

S9 - SQL Queries (in MySQL)

Problem Statement:

SQL Queries

Create following tables with suitable constraints. Insert data and solve the following queries:

CUSTOMERS(*CNo*, *Cname*, *Ccity*, *CMobile*)

ITEMS(*INo*, *Iname*, *Itype*, *Iprice*, *Icount*)

PURCHASE(*PNo*, *Pdate*, *Pquantity*, *Cno*, *INo*)

1. List all stationary items with price between 400/- to 1000/-
2. Change the mobile number of customer "Gopal"
3. Display the item with maximum price
4. Display all purchases sorted from the most recent to the oldest
5. Count the number of customers in every city
6. Display all purchased quantity of Customer Maya
7. Create view which shows *Iname*, *Price* and *Count* of all stationary items in descending order of price.

Creating the database

```
CREATE DATABASE Store3;  
USE Store3;
```

Creating tables:

```
CREATE TABLE Customers (  
    CNo INT,  
    Cname VARCHAR(255),  
    Ccity VARCHAR(255),  
    Cmobile BIGINT,  
    PRIMARY KEY (CNo)  
);
```

```
CREATE TABLE Items (  
    INo INT,  
    Iname VARCHAR(255),  
    Itype VARCHAR(255),  
    Iprice INT,  
    Icount INT,  
    PRIMARY KEY (Icount)  
);
```

```
CREATE TABLE Purchase (  
    PNo INT,  
    Pdate DATE,
```

```
Pquantity INT,  
Cno INT,  
INo INT,  
PRIMARY KEY (PNo),  
FOREIGN KEY (Cno) REFERENCES Customers (CNo)  
);
```

[!WARNING]

Notice inconsistent naming for columns? We're just doing it by the books. Blame the one who made these [problem statements](#).

Inserting data

```
INSERT INTO Customers VALUES  
(1, 'Kalas', 'Pune', 9857265240),  
(2, 'Himanshu', 'Chennai', 9857265241),  
(3, 'Gopal', 'Mumbai', 9857265245),  
(4, 'Maya', 'Mumbai', 9857265243),  
(5, 'Mehul', 'Pune', 9857265244);  
  
INSERT INTO Items VALUES  
(101, 'Stamp', 'Collectables', 850, 10),  
(102, 'Pen', 'Instruments', 549, 50),  
(103, 'Sticky notes', 'Writing', 150, 200),  
(104, 'Geometry box', 'Instruments', 1350, 70),  
(105, 'Pencil', 'Instruments', 670, 45);  
  
INSERT INTO Purchase VALUES  
(201, '2024-11-05', 4, 1, 101),  
(202, '2024-11-07', 3, 2, 102),  
(203, '2024-11-08', 20, 3, 103),  
(204, '2024-10-29', 1, 4, 104),  
(205, '2024-11-10', 7, 5, 105);
```

Queries

1. List all stationary items with price between 400/- to 1000/-

```
SELECT Iname FROM Items WHERE Iprice BETWEEN 400 AND 1000;
```

2. Change the mobile number of customer "Gopal"

```
UPDATE Customers SET CMobile = 9857265242 WHERE Cname = "Gopal";  
SELECT * FROM Customers WHERE Cname = "Gopal";
```

3. Display the item with maximum price

```
SELECT Iname FROM Items WHERE Iprice = (SELECT MAX(Iprice) FROM Items);
```

4. Display all purchases sorted from the most recent to the oldest

```
SELECT * FROM Purchase ORDER BY Pdate DESC;
```

5. Count the number of customers in every city

```
SELECT COUNT(*), Cci ty FROM Customers GROUP BY Cci ty;
```

6. Display all purchased quantity of Customer Maya

```
SELECT Iname, Pquantity FROM Purchase INNER JOIN Customers ON Purchase.Cno =  
Customers.CNo INNER JOIN Items ON Purchase.INo = Items.INo WHERE Cname = "Maya";
```

7. Create view which shows Iname, Price and Count of all stationary items in descending order of price.

```
CREATE VIEW itemView AS SELECT Iname, Iprice, Icount FROM Items ORDER BY Iprice DESC;  
SELECT * FROM itemView;
```
