/*Assignment 1 */

/* Learn about how to Create Procedures,Variables and nested statements with BEGIN & END statements */

/ * Step1 : Create the following procedure in MySQL editor */

```
DELIMITER $$
CREATE PROCEDURE myproc_local _Variables()
BEGIN   /* declare local variables */
DECLARE a INT DEFAULT 10;
DECLARE b, c INT;    /* using the local variables */
SET a = a + 100;
SET b = 2;
SET c = a + b;
BEGIN      /* local variable in nested block */
  DECLARE c INT;
  SET c = 5;
  /* local variable c takes precedence over the one of the same name declared in the enclosing block. */
  SELECT a, b, c;
END;
SELECT a, b, c;
END$$
```

/* Call the procedure */
/* Step 2. Run the call statment */

```
call myproc_Local_Variables() $$
```

/* Assignment 2 */

/* Use user variables. They are referenced with an ampersand (@) prefixed to the user variable name */
/* Step 1 */

```
DELIMITER $$
CREATE PROCEDURE myproc_User_Variables()
BEGIN
SET @x = 15;
SET @y = 10;
SELECT @x, @y, @x-@y;
END$$
```

/* step2 */

Call procedure myproc_User_Variables()

/* Assignment 3 */

```
DROP TABLE products;

CREATE  Table products(prod_id  int,Prod_name varchar(30),Prod_cost  int,prod_location  int);

INSERT INTO products Values(100, 'Pencil', 3500, 20);
INSERT INTO products Values(120, 'Book', 500, 25) ;
INSERT INTO products Values(130, 'Table', 750, 20);
INSERT INTO products Values(145, 'chair', 250, 30);
```

```
SELECT * FROM products;

/* Different ways of sending the parameters through the stored procedures  */

/* 3.1 Step 1. parameters sending through procedure . var1 takes the value from the call statement below. eg: var1 = 2      */

DELIMITER $$ ;
CREATE PROCEDURE myproc_IN (IN var1 INT)  BEGIN
   SELECT * FROM products LIMIT var1;
END$$

/* 3.1 Step 2.    send parameter as  number of rows to be displayed  */

CALL myproc_in(2)$$


/*  3.2  Step1 : parameters sending through procedure  */

CREATE PROCEDURE my_userproc_out(OUT highest_cost INT, OUT NumOfItems INT)
BEGIN
  SELECT MAX(Prod_cost) INTO highest_cost FROM products;
  SELECT COUNT(Prod_name) INTO NumOfItems FROM products WHERE prod_location = 20;
END$$

/* 3.2  Step2 :send the value from terminal.Here @M takes value from the terminal ans send as parameter to procedure variable highest_cost  */

 CALL my_userproc_out(@Val1, @Val2) $$
 SELECT @Val1, @Val2 $$

/*  3.3 Case statements  */
/* In this procedure,pcost variable is passed as value through IN parameter. Within the procedure, there is CASE statement along with two WHEN and an ELSE which will test the condition and return the count value in no_items */
/* Step1 */

DELIMITER $$ ;
CREATE PROCEDURE Prod_case(INOUT no_items INT, IN pcost INT)
BEGIN
  CASE
  WHEN (pcost>500) THEN (SELECT COUNT(prod_id) INTO no_items FROM products WHERE
prod_cost>500);
  WHEN (pcost<1000) THEN (SELECT COUNT(prod_id) INTO no_items FROM products WHERE
prod_cost<1000);
  ELSE (SELECT COUNT(prod_id) INTO no_items  FROM products WHERE prod_cost=3000);
  END CASE;
END$$


/* Step 2. Run the following statements. 500,1000 are  sent as values to pcost.  @C is passed as OUT parameter  */
CALL Prod_case(@C,500) $$
SELECT @C $$
CALL Prod_case(@C,1000) $$
SELECT @C $$
```