**Assignment 4**

Aim: To implement the stored procedures and functions

1. Write a procedure to print your name N-number of times

mysql> DELIMITER //

mysql> CREATE PROCEDURE PrintName(IN Name Varchar(30),N INT)

-> BEGIN

-> LOOP\_LABEL:LOOP

-> IF N=0 THEN

-> LEAVE LOOP\_LABEL;

-> END IF;

-> SET N=N-1;

-> SELECT Name;

-> END LOOP;

-> END //

Query OK, 0 rows affected (0.03 sec)

mysql> CALL PrintName("Shri Patil",5);

-> //

+------------+

| Name |

+------------+

| Shri Patil |

+------------+

1 row in set (0.01 sec)

+------------+

| Name |

+------------+

| Shri Patil |

+------------+

1 row in set (0.02 sec)

+------------+

| Name |

+------------+

| Shri Patil |

+------------+

1 row in set (0.02 sec)

+------------+

| Name |

+------------+

| Shri Patil |

+------------+

1 row in set (0.02 sec)

+------------+

| Name |

+------------+

| Shri Patil |

+------------+

1 row in set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

2. Write a procedure to update the cost of the given book. (bookid and newCost are input parameters). Use exception handling.

mysql>DELIMITER //

mysql> CREATE PROCEDURE UPDATECOST(IN ID INT,NEWCOST INT)

-> BEGIN

-> DECLARE EXIT HANDLER FOR SQLEXCEPTION SELECT 'Table not found';

-> UPDATE BOOKS SET BOOKCOST=NEWCOST WHERE BOOK\_ID=ID;

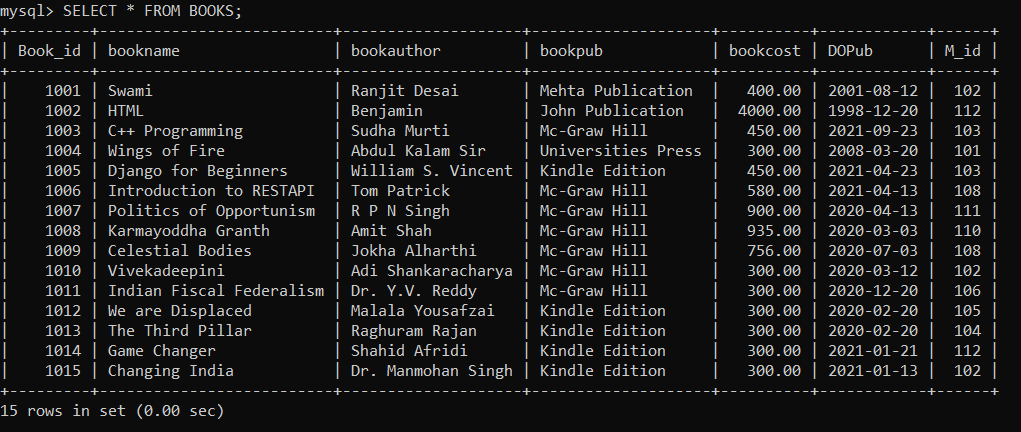
-> END //

Query OK, 0 rows affected (0.01 sec)

mysql>DELIMITER ;

mysql> CALL UPDATECOST(1001,400);

Query OK, 1 row affected (0.02 sec)



3. Write a function which accepts the Member\_Name and returns the number of books issued by him/ her

mysql> CREATE PROCEDURE NO\_BOOKISSUE(IN ID INT,OUT N INT)

-> BEGIN

-> select COUNT(M\_ID) INTO N FROM ISSUE WHERE M\_ID=ID;

-> END //

Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;

mysql> CALL NO\_BOOKISSUE(102,@N);

Query OK, 1 row affected (0.01 sec)

mysql> SELECT @N;

+------+

| @N |

+------+

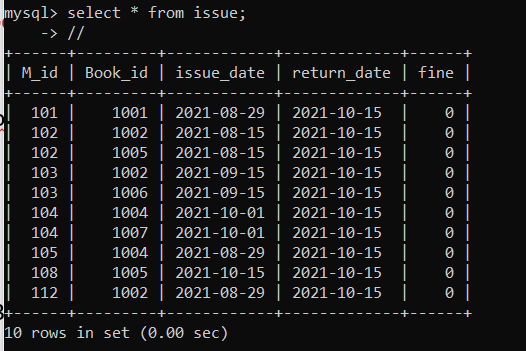
| 2 |

+------+

1 row in set (0.00 sec)

4. Consider the following tables: Books(BNo, Bname, Publisher, cost, DOP, status) IssueReturn(Mem\_ID, BNo, Issue\_Dt, Return\_Dt, fine) Write a procedure Book\_Return which accepts the mem\_ID, BNo and Issue\_Dt as input parameters and performs the book-return operation by resetting the status of the book and calculating the fine. Use exception handling. The procedure should calculate a fine as follows: Check the number of days (from date of issue), - If days are between 15 to 30 then fine amount will be Rs 5 per day. - If no. of days>30, per day fine will be Rs 50 per day & for days > 30, Rs. 5 per day. After submitting the book, status will change from I to R

WithCursor:



CREATE PROCEDURE Book\_Return()

BEGIN

DECLARE MID INT;

DECLARE BID INT;

Declare no\_ofdays int(3) default 0;

Declare fine\_Amt int(3) default 0;

DECLARE STATUSR VARCHAR(3) default 'I';

DECLARE I\_DATE DATE;

DECLARE is\_done INT DEFAULT 0;

DECLARE C1 CURSOR FOR SELECT issue.M\_id,issue.Book\_id,issue\_date,status FROM ISSUE,books where books.book\_id=issue.book\_id;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET is\_done=1;

OPEN C1;

L1:LOOP

FETCH C1 INTO MID,BID,I\_DATE,STATUSR;

IF is\_done=1 THEN

LEAVE L1;

END IF;

select datediff(curdate(),I\_date) INTO no\_ofdays;

IF (no\_ofdays>=15 AND no\_ofdays<=30 AND STATUSR='I') THEN

set FINE\_AMT=no\_ofdays\*5;

elseif (no\_ofdays>30 AND STATUSR='I') THEN

set Fine\_AMT=(15\*5)+((no\_ofdays-15)\*50);

END IF;

update issue set fine=FINE\_AMT,return\_date=Curdate() where book\_id=BID and m\_id=MID;

update books set status='R' where book\_id=BID;

END LOOP L1;

CLOSE C1;

SELECT 'SUCCESSFULL';

END //

mysql> delimiter ;

mysql> call book\_return();

+-------------+

| SUCCESSFULL |

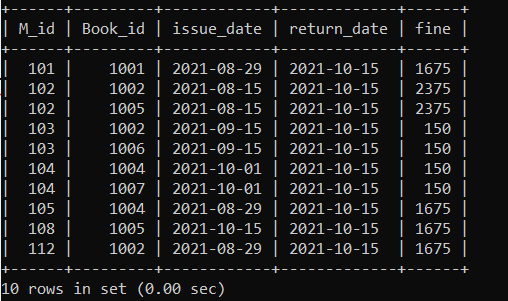
+-------------+

| SUCCESSFULL |

+-------------+

1 row in set (0.08 sec)

mysql> select \* from issue;

s

Without cursor:

CREATE PROCEDURE Book\_Return(IN MID int ,BID int,I\_date date)

BEGIN

Declare no\_ofdays int default 0;

Declare fine\_Amt int default 0;

DECLARE STATUSR VARCHAR(30) default 'I';

DECLARE is\_done INT DEFAULT 0;

SELECT status into STATUSR FROM books where book\_id=bid;

select datediff(curdate(),I\_date) INTO no\_ofdays;

IF (no\_ofdays>=15 AND no\_ofdays<=30 AND STATUSR='I') THEN

set FINE\_AMT=no\_ofdays\*5;

elseif (no\_ofdays>30 AND STATUSR='I') THEN

set Fine\_AMT=(15\*5)+((no\_ofdays-15)\*50);

END IF;

update issue set fine=FINE\_AMT,return\_date=Curdate() where book\_id=BID and m\_id=MID;

update books set status='R' where book\_id=BID;

SELECT 'SUCCESSFULL';

END //

5. Create table Result(RNo, Sname,tot-marks, class) Populate the table with total marks out of 1500. Put NULL in the class field. Write a stored procedure to fill the class of each student as follows: class = Distinction if marks990 class = First Class if marks900 class = Higher second class if marks825 class = Second class if marks600 If marks<600 then class=fail.

mysql> create table result(RNo int primary key,Sname varchar(3)not null,Total\_marks int not null ,class varchar(3));

Query OK, 0 rows affected (0.12 sec)

mysql> Insert into resultT values(1,"gayatri",990,"ND");

Query OK, 1 row affected (0.05 sec)

mysql> Insert into resultT values(2,"Dhiraj",1340,"ND");

Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(3,"Akhila",1110,"ND");

Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(4,"Mandar",1140,"ND");

Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(5,"shreya",410,"ND");

Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(6,"Vishnu",890,"ND");

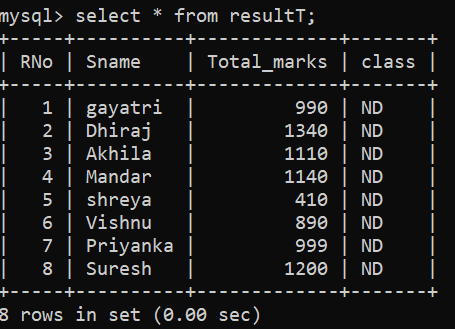
Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(7,"Priyanka",999,"ND");

Query OK, 1 row affected (0.01 sec)

mysql> Insert into resultT values(8,"Suresh",1200,"ND");

Query OK, 1 row affected (0.01 sec)



With cursor:

mysql> CREATE PROCEDURE set\_class()

-> BEGIN

-> DECLARE mks int;

-> DECLARE no int;

-> DECLARE is\_done INT DEFAULT 0;

-> DECLARE C1 CURSOR FOR SELECT rno,total\_marks from resultt;

-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET is\_done=1;

-> OPEN C1;

-> L1:LOOP

-> FETCH C1 INTO no,mks;

-> IF is\_done=1 THEN

-> LEAVE L1;

-> END IF;

-> IF mks<=1500 and mks>=990 then

-> UPDATE resultT SET class="DISTINCTION" WHERE RNo=no;

-> ELSEIF mks>=900 and mks<990 then

-> UPDATE resultT SET class="FIRST CLASS" WHERE RNo=no;

-> ELSEIF mks>=825 and mks<900 then

-> UPDATE resultT SET class="HIGHER SECOND CLASS" WHERE RNo=no;

-> ELSEIF mks>=600 and mks<825 then

-> UPDATE resultT SET class="SECOND CLASS" WHERE RNo=no;

-> ELSE

-> UPDATE resultT SET class="FAIL" WHERE RNo=no;

-> END IF;

-> END LOOP L1;

-> CLOSE C1;

-> SELECT 'SUCCESSFUL';

-> END

-> //

Query OK, 0 rows affected (0.01 sec)

mysql> call set\_class();

-> //

+- - - - - - - - - - - - +

| SUCCESSFUL |

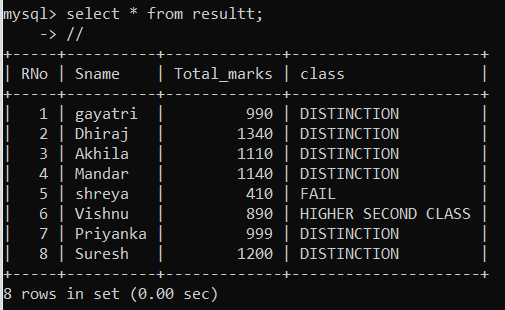
+- - - - - - - - - - - - +

| SUCCESSFUL |

+ - - - - - - - - - - - - +

1 row in set (0.03 sec)

Query OK, 0 rows affected (0.04 sec)



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Without cursor:

mysql> CREATE PROCEDURE CLASSCREATION(IN MID INT)

-> BEGIN

-> DECLARE marks INT;

-> SELECT Total\_marks INTO marks FROM RESULTT WHERE RNo=MID;

-> IF marks>=990 and marks<=1500 then

-> UPDATE resultT SET class="DISTINCTION" WHERE RNo=MID;

-> ELSEIF marks>=900 AND marks<990 then

-> UPDATE resultT SET class="FIRST CLASS" WHERE RNo=MID;

-> ELSEIF marks>=825 AND marks<900 then

-> UPDATE resultT SET class="HIGHER SECOND CLASS" WHERE RNo=MID;

-> ELSEIF marks>=600 AND marks<825 then

-> UPDATE resultT SET class="SECOND CLASS" WHERE RNo=MID;

-> ELSE

-> UPDATE resultT SET class="FAIL" WHERE RNo=MID;

-> END IF;

-> END //

Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;

mysql> CALL CLASSCREATION(1);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(2);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(3);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(4);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(5);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(6);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(7);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CLASSCREATION(8);

Query OK, 1 row affected (0.01 sec)

mysql> SELECT \* FROM RESULTT;

