT branch_name,

SUM(balance * 0.04) AS Total_Interest

FROM account

GROUP BY branch_name;



```
MariaDB [Bank]> SELECT branch_name,
    -> SUM(balance * 0.04) AS Total_Interest
    -> FROM account
    -> GROUP BY branch_name;
+-------------+----------------+
| branch_name | Total_Interest |
+-------------+----------------+
| Brighton    |          66.00 |
| Downtown    |          20.00 |
| Mianus      |          28.00 |
| Perryridge  |          16.00 |
| Redwood     |          28.00 |
| Round Hill  |          14.00 |
+-------------+----------------+
6 rows in set (0.093 sec)
```

4. **Find account numbers with the highest balances for each city in the database.**

SELECT a.account_number, b.branch_city, a.balance

FROM account a

JOIN branch b ON a.branch_name = b.branch_name

WHERE (b.branch_city, a.balance) IN

(

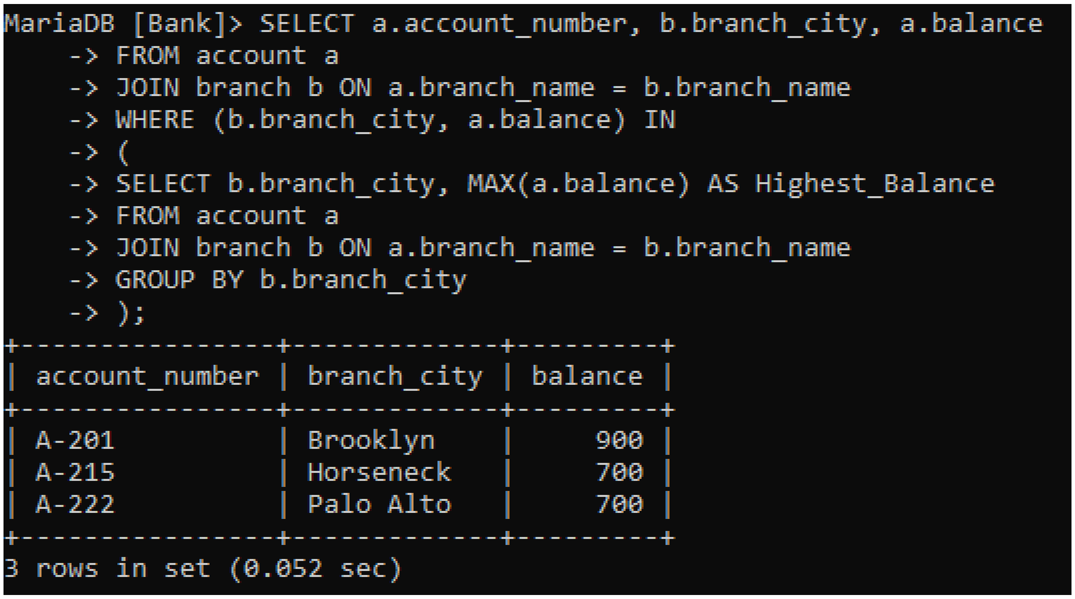
SELECT b.branch_city, MAX(a.balance) AS Highest_Balance

FROM account a

JOIN branch b ON a.branch_name = b.branch_name

GROUP BY b.branch_city

);



```
MariaDB [Bank]> SELECT a.account_number, b.branch_city, a.balance
    -> FROM account a
    -> JOIN branch b ON a.branch_name = b.branch_name
    -> WHERE (b.branch_city, a.balance) IN
    -> (
    -> SELECT b.branch_city, MAX(a.balance) AS Highest_Balance
    -> FROM account a
    -> JOIN branch b ON a.branch_name = b.branch_name
    -> GROUP BY b.branch_city
    -> );
+----------------+-------------+---------+
| account_number | branch_city | balance |
+----------------+-------------+---------+
| A-201          | Brooklyn    |     900 |
| A-215          | Horseneck   |     700 |
| A-222          | Palo Alto   |     700 |
+----------------+-------------+---------+
3 rows in set (0.052 sec)
```

5. **Show the loan number, loan amount, and name of customers with the top 5 highest loan amounts. The data should be sorted by increasing amounts, then decreasing loan numbers in case of the same loan amount. [Hint for top 5: Check the "limit" keyword in mysql**

SELECT l.loan_number, l.amount AS Loan_Amount, c.customer_name AS Customer_Name

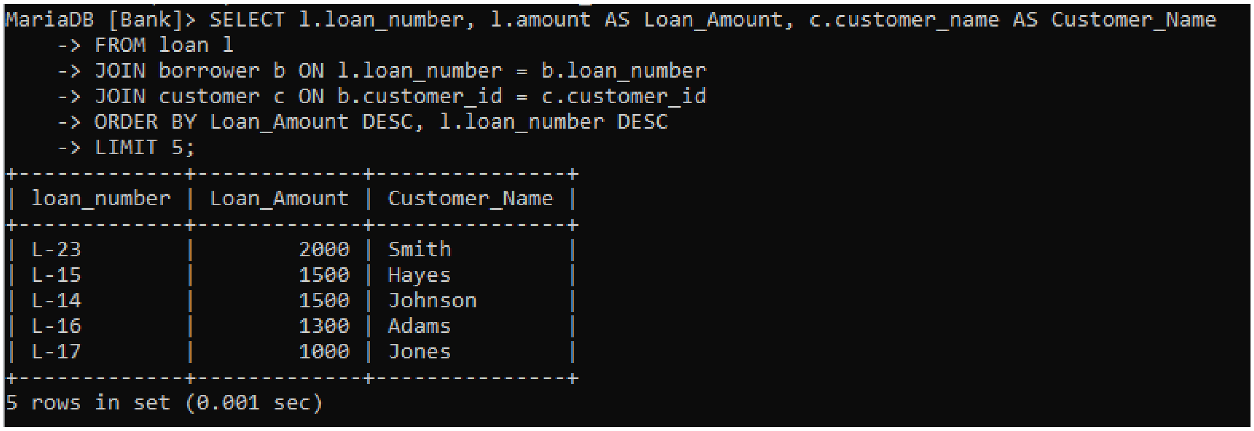
FROM loan l

JOIN borrower b ON l.loan_number = b.loan_number

JOIN customer c ON b.customer_id = c.customer_id

ORDER BY Loan_Amount DESC, l.loan_number DESC

LIMIT 5;



```
MariaDB [Bank]> SELECT l.loan_number, l.amount AS Loan_Amount, c.customer_name AS Customer_Name
    -> FROM loan l
    -> JOIN borrower b ON l.loan_number = b.loan_number
    -> JOIN customer c ON b.customer_id = c.customer_id
    -> ORDER BY Loan_Amount DESC, l.loan_number DESC
    -> LIMIT 5;
+-------------+-------------+---------------+
| loan_number | Loan_Amount | Customer_Name |
+-------------+-------------+---------------+
| L-23        |        2000 | Smith         |
| L-15        |        1500 | Hayes         |
| L-14        |        1500 | Johnson       |
| L-16        |        1300 | Adams         |
| L-17        |        1000 | Jones         |
+-------------+-------------+---------------+
5 rows in set (0.001 sec)
```

6. **Find the names of customers with an account and also a loan at the Perryridge branch.**

SELECT DISTINCT c.customer_name

FROM customer c

JOIN depositor d ON c.customer_id = d.customer_id

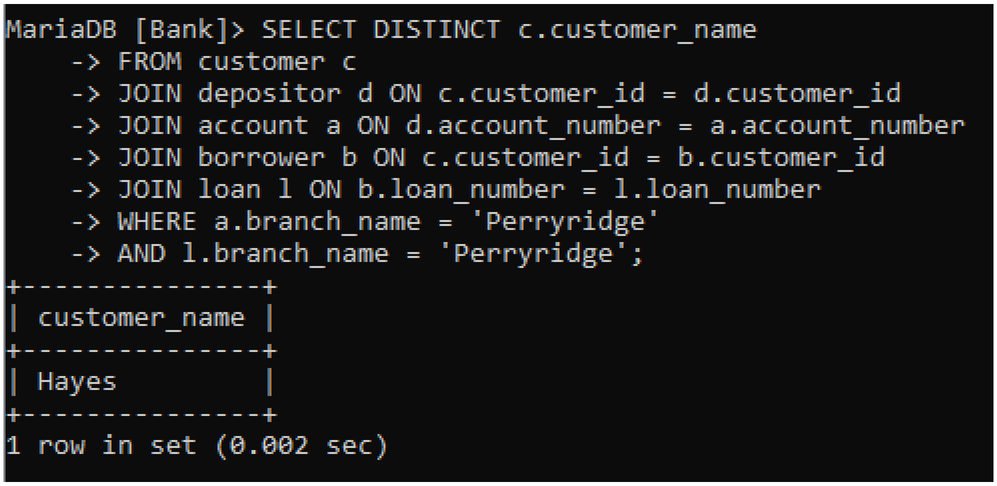
JOIN account a ON d.account_number = a.account_number

JOIN borrower b ON c.customer_id = b.customer_id

JOIN loan l ON b.loan_number = l.loan_number

WHERE a.branch_name = 'Perryridge'

AND l.branch_name = 'Perryridge';



```
MariaDB [Bank]> SELECT DISTINCT c.customer_name
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> JOIN borrower b ON c.customer_id = b.customer_id
    -> JOIN loan l ON b.loan_number = l.loan_number
    -> WHERE a.branch_name = 'Perryridge'
    -> AND l.branch_name = 'Perryridge';
+---------------+
| customer_name |
+---------------+
| Hayes         |
+---------------+
1 row in set (0.002 sec)
```

7. **Find the total loan amount of all customers having at least 2 loans from the bank. Show in format customer name, total_loan.**

SELECT c.customer_name AS Customer_Name,

SUM(l.amount) AS Total_Loan

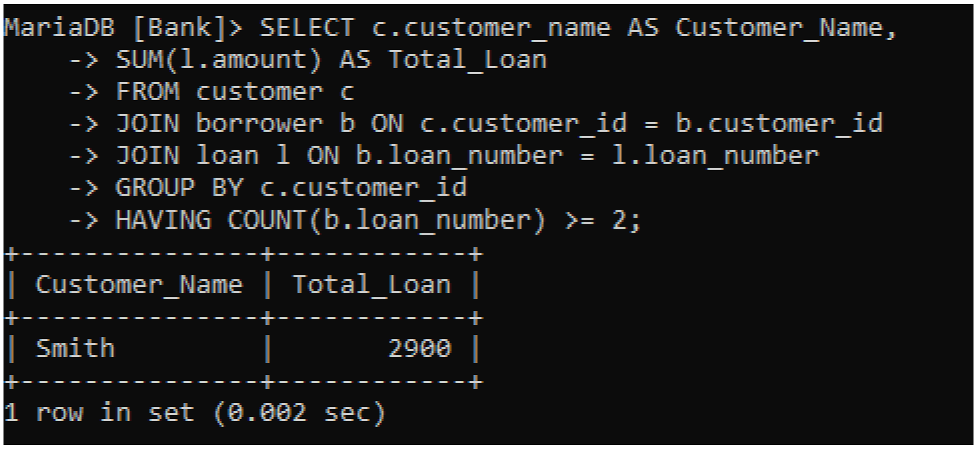
FROM customer c

JOIN borrower b ON c.customer_id = b.customer_id

JOIN loan l ON b.loan_number = l.loan_number

GROUP BY c.customer_id

HAVING COUNT(b.loan_number) >= 2;

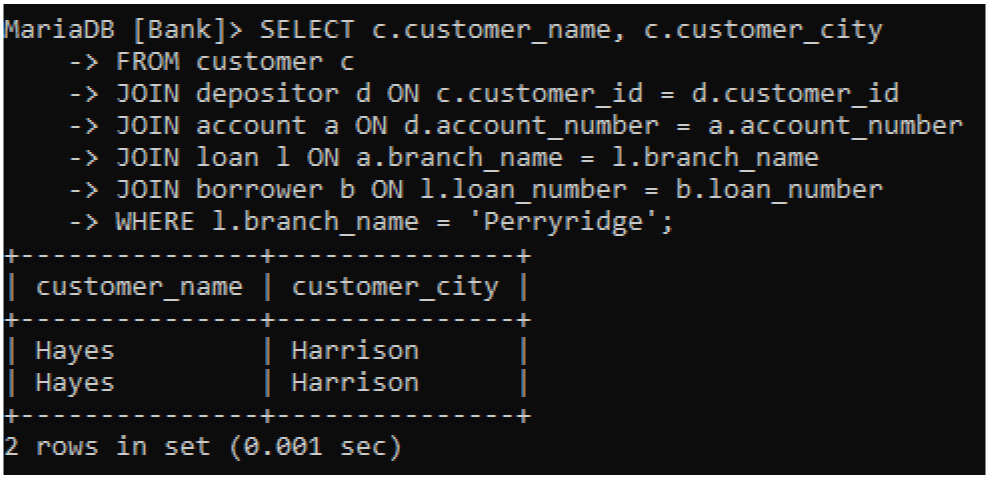


```
MariaDB [Bank]> SELECT c.customer_name AS Customer_Name,
    -> SUM(l.amount) AS Total_Loan
    -> FROM customer c
    -> JOIN borrower b ON c.customer_id = b.customer_id
    -> JOIN loan l ON b.loan_number = l.loan_number
    -> GROUP BY c.customer_id
    -> HAVING COUNT(b.loan_number) >= 2;
+---------------+------------+
| Customer_Name | Total_Loan |
+---------------+------------+
| Smith         |       2900 |
+---------------+------------+
1 row in set (0.002 sec)
```

Task 6 (LAB Sheet):

1. **Find names and cities of customers who have a loan at Perryridge branch**

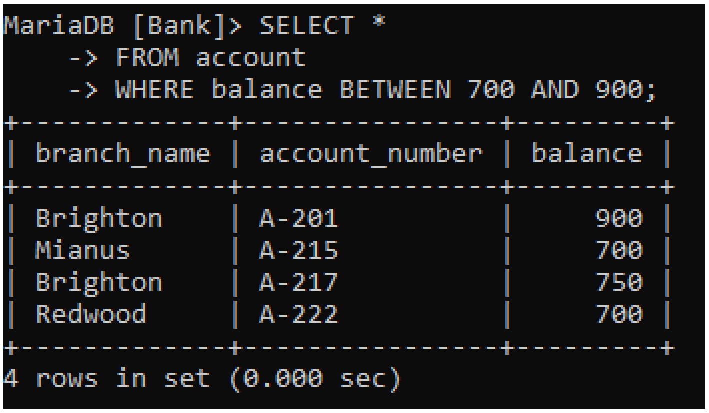
```
SELECT c.customer_name, c.customer_city
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN account a ON d.account_number = a.account_number
JOIN loan l ON a.branch_name = l.branch_name
JOIN borrower b ON l.loan_number = b.loan_number
WHERE l.branch_name = 'Perryridge';
```



```
MariaDB [Bank]> SELECT c.customer_name, c.customer_city
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> JOIN loan l ON a.branch_name = l.branch_name
    -> JOIN borrower b ON l.loan_number = b.loan_number
    -> WHERE l.branch_name = 'Perryridge';
+---------------+---------------+
| customer_name | customer_city |
+---------------+---------------+
| Hayes         | Harrison      |
| Hayes         | Harrison      |
+---------------+---------------+
2 rows in set (0.001 sec)
```

2. **Find the accounts with balances between 700 and 900.**

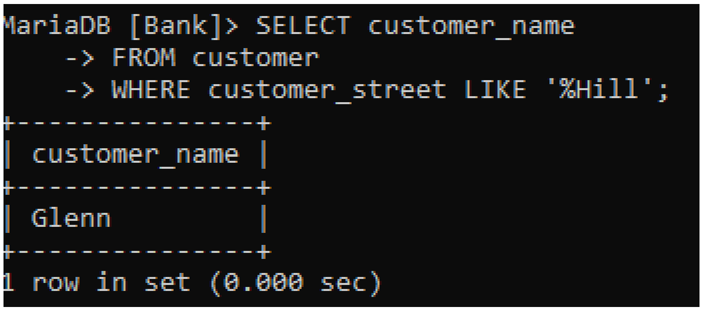
```
SELECT *
FROM account
WHERE balance BETWEEN 700 AND 900;
```



```
MariaDB [Bank]> SELECT *
    -> FROM account
    -> WHERE balance BETWEEN 700 AND 900;
+-------------+----------------+---------+
| branch_name | account_number | balance |
+-------------+----------------+---------+
| Brighton    | A-201          |     900 |
| Mianus      | A-215          |     700 |
| Brighton    | A-217          |     750 |
| Redwood     | A-222          |     700 |
+-------------+----------------+---------+
4 rows in set (0.000 sec)
```

3. **Find the names of customers on streets with names ending in "Hill".**

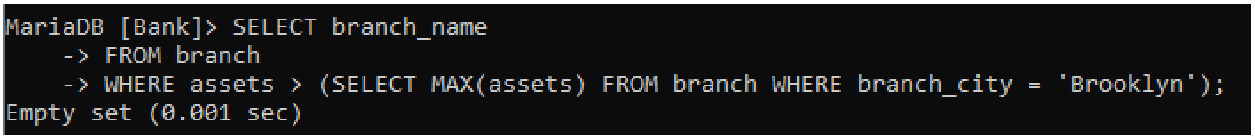
   SELECT customer_name
   FROM customer
   WHERE customer_street LIKE '%Hill';



```
MariaDB [Bank]> SELECT customer_name
    -> FROM customer
    -> WHERE customer_street LIKE '%Hill';
+---------------+
| customer_name |
+---------------+
| Glenn         |
+---------------+
1 row in set (0.000 sec)
```

4. **Find the names of branches whose assets are greater than the assets of some branch in Brooklyn.**

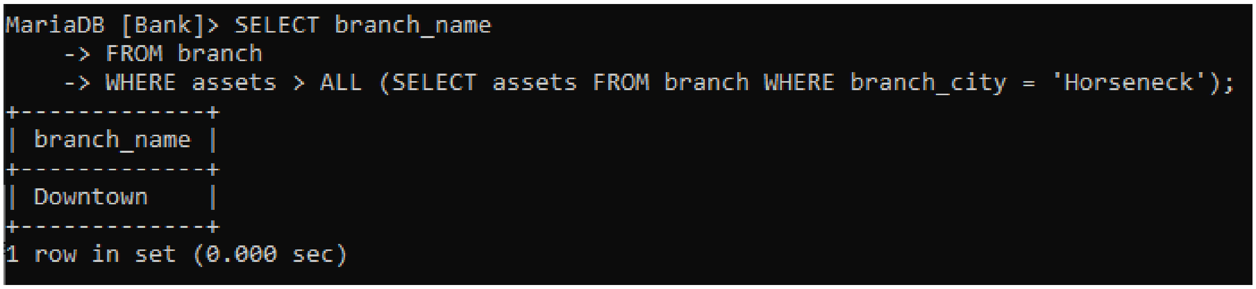
   SELECT branch_name
   FROM branch
   WHERE assets > (SELECT MAX(assets) FROM branch WHERE branch_city = 'Brooklyn');



```
MariaDB [Bank]> SELECT branch_name
    -> FROM branch
    -> WHERE assets > (SELECT MAX(assets) FROM branch WHERE branch_city = 'Brooklyn');
Empty set (0.001 sec)
```

5. **Find the set of names of branches whose assets are greater than the assets of all branches in Horseneck.**

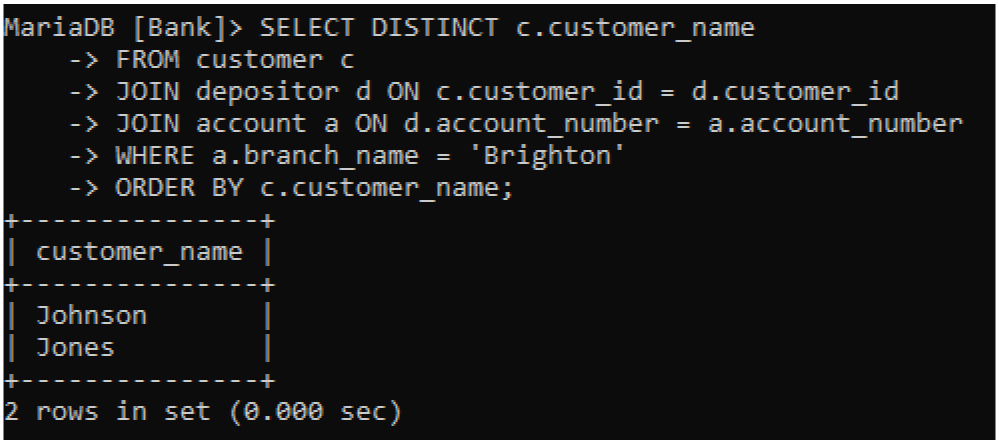
   SELECT branch_name
   FROM branch
   WHERE assets > ALL (SELECT assets FROM branch WHERE branch_city = 'Horseneck');



```
MariaDB [Bank]> SELECT branch_name
    -> FROM branch
    -> WHERE assets > ALL (SELECT assets FROM branch WHERE branch_city = 'Horseneck');
+-------------+
| branch_name |
+-------------+
| Downtown    |
+-------------+
1 row in set (0.000 sec)
```

6. **Find the set of names of customers at Brighton branch, in alphabetical order.**

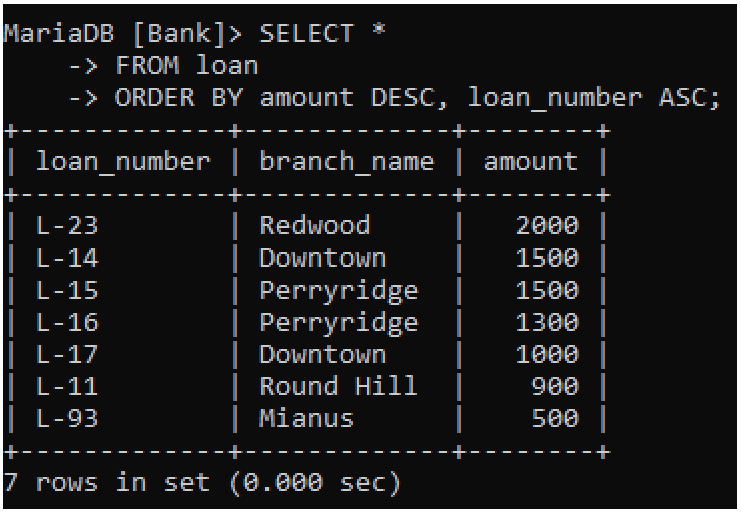
```
SELECT DISTINCT c.customer_name
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN account a ON d.account_number = a.account_number
WHERE a.branch_name = 'Brighton'
ORDER BY c.customer_name;
```



```
MariaDB [Bank]> SELECT DISTINCT c.customer_name
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> WHERE a.branch_name = 'Brighton'
    -> ORDER BY c.customer_name;
+---------------+
| customer_name |
+---------------+
| Johnson       |
| Jones         |
+---------------+
2 rows in set (0.000 sec)
```

7. **Show the loan data, ordered by decreasing amounts, then increasing loan numbers.**

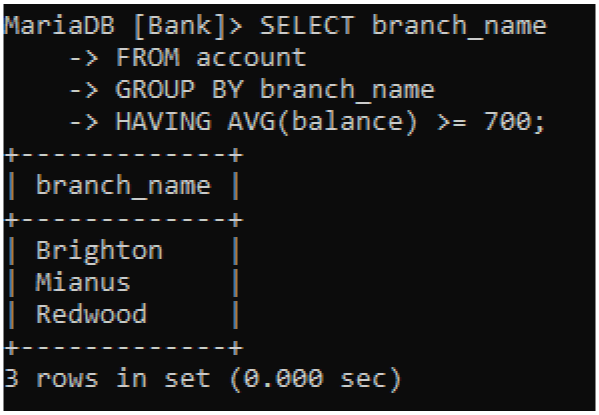
```
SELECT *
FROM loan
ORDER BY amount DESC, loan_number ASC;
```



```
MariaDB [Bank]> SELECT *
    -> FROM loan
    -> ORDER BY amount DESC, loan_number ASC;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L-23        | Redwood     |   2000 |
| L-14        | Downtown    |   1500 |
| L-15        | Perryridge  |   1500 |
| L-16        | Perryridge  |   1300 |
| L-17        | Downtown    |   1000 |
| L-11        | Round Hill  |    900 |
| L-93        | Mianus      |    500 |
+-------------+-------------+--------+
7 rows in set (0.000 sec)
```

8. **Find the names of branches having at least one account, with average balances greater than or equal 700.**

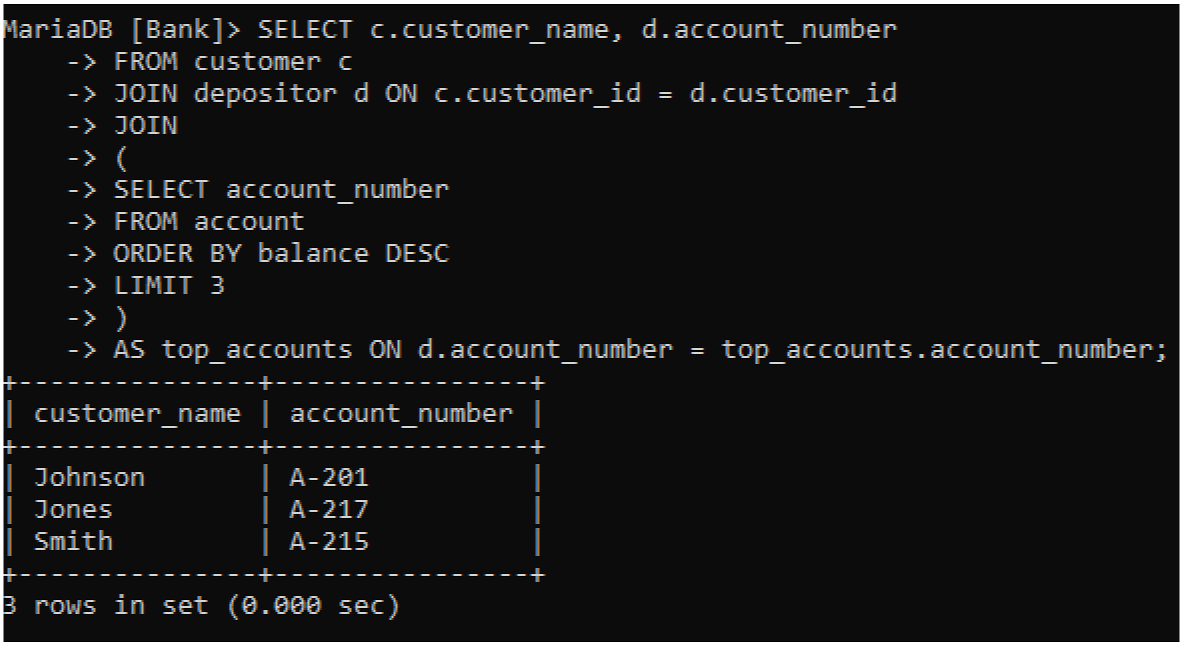
```
SELECT branch_name
FROM account
GROUP BY branch_name
HAVING AVG(balance) >= 700;
```



```
MariaDB [Bank]> SELECT branch_name
    -> FROM account
    -> GROUP BY branch_name
    -> HAVING AVG(balance) >= 700;
+-------------+
| branch_name |
+-------------+
| Brighton    |
| Mianus      |
| Redwood     |
+-------------+
3 rows in set (0.000 sec)
```

9. **Find the names and account number of customers who have the 3 highest balances in their accounts.**

```
SELECT c.customer_name, d.account_number
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN
(
SELECT account_number
FROM account
ORDER BY balance DESC
LIMIT 3
)
AS top_accounts ON d.account_number = top_accounts.account_number;
```

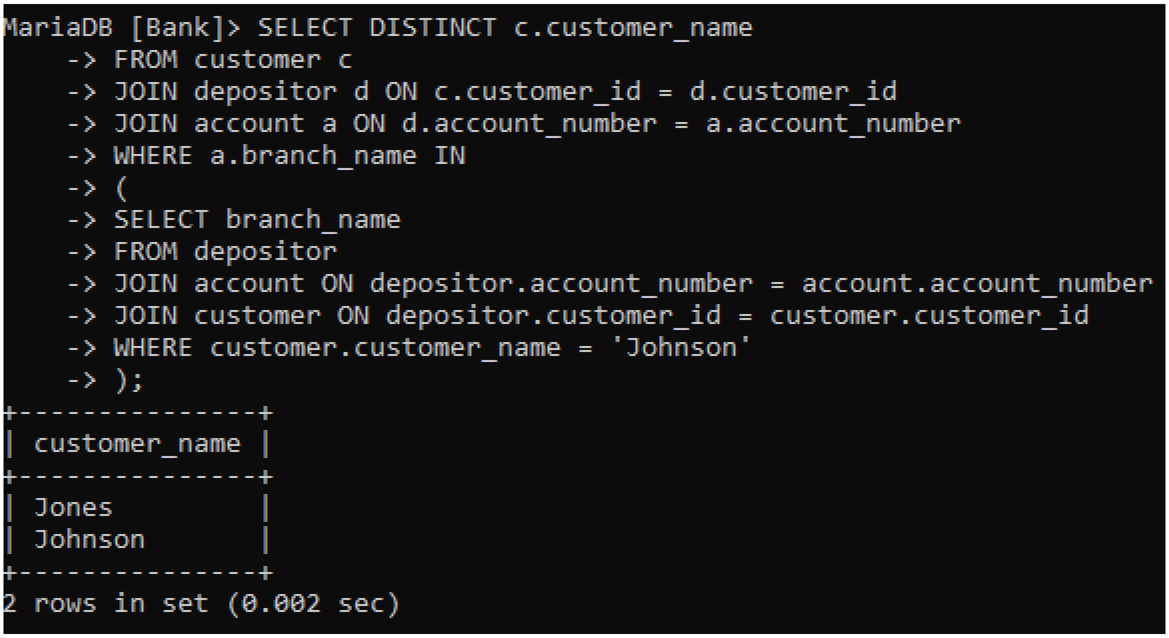


```
MariaDB [Bank]> SELECT c.customer_name, d.account_number
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN
    -> (
    -> SELECT account_number
    -> FROM account
    -> ORDER BY balance DESC
    -> LIMIT 3
    -> )
    -> AS top_accounts ON d.account_number = top_accounts.account_number;
+---------------+----------------+
| customer_name | account_number |
+---------------+----------------+
| Johnson       | A-201          |
| Jones         | A-217          |
| Smith         | A-215          |
+---------------+----------------+
3 rows in set (0.000 sec)
```

Task 7 (LAB Sheet):

1. **Find the names of customers with accounts at a branch where Johnson has an account.**

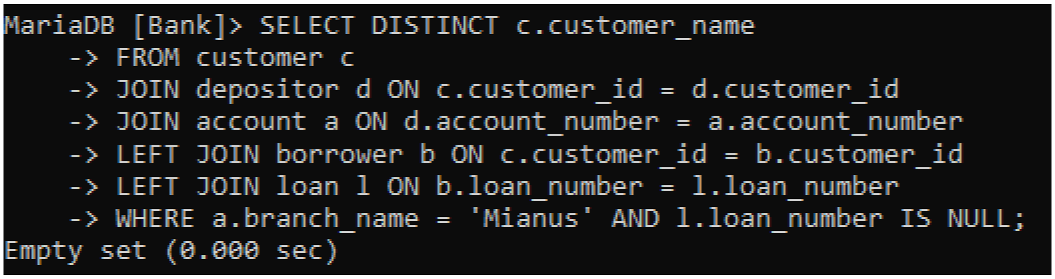
```
SELECT DISTINCT c.customer_name
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN account a ON d.account_number = a.account_number
WHERE a.branch_name IN
(
SELECT branch_name
FROM depositor
JOIN account ON depositor.account_number = account.account_number
JOIN customer ON depositor.customer_id = customer.customer_id
WHERE customer.customer_name = 'Johnson'
);
```



```
MariaDB [Bank]> SELECT DISTINCT c.customer_name
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> WHERE a.branch_name IN
    -> (
    -> SELECT branch_name
    -> FROM depositor
    -> JOIN account ON depositor.account_number = account.account_number
    -> JOIN customer ON depositor.customer_id = customer.customer_id
    -> WHERE customer.customer_name = 'Johnson'
    -> );
+---------------+
| customer_name |
+---------------+
| Jones         |
| Johnson       |
+---------------+
2 rows in set (0.002 sec)
```

2. **Find the names of customers with an account but not a loan at Mianus branch.**

```
SELECT DISTINCT c.customer_name
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN account a ON d.account_number = a.account_number
LEFT JOIN borrower b ON c.customer_id = b.customer_id
LEFT JOIN loan l ON b.loan_number = l.loan_number
WHERE a.branch_name = 'Mianus' AND l.loan_number IS NULL;
```

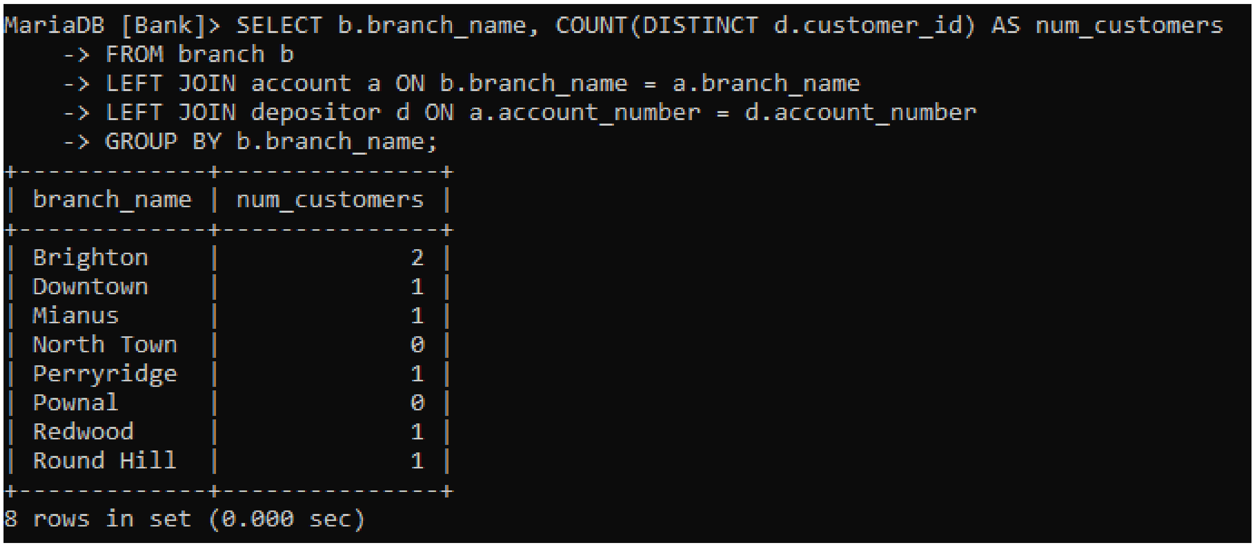


```
MariaDB [Bank]> SELECT DISTINCT c.customer_name
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> LEFT JOIN borrower b ON c.customer_id = b.customer_id
    -> LEFT JOIN loan l ON b.loan_number = l.loan_number
    -> WHERE a.branch_name = 'Mianus' AND l.loan_number IS NULL;
Empty set (0.000 sec)
```

3. **Find the names of each branch and the number of customers having at least one account at that branch.**

```
SELECT b.branch_name, COUNT(DISTINCT d.customer_id) AS num_customers
FROM branch b
LEFT JOIN account a ON b.branch_name = a.branch_name
LEFT JOIN depositor d ON a.account_number = d.account_number
```

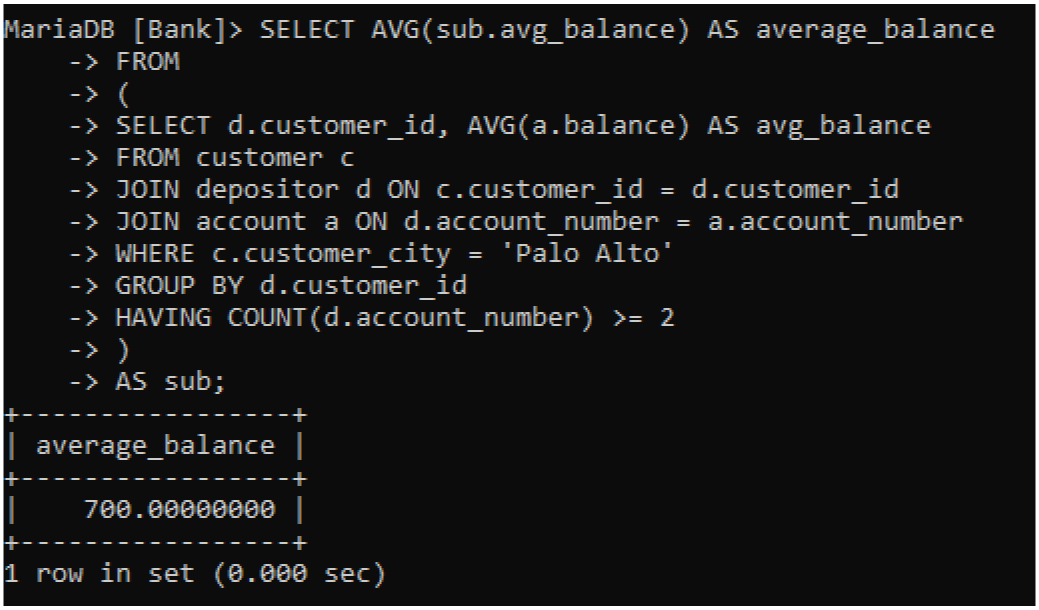
GROUP BY b.branch_name;



```
MariaDB [Bank]> SELECT b.branch_name, COUNT(DISTINCT d.customer_id) AS num_customers
    -> FROM branch b
    -> LEFT JOIN account a ON b.branch_name = a.branch_name
    -> LEFT JOIN depositor d ON a.account_number = d.account_number
    -> GROUP BY b.branch_name;
+-------------+---------------+
| branch_name | num_customers |
+-------------+---------------+
| Brighton    |             2 |
| Downtown    |             1 |
| Mianus      |             1 |
| North Town  |             0 |
| Perryridge  |             1 |
| Pownal      |             0 |
| Redwood     |             1 |
| Round Hill  |             1 |
+-------------+---------------+
8 rows in set (0.000 sec)
```

4. **Find the average balance of all customers in ‘Palo Alto’ having at least 2 accounts**

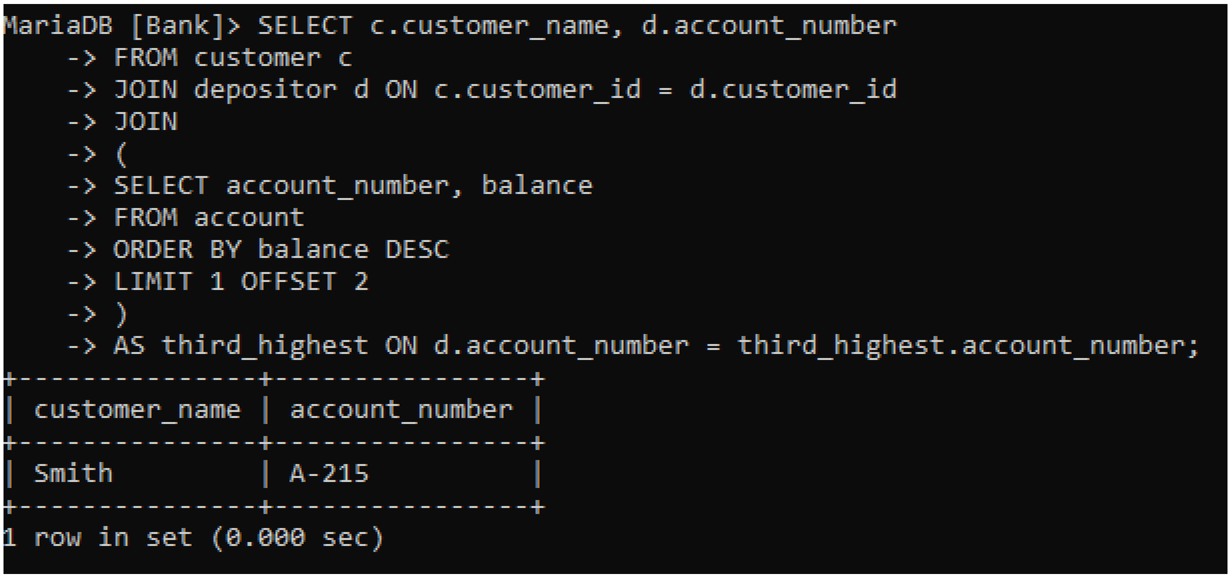
```
SELECT AVG(sub.avg_balance) AS average_balance
FROM
(
SELECT d.customer_id, AVG(a.balance) AS avg_balance
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN account a ON d.account_number = a.account_number
WHERE c.customer_city = 'Palo Alto'
GROUP BY d.customer_id
HAVING COUNT(d.account_number) >= 2
)
AS sub;
```



```
MariaDB [Bank]> SELECT AVG(sub.avg_balance) AS average_balance
    -> FROM
    -> (
    -> SELECT d.customer_id, AVG(a.balance) AS avg_balance
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN account a ON d.account_number = a.account_number
    -> WHERE c.customer_city = 'Palo Alto'
    -> GROUP BY d.customer_id
    -> HAVING COUNT(d.account_number) >= 2
    -> )
    -> AS sub;
+-----------------+
| average_balance |
+-----------------+
|    700.00000000 |
+-----------------+
1 row in set (0.000 sec)
```

5. **Find the name and account number of the customer who has the 3rd highest balance in their account.**

```
SELECT c.customer_name, d.account_number
FROM customer c
JOIN depositor d ON c.customer_id = d.customer_id
JOIN
(
SELECT account_number, balance
FROM account
ORDER BY balance DESC
LIMIT 1 OFFSET 2
)
AS third_highest ON d.account_number = third_highest.account_number;
```



```
MariaDB [Bank]> SELECT c.customer_name, d.account_number
    -> FROM customer c
    -> JOIN depositor d ON c.customer_id = d.customer_id
    -> JOIN
    -> (
    -> SELECT account_number, balance
    -> FROM account
    -> ORDER BY balance DESC
    -> LIMIT 1 OFFSET 2
    -> )
    -> AS third_highest ON d.account_number = third_highest.account_number;
+---------------+----------------+
| customer_name | account_number |
+---------------+----------------+
| Smith         | A-215          |
+---------------+----------------+
1 row in set (0.000 sec)
```