Project on: Fashion store

Name: Hanisha Jain

Batch: 2020-21

CERTIFICATE

CLASS: XII YEAR: 2020-2021

This is to certify that Fashion Store Project is successfully completed by **Hanish Jain** of Class: XII, from 2020-21 batch

This project was solely completed and code by

Hanisha Jain .

Head Teacher	External	Internal Examiner
Signature:	Examiner	(Subject Teacher)

Date: / / 21 Department of: COMPUTER SCI.

Principal

Acknowledgement:

I, Hanisha Jain of class XIIth

would like to express my sincere gratitude to my computer science teacher Mrs.Pooja, PGT COMPUTER SCIENCE, for her vital support, guidance and encouragement – without which this project would not have come forth.

We would also like to express my gratitude to my school Poddar Brio International for letting me use the school laboratory.

INDEX

Fashion stores

BRIEF OVERVIEW OF PROJECT..

The main objective of the python project on fashion store management is to manage the details of sales, discounts, payments, products, and inventory digitally.

The project is totally built at administrative end and only administrator is guaranteed the access.

The purpose of the project is to build an application program to reduce the manual work for managing the sales, discounts, stock, and payments.

It tracks all the details about stocks, products, and inventory; it also prints various reports as per input given by the user.

INPUT DATA AND VALIDATION OF PROJECT

1. All the fields such as sales payments discounts are validated and does not take invalid values.
2. Each form of sales, discounts, stock cannot accept the blank values.
3. Avoiding errors in data.
4. Controlling amount of input.

SOFTWARE AND HARDWARE REQUIREMENTS:

Data file handling has been effectively used in the program. The database is a collection of interrelated data to serve multiple applications. That is database programs create files of information. So we see that files are worked with most, inside the program.

DBMS: The software required for the management of data is called as DBMS. It has3 models:

- Relation model
- Hierarchical model
- Network model

RELATIONAL MODEL: It's based on the concept on relation. Relation is the table that consists of rows and columns. The rows of the table are called tuple and the columns of the table are called attribute. Numbers of rows in the table is called as cardinality. Number of columns in the table is called as degree.

HIERARCHICAL MODEL: In this type of model, we have multiple records for each record. A particular record has one parent record. No chide record can exist without parent record. In this, the records are organized in tree.

NETWORK MODEL: In this, the data is represented by collection of records and relationship is represented by (ink or association.

CHARACTERISTICS OF DBMS:

- It reduces the redundancy
- Reduction of data in inconsistency
- Data sharing
- Data standardization

DIFFERENT TYPES OF FILES: -BASED ON ACCESS:

- Sequential file
- Serial file
- Random (direct access) file BASED ON STORAGE:-
- Text file
- Binary File

NEED OF COMPUTERISATION

Over the decades computers and fashion have developed gradually, changed with time, taste and trend. But nobody knew that a time will come when both these fields will complement each other so well. Today fashion design has reached new heights by computer aided methods of design. As a result of which, computer industry has got its new customer. Computer technology is making waves in the fashion design zone. From determining textile weaves to sizing designs; computers are a vital component of the fashion industry. Computer aided design (CAD) programs reduce the demand for manual sketches. New software programs continue to replace old manual skills. Going by the wayside are "old fashioned" flat pattern construction, pencil sketching and traditional math-based pattern sizing. Those who lag in math and falter at sketching can now breathe a little easier. Manually figuring size adjustments and cutting pattern pieces instils that knowledge. Software programs constantly evolve. A program used today may be obsolete within several years. Being trained on today's software does not guarantee it will be used when you are ready to go out into the field. Understanding calculations is timeless, as is computer competency. Software, however, shifts rapidly.

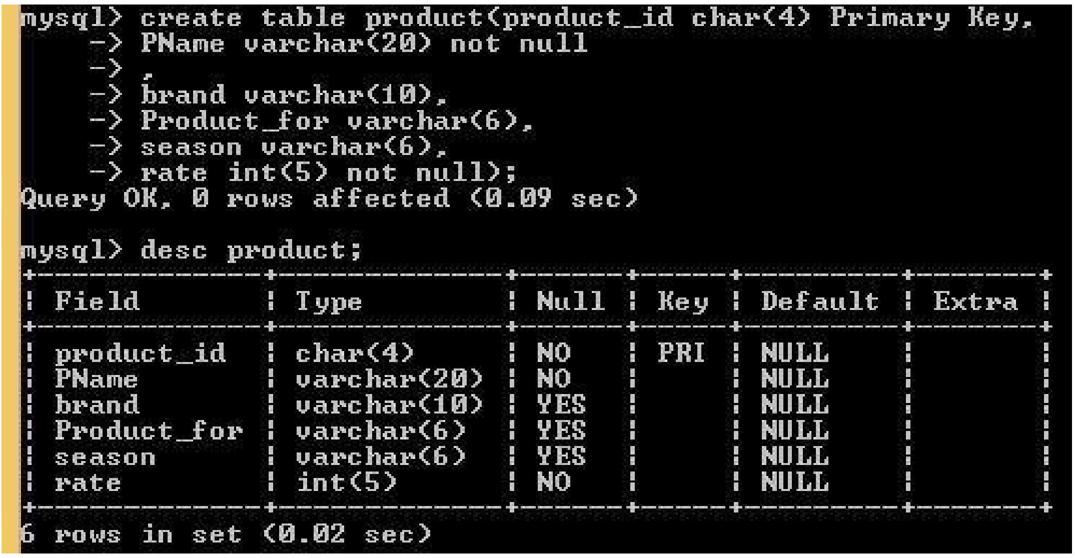
1. It generates the report on sales, discounts and stocks.
2. Provides filter report on payments, inventory and products.
3. We can easily export PDF on sales, products and stocks.
4. Applications can also provide excel export for sales and discounts.
5. It deals with monitoring the information and transaction of products.
6. It increases the efficiency of managing sales and discount.
7. It has higher efficiency of editing, adding and updating of records.
8. Provides the searching facilities on various factors.

1. Excel export has not been developed for stocks and products.
2. The transactions are executed in offline mode only.
3. Online transactions for sales, discounts, or other data modifications are not possible.
4. Offline reports of sales, products, discounts and stocks cannot be generated due to batch mode execution.

Source code screening…

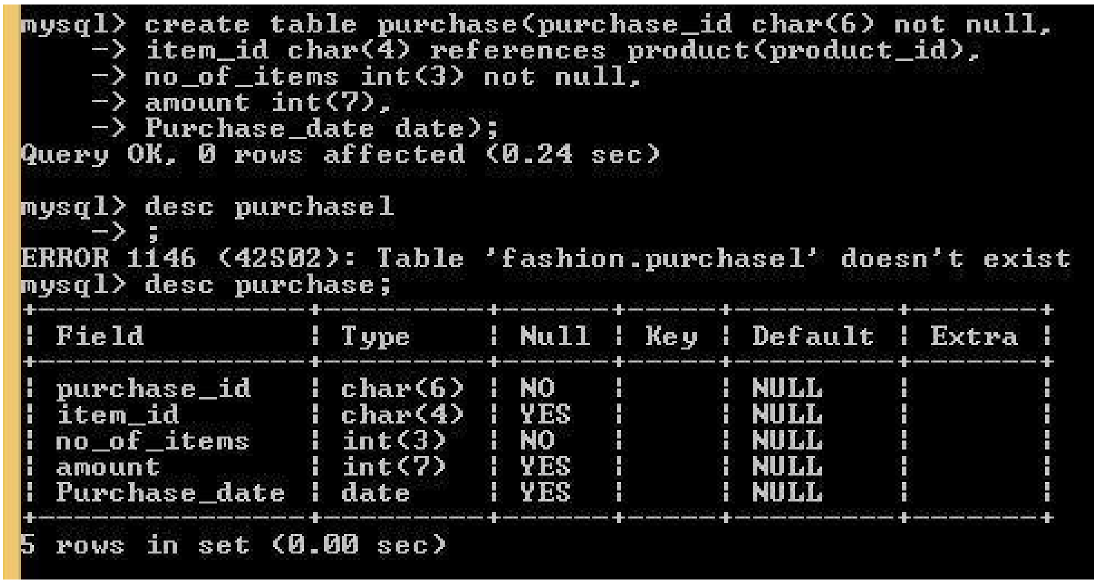
DBMS: MySQL
Host: local host
User: root
Pass: root
Database: fashion
Table Structure: (Images Bellow)

1. ***Product table***



```
mysql> create table product(product_id char(4) Primary Key,
    -> PName varchar(20) not null
    -> ,
    -> brand varchar(10),
    -> Product_for varchar(6),
    -> season varchar(6),
    -> rate int(5) not null);
Query OK, 0 rows affected (0.09 sec)

mysql> desc product;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| product_id  | char(4)     | NO   | PRI | NULL    |       |
| PName       | varchar(20) | NO   |     | NULL    |       |
| brand       | varchar(10) | YES  |     | NULL    |       |
| Product_for | varchar(6)  | YES  |     | NULL    |       |
| season      | varchar(6)  | YES  |     | NULL    |       |
| rate        | int(5)      | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

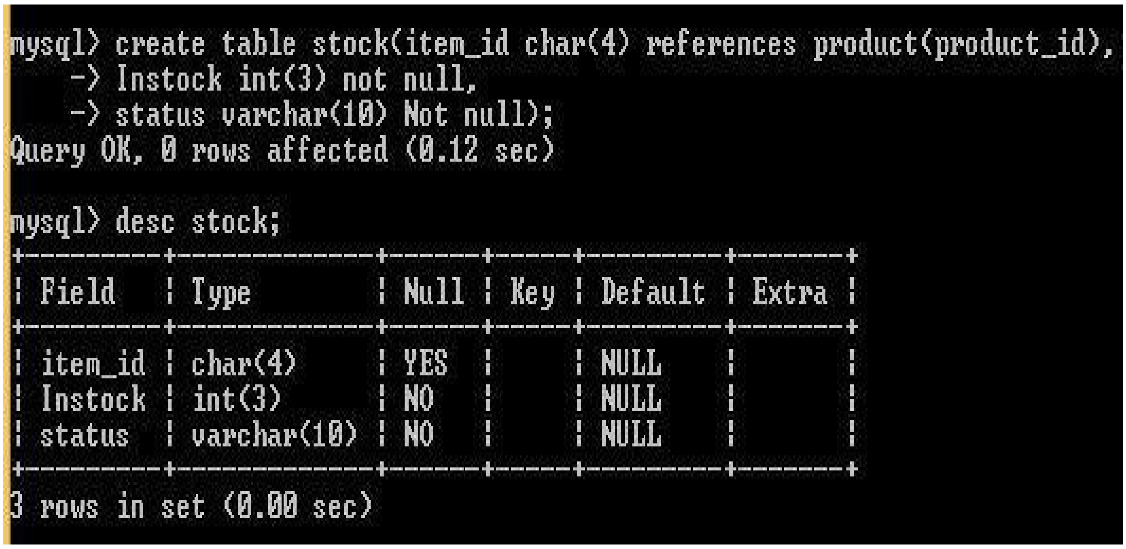
2. *Purchase table*



```
mysql> create table purchase(purchase_id char(6) not null,
    -> item_id char(4) references product(product_id),
    -> no_of_items int(3) not null,
    -> amount int(7),
    -> Purchase_date date);
Query OK, 0 rows affected (0.24 sec)

mysql> desc purchasel
    -> ;
ERROR 1146 (42S02): Table 'fashion.purchasel' doesn't exist
mysql> desc purchase;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| purchase_id   | char(6) | NO   |     | NULL    |       |
| item_id       | char(4) | YES  |     | NULL    |       |
| no_of_items   | int(3)  | NO   |     | NULL    |       |
| amount        | int(7)  | YES  |     | NULL    |       |
| Purchase_date | date    | YES  |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

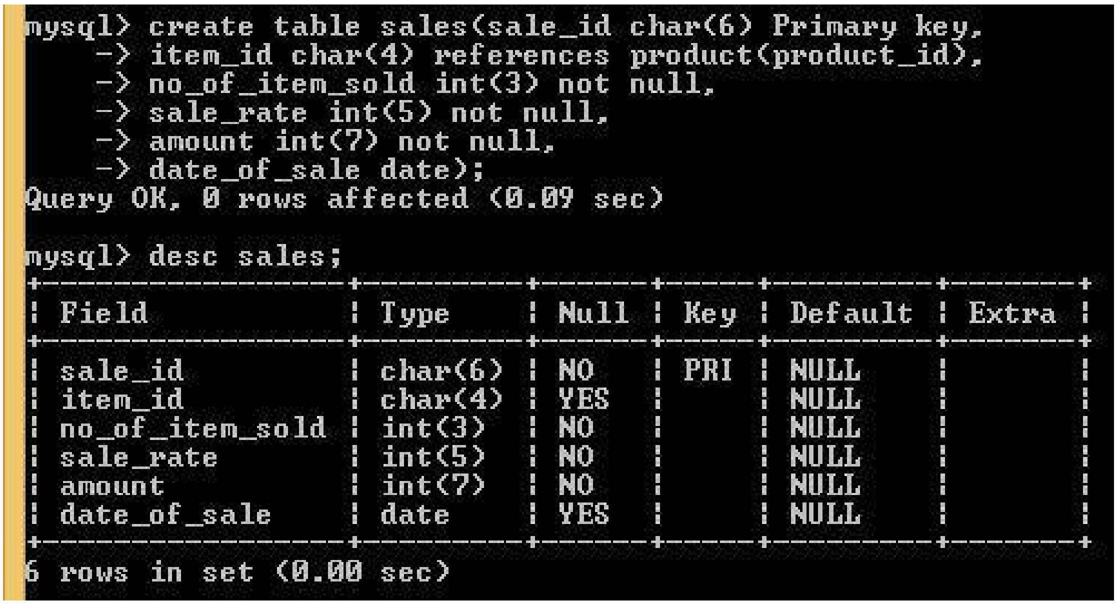
Note: In Purchase table take the purchase ID as varchar (16)

3. *Stock table*



```
mysql> create table stock(item_id char(4) references product(product_id),
    -> Instock int(3) not null,
    -> status varchar(10) Not null);
Query OK, 0 rows affected (0.12 sec)

mysql> desc stock;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| item_id | char(4)     | YES  |     | NULL    |       |
| Instock | int(3)      | NO   |     | NULL    |       |
| status  | varchar(10) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4. ***Purchase table***



```
mysql> create table sales(sale_id char(6) Primary key,
    -> item_id char(4) references product(product_id),
    -> no_of_item_sold int(3) not null,
    -> sale_rate int(5) not null,
    -> amount int(7) not null,
    -> date_of_sale date);
Query OK, 0 rows affected (0.09 sec)

mysql> desc sales;
+-----------------+---------+------+-----+---------+-------+
| Field           | Type    | Null | Key | Default | Extra |
+-----------------+---------+------+-----+---------+-------+
| sale_id         | char(6) | NO   | PRI | NULL    |       |
| item_id         | char(4) | YES  |     | NULL    |       |
| no_of_item_sold | int(3)  | NO   |     | NULL    |       |
| sale_rate       | int(5)  | NO   |     | NULL    |       |
| amount          | int(7)  | NO   |     | NULL    |       |
| date_of_sale    | date    | YES  |     | NULL    |       |
+-----------------+---------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Note: In Purchase table take the sale_id as varchar (16)

Python code:

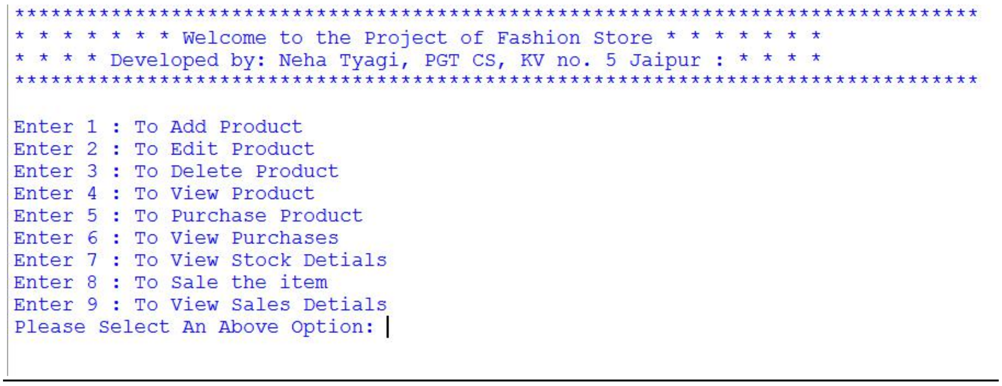
```
#NAME: Hanisha Jain
#Subject: Computer Science
#Coded, completed and submitted by: Hanisha Jain
#Submitted to: Pooja Ma'am

import os
import platform
import mysql.connector
import pandas as pd
import datetime

mydb=mysql.connector.connect(host="localhost",\
        user="root",\
        passwd="root",\
        database="fashion")
mycursor=mydb.cursor()

def AddProduct():
        L=[]
        stk=[]
        pid=input("Enter the Product ID : ")
        L.append(pid)
        IName=input("Enter the Product Name : ")

        L.append(IName)
        brnd=input("Enter the Product Brand Name : ")
        L.append(brnd)
        fr=input("Enter Male/Female/Kids : ")
        L.append(fr)
        sn=input("Enter Winter/Summer : ")
        L.append(sn)
        rate=int(input("Enter the Rates for Product :"))
        L.append(rate)
        product=(L)
        sql="Insert into product
        (product_id,PName,brand,Product_for,Season,rate)values(%s,%s,%s,%s,%s,%s)"
        mycursor.execute(sql,product)
        mydb.commit()
        stk.append(pid)
        stk.append(0)
        stk.append("No")
        st=(stk)
        sql="insert into stock(item_id, Instock, status) values(%s,%s,%s)"
        mycursor.execute(sql,st)
        mydb.commit()
        print("One Product inserted ")
```

```
def EditProduct():
    pid=input("Enter product ID to be edited : ")
    sql="select * from product where product_id=%s"
    ed=(pid,)
    mycursor.execute(sql,ed)
    res=mycursor.fetchall()
    for x in res:
        print(x)
    print("")
    fld=input("Enter the field which you want to edit : ")
    val=input("Enter the value you want to set : ")
    sql="Update product set " + fld +"='" + val + "' where product_id='" + pid + "'"
    sq=sql
    mycursor.execute(sql)
    print("Editing Don : ")
    print("After correction the record is : ")
    sql="select * from product where product_id=%s"
    ed=(pid,)
    mycursor.execute(sql,ed)
    res=mycursor.fetchall()
    for x in res:
        print(x)
    mydb.commit()

def DelProduct():
    pid=input("Enter the Product)id to be deleted : ")
    sql="delete from sales where item_id=%s"
    id=(pid,)
    mycursor.execute(sql,id)
    mydb.commit()
    sql="delete from purchase where item_id=%s"
    mycursor.execute(sql,id)
    mydb.commit()
    sql="delete from stock where item_id=%s"
    mycursor.execute(sql,id)
    mydb.commit()
    sql="delete from product where product_id=%s"
    mycursor.execute(sql,id)
    mydb.commit()
    print("One Item Deleted")

def ViewProduct():
    print("Display Menu: Select the category to display the data")
    print("1. All Details")
    print("2. Product Name:")
    print("3. Product Brand:")
    print("4. Product For:")
    print("5. Product Season:")
    print("6. Product ID:")
    x=0
    ch=int(input("Enter your choice to display : "))
    if ch==1:
        sql="select * from product"
        mycursor.execute(sql)
        res=mycursor.fetchall()
        for x in res:
            print(x)
        x=1
    elif ch==2:
        var='PName'
        val=input("Enter the name of Product : ")
    elif ch==3:
        var='brand'
        val=input("Enter the name of Brand : ")
    elif ch==4:
        var='Product_for'
        val=input("Enter Male/Femal/Kids : ")
    elif ch==5:
        var='season'

        val=input("Enter the Season : ")
    elif ch==6:
        var='product_id'
        val=input("Enter the Product_id : ")
    if x==0:
        sql="select * from product where " + var + " = %s"
        sq=sql
        tp=(val,)
        mycursor.execute(sq,tp)
        res=mycursor.fetchall()
        for x in res:
            print(x)

def PurchaseProduct():
    mn=""
    dy=""
    now=datetime.datetime.now()
    purchaseID="P"+str(now.year)+str(now.month)+str(now.day)+str(now.hour)+str(now.minut
    e)+str(now.second)
    L=[]
    Lst=[]
    L.append(purchaseID)
    itemId=input("Enter Product ID : ")
    L.append(itemId)
    itemNo=int(input("Enter the number of Items : "))
    L.append(itemNo)
    sql="select rate from product where product_id=%s"
    pid=(itemId,)
    mycursor.execute(sql,pid)
    res=mycursor.fetchone()
    for x in res:
        print("rate is : ", x)
    amount=x*itemNo
    print("Amount is :", amount)
```

```
        L.append(amount)
        mnth=now.month
        if mnth<=9:
                mn="0"+str(mnth)
        else:
                mn=str(mnth)
        day=now.day
        if day<=9:
                dy="0"+str(day)
        else:
            dy=str(day)

            dt=str(now.year)+"-"+mn+"-"+dy
            L.append(dt)
            tp=(L)
            sql="insert into
            purchase(purchase_id,item_id,no_of_items,amount,Purchase_date)values(%s,%s,%s,%s,
            %s)"
            mycursor.execute(sql,tp)
            mydb.commit()
            sql="Select Instock from stock where item_id=%s"
            mycursor.execute(sql,pid)
            res=mycursor.fetchall()
            status="No"
            for x in res:
                print(x)
             instock=x[0]+itemNo
             if instock>0:
                status="Yes"
             Lst.append(instock)
             Lst.append(status)
            Lst.append(itemId)
            tp=(Lst)
           sql="update stock set instock=%s,status=%s where item_id=%s"
           mycursor.execute(sql,tp)
           mydb.commit()
          print("1 Item purchased and saved in Database")

def ViewPurchase():
        item=input("Enter Product Name : ")
        sql="select product.product_id,
        product.PName,product.brand,purchase.no_of_items,purchase.purchase_date,purchase.am
        ount from product INNER JOIN purchase ON product.product_id=purchase.item_id and
        product.PName=%s"
        itm=(item,)
        mycursor.execute(sql,itm)
        res=mycursor.fetchall()
        for x in res:
                print(x)

def ViewStock():
        item=input("Enter Product Name : ")
        sql="select product.product_id,product.PName,stock.Instock,\
        stock.status from stock, product where \
        product.product_id=stock.item_id and product.PName=%s"
        itm=(item,)
        mycursor.execute(sql,itm)
        res=mycursor.fetchall()
        for x in res:
            print(x)

def SaleProduct():
        now=datetime.datetime.now()
        saleID="S"+str(now.year)+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+str
        (now.second)
        L=[]
        L.append(saleID)
        itemId=input("Enter Product ID : ")
        L.append(itemId)
        itemNo=int(input("Enter the number of Items : "))
        L.append(itemNo)
        sql="select rate from product where product_id=%s"
        pid=(itemId,)
        mycursor.execute(sql,pid)
        res=mycursor.fetchall()
        for x in res:
                print("The rate of item is :",x)
        dis=int(input("Enter the discount : "))
        saleRate=x[0]-(x[0]*dis/100)
        L.append(saleRate)
        amount=itemNo*saleRate
        L.append(amount)
        mnth=now.month
                if mnth<=9:
                        mn="0"+str(mnth)
                else:
                        mn=str(mnth)
                day=now.day
                if day<=9:
                        dy="0"+str(day)
                else:
                        dy=str(day)
                dt=str(now.year)+"-"+mn+"-"+dy
                L.append(dt)
                tp=(L)
                sql="insert into sales (sale_id, item_id,no_of_item_sold,\
                sale_rate,amount,date_of_sale) values(%s,%s,%s,%s,%s,%s)"
                mycursor.execute(sql,tp)
                mydb.commit()
                sql="Select Instock from stock where item_id=%s"
                mycursor.execute(sql,pid)
                res=mycursor.fetchall()
                for x in res:
                        print("Total Items in Stock are : ",x)
                        instock=x[0]-itemNo

                    if instock>0:
                        status="Yes"
                    tp=(instock,status,itemId)
                    sql="update stock set instock=%s,status=%s where item_id=%s"
                    print("Remaining Items in Stock are : ",instock)
                    mycursor.execute(sql,tp)
                    mydb.commit()

def ViewSales():
        item=input("Enter Product Name : ")
        sql="select product.product_id, product.PName,product.brand,\
        sales.no_of_item_sold,sales.date_of_sale,sales.amount \
        from sales, product where product.product_id=sales.item_id \
        and product.PName=%s"
        itm=(item,)
        mycursor.execute(sql,itm)
        res=mycursor.fetchall()
        for x in res:
                print(x)

def MenuSet(): #Function For The SFashion Store System
        print("Enter 1 : To Add Product ")
        print("Enter 2 : To Edit Product ")
        print("Enter 3 : To Delete Product ")
        print("Enter 4 : To View Product ")
        print("Enter 5 : To Purchase Product")
        print("Enter 6 : To View Purchases")
        print("Enter 7 : To View Stock Detials")
        print("Enter 8 : To Sale the item")
        print("Enter 9 : To View Sales Detials")
        try: #Using Exceptions For Validation
            userInput = int(input("Please Select An Above Option: ")) #Will Take Input From User
        except ValueError:
             exit("\nHy! That's Not A Number") #Error Message
        else:
                print("\n") #Print New Line
        if(userInput == 1):
                AddProduct()
        elif(userInput == 2):
                EditProduct()
        elif (userInput==3):
                DelProduct()
        elif (userInput==4):
                ViewProduct()
        elif (userInput==5):
                PurchaseProduct()
        elif (userInput==6):
            ViewPurchase()
        elif (userInput==7):
            ViewStock()
        elif (userInput==8):
            SaleProduct()
        elif (userInput==9):
            ViewSales()
        else:
            print("Enter correct choice. . . ")




MenuSet()

def runAgain():
        runAgn = input("\nwant To Run Again Y/n: ")
        while(runAgn.lower() == 'y'):
            if(platform.system() == "Windows"):
                print(os.system('cls'))
            else:
                print(os.system('clear'))
            MenuSet()
            runAgn = input("\nwant To Run Again Y/n: ")
runAgain()
```

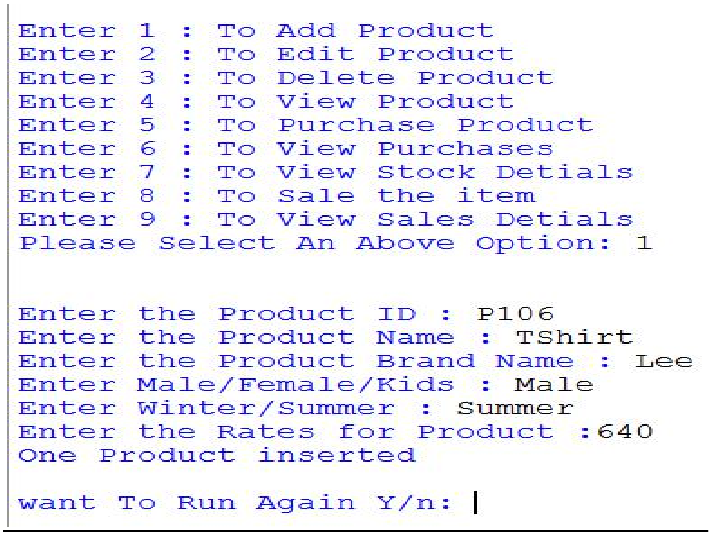
Output of code… :

Main menu:



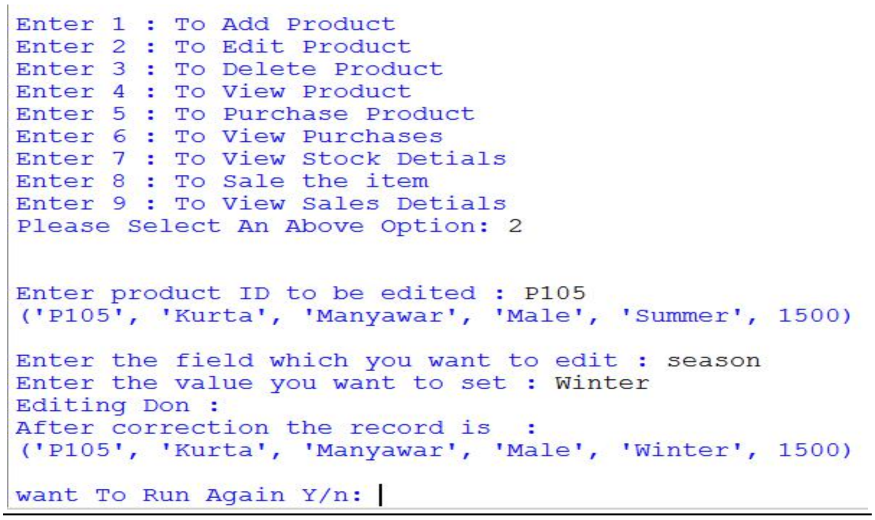
```
********************************************************************************
* * * * * * * Welcome to the Project of Fashion Store * * * * * * *
* * * * Developed by: Neha Tyagi, PGT CS, KV no. 5 Jaipur : * * * *
********************************************************************************

Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: |
```

Add product:



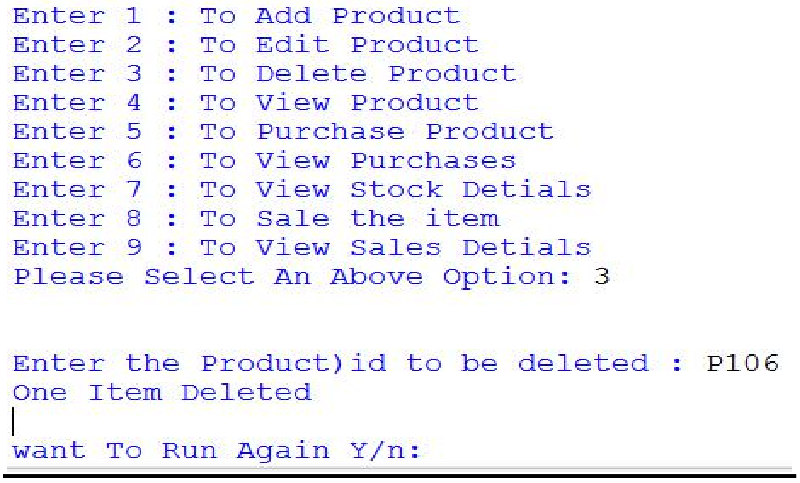
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 1


Enter the Product ID : P106
Enter the Product Name : TShirt
Enter the Product Brand Name : Lee
Enter Male/Female/Kids : Male
Enter Winter/Summer : Summer
Enter the Rates for Product :640
One Product inserted

want To Run Again Y/n: |
```

Edit product:



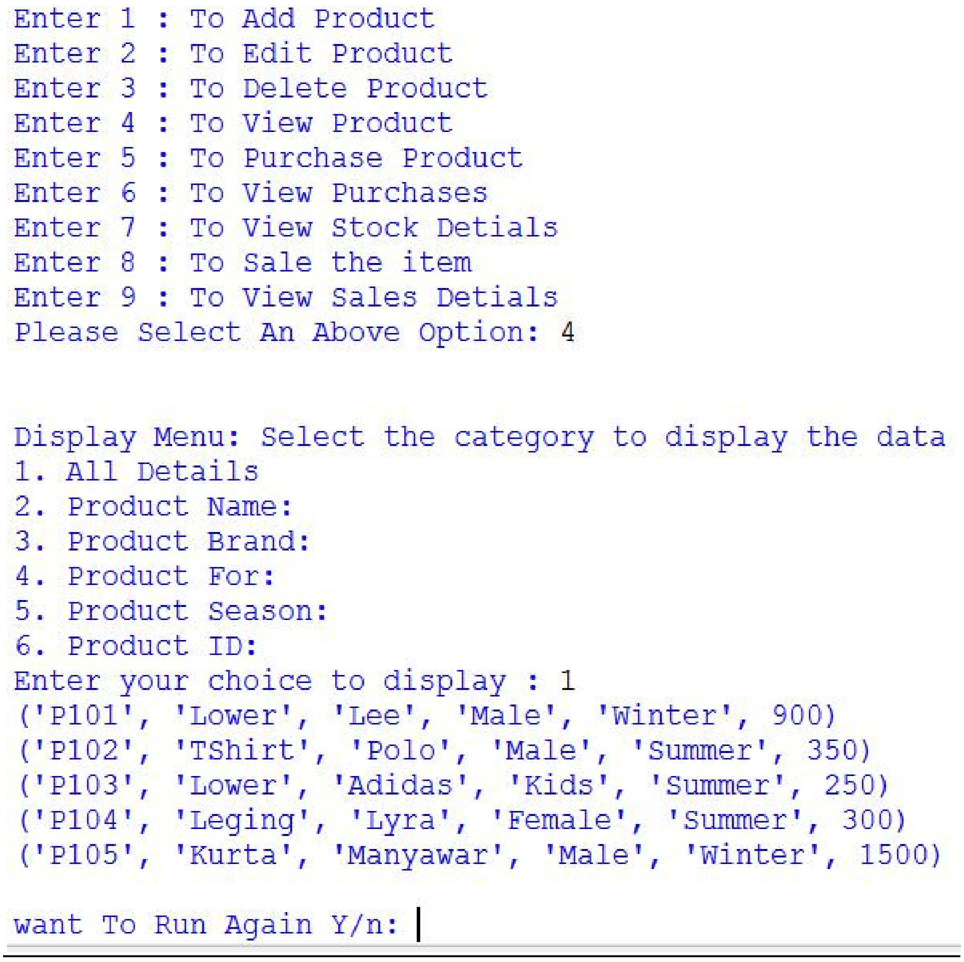
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 2


Enter product ID to be edited : P105
('P105', 'Kurta', 'Manyawar', 'Male', 'Summer', 1500)

Enter the field which you want to edit : season
Enter the value you want to set : Winter
Editing Don :
After correction the record is  :
('P105', 'Kurta', 'Manyawar', 'Male', 'Winter', 1500)

want To Run Again Y/n: 
```

Delete product:



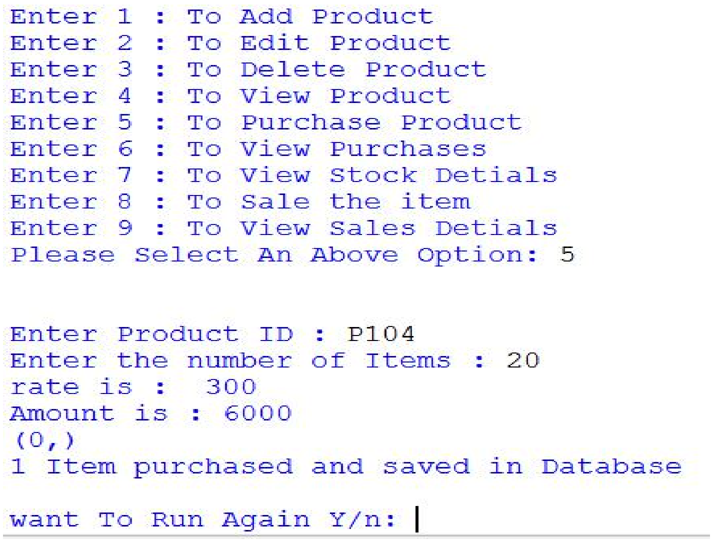
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 3


Enter the Product)id to be deleted : P106
One Item Deleted

want To Run Again Y/n:
```

View product:



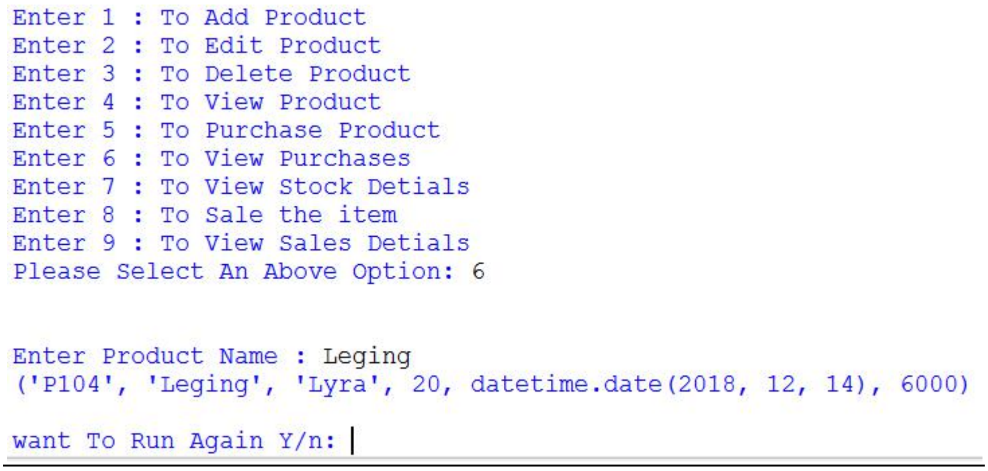
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 4


Display Menu: Select the category to display the data
1. All Details
2. Product Name:
3. Product Brand:
4. Product For:
5. Product Season:
6. Product ID:
Enter your choice to display : 1
('P101', 'Lower', 'Lee', 'Male', 'Winter', 900)
('P102', 'TShirt', 'Polo', 'Male', 'Summer', 350)
('P103', 'Lower', 'Adidas', 'Kids', 'Summer', 250)
('P104', 'Leging', 'Lyra', 'Female', 'Summer', 300)
('P105', 'Kurta', 'Manyawar', 'Male', 'Winter', 1500)

want To Run Again Y/n:
```

__Purchase product:__



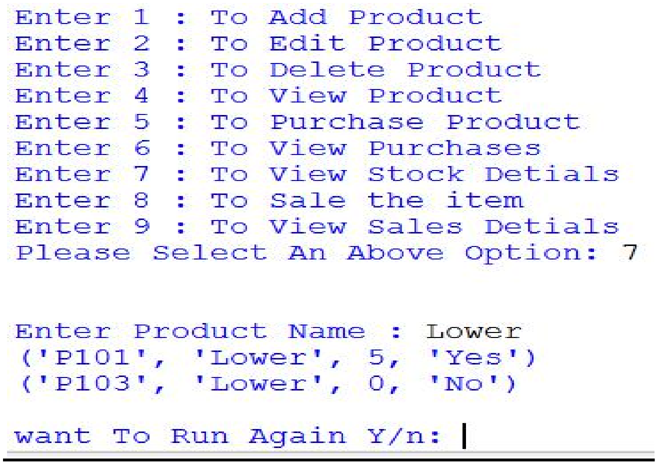
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 5


Enter Product ID : P104
Enter the number of Items : 20
rate is :  300
Amount is : 6000
(0,)
1 Item purchased and saved in Database

want To Run Again Y/n:
```

View purchase:



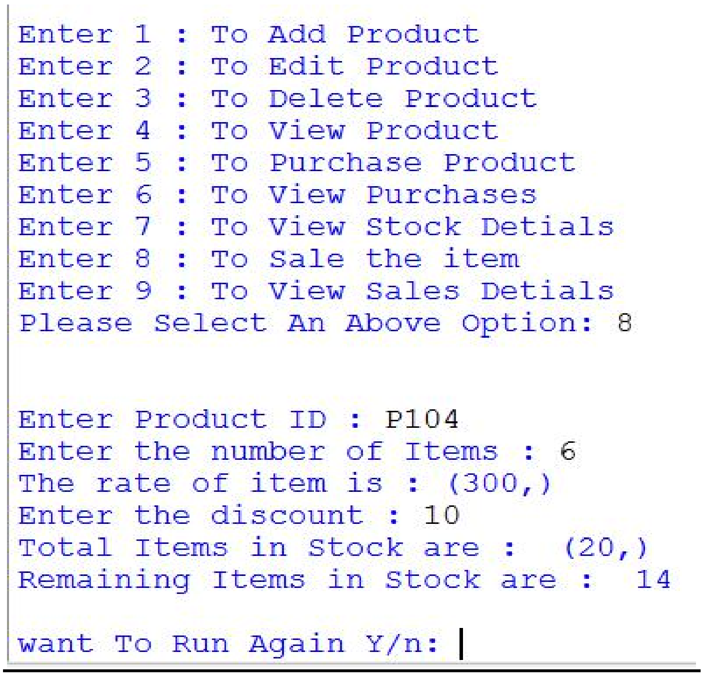
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 6


Enter Product Name : Leging
('P104', 'Leging', 'Lyra', 20, datetime.date(2018, 12, 14), 6000)

want To Run Again Y/n: |
```

View stock details:



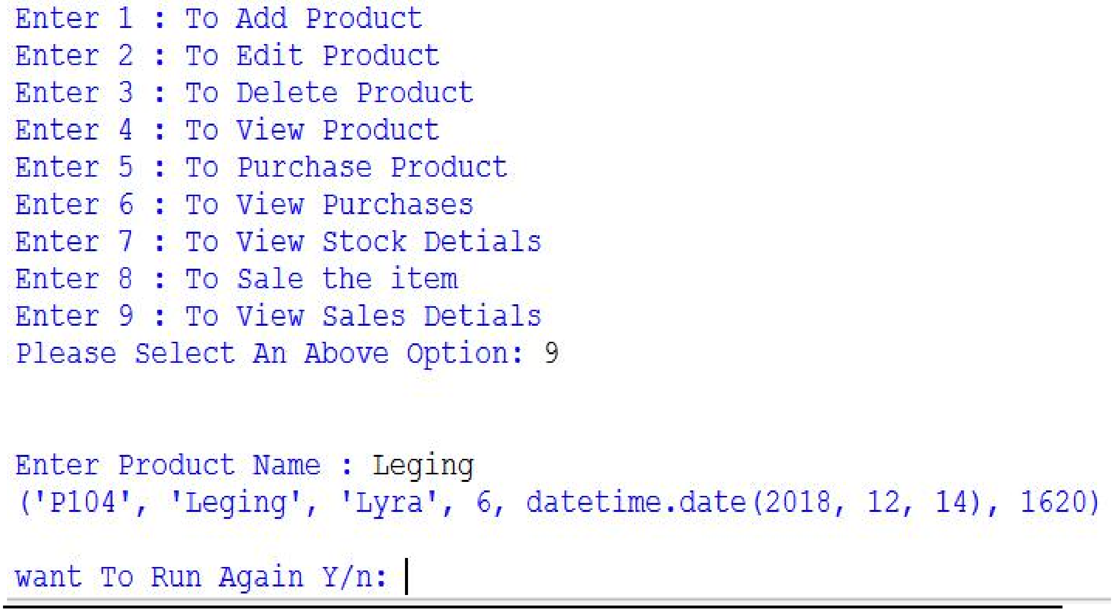
```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 7


Enter Product Name : Lower
('P101', 'Lower', 5, 'Yes')
('P103', 'Lower', 0, 'No')

want To Run Again Y/n: |
```

Sale item:



```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 8


Enter Product ID : P104
Enter the number of Items : 6
The rate of item is : (300,)
Enter the discount : 10
Total Items in Stock are :  (20,)
Remaining Items in Stock are :  14

want To Run Again Y/n:
```

View sales details:



```
Enter 1 : To Add Product
Enter 2 : To Edit Product
Enter 3 : To Delete Product
Enter 4 : To View Product
Enter 5 : To Purchase Product
Enter 6 : To View Purchases
Enter 7 : To View Stock Detials
Enter 8 : To Sale the item
Enter 9 : To View Sales Detials
Please Select An Above Option: 9


Enter Product Name : Leging
('P104', 'Leging', 'Lyra', 6, datetime.date(2018, 12, 14), 1620)

want To Run Again Y/n:
```

Future enhancements

1. The process of gathering information, diagnosing the problems, then interpreting facts is known as System analysis. It includes recommending system improvements needed, based on the same data.
2. The system is observed as a whole; the inputs need to be identified firstly before tuning them and then the system is subjected to study as a whole to identify the problem areas.
3. Although tunings any system as a whole is a complex procedure, but tuning individual statements is not the best as something that is correct for one input may hurt another inputs performance.
4. The solutions are given as a proposal. The suggestion is revised on user request and optimal changes are made. This loop terminates as soon as the user is gratified with the proposal.
5. So on the whole, system analysis is done to improve the system performance by monitoring it and obtaining the best throughput possible from it.
6. It would provide more and more details of products which the consumer or the user wants.

Bibliography

SITES REFFERED FOR SOME BASIC INFORMATON:

1. *https://en.wikipedia.org*
2. *https://www.geeksforgeeks.org/*

BOOKS REFFERED:

1. ***Introduction to Python (class:xi),Sumita Arora***
2. ***Introduction to python(class:xii),Sumita Arora and Preeti Arora***

****. Also under the guidance of our subject teacher***

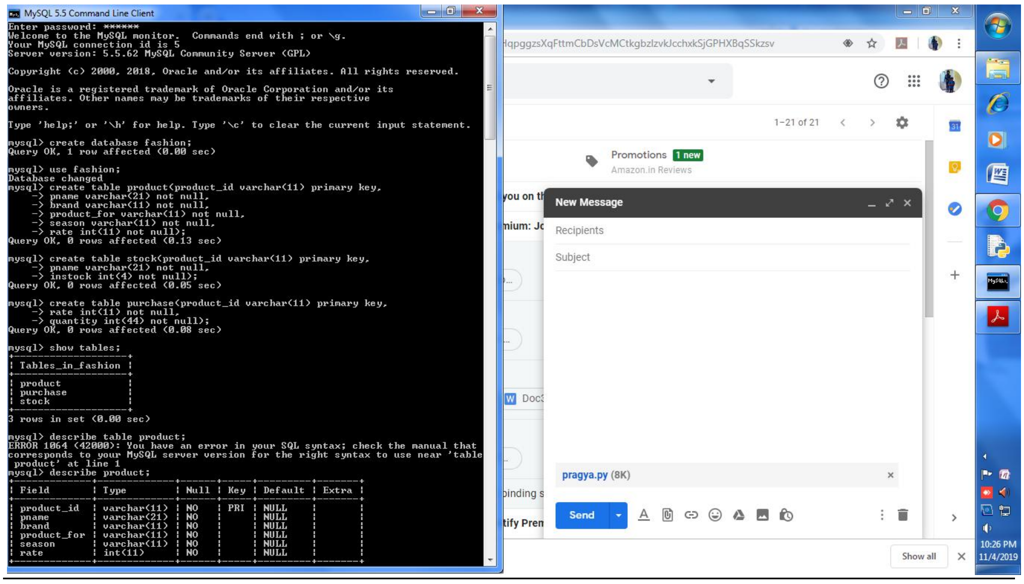
Project on : Fashion Stores

Made by:

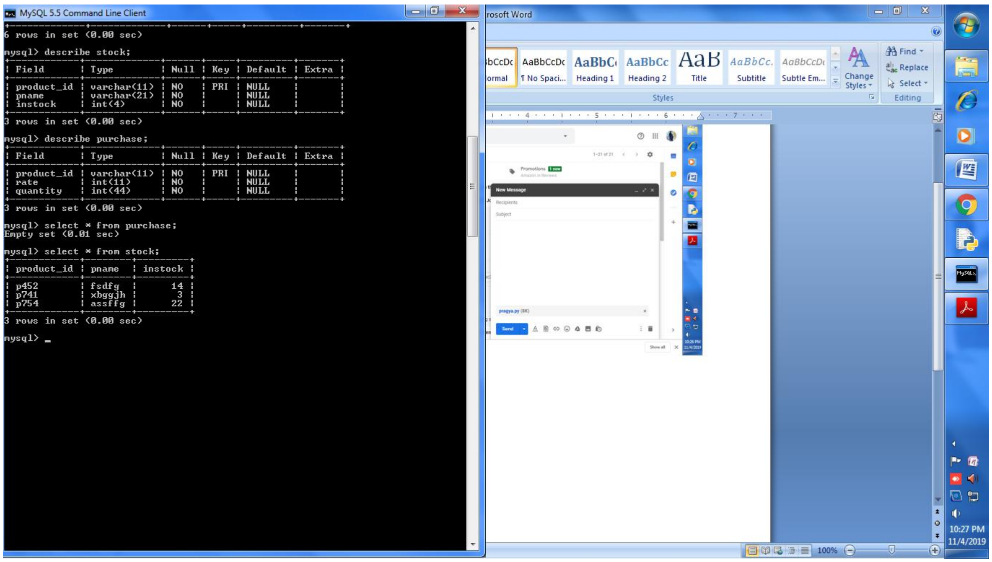
Hanisha Jain

Class:xii

Brio Poddar International School



```
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database fashion;
Query OK, 1 row affected (0.00 sec)

mysql> use fashion;
Database changed
mysql> create table product(product_id varchar(11) primary key,
    -> pname varchar(21) not null,
    -> brand varchar(11) not null,
    -> product_for varchar(11) not null,
    -> season varchar(11) not null,
    -> rate int(11) not null);
Query OK, 0 rows affected (0.13 sec)

mysql> create table stock(product_id varchar(11) primary key,
    -> pname varchar(21) not null,
    -> instock int(4) not null);
Query OK, 0 rows affected (0.05 sec)

mysql> create table purchase(product_id varchar(11) primary key,
    -> rate int(11) not null,
    -> quantity int(44) not null);
Query OK, 0 rows affected (0.08 sec)

mysql> show tables;
+-------------------+
| Tables_in_fashion |
+-------------------+
| product           |
| purchase          |
| stock             |
+-------------------+
3 rows in set (0.00 sec)

mysql> describe table product;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'table
 product' at line 1
mysql> describe product;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| product_id  | varchar(11) | NO   | PRI | NULL    |       |
| pname       | varchar(21) | NO   |     | NULL    |       |
| brand       | varchar(11) | NO   |     | NULL    |       |
| product_for | varchar(11) | NO   |     | NULL    |       |
| season      | varchar(11) | NO   |     | NULL    |       |
| rate        | int(11)     | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```



```
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe stock;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| product_id | varchar(11) | NO   | PRI | NULL    |       |
| pname      | varchar(21) | NO   |     | NULL    |       |
| instock    | int(4)      | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe purchase;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| product_id | varchar(11) | NO   | PRI | NULL    |       |
| rate       | int(11)     | NO   |     | NULL    |       |
| quantity   | int(44)     | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> select * from purchase;
Empty set (0.01 sec)

mysql> select * from stock;
+------------+--------+---------+
| product_id | pname  | instock |
+------------+--------+---------+
| p452       | fsdfg  |      14 |
| p741       | xbggjh |       3 |
| p754       | assffg |      22 |
+------------+--------+---------+
3 rows in set (0.00 sec)

mysql> _
```