Government Engineering College, Thrissur

CS333– Application Software Development Lab
Documentation -

Exp 12 – Control Structures
Exp 13 – Procedures and Functions

Date of Submission
17 October 2020

Submitted By
Kowsik Nandagopan D
Roll No 31
TCR18CS031
GECT CSE S5

17 Oct 2020

Experiment 11

AIM

Implementation of various control structures using PL/SQL

Description

Implement Control Structures in MySQL to get valuable insight from the previously created database. Functions available in MySQL are -

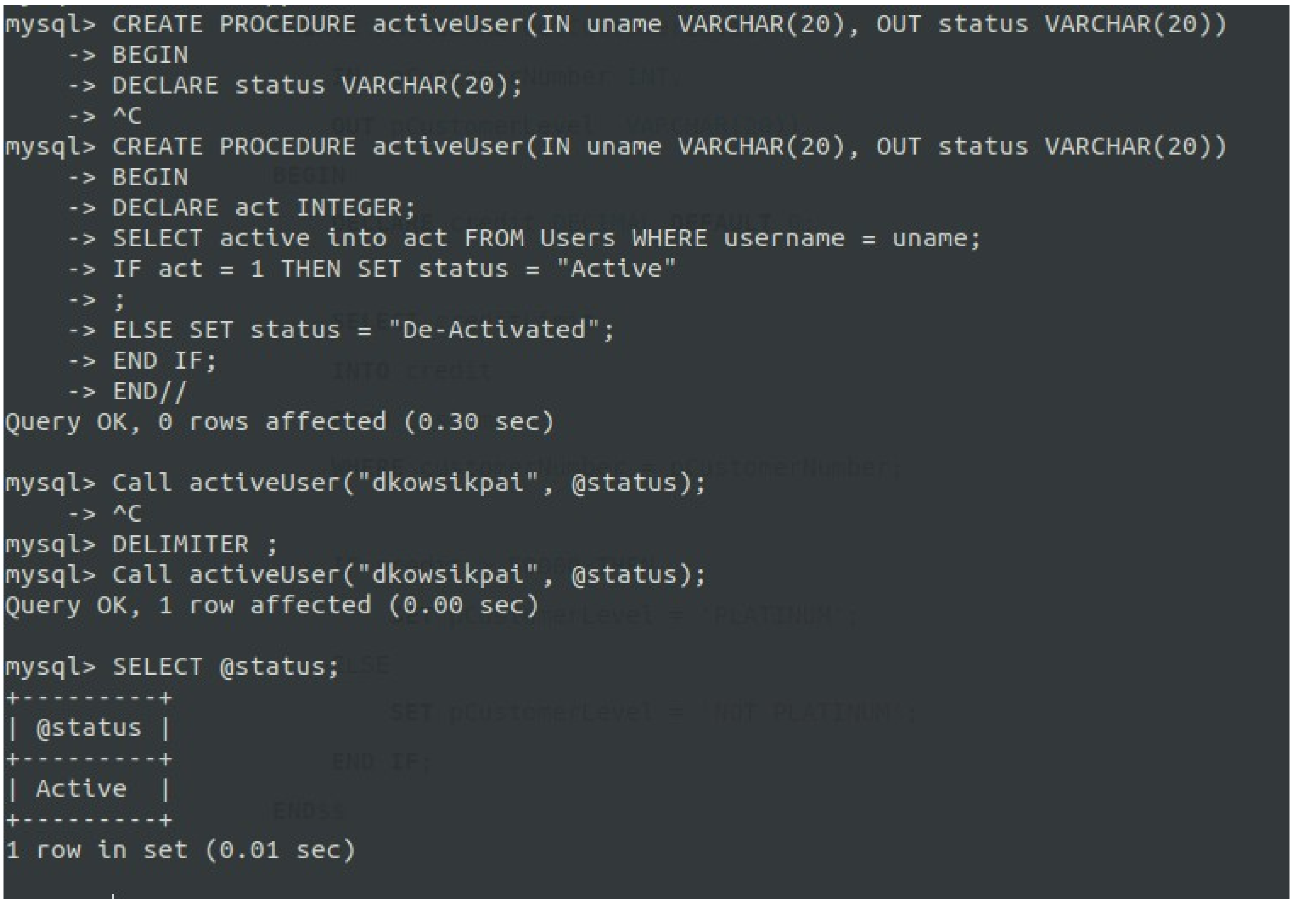
- CASE
- IF
- ITERATE
- LEAVE
- REPEAT
- WHILE
- LOOP

Output / Screenshots

1. IF - THEN - ELSE – THEN

Create a procedure using the control flow that gives the result whether the user is active or not

P. T. O

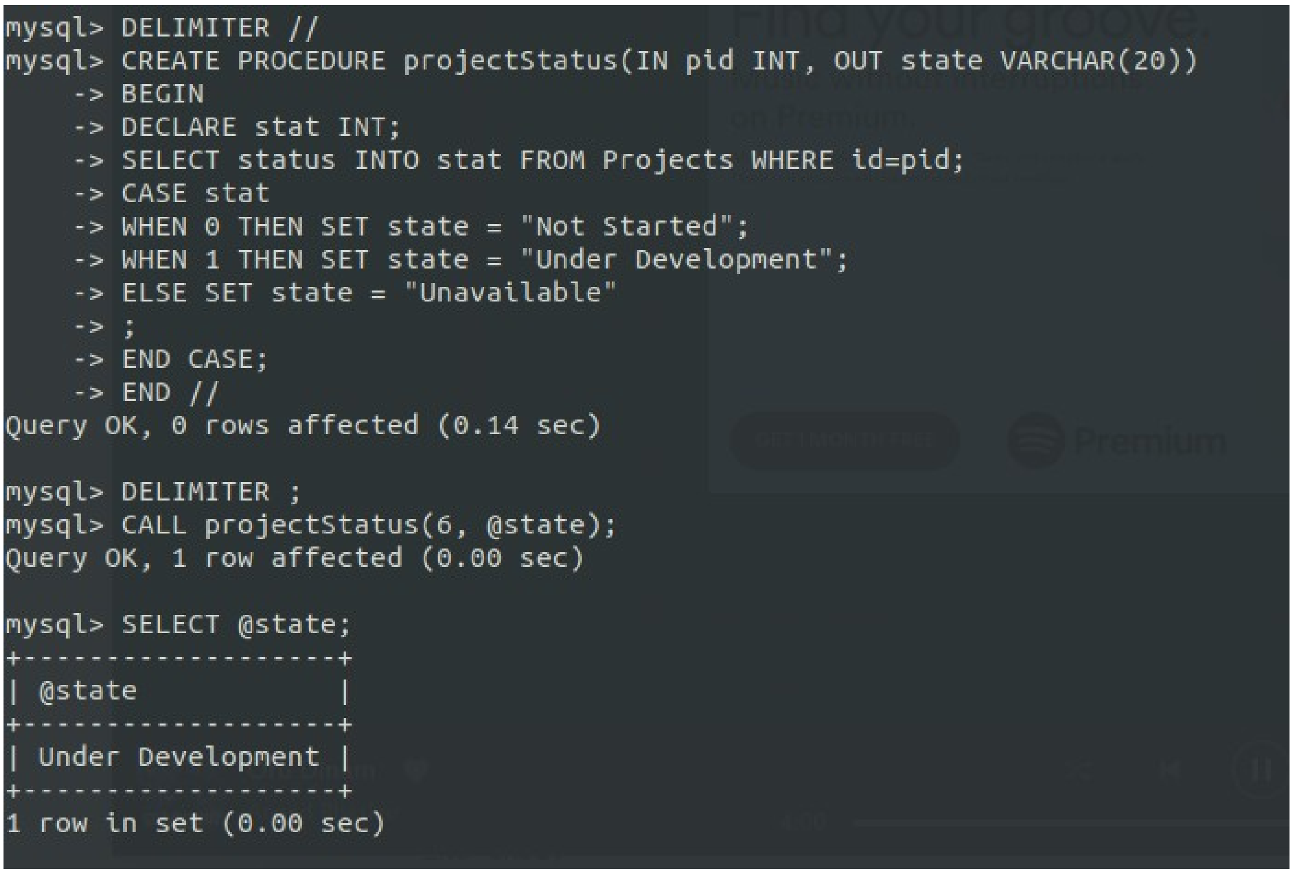


```
mysql> CREATE PROCEDURE activeUser(IN uname VARCHAR(20), OUT status VARCHAR(20))
    -> BEGIN
    -> DECLARE status VARCHAR(20);
    -> ^C
mysql> CREATE PROCEDURE activeUser(IN uname VARCHAR(20), OUT status VARCHAR(20))
    -> BEGIN
    -> DECLARE act INTEGER;
    -> SELECT active into act FROM Users WHERE username = uname;
    -> IF act = 1 THEN SET status = "Active"
    -> ;
    -> ELSE SET status = "De-Activated";
    -> END IF;
    -> END//
Query OK, 0 rows affected (0.30 sec)

mysql> Call activeUser("dkowsikpai", @status);
    -> ^C
mysql> DELIMITER ;
mysql> Call activeUser("dkowsikpai", @status);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @status;
+---------+
| @status |
+---------+
| Active  |
+---------+
1 row in set (0.01 sec)
```

2. CASE

Get the project status from the procedural call

P. T. O

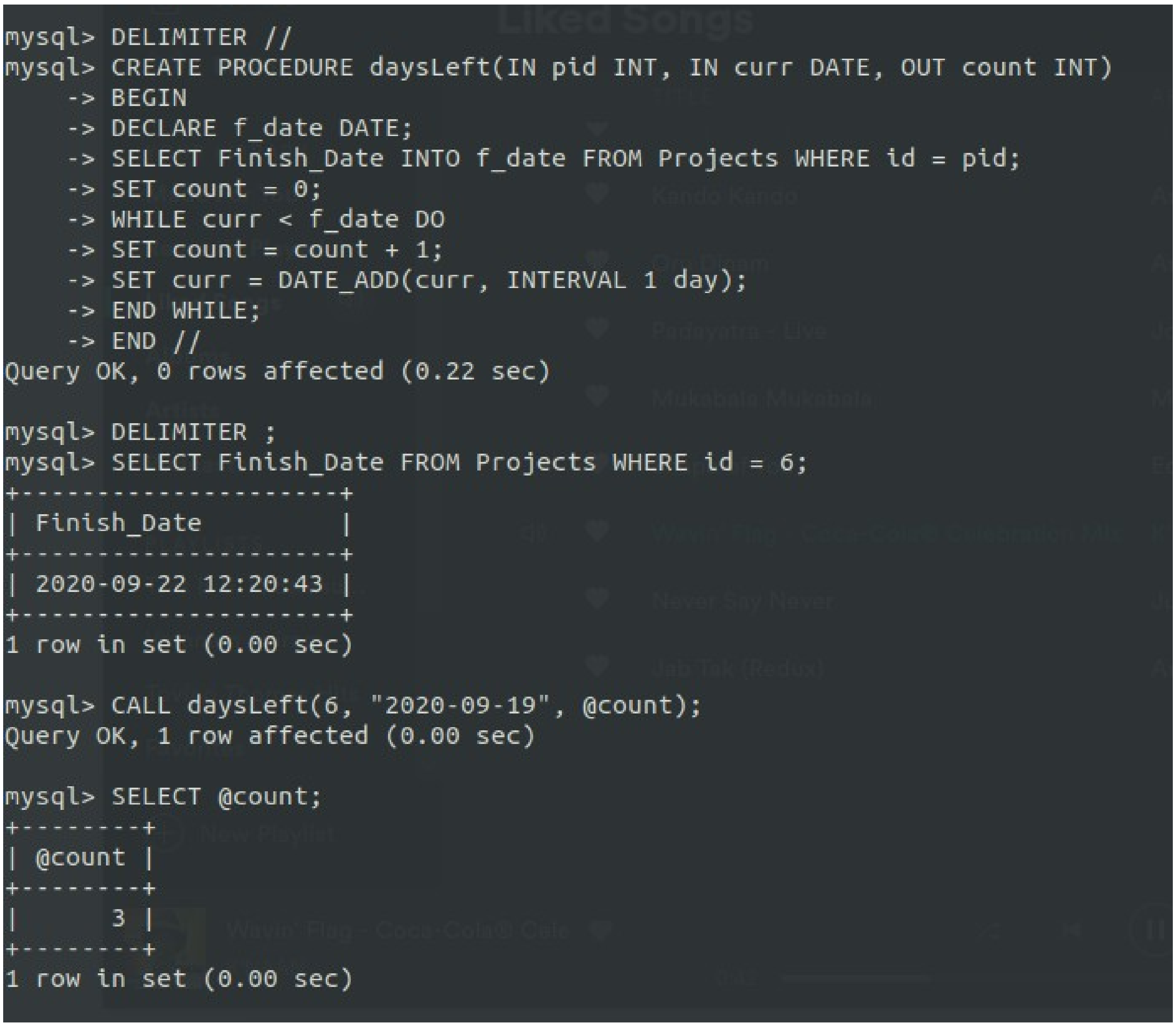


```
mysql> DELIMITER //
mysql> CREATE PROCEDURE projectStatus(IN pid INT, OUT state VARCHAR(20))
    -> BEGIN
    -> DECLARE stat INT;
    -> SELECT status INTO stat FROM Projects WHERE id=pid;
    -> CASE stat
    -> WHEN 0 THEN SET state = "Not Started";
    -> WHEN 1 THEN SET state = "Under Development";
    -> ELSE SET state = "Unavailable"
    -> ;
    -> END CASE;
    -> END //
Query OK, 0 rows affected (0.14 sec)

mysql> DELIMITER ;
mysql> CALL projectStatus(6, @state);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @state;
+-------------------+
| @state            |
+-------------------+
| Under Development |
+-------------------+
1 row in set (0.00 sec)
```

3. WHILE

Get the number of days left to finish date from the given date

P. T. O

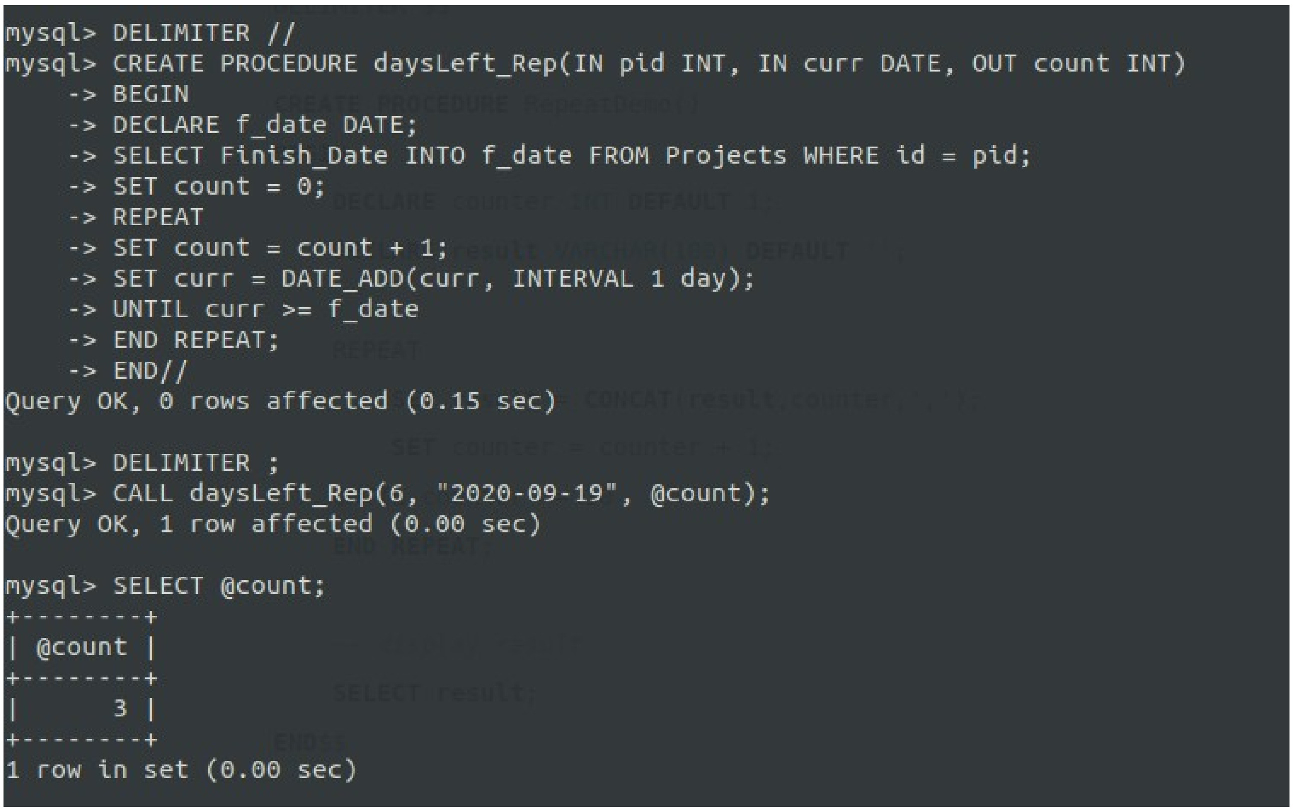


```
mysql> DELIMITER //
mysql> CREATE PROCEDURE daysLeft(IN pid INT, IN curr DATE, OUT count INT)
    -> BEGIN
    -> DECLARE f_date DATE;
    -> SELECT Finish_Date INTO f_date FROM Projects WHERE id = pid;
    -> SET count = 0;
    -> WHILE curr < f_date DO
    -> SET count = count + 1;
    -> SET curr = DATE_ADD(curr, INTERVAL 1 day);
    -> END WHILE;
    -> END //
Query OK, 0 rows affected (0.22 sec)

mysql> DELIMITER ;
mysql> SELECT Finish_Date FROM Projects WHERE id = 6;
+---------------------+
| Finish_Date         |
+---------------------+
| 2020-09-22 12:20:43 |
+---------------------+
1 row in set (0.00 sec)

mysql> CALL daysLeft(6, "2020-09-19", @count);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @count;
+--------+
| @count |
+--------+
|      3 |
+--------+
1 row in set (0.00 sec)
```

4. REPEAT

Use the Repeat to do the daysLeft

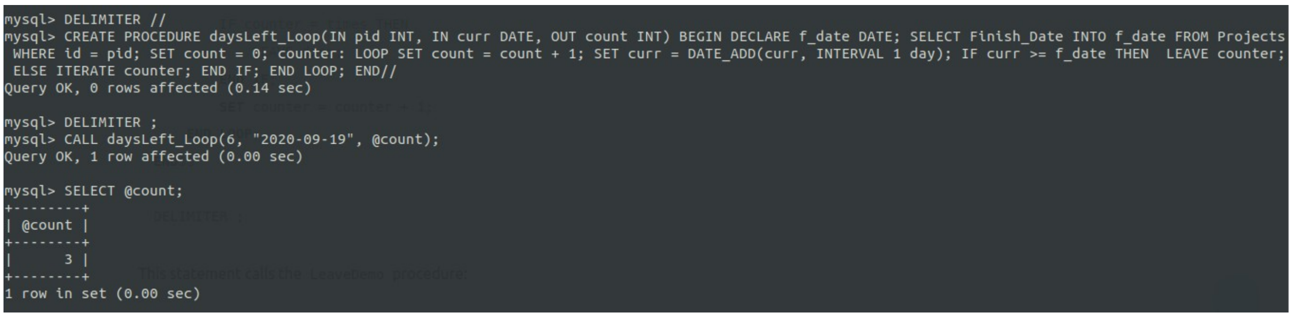
P. T. O



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE daysLeft_Rep(IN pid INT, IN curr DATE, OUT count INT)
    -> BEGIN
    -> DECLARE f_date DATE;
    -> SELECT Finish_Date INTO f_date FROM Projects WHERE id = pid;
    -> SET count = 0;
    -> REPEAT
    -> SET count = count + 1;
    -> SET curr = DATE_ADD(curr, INTERVAL 1 day);
    -> UNTIL curr >= f_date
    -> END REPEAT;
    -> END//
Query OK, 0 rows affected (0.15 sec)

mysql> DELIMITER ;
mysql> CALL daysLeft_Rep(6, "2020-09-19", @count);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @count;
+--------+
| @count |
+--------+
|      3 |
+--------+
1 row in set (0.00 sec)
```

5. LOOP, ITERATE, LEAVE

DaysLeft using Loop, Iterate, Leave



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE daysLeft_Loop(IN pid INT, IN curr DATE, OUT count INT) BEGIN DECLARE f_date DATE; SELECT Finish_Date INTO f_date FROM Projects
 WHERE id = pid; SET count = 0; counter: LOOP SET count = count + 1; SET curr = DATE_ADD(curr, INTERVAL 1 day); IF curr >= f_date THEN  LEAVE counter;
 ELSE ITERATE counter; END IF; END LOOP; END//
Query OK, 0 rows affected (0.14 sec)

mysql> DELIMITER ;
mysql> CALL daysLeft_Loop(6, "2020-09-19", @count);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @count;
+--------+
| @count |
+--------+
|      3 |
+--------+
1 row in set (0.00 sec)
```

17 Oct 2020

Experiment 12

AIM

Creation of Procedures and Functions

Description

Implement Creation of Procedures and Functions in MySQL to get valuable insight from the previously created database. Functions and procedures in MySQL are different.

Stored Procedure	User Defined Functions
Can return **zero, single or multiple** values	Must return single value
We can **use transaction**	We cannot use transaction
We can have **Input / Output parameters**	We can have only Input parameter
We can **call function from procedure**	We cannot call Procedure
We **cannot use** Procedure with SELECT/ WHERE/ HAVING statements	We can use it
We can use **exception handing** using Try-Catch	We cannot use Try Catch

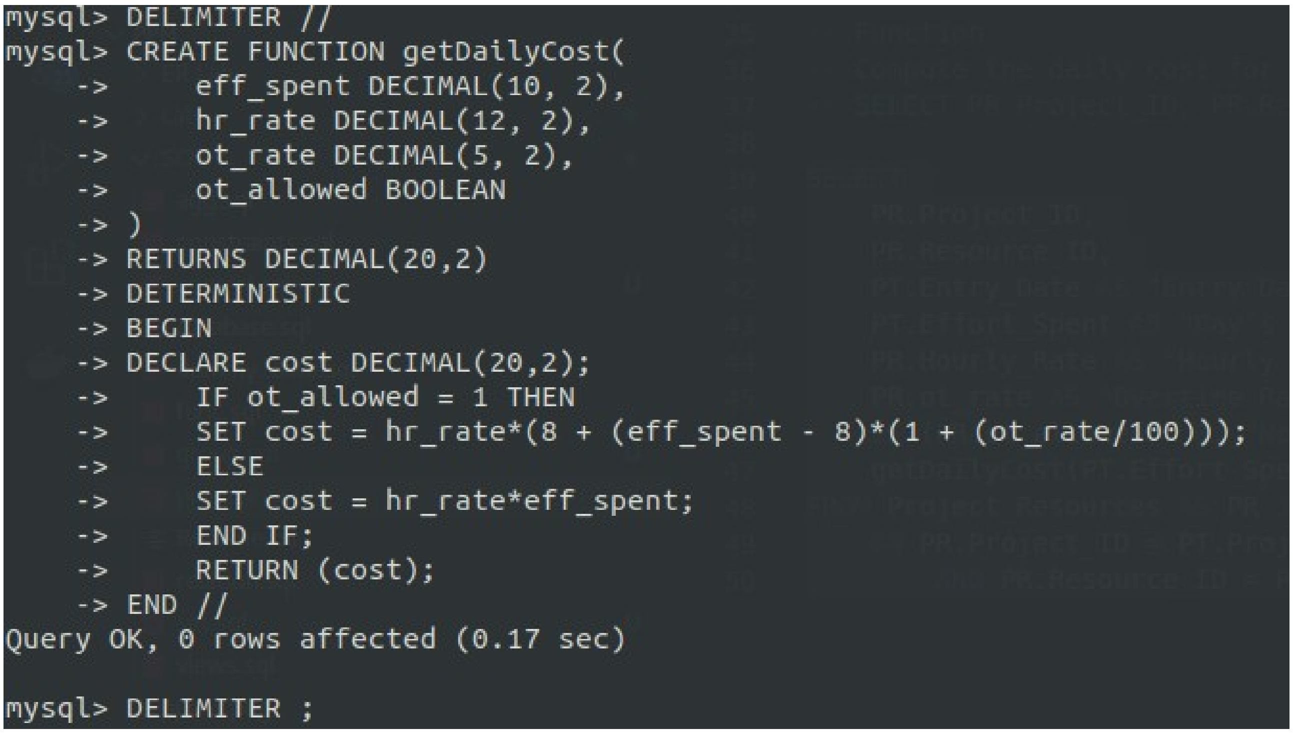
Output / Screenshots

1. Function

Compute the daily cost for each resource by using the transaction table and resource table.

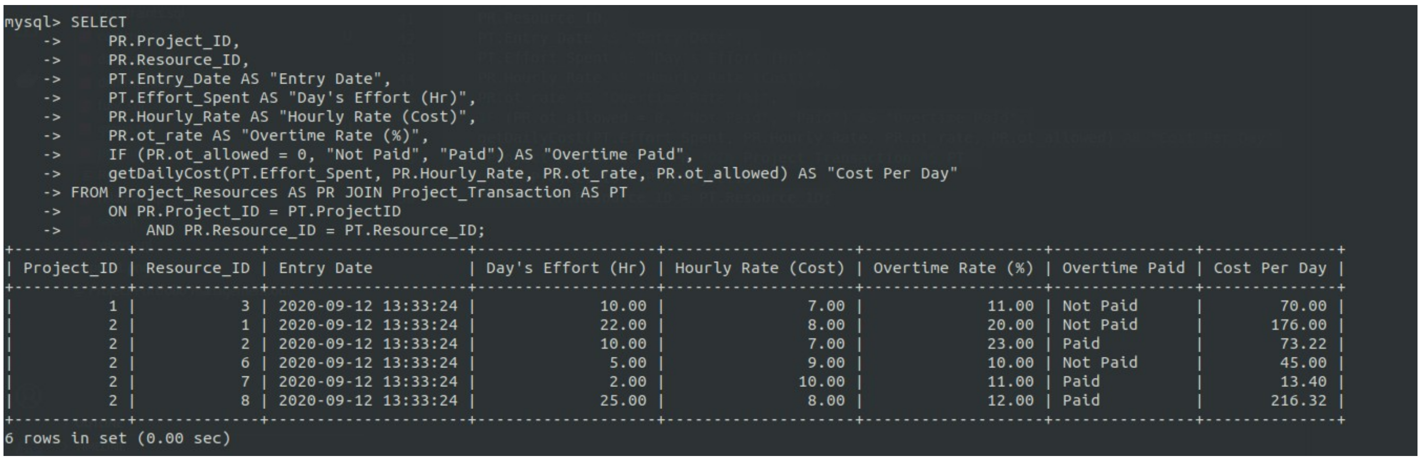
-- Table that will be used

```
SELECT PR.Project_ID, PR.Resource_ID, PR.Hourly_Rate, PR.ot_allowed,
PR.ot_rate, PT.Effort_Spent, PT.Entry_Date
        FROM Project_Resources AS PR JOIN Project_Transaction AS PT ON
PR.Project_ID = PT.ProjectID AND PR.Resource_ID
                = PT.Resource_ID;
```



```
mysql> DELIMITER //
mysql> CREATE FUNCTION getDailyCost(
    ->     eff_spent DECIMAL(10, 2),
    ->     hr_rate DECIMAL(12, 2),
    ->     ot_rate DECIMAL(5, 2),
    ->     ot_allowed BOOLEAN
    -> )
    -> RETURNS DECIMAL(20,2)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE cost DECIMAL(20,2);
    ->     IF ot_allowed = 1 THEN
    ->     SET cost = hr_rate*(8 + (eff_spent - 8)*(1 + (ot_rate/100)));
    ->     ELSE
    ->     SET cost = hr_rate*eff_spent;
    ->     END IF;
    ->     RETURN (cost);
    -> END //
Query OK, 0 rows affected (0.17 sec)

mysql> DELIMITER ;
```

Calling Function

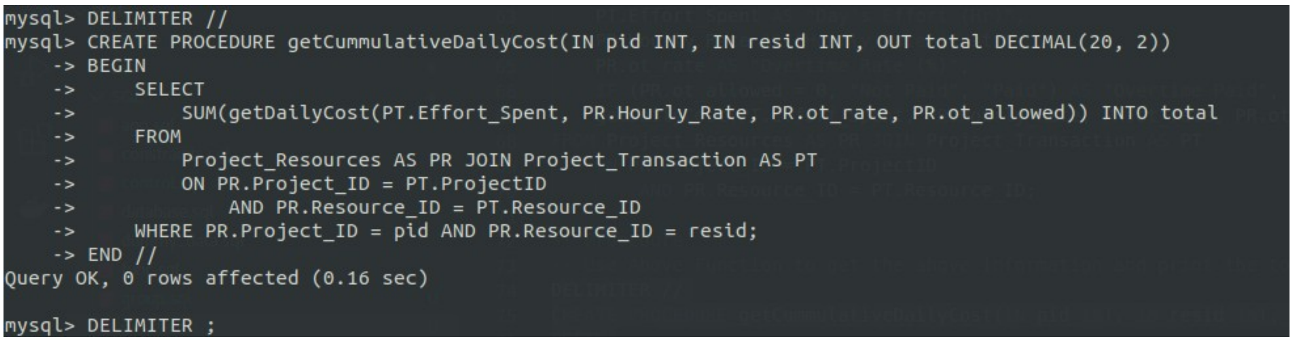


```
mysql> SELECT
    ->     PR.Project_ID,
    ->     PR.Resource_ID,
    ->     PT.Entry_Date AS "Entry Date",
    ->     PT.Effort_Spent AS "Day's Effort (Hr)",
    ->     PR.Hourly_Rate AS "Hourly Rate (Cost)",
    ->     PR.ot_rate AS "Overtime Rate (%)",
    ->     IF (PR.ot_allowed = 0, "Not Paid", "Paid") AS "Overtime Paid",
    ->     getDailyCost(PT.Effort_Spent, PR.Hourly_Rate, PR.ot_rate, PR.ot_allowed) AS "Cost Per Day"
    -> FROM Project_Resources AS PR JOIN Project_Transaction AS PT
    ->     ON PR.Project_ID = PT.ProjectID
    ->         AND PR.Resource_ID = PT.Resource_ID;
+------------+-------------+---------------------+-------------------+--------------------+-------------------+---------------+--------------+
| Project_ID | Resource_ID | Entry Date          | Day's Effort (Hr) | Hourly Rate (Cost) | Overtime Rate (%) | Overtime Paid | Cost Per Day |
+------------+-------------+---------------------+-------------------+--------------------+-------------------+---------------+--------------+
|          1 |           3 | 2020-09-12 13:33:24 |             10.00 |               7.00 |             11.00 | Not Paid      |        70.00 |
|          2 |           1 | 2020-09-12 13:33:24 |             22.00 |               8.00 |             20.00 | Not Paid      |       176.00 |
|          2 |           2 | 2020-09-12 13:33:24 |             10.00 |               7.00 |             23.00 | Paid          |        73.22 |
|          2 |           6 | 2020-09-12 13:33:24 |              5.00 |               9.00 |             10.00 | Not Paid      |        45.00 |
|          2 |           7 | 2020-09-12 13:33:24 |              2.00 |              10.00 |             11.00 | Paid          |        13.40 |
|          2 |           8 | 2020-09-12 13:33:24 |             25.00 |               8.00 |             12.00 | Paid          |       216.32 |
+------------+-------------+---------------------+-------------------+--------------------+-------------------+---------------+--------------+
6 rows in set (0.00 sec)
```

P. T. O

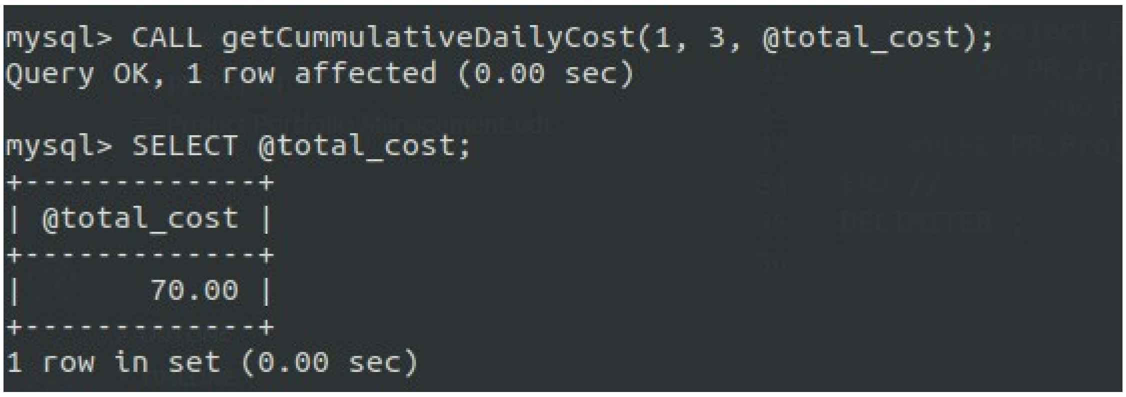
2. Procedure

Use above Function to get the above information and print the total cost for a particular resource for a particular project



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE getCummulativeDailyCost(IN pid INT, IN resid INT, OUT total DECIMAL(20, 2))
    -> BEGIN
    ->     SELECT
    ->         SUM(getDailyCost(PT.Effort_Spent, PR.Hourly_Rate, PR.ot_rate, PR.ot_allowed)) INTO total
    ->     FROM
    ->         Project_Resources AS PR JOIN Project_Transaction AS PT
    ->         ON PR.Project_ID = PT.ProjectID
    ->             AND PR.Resource_ID = PT.Resource_ID
    ->     WHERE PR.Project_ID = pid AND PR.Resource_ID = resid;
    -> END //
Query OK, 0 rows affected (0.16 sec)

mysql> DELIMITER ;
```

Calling Procedure



```
mysql> CALL getCummulativeDailyCost(1, 3, @total_cost);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @total_cost;
+-------------+
| @total_cost |
+-------------+
|       70.00 |
+-------------+
1 row in set (0.00 sec)
```