National Institute Of Technology Karnataka

IT252 - Database Systems

BANK MANAGEMENT DATABASE SYSTEM

Team:

- Pratham Nayak - 191IT241
- Aprameya Dash - 191IT209
- Suyash Chintawar - 191IT109

Mentor : Dr. Shrutilipi Bhattacharjee

INTRODUCTION

A bank management database system is a database system that allows an enterprise (or user) to store, manage and manipulate data relevant to banks in a database. Database management systems are specifically designed for storage and retrieval of large amounts of data. Further, banks also handle huge amounts of data. So, a bank management database system allows them to store that data, operate on it, and retrieve it when needed, fast enough to satisfy the needs of the customers.

The aim of this project is to create a bank management database system capable of storing different types of data which are commonly stored for banking purposes. Our database contains a total of 7 tables or relational schemas containing data about different topics. Further, as per the requirement of the project, triggers, simple and complex queries, functions, stored procedures etc. have also been created.

The 7 entity sets in the database are:

1. **Customer**(**cid**, first_name, last_name, city, occupation, dob)
2. **Branch**(**bid**, name, city)
3. **Employee**(**eid**, first_name, last_name, salary)
4. **Account**(**aid**, bid, balance, opening_date, type, status)
5. **Transactions**(**tid**, transaction_date, transaction_medium, transaction_type, transaction_amount)
6. **Loan**(**lid**, loan_amount, loan_interest, loan_pending, loan_status)
7. **Repayment**(**rid**, repayment_date, repayment_amount)

The relations associated with the entity sets are:

1. **loan_repayment** : One to Many relation between ‘Loan’ and ‘Repayment’ with total participation of ‘Repayment’ entity set.
2. **works_in**: One to Many relation between ‘Branch’ and ‘Employee’ with total participation of ‘Employee’ entity set.
3. **offers**: One to Many relation between ‘Branch’ and ‘Loan’ with total participation of ‘Loan’ entity set.
4. **borrows**: One to Many relation between ‘Customer’ and ‘Loan’ with total participation of ‘Loan’ entity set.
5. **has**: One to Many relation between ‘Branch’ and ‘Account’ with total participation of ‘Account’ entity set.
6. **creates**: One to Many relation between ‘Customer’ and ‘Account’ with total participation of ‘Account’ entity set.
7. **carried_out_by**: One to Many relation between ‘Account’ and ‘Transaction’ with total participation of ‘Transaction’ entity set.

ER DIAGRAM

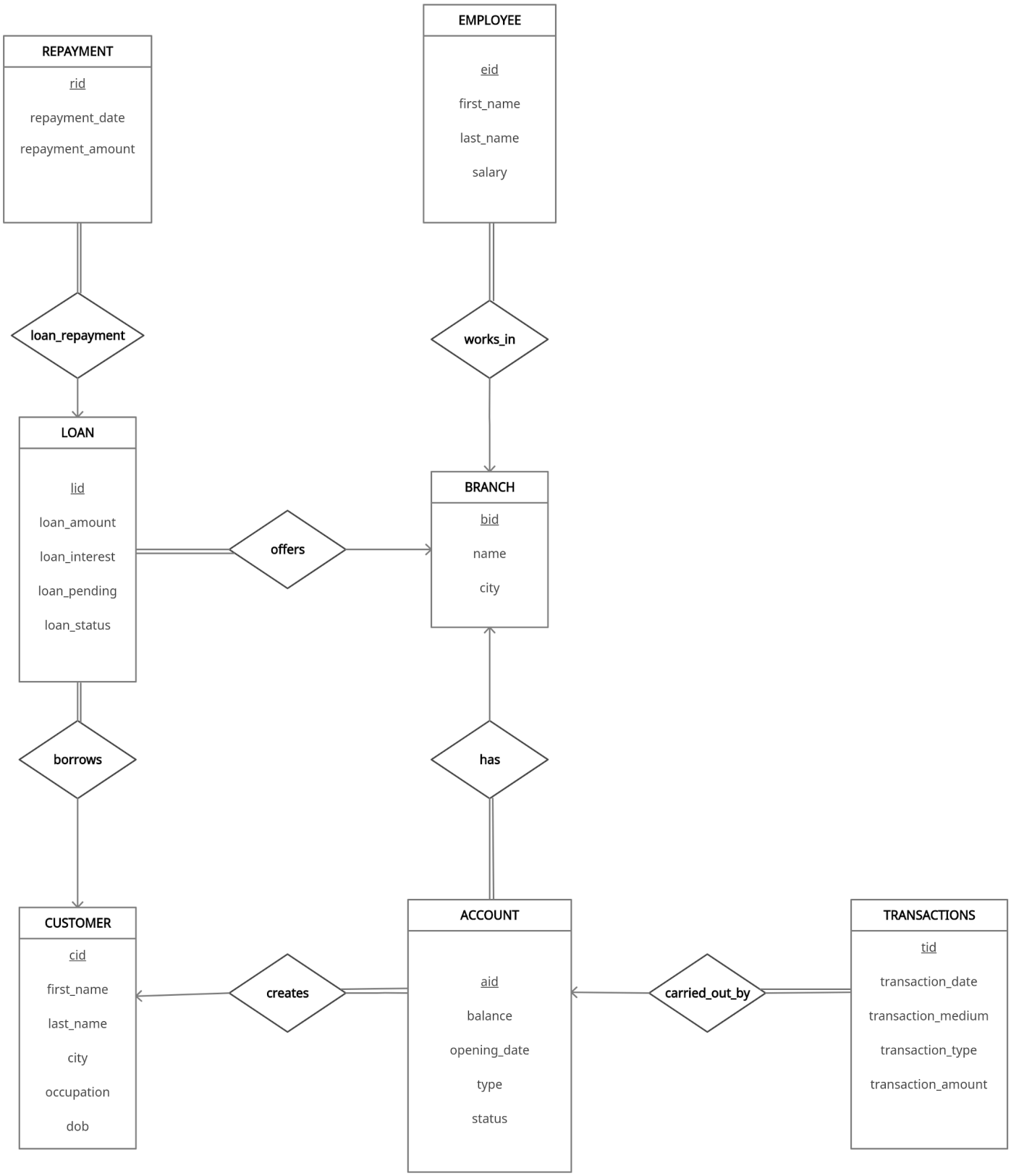
REPAYMENT
rid
repayment_date
repayment_amount
EMPLOYEE
eid
first_name
last_name
salary
loan_repayment
works_in
LOAN
lid
loan_amount
loan_interest
loan_pending
loan_status
offers
BRANCH
bid
name
city
borrows
has
CUSTOMER
cid
first_name
last_name
city
occupation
dob
creates
ACCOUNT
aid
balance
opening_date
type
status
carried_out_by
TRANSACTIONS
tid
transaction_date
transaction_medium
transaction_type
transaction_amount

TABLE CONTENTS

1. **Customer**(**cid**, first_name, last_name, city, occupation, dob)

```
mysql> DESC Customer;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| cid        | varchar(6)  | NO   | PRI | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| city       | varchar(50) | YES  |     | NULL    |       |
| occupation | varchar(50) | YES  |     | NULL    |       |
| dob        | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> SELECT * FROM Customer;
+--------+------------+-----------+---------+------------+------------+
| cid    | first_name | last_name | city    | occupation | dob        |
+--------+------------+-----------+---------+------------+------------+
| C00001 | Ramesh     | Sharma    | Delhi   | Service    | 1986-12-06 |
| C00002 | Avinash    | Minha     | Delhi   | Service    | 1984-10-16 |
| C00003 | Rahul      | Rastogi   | Delhi   | Student    | 1991-09-26 |
| C00004 | Parul      | Gandhi    | Delhi   | Housewife  | 1986-11-03 |
| C00005 | Naveen     | Aedekar   | Mumbai  | Service    | 1986-09-19 |
| C00006 | Chitresh   | Barwe     | Mumbai  | Student    | 2002-11-06 |
| C00007 | Amit       | Borkar    | Mumbai  | Student    | 1991-09-06 |
| C00008 | Nisha      | Damle     | Mumbai  | Service    | 1985-12-03 |
| C00009 | Abhishek   | Dutta     | Kolkata | Service    | 1983-05-22 |
| C00010 | Shankar    | Nair      | Chennai | Service    | 1986-07-12 |
+--------+------------+-----------+---------+------------+------------+
10 rows in set (0.00 sec)
```

2. **Branch**(**bid**, name, city)

```
mysql> DESC Branch;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| bid   | varchar(6)  | NO   | PRI | NULL    |       |
| name  | varchar(50) | YES  |     | NULL    |       |
| city  | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM Branch;
+--------+-----------------------+---------+
| bid    | name                  | city    |
+--------+-----------------------+---------+
| B00001 | Asaf ali road         | Delhi   |
| B00002 | New delhi main branch | Delhi   |
| B00003 | Delhi cantt           | Delhi   |
| B00004 | Jasola                | Delhi   |
| B00005 | Mahim                 | Mumbai  |
| B00006 | Vile parle            | Mumbai  |
| B00007 | Mandvi                | Mumbai  |
| B00008 | Jadavpur              | Kolkata |
| B00009 | Kodambakkam           | Chennai |
+--------+-----------------------+---------+
9 rows in set (0.00 sec)
```

3. **Employee**(**eid**, first_name, last_name, salary)

```
mysql> DESC Employee;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| eid        | varchar(6)  | NO   | PRI | NULL    |       |
| bid        | varchar(6)  | YES  | MUL | NULL    |       |
| first_name | varchar(50) | YES  |     | NULL    |       |
| last_name  | varchar(50) | YES  |     | NULL    |       |
| salary     | double      | YES  |     | NULL    |       |
| job        | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 100000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     |  75000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 125000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 100000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   |  80000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)
```

4. **Account**(**aid**, bid, balance, opening_date, type, status)

```
mysql> DESC Account;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| aid          | varchar(6)  | NO   | PRI | NULL    |       |
| cid          | varchar(6)  | YES  | MUL | NULL    |       |
| bid          | varchar(6)  | YES  | MUL | NULL    |       |
| balance      | double      | YES  |     | NULL    |       |
| opening_date | date        | YES  |     | NULL    |       |
| type         | varchar(10) | YES  |     | NULL    |       |
| status       | varchar(10) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql> SELECT * FROM Account;
+--------+--------+--------+---------+--------------+--------+-----------+
| aid    | cid    | bid    | balance | opening_date | type   | status    |
+--------+--------+--------+---------+--------------+--------+-----------+
| A00001 | C00001 | B00001 |    7000 | 2012-12-15   | Saving | Active    |
| A00002 | C00002 | B00001 |   13000 | 2015-06-12   | Saving | Active    |
| A00003 | C00003 | B00002 |   10000 | 2013-05-17   | Saving | Active    |
| A00004 | C00002 | B00005 |   20000 | 2015-01-27   | Saving | Active    |
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active    |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active    |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated|
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated|
| A00010 | C00004 | B00002 |   20000 | 2017-03-01   | Saving | Active    |
+--------+--------+--------+---------+--------------+--------+-----------+
10 rows in set (0.00 sec)
```

5. **Transactions**(**tid**, transaction_date, transaction_medium, transaction_type, transaction_amount)

```
mysql> DESC Transactions;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| tid                | varchar(6)  | NO   | PRI | NULL    |       |
| aid                | varchar(6)  | YES  | MUL | NULL    |       |
| transaction_date   | date        | YES  |     | NULL    |       |
| transaction_medium | varchar(20) | YES  |     | NULL    |       |
| transaction_type   | varchar(20) | YES  |     | NULL    |       |
| transaction_amount | double      | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Transactions;
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | A00001 | 2013-03-14       | Cheque             | Deposit          |               3000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
12 rows in set (0.00 sec)
```

6. **Loan**(**lid**, loan_amount, loan_interest, loan_pending, loan_status)

```
mysql> DESC Loan;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| lid           | varchar(6)  | NO   | PRI | NULL    |       |
| cid           | varchar(6)  | YES  | MUL | NULL    |       |
| bid           | varchar(6)  | YES  | MUL | NULL    |       |
| loan_amount   | double      | YES  |     | NULL    |       |
| loan_interest | double      | YES  |     | NULL    |       |
| loan_pending  | double      | YES  |     | NULL    |       |
| loan_status   | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Loan;
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       670000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
6 rows in set (0.00 sec)
```

7. **Repayment**(**rid**, repayment_date, repayment_amount)

```
mysql> DESC Repayment;
+------------------+------------+------+-----+---------+-------+
| Field            | Type       | Null | Key | Default | Extra |
+------------------+------------+------+-----+---------+-------+
| rid              | varchar(6) | NO   | PRI | NULL    |       |
| lid              | varchar(6) | YES  | MUL | NULL    |       |
| repayment_data   | date       | YES  |     | NULL    |       |
| repayment_amount | double     | YES  |     | NULL    |       |
+------------------+------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM Repayment;
+--------+--------+----------------+------------------+
| rid    | lid    | repayment_data | repayment_amount |
+--------+--------+----------------+------------------+
| R00001 | L00001 | 2020-12-15     |             5000 |
| R00002 | L00002 | 2021-01-17     |            10000 |
| R00003 | L00003 | 2021-02-19     |            10000 |
| R00004 | L00004 | 2020-11-21     |            15000 |
| R00005 | L00005 | 2021-03-25     |            10000 |
| R00006 | L00006 | 2020-09-28     |            20000 |
+--------+--------+----------------+------------------+
6 rows in set (0.00 sec)
```

SQL QUERIES

SIMPLE QUERIES

1. Write a query to display the details of all transactions which were done by cash.

```
mysql> SELECT *
    -> FROM Transactions
    -> WHERE
    -> transaction_medium = 'Cash';
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
10 rows in set (0.00 sec)
```

2. Write a query to display account number, balance, type of all accounts where the account status is terminated.

```
mysql> SELECT aid, balance, type
    -> FROM Account
    -> WHERE
    -> status = 'Terminated';
+--------+---------+--------+
| aid    | balance | type   |
+--------+---------+--------+
| A00008 |   40000 | Saving |
| A00009 |   30000 | Saving |
+--------+---------+--------+
2 rows in set (0.00 sec)
```

3. Write a query to get the cid,first names,last names of the students who live in Mumbai.

```
mysql> SELECT cid, first_name, last_name
    -> FROM Customer
    -> WHERE
    -> occupation = 'Student'
    -> AND
    -> city='Mumbai';
+--------+------------+-----------+
| cid    | first_name | last_name |
+--------+------------+-----------+
| C00006 | Chitresh   | Barwe     |
| C00007 | Amit       | Borkar    |
+--------+------------+-----------+
2 rows in set (0.00 sec)
```

4. Write a query to display the details of all Employees who have salaries greater than Rs.100000. Print in descending order of their last names.

```
mysql> SELECT *
    -> FROM Employee
    -> WHERE
    -> salary > 100000
    -> ORDER BY last_name
    -> DESC;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00007 | B00006 | Shashank   | Patel     | 125000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
4 rows in set (0.00 sec)
```

5. Write a query to display the account number, balance and account opening dates of all accounts which were created before 15 May 2012.

```
mysql> SELECT aid, balance, opening_date
    -> FROM Account
    -> WHERE
    -> opening_date < '2012-05-15';
+--------+---------+--------------+
| aid    | balance | opening_date |
+--------+---------+--------------+
| A00006 |   50000 | 2010-08-12   |
| A00008 |   40000 | 2009-11-09   |
| A00009 |   30000 | 2008-11-30   |
+--------+---------+--------------+
3 rows in set (0.00 sec)
```

6. Write a query to display all employees who work at a bank branch in Delhi.

```
mysql> SELECT *
    -> FROM Employee
    -> WHERE
    -> bid=ANY(
    ->   SELECT bid FROM Branch
    ->   WHERE city='Delhi'
    -> );
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 100000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
+--------+--------+------------+-----------+--------+---------+
5 rows in set (0.00 sec)
```

7. Write a query to display the id, date, medium and amount of all transactions done by account 'A00001'.

```
mysql> SELECT tid, transaction_date, transaction_medium, transaction_type, transaction_amount
    -> FROM Transactions
    -> WHERE
    -> aid='A00001';
+--------+------------------+--------------------+------------------+--------------------+
| tid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+------------------+--------------------+------------------+--------------------+
| T00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00007 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | 2013-03-14       | Cheque             | Deposit          |               3000 |
| T00009 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
+--------+------------------+--------------------+------------------+--------------------+
6 rows in set (0.00 sec)
```

8. Write a query to display all info of the customers whose first names start with either ‘A’ or ‘N’.

```
mysql> SELECT *
    -> FROM Customer
    -> WHERE
    -> first_name LIKE 'A%'
    -> OR
    -> first_name LIKE 'N%';
+--------+------------+-----------+---------+------------+------------+
| cid    | first_name | last_name | city    | occupation | dob        |
+--------+------------+-----------+---------+------------+------------+
| C00002 | Avinash    | Minha     | Delhi   | Service    | 1984-10-16 |
| C00005 | Naveen     | Aedekar   | Mumbai  | Service    | 1986-09-19 |
| C00007 | Amit       | Borkar    | Mumbai  | Student    | 1991-09-06 |
| C00008 | Nisha      | Damle     | Mumbai  | Service    | 1985-12-03 |
| C00009 | Abhishek   | Dutta     | Kolkata | Service    | 1983-05-22 |
+--------+------------+-----------+---------+------------+------------+
5 rows in set (0.00 sec)
```

9. Write a query to select all the branches which are located in Mumbai. Print their data in ascending order of their names.

```
mysql> SELECT *
    -> FROM Branch
    -> WHERE
    -> city='Mumbai'
    -> ORDER BY
    -> name ASC;
+--------+------------+--------+
| bid    | name       | city   |
+--------+------------+--------+
| B00005 | Mahim      | Mumbai |
| B00007 | Mandvi     | Mumbai |
| B00006 | Vile parle | Mumbai |
+--------+------------+--------+
3 rows in set (0.00 sec)
```

10. Write a query to display the details of all loans which are still pending.

```
mysql> SELECT *
    -> FROM Loan
    -> WHERE
    -> loan_status='pending';
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       670000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
6 rows in set (0.00 sec)
```

COMPLEX QUERIES

1. Write a query to get all info of the customers who have occupations same as the occupation of any of the customers whose first names start with ‘A’.

```
mysql> SELECT *
    -> FROM Customer
    -> WHERE occupation IN(
    ->     SELECT occupation
    ->     FROM Customer
    ->     WHERE first_name LIKE 'A%'
    -> );
+--------+------------+-----------+---------+------------+------------+
| cid    | first_name | last_name | city    | occupation | dob        |
+--------+------------+-----------+---------+------------+------------+
| C00001 | Ramesh     | Sharma    | Delhi   | Service    | 1986-12-06 |
| C00002 | Avinash    | Minha     | Delhi   | Service    | 1984-10-16 |
| C00003 | Rahul      | Rastogi   | Delhi   | Student    | 1991-09-26 |
| C00005 | Naveen     | Aedekar   | Mumbai  | Service    | 1986-09-19 |
| C00006 | Chitresh   | Barwe     | Mumbai  | Student    | 2002-11-06 |
| C00007 | Amit       | Borkar    | Mumbai  | Student    | 1991-09-06 |
| C00008 | Nisha      | Damle     | Mumbai  | Service    | 1985-12-03 |
| C00009 | Abhishek   | Dutta     | Kolkata | Service    | 1983-05-22 |
| C00010 | Shankar    | Nair      | Chennai | Service    | 1986-07-12 |
+--------+------------+-----------+---------+------------+------------+
9 rows in set (0.00 sec)
```

2. Write a query to get Employee details of the employee who has second highest salary amongst all employees.

```
mysql> SELECT *
    -> FROM Employee
    -> WHERE salary = (
    ->     SELECT max(salary)
    ->     FROM Employee
    ->     WHERE salary <> (
    ->         SELECT max(salary)
    ->         FROM Employee
    ->     )
    -> );
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
+--------+--------+------------+-----------+--------+---------+
1 row in set (0.00 sec)
```

3. Select the customers' id,first name,last name and occupation who have accounts which were opened after the year 2010 and balance greater than minimum balance amongst all accounts.

```
mysql> SELECT cid, first_name, last_name, occupation
    -> FROM Customer
    -> WHERE cid IN(
    ->     SELECT cid
    ->     FROM Account
    ->     WHERE opening_date > '2010-12-31'
    ->     AND balance > (
    ->         SELECT min(balance)
    ->         FROM Account
    ->     )
    -> );
+--------+------------+-----------+------------+
| cid    | first_name | last_name | occupation |
+--------+------------+-----------+------------+
| C00002 | Avinash    | Minha     | Service    |
| C00003 | Rahul      | Rastogi   | Student    |
| C00006 | Chitresh   | Barwe     | Student    |
| C00007 | Amit       | Borkar    | Student    |
| C00004 | Parul      | Gandhi    | Housewife  |
+--------+------------+-----------+------------+
5 rows in set (0.00 sec)
```

4. Select info of all types of transactions which have transaction amounts equal to any of the withdrawal amounts.

```
mysql> SELECT *
    -> FROM Transactions
    -> WHERE transaction_amount IN(
    ->     SELECT transaction_amount
    ->     FROM Transactions
    ->     WHERE transaction_type = 'Withdrawal'
    -> );
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
9 rows in set (0.01 sec)
```

5. Write a query to display the customer id, first name, last name and city of all customers which do not have any bank accounts.

```
mysql> SELECT cid, first_name, last_name, city
    -> FROM Customer C
    -> WHERE NOT EXISTS(
    ->     SELECT *
    ->     FROM Account A
    ->     WHERE A.cid = C.cid
    -> );
+--------+------------+-----------+---------+
| cid    | first_name | last_name | city    |
+--------+------------+-----------+---------+
| C00005 | Naveen     | Aedekar   | Mumbai  |
| C00008 | Nisha      | Damle     | Mumbai  |
| C00009 | Abhishek   | Dutta     | Kolkata |
| C00010 | Shankar    | Nair      | Chennai |
+--------+------------+-----------+---------+
4 rows in set (0.00 sec)
```

6. Write a query to select all info of the accounts which have current balance greater than average balance of all other accounts.

```
mysql> SELECT *
    -> FROM Account A1
    -> WHERE balance > (
    ->     SELECT avg(balance)
    ->     FROM Account A2
    ->     WHERE A2.aid <> A1.aid
    -> );
+--------+--------+--------+---------+--------------+--------+------------+
| aid    | cid    | bid    | balance | opening_date | type   | status     |
+--------+--------+--------+---------+--------------+--------+------------+
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active     |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended  |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active     |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated |
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated |
+--------+--------+--------+---------+--------------+--------+------------+
5 rows in set (0.00 sec)
```

7. Write a query to display the customer id, first name, last name and occupation of all customers which have taken a loan.

```
mysql> SELECT cid, first_name, last_name, occupation
    -> FROM Customer C
    -> WHERE EXISTS(
    ->     SELECT *
    ->     FROM Loan L
    ->     WHERE L.cid = C.cid
    -> );
+--------+------------+-----------+------------+
| cid    | first_name | last_name | occupation |
+--------+------------+-----------+------------+
| C00001 | Ramesh     | Sharma    | Service    |
| C00002 | Avinash    | Minha     | Service    |
| C00009 | Abhishek   | Dutta     | Service    |
| C00010 | Shankar    | Nair      | Service    |
+--------+------------+-----------+------------+
4 rows in set (0.00 sec)
```

VIEWS

1. Create a view to display all details of all branches of banks present in Delhi.

```
mysql> DROP VIEW IF EXISTS delhi_branch;
Query OK, 0 rows affected (0.11 sec)

mysql> CREATE VIEW delhi_branch AS
    -> SELECT *
    -> FROM Branch
    -> WHERE city = 'Delhi';
Query OK, 0 rows affected (0.09 sec)

mysql> SELECT *
    -> FROM delhi_branch;
+--------+-----------------------+-------+
| bid    | name                  | city  |
+--------+-----------------------+-------+
| B00001 | Asaf ali road         | Delhi |
| B00002 | New delhi main branch | Delhi |
| B00003 | Delhi cantt           | Delhi |
| B00004 | Jasola                | Delhi |
+--------+-----------------------+-------+
4 rows in set (0.00 sec)
```

2. Create a view to display all details of customers who live in Mumbai.

```
mysql> DROP VIEW IF EXISTS mumbai_customers;
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE VIEW mumbai_customers AS
    -> SELECT *
    -> FROM Customer
    -> WHERE city = 'Mumbai';
Query OK, 0 rows affected (0.09 sec)

mysql> SELECT *
    -> FROM mumbai_customers;
+--------+------------+-----------+--------+------------+------------+
| cid    | first_name | last_name | city   | occupation | dob        |
+--------+------------+-----------+--------+------------+------------+
| C00005 | Naveen     | Aedekar   | Mumbai | Service    | 1986-09-19 |
| C00006 | Chitresh   | Barwe     | Mumbai | Student    | 2002-11-06 |
| C00007 | Amit       | Borkar    | Mumbai | Student    | 1991-09-06 |
| C00008 | Nisha      | Damle     | Mumbai | Service    | 1985-12-03 |
+--------+------------+-----------+--------+------------+------------+
4 rows in set (0.01 sec)
```

3. Create a view to display **aid, balance ,opening_date** of all active bank accounts.

```
mysql> DROP VIEW IF EXISTS active_accounts;
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE VIEW active_accounts AS
    -> SELECT aid, balance, opening_date
    -> FROM Account
    -> WHERE status = 'Active';
Query OK, 0 rows affected (0.18 sec)

mysql> SELECT *
    -> FROM active_accounts;
+--------+---------+--------------+
| aid    | balance | opening_date |
+--------+---------+--------------+
| A00002 |   13000 | 2015-06-12   |
| A00003 |   10000 | 2013-05-17   |
| A00004 |   20000 | 2015-01-27   |
| A00005 |   60000 | 2012-12-17   |
| A00007 |   34000 | 2012-10-02   |
| A00010 |   20000 | 2017-03-01   |
+--------+---------+--------------+
6 rows in set (0.00 sec)
```

4. Create a view to display all details of deposit transactions.

```
mysql> DROP VIEW IF EXISTS deposit_transactions;
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE VIEW deposit_transactions AS
    -> SELECT *
    -> FROM Transactions
    -> WHERE transaction_type = 'Deposit';
Query OK, 0 rows affected (0.08 sec)

mysql> SELECT *
    -> FROM deposit_transactions;
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | A00001 | 2013-03-14       | Cheque             | Deposit          |               3000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
7 rows in set (0.00 sec)
```

5. Create a view to display all details of customers born before the year 1990.

```
mysql> DROP VIEW IF EXISTS dob_before_1990_customers;
Query OK, 0 rows affected (0.16 sec)

mysql> CREATE VIEW dob_before_1990_customers AS
    -> SELECT *
    -> FROM Customer
    -> WHERE dob < '1990-01-01';
Query OK, 0 rows affected (0.09 sec)

mysql> SELECT *
    -> FROM dob_before_1990_customers;
+--------+------------+-----------+---------+------------+------------+
| cid    | first_name | last_name | city    | occupation | dob        |
+--------+------------+-----------+---------+------------+------------+
| C00001 | Ramesh     | Sharma    | Delhi   | Service    | 1986-12-06 |
| C00002 | Avinash    | Minha     | Delhi   | Service    | 1984-10-16 |
| C00004 | Parul      | Gandhi    | Delhi   | Housewife  | 1986-11-03 |
| C00005 | Naveen     | Aedekar   | Mumbai  | Service    | 1986-09-19 |
| C00008 | Nisha      | Damle     | Mumbai  | Service    | 1985-12-03 |
| C00009 | Abhishek   | Dutta     | Kolkata | Service    | 1983-05-22 |
| C00010 | Shankar    | Nair      | Chennai | Service    | 1986-07-12 |
+--------+------------+-----------+---------+------------+------------+
7 rows in set (0.00 sec)
```

STORED PROCEDURES

1. Create a stored procedure to change the account **status** using **aid** and new status as a parameter.

```
mysql> DROP procedure IF EXISTS change_account_status;
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> SELECT * FROM Account;
+--------+--------+--------+---------+--------------+--------+------------+
| aid    | cid    | bid    | balance | opening_date | type   | status     |
+--------+--------+--------+---------+--------------+--------+------------+
| A00001 | C00001 | B00001 |    7000 | 2012-12-15   | Saving | Active     |
| A00002 | C00002 | B00001 |   13000 | 2015-06-12   | Saving | Active     |
| A00003 | C00003 | B00002 |   10000 | 2013-05-17   | Saving | Active     |
| A00004 | C00002 | B00005 |   20000 | 2015-01-27   | Saving | Active     |
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active     |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended  |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active     |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated |
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated |
| A00010 | C00004 | B00002 |   20000 | 2017-03-01   | Saving | Active     |
+--------+--------+--------+---------+--------------+--------+------------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE procedure
    -> change_account_status(IN account_no varchar(6),IN new_status varchar(10))
    -> BEGIN
    ->   UPDATE Account
    ->   SET
    ->   status = new_status
    ->   WHERE
    ->   aid = account_no;
    -> END$$
Query OK, 0 rows affected (0.09 sec)

mysql> DELIMITER ;
mysql> CALL change_account_status('A00001','Terminated');
Query OK, 1 row affected (0.04 sec)

mysql> SELECT * FROM Account;
+--------+--------+--------+---------+--------------+--------+------------+
| aid    | cid    | bid    | balance | opening_date | type   | status     |
+--------+--------+--------+---------+--------------+--------+------------+
| A00001 | C00001 | B00001 |    7000 | 2012-12-15   | Saving | Terminated |
| A00002 | C00002 | B00001 |   13000 | 2015-06-12   | Saving | Active     |
| A00003 | C00003 | B00002 |   10000 | 2013-05-17   | Saving | Active     |
| A00004 | C00002 | B00005 |   20000 | 2015-01-27   | Saving | Active     |
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active     |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended  |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active     |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated |
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated |
| A00010 | C00004 | B00002 |   20000 | 2017-03-01   | Saving | Active     |
+--------+--------+--------+---------+--------------+--------+------------+
10 rows in set (0.00 sec)
```

2. Create a stored procedure to promote or demote an employee using **eid** and new job as a parameter.

```
mysql> DROP procedure IF EXISTS change_employee_designation;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 100000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     |  75000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 125000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 100000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   |  80000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE procedure
    -> change_employee_designation(IN employee_id varchar(6),IN new_job varchar(20))
    -> BEGIN
    ->   UPDATE Employee
    ->   SET
    ->   job = new_job
    ->   WHERE
    ->   eid = employee_id;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> CALL change_employee_designation('E00004','Manager');
Query OK, 1 row affected (0.05 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 100000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Manager |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     |  75000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 125000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 100000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   |  80000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)
```

3. Create a stored procedure to increase the salaries of the employees who belong to a particular job by a bonus amount. Take **job** , **bonus_amount** as parameters to the procedure.

```
mysql> DROP procedure IF EXISTS update_employee_salary;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 100000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     |  75000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 125000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 100000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   |  80000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE procedure
    -> update_employee_salary(IN curr_job varchar(20),IN bonus_amount int)
    -> BEGIN
    ->   UPDATE Employee
    ->   SET
    ->   salary = salary + bonus_amount
    ->   WHERE
    ->   job = curr_job;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> CALL update_employee_salary('Clerk',25000);
Query OK, 5 rows affected (0.04 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 125000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     | 100000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 150000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 125000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   | 105000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)
```

4. Create a stored procedure to display loan repayment details of a given loan using **lid** as a parameter.

```
mysql> DROP procedure IF EXISTS display_loan_repayment;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> SELECT * FROM Repayment;
+--------+--------+----------------+------------------+
| rid    | lid    | repayment_data | repayment_amount |
+--------+--------+----------------+------------------+
| R00001 | L00001 | 2020-12-15     |             5000 |
| R00002 | L00002 | 2021-01-17     |            10000 |
| R00003 | L00003 | 2021-02-19     |            10000 |
| R00004 | L00004 | 2020-11-21     |            15000 |
| R00005 | L00005 | 2021-03-25     |            10000 |
| R00006 | L00006 | 2020-09-28     |            20000 |
+--------+--------+----------------+------------------+
6 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE procedure
    -> display_loan_repayment(IN loan_id varchar(6))
    -> BEGIN
    ->   SELECT *
    ->   FROM Repayment
    ->   WHERE
    ->   lid = loan_id;
    -> END$$
Query OK, 0 rows affected (0.08 sec)

mysql> DELIMITER ;
mysql> CALL display_loan_repayment('L00002');
+--------+--------+----------------+------------------+
| rid    | lid    | repayment_data | repayment_amount |
+--------+--------+----------------+------------------+
| R00002 | L00002 | 2021-01-17     |            10000 |
+--------+--------+----------------+------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

5. Create a stored procedure to display all the transactions done between a time interval using start and end date as parameters.

```
mysql> DROP procedure IF EXISTS display_transactions;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> SELECT * FROM Transactions;
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | A00001 | 2013-03-14       | Cheque             | Deposit          |               3000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
12 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE procedure
    -> display_transactions(IN start_date DATE,IN end_date DATE)
    -> BEGIN
    ->   SELECT *
    ->   FROM Transactions
    ->   WHERE
    ->   transaction_date BETWEEN
    ->   start_date AND end_date;
    -> END$$
Query OK, 0 rows affected (0.12 sec)

mysql> DELIMITER ;
mysql> CALL display_transactions('2013-01-01','2013-02-01');
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

STORED FUNCTIONS

1. Create a function to return the sum of all **loan_amount**s of a branch using **bid** as an input parameter.

```
mysql> DROP function IF EXISTS get_loan_sum;
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> SELECT * FROM Loan;
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       670000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
6 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE function
    -> get_loan_sum(branch_id varchar(6))
    -> RETURNS int DETERMINISTIC
    -> BEGIN
    ->    DECLARE loan_sum int DEFAULT 0;
    ->    SELECT sum(loan_amount)
    ->    INTO loan_sum
    ->    FROM Loan
    ->    WHERE
    ->    bid = branch_id;
    ->    return loan_sum;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> SELECT get_loan_sum('B00002');
+-------------------------+
| get_loan_sum('B00002') |
+-------------------------+
|                  200000 |
+-------------------------+
1 row in set (0.00 sec)
```

2. Create a function to return the number of employees of a branch using **bid** as an input parameter.

```
mysql> DROP function IF EXISTS get_num_employees;
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 125000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     | 100000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 150000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 125000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   | 105000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE function
    -> get_num_employees(branch_id varchar(6))
    -> RETURNS int DETERMINISTIC
    -> BEGIN
    ->    DECLARE num_employees int DEFAULT 0;
    ->    SELECT count(*)
    ->    INTO num_employees
    ->    FROM Employee
    ->    WHERE
    ->    bid = branch_id;
    ->    return num_employees;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> SELECT get_num_employees('B00002');
+-----------------------------+
| get_num_employees('B00002') |
+-----------------------------+
|                           2 |
+-----------------------------+
1 row in set (0.00 sec)
```

3. Create a function to display **transaction_medium** based on **tid** provided as input parameter.

```
mysql> DROP function IF EXISTS get_transaction_medium;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> SELECT * FROM Transactions;
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | A00001 | 2013-03-14       | Cheque             | Deposit          |               3000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
12 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE function
    -> get_transaction_medium(transaction_id varchar(6))
    -> RETURNS varchar(20) DETERMINISTIC
    -> BEGIN
    ->    DECLARE output varchar(20);
    ->    SELECT transaction_medium
    ->    INTO output
    ->    FROM Transactions
    ->    WHERE
    ->    tid = transaction_id;
    ->    return output;
    -> END$$
Query OK, 0 rows affected (0.08 sec)

mysql> DELIMITER ;
mysql> SELECT get_transaction_medium('T00004');
+----------------------------------+
| get_transaction_medium('T00004') |
+----------------------------------+
| Cash                             |
+----------------------------------+
1 row in set (0.00 sec)
```

4. Create a function to display the full name of a customer based on **cid** as input parameters to the function.

```
mysql> DROP function IF EXISTS get_customer_name;
Query OK, 0 rows affected, 1 warning (0.16 sec)

mysql> SELECT * FROM Customer;
+--------+------------+-----------+---------+------------+------------+
| cid    | first_name | last_name | city    | occupation | dob        |
+--------+------------+-----------+---------+------------+------------+
| C00001 | Ramesh     | Sharma    | Delhi   | Service    | 1986-12-06 |
| C00002 | Avinash    | Minha     | Delhi   | Service    | 1984-10-16 |
| C00003 | Rahul      | Rastogi   | Delhi   | Student    | 1991-09-26 |
| C00004 | Parul      | Gandhi    | Delhi   | Housewife  | 1986-11-03 |
| C00005 | Naveen     | Aedekar   | Mumbai  | Service    | 1986-09-19 |
| C00006 | Chitresh   | Barwe     | Mumbai  | Student    | 2002-11-06 |
| C00007 | Amit       | Borkar    | Mumbai  | Student    | 1991-09-06 |
| C00008 | Nisha      | Damle     | Mumbai  | Service    | 1985-12-03 |
| C00009 | Abhishek   | Dutta     | Kolkata | Service    | 1983-05-22 |
| C00010 | Shankar    | Nair      | Chennai | Service    | 1986-07-12 |
+--------+------------+-----------+---------+------------+------------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE function
    -> get_customer_name(customer_id varchar(6))
    -> RETURNS varchar(100) DETERMINISTIC
    -> BEGIN
    ->    DECLARE full_name varchar(100);
    ->    DECLARE fname varchar(50);
    ->    DECLARE lname varchar(50);
    ->    SELECT first_name,last_name
    ->    INTO fname,lname
    ->    FROM Customer
    ->    WHERE
    ->    cid = customer_id;
    ->    SET full_name = concat(fname,' ',lname);
    ->    return full_name;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> SELECT get_customer_name('C00002');
+----------------------------+
| get_customer_name('C00002') |
+----------------------------+
| Avinash Minha              |
+----------------------------+
1 row in set (0.00 sec)
```

5. Create a function that takes an **eid** as input and returns the net salary as output. Net salary = **salary** - tax, if **salary** <= 80000 then tax is 5%, if 80000 < **salary** <= 150000 then tax is 10%, else the tax is 18%

```
mysql> DROP function IF EXISTS get_net_salary;
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> SELECT * FROM Employee;
+--------+--------+------------+-----------+--------+---------+
| eid    | bid    | first_name | last_name | salary | job     |
+--------+--------+------------+-----------+--------+---------+
| E00001 | B00001 | Suresh     | Sharma    | 125000 | Clerk   |
| E00002 | B00002 | Arun       | Verma     | 150000 | Manager |
| E00003 | B00002 | Ramesh     | Tripathi  | 200000 | Manager |
| E00004 | B00003 | Sachin     | Rastogi   |  50000 | Intern  |
| E00005 | B00004 | Pankaj     | Dwivedi   |  50000 | Intern  |
| E00006 | B00005 | Virat      | Kumar     | 100000 | Clerk   |
| E00007 | B00006 | Shashank   | Patel     | 150000 | Clerk   |
| E00008 | B00007 | Arjun      | Kumar     | 125000 | Clerk   |
| E00009 | B00008 | Nitin      | Saxena    | 200000 | Manager |
| E00010 | B00009 | Nishant    | Chandra   | 105000 | Clerk   |
+--------+--------+------------+-----------+--------+---------+
10 rows in set (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE function
    -> get_net_salary(employee_id varchar(6))
    -> RETURNS DOUBLE DETERMINISTIC
    -> BEGIN
    ->    DECLARE tax DOUBLE;
    ->    DECLARE net_salary DOUBLE;
    ->    DECLARE esal DOUBLE;
    ->    SELECT salary
    ->    INTO esal
    ->    FROM Employee
    ->    WHERE
    ->    eid = employee_id;
    ->    IF esal<=80000 THEN
    ->      SET tax = 0.05*esal;
    ->    ELSEIF esal>80000 and esal<=150000 THEN
    ->      SET tax = 0.1*esal;
    ->    ELSEIF esal>150000 THEN
    ->      SET tax=0.18*esal;
    ->    END IF;
    ->    SET net_salary = esal - tax;
    ->    return net_salary;
    -> END$$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> SELECT get_net_salary('E00002');
+--------------------------+
| get_net_salary('E00002') |
+--------------------------+
|                   135000 |
+--------------------------+
1 row in set (0.01 sec)
```

TRIGGERS

1. Create a Trigger to check and update the account **balance** in the **Account** table before inserting a new transaction entry into the **Transaction** table.

```
mysql> DROP TRIGGER IF EXISTS before_transaction_insert;
Query OK, 0 rows affected (0.11 sec)

mysql> DELIMITER $$
mysql> CREATE TRIGGER before_transaction_insert
    -> BEFORE INSERT
    -> ON Transactions
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE account_balance DOUBLE;
    ->     DECLARE account_status VARCHAR(10);
    ->     SELECT balance INTO account_balance FROM Account WHERE aid = new.aid;
    ->     SELECT status INTO account_status FROM Account WHERE aid = new.aid;
    ->     IF account_status = 'Active' THEN
    ->         IF new.transaction_type = 'Deposit' THEN
    ->             UPDATE Account
    ->             SET balance = account_balance + new.transaction_amount
    ->             WHERE aid = new.aid;
    ->         ELSE
    ->             IF account_balance >= new.transaction_amount THEN
    ->                 UPDATE Account
    ->                 SET balance = account_balance - new.transaction_amount
    ->                 WHERE aid = new.aid;
    ->             ELSE
    ->                 SIGNAL SQLSTATE '45000' SET message_text = 'Insufficient Balance';
    ->             END IF;
    ->         END IF;
    ->     ELSE
    ->         SIGNAL SQLSTATE '45000' SET message_text = 'Account Not Active';
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.11 sec)

mysql> DELIMITER ;
mysql> SELECT *
    -> FROM Account;
+--------+--------+--------+---------+--------------+--------+------------+
| aid    | cid    | bid    | balance | opening_date | type   | status     |
+--------+--------+--------+---------+--------------+--------+------------+
| A00001 | C00001 | B00001 |    7000 | 2012-12-15   | Saving | Terminated |
| A00002 | C00002 | B00001 |   13000 | 2015-06-12   | Saving | Active     |
| A00003 | C00003 | B00002 |   10000 | 2013-05-17   | Saving | Active     |
| A00004 | C00002 | B00005 |   20000 | 2015-01-27   | Saving | Active     |
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active     |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended  |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active     |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated |
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated |
| A00010 | C00004 | B00002 |   20000 | 2017-03-01   | Saving | Active     |
+--------+--------+--------+---------+--------------+--------+------------+
10 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Transactions VALUES('T00013','A00002','2013-03-26','Cheque','Deposit',10000);
Query OK, 1 row affected (0.05 sec)

mysql> SELECT *
    -> FROM Transactions;
+--------+--------+------------------+--------------------+------------------+--------------------+
| tid    | aid    | transaction_date | transaction_medium | transaction_type | transaction_amount |
+--------+--------+------------------+--------------------+------------------+--------------------+
| T00001 | A00001 | 2013-01-01       | Cheque             | Deposit          |               2000 |
| T00002 | A00001 | 2013-02-01       | Cash               | Withdrawal       |               1000 |
| T00003 | A00002 | 2013-01-01       | Cash               | Deposit          |               2000 |
| T00004 | A00002 | 2013-02-01       | Cash               | Deposit          |               3000 |
| T00005 | A00007 | 2013-01-11       | Cash               | Deposit          |               7000 |
| T00006 | A00007 | 2013-01-13       | Cash               | Deposit          |               9000 |
| T00007 | A00001 | 2013-03-13       | Cash               | Deposit          |               4000 |
| T00008 | A00001 | 2013-03-14       | Cheque             | Deposit          |               3000 |
| T00009 | A00001 | 2013-03-21       | Cash               | Withdrawal       |               9000 |
| T00010 | A00001 | 2013-03-22       | Cash               | Withdrawal       |               2000 |
| T00011 | A00002 | 2013-03-25       | Cash               | Withdrawal       |               7000 |
| T00012 | A00007 | 2013-03-26       | Cash               | Withdrawal       |               2000 |
| T00013 | A00002 | 2013-03-26       | Cheque             | Deposit          |              10000 |
+--------+--------+------------------+--------------------+------------------+--------------------+
13 rows in set (0.00 sec)

mysql> SELECT *
    -> FROM Account;
+--------+--------+--------+---------+--------------+--------+------------+
| aid    | cid    | bid    | balance | opening_date | type   | status     |
+--------+--------+--------+---------+--------------+--------+------------+
| A00001 | C00001 | B00001 |    7000 | 2012-12-15   | Saving | Terminated |
| A00002 | C00002 | B00001 |   23000 | 2015-06-12   | Saving | Active     |
| A00003 | C00003 | B00002 |   10000 | 2013-05-17   | Saving | Active     |
| A00004 | C00002 | B00005 |   20000 | 2015-01-27   | Saving | Active     |
| A00005 | C00006 | B00006 |   60000 | 2012-12-17   | Saving | Active     |
| A00006 | C00007 | B00007 |   50000 | 2010-08-12   | Saving | Suspended  |
| A00007 | C00007 | B00001 |   34000 | 2012-10-02   | Saving | Active     |
| A00008 | C00001 | B00003 |   40000 | 2009-11-09   | Saving | Terminated |
| A00009 | C00003 | B00007 |   30000 | 2008-11-30   | Saving | Terminated |
| A00010 | C00004 | B00002 |   20000 | 2017-03-01   | Saving | Active     |
+--------+--------+--------+---------+--------------+--------+------------+
10 rows in set (0.00 sec)
```

2. Create a Trigger to set **loan_interest** and **loan_pending** in the **Loan** table before inserting a new loan entry according to the following condition -
    a. If **loan_amount** <= 100000 then **loan_interest** is 5%
    b. If 100000 < **loan_amount** <= 500000 then **loan_interest** is 10%
    c. If **loan_amount** > 500000 then **loan_interest** is 15%

```
mysql> DROP TRIGGER IF EXISTS before_loan_insert;
Query OK, 0 rows affected (0.12 sec)

mysql> DELIMITER $$
mysql> CREATE TRIGGER before_loan_insert
    -> BEFORE INSERT
    -> ON Loan
    -> FOR EACH ROW
    -> BEGIN
    ->     IF new.loan_amount <= 100000 THEN
    ->         SET new.loan_interest = 0.05;
    ->     ELSEIF new.loan_amount <= 500000 THEN
    ->         SET new.loan_interest = 0.10;
    ->     ELSE
    ->         SET new.loan_interest = 0.15;
    ->     END IF;
    ->     SET new.loan_pending = new.loan_amount + new.loan_amount * new.loan_interest;
    -> END $$
Query OK, 0 rows affected (0.09 sec)

mysql> DELIMITER ;
mysql> INSERT INTO Loan VALUES('L00007','C00002','B00002',300000,null,null,'Pending');
Query OK, 1 row affected (0.06 sec)

mysql> SELECT *
    -> FROM Loan;
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       670000 | Pending     |
| L00007 | C00002 | B00002 |      300000 |           0.1 |       330000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
7 rows in set (0.00 sec)
```

3. Create a Trigger to check and update **loan_pending** and **loan_status** in the **Loan** table before inserting a new repayment entry into the **Repayment** table.

```
mysql> DROP TRIGGER IF EXISTS before_repayment_insert;
Query OK, 0 rows affected (0.13 sec)

mysql> DELIMITER $$
mysql> CREATE TRIGGER before_repayment_insert
    -> BEFORE INSERT
    -> ON Repayment
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE pending DOUBLE;
    ->     SELECT loan_pending INTO pending FROM Loan WHERE lid = new.lid;
    ->     IF new.repayment_amount < pending THEN
    ->         UPDATE Loan
    ->         SET loan_pending = pending - new.repayment_amount
    ->         WHERE lid = new.lid;
    ->     ELSEIF new.repayment_amount = pending THEN
    ->         UPDATE Loan
    ->         SET loan_pending = 0, loan_status = 'Payed'
    ->         WHERE lid = new.lid;
    ->     ELSE
    ->         SIGNAL SQLSTATE '45000' SET message_text = 'Excess Amount Repayed';
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.09 sec)

mysql> DELIMITER ;
mysql> SELECT *
    -> FROM Loan;
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       670000 | Pending     |
| L00007 | C00002 | B00002 |      300000 |           0.1 |       330000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
7 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Repayment VALUES('R00007','L00006','2020-10-28',15000);
Query OK, 1 row affected (0.04 sec)

mysql> SELECT *
    -> FROM Repayment;
+--------+--------+----------------+------------------+
| rid    | lid    | repayment_data | repayment_amount |
+--------+--------+----------------+------------------+
| R00001 | L00001 | 2020-12-15     |             5000 |
| R00002 | L00002 | 2021-01-17     |            10000 |
| R00003 | L00003 | 2021-02-19     |            10000 |
| R00004 | L00004 | 2020-11-21     |            15000 |
| R00005 | L00005 | 2021-03-25     |            10000 |
| R00006 | L00006 | 2020-09-28     |            20000 |
| R00007 | L00006 | 2020-10-28     |            15000 |
+--------+--------+----------------+------------------+
7 rows in set (0.01 sec)

mysql> SELECT *
    -> FROM Loan;
+--------+--------+--------+-------------+---------------+--------------+-------------+
| lid    | cid    | bid    | loan_amount | loan_interest | loan_pending | loan_status |
+--------+--------+--------+-------------+---------------+--------------+-------------+
| L00001 | C00001 | B00001 |      100000 |          0.05 |       100000 | Pending     |
| L00002 | C00002 | B00002 |      200000 |           0.1 |       210000 | Pending     |
| L00003 | C00009 | B00008 |      400000 |           0.1 |       430000 | Pending     |
| L00004 | C00010 | B00009 |      500000 |           0.1 |       535000 | Pending     |
| L00005 | C00001 | B00003 |      600000 |          0.15 |       680000 | Pending     |
| L00006 | C00002 | B00001 |      600000 |          0.15 |       655000 | Pending     |
| L00007 | C00002 | B00002 |      300000 |           0.1 |       330000 | Pending     |
+--------+--------+--------+-------------+---------------+--------------+-------------+
7 rows in set (0.00 sec)
```