

# DBMS LAB-9

**NAME:- ABHISHIKTH BODA**

**ROLL NUMBER:- S20210010044**

**DATE:- 02-11-2022**

## Example 1:-

### Function:-

```
mysql> create function my_fun(value numeric(12,2))
-> returns int
-> deterministic
-> begin
-> declare r int;
-> select count(*) into r from account_relation where
-> balance>value;
-> return (r);
-> end<>
Query OK, 0 rows affected (0.01 sec)
```

### Execution:-

```
mysql> select my_fun(500);
-> select balance,my_fun(balance) from account order
-> by balance;
-> <>
+-----+
| my_fun(500) |
+-----+
|          4 |
+-----+
1 row in set (0.01 sec)
```

## Example 2:-

### Function:-

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE emp_insert(
  -> IN name VARCHAR(20),
  -> IN id INT
  -> )
  -> BEGIN
  -> -- exit if the duplicate key occurs
  -> DECLARE EXIT HANDLER FOR 1062
  -> BEGIN
  -> SELECT CONCAT('Duplicate key (' ,name,',',',id,') occurred') AS
  -> message; END;
  -> -- insert a new row into the EMP table
  -> INSERT INTO emp(emp_id,emp_name)
  -> VALUES(id,name);
  -> -- return the row inserted recently
  -> Select * from emp where emp_id=id;
  -> END $$
Query OK, 0 rows affected (0.00 sec)

mysql> _
```

### Execution:-

```
mysql> Call emp_insert ('Jo',30);
+-----+-----+
| emp_name | emp_id |
+-----+-----+
| Jo      | 30     |
+-----+-----+
1 row in set (0.01 sec)

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

```
mysql> Call emp_insert('mart',20);
+-----+-----+
| emp_name | emp_id |
+-----+-----+
| mart     | 20     |
| mart     | 20     |
+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql> Call emp_insert('John',10);
+-----+-----+
| emp_name | emp_id |
+-----+-----+
| John     | 10     |
+-----+-----+
1 row in set (0.02 sec)

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

```
mysql> Call emp_insert('mart',20);
+-----+-----+
| emp_name | emp_id |
+-----+-----+
| mart     | 20     |
+-----+-----+
1 row in set (0.01 sec)

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

## Exercise 1:-

### Question 1:-

Display all account numbers and 'Level' of the customer based on the balance amount. (Level='PLATINUM' if balance >=900 ; Level='Gold' if balance >=700 and <900; Level='Silver' if balance <700).

### Answer:-

### Function:-

```
mysql> delimiter //
mysql> create function customer_level(balance int)
    -> returns varchar(10)
    -> deterministic
    -> begin
    -> declare level varchar(10);
    -> if balance >= 900 then
    -> set level = 'PLATINUM';
    -> elseif balance >= 700 and balance < 900 then
    -> set level = 'Gold';
    -> elseif balance < 700 then
    -> set level = 'Silver';
    -> end if;
    -> return (level);
    -> end//
Query OK, 0 rows affected (0.01 sec)
```

### Execution:-

```
mysql> select accountnumber, customer_level(balance) from account_relation;
+-----+-----+
| accountnumber | customer_level(balance) |
+-----+-----+
| A-101         | Silver                  |
| A-102         | Silver                  |
| A-201         | PLATINUM                |
| A-215         | Gold                    |
| A-217         | Gold                    |
| A-222         | Gold                    |
| A-305         | Silver                  |
+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

## Question 2:-

Display the experience of each of the employees along with other details using a stored function.

## Answer:-

### Function:-

```
mysql> delimiter <>
mysql> create function experience(doj date) returns int
-> deterministic
-> begin
-> declare exp int;
-> set exp = year(current_date()) - year(doj);
-> return (exp);
-> end<>
Query OK, 0 rows affected (0.01 sec)
```

### Execution:-

```
mysql> select emp_id,emp_name,emp_dept,emp_age,place,income,doj,experience(doj) from employee;<>
+-----+-----+-----+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_dept | emp_age | place | income | doj | experience(doj) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 2505 | peter | Finance | 32 | Newyork | 100000 | 2002-08-25 | 20 |
| 2506 | Mark | HR | 32 | California | 120000 | 1980-03-25 | 42 |
| 2507 | Donald | Finance | 28 | Arizona | 100000 | 1995-12-26 | 27 |
| 2508 | Obama | Management | 35 | Florida | 500000 | 1990-10-30 | 32 |
| 2509 | Linklon | HR | 25 | Georgia | 25000 | 2008-08-08 | 14 |
| 2510 | Kane | Sales | 29 | Alaska | 30000 | 2000-01-01 | 22 |
| 2511 | Adam | Management | 38 | California | 54000 | 2020-10-25 | 2 |
| 2512 | Mac | Finance | 40 | Florida | 280000 | 1970-06-09 | 52 |
| 2513 | Manas | Accounts | 29 | India | 600000 | 1990-12-11 | 32 |
| 2514 | Vasin | Accounts | 30 | India | 800000 | 1989-10-10 | 33 |
| 2515 | peter | Finance | 32 | Newyork | 100000 | 1989-10-10 | 33 |
| 2516 | Mark | HR | 32 | California | 120000 | 1990-12-11 | 32 |
| 2517 | Donald | Finance | 28 | Arizona | 100000 | 1970-06-09 | 52 |
| 2518 | Obama | Management | 35 | Florida | 500000 | 2020-10-25 | 2 |
| 2519 | Linklon | HR | 25 | Georgia | 25000 | 2000-01-01 | 22 |
| 2520 | Kane | Sales | 29 | Alaska | 30000 | 2008-08-08 | 14 |
| 2521 | Adam | Management | 38 | California | 54000 | 1990-10-30 | 32 |
| 2522 | Mac | Finance | 40 | Florida | 280000 | 1995-12-26 | 27 |
| 2523 | Manas | Accounts | 29 | India | 600000 | 1980-03-25 | 42 |
| 2524 | Vasin | Accounts | 30 | India | 800000 | 2002-08-25 | 20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
20 rows in set (0.00 sec)

mysql> _
```

### Question 3:-

Count the number of borrowers who are having a loan amount of more than 1000 using a stored function.

### Answer:-

#### Function:-

```
mysql> delimiter <>
mysql> create function no_of_borrowers() returns int
  -> deterministic
  -> begin
  -> declare n int;
  -> select count(*) into n from loan_relation
  -> where amount>1000;
  -> return n;
  -> end<>
Query OK, 0 rows affected (0.01 sec)
```

#### Execution:-

```
mysql> select no_of_borrowers() from loan_relation;<>
+-----+
| no_of_borrowers() |
+-----+
| 4 |
| 4 |
| 4 |
| 4 |
| 4 |
| 4 |
| 4 |
+-----+
7 rows in set (0.01 sec)
```

## Question 4:-

Find the customer name who is having the highest balance in his account using a stored function.

## Answer:-

## Function:-

```
mysql> create function find_name() returns varchar(25)
-> deterministic
-> begin
-> declare name varchar(25);
-> select customername into name from
-> depositor_relation natural join account_relation
-> where balance = (select max(balance) from account_relation);
-> return name;
-> end<>
Query OK, 0 rows affected (0.01 sec)
```

## Execution:-

```
mysql> select highbalance_customer() from customer;
+-----+
| highbalance_customer() |
+-----+
| Johnson                |
+-----+
```

## Exercise 2:-

### Question 1:-

Create a table with at least one field with non-null constraint and then define the required procedure and error handler which will exit and print "Trying to populate a non-null column with null value" whenever an attempt to insert null value for that field will be done.

### Answer:-

### Function:-

```
mysql> create procedure test_insert(IN col1 varchar(20), IN col2 int)
-> begin
-> declare exit handler for 1048
-> begin
-> select "Trying to populate a non-null column with null value"
-> as message;
-> end;
-> insert into test values(col1, col2);
-> end<>
Query OK, 0 rows affected (0.01 sec)
```

### Execution:-

```
mysql> create table test(col1 varchar(20) NOT NULL,col2 int);
-> <>\
Query OK, 0 rows affected (0.03 sec)

mysql> call test_insert(NULL,100);<>
+-----+
| message                                     |
+-----+
| Trying to populate a non-null column with null value |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

←THE END→