**Tech-Shop**

an electronic gadgets shop

**Task:1. Database Design:**

**ER Diagram  
A screenshot of a computer

Description automatically generated**

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

**Ans:-**

CREATE TABLE Customers (

CustomerID int(3) PRIMARY KEY,

Firstname varchar(50),

Lastname varchar(50),

Email varchar(75),

Ph\_no varchar(10),

Address varchar(225)

);

CREATE TABLE Products (

ProductID int(3) PRIMARY KEY,

ProductName varchar(50),

Description varchar(249),

Price decimal(10,2)

);

CREATE TABLE Orders (

OrderID int PRIMARY KEY,

CustomerID int(3),

Orderdate date,

TotalAmount decimal(10, 2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE Orderdetails (

OrderdetailID int(3) PRIMARY KEY,

OrderID int(3),

ProductID int(3),

Quantity int(5),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

CREATE TABLE Inventory (

InventoryID int(5) PRIMARY KEY,

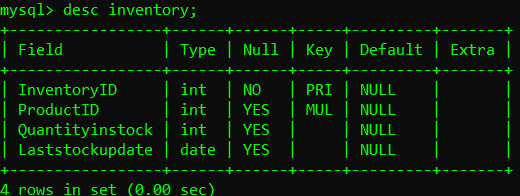
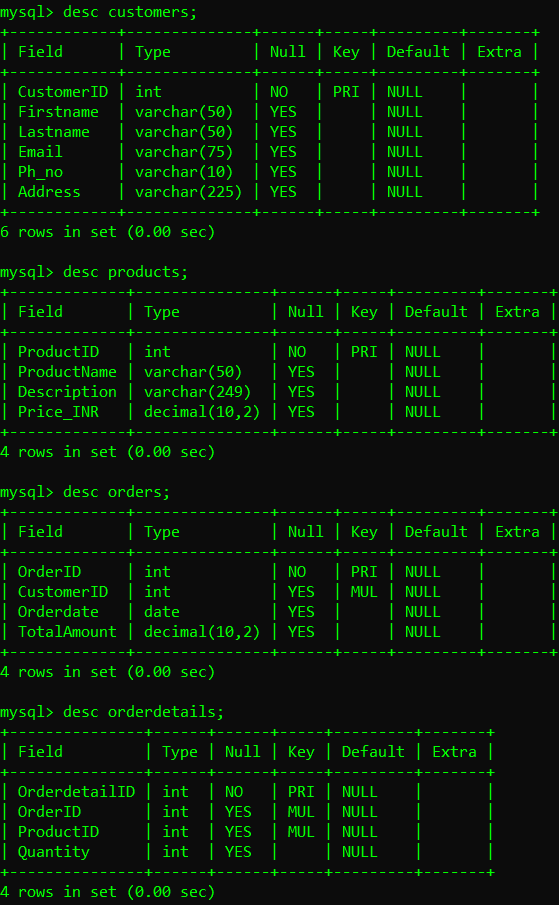
ProductID int(3),

Quantityinstock int(5),

Laststockupdate date,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

**OUTPUT:-** 

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

**Customers**

INSERT INTO Customers (CustomerID,Firstname,Lastname,Email,Ph\_no,Address)

VALUES

(101, 'Eshwar', 'V', 'eshwar@gmail.com', '9782154201', '32,JP nagar,banglore-49'),

(102, 'Reddy', 'V', 'reddy123@gmail.com','9587412368', '45,JJ nagar,banglore'),

(103, 'Tom', 'Son', 'tom143@gmail.com', '9586741230', '36,LB nagar,banglore-49'),

(104, 'Jhon', 'Son', 'jhon456@gmail.com', '9513687421', '97,NR layout,hyd-07'),

(105, 'King', 'Simth', 'king956@gmail.com', '6365987412', '54,JP nagar,banglore-49'),

(106, 'NTR', 'Anna', 'ntrfans@gmail.com', '7898745621', '9,NR layout,hyd-07'),

(107, 'Pavan', 'Kumar', 'pavan@gmail.com', '7775648921', '13,RR nagar,hyd-04'),

(108, 'Pavi', 'Sharma', 'ravis123@gmail.com', '9694562100', '49,MR layout,banglore-48'),

(109, 'Hunk', 'Kumar', 'kumar123@gmail.com', '6457871188', '876 jp nagar,banglore-49'),

(110, 'Muni', 'Reddy', 'muni23@gmail.com', '8364942155', '24,PK nagar,banglore-04');

**Customers OUTPUT: -**

**A screen shot of a computer

Description automatically generated**

**Products**

INSERT INTO Products (ProductID, Productname, Description, Price\_INR)

VALUES

(111, 'HP Gaming Laptop', 'High-performance and Best Display laptop', 55999.00),

(112, 'Nothing Phone(2a)', 'Latest Model with Glyph Ligthing ', 45999.99),

(113, 'Boat 161 ANC Headphones', 'ANC with 4mics Wireless Headphones', 1499.99),

(114, 'Asus Monitor', '4K Curved Display', 85299.00),

(115, 'Acer Keyboard', 'gaming keyboard with RCB Light', 19499.99),

(116, 'Asus Mouse', 'Best Wireless Mouse for Gaming', 8999.99),

(117, 'HP Printer', 'All-in-one Printer', 15549.90),

(118, 'Apple Adopter', 'With 120w Fast Charging',9399.99),

(119, 'Apple Series 8Watch', 'With Two Finger tap Opreating', 35599.99),

(120, 'Apple 16 pro max', 'With New Action Button Feature', 169499.99);

**Products OUTPUT: -**

A screenshot of a computer

Description automatically generated

**Orders**

INSERT INTO Orders (OrderID, CustomerID, Orderdate, TotalAmount)

VALUES

(201, 101, '2024-09-26', 55999.00),

(202, 102, '2024-09-24', 1499.99),

(203, 103, '2024-09-29', 45999.99),

(204, 104, '2024-09-27', 169499.99),

(205, 104, '2024-09-27', 9399.99),

(206, 106, '2024-09-28', 85299.00),

(207, 101, '2024-09-26', 15549.90),

(208, 108, '2024-09-25', 19499.99),

(209, 104, '2024-09-27', 35599.99),

(210, 106, '2024-09-28', 8999.99);

**Order OUTPUT: -**

**A screenshot of a computer screen

Description automatically generated**

**OrderDetails**

INSERT INTO Orderdetails (OrderdetailID, OrderID, ProductID, Quantity)

VALUES

(211, 201, 111, 1),

(212, 202, 113, 3),

(213, 203, 112, 1),

(214, 204, 120, 1),

(215, 205, 118, 2),

(216, 206, 114, 4),

(217, 207, 117, 2),

(218, 208, 115, 1),

(219, 209, 119, 1),

(220, 210, 116, 1);

**Order details OUTPUT: -**

**A screenshot of a computer screen

Description automatically generated**

**Inventory**

INSERT INTO Inventory (InventoryID, ProductID, Quantityinstock, Laststockupdate)

VALUES

(301, 111, 15, '2024-09-16'),

(302, 112, 25, '2024-09-18'),

(303, 113, 90, '2024-09-20'),

(304, 114, 15, '2024-09-19'),

(305, 115, 80, '2024-09-22'),

(306, 116, 75, '2024-09-19'),

(307, 117, 10, '2024-09-16'),

(308, 118, 50, '2024-09-21'),

(309, 119, 20, '2024-09-21'),

(310, 120, 10, '2024-09-21');

**Inventory OUTPUT: -**

A black screen with green numbers and numbers

Description automatically generated

**Tasks 2**

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

**Ans :** SELECT Firstname, Lastname, Email FROM customers;

A screen shot of a computer

Description automatically generated

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

**Ans** SELECT o.OrderDate, c.Firstname, c.Lastname FROM orders o,Customers c

WHERE o.CustomerID=c.CustomerID;

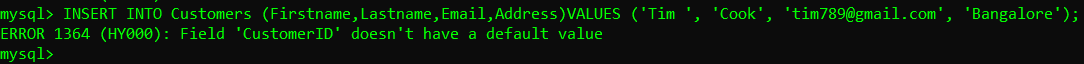
A screen shot of a computer screen

Description automatically generated

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

**Ans:-** Not possible because CustomerID should have a value has it is primary key but

in question CustomerID is not mentioned.



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

**Ans:-** UPDATE Products SET Price\_INR = Price\_INR \* 1.10;

A screenshot of a computer

Description automatically generated

5. 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.

**Ans:-** DELETE from Orderdetails WHERE OrderID=203; DELETE from Order

WHERE OrderID=203;

A screenshot of a computer screen

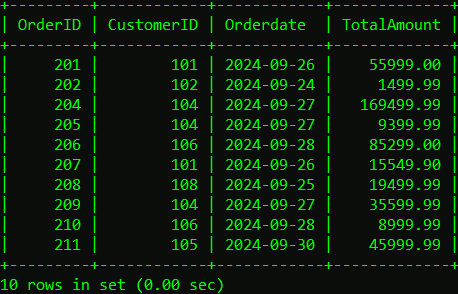
Description automatically generatedA screenshot of a computer screen

Description automatically generated

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.

**Ans:-** INSERT INTO Orders (OrderID,CustomerID,Orderdate,TotalAmount)

VALUES (211, 105, '2024-09-30', 45999.99);



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.

**Ans:-** UPDATE Customers SET email =' hunkkumar@hotmail.com', address =

'99 ,RR nagar,banglore-57' WHERE customerid=109;

A screen shot of a computer

Description automatically generated

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.

**Ans** SELECT od.Orderid, SUM(od.Quantity \* p.price\_INR) AS TotalAmount

FROM Orderdetails od, Products p WHERE od.ProductID = p.ProductID

GROUP BY od.OrderID;

A screenshot of a computer screen

Description automatically generated

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.

**Ans:-** DELETE FROM Orderdetails WHERE OrderID IN( SELECT OrderID FROM

Orders WHERE CustomerID = 103);

DELETE FROM Orders WHERE CustomerID = 103;

A screenshot of a computer screen

Description automatically generated A screenshot of a computer screen

Description automatically generated

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.

**Ans:-** INSERT INTO Products (ProductID, Productname, Description, Price\_INR)

VALUES (121, 'Samsung Galaxy s23', 'Latest model with AI features', 69999.00);

A screenshot of a computer

Description automatically generated

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.

**Ans:-** ALTER TABLE Orders ADD status VARCHAR(20);

UPDATE Orders SET Status = CASE

WHEN customeriID=104 THEN 'Shipped'

ELSE 'Not Shipped'

END;

A black screen with green numbers and numbers

Description automatically generated

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.

**Ans:-** ALTER TABLE Customers ADD Ordercount INT DEFAULT 0;

UPDATE Customers c SET Ordercount=(SELECT COUNT(DISTINCT od.OrderID)

FROM Orderdetails od

JOIN Order o ON od.OrderID=o.OrderID WHERE o.CustomerID=c.CustomerID);

A screenshot of a computer screen

Description automatically generated

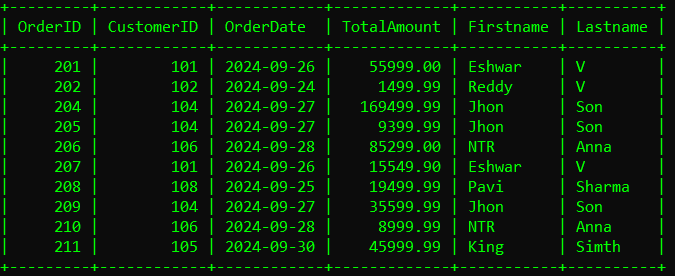
**Tasks 3**

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

**Ans:-** SELECT o.OrderID ,o.CustomerID,o.OrderDate , o.TotalAmount ,c.Firstname

,c.Lastname FROM Orders o INNER JOIN Customers c WHERE

o.customerID=c.customerID;



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

**Ans:-** SELECT p.Productname,SUM(od.Quantity\*p.Price\_INR) AS TotalRevenu

FROM Products p JOIN Orderdetails od WHERE p.ProductID = od.ProductID

GROUP BY Productname;

A screenshot of a computer program

Description automatically generated

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

**Ans**: SELECT DISTINCT c.Firstname,c.Lastname,c.Ph\_no from Customers c

JOIN Orders o WHERE c.CustomerID = o.CustomerID;

A screenshot of a computer screen

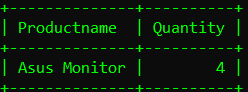
Description automatically generated

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.

**Ans:-** SELECT p.Productname, od.Quantity from Products p

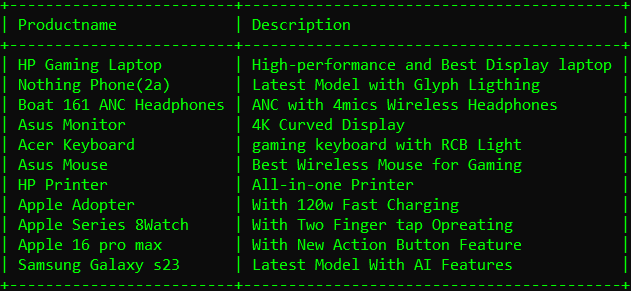
INNER JOIN Orderdetails od WHERE p.productID=od.productID

ORDER BY od.Quantity desc limit 1;



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

Ans :- SELECT Productname, Description from products;



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

**Ans:-** SELECT c.Firstname, c.Lastname, AVG (o.TotalAmount) AS Avg\_order\_value

FROM Orders o INNER JOIN Customers c WHERE o.CustomerID = c.CustomerID

GROUP BY c.Firstname, c.Lastname ORDER BY Avg\_order\_value desc;

A screenshot of a computer screen

Description automatically generated

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

**Ans** SELECT o.OrderID,c.Firstname, c.Lastname, c.Ph\_no, o.TotalAmount AS

TotalRevenue FROM Orders o

JOIN Customers c WHERE o.CustomerID = c.CustomerID

ORDER BY o.TotalAmount DESC LIMIT 1;

A screen shot of a black background with green text

Description automatically generated

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

**Ans:-** SELECT p.Productname ,od.Quantity AS No\_of\_times\_ordered from Products p

INNER JOIN Orderdetails od WHERE p.ProductID=od.ProductID ;

A screen shot of a computer

Description automatically generated

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.

**Ans:-** SELECT c.Firstname, c.Lastname FROM orders o

JOIN Customers c ON o.CustomerID = c.CustomerID

JOIN Orderdetails od ON o.OrderID = od.OrderID

JOIN Products p ON od.ProductID = p.ProductID

WHERE p.Productname = 'HP Printer';

A black screen with green lines

Description automatically generated

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.

**Ans :-** SELECT SUM(o.TotalAmount) Totalrevenu FROM Orders o

WHERE o.Orderdate BETWEEN '2024-09-26' AND '2024-09-30';

A screenshot of a computer screen

Description automatically generated

**Tasks 4**

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

**Ans:-** SELECT c.CustomerID,c.Firstname,c.Lastname FROM Customers c

WHERE c.CustomerID NOT IN (SELECT o.CustomerID FROM Orders o);

A screenshot of a computer screen

Description automatically generated

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

**Ans:-** SELECT SUM(Quantityinstock) TotalProductsAvailable FROM Inventory;

A screen shot of a black background

Description automatically generated

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

**Ans:-** SELECT SUM(totalamount) TotalRevenue FROM Orders;

A screenshot of a computer screen

Description automatically generated

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.

**Ans** SELECT p.Productname, AVG(od.Quantity) Avg\_Quantity\_Order FROM

Orderdetails od INNER JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.Productname;

A screen shot of a computer

Description automatically generated

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.

**Ans** SELECT CustomerID,SUM(TotalAmount) AS Totalrevenue FROM Orders

WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 104)

GROUP BY CustomerID;

A screen shot of numbers and letters

Description automatically generated

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.

**Ans:-** SELECT c.Firstname, c.Lastname, COUNT(o.OrderID) AS Num\_of\_orders

FROM Customers c INNER JOIN Orders o WHERE c.CustomerID = o.CustomerID

GROUP BY c.CustomerID, c.Firstname, c.Lastname

ORDER BY Num\_of\_orders desc;

A black screen with green lines and white text

Description automatically generated

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.

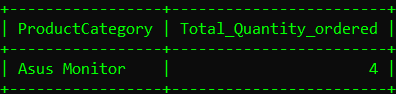
**Ans:-** SELECT p.Productname AS ProductCategory,SUM(od.Quantity)

AS Total\_Quantity\_ordered FROM OrderDetails od

JOIN Products p WHERE od.ProductID = p.ProductID

GROUP BY p.Productname

ORDER BY Total\_Quantity\_ordered desc LIMIT 1;



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.

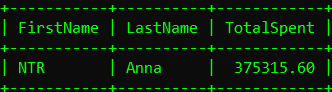
**Ans** SELECT c.Firstname,c.Lastname, (od.Quantity \* p.Price\_INR) AS

Total\_Amt\_Spent FROM Orderdetails od JOIN Products p ON od.ProductID =

p.ProductID JOIN Orders o ON od.OrderID = o.OrderID

JOIN Customers c ON o.CustomerID = c.CustomerID

ORDER BY Total\_Amt\_Spent desc LIMIT 1;



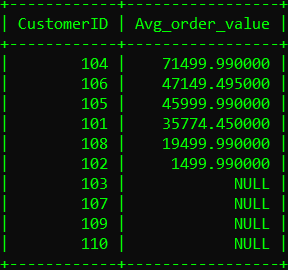
9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.

**Ans** SELECT c.CustomerID, SUM(o.TotalAmount) / COUNT(o.OrderID) AS

Avg\_order\_value FROM Customers c

LEFT JOIN Orders o ON c.CustomerID =o.CustomerID

GROUP BY c.CustomerID ORDER BY Avg\_order\_value desc;



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.

Ans:- SELECT c.Firstname, c.Lastname, (SELECT COUNT(o.OrderID)

FROM Orders o WHERE o.CustomerID = c.CustomerID)

AS Order\_count FROM Customers c ORDER BY Order\_count desc;

A screenshot of a computer screen

Description automatically generated