Description:-

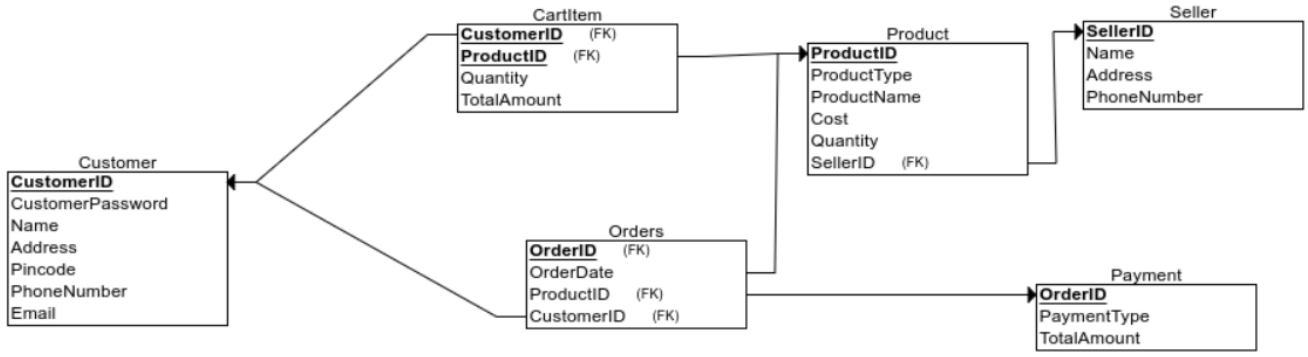
The E-commerce Database Management System provides all the basic functions of a modern e-commerce system.
It consists of various tables:-

- **Customer:-** Contains the customer details
- **Product:-** Contains product details along with its seller
- **Seller:-** Contains the seller details along with contact information.
- **CartItem:-** Contains the products each customer has in their cart.
- **Payment:-** Contains the order id along with the Type of payment made and Total Amount of the order
- **Orders:-** Stores the customer, product and order id to uniquely identify each order.

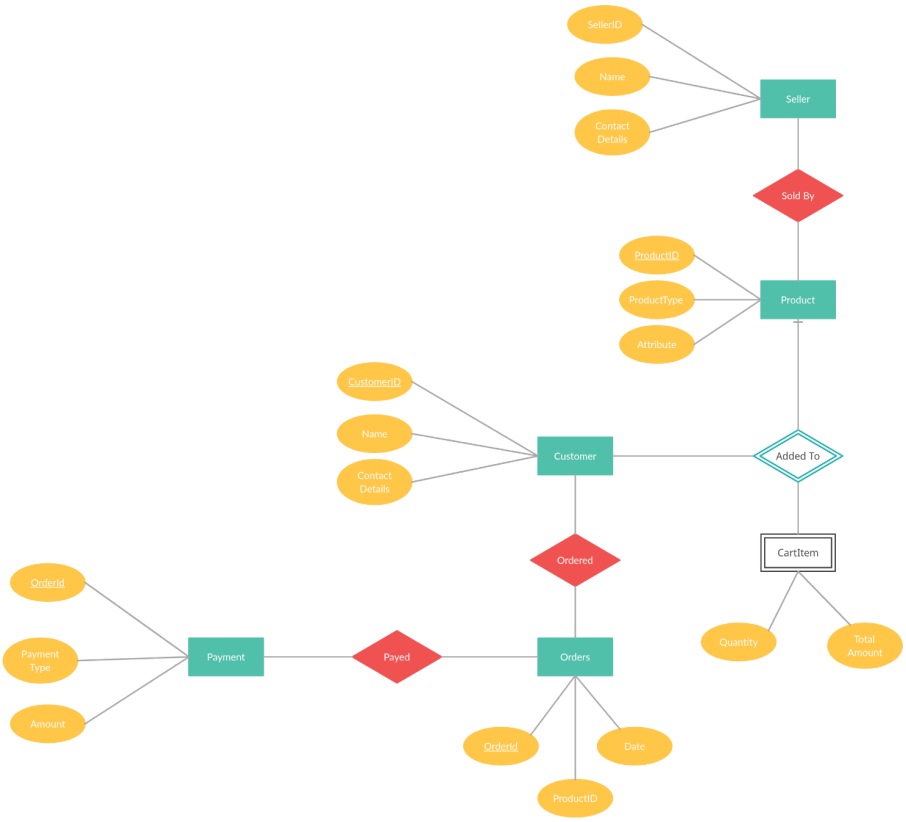
Major Functionality Provided:-

- Check the details of all customers, products, orders, payments, sellers and cart items of the customers
- Allows a customer to add and remove items from cart
- Allows customer to choose mode of payment before an order
- Provide discount to customer depending on order amount
- Provide customer an option to get refund of an order

Relational Database Schema



ER Diagram:-



Tables:-

Customer:-

```
mysql> SELECT * FROM Customer;
+------------+------------------+--------+-----------------------+---------+-------------+-------------------+
| CustomerID | CustomerPassword | Name   | Address               | Pincode | PhoneNumber | Email             |
+------------+------------------+--------+-----------------------+---------+-------------+-------------------+
| 10001      | abvpwd123        | Rajesh | East Phase New Delhi  |  661023 | 8456712345  | rajesh@gmail.com  |
| 10002      | axxcvpd3         | Ajit   | Juhu Mumbai           |  681023 | 8456652345  | ajit@gmail.com    |
| 10003      | bbvxwd123        | Rahul  | Lions Road Bangalore  |  661323 | 8126712345  | rahul@gmail.com   |
| 10004      | abbcvp245        | Arjun  | New Delhi             |  661023 | 8456712345  | arjun@gmail.com   |
| 10005      | abvpwd123        | Raheem | Old City Lucknow      |  781023 | 8456719845  | raheem@gmail.com  |
| 10006      | abvdgd1675       | Charu  | Kings Road Chandigarh |  661723 | 8106712345  | charu@gmail.com   |
| 10007      | lskvpds79        | Sonam  | New Delhi             |  661023 | 8456719785  | sonam@gmail.com   |
| 10008      | bbvpwd320        | Priya  | Rajeshwari Kolkata    |  721023 | 8896714345  | priya@gmail.com   |
| 10009      | ab1234psh        | Anil   | Colaba Mumbai         |  661093 | 8456719845  | anil@gmail.com    |
| 10010      | xxv123pqr        | Riya   | Rajouri NewDelhi      |  760823 | 8419812345  | riya@gmail.com    |
+------------+------------------+--------+-----------------------+---------+-------------+-------------------+
10 rows in set (0.00 sec)
```

Product:-

```
mysql> SELECT * FROM Product;
+-----------+-------------+-------------------+--------+----------+----------+
| ProductID | ProductType | ProductName       | Cost   | Quantity | SellerID |
+-----------+-------------+-------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter      |    600 |       10 | sid101   |
| pid1002   | Fashion     | Makeup            |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X          | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream         |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad |    200 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia       |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator   |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter       |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke              |     40 |       50 | sid101   |
| pid1010   | Eatables    | Bread             |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri           |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle            |    120 |       18 | sid101   |
+-----------+-------------+-------------------+--------+----------+----------+
12 rows in set (0.00 sec)
```

Seller:-

```
mysql> SELECT * FROM Seller;
+----------+------------+------------------+-------------+
| SellerID | Name       | Address          | PhoneNumber |
+----------+------------+------------------+-------------+
| sid100   | Aman       | At post Delhi    | 9923657432  |
| sid101   | Zaad       | At post Mumbai   | 9923657433  |
| sid102   | Nazir      | At post Bhiwandi | 9923657434  |
| sid103   | Aadil      | At post Pune     | 9923657435  |
| sid104   | Anil       | At post Bhopal   | 9923657436  |
| sid105   | Yogesh     | At post Mumbai   | 9923657437  |
| sid106   | Kiran      | At post Delhi    | 9923657438  |
| sid107   | Himanshu   | At post Bhiwandi | 9923657439  |
| sid108   | Mohan      | At post Chennai  | 9923657440  |
| sid109   | Bhanupriya | At post Banglore | 9923657430  |
+----------+------------+------------------+-------------+
10 rows in set (0.00 sec)
```

CartItem:

```
mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
10 rows in set (0.00 sec)
```

Payment:

```
mysql> SELECT * FROM Payment;
+---------+---------------+-------------+
| OrderID | PaymentType   | TotalAmount |
+---------+---------------+-------------+
| 1000001 | Net Banking   |         800 |
| 1000002 | Credit/Debit  |      100000 |
| 1000003 | Net Banking   |       75000 |
| 1000004 | UPI           |         600 |
| 1000005 | COD           |       45000 |
| 1000006 | COD           |         600 |
| 1000007 | UPI           |         200 |
| 1000008 | Credit/Debit  |          40 |
| 1000009 | COD           |        5000 |
| 1000010 | UPI           |         120 |
| 1000011 | UPI           |          10 |
| 1000012 | COD           |          10 |
| 1000013 | Net Banking   |         200 |
| 1000014 | COD           |          30 |
| 1000015 | UPI           |          30 |
+---------+---------------+-------------+
15 rows in set (0.00 sec)
```

Orders:

```
mysql> SELECT * FROM Orders;
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000002 | 10001      | pid1003   | 2019-10-23 |
| 1000003 | 10005      | pid1006   | 2018-10-15 |
| 1000004 | 10003      | pid1001   | 2019-10-05 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000006 | 10007      | pid1001   | 2019-10-08 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000008 | 10003      | pid1009   | 2019-10-02 |
| 1000009 | 10001      | pid1007   | 2019-10-14 |
| 1000010 | 10009      | pid1012   | 2018-10-02 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000012 | 10007      | pid1011   | 2018-10-02 |
| 1000013 | 10005      | pid1005   | 2018-10-17 |
| 1000014 | 10001      | pid1004   | 2018-10-18 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
+---------+------------+-----------+------------+
15 rows in set (0.00 sec)
```

Simple queries

1. **To get the details of all the customers of a particular city ‘Mumbai'**

```
mysql> SELECT CustomerID,Name,PhoneNumber,Email
    -> FROM Customer
    -> WHERE Address LIKE '%Mumbai%';
+------------+------+-------------+----------------+
| CustomerID | Name | PhoneNumber | Email          |
+------------+------+-------------+----------------+
| 10002      | Ajit | 8456652345  | ajit@gmail.com |
| 10009      | Anil | 8456719845  | anil@gmail.com |
+------------+------+-------------+----------------+
2 rows in set (0.00 sec)
```

2. **To decrease the price of all products of seller with SellerID ‘sid101’ by 10%**

```
mysql> SELECT * FROM Product;
+-----------+-------------+--------------------+--------+----------+----------+
| ProductID | ProductType | ProductName        | Cost   | Quantity | SellerID |
+-----------+-------------+--------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter       |    600 |       10 | sid101   |
| pid1002   | Fashion     | Makeup             |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X           | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream          |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad  |    200 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia        |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator    |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter        |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke               |     40 |       50 | sid101   |
| pid1010   | Eatables    | Bread              |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri            |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle             |    120 |       18 | sid101   |
+-----------+-------------+--------------------+--------+----------+----------+
12 rows in set (0.00 sec)

mysql> UPDATE Product SET cost=0.9*cost
    -> WHERE SellerID='sid101';
Query OK, 4 rows affected (0.20 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql>
mysql> SELECT * FROM Product;
+-----------+-------------+--------------------+--------+----------+----------+
| ProductID | ProductType | ProductName        | Cost   | Quantity | SellerID |
+-----------+-------------+--------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter       |    540 |       10 | sid101   |
| pid1002   | Fashion     | Makeup             |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X           | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream          |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad  |    180 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia        |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator    |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter        |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke               |     36 |       50 | sid101   |
| pid1010   | Eatables    | Bread              |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri            |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle             |    108 |       18 | sid101   |
+-----------+-------------+--------------------+--------+----------+----------+
12 rows in set (0.00 sec)
```

3. **To get the different types of payment accepted**

```
mysql> SELECT DISTINCT PaymentType AS DifferentTypesOfPaymentsAccepted
    -> FROM Payment;
+----------------------------------+
| DifferentTypesOfPaymentsAccepted |
+----------------------------------+
| Net Banking                      |
| Credit/Debit                     |
| UPI                              |
| COD                              |
+----------------------------------+
4 rows in set (0.00 sec)
```

4. **To get the costliest product sold by each seller**

```
mysql> SELECT P.ProductID,P.ProductName,S.SellerID,S.Name,P.Cost
    -> FROM Product P,Seller S
    -> WHERE P.SellerID=S.SellerID
    -> GROUP BY P.SellerID
    -> HAVING Cost=MAX(Cost);
+-----------+-----------------+----------+------------+--------+
| ProductID | ProductName     | SellerID | Name       | Cost   |
+-----------+-----------------+----------+------------+--------+
| pid1001   | Harry Potter    | sid101   | Zaad       |    540 |
| pid1008   | TVS Jupiter     | sid102   | Nazir      |  45000 |
| pid1010   | Bread           | sid103   | Aadil      |     20 |
| pid1004   | Ice Cream       | sid104   | Anil       |     30 |
| pid1002   | Makeup          | sid105   | Yogesh     |    800 |
| pid1006   | Sony Bravia     | sid106   | Kiran      |  75000 |
| pid1007   | Adidas Predator | sid108   | Mohan      |   5000 |
| pid1003   | iPhone X        | sid109   | Bhanupriya | 100000 |
+-----------+-----------------+----------+------------+--------+
8 rows in set (0.07 sec)
```

5. **To Find all the distinct locations where the various sellers are located**

```
mysql> SELECT DISTINCT Address
    -> FROM Seller;
+-------------------+
| Address           |
+-------------------+
| At post Delhi     |
| At post Mumbai    |
| At post Bhiwandi  |
| At post Pune      |
| At post Bhopal    |
| At post Chennai   |
| At post Banglore  |
+-------------------+
7 rows in set (0.00 sec)
```

6. **To get the number of cart items of customers with at least 1 cart item in desc order**

```
mysql> SELECT C.CustomerID, C.Name, COUNT(*) AS NumberOfItems
    -> FROM Customer C,CartItem CI
    -> WHERE C.CustomerID=CI.CustomerID
    -> GROUP BY C.CustomerID
    -> ORDER BY NumberOfItems desc;
+------------+--------+---------------+
| CustomerID | Name   | NumberOfItems |
+------------+--------+---------------+
| 10002      | Ajit   |             2 |
| 10004      | Arjun  |             2 |
| 10008      | Priya  |             2 |
| 10001      | Rajesh |             1 |
| 10006      | Charu  |             1 |
| 10007      | Sonam  |             1 |
| 10009      | Anil   |             1 |
+------------+--------+---------------+
7 rows in set (0.00 sec)
```

7. **To get the details of all orders placed after 2019**

```
mysql> SELECT  * FROM Orders
    -> WHERE OrderDate > '2019-12-31';
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
+---------+------------+-----------+------------+
5 rows in set (0.00 sec)
```

8. **To get the count of different types of products available along with their count**

```
mysql> SELECT ProductType, COUNT(*) AS NumberOfProducts
    -> FROM Product
    -> GROUP BY ProductType;
+-------------+-------------------+
| ProductType | NumberOfProducts  |
+-------------+-------------------+
| Book        |                 3 |
| Fashion     |                 1 |
| Electronics |                 2 |
| Eatables    |                 4 |
| Footwear    |                 1 |
| Vehicle     |                 1 |
+-------------+-------------------+
6 rows in set (0.00 sec)
```

9. **To get the average cost of every type of product**

```
mysql> SELECT ProductType,AVG(COST) AS AverageCost
    -> FROM Product
    -> GROUP BY ProductType;
+-------------+-------------+
| ProductType | AverageCost |
+-------------+-------------+
| Book        |    276.0000 |
| Fashion     |    800.0000 |
| Electronics |  87500.0000 |
| Eatables    |     24.0000 |
| Footwear    |   5000.0000 |
| Vehicle     |  45000.0000 |
+-------------+-------------+
6 rows in set (0.00 sec)
```

10. **To get the details of orders of a particular customer with name Rajesh**

```
mysql> SELECT C.Name,O.OrderID,O.ProductID,O.OrderDate
    -> FROM Customer as C, Orders as O
    -> WHERE C.Name='Rajesh' and C.CustomerID=O.CustomerID;
+--------+---------+-----------+------------+
| Name   | OrderID | ProductID | OrderDate  |
+--------+---------+-----------+------------+
| Rajesh | 1000002 | pid1003   | 2019-10-23 |
| Rajesh | 1000009 | pid1007   | 2019-10-14 |
| Rajesh | 1000014 | pid1004   | 2018-10-18 |
+--------+---------+-----------+------------+
3 rows in set (0.00 sec)
```

Complex Queries

1. **To find customers who have placed orders as well as have items in their cart**

```
mysql> SELECT * FROM Customer as C
    -> WHERE C.CustomerID IN(SELECT O.CustomerID
    ->                       From Orders AS O, CartItem as CI
    ->                       WHERE O.CustomerID=C.CustomerID AND C.CustomerID=CI.CustomerID);
+------------+------------------+--------+----------------------+---------+-------------+------------------+
| CustomerID | CustomerPassword | Name   | Address              | Pincode | PhoneNumber | Email            |
+------------+------------------+--------+----------------------+---------+-------------+------------------+
| 10001      | abvpwd123        | Rajesh | East Phase New Delhi |  661023 | 8456712345  | rajesh@gmail.com |
| 10004      | abbcvp245        | Arjun  | New Delhi            |  661023 | 8456712345  | arjun@gmail.com  |
| 10006      | abvdgd1675       | Charu  | Kings Road Chandigarh |  661723 | 8106712345  | charu@gmail.com  |
| 10007      | lskvpds79        | Sonam  | New Delhi            |  661023 | 8456719785  | sonam@gmail.com  |
| 10009      | ab1234psh        | Anil   | Colaba Mumbai        |  661093 | 8456719845  | anil@gmail.com   |
+------------+------------------+--------+----------------------+---------+-------------+------------------+
5 rows in set (0.03 sec)
```

2. **To find number of different types of payments made by customers from Mumbai and Delhi**

```
mysql> SELECT P.PaymentType,COUNT(*) AS NumberOfPayments
    -> FROM Payment P, Orders O
    -> WHERE P.OrderID=O.OrderID AND
    -> O.CustomerID IN(SELECT CustomerID
    ->                 FROM Customer
    ->                 WHERE CustomerID=O.CustomerID AND
    ->                 Address LIKE '%Mumbai%' OR Address LIKE '%Delhi%')
    -> GROUP BY P.PaymentType;
+--------------+-------------------+
| PaymentType  | NumberOfPayments  |
+--------------+-------------------+
| Credit/Debit |                 1 |
| COD          |                 4 |
| UPI          |                 2 |
+--------------+-------------------+
3 rows in set (0.00 sec)
```

3. **To find the highest order of all customers**

```
mysql> SELECT C.CustomerID, C.Name, O.OrderID, P.TotalAmount
    -> FROM Customer AS C, Orders AS O, Payment AS P
    -> WHERE O.CustomerID=C.CustomerID AND P.OrderID=O.OrderID
    -> AND P.TotalAmount=(SELECT MAX(P1.TotalAmount)
    ->                    FROM Payment AS P1, Orders AS O1
    ->                    WHERE P1.OrderID=O1.OrderID AND O1.CustomerID=C.CustomerID);
+------------+--------+---------+-------------+
| CustomerID | Name   | OrderID | TotalAmount |
+------------+--------+---------+-------------+
| 10001      | Rajesh | 1000002 |      100000 |
| 10003      | Rahul  | 1000001 |         800 |
| 10004      | Arjun  | 1000011 |          10 |
| 10005      | Raheem | 1000003 |       75000 |
| 10006      | Charu  | 1000005 |       45000 |
| 10007      | Sonam  | 1000006 |         600 |
| 10009      | Anil   | 1000010 |         120 |
+------------+--------+---------+-------------+
7 rows in set (0.00 sec)
```

4. **To find the customer with maximum orders placed**

```
mysql> SELECT C.CustomerID, C.Name,COUNT(*) AS NoOfOrders FROM Customer C
    -> INNER JOIN Orders O
    -> ON O.CustomerID=C.CustomerID
    -> GROUP BY C.CustomerID
    -> HAVING NoOfOrders>=ALL(SELECT COUNT(*) FROM
    ->                        Orders
    ->                        GROUP BY CustomerID);
+------------+-------+------------+
| CustomerID | Name  | NoOfOrders |
+------------+-------+------------+
| 10003      | Rahul |          4 |
+------------+-------+------------+
1 row in set (0.00 sec)
```

5. **All customers that have bought product from sellers that sell more than 1 product**

```
mysql> SELECT DISTINCT C.CustomerID, C.Name FROM Customer C,Orders O, Product P,Seller S
    -> WHERE O.CustomerID=C.CustomerID AND O.ProductID=P.ProductID AND P.SellerID=S.SellerID AND
    -> P.SellerID IN(SELECT SellerID
    ->               FROM Product
    ->               GROUP BY SellerID
    ->               HAVING COUNT(*)>1);
+------------+--------+
| CustomerID | Name   |
+------------+--------+
| 10003      | Rahul  |
| 10007      | Sonam  |
| 10005      | Raheem |
| 10009      | Anil   |
| 10004      | Arjun  |
+------------+--------+
5 rows in set (0.00 sec)
```

6. **To find orders of the customer who has placed highest number of orders**

```
mysql> SELECT O.CustomerID, O.OrderID, O.ProductID, O.OrderDate FROM Orders AS O
    -> WHERE O.CustomerID = (SELECT CustomerID
    ->                       FROM Orders
    ->                       GROUP BY CustomerID
    ->                       ORDER BY COUNT(CustomerID) DESC LIMIT 1);
+------------+---------+-----------+------------+
| CustomerID | OrderID | ProductID | OrderDate  |
+------------+---------+-----------+------------+
| 10003      | 1000001 | pid1002   | 2020-10-12 |
| 10003      | 1000004 | pid1001   | 2019-10-05 |
| 10003      | 1000008 | pid1009   | 2019-10-02 |
| 10003      | 1000015 | pid1004   | 2020-10-02 |
+------------+---------+-----------+------------+
4 rows in set (0.00 sec)
```

7. **To find Customers who have not purchased anything**

```
mysql> SELECT * FROM Customer
    -> WHERE CustomerID!=ALL(SELECT CustomerID
    ->                       FROM Orders);
+------------+------------------+-------+--------------------+---------+-------------+-----------------+
| CustomerID | CustomerPassword | Name  | Address            | Pincode | PhoneNumber | Email           |
+------------+------------------+-------+--------------------+---------+-------------+-----------------+
| 10002      | axxcvpd3         | Ajit  | Juhu Mumbai        |  681023 | 8456652345  | ajit@gmail.com  |
| 10008      | bbvpwd320        | Priya | Rajeshwari Kolkata |  721023 | 8896714345  | priya@gmail.com |
| 10010      | xxv123pqr        | Riya  | Rajouri NewDelhi   |  760823 | 8419812345  | riya@gmail.com  |
+------------+------------------+-------+--------------------+---------+-------------+-----------------+
3 rows in set (0.00 sec)
```

Views:-

1. **To show for each order the customer and seller by name**

```
mysql> CREATE VIEW OrderDet AS
    -> SELECT O.OrderID, C.Name AS CustomerName, S.Name AS SellerName
    -> FROM Orders AS O, Customer AS C, Product AS P, Seller AS S
    -> WHERE O.CustomerID=C.CustomerID AND O.ProductID=P.ProductID AND P.SellerID=S.SellerID;
Query OK, 0 rows affected (0.16 sec)

mysql>
mysql> SELECT * FROM OrderDet;
+---------+--------------+------------+
| OrderID | CustomerName | SellerName |
+---------+--------------+------------+
| 1000004 | Rahul        | Zaad       |
| 1000006 | Sonam        | Zaad       |
| 1000007 | Raheem       | Zaad       |
| 1000013 | Raheem       | Zaad       |
| 1000008 | Rahul        | Zaad       |
| 1000010 | Anil         | Zaad       |
| 1000005 | Charu        | Nazir      |
| 1000011 | Arjun        | Aadil      |
| 1000012 | Sonam        | Aadil      |
| 1000014 | Rajesh       | Anil       |
| 1000015 | Rahul        | Anil       |
| 1000001 | Rahul        | Yogesh     |
| 1000003 | Raheem       | Kiran      |
| 1000009 | Rajesh       | Mohan      |
| 1000002 | Rajesh       | Bhanupriya |
+---------+--------------+------------+
15 rows in set (0.00 sec)
```

2. **To show products which value less than x amt in increasing order of price**

```
mysql> CREATE VIEW ProductsWithPriceLessThanRs1000 AS
    -> SELECT * FROM Product
    -> WHERE Cost<1000
    -> ORDER BY Cost;
Query OK, 0 rows affected (0.47 sec)

mysql>
mysql> SELECT * FROM ProductsWithPriceLessThanRs1000;
+-----------+-------------+--------------------+------+----------+----------+
| ProductID | ProductType | ProductName        | Cost | Quantity | SellerID |
+-----------+-------------+--------------------+------+----------+----------+
| pid1011   | Eatables    | Bisleri            |   10 |      100 | sid103   |
| pid1010   | Eatables    | Bread              |   20 |      100 | sid103   |
| pid1004   | Eatables    | Ice Cream          |   30 |       15 | sid104   |
| pid1009   | Eatables    | Coke               |   36 |       50 | sid101   |
| pid1012   | Book        | Tinkle             |  108 |       18 | sid101   |
| pid1005   | Book        | Rich Dad Poor Dad  |  180 |       15 | sid101   |
| pid1001   | Book        | Harry Potter       |  540 |       10 | sid101   |
| pid1002   | Fashion     | Makeup             |  800 |       18 | sid105   |
+-----------+-------------+--------------------+------+----------+----------+
8 rows in set (0.00 sec)
```

3. **To find customer with highest purchase of the day**

```
mysql> CREATE VIEW HighestPurchaseOfDay AS
    -> SELECT C.CustomerID, C.Name, MAX(P.Cost) as PurchaseAmount, O.OrderDate
    -> FROM Customer C, Orders O, Product P
    -> WHERE O.CustomerID=C.CustomerID AND O.ProductID=P.ProductID
    -> GROUP BY O.OrderDate;
Query OK, 0 rows affected (0.15 sec)

mysql> SET sql_mode = '';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM HighestPurchaseOfDay;
+------------+--------+----------------+------------+
| CustomerID | Name   | PurchaseAmount | OrderDate  |
+------------+--------+----------------+------------+
| 10003      | Rahul  |            800 | 2020-10-12 |
| 10001      | Rajesh |         100000 | 2019-10-23 |
| 10005      | Raheem |          75000 | 2018-10-15 |
| 10003      | Rahul  |            540 | 2019-10-05 |
| 10006      | Charu  |          45000 | 2020-10-06 |
| 10007      | Sonam  |            540 | 2019-10-08 |
| 10003      | Rahul  |             36 | 2019-10-02 |
| 10001      | Rajesh |           5000 | 2019-10-14 |
| 10009      | Anil   |            108 | 2018-10-02 |
| 10004      | Arjun  |             10 | 2020-10-13 |
| 10005      | Raheem |            180 | 2018-10-17 |
| 10001      | Rajesh |             30 | 2018-10-18 |
| 10003      | Rahul  |             30 | 2020-10-02 |
+------------+--------+----------------+------------+
13 rows in set (0.00 sec)
```

4. **To find the lowest cost items order by seller**

```
mysql> CREATE VIEW CheapestProductOfSeller AS
    -> SELECT S.SellerID, P.ProductID, P.ProductName, P.Cost
    -> FROM Product AS P, Seller AS S
    -> WHERE P.SellerID=S.SellerID AND P.ProductID = (SELECT ProductID
    ->                                                FROM Product
    ->                                                WHERE SellerID=S.SellerID
    ->                                                ORDER BY COST LIMIT 1);
Query OK, 0 rows affected (0.15 sec)

mysql>
mysql> SELECT * FROM CheapestProductOfSeller;
+----------+-----------+-----------------+--------+
| SellerID | ProductID | ProductName     | Cost   |
+----------+-----------+-----------------+--------+
| sid101   | pid1009   | Coke            |     36 |
| sid102   | pid1008   | TVS Jupiter     |  45000 |
| sid103   | pid1011   | Bisleri         |     10 |
| sid104   | pid1004   | Ice Cream       |     30 |
| sid105   | pid1002   | Makeup          |    800 |
| sid106   | pid1006   | Sony Bravia     |  75000 |
| sid108   | pid1007   | Adidas Predator |   5000 |
| sid109   | pid1003   | iPhone X        | 100000 |
+----------+-----------+-----------------+--------+
8 rows in set (0.01 sec)
```

5. **To find highest payment of a particular type**

```
mysql> CREATE VIEW HighestPayType AS
    -> SELECT PaymentType,Max(TotalAmount) FROM Payment  GROUP BY PaymentType ;
Query OK, 0 rows affected (0.10 sec)

mysql>
mysql> SELECT * FROM HighestPayType;
+--------------+------------------+
| PaymentType  | Max(TotalAmount) |
+--------------+------------------+
| Net Banking  |            75000 |
| Credit/Debit |           100000 |
| UPI          |              600 |
| COD          |            45000 |
+--------------+------------------+
4 rows in set (0.00 sec)
```

Procedures:-

1. **Insert into cartItem**

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE AddToCart(IN ProductID VARCHAR(7), IN CustomerID VARCHAR(7), IN Quantity INT)
    -> BEGIN
    ->     DECLARE TotalAmount int;
a    ->      DECLARE Stock int;
    ->     DECLARE ErrorMessage VARCHAR(255);
    ->     SET ErrorMessage = 'Not Enough Stock Available';
    ->     SELECT P.Quantity INTO Stock FROM Product AS P WHERE P.ProductID=ProductID;
    ->     IF Quantity<=Stock THEN
    ->         SELECT P.Cost*Quantity INTO TotalAmount From Product as P WHERE P.ProductID=ProductID;
    ->         INSERT INTO CartItem VALUES(CustomerID,ProductID,Quantity,TotalAmount);
    ->     ELSE
    ->         SIGNAL SQLSTATE '45000' SET message_text = ErrorMessage;
    ->     END IF;
    -> END//
Query OK, 0 rows affected (0.29 sec)

mysql> DELIMITER ;
mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
10 rows in set (0.00 sec)

mysql> CALL AddToCart('pid1011','10001',20);
Query OK, 1 row affected (0.13 sec)

mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       20 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
11 rows in set (0.01 sec)
```

2. Delete from cartItem

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE RemoveFromCart(IN ProductID VARCHAR(7),IN CustomerID VARCHAR(7),IN NoOfItems INT)
    -> BEGIN
         ->   DECLARE Quan INT;
    ->   DECLARE QLeft INT;
    ->   SELECT C.Quantity INTO Quan
    ->   FROM CartItem C
    ->   WHERE C.CustomerID=CustomerID AND C.ProductID=ProductID;
    ->   IF NoOfItems<Quan THEN SET QLeft=Quan-NoOfItems;
    ->   ELSE SET QLeft=0;
    ->   END IF;
    ->   UPDATE CartItem C
    ->   SET C.Quantity=QLeft
    ->   WHERE C.ProductID=ProductID AND C.CustomerID=CustomerID;
    -> END //
Query OK, 0 rows affected (0.27 sec)

mysql> DELIMITER ;
mysql> SELECT * FROM CartItem;
P+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       20 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
11 rows in set (0.00 sec)

mysql> CALL RemoveFromCart('pid1011','10001',5);
Query OK, 1 row affected (0.13 sec)

mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       15 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
11 rows in set (0.01 sec)
```

3. **Get Sellers products of a particular seller**

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetSellerProducts(IN SellerID VARCHAR(6))
    -> BEGIN
    ->   SELECT * FROM Product AS P
    ->   WHERE P.SellerID=SellerID;
    -> END//
Query OK, 0 rows affected (1.03 sec)

mysql> DELIMITER ;
mysql> CALL GetSellerProducts('sid101');
+-----------+-------------+-------------------+------+----------+----------+
| ProductID | ProductType | ProductName       | Cost | Quantity | SellerID |
+-----------+-------------+-------------------+------+----------+----------+
| pid1001   | Book        | Harry Potter      |  540 |       10 | sid101   |
| pid1005   | Book        | Rich Dad Poor Dad |  180 |       15 | sid101   |
| pid1009   | Eatables    | Coke              |   36 |       50 | sid101   |
| pid1012   | Book        | Tinkle            |  108 |       18 | sid101   |
+-----------+-------------+-------------------+------+----------+----------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

4. **Get cart items of a particular customer**

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetCustomersCartItems(IN CustomerID VARCHAR(6))
    -> BEGIN
    ->   SELECT * FROM CartItem AS CI
    ->   WHERE CI.CustomerID=CustomerID;
    -> END//
Query OK, 0 rows affected (0.26 sec)

mysql> DELIMITER ;
mysql> CALL GetCustomersCartItems('10001');
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       15 |         200 |
+------------+-----------+----------+-------------+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

5. **Customers orders**

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetCustomersOrders(IN CustomerID VARCHAR(6))
    -> BEGIN
    ->   SELECT OrderID,ProductID,OrderDate FROM Orders AS O
    ->   WHERE O.CustomerID=CustomerID;
    -> END//
Query OK, 0 rows affected (0.26 sec)

mysql> DELIMITER ;
mysql> CALL GetCustomersOrders('10003');
+---------+-----------+------------+
| OrderID | ProductID | OrderDate  |
+---------+-----------+------------+
| 1000001 | pid1002   | 2020-10-12 |
| 1000004 | pid1001   | 2019-10-05 |
| 1000008 | pid1009   | 2019-10-02 |
| 1000015 | pid1004   | 2020-10-02 |
+---------+-----------+------------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

Functions:-

1. **Get total price of items in a customers cart**

```
mysql> DELIMITER //
mysql> CREATE FUNCTION GetCartCost(customerID VARCHAR(7))
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->   DECLARE TotalCost INT DEFAULT 0;
    ->   SELECT SUM(TotalAmount) INTO TotalCost
    ->   FROM CartItem AS CI
    ->   WHERE CI.CustomerID=customerID;
    ->   RETURN TotalCost;
    -> END //
Query OK, 0 rows affected (0.85 sec)

mysql> DELIMITER ;
mysql> SELECT GetCartCost('10002') AS TotalAmountInCart;
+-------------------+
| TotalAmountInCart |
+-------------------+
|               490 |
+-------------------+
1 row in set (0.00 sec)

mysql> SELECT GetCartCost('10008') AS TotalAmountInCart;
+-------------------+
| TotalAmountInCart |
+-------------------+
|               220 |
+-------------------+
1 row in set (0.00 sec)
```

2. **Find the ProductLevel of a given product (Less than 10000 Affordable, Less than equal to 70000 Mid-Range, Greater than 70000 Expensive)**

```
mysql> DELIMITER //
mysql> CREATE FUNCTION GetProductLevel(PName VARCHAR(20))
    -> RETURNS VARCHAR(15)
    -> DETERMINISTIC
    -> BEGIN
    ->   DECLARE val INT DEFAULT 0;
    ->   DECLARE level varchar(15);
    ->   SELECT Cost INTO val FROM Product WHERE ProductName=PName;
    ->   IF val<10000 THEN
    ->   SET level='AFFORDABLE';
    ->   ELSEIF val<=70000 THEN
    ->   SET level='MID-RANGE';
    ->   ELSE
    ->   SET level='EXPENSIVE';
    ->   END IF;
    ->   RETURN (level);
    -> END //
Query OK, 0 rows affected (0.23 sec)

mysql> DELIMITER ;
mysql> SELECT GetProductLevel('Sony Bravia') AS ProductLevel;
+--------------+
| ProductLevel |
+--------------+
| EXPENSIVE    |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT GetProductLevel('Bisleri') AS ProductLevel;
+---------------+
| ProductLevel  |
+---------------+
| AFFORDABLE    |
+---------------+
1 row in set (0.00 sec)
```

3. **Find the most expensive product**

```
mysql> DELIMITER //
mysql> CREATE FUNCTION GetExpensiveProduct()
    -> RETURNS varchar(20)
    -> DETERMINISTIC
    -> BEGIN
    ->   DECLARE val varchar(20);
    ->   SELECT ProductName INTO val FROM Product WHERE Cost=(SELECT MAX(Cost) FROM Product P) ;
    ->   RETURN val;
    -> END //
Query OK, 0 rows affected (1.74 sec)

mysql> DELIMITER ;
mysql> SELECT GetExpensiveProduct() AS ExpensiveProduct;
+------------------+
| ExpensiveProduct |
+------------------+
| iPhone X         |
+------------------+
1 row in set (0.00 sec)
```

4. **Find the stock of a particular product.**

```
mysql> DELIMITER //
mysql> CREATE FUNCTION GetStock(ProductID VARCHAR(7))
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->   DECLARE Stock INT;
    ->   SELECT Quantity INTO Stock FROM Product P
    ->   WHERE P.ProductID=ProductID;
    ->   RETURN Stock;
    -> END //
Query OK, 0 rows affected (0.27 sec)

mysql> DELIMITER ;
mysql> SELECT GetStock('pid1003') AS Stock;
+-------+
| Stock |
+-------+
|    40 |
+-------+
1 row in set (0.01 sec)

mysql> SELECT GetStock('pid1006') AS Stock;
+-------+
| Stock |
+-------+
|     7 |
+-------+
1 row in set (0.00 sec)
```

5. **Find the phone number of seller of a Product**

```
mysql> DELIMITER //
mysql> CREATE FUNCTION GetContactDetailOfSeller(ProductID VARCHAR(7))
    -> RETURNS VARCHAR(10)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE ContactNo VARCHAR(10);
    -> SELECT S.PhoneNumber INTO ContactNo FROM Product P, Seller S
    -> WHERE P.ProductID=ProductID AND S.SellerID=P.SellerID;
    -> RETURN ContactNo;
    -> END //
Query OK, 0 rows affected (1.20 sec)

mysql> DELIMITER ;
mysql> SELECT GetContactDetailOfSeller('pid1005') AS ContactNo;
+------------+
| ContactNo  |
+------------+
| 9923657433 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT GetContactDetailOfSeller('pid1011') AS ContactNo;
+------------+
| ContactNo  |
+------------+
| 9923657435 |
+------------+
1 row in set (0.01 sec)
```

Triggers:-

1. **Buy product (Update payment,product, orders tables on remove from CartItem)**

```
mysql> DELIMITER //
mysql> CREATE TRIGGER BuyProductTrigger
    -> AFTER DELETE ON CartItem
    -> FOR EACH ROW
    -> BEGIN
    ->   DECLARE Quantity INT;
    ->   DECLARE Total INT;
    ->   DECLARE Price INT;
    ->   DECLARE PaymentID VARCHAR(7);
    ->   SELECT Cost INTO Price FROM Product WHERE ProductID=old.ProductID;
    ->   SET Total=Price*old.Quantity;
    ->   INSERT INTO Payment(PaymentType,TotalAmount) VALUES ('COD',Total);
    ->   SELECT OrderID INTO PaymentID FROM Payment ORDER BY OrderID DESC LIMIT 1;
    ->   INSERT INTO Orders VALUES (PaymentID,old.CustomerID,old.ProductID,CURDATE());
    ->   UPDATE Product P SET P.Quantity=P.Quantity-old.Quantity
    ->   WHERE P.ProductID=old.ProductID;
    -> END //
Query OK, 0 rows affected (0.64 sec)

mysql> DELIMITER ;
mysql>
mysql> DROP PROCEDURE IF EXISTS BuyProduct;
Query OK, 0 rows affected (1.25 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE BuyProduct(IN CustomerID VARCHAR(7),IN ProductID VARCHAR(7),IN PaymentType VARCHAR(7))
    -> BEGIN
    ->   DECLARE PaymentID VARCHAR(7);
    ->   DELETE FROM CartItem C
    ->   WHERE C.CustomerID=CustomerID AND C.ProductID=ProductID;
    ->   SELECT 'Your Order Has Been Placed' as OrderPlaced;
    ->   SELECT OrderID INTO PaymentID FROM Payment ORDER BY OrderID DESC LIMIT 1;
    ->   UPDATE Payment P SET P.PaymentType=PaymentType WHERE P.OrderID=PaymentID;
    -> END //
Query OK, 0 rows affected (0.41 sec)

mysql> DELIMITER ;
```

BuyProduct Procedure will delete entry from cartItem and the BuyProductTrigger will add corresponding entry in Payment and Orders and update Product table

Below query depicts the same

Before calling BuyProduct:

```
mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       15 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1009   |        3 |         120 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
11 rows in set (0.01 sec)

mysql> SELECT * FROM Product;
+-----------+-------------+-------------------+--------+----------+----------+
| ProductID | ProductType | ProductName       | Cost   | Quantity | SellerID |
+-----------+-------------+-------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter      |    540 |       10 | sid101   |
| pid1002   | Fashion     | Makeup            |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X          | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream         |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad |    180 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia       |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator   |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter       |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke              |     36 |       50 | sid101   |
| pid1010   | Eatables    | Bread             |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri           |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle            |    108 |       18 | sid101   |
+-----------+-------------+-------------------+--------+----------+----------+
12 rows in set (0.00 sec)

mysql> SELECT * FROM Orders;
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000002 | 10001      | pid1003   | 2019-10-23 |
| 1000003 | 10005      | pid1006   | 2018-10-15 |
| 1000004 | 10003      | pid1001   | 2019-10-05 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000006 | 10007      | pid1001   | 2019-10-08 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000008 | 10003      | pid1009   | 2019-10-02 |
| 1000009 | 10001      | pid1007   | 2019-10-14 |
| 1000010 | 10009      | pid1012   | 2018-10-02 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000012 | 10007      | pid1011   | 2018-10-02 |
| 1000013 | 10005      | pid1005   | 2018-10-17 |
| 1000014 | 10001      | pid1004   | 2018-10-18 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
+---------+------------+-----------+------------+
15 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM Payment;
+---------+-------------+-------------+
| OrderID | PaymentType | TotalAmount |
+---------+-------------+-------------+
| 1000001 | Net Banking |         800 |
| 1000002 | Credit/Debit |      100000 |
| 1000003 | Net Banking |       75000 |
| 1000004 | UPI         |         600 |
| 1000005 | COD         |       45000 |
| 1000006 | COD         |         600 |
| 1000007 | UPI         |         200 |
| 1000008 | Credit/Debit |         40 |
| 1000009 | COD         |        5000 |
| 1000010 | UPI         |         120 |
| 1000011 | UPI         |          10 |
| 1000012 | COD         |          10 |
| 1000013 | Net Banking |         200 |
| 1000014 | COD         |          30 |
| 1000015 | UPI         |          30 |
+---------+-------------+-------------+
15 rows in set (0.00 sec)

mysql> CALL BuyProduct(10008,'pid1009','Net Banking');
+---------------------------+
| OrderPlaced               |
+---------------------------+
| Your Order Has Been Placed |
+---------------------------+
1 row in set (0.34 sec)

Query OK, 1 row affected, 1 warning (1.84 sec)

mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       15 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
10 rows in set (0.00 sec)
```

After calling BuyProduct (Trigger will update Orders and Payment Table):

```
mysql> SELECT * FROM Product;
+-----------+-------------+-------------------+--------+----------+----------+
| ProductID | ProductType | ProductName       | Cost   | Quantity | SellerID |
+-----------+-------------+-------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter      |    540 |       10 | sid101   |
| pid1002   | Fashion     | Makeup            |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X          | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream         |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad |    180 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia       |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator   |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter       |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke              |     36 |       47 | sid101   |
| pid1010   | Eatables    | Bread             |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri           |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle            |    108 |       18 | sid101   |
+-----------+-------------+-------------------+--------+----------+----------+
12 rows in set (0.00 sec)

mysql> SELECT * FROM Orders;
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000002 | 10001      | pid1003   | 2019-10-23 |
| 1000003 | 10005      | pid1006   | 2018-10-15 |
| 1000004 | 10003      | pid1001   | 2019-10-05 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000006 | 10007      | pid1001   | 2019-10-08 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000008 | 10003      | pid1009   | 2019-10-02 |
| 1000009 | 10001      | pid1007   | 2019-10-14 |
| 1000010 | 10009      | pid1012   | 2018-10-02 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000012 | 10007      | pid1011   | 2018-10-02 |
| 1000013 | 10005      | pid1005   | 2018-10-17 |
| 1000014 | 10001      | pid1004   | 2018-10-18 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
| 1000016 | 10008      | pid1009   | 2021-04-04 |
+---------+------------+-----------+------------+
16 rows in set (0.00 sec)

mysql> SELECT * FROM Payment;
+---------+--------------+-------------+
| OrderID | PaymentType  | TotalAmount |
+---------+--------------+-------------+
| 1000001 | Net Banking  |         800 |
| 1000002 | Credit/Debit |      100000 |
| 1000003 | Net Banking  |       75000 |
| 1000004 | UPI          |         600 |
| 1000005 | COD          |       45000 |
| 1000006 | COD          |         600 |
| 1000007 | UPI          |         200 |
| 1000008 | Credit/Debit |          40 |
| 1000009 | COD          |        5000 |
| 1000010 | UPI          |         120 |
| 1000011 | UPI          |          10 |
| 1000012 | COD          |          10 |
| 1000013 | Net Banking  |         200 |
| 1000014 | COD          |          30 |
| 1000015 | UPI          |          30 |
| 1000016 | Net Ban      |         108 |
+---------+--------------+-------------+
16 rows in set (0.00 sec)
```

2. Discount

```
mysql> DELIMITER //
mysql> CREATE TRIGGER ApplyDiscount
    -> BEFORE INSERT ON Payment
    -> FOR EACH ROW
    -> BEGIN
    ->    IF new.TotalAmount>=75000 THEN
    ->    SET new.TotalAmount=0.9*new.TotalAmount;
    -> END IF;
    -> END//
Query OK, 0 rows affected (0.88 sec)

mysql> DELIMITER ;
mysql>
mysql> SELECT * FROM CartItem;
+------------+-----------+----------+-------------+
| CustomerID | ProductID | Quantity | TotalAmount |
+------------+-----------+----------+-------------+
| 10001      | pid1001   |        2 |        1200 |
| 10001      | pid1011   |       15 |         200 |
| 10002      | pid1004   |        3 |          90 |
| 10002      | pid1005   |        2 |         400 |
| 10004      | pid1006   |        1 |       75000 |
| 10004      | pid1009   |        5 |         200 |
| 10006      | pid1002   |        1 |         800 |
| 10007      | pid1005   |        1 |         200 |
| 10008      | pid1011   |       10 |         100 |
| 10009      | pid1003   |        1 |      100000 |
+------------+-----------+----------+-------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Product;
+-----------+-------------+--------------------+--------+----------+----------+
| ProductID | ProductType | ProductName        | Cost   | Quantity | SellerID |
+-----------+-------------+--------------------+--------+----------+----------+
| pid1001   | Book        | Harry Potter       |    540 |       10 | sid101   |
| pid1002   | Fashion     | Makeup             |    800 |       18 | sid105   |
| pid1003   | Electronics | iPhone X           | 100000 |       40 | sid109   |
| pid1004   | Eatables    | Ice Cream          |     30 |       15 | sid104   |
| pid1005   | Book        | Rich Dad Poor Dad  |    180 |       15 | sid101   |
| pid1006   | Electronics | Sony Bravia        |  75000 |        7 | sid106   |
| pid1007   | Footwear    | Adidas Predator    |   5000 |       20 | sid108   |
| pid1008   | Vehicle     | TVS Jupiter        |  45000 |        5 | sid102   |
| pid1009   | Eatables    | Coke               |     36 |       47 | sid101   |
| pid1010   | Eatables    | Bread              |     20 |      100 | sid103   |
| pid1011   | Eatables    | Bisleri            |     10 |      100 | sid103   |
| pid1012   | Book        | Tinkle             |    108 |       18 | sid101   |
+-----------+-------------+--------------------+--------+----------+----------+
12 rows in set (0.00 sec)

mysql> CALL BuyProduct(10009,'pid1003','COD');
+---------------------------+
| OrderPlaced               |
+---------------------------+
| Your Order Has Been Placed |
+---------------------------+
1 row in set (0.42 sec)

Query OK, 0 rows affected (0.42 sec)
```

```
mysql> SELECT * FROM Payment;
+---------+--------------+-------------+
| OrderID | PaymentType  | TotalAmount |
+---------+--------------+-------------+
| 1000001 | Net Banking  |         800 |
| 1000002 | Credit/Debit |      100000 |
| 1000003 | Net Banking  |       75000 |
| 1000004 | UPI          |         600 |
| 1000005 | COD          |       45000 |
| 1000006 | COD          |         600 |
| 1000007 | UPI          |         200 |
| 1000008 | Credit/Debit |          40 |
| 1000009 | COD          |        5000 |
| 1000010 | UPI          |         120 |
| 1000011 | UPI          |          10 |
| 1000012 | COD          |          10 |
| 1000013 | Net Banking  |         200 |
| 1000014 | COD          |          30 |
| 1000015 | UPI          |          30 |
| 1000016 | Net Ban      |         108 |
| 1000017 | COD          |       90000 |
+---------+--------------+-------------+
17 rows in set (0.00 sec)
```

3. Refund

```
mysql> DELIMITER //
mysql> CREATE TRIGGER Refund
    -> AFTER DELETE ON Orders
    -> FOR EACH ROW
    -> BEGIN
    ->   DELETE FROM Payment
    ->   WHERE OrderID=old.OrderID;
    -> END//
Query OK, 0 rows affected (1.33 sec)

mysql> DELIMITER ;
mysql>
mysql> DROP PROCEDURE IF EXISTS ReturnOrder;
Query OK, 0 rows affected (0.29 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE ReturnOrder(IN OrderID VARCHAR(7))
    -> BEGIN
    ->   DELETE FROM Orders AS O
    ->   WHERE O.OrderID = OrderID;
    -> END//
Query OK, 0 rows affected (0.36 sec)

mysql> DELIMITER ;
mysql>
mysql> SELECT * FROM Orders;
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000002 | 10001      | pid1003   | 2019-10-23 |
| 1000003 | 10005      | pid1006   | 2018-10-15 |
| 1000004 | 10003      | pid1001   | 2019-10-05 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000006 | 10007      | pid1001   | 2019-10-08 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000008 | 10003      | pid1009   | 2019-10-02 |
| 1000009 | 10001      | pid1007   | 2019-10-14 |
| 1000010 | 10009      | pid1012   | 2018-10-02 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000012 | 10007      | pid1011   | 2018-10-02 |
| 1000013 | 10005      | pid1005   | 2018-10-17 |
| 1000014 | 10001      | pid1004   | 2018-10-18 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
| 1000016 | 10008      | pid1009   | 2021-04-04 |
| 1000017 | 10009      | pid1003   | 2021-04-04 |
+---------+------------+-----------+------------+
17 rows in set (0.00 sec)

mysql> SELECT * FROM Payment;
+---------+-------------+-------------+
| OrderID | PaymentType | TotalAmount |
+---------+-------------+-------------+
| 1000001 | Net Banking |         800 |
| 1000002 | Credit/Debit |      100000 |
| 1000003 | Net Banking |        75000 |
| 1000004 | UPI         |          600 |
| 1000005 | COD         |        45000 |
| 1000006 | COD         |          600 |
| 1000007 | UPI         |          200 |
| 1000008 | Credit/Debit |          40 |
| 1000009 | COD         |         5000 |
| 1000010 | UPI         |          120 |
| 1000011 | UPI         |           10 |
| 1000012 | COD         |           10 |
| 1000013 | Net Banking |          200 |
| 1000014 | COD         |           30 |
| 1000015 | UPI         |           30 |
| 1000016 | Net Ban     |          108 |
| 1000017 | COD         |        90000 |
+---------+-------------+-------------+
17 rows in set (0.01 sec)
```

RefundOrder Procedure will Delete the Order entry from Orders table
And Refund trigger will remove corresponding entry from Payment

```
mysql> CALL ReturnOrder('1000004');
Query OK, 1 row affected (0.29 sec)

mysql> SELECT * FROM Orders;
+---------+------------+-----------+------------+
| OrderID | CustomerID | ProductID | OrderDate  |
+---------+------------+-----------+------------+
| 1000001 | 10003      | pid1002   | 2020-10-12 |
| 1000002 | 10001      | pid1003   | 2019-10-23 |
| 1000003 | 10005      | pid1006   | 2018-10-15 |
| 1000005 | 10006      | pid1008   | 2020-10-06 |
| 1000006 | 10007      | pid1001   | 2019-10-08 |
| 1000007 | 10005      | pid1005   | 2020-10-12 |
| 1000008 | 10003      | pid1009   | 2019-10-02 |
| 1000009 | 10001      | pid1007   | 2019-10-14 |
| 1000010 | 10009      | pid1012   | 2018-10-02 |
| 1000011 | 10004      | pid1011   | 2020-10-13 |
| 1000012 | 10007      | pid1011   | 2018-10-02 |
| 1000013 | 10005      | pid1005   | 2018-10-17 |
| 1000014 | 10001      | pid1004   | 2018-10-18 |
| 1000015 | 10003      | pid1004   | 2020-10-02 |
| 1000016 | 10008      | pid1009   | 2021-04-04 |
| 1000017 | 10009      | pid1003   | 2021-04-04 |
+---------+------------+-----------+------------+
16 rows in set (0.00 sec)

mysql> SELECT * FROM Payment;
+---------+-------------+-------------+
| OrderID | PaymentType | TotalAmount |
+---------+-------------+-------------+
| 1000001 | Net Banking |         800 |
| 1000002 | Credit/Debit |      100000 |
| 1000003 | Net Banking |        75000 |
| 1000005 | COD         |        45000 |
| 1000006 | COD         |          600 |
| 1000007 | UPI         |          200 |
| 1000008 | Credit/Debit |          40 |
| 1000009 | COD         |         5000 |
| 1000010 | UPI         |          120 |
| 1000011 | UPI         |           10 |
| 1000012 | COD         |           10 |
| 1000013 | Net Banking |          200 |
| 1000014 | COD         |           30 |
| 1000015 | UPI         |           30 |
| 1000016 | Net Ban     |          108 |
| 1000017 | COD         |        90000 |
+---------+-------------+-------------+
16 rows in set (0.00 sec)
```