No 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, borrower b, loan l WHERE c.customer_id = b.customer_id and b.loan_number = l.loan_number and l.branch_name = 'Downtown';
| | ```
MariaDB [Bank_22101698]> SELECT c.customer_name, l.loan_number from customer c, borrower b, loan l
    ->  WHERE c.customer_id = b.customer_id and b.loan_number = l.loan_number and l.branch_name = 'Downtown'
+---------------+-------------+
| customer_name | loan_number |
+---------------+-------------+
| Johnson       | L-14        |
| Jones         | L-17        |
| Williams      | L-17        |
+---------------+-------------+
3 rows in set (0.002 sec)
``` |


No 2	Find all the possible pairs of customers who are from the same city. show in the format Customer1, Customer2, City.
	SELECT C1.customer_name as Customer1, C2.customer_name as Customer2, c1.customer_city as City FROM customer C1, customer C2 WHERE C1.customer_city = C2.customer_city AND C1.customer_id != C2.customer_id;
| | ```
MariaDB [Bank_22101698]> SELECT C1.customer_name as Customer1, C2.customer_name as Customer2, c1.customer_city as City
    -> FROM customer C1, customer C2 WHERE C1.customer_city = C2.customer_city AND C1.customer_id != C2.customer_id;
+-----------+-----------+------------+
| Customer1 | Customer2 | City       |
+-----------+-----------+------------+
| Hayes     | Jones     | Harrison   |
| Curry     | Smith     | Rye        |
| Jones     | Hayes     | Harrison   |
| Smith     | Curry     | Rye        |
| Adams     | Lindsay   | Pittsfield |
| Green     | Turner    | Stamford   |
| Lindsay   | Adams     | Pittsfield |
| Turner    | Green     | Stamford   |
+-----------+-----------+------------+
``` |

No 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 as Branch_name, SUM(balance*(4/100)) as Total_Interest FROM account group by branch_name;
	`` MariaDB [Bank_22101698]> SELECT branch_name as Branch_name, SUM(balance*(4/100)) as Total_Interest -> FROM account group by branch_name; +-------------+----------------+	Branch_name	Total_Interest	+-------------+----------------+	Brighton	66.0000		Downtown	20.0000		Mianus	28.0000		Perryridge	16.0000		Redwood	28.0000		Round Hill	14.0000	+-------------+----------------+ 6 rows in set (0.001 sec) ```

No 4	Find account numbers with the highest balances for each city in the database
	SELECT a.account_number FROM account a, customer c, depositor d WHERE a.account_number = d.account_number AND c.customer_id = d.customer_id AND (a.balance, c.customer_city) IN (SELECT MAX(a.balance), c.customer_city FROM account a, depositor d, customer c WHERE a.account_number = d.account_number AND c.customer_id = d.customer_id GROUP BY c.customer_city);
	`` MariaDB [Bank_22101698]> SELECT a.account_number FROM account a, customer c, depositor d -> WHERE a.account_number = d.account_number AND c.customer_id = d.customer_id -> AND (a.balance, c.customer_city) IN (SELECT MAX(a.balance), c.customer_city -> FROM account a, depositor d, customer c -> WHERE a.account_number = d.account_number AND c.customer_id = d.customer_id -> GROUP BY c.customer_city); +----------------+	account_number	+----------------+	A-217		A-215		A-222		A-305		A-201	+----------------+ 5 rows in set (0.001 sec) ```

No 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, c.customer_name FROM customer c, loan l, borrower b WHERE l.loan_number=b.loan_number and b.customer_id = c.customer_id ORDER BY l.amount DESC, l.loan_number DESC limit 5;
| | ```
MariaDB [Bank_22101698]> SELECT l.loan_number, l.amount, c.customer_name FROM
    -> customer c, loan l, borrower b WHERE l.loan_number = b.loan_number and
    -> b.customer_id = c.customer_id ORDER BY l.amount DESC, l.loan_number DESC limit 5;
+-------------+--------+---------------+
| loan_number | 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)
``` |

No 6	Find the names of customers with an account and also a loan at the Perryridge branch.
	SELECT c.customer_name FROM customer c, account a, depositor d, loan l, borrower b WHERE c.customer_id = d.customer_id AND d.customer_id = b.customer_id AND d.account_number = a.account_number AND l.loan_number = b.loan_number AND a.branch_name = 'Perryridge' AND l.branch_name = 'Perryridge';

```
MariaDB [Bank_22101698]> SELECT c.customer_name
    -> FROM customer c, account a, depositor d, loan l, borrower b
    -> WHERE c.customer_id = d.customer_id
    -> AND d.customer_id = b.customer_id
    -> AND d.account_number = a.account_number
    -> AND l.loan_number = b.loan_number
    -> AND a.branch_name = 'Perryridge'
    -> AND l.branch_name = 'Perryridge';
+---------------+
| customer_name |
+---------------+
| Hayes         |
+---------------+
1 row in set (0.018 sec)
```

No 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, loan l, borrower b WHERE c.customer_id = b.customer_id and b.loan_number = l.loan_number GROUP BY b.customer_id HAVING COUNT(*)>=2;

```
MariaDB [Bank_22101698]> SELECT c.customer_name as customer_name, SUM(l.amount) as total_loan
    -> FROM customer c, loan l, borrower b WHERE c.customer_id = b.customer_id
    -> and b.loan_number = l.loan_number GROUP BY b.customer_id HAVING COUNT(*)>=2;
+---------------+------------+
| customer_name | total_loan |
+---------------+------------+
| Smith         |       2900 |
+---------------+------------+
1 row in set (0.001 sec)
```