1. Write a procedure that counts the number of account numbers having fixed deposit facility in their respective branches. [5 points]

```
mysql> CREATE PROCEDURE ACC_COUNT()
    -> BEGIN
    -> SELECT COUNT(ACC_NO) FROM ACCOUNT INNER JOIN ASSETS ON
ACCOUNT.BR_NAME = ASSETS.BR_NAME WHERE
ASSETS.FACILITIES="fixeddeposit";
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL ACC_COUNT();
    -> $$
+---------------+
| COUNT(ACC_NO) |
+---------------+
|             6 |
+---------------+
1 row in set (0.29 sec)
```

2. Write a procedure that prints the name of the customers who have borrowed a house loan of more than a certain "amount". This "amount" will be given by the user. [5 points]

```
mysql> CREATE PROCEDURE CUST_LOAN(IN amount INT)
    -> BEGIN
    -> SELECT DISTINCT C.NAME FROM CUSTOMER C INNER JOIN
BORROWER B ON C.C_ID = B.CUST_ID INNER JOIN LOAN L ON L.LN_NO =
B.LOAN_NO WHERE L.LN_NO LIKE "%hou%" AND L.AMOUNT > amount;
    -> END$$

mysql> CALL CUST_LOAN(1000000)$$
+----------+
| NAME     |
+----------+
| himani   |
| shankar  |
| saritha  |
| sachitra |
+----------+
4 rows in set (0.00 sec)
```

3. Write a procedure that finds the activity of the customers that have a bank account. Take "gender" as input to the procedure. A customer is said to be active if he/she accesses his/her account after "28 May, 2012", otherwise said to be inactive. [7 points]

```
mysql> CREATE PROCEDURE ACTIVITY(IN gender enum('f','m'))
    -> BEGIN
    -> SELECT COUNT(*) AS ACTIVE_CUSTOMERS FROM CUSTOMER WHERE C_ID IN (SELECT DISTINCT C.C_ID FROM CUSTOMER C INNER JOIN DEPOSITOR D ON C.C_ID = D.CUST_ID WHERE ACCESS_DT > '2012-05-28' and C.GENDER = gender);
    -> SELECT COUNT(*) AS INACTIVE_CUSTOMERS FROM CUSTOMER WHERE C_ID IN (SELECT DISTINCT C.C_ID FROM CUSTOMER C INNER JOIN DEPOSITOR D ON C.C_ID = D.CUST_ID WHERE ACCESS_DT < '2012-05-28' and C.GENDER = gender);
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL ACTIVITY('m')$$
+------------------+
| ACTIVE_CUSTOMERS |
+------------------+
|                0 |
+------------------+
1 row in set (0.00 sec)

+--------------------+
| INACTIVE_CUSTOMERS |
+--------------------+
|                  1 |
+--------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL ACTIVITY('f')$$
+------------------+
| ACTIVE_CUSTOMERS |
+------------------+
|                2 |
+------------------+
1 row in set (0.00 sec)

+--------------------+
| INACTIVE_CUSTOMERS |
+--------------------+
|                 3 |
+-------------------+
1 row in set (0.00 sec)

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

4. Write a procedure to find the second largest sum of all the loans given by a branch. [8 points]

```
mysql> create procedure SUMLOAN()
    -> BEGIN
    -> CREATE TABLE LOAN_SUM(BR_NAME VARCHAR(10), AMOUNT_SUM DECIMAL(10,0));
    -> INSERT INTO LOAN_SUM SELECT BR_NAME, SUM(AMOUNT) AS AMOUNT_SUM FROM LOAN GROUP BY BR_NAME;
    -> SELECT MAX(AMOUNT_SUM) FROM LOAN_SUM WHERE AMOUNT_SUM NOT IN (SELECT MAX(AMOUNT_SUM) FROM LOAN_SUM);
    -> DROP TABLE LOAN_SUM;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL SUMLOAN()$$
+-----------------+
| MAX(AMOUNT_SUM) |
+-----------------+
|         9000000 |
+-----------------+
1 row in set (0.51 sec)
```

5. Find the calatalan number of the smallest payment made for any personal loan. Divide the payment by 1000 before finding the catalan number.
Catalan number = choose(2n,n) / (n+1). [7 points]

```
mysql> create procedure fact(in x int, out result int)
    -> begin
    -> DECLARE i INT;
    -> SET result = 1;
    -> SET i = 1;
    ->
    -> WHILE i <= x DO
    -> SET result = result * i;
    -> SET i = i + 1;
    -> END WHILE;
    -> end$$
Query OK, 0 rows affected (0.00 sec)

mysql> create procedure cat()
    -> begin
    -> declare var decimal;
    -> select min(AMOUNT) into var from PAYMENT where L_NO like "per%";
    -> set var = var/1000;
    -> call fact(2*var, @ans1);
    -> call fact(var, @ans2);
    -> call fact(var+1, @ans3);
    -> select @ans1/(@ans2*@ans3) as Catalan;
    -> end$$
Query OK, 0 rows affected (0.00 sec)

mysql> call cat()$$
+---------------------+
|             Catalan |
+---------------------+
|             42.0000 |
+---------------------+
1 row in set (0.00 sec)

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

6. Find the type(i.e., car loan or home loan or personal loan) and the cities of their respective branches for those loans whose some of the payment has been made.
[Hints: Iterate through the payment table and find the type of the loans and the cities.
Limit x,y : outputs first y rows starting from the row number x.
Locate(substring, string): find the first location of the substring in the string. The staring position is 1. If substring is not present, it gives 0 as output.
Eq: Locate (‘at’,’attitude’) = 1] [13 points]

```
mysql> CREATE PROCEDURE LOAN_INFO()
    -> BEGIN
    -> DECLARE n INT DEFAULT 0;
    -> DECLARE i INT DEFAULT 0;
    -> DECLARE LOAN_NO VARCHAR(20);
    -> SELECT COUNT(*) INTO n FROM PAYMENT;
    -> SET i = 0;
    -> WHILE i < n DO
    -> SELECT L_NO INTO LOAN_NO FROM PAYMENT LIMIT i,1;
    -> IF LOCATE('car',LOAN_NO) > 0 THEN
    -> SELECT 'CAR LOAN' AS LOAN_TYPE ;
    -> ELSEIF LOCATE('hou',LOAN_NO) > 0 THEN
    -> SELECT 'HOUSE LOAN' AS LOAN_TYPE;
    -> ELSE
    -> SELECT 'PERSONAL LOAN' AS LOAN_TYPE;
    -> END IF;
    -> SELECT B.CITY FROM BRANCH B INNER JOIN LOAN L ON
B.BRN_NAME = L.BR_NAME WHERE L.LN_NO = LOAN_NO;
    -> SET i = i+1;
    -> END WHILE;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL LOAN_INFO()$$
+-----------+
| LOAN_TYPE |
+-----------+
| CAR LOAN  |
+-----------+
1 row in set (0.00 sec)

+--------------+
| CITY         |
+--------------+
| Vijayanagaram |
+--------------+
1 row in set (0.00 sec)

+-----------+
| LOAN_TYPE |
+-----------+
| CAR LOAN  |
+-----------+
1 row in set (0.00 sec)

+--------------+
| CITY         |
+--------------+
| Vijayanagaram |
+--------------+
1 row in set (0.00 sec)

+-----------+
| LOAN_TYPE |
+-----------+
| CAR LOAN  |
+-----------+
1 row in set (0.00 sec)

+---------------+
| CITY          |
+---------------+
| Vijayanagaram |
+---------------+
1 row in set (0.00 sec)

+-----------+
| LOAN_TYPE |
+-----------+
| CAR LOAN  |
+-----------+
1 row in set (0.00 sec)

+------------+
| CITY       |
+------------+
| Srikakulam |
+------------+
1 row in set (0.00 sec)

+------------+
| LOAN_TYPE  |
+------------+
| HOUSE LOAN |
+------------+
1 row in set (0.00 sec)

+------------+
| CITY       |
+------------+
| Vijayawada |
+------------+
1 row in set (0.00 sec)

+------------+
| LOAN_TYPE  |
+------------+
| HOUSE LOAN |
+------------+
1 row in set (0.00 sec)

+------------+
| CITY       |
+------------+
| Vijayawada |
+------------+
1 row in set (0.00 sec)

+------------+
| LOAN_TYPE  |
+------------+
| HOUSE LOAN |
+------------+
1 row in set (0.00 sec)

+------------+
| CITY       |
+------------+
| Vijayawada |
+------------+
1 row in set (0.00 sec)

+---------------+
| LOAN_TYPE     |
+---------------+
| PERSONAL LOAN |
+---------------+
1 row in set (0.00 sec)

+-------+
| CITY  |
+-------+
| Eluru |
+-------+
1 row in set (0.00 sec)

+---------------+
| LOAN_TYPE     |
+---------------+
| PERSONAL LOAN |
+---------------+
1 row in set (0.01 sec)

+-------+
| CITY  |
+-------+
| Eluru |
+-------+
1 row in set (0.01 sec)

+---------------+
| LOAN_TYPE     |
+---------------+
| PERSONAL LOAN |
+---------------+
1 row in set (0.01 sec)

+----------+
| CITY     |
+----------+
| Kakinada |
+----------+
1 row in set (0.01 sec)
```