**LAB CYCLE 3**

**Experiment No. 5**

**20MCA134 ADVANCED DBMS LAB**

**Familiarization of Stored Procedure, Function, Cursor and Triggers**

1. Write a stored procedure to read three numbers and find the greatest among them.

**Source code**:

DROP PROCEDURE IF EXISTS Greatest;

DELIMITER //

CREATE PROCEDURE Greatest(IN A INT, IN B INT, IN C INT, OUT D INT)

BEGIN

IF A>=B AND A>=C THEN

SET D=A;

ELSEIF B>=C THEN

SET D=B;

ELSE

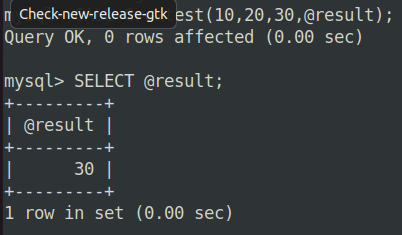
SET D=C;

END IF;

END//

DELIMITER ;

**Output**:



2. Write a stored procedure to read two numbers and print all the numbers between them.

**Source code**:

DROP PROCEDURE IF EXISTS PrintNumbersBetween;

DELIMITER //

CREATE PROCEDURE PrintNumbersBetween(IN A INT, IN B INT)

BEGIN

DECLARE RESULT VARCHAR (100);

DECLARE I INT;

SET I=A+1;

SET RESULT="";

WHILE I<B DO

SET RESULT=CONCAT(RESULT, I, " ");

SET I=I+1;

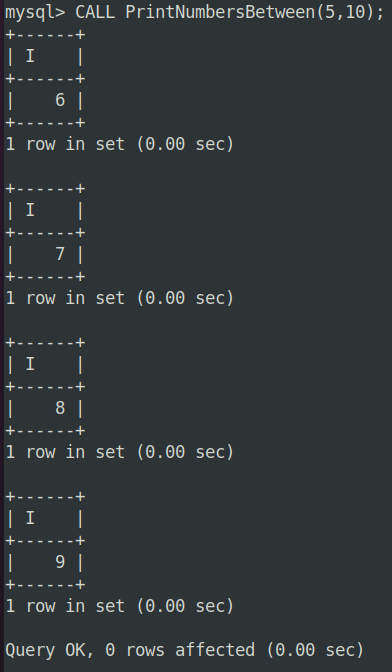
END WHILE;

SELECT RESULT;

END//

DELIMITER ;

**Output**:



3. Write a stored procedure to read N and find the sum of the series 1+2+3 +... N

**Source code**:

DROP PROCEDURE IF EXISTS SumOfN;

DELIMITER //

CREATE PROCEDURE SumOfN(IN N INT)

BEGIN

DECLARE I INT;

DECLARE SUM INT;

SET I=1;

SET SUM=0;

WHILE I<=N DO

SET SUM=SUM+I;

SET I=I+1;

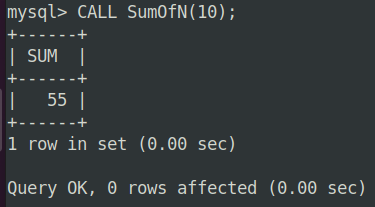
END WHILE;

SELECT SUM;

END//

DELIMITER ;

**Output**:



4. Write a stored procedure to read a mark and display the grade

**Source code**:

DROP PROCEDURE IF EXISTS MarkGrade;

DELIMITER //

CREATE PROCEDURE MarkGrade(IN MARK INT)

BEGIN

DECLARE GRADE VARCHAR (1);

IF MARK>90 THEN

SET GRADE='S';

ELSEIF MARK>80 THEN

SET GRADE='A';

ELSEIF MARK>70 THEN

SET GRADE='B';

ELSEIF MARK>60 THEN

SET GRADE='C';

ELSEIF MARK>50 THEN

SET GRADE='D';

ELSE

SET GRADE='F';

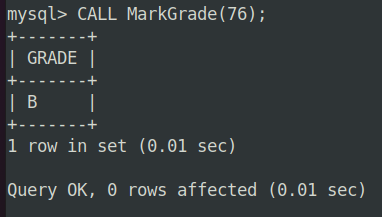
END IF;

SELECT GRADE;

END//

DELIMITER ;

**Output**:



5. Write a stored procedure to read a number and invert the given number

**Source code**:

DROP PROCEDURE IF EXISTS InvertNumber;

DELIMITER //

CREATE PROCEDURE InvertNumber(IN NUM INT)

BEGIN

DECLARE INVERSE INT DEFAULT 0;

DECLARE REM INT;

WHILE NUM>0 DO

SET REM=NUM%10;

SET NUM=NUM DIV 10;

SET INVERSE=(INVERSE\*10)+REM;

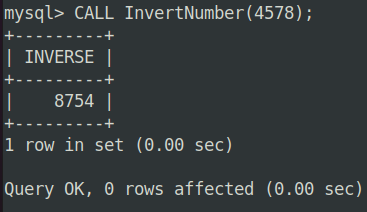
END WHILE;

SELECT INVERSE;

END//

DELIMITER ;

**Output**:



6. Create a procedure which will receive account\_id and amount to withdraw. If the account does not exist, it will display a message. Otherwise, if the account exists, it will allow the withdrawal only if the new balance after the withdrawal is at least 1000.

**Source code**:

**Output**:

7.

**Source code**:

**Output**:

8.

**Source code**:

**Output**: