DBMS – LAB -09

NAME : RAHUL VARMA

ROLL NO: S20200010212

SECTION: C

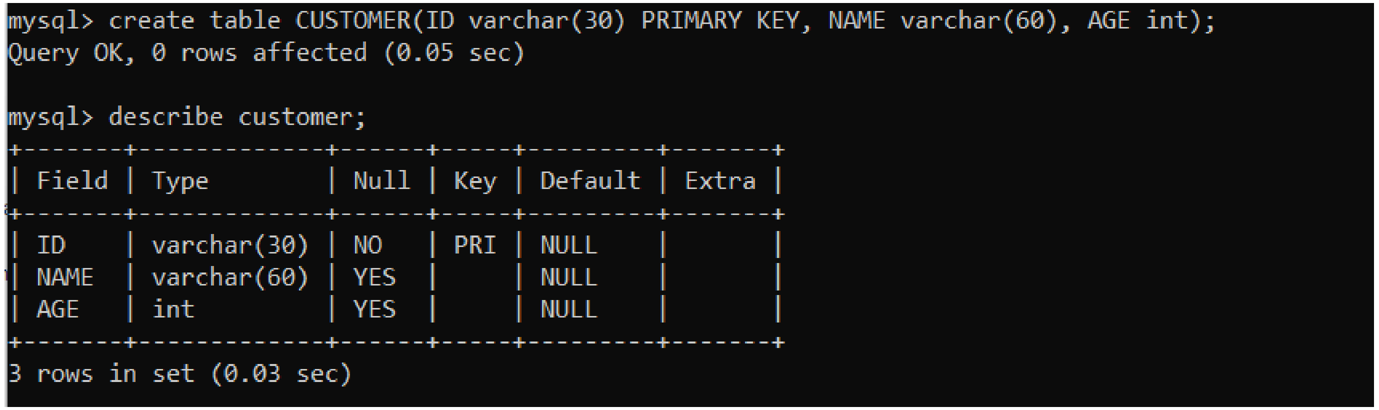
TASK: (LAB EXERCISES)

SQL COMMANDS:

Topic: STORED FUNCTIONS AND ERROR HANDLING

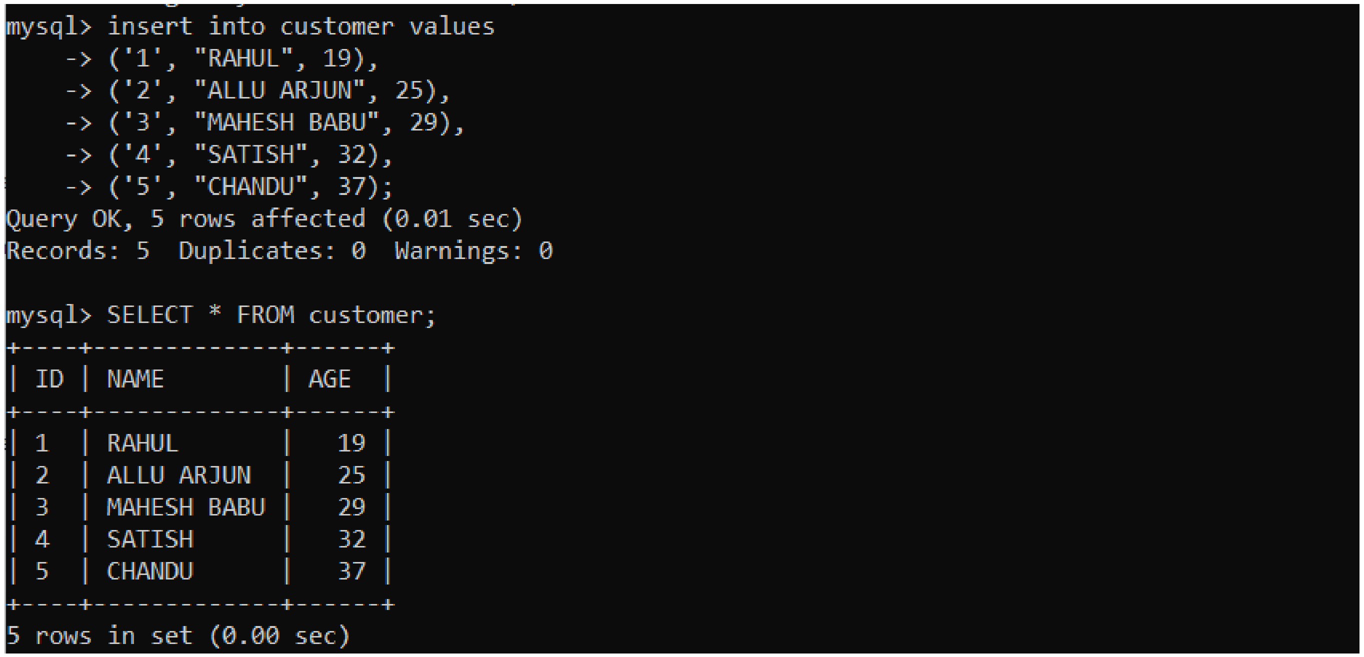
1. Create a function that returns the **customer occupation** based on the **age**.
Age>=35, Carpenter
20<age>=30< Actor
30<age>35 Engineer
Below 20 years student

CREATING A TABLE:



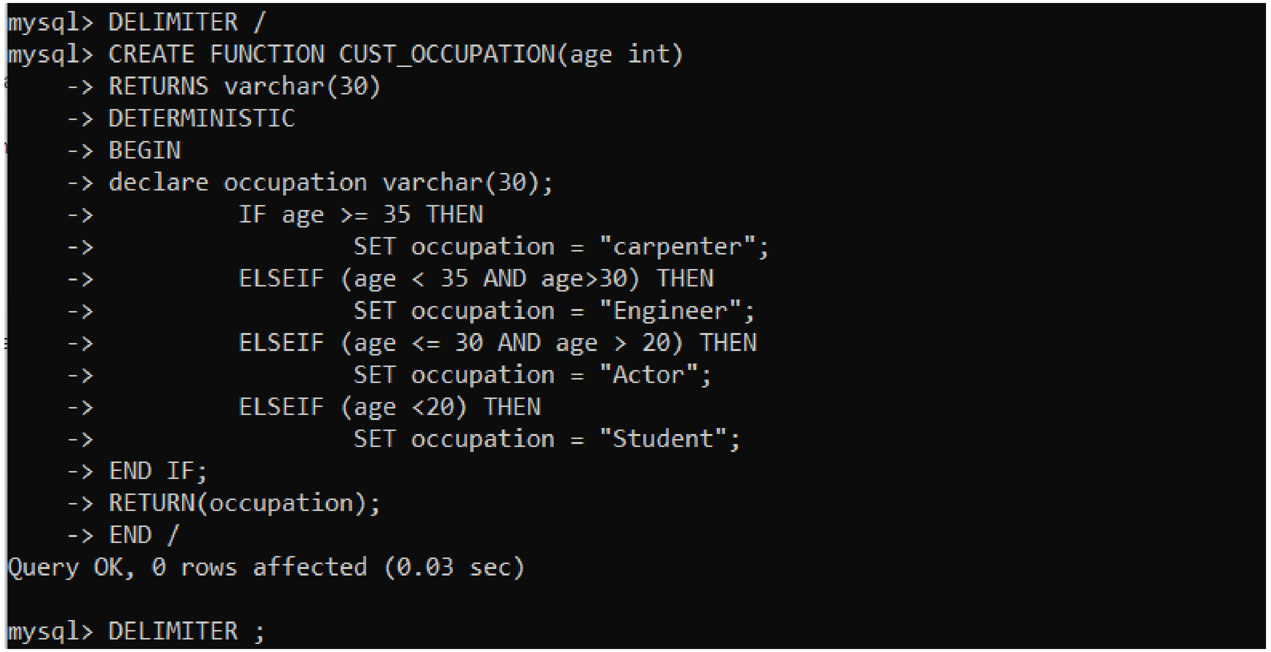
```
mysql> create table CUSTOMER(ID varchar(30) PRIMARY KEY, NAME varchar(60), AGE int);
Query OK, 0 rows affected (0.05 sec)

mysql> describe customer;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ID    | varchar(30) | NO   | PRI | NULL    |       |
| NAME  | varchar(60) | YES  |     | NULL    |       |
| AGE   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

INSERTING VALUES IN THE CUSTOMER TABLE:



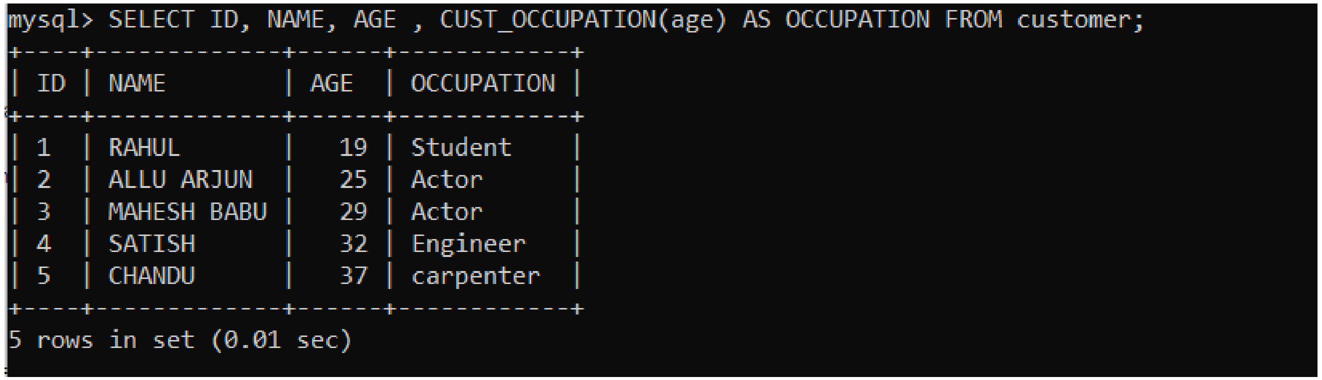
```
mysql> insert into customer values
    -> ('1', "RAHUL", 19),
    -> ('2', "ALLU ARJUN", 25),
    -> ('3', "MAHESH BABU", 29),
    -> ('4', "SATISH", 32),
    -> ('5', "CHANDU", 37);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM customer;
+----+-------------+------+
| ID | NAME        | AGE  |
+----+-------------+------+
| 1  | RAHUL       |   19 |
| 2  | ALLU ARJUN  |   25 |
| 3  | MAHESH BABU |   29 |
| 4  | SATISH      |   32 |
| 5  | CHANDU      |   37 |
+----+-------------+------+
5 rows in set (0.00 sec)
```

CREATING FUNCTION FOR STORED FUNCTIONS:



```
mysql> DELIMITER /
mysql> CREATE FUNCTION CUST_OCCUPATION(age int)
    -> RETURNS varchar(30)
    -> DETERMINISTIC
    -> BEGIN
    -> declare occupation varchar(30);
    ->          IF age >= 35 THEN
    ->                  SET occupation = "carpenter";
    ->          ELSEIF (age < 35 AND age>30) THEN
    ->                  SET occupation = "Engineer";
    ->          ELSEIF (age <= 30 AND age > 20) THEN
    ->                  SET occupation = "Actor";
    ->          ELSEIF (age <20) THEN
    ->                  SET occupation = "Student";
    -> END IF;
    -> RETURN(occupation);
    -> END /
Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER ;
```

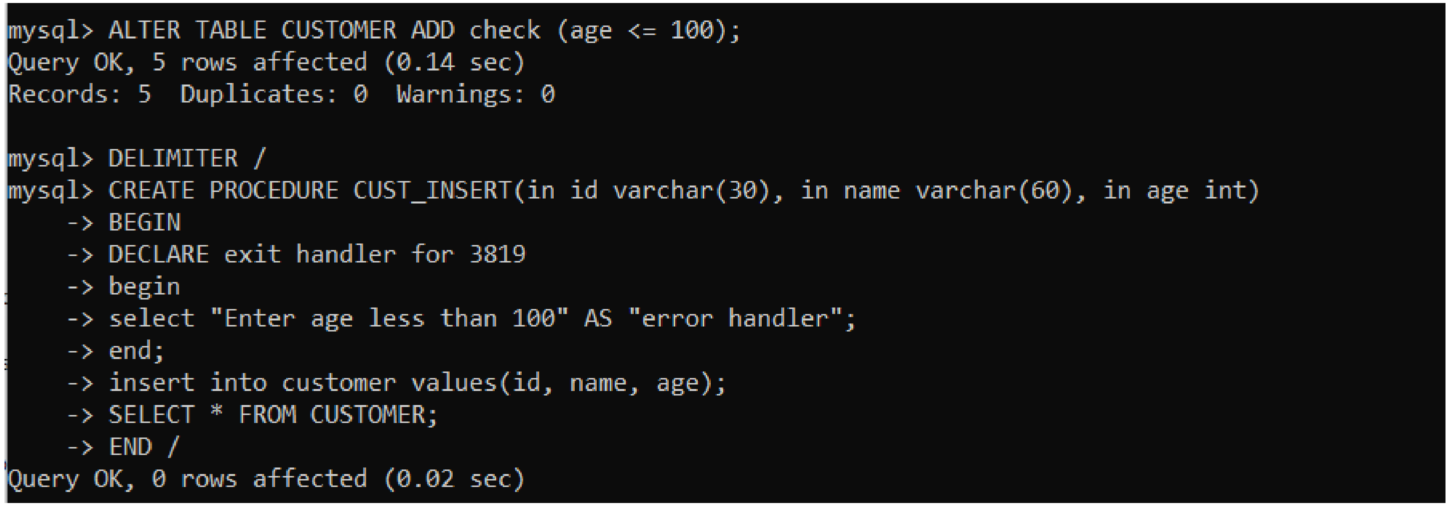
USING THE CREATED FUNCTION:



```
mysql> SELECT ID, NAME, AGE , CUST_OCCUPATION(age) AS OCCUPATION FROM customer;
+----+-------------+-----+------------+
| ID | NAME        | AGE | OCCUPATION |
+----+-------------+-----+------------+
| 1  | RAHUL       |  19 | Student    |
| 2  | ALLU ARJUN  |  25 | Actor      |
| 3  | MAHESH BABU |  29 | Actor      |
| 4  | SATISH      |  32 | Engineer   |
| 5  | CHANDU      |  37 | carpenter  |
+----+-------------+-----+------------+
5 rows in set (0.01 sec)
```

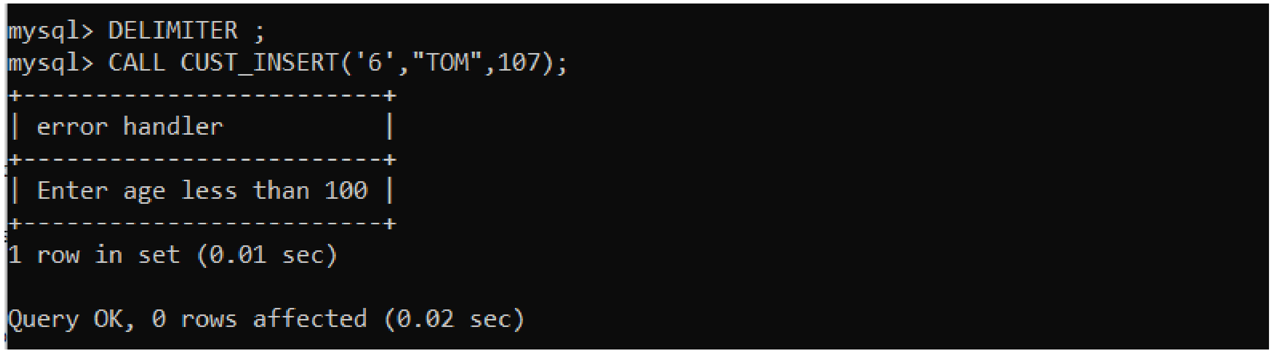
2. Declare an error handler for customer table whenever user inputs the customer's age above 100 years.

ALTERING TABLE FOR ERROR HANDLING
AND CREATING ERROR HANDLING USING PROCEDURE:

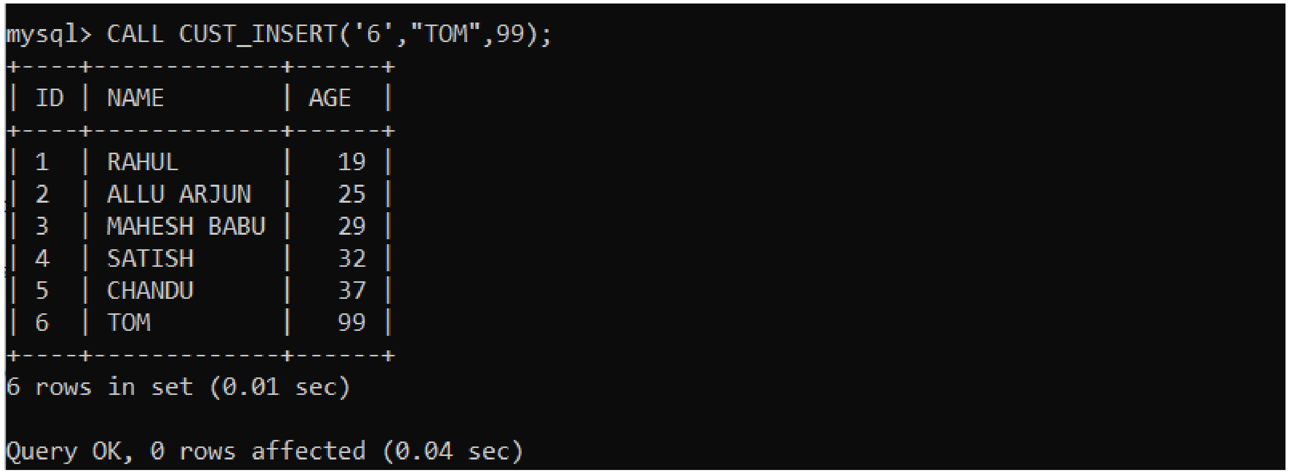


```
mysql> ALTER TABLE CUSTOMER ADD check (age <= 100);
Query OK, 5 rows affected (0.14 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> DELIMITER /
mysql> CREATE PROCEDURE CUST_INSERT(in id varchar(30), in name varchar(60), in age int)
    -> BEGIN
    -> DECLARE exit handler for 3819
    -> begin
    -> select "Enter age less than 100" AS "error handler";
    -> end;
    -> insert into customer values(id, name, age);
    -> SELECT * FROM CUSTOMER;
    -> END /
Query OK, 0 rows affected (0.02 sec)
```

CALLING THE CREATED PROCEDURE:



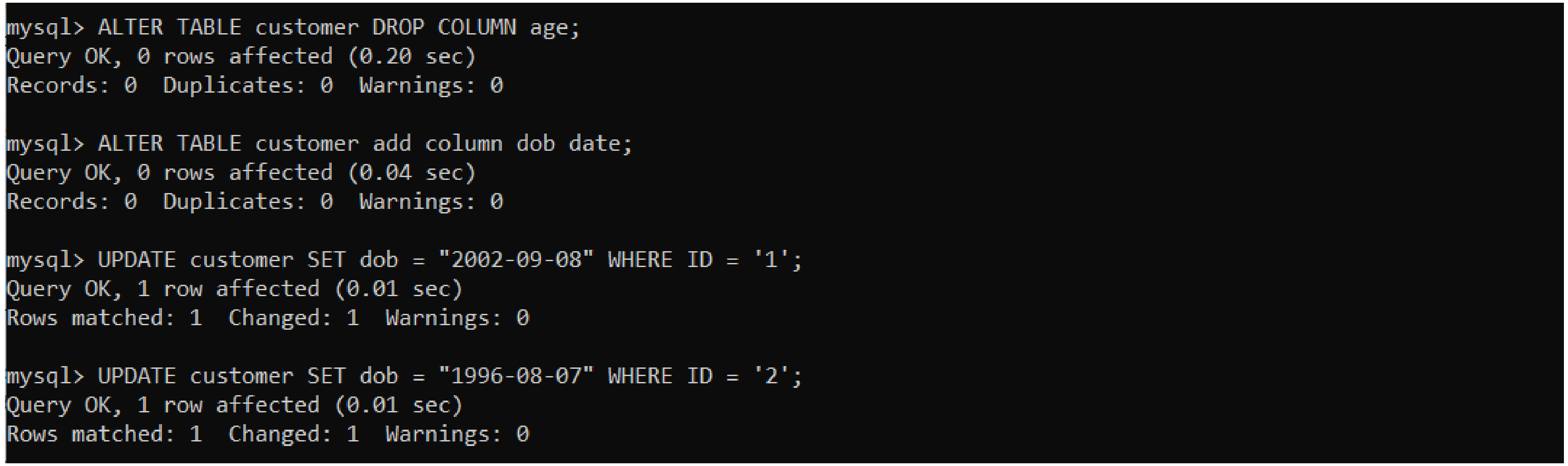
```
mysql> DELIMITER ;
mysql> CALL CUST_INSERT('6',"TOM",107);
+-------------------------+
| error handler           |
+-------------------------+
| Enter age less than 100 |
+-------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```



```
mysql> CALL CUST_INSERT('6',"TOM",99);
+----+-------------+------+
| ID | NAME        | AGE  |
+----+-------------+------+
| 1  | RAHUL       |   19 |
| 2  | ALLU ARJUN  |   25 |
| 3  | MAHESH BABU |   29 |
| 4  | SATISH      |   32 |
| 5  | CHANDU      |   37 |
| 6  | TOM         |   99 |
+----+-------------+------+
6 rows in set (0.01 sec)

Query OK, 0 rows affected (0.04 sec)
```

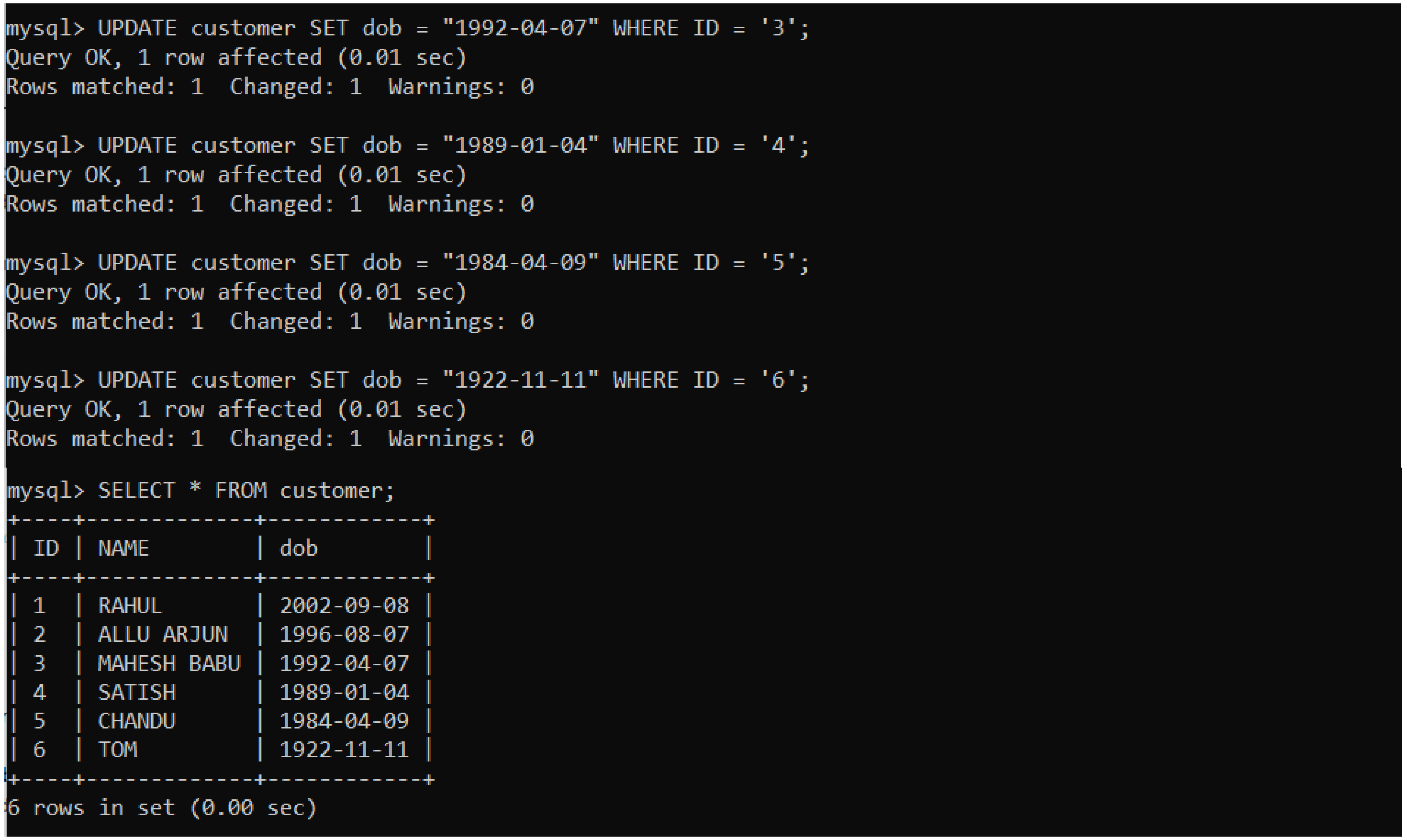
3. Create a function to calculate the age of the all customers based on DateOfBirth(For this program, alter the customer table such that, remove “age” column and add ”dob” column)

ALTERING TABLE ACCORDING TO THE QUESTION:



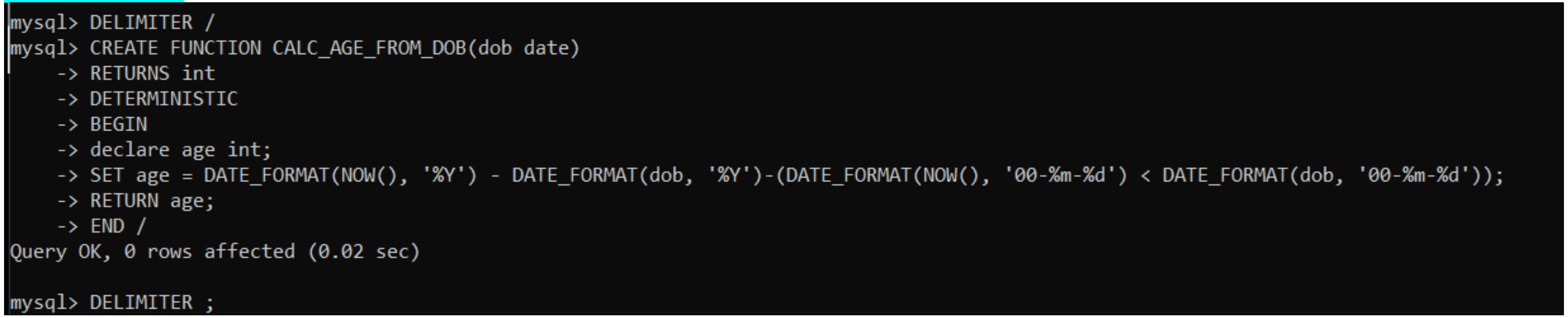
```
mysql> ALTER TABLE customer DROP COLUMN age;
Query OK, 0 rows affected (0.20 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE customer add column dob date;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE customer SET dob = "2002-09-08" WHERE ID = '1';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE customer SET dob = "1996-08-07" WHERE ID = '2';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

ADDING DOB VALUES FOR EACH RECORD:



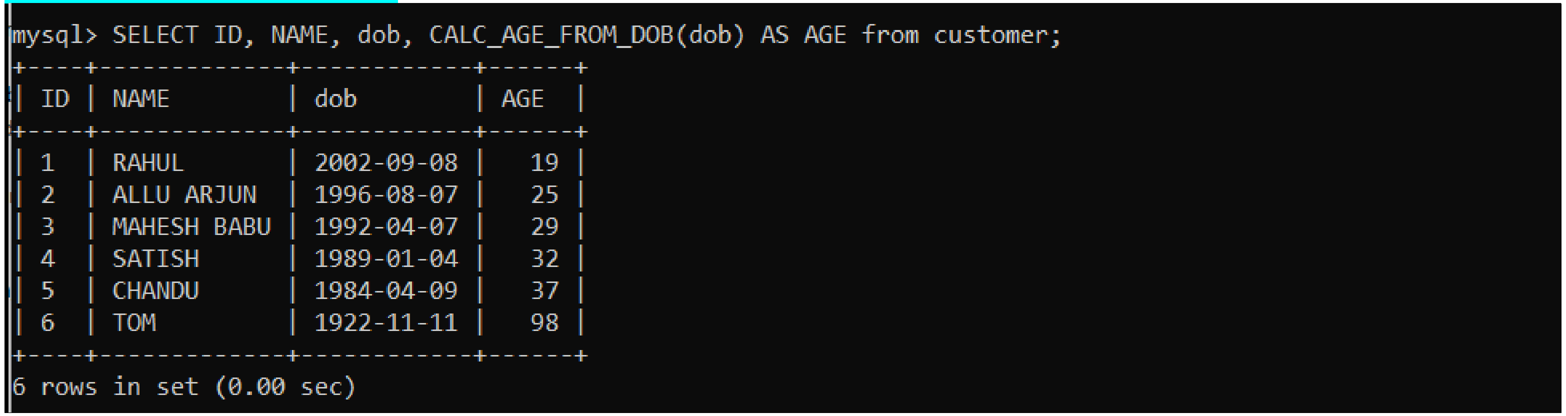
```
mysql> UPDATE customer SET dob = "1992-04-07" WHERE ID = '3';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE customer SET dob = "1989-01-04" WHERE ID = '4';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE customer SET dob = "1984-04-09" WHERE ID = '5';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE customer SET dob = "1922-11-11" WHERE ID = '6';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM customer;
+----+-------------+------------+
| ID | NAME        | dob        |
+----+-------------+------------+
| 1  | RAHUL       | 2002-09-08 |
| 2  | ALLU ARJUN  | 1996-08-07 |
| 3  | MAHESH BABU | 1992-04-07 |
| 4  | SATISH      | 1989-01-04 |
| 5  | CHANDU      | 1984-04-09 |
| 6  | TOM         | 1922-11-11 |
+----+-------------+------------+
6 rows in set (0.00 sec)
```

FUNCTION:



```
mysql> DELIMITER /
mysql> CREATE FUNCTION CALC_AGE_FROM_DOB(dob date)
    -> RETURNS int
    -> DETERMINISTIC
    -> BEGIN
    -> declare age int;
    -> SET age = DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(dob, '%Y')-(DATE_FORMAT(NOW(), '00-%m-%d') < DATE_FORMAT(dob, '00-%m-%d'));
    -> RETURN age;
    -> END /
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
```

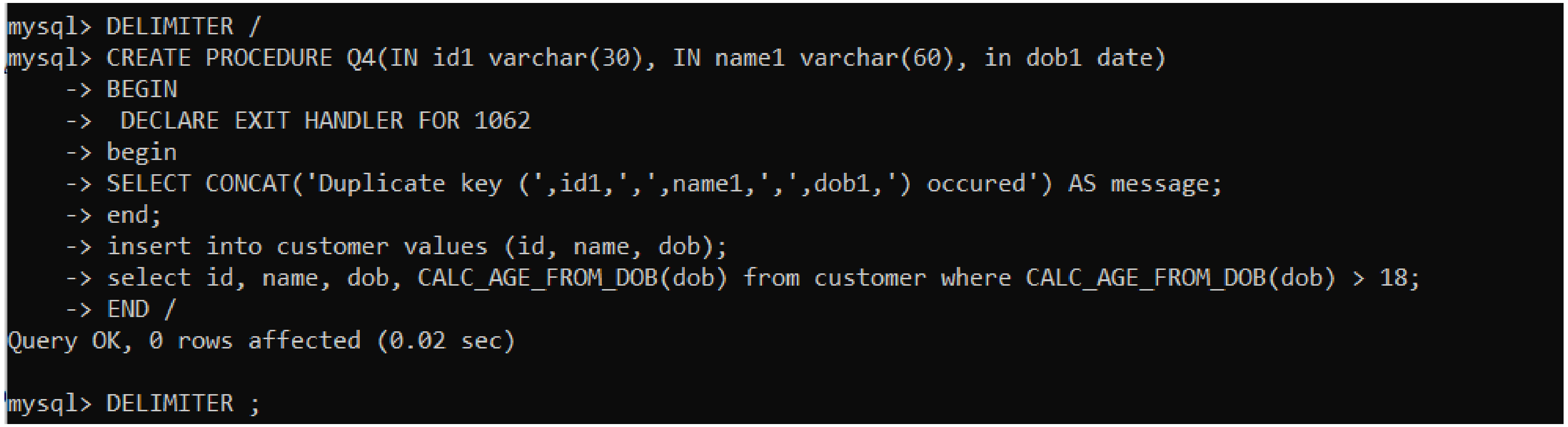
AGES FROM THE DOB:



```
mysql> SELECT ID, NAME, dob, CALC_AGE_FROM_DOB(dob) AS AGE from customer;
+----+-------------+------------+------+
| ID | NAME        | dob        | AGE  |
+----+-------------+------------+------+
| 1  | RAHUL       | 2002-09-08 |   19 |
| 2  | ALLU ARJUN  | 1996-08-07 |   25 |
| 3  | MAHESH BABU | 1992-04-07 |   29 |
| 4  | SATISH      | 1989-01-04 |   32 |
| 5  | CHANDU      | 1984-04-09 |   37 |
| 6  | TOM         | 1922-11-11 |   98 |
+----+-------------+------------+------+
6 rows in set (0.00 sec)
```

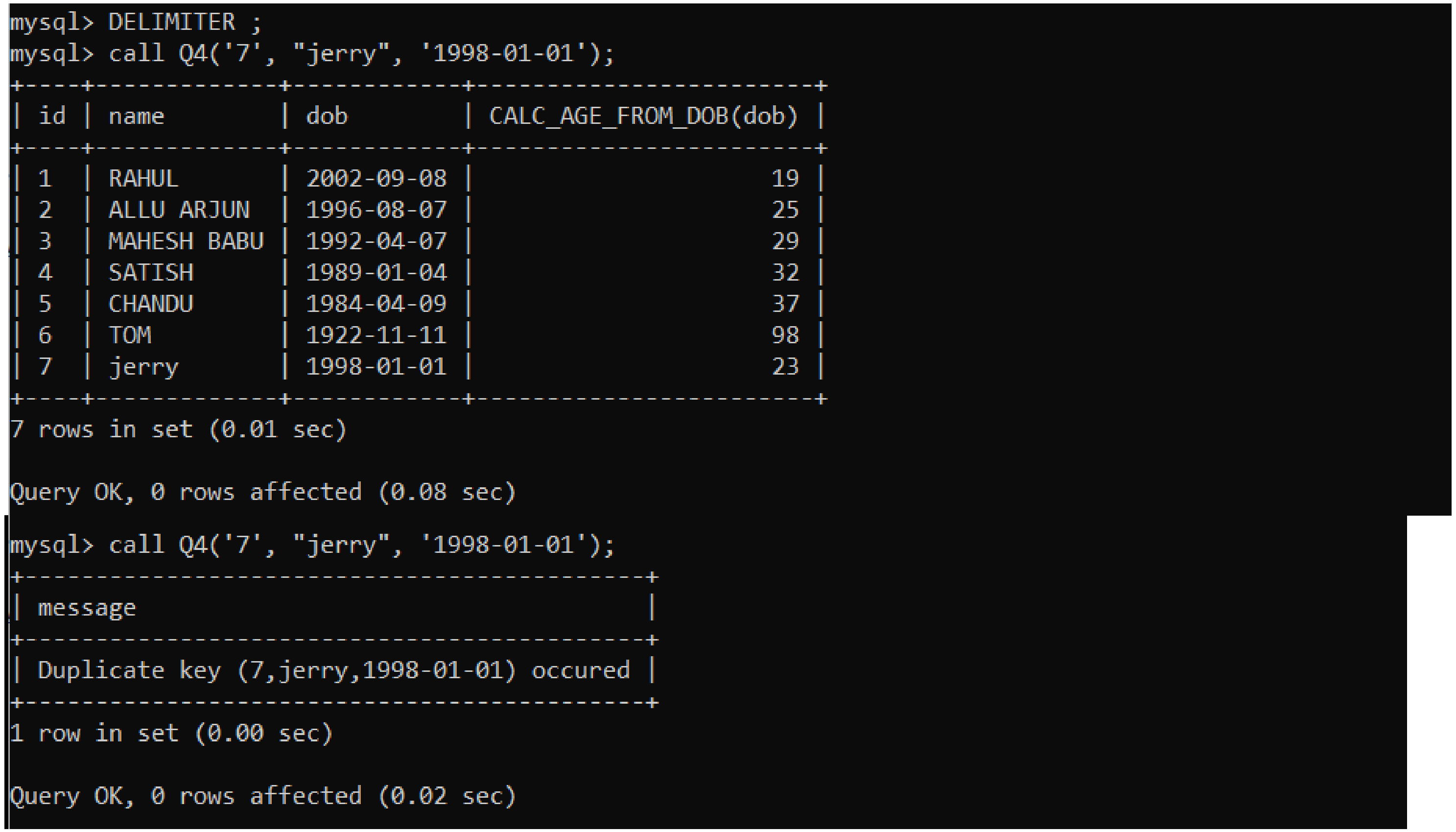
4. Create an error handler that terminates the stored procedure whenever a duplicate key occurs and list out the number of customers who are majors.

PROCEDURE:



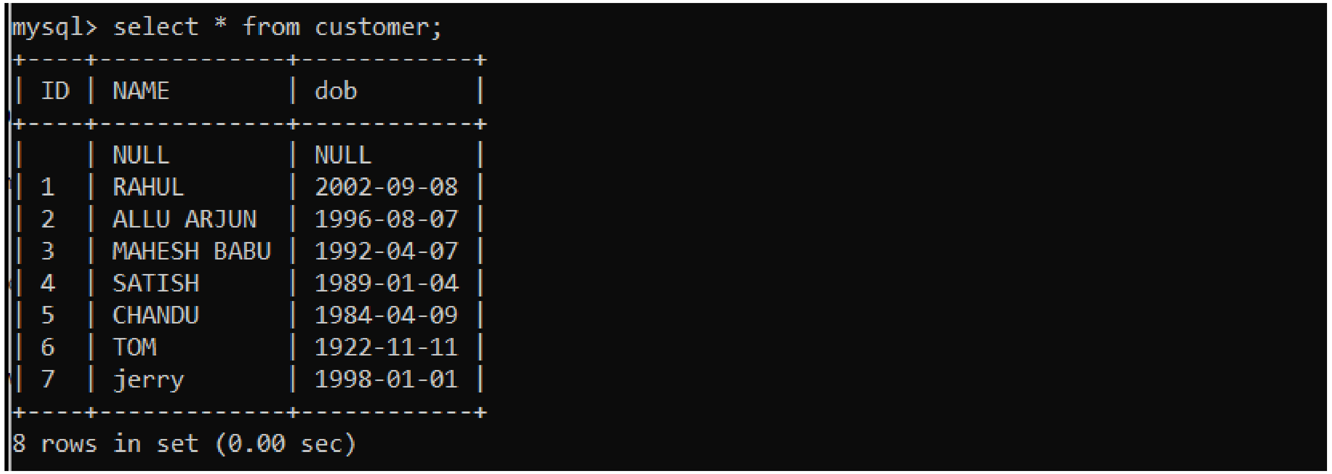
```
mysql> DELIMITER /
mysql> CREATE PROCEDURE Q4(IN id1 varchar(30), IN name1 varchar(60), in dob1 date)
    -> BEGIN
    ->  DECLARE EXIT HANDLER FOR 1062
    -> begin
    -> SELECT CONCAT('Duplicate key (',id1,',',name1,',',dob1,') occured') AS message;
    -> end;
    -> insert into customer values (id, name, dob);
    -> select id, name, dob, CALC_AGE_FROM_DOB(dob) from customer where CALC_AGE_FROM_DOB(dob) > 18;
    -> END /
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
```

CALLING PROCEDURE AND CALLING AGAIN SAME PROCEDURE:



```
mysql> DELIMITER ;
mysql> call Q4('7', "jerry", '1998-01-01');
+----+-------------+------------+-----------------------+
| id | name        | dob        | CALC_AGE_FROM_DOB(dob) |
+----+-------------+------------+-----------------------+
| 1  | RAHUL       | 2002-09-08 |                    19 |
| 2  | ALLU ARJUN  | 1996-08-07 |                    25 |
| 3  | MAHESH BABU | 1992-04-07 |                    29 |
| 4  | SATISH      | 1989-01-04 |                    32 |
| 5  | CHANDU      | 1984-04-09 |                    37 |
| 6  | TOM         | 1922-11-11 |                    98 |
| 7  | jerry       | 1998-01-01 |                    23 |
+----+-------------+------------+-----------------------+
7 rows in set (0.01 sec)

Query OK, 0 rows affected (0.08 sec)

mysql> call Q4('7', "jerry", '1998-01-01');
+----------------------------------------------+
| message                                      |
+----------------------------------------------+
| Duplicate key (7,jerry,1998-01-01) occured |
+----------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

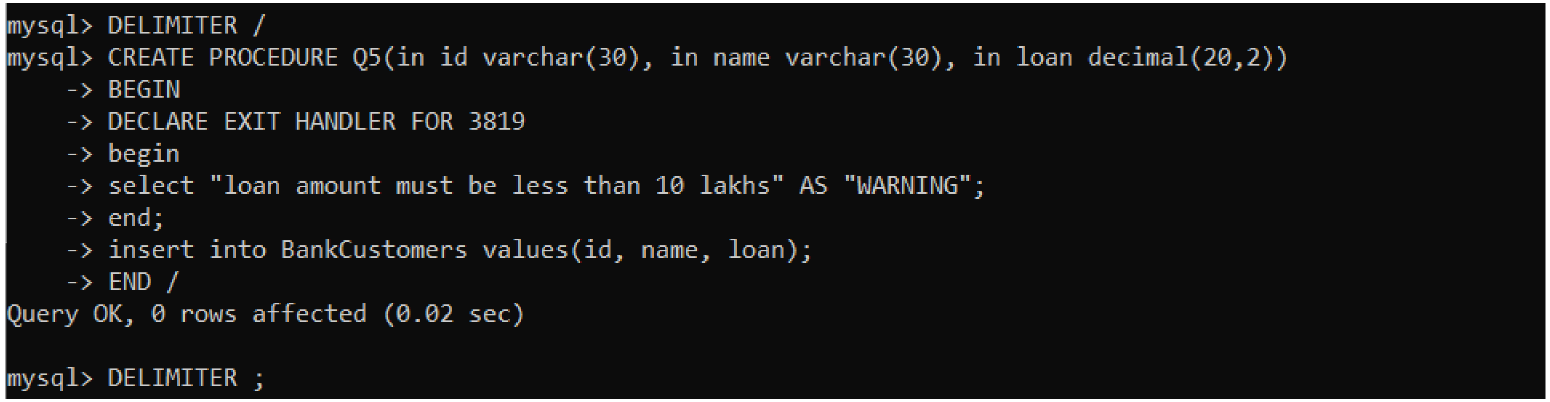
ONLY ONCE THE RECORD IS ENTERED:



```
mysql> select * from customer;
+----+-------------+------------+
| ID | NAME        | dob        |
+----+-------------+------------+
|    | NULL        | NULL       |
| 1  | RAHUL       | 2002-09-08 |
| 2  | ALLU ARJUN  | 1996-08-07 |
| 3  | MAHESH BABU | 1992-04-07 |
| 4  | SATISH      | 1989-01-04 |
| 5  | CHANDU      | 1984-04-09 |
| 6  | TOM         | 1922-11-11 |
| 7  | jerry       | 1998-01-01 |
+----+-------------+------------+
8 rows in set (0.00 sec)
```

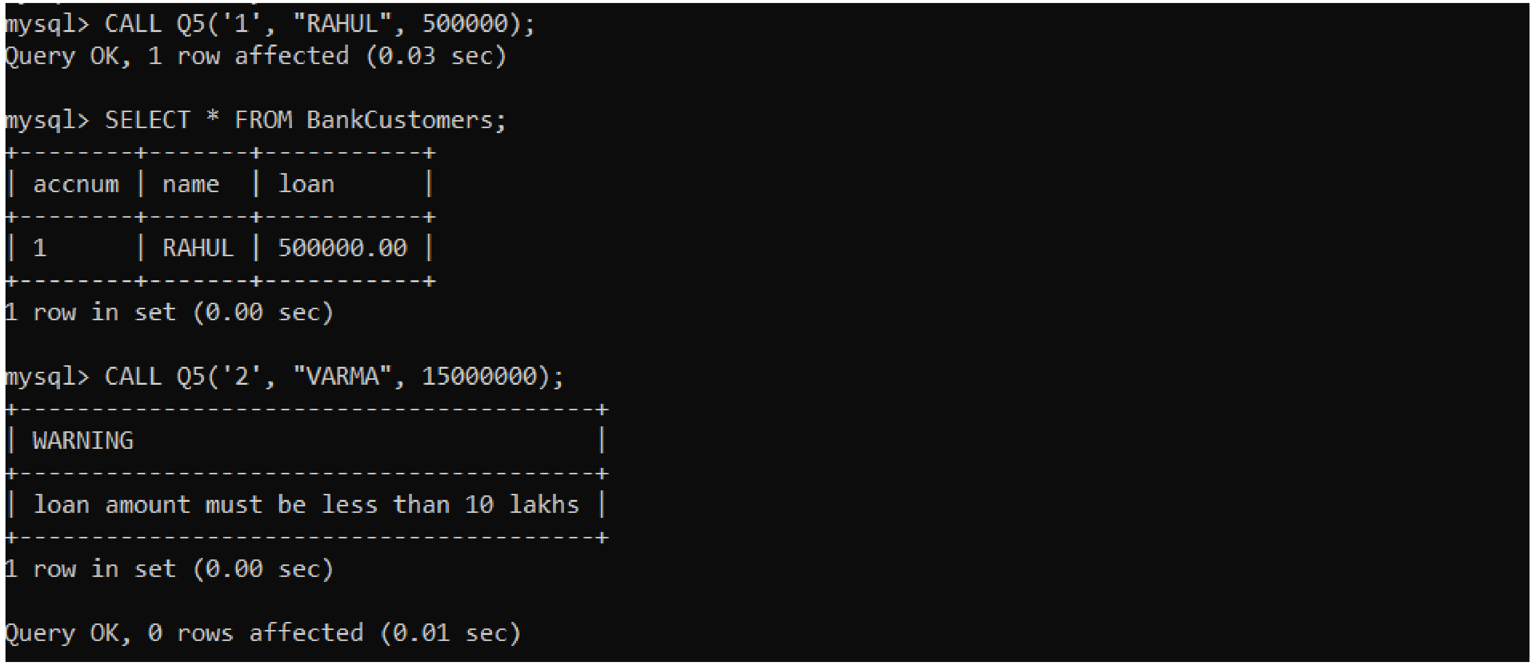
5. Consider a table with the schema BankCustomers (accNum, name and loan). Raise an exception when the customer initiates loan amount above

CREATING PROCEDURE:



```
mysql> DELIMITER /
mysql> CREATE PROCEDURE Q5(in id varchar(30), in name varchar(30), in loan decimal(20,2))
    -> BEGIN
    -> DECLARE EXIT HANDLER FOR 3819
    -> begin
    -> select "loan amount must be less than 10 lakhs" AS "WARNING";
    -> end;
    -> insert into BankCustomers values(id, name, loan);
    -> END /
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
```

CALLING PROCEDURE:



```
mysql> CALL Q5('1', "RAHUL", 500000);
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM BankCustomers;
+--------+-------+-----------+
| accnum | name  | loan      |
+--------+-------+-----------+
| 1      | RAHUL | 500000.00 |
+--------+-------+-----------+
1 row in set (0.00 sec)

mysql> CALL Q5('2', "VARMA", 15000000);
+-------------------------------------+
| WARNING                             |
+-------------------------------------+
| loan amount must be less than 10 lakhs |
+-------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

THANK YOU