

Assignment -1

TechShop, an electronic gadgets shop

Submitted by : Mridul Bhardwaj

Task:1. Database Design:

1. Create the database named "TechShop"

```
mysql> create database TechShop;  
Query OK, 1 row affected (0.04 sec)  
  
mysql>
```

2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.

```
mysql> desc Customers;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| CustomerID | int       | NO   | PRI | NULL    |       |  
| FirstName  | text     | YES  |     | NULL    |       |  
| LastName   | text     | YES  |     | NULL    |       |  
| Email      | text     | YES  |     | NULL    |       |  
| Phone      | int      | YES  |     | NULL    |       |  
| Address    | varchar(99) | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
6 rows in set (0.04 sec)
```

```
mysql> desc Products;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| ProductID  | int       | NO   | PRI | NULL    |       |  
| ProductName | text     | YES  |     | NULL    |       |  
| Description | text     | YES  |     | NULL    |       |  
| Price      | float    | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

```
mysql> desc Orders;
```

Field	Type	Null	Key	Default	Extra
OrderID	int	NO	PRI	NULL	
CustomerID	int	YES	MUL	NULL	
OrderDate	date	YES		NULL	
TotalAmount	decimal(10,2)	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> desc OrderDetails;
```

Field	Type	Null	Key	Default	Extra
OrderDetailID	int	NO	PRI	NULL	
OrderID	int	YES	MUL	NULL	
ProductID	int	YES	MUL	NULL	
Quantity	decimal(10,0)	YES		NULL	

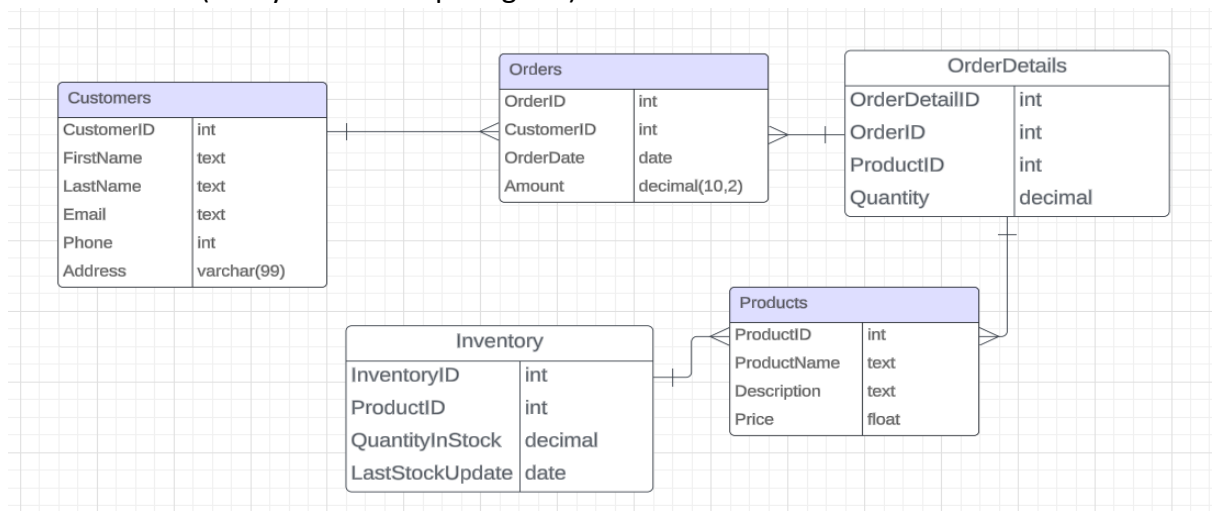
4 rows in set (0.01 sec)

```
mysql> desc Inventory;
```

Field	Type	Null	Key	Default	Extra
InventoryID	int	NO	PRI	NULL	
ProductID	int	YES	MUL	NULL	
QuantityInStock	decimal(10,0)	YES		NULL	
LastStockUpdate	date	YES		NULL	

4 rows in set (0.01 sec)

3. Create an ERD (Entity Relationship Diagram) for the database.



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

```
mysql> alter table Customers add primary key(CustomerID);
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table Products add primary key(ProductID);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> create table Orders (OrderID int primary key, CustomerID int, OrderDate date, TotalAmount decimal(10,2), foreign key(CustomerID) references Customers (CustomerID));
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> create table OrderDetails (OrderDetailID int primary key, OrderID int, ProductID int, Quantity decimal, foreign key(OrderID) references Orders(OrderID), foreign key(ProductID) references Products(ProductID));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> show tables;
```

```
mysql> create table Inventory(InventoryID int primary key, ProductID int, QuantityInStock decimal, LastStockUpdate date, foreign key(ProductID) references Products(ProductID));
Query OK, 0 rows affected (0.09 sec)
```

5. Insert at least 10 sample records into each of the following tables.

a. Customers

```
mysql> insert into Customers (CustomerID, FirstName, LastName, Email, Phone, Address) values
-> (5, "Divya", "Raghav", "div@abc.com", 1056784320, "56 Neerja ");
Query OK, 1 row affected (0.01 sec)

mysql> insert into Customers (CustomerID, FirstName, LastName, Email, Phone, Address) values
-> (6, "Dev", "Verma", "dec@xyz.com", 1045786320, "45 Palash");
Query OK, 1 row affected (0.01 sec)

mysql> insert into Customers (CustomerID, FirstName, LastName, Email, Phone, Address) values
-> (7, "Gagan", "Soni", "gagan@abc.com", 1076543210, "50 Sarojini"),
-> (8, "Sachin", "Tyagi", "sachin@pqr.com", 1894561230, "78 Palika Bazar"),
-> (9, "Rishi", "Thakur", "rishi@exy.com", 1745961230, "74 Saket"),
-> (10, "Radha", "Mohan", "radha@qxy.com", 1245874523, "89 Ecovillage");
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Customers;;
```

CustomerID	FirstName	LastName	Email	Phone	Address
1	Akash	Nigam	akash@example.com	1234567890	17 Om Nagar
2	Aman	Sharma	aman@yahoo.com	1478523690	140 Anmol Vihar
3	Chetan	Singh	chetan@outlook.com	1237894560	152 Gokul
4	Astha	Sharma	astha@yahoo.com	1478523690	10 Anmol Vihar
5	Divya	Raghav	div@abc.com	1056784320	56 Neerja
6	Dev	Verma	dec@xyz.com	1045786320	45 Palash
7	Gagan	Soni	gagan@abc.com	1076543210	50 Sarojini
8	Sachin	Tyagi	sachin@pqr.com	1894561230	78 Palika Bazar
9	Rishi	Thakur	rishi@exy.com	1745961230	74 Saket
10	Radha	Mohan	radha@qxy.com	1245874523	89 Ecovillage

```
10 rows in set (0.01 sec)
```

b. Products

```
mysql> insert into Products (ProductID, ProductName, Description, Price) values
-> (101, "Speaker", "Bluetooth portable", 99.99);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> insert into Products (ProductID, ProductName, Description, Price) values
-> (102, "Mic", "Wireless", 50.55),
-> (103, "TV", "Smart and Ultra", 450.55),
-> (104, "Remote", "Chargable", 40.55),
-> (105, "Laptop", "16 inch and high battery", 250.00),
-> (106, "LED", "32 inch", 540.00),
-> (107, "Mouse", "Wireless", 80.85),
-> (108, "Tablet", "High resolution Camera", 150.55),
-> (109, "Watch", "Smart", 62.45),
-> (110, "Fridge", "Double door", 500.22);
Query OK, 9 rows affected (0.01 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Products;
```

ProductID	ProductName	Description	Price
101	Speaker	Bluetooth portable	99.99
102	Mic	Wireless	50.55
103	TV	Smart and Ultra	450.55
104	Remote	Chargable	40.55
105	Laptop	16 inch and high battery	250
106	LED	32 inch	540
107	Mouse	Wireless	80.85
108	Tablet	High resolution Camera	150.55
109	Watch	Smart	62.45
110	Fridge	Double door	500.22

```
10 rows in set (0.00 sec)
```

c. Orders

```
mysql> insert into Orders values
-> (201,1,'2024-01-01',1500.00),
-> (202,2,'2024-01-02',750.00),
-> (203,3,'2024-01-03',900.00),
-> (204,4,'2024-01-04',100.00),
-> (205,5,'2024-01-05',1000.55),
-> (206,6,'2024-01-06',1050.00),
-> (207,7,'2024-01-07',150.00),
-> (208,8,'2024-01-08',300.00),
-> (209,9,'2024-01-09',120.00),
-> (210,10,'2024-01-10',90.00);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 201 | 1 | 2024-01-01 | 1500.00 |
| 202 | 2 | 2024-01-02 | 750.00 |
| 203 | 3 | 2024-01-03 | 900.00 |
| 204 | 4 | 2024-01-04 | 100.00 |
| 205 | 5 | 2024-01-05 | 1000.55 |
| 206 | 6 | 2024-01-06 | 1050.00 |
| 207 | 7 | 2024-01-07 | 150.00 |
| 208 | 8 | 2024-01-08 | 300.00 |
| 209 | 9 | 2024-01-09 | 120.00 |
| 210 | 10 | 2024-01-10 | 90.00 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

d. OrderDetails

```
mysql> insert into OrderDetails values
-> (301,201,101,10),
-> (302,202,102,12),
-> (303,203,103,14),
-> (304,204,104,16),
-> (305,205,105,18),
-> (306,206,106,20),
-> (307,207,107,22),
-> (308,208,108,24),
-> (309,209,109,26),
-> (310,210,110,28);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from OrderDetails;
+-----+-----+-----+-----+
| OrderDetailID | OrderID | ProductID | Quantity |
+-----+-----+-----+-----+
| 301 | 201 | 101 | 10 |
| 302 | 202 | 102 | 12 |
| 303 | 203 | 103 | 14 |
| 304 | 204 | 104 | 16 |
| 305 | 205 | 105 | 18 |
| 306 | 206 | 106 | 20 |
| 307 | 207 | 107 | 22 |
| 308 | 208 | 108 | 24 |
| 309 | 209 | 109 | 26 |
| 310 | 210 | 110 | 28 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

e. Inventory

```
mysql> insert into Inventory values
-> (401,101,4,'2024-01-01'),
-> (402,102,6,'2024-01-02'),
-> (403,103,8,'2024-01-03'),
-> (404,104,10,'2024-01-04'),
-> (405,105,12,'2024-01-05'),
-> (406,106,14,'2024-01-06'),
-> (407,107,16,'2024-01-07'),
-> (408,108,18,'2024-01-08'),
-> (409,109,20,'2024-01-09'),
-> (410,110,22,'2024-01-10');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Inventory;
+-----+-----+-----+-----+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-----+-----+-----+-----+
| 401 | 101 | 4 | 2024-01-01 |
| 402 | 102 | 6 | 2024-01-02 |
| 403 | 103 | 8 | 2024-01-03 |
| 404 | 104 | 10 | 2024-01-04 |
| 405 | 105 | 12 | 2024-01-05 |
| 406 | 106 | 14 | 2024-01-06 |
| 407 | 107 | 16 | 2024-01-07 |
| 408 | 108 | 18 | 2024-01-08 |
| 409 | 109 | 20 | 2024-01-09 |
| 410 | 110 | 22 | 2024-01-10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Tasks 2: Select, Where, Between, AND, LIKE:

1. Write an SQL query to retrieve the names and emails of all customers.

```
mysql> SELECT FirstName, Email from Customers;
+-----+-----+
| FirstName | Email |
+-----+-----+
| Akash | akash@example.com |
| Aman | aman@yahoo.com |
| Chetan | chetan@outlook.com |
| Astha | astha@yahoo.com |
| Divya | div@abc.com |
| Dev | dec@xyz.com |
| Gagan | gagan@abc.com |
| Sachin | sachin@pqr.com |
| Rishi | rishi@exy.com |
| Radha | radha@qxy.com |
+-----+-----+
10 rows in set (0.01 sec)
```

2. Write an SQL query to list all orders with their order dates and corresponding customer names.

```
mysql> select Orders.OrderID, Orders.OrderDate, Customers.FirstName from Orders join Customers on Orders.CustomerID = Customers.CustomerID;
+-----+-----+-----+
| OrderID | OrderDate | FirstName |
+-----+-----+-----+
| 201 | 2024-01-01 | Akash |
| 202 | 2024-01-02 | Aman |
| 203 | 2024-01-03 | Chetan |
| 204 | 2024-01-04 | Astha |
| 205 | 2024-01-05 | Divya |
| 206 | 2024-01-06 | Dev |
| 207 | 2024-01-07 | Gagan |
| 208 | 2024-01-08 | Sachin |
| 209 | 2024-01-09 | Rishi |
| 210 | 2024-01-10 | Radha |
+-----+-----+-----+
10 rows in set (0.02 sec)
```

- Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

```
mysql> insert into Customers values (11, "Vijay", "Kumar", "vijay@abc.com", 1237890456, "78 Rohini");
Query OK, 1 row affected (0.03 sec)
```

- Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.

```
mysql> update Products set price = price * 1.10;
Query OK, 10 rows affected (0.02 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from Products;
+-----+-----+-----+-----+
| ProductID | ProductName | Description | Price |
+-----+-----+-----+-----+
| 101 | Speaker | Bluetooth portable | 109.989 |
| 102 | Mic | Wireless | 55.605 |
| 103 | TV | Smart and Ultra | 495.605 |
| 104 | Remote | Chargable | 44.605 |
| 105 | Laptop | 16 inch and high battery | 275 |
| 106 | LED | 32 inch | 594 |
| 107 | Mouse | Wireless | 88.935 |
| 108 | Tablet | High resolution Camera | 165.605 |
| 109 | Watch | Smart | 68.695 |
| 110 | Fridge | Double door | 550.242 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

- Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

```
mysql> delimiter !!
mysql> create procedure DeleteOrder(in CustomerIDParam int)
-> begin
-> delete from OrderDetails
-> where OrderID in (select OrderID from Orders where CustomerID = CustomerIDParam);
-> delete from Orders where CustomerID = CustomerIDParam;
-> end !!
Query OK, 0 rows affected (0.07 sec)

mysql> delimiter ;
mysql> set @m = 4;
Query OK, 0 rows affected (0.01 sec)

mysql> call DeleteOrder(@m);
Query OK, 1 row affected (0.02 sec)
```

6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

```
mysql> insert into Orders values (211,11,'2024-01-11',250.00);
Query OK, 1 row affected (0.01 sec)
```

7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.

```
mysql> delimiter @@
mysql> create procedure UpdateContact(inout NewEmail text, inout UpdateCustID int)
-> begin
-> update Customers set email = NewEmail where CustomerID = UpdateCustID;
-> end @@
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> set @E = 'sajan@abc.com';
Query OK, 0 rows affected (0.00 sec)

mysql> set @id = '1';
Query OK, 0 rows affected (0.00 sec)

mysql> call UpdateContact(@E, @id);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address
1	Akash	Nigam	sajan@abc.com	1234567890	17 Om Nagar
2	Aman	Sharma	aman@yahoo.com	1478523690	140 Anmol Vihar
3	Chetan	Singh	chetan@outlook.com	1237894560	152 Gokul
4	Astha	Sharma	astha@yahoo.com	1478523690	10 Anmol Vihar
5	Divya	Raghav	div@abc.com	1056784320	56 Neerja
6	Dev	Verma	dec@xyz.com	1045786320	45 Palash
7	Gagan	Soni	gagan@abc.com	1076543210	50 Sarojini
8	Sachin	Tyagi	sachin@pqr.com	1894561230	78 Palika Bazar
9	Rishi	Thakur	rishi@exy.com	1745961230	74 Saket
10	Radha	Mohan	radha@qxy.com	1245874523	89 Ecovillage
11	Vijay	Kumar	vijay@abc.com	1237890456	78 Rohini

```
11 rows in set (0.00 sec)
```

8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.


```
mysql> update Orders
-> set TotalAmount = (
-> select sum(od.quantity*p.price)
-> from OrderDetails od
-> join Products p on od.ProductID = p.ProductID
-> where od.OrderID = Orders.OrderID)
-> where OrderID in(select distinct OrderID from OrderDetails);
Query OK, 10 rows affected, 8 warnings (0.05 sec)
Rows matched: 10 Changed: 10 Warnings: 8

mysql> select * from Orders;
+-----+-----+-----+-----+
| OrderID | CustomerID | OrderDate | TotalAmount |
+-----+-----+-----+-----+
| 201 | 1 | 2024-01-01 | 1099.89 |
| 202 | 2 | 2024-01-02 | 667.26 |
| 203 | 3 | 2024-01-03 | 6938.47 |
| 204 | 4 | 2024-01-04 | 713.68 |
| 205 | 5 | 2024-01-05 | 4950.00 |
| 206 | 6 | 2024-01-06 | 11880.00 |
| 207 | 7 | 2024-01-07 | 1956.57 |
| 208 | 8 | 2024-01-08 | 3974.52 |
| 209 | 9 | 2024-01-09 | 1786.07 |
| 210 | 10 | 2024-01-10 | 15406.78 |
| 211 | 11 | 2024-01-11 | 250.00 |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.

```
mysql> create procedure DelCustOrders(in CustID int)
-> begin
-> delete from OrderDetails where OrderID in(select OrderID from Orders where CustomerID = CustID);
-> delete from Orders where CustomerID = CustID;
-> end##
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> set @m = '2';
Query OK, 0 rows affected (0.00 sec)

mysql> call DelCustOrders(@m);
Query OK, 1 row affected (0.01 sec)
```

10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

```
mysql> insert into Products values (111, "Smartphone", "High Battery Life", 170.00);
Query OK, 1 row affected (0.01 sec)
```

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

```
mysql> select OrderID, OrderDate, if(OrderDate > '2024-01-07', 'Shipped','Pending') from Orders;
```

OrderID	OrderDate	if(OrderDate > '2024-01-07', 'Shipped','Pending')
201	2024-01-01	Pending
203	2024-01-03	Pending
205	2024-01-05	Pending
206	2024-01-06	Pending
207	2024-01-07	Pending
208	2024-01-08	Shipped
209	2024-01-09	Shipped
210	2024-01-10	Shipped
211	2024-01-11	Shipped

```
9 rows in set (0.00 sec)
```

12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.

```
mysql> select count(OrderID) from Orders, Customers where Orders.CustomerID = Customers.CustomerID;
```

count(OrderID)
11

```
1 row in set (0.01 sec)
```

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

- Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

```
mysql> SELECT Orders.orderid, Orders.orderdate, Customers.customerid, Customers.firstname
-> FROM Orders
-> JOIN Customers ON Orders.customerid = Customers.customerid;
```

orderid	orderdate	customerid	firstname
201	2024-01-01	1	Akash
203	2024-01-03	3	Chetan
205	2024-01-05	5	Divya
206	2024-01-06	6	Dev
207	2024-01-07	7	Gagan
208	2024-01-08	8	Sachin
209	2024-01-09	9	Rishi
210	2024-01-10	10	Radha
211	2024-01-11	11	Vijay

```
9 rows in set (0.01 sec)
```

- Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue

```
mysql> select Products.ProductID, ProductName, sum(Quantity * Price) as TotalRevenue
-> from OrderDetails
-> join Products on OrderDetails.ProductID = Products.ProductID
-> group by Products.ProductID, ProductName;
```

ProductID	ProductName	TotalRevenue
101	Speaker	1099.8899841308594
103	TV	6938.4697265625
105	Laptop	4950
106	LED	11880
107	Mouse	1956.5699462890625
108	Tablet	3974.520263671875
109	Watch	1786.0699920654297
110	Fridge	15406.776123046875

```
8 rows in set (0.01 sec)
```

3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

```
mysql> select FirstName, LastName, Email from Customers
-> where CustomerID in (select distinct CustomerID from Orders);
```

FirstName	LastName	Email
Akash	Nigam	sajan@abc.com
Chetan	Singh	chetan@outlook.com
Divya	Raghav	div@abc.com
Dev	Verma	dec@xyz.com
Gagan	Soni	gagan@abc.com
Sachin	Tyagi	sachin@pqr.com
Rishi	Thakur	rishi@exy.com
Radha	Mohan	radha@qxy.com
Vijay	Kumar	vijay@abc.com

9 rows in set (0.00 sec)

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

```
mysql> select OrderDetails.productID, ProductName, sum(quantity) as TotalQuantityOrdered from OrderDetails
-> join Products on OrderDetails.ProductID = Products.ProductID
-> group by ProductID, ProductName order by TotalQuantityOrdered desc
-> limit 1;
```

productID	ProductName	TotalQuantityOrdered
110	Fridge	28

1 row in set (0.01 sec)

5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

```
mysql> select ProductName, description from Products;
```

ProductName	description
Speaker	Bluetooth portable
Mic	Wireless
TV	Smart and Ultra
Remote	Chargable
Laptop	16 inch and high battery
LED	32 inch
Mouse	Wireless
Tablet	High resolution Camera
Watch	Smart
Fridge	Double door
Smartphone	High Battery Life

11 rows in set (0.01 sec)

6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.

```
mysql> select Customers.CustomerID,FirstName,LastName, avg(TotalAmount) as AverageOrderValue
-> from Orders
-> join Customers on Orders.CustomerID = Customers.CustomerID
-> group by Customers.CustomerID, FirstName, LastName;
```

CustomerID	FirstName	LastName	AverageOrderValue
1	Akash	Nigam	1099.890000
3	Chetan	Singh	6938.470000
5	Divya	Raghav	4950.000000
6	Dev	Verma	11880.000000
7	Gagan	Soni	1956.570000
8	Sachin	Tyagi	3974.520000
9	Rishi	Thakur	1786.070000
10	Radha	Mohan	15406.780000
11	Vijay	Kumar	250.000000

9 rows in set (0.00 sec)

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.

```
mysql> select OrderID,OrderDate,FirstName,LastName,max(TotalAmount) as MaxTotalRevenue
-> from Orders
-> join Customers on Orders.CustomerID = Customers.CustomerID
-> group by OrderID
-> order by MaxTotalRevenue desc
-> limit 1;
```

OrderID	OrderDate	FirstName	LastName	MaxTotalRevenue
210	2024-01-10	Radha	Mohan	15406.78

1 row in set (0.00 sec)

8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

```
mysql> select Products.ProductID,ProductName,count(OrderDetails.OrderID) as OrderCount
-> from Products
-> left join OrderDetails on Products.ProductID = OrderDetails.ProductID
-> group by ProductID,ProductName;
```

ProductID	ProductName	OrderCount
101	Speaker	1
102	Mic	0
103	TV	1
104	Remote	0
105	Laptop	1
106	LED	1
107	Mouse	1
108	Tablet	1
109	Watch	1
110	Fridge	1
111	Smartphone	0

11 rows in set (0.00 sec)

9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.

```
mysql> select FirstName,LastName,Email from Customers
-> where CustomerID in(select distinct Orders.CustomerID from Orders
-> join OrderDetails on Orders.OrderID = OrderDetails.OrderID
-> where OrderDetails.ProductID = (select ProductID from Products where ProductName='fridge'));
+-----+-----+-----+
| FirstName | LastName | Email      |
+-----+-----+-----+
| Radha     | Mohan    | radha@qxy.com |
+-----+-----+-----+
1 row in set (0.01 sec)
```

10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

```
mysql> delimiter $$
mysql> create procedure TotalRevenueForTimePeriod(in date1 date, in date2 date)
-> begin
-> select sum(TotalAmount) as TotalRevenue from Orders where OrderDate between date1 and date2;
-> end $$
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> call TotalRevenueForTimePeriod('2024-01-06','2024-01-10');
+-----+
| TotalRevenue |
+-----+
| 35003.94 |
+-----+
1 row in set (0.00 sec)

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

Task 4. Subquery and its type:

1. Write an SQL query to find out which customers have not placed any orders.

```
mysql> select CustomerID,FirstName,LastName from Customers
-> where CustomerID not in(select distinct CustomerID from Orders);
+-----+-----+-----+
| CustomerID | FirstName | LastName |
+-----+-----+-----+
| 2          | Aman     | Sharma   |
| 4          | Astha    | Sharma   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Write an SQL query to find the total number of products available for sale.

```
mysql> select count(ProductName) as TotalProduct from Products;
+-----+
| TotalProduct |
+-----+
| 11          |
+-----+
1 row in set (0.01 sec)
```

3. Write an SQL query to calculate the total revenue generated by TechShop.

```
mysql> select sum(TotalAmount) as TotalRevenue from Orders;
+-----+
| TotalRevenue |
+-----+
|      48242.30 |
+-----+
1 row in set (0.00 sec)
```

4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

```
mysql> delimiter @@
mysql> create procedure ShowAvgQuant(in category varchar(50))
    -> begin
    -> select avg(Quantity) as AvgQuantityOrdered from OrderDetails
    -> join Products on OrderDetails.ProductID = Products.ProductID
    -> where ProductName = category;
    -> end @@
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call ShowAvgQuant('Mouse');
+-----+
| AvgQuantityOrdered |
+-----+
|           22.0000 |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

```
mysql> delimiter !!
mysql> create procedure RevenuePerCust(in id int)
    -> begin
    -> select Customers.CustomerID, FirstName, LastName, sum(TotalAmount) as TotalRevenue from Orders
    -> join Customers on Orders.CustomerID = Customers.CustomerID where Orders.CustomerID = id
    -> group by Orders.CustomerID;
    -> end !!
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call RevenuePerCust(7);
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | TotalRevenue |
+-----+-----+-----+-----+
|          7 | Gagan    | Soni     |      1956.57 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

```
mysql> select Customers.CustomerID,FirstName,LastName,count(OrderID) as OrderCount from Orders
-> join Customers on Orders.CustomerID = Customers.CustomerID
-> group by OrderID order by OrderCount desc
-> limit 1;
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | OrderCount |
+-----+-----+-----+-----+
|          1 | Akash    | Nigam    |           1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

```
mysql> select Products.ProductID,sum(Quantity) as TotalQuantityOrdered from OrderDetails
-> join Products on OrderDetails.ProductID = Products.ProductID
-> group by ProductID
-> order by TotalQuantityOrdered desc
-> limit 1;
+-----+-----+
| ProductID | TotalQuantityOrdered |
+-----+-----+
|         110 |                28 |
+-----+-----+
1 row in set (0.00 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

```
mysql> select Customers.CustomerID,FirstName,LastName,sum(TotalAmount) as TotalSpent from Customers
-> join Orders on Customers.CustomerID = Customers.CustomerID
-> group by CustomerID
-> order by TotalSpent desc
-> limit 1;
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | TotalSpent |
+-----+-----+-----+-----+
|          10 | Radha    | Mohan    | 15406.78 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.

```
mysql> select Customers.CustomerID,FirstName,LastName,avg(TotalAmount) as AvgOrderValue from Orders
-> join Customers on Orders.CustomerID = Customers.CustomerID
-> group by CustomerID,FirstName,LastName;
+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | AvgOrderValue |
+-----+-----+-----+-----+
|          1 | Akash    | Nigam    | 1099.890000 |
|          3 | Chetan   | Singh    | 6938.470000 |
|          5 | Divya    | Raghav    | 4950.000000 |
|          6 | Dev      | Verma     | 11880.000000 |
|          7 | Gagan    | Soni     | 1956.570000 |
|          8 | Sachin   | Tyagi     | 3974.520000 |
|          9 | Rishi    | Thakur    | 1786.070000 |
|         10 | Radha    | Mohan    | 15406.780000 |
|         11 | Vijay    | Kumar     | 250.000000 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

```
mysql> select Customers.CustomerID,FirstName,LastName,count(OrderID) as TotalOrders from Customers
-> join Orders on Customers.CustomerID = Orders.CustomerID
-> group by CustomerID;
```

CustomerID	FirstName	LastName	TotalOrders
1	Akash	Nigam	1
3	Chetan	Singh	1
5	Divya	Raghav	1
6	Dev	Verma	1
7	Gagan	Soni	1
8	Sachin	Tyagi	1
9	Rishi	Thakur	1
10	Radha	Mohan	1
11	Vijay	Kumar	1

```
9 rows in set (0.00 sec)
```