EXERCISE 5

Date: 14/03/2024

Create a database **STORE** with table named **Product**(PdtId, PName, Price, Quantity) and create a stored procedure called **Insertproduct** that inserts a new product into the database, under some conditions. The stored procedure has input parameters (barcode, product name, price, and quantityInStock). The shared procedure should insert a row in the Product table only if the price is greater than 0 and the quantity is greater or equal to 0. If the conditions are not satisfied, the stored procedure just terminates (no errors generated).

Table structure of Product: -

Field	Type	Null	Key	Default	Extra
PdtId	varchar(4)	NO	PRI	NULL	
PName	varchar(20)	YES		NULL	
Price	double(5,2)	YES		NULL	
Qunatity	int	YES		NULL	

Inserted Values: -

PdtId	PName	Price	Qunatity
1001	Pen	10.5	10
1002	Pencil	5.75	15
1003	Book	25	5

SQL Query to create database and table:

- Create Database

➢ CREATE DATABASE STORE;
➢ USE STORE;

- Create Table: Product

➢ CREATE TABLE Product(PdtId varchar(4) primary key,PName varchar(20), Price double(5,2),Qunatity int(3));
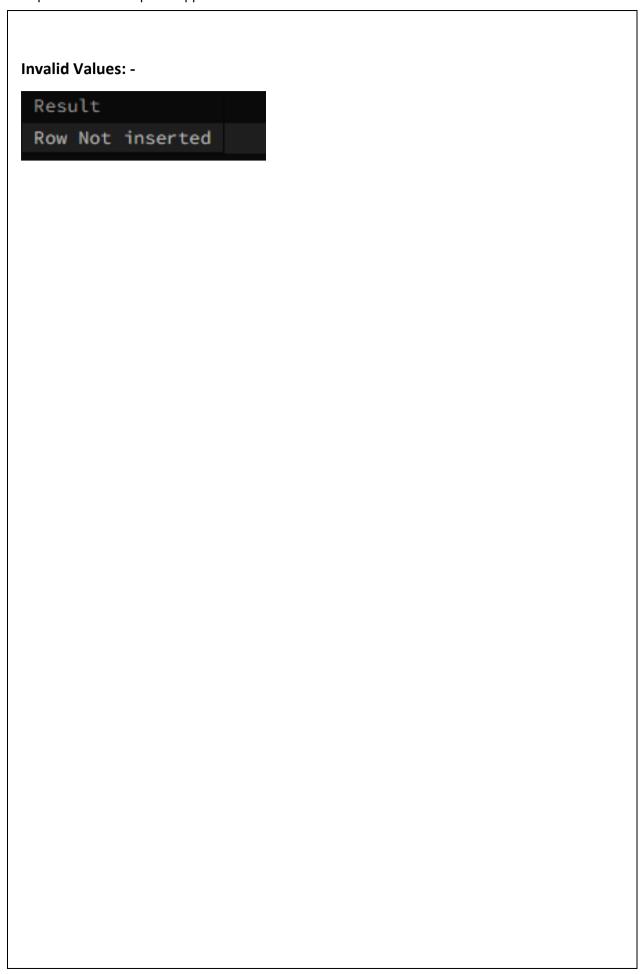
Procedure to insert values to the table:

```
CREATE PROCEDURE `Insertproduct`(in barcode int,in name
varchar(20),in price double(5,2), in quantity int(3))
BEGIN
    declare Result varchar(20);
        if price>0 and quantity >0
        then
            insert into Product values
(barcode,name,price,quantity);
        else
            set Result= "Row Not inserted";
            select Result;
        end if;
END
```

SQL Query to call the procedure:

➢ call Insertproduct(1001,"Pen",10.5,10);
➢ call Insertproduct(1002,"Pencil",5.75,15);
➢ call Insertproduct(1003,"Book",25,5);

➢ select * from Product;

Invalid Values: -

```
Result

Row Not inserted
```

- Inserting Invalid Values

➢ call Insertproduct(1004,"Eraser",-2,20);
➢ call Insertproduct(1005,"Paper",1.25,-10);
➢ call Insertproduct(1006,"Sharpner",-2.5,-20);

RESULT: -

Procedure and queries are executed and output is verified.