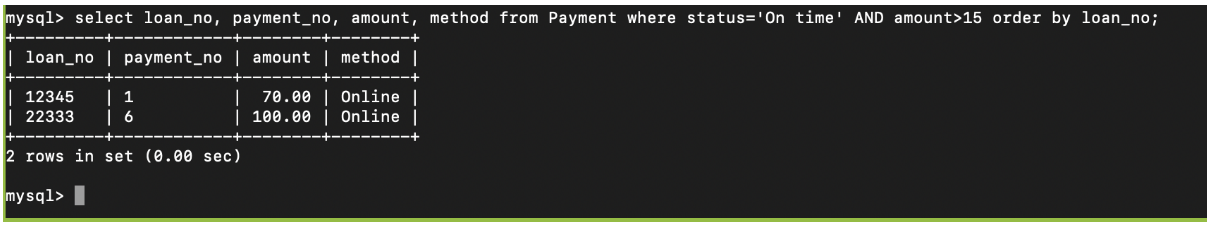
1. List each the loan payment (loan_no, payment_no, amount, method) that was paid on time and the payment was more than $15, order the result by loan_no in ascending order. (the status of the payment is “On time”)

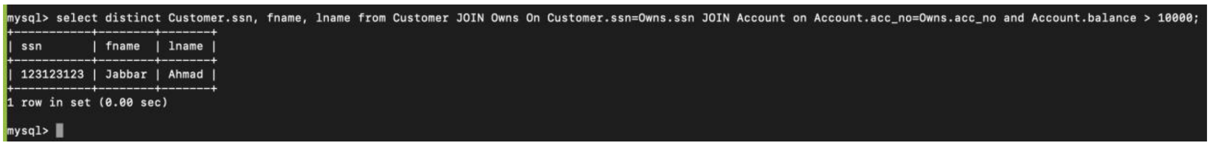
```
select loan_no, payment_no, amount, method from Payment
where status='On time' AND amount>15 order by loan_no;
```



```
mysql> select loan_no, payment_no, amount, method from Payment where status='On time' AND amount>15 order by loan_no;
+---------+------------+--------+--------+
| loan_no | payment_no | amount | method |
+---------+------------+--------+--------+
| 12345   | 1          |  70.00 | Online |
| 22333   | 6          | 100.00 | Online |
+---------+------------+--------+--------+
2 rows in set (0.00 sec)

mysql> 
```

2. List all the customers (ssn, first name, last name) who have at least one account with balance more than $10000. Don’t list a customer twice if he/she has more than one account satisfy the condition.

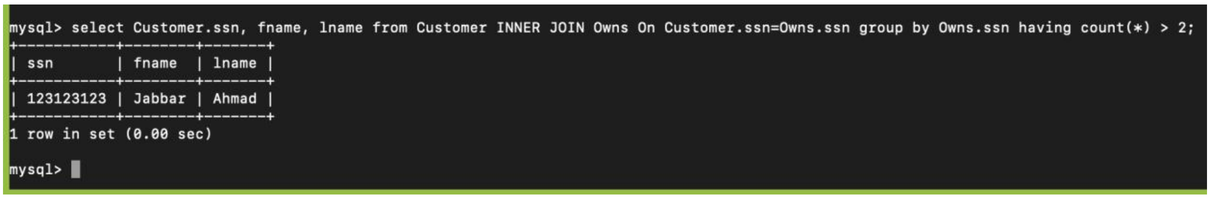
```
select distinct Customer.ssn, fname, lname from Customer
JOIN Owns On Customer.ssn=Owns.ssn JOIN Account on
Account.acc_no=Owns.acc_no and Account.balance > 10000;
```



```
mysql> select distinct Customer.ssn, fname, lname from Customer JOIN Owns On Customer.ssn=Owns.ssn JOIN Account on Account.acc_no=Owns.acc_no and Account.balance > 10000;
+-----------+--------+-------+
| ssn       | fname  | lname |
+-----------+--------+-------+
| 123123123 | Jabbar | Ahmad |
+-----------+--------+-------+
1 row in set (0.00 sec)

mysql> 
```

3. List all customers (ssn, first name, last name) who own more than 2 accounts. If an account is shared, you should also count it in.

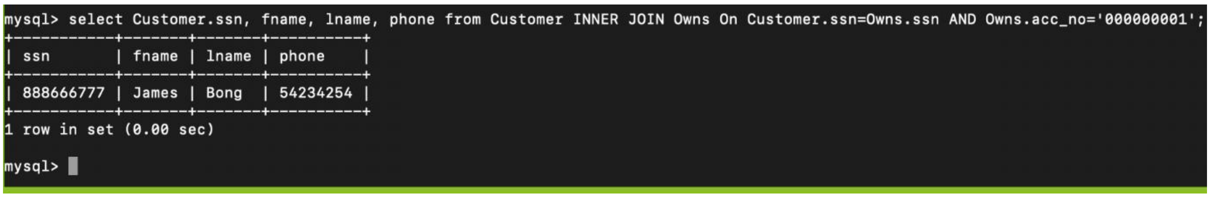
```
select Customer.ssn, fname, lname from Customer INNER JOIN
Owns On Customer.ssn=Owns.ssn group by Owns.ssn having
count(*) > 2;
```



```
mysql> select Customer.ssn, fname, lname from Customer INNER JOIN Owns On Customer.ssn=Owns.ssn group by Owns.ssn having count(*) > 2;
+-----------+--------+-------+
| ssn       | fname  | lname |
+-----------+--------+-------+
| 123123123 | Jabbar | Ahmad |
+-----------+--------+-------+
1 row in set (0.00 sec)

mysql> 
```

4. List all the customers (ssn, first name, last name, phone number) who own account ‘000000001’.

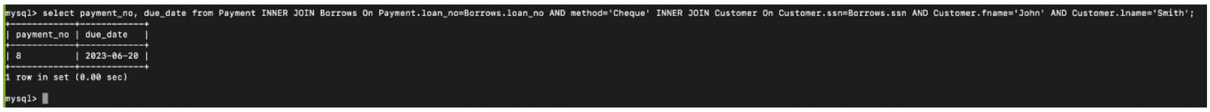
```
select Customer.ssn, fname, lname, phone from Customer INNER
JOIN Owns On Customer.ssn=Owns.ssn AND
Owns.acc_no='000000001';
```



```
mysql> select Customer.ssn, fname, lname, phone from Customer INNER JOIN Owns On Customer.ssn=Owns.ssn AND Owns.acc_no='000000001';
+-----------+-------+-------+----------+
| ssn       | fname | lname | phone    |
+-----------+-------+-------+----------+
| 888666777 | James | Bong  | 54234254 |
+-----------+-------+-------+----------+
1 row in set (0.00 sec)

mysql> █
```

5. List each the loan payment (payment number, due date) of “John Smith” that has been paid by check.

```
select payment_no, due_date from Payment INNER JOIN Borrows
On Payment.loan_no=Borrows.loan_no AND method='Cheque' INNER
JOIN Customer On Customer.ssn=Borrows.ssn AND
Customer.fname='John' AND Customer.lname='Smith';
```



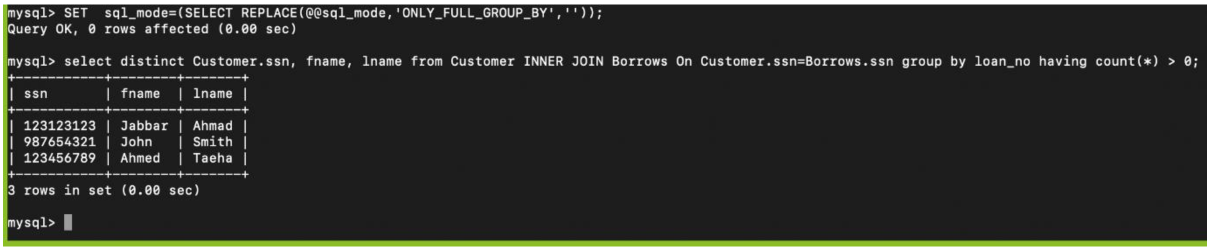
```
mysql> select payment_no, due_date from Payment INNER JOIN Borrows On Payment.loan_no=Borrows.loan_no AND method='Cheque' INNER JOIN Customer On Customer.ssn=Borrows.ssn AND Customer.fname='John' AND Customer.lname='Smith';
+------------+------------+
| payment_no | due_date   |
+------------+------------+
| 8          | 2023-06-20 |
+------------+------------+
1 row in set (0.00 sec)

mysql> █
```

6. List each the customer (ssn, first name, last name) who has at least one loan. List a customer only once.

This query sets sql mode from Full group by to empty. Just to make group by query work.

```
SET sql_mode=(SELECT
REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
```

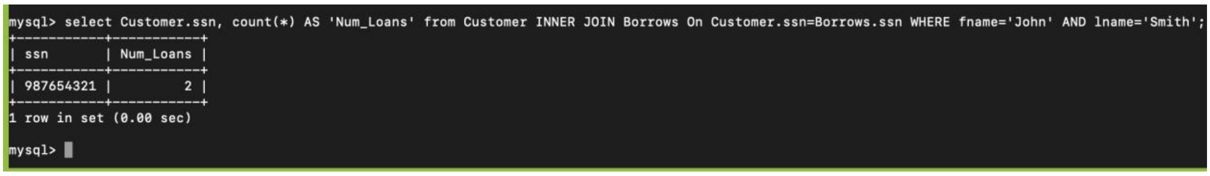
```
select distinct Customer.ssn, fname, lname from Customer
INNER JOIN Borrows On Customer.ssn=Borrows.ssn group by
loan_no having count(*) > 0;
```



```
mysql> SET  sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
Query OK, 0 rows affected (0.00 sec)

mysql> select distinct Customer.ssn, fname, lname from Customer INNER JOIN Borrows On Customer.ssn=Borrows.ssn group by loan_no having count(*) > 0;
+-----------+--------+-------+
| ssn       | fname  | lname |
+-----------+--------+-------+
| 123123123 | Jabbar | Ahmad |
| 987654321 | John   | Smith |
| 123456789 | Ahmed  | Taeha |
+-----------+--------+-------+
3 rows in set (0.00 sec)

mysql> █
```

7. Retrieve all the loans that “John Smith” has. If a loan is shared by him and other customers, you should count it in. Please list John’s ssn and number of loans he has.

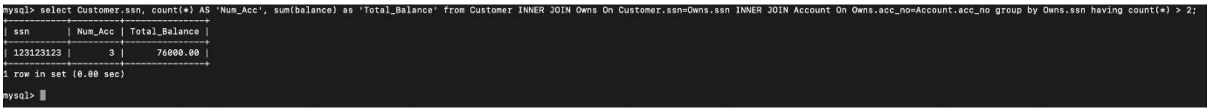
```
select Customer.ssn, count(*) AS 'Num_Loans' from Customer
INNER JOIN Borrows On Customer.ssn=Borrows.ssn WHERE
fname='John' AND lname='Smith';
```



```
mysql> select Customer.ssn, count(*) AS 'Num_Loans' from Customer INNER JOIN Borrows On Customer.ssn=Borrows.ssn WHERE fname='John' AND lname='Smith';
+-----------+-----------+
| ssn       | Num_Loans |
+-----------+-----------+
| 987654321 |         2 |
+-----------+-----------+
1 row in set (0.00 sec)

mysql> 
```

8. Retrieve the total account balance of a customer if a customer has more than 2 accounts. If an account is shared, you should also count it in. List ssn, number of accounts, and total balance.

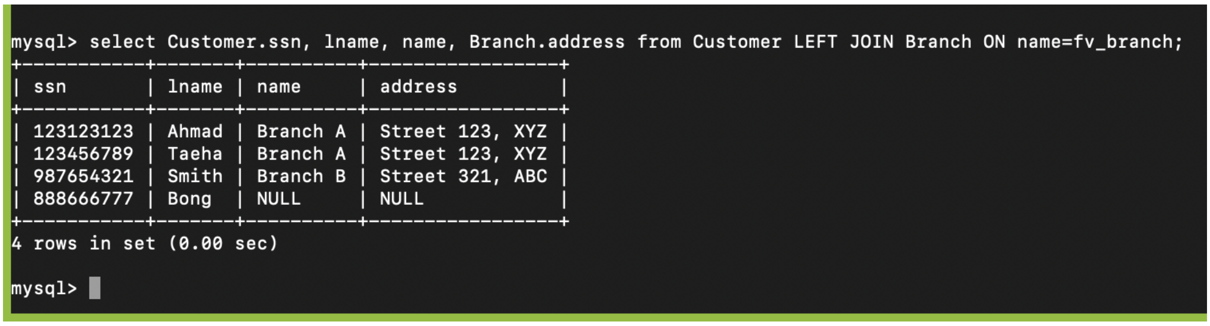
```
select Customer.ssn, count(*) AS 'Num_Acc', sum(balance) as
'Total_Balance' from Customer INNER JOIN Owns On
Customer.ssn=Owns.ssn INNER JOIN Account On
Owns.acc_no=Account.acc_no group by Owns.ssn having count(*)
> 2;
```



```
mysql> select Customer.ssn, count(*) AS 'Num_Acc', sum(balance) as 'Total_Balance' from Customer INNER JOIN Owns On Customer.ssn=Owns.ssn INNER JOIN Account On Owns.acc_no=Account.acc_no group by Owns.ssn having count(*) > 2;
+-----------+---------+---------------+
| ssn       | Num_Acc | Total_Balance |
+-----------+---------+---------------+
| 123123123 |       3 |      76000.00 |
+-----------+---------+---------------+
1 row in set (0.00 sec)

mysql> 
```

9. List each customer with his/her favorite branch (customer SSN, Customer Last Name, Branch Name, Branch address). If a customer does not have a favorite branch, then the branch name and branch address should be NULL.

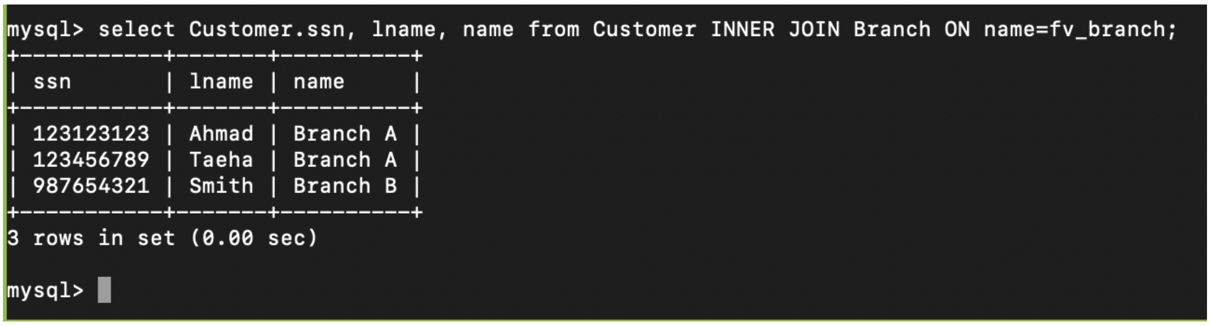
```
select Customer.ssn, lname, name, Branch.address from
Customer LEFT JOIN Branch ON name=fv_branch;
```



```
mysql> select Customer.ssn, lname, name, Branch.address from Customer LEFT JOIN Branch ON name=fv_branch;
+-----------+-------+----------+-----------------+
| ssn       | lname | name     | address         |
+-----------+-------+----------+-----------------+
| 123123123 | Ahmad | Branch A | Street 123, XYZ |
| 123456789 | Taeha | Branch A | Street 123, XYZ |
| 987654321 | Smith | Branch B | Street 321, ABC |
| 888666777 | Bong  | NULL     | NULL            |
+-----------+-------+----------+-----------------+
4 rows in set (0.00 sec)

mysql> 
```

10. For each customer who has a favorite branch, list the customer SSN, customer last name, and the name of his/her favorite branch.

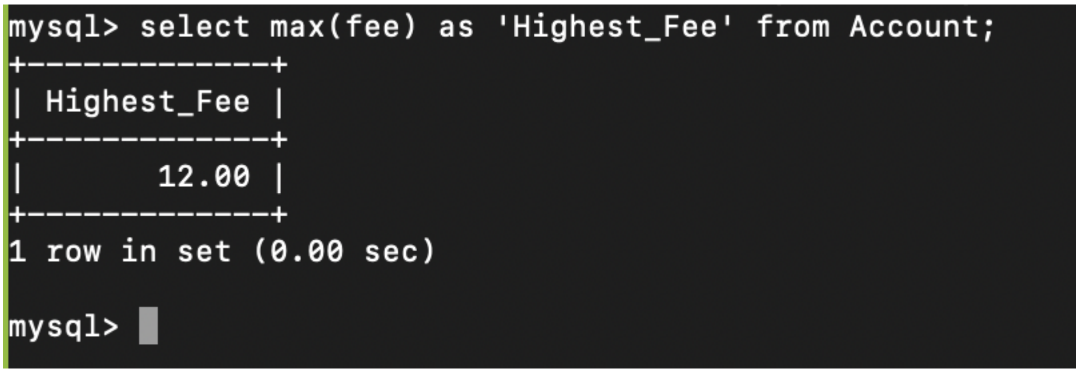
```
select Customer.ssn, lname, name from Customer INNER JOIN
Branch ON name=fv_branch;
```



```
mysql> select Customer.ssn, lname, name from Customer INNER JOIN Branch ON name=fv_branch;
+-----------+-------+----------+
| ssn       | lname | name     |
+-----------+-------+----------+
| 123123123 | Ahmad | Branch A |
| 123456789 | Taeha | Branch A |
| 987654321 | Smith | Branch B |
+-----------+-------+----------+
3 rows in set (0.00 sec)

mysql> 
```

11. Retrieve the highest fee among all accounts.

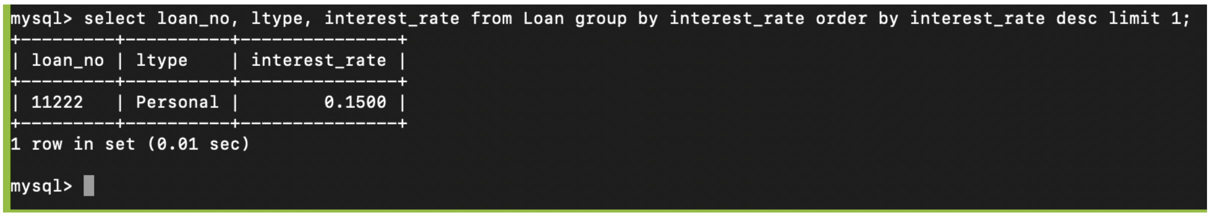
```
select max(fee) as 'Highest_Fee' from Account;
```



```
mysql> select max(fee) as 'Highest_Fee' from Account;
+-------------+
| Highest_Fee |
+-------------+
|       12.00 |
+-------------+
1 row in set (0.00 sec)

mysql> 
```

12. Retrieve the loan that has the highest interest rate among all loans. List the loan number, type, and the interest rate. If there is a tie of highest interest rate, list all loans with the highest interest rate.

```
select loan_no, ltype, interest_rate from loan group by
interest_rate order by interest_rate desc limit 1;
```



```
mysql> select loan_no, ltype, interest_rate from Loan group by interest_rate order by interest_rate desc limit 1;
+---------+----------+---------------+
| loan_no | ltype    | interest_rate |
+---------+----------+---------------+
| 11222   | Personal |        0.1500 |
+---------+----------+---------------+
1 row in set (0.01 sec)

mysql> 
```