TechShop-Electronic Gadgets

**Task:1. Database Design:**

1. Create the database named "TechShop.

CREATE DATABASE techshopdb;

USE techshopdb;

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

based on the provided schema.

CREATE TABLE Customers(

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(20) NOT NULL ,

LastName VARCHAR(20) DEFAULT NULL,

Email VARCHAR(65) NOT NULL,

Phone DOUBLE DEFAULT NULL,

Address VARCHAR(65) DEFAULT NULL

);

CREATE TABLE Products(

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

ProductName VARCHAR(25) NOT NULL,

Description VARCHAR(100) NOT NULL,

Price FLOAT NOT NULL

);

CREATE TABLE Orders(

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID int NOT NULL,

OrderDate DATE NOT NULL,

TotalAmount FLOAT DEFAULT 0,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE

);

CREATE TABLE OrderDetails(

OrderDetailID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT NOT NULL,

ProductID INT NOT NULL,

Quantity INT NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

CREATE TABLE Inventory(

InventoryID INT PRIMARY KEY AUTO\_INCREMENT,

ProductID INT NOT NULL UNIQUE,

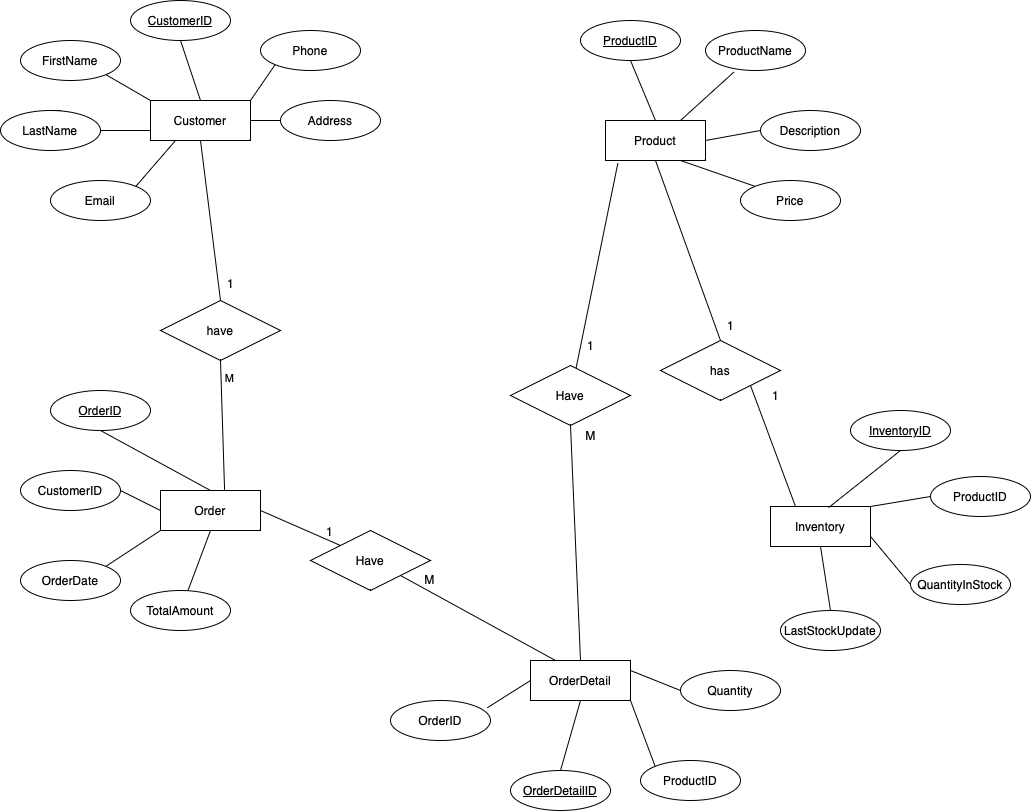
QuantityInStock INT DEFAULT NULL,

LastStockUpdate Date DEFAULT NULL,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

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



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

* Created Foreign Keys and Primary Keys while creating tables.

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

a. Customers

INSERT INTO Customers VALUES

(1,'Sai','Teja','saiteja@abc.com','9392731814','Visakhapatnam'),

(2,'Babu','Rao','babu@abc.com','9390639493','Visakhapatnam'),

(3,'Durga','Prasad','durga@abc.com','6304213633','Vijayawada'),

(4,'Sagar','Mallipudi','sagar@xyz.com','8688881973','Tanuku'),

(5,'Hari','Kiran','hk08@abc.com','9014801838','Vizianagaram'),

(6,'Yash','alamuri','yash10@abc.com','8341145584','East Godavari'),

(7,'Hari','Prakash','hp27@abc.com','8919393084','Hyderabad'),

(8,'Pramod','Peketi','pramod24@abc.com','9704845977','Chennai'),

(9,'Sundar','Miriyala','sundar@xyz.com','9391177179','Banglore'),

(10,'John','Podapati','bilaljohn@xyz.com','9390233211','Onlgole')

;

b. Products

INSERT INTO Products VALUES

(101,'Watch','Fastrack',2000),

(102,'Battery','Li-ion with 3000mAh',1700),

(103,'Temper Glass','11D glass',300),

(104,'Charger','Samsung Charger 33W',1200),

(105,'Mobile','Nothing Phone 2',45000),

(106,'Mobile Case','Transparent Case for Nothing Mobile',500),

(107,'Air pods','Apple Air pods',16000),

(108,'Wired Earphones','Boat Earphones Best Quality',550),

(109,'SmartWatch','Haino Techo watch',4000),

(110,'Laptop','MacBook Pro with M3 Chip',150000),

(111,'Keyboard','Lenovo Keyboard',1000),

(112,'Mouse','Lenovo Mouse',500);

c. Orders

INSERT INTO Orders VALUES

(201,5,'2021-04-07',2000),

(202,8,'2021-05-28',2900),

(203,10,'2022-01-17',16000),

(204,4,'2022-02-14',2000),

(205,6,'2022-05-12',5000),

(206,1,'2023-10-18',150000),

(207,5,'2023-03-24',5000),

(208,4,'2023-08-19',1500),

(209,9,'2023-10-27',6000),

(210,7,'2023-07-23',2500),

(211,3,'2022-05-27',2800),

(212,1,'2023-09-13',7500);

d. OrderDetails

INSERT INTO OrderDetails VALUES

(301,202,103,2),

(302,203,104,4),

(303,204,103,2),

(304,202,101,2),

(305,201,102,3),

(306,205,107,5),

(307,207,107,2),

(308,206,104,3),

(309,208,102,2),

(310,209,105,2);

e. Inventory

INSERT INTO Inventory VALUES

(401,102,10,'2023-10-12'),

(402,103,60,'2023-11-27'),

(403,105,17,'2022-12-04'),

(404,101,20,'2021-10-18'),

(405,106,50,'2023-12-04'),

(406,110,20,'2023-01-23'),

(407,108,20,'2023-06-15'),

(408,104,14,'2023-04-29'),

(409,107,8,'2023-09-26'),

(410,109,20,'2023-10-18');

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

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

A.

SELECT firstname 'Name of Customer', email 'Customer Email'

FROM

Customers;

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

A.

SELECT o.orderid, o.orderdate, c.firstname 'Customer Name'

FROM orders o, customers c

WHERE o.customerid = c.customerid;

3.Write an SQL query to insert a new customer record into the "Customers" table. Include

customer information such as name, email, and address.

A.

INSERT INTO customers(firstname,lastname,email,address) values ('anav','ahb','[avana@cwp.com](mailto:avana@cwp.com)','Pune');

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

increasing them by 10%.

A.

UPDATE products SET price = price\*(110/100);

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.

A.

SET @orderidtodel= 209;

DELETE FROM orderdetails WHERE orderdetailid=@orderidtodel;

DELETE FROM orders WHERE orderid=@orderidtodel;

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.

A.

INSERT INTO orders(customerid,orderdate,totalamount) VALUES (3,'2023-12-07',4500);

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.

A.

SET @emailtoupdate='teja@hexa.com';

SET @addresstoupdate='Tamil Nadu';

SET @custid=1;

UPDATE customers SET email = @emailtoupdate WHERE customerid=@custid;

UPDATE customers SET address = @addresstoupdate WHERE customerid=@custid;

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.

A.

UPDATE orders

SET totalamount = (SELECT SUM(price\*quantity)

FROM products p,orderdetails od

WHERE p.productid = od.productid AND orders.orderid=od.orderid);

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

A.

SET @customid=7;

DELETE FROM orderdetails WHERE orderid IN (SELECT orderid

FROM orders

WHERE customerid=@customid);

DELETE FROM orders WHERE customerid=@customid;

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.

A.

INSERT INTO products(productname,description,price) VALUES

('Display','Laptop Display For HP',9000);

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.

A.

ALTER TABLE orders ADD status VARCHAR(20) DEFAULT NULL AFTER totalamount;

UPDATE orders SET status = 'Pending';

SET @orderidtochange=208;

SET @newstatus='Shipped';

UPDATE orders SET status = @newstatus WHERE orderid=@orderidtochange;

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.

A.

ALTER TABLE customers ADD num\_orders INT DEFAULT NULL AFTER address;

UPDATE customers

SET num\_orders=(SELECT COUNT(orderid)

FROM orders

WHERE customers.customerid=orders.customerid

GROUP BY customerid);

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

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

customer name) for each order.

A.

SELECT o.\*,CONCAT(c.firstname,' ',c.lastname) 'Name' FROM orders o

JOIN

Customers c

ON 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.

A.

SELECT productname,SUM(price\*quantity) as 'totalrevenue' FROM products p

JOIN

orderdetails od

ON p.productid=od.productid

GROUP BY od.productid,productname;

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

names and contact information.

A.

SELECT CONCAT(firstname,' ',lastname) 'Name',Phone FROM customers c

JOIN

(SELECT DISTINCT(customerid) 'custid' FROM orders o) AS o

ON o.custid=c.customerid;

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.

A.

SELECT productname ,sum(quantity) AS 'TotalQuantityOrdered' FROM products p

JOIN

orderdetails od

ON p.productid=od.productid

GROUP BY p.productname

ORDER BY TotalQuantityOrdered DESC

LIMIT 0,1;

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

categories.

A.

ALTER TABLE products ADD category VARCHAR(20) DEFAULT NULL;

UPDATE products SET category = 'Watches' WHERE productname='Watch';

UPDATE products SET category = 'MobileAccessories' WHERE productid BETWEEN 102 AND 106;

UPDATE products SET category = 'Bluetoothdevices' WHERE productid BETWEEN 107 AND 109;

UPDATE products SET category = 'LaptopAccessories' WHERE productid > 109;

SELECT productname,category 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.

A.

SELECT CONCAT(firstname,' ',lastname) 'Customer Name' , AVG(totalamount) 'average\_order\_value'

FROM customers c

JOIN

orders o

ON o.customerid=c.customerid

GROUP BY c.customerid;

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

customer information, and the total revenue.

A.

SELECT orderid,firstname 'Name',phone,address, totalamount from orders o

JOIN

customers c

ON o.customerid = c.customerid

ORDER BY totalamount DESC LIMIT 1;

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

ordered.

A.

SELECT productname,COUNT(od.productid) 'numtimes\_ordered'

FROm products p

JOIN

orderdetails od

ON p.productid=od.productid

GROUP BY od.productid;

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.

SET @prodtname = 'Battery';

SELECT DISTINCT(customername) FROM

(SELECT CONCAT(firstname,' ',lastname) 'Customername',orderid FROM customers c,orders o

WHERE c.customerid=o.customerid) AS cn

JOIN

(SELECT od.orderid,od.productid,productname FROM orderdetails od,products p

WHERE od.productid=p.productid) AS pn

ON cn.orderid=pn.orderid

WHERE productname=@prodtname;

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.

SET @startdate = '2021-03-18';

SET @enddate = '2021-06-18';

SELECT sum(totalamount)

FROM orders

WHERE orders.orderdate BETWEEN @startdate AND @enddate;

**Task 4. Subquery and its type:**

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

Ans.

SELECT CONCAT(firstname,' ',lastname) 'Customername' FROM customers

WHERE customerid NOT IN (SELECT customerid FROM orders);

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

Ans.

/\* Creating a temporary with updated quantity of stock\*/

CREATE TEMPORARY TABLE tempquantity AS

SELECT

l.productid, (MIN(qis) - SUM(quantity)) 'qanis'

FROM

(SELECT

productid, quantityinstock 'qis', laststockupdate 'lsu'

FROM

inventory) AS l,

(SELECT

o.orderid, orderdate, productid, quantity

FROM

orders o

JOIN orderdetails od ON od.orderid = o.orderid) AS tb

WHERE

l.lsu < tb.orderdate

AND l.productid = tb.productid

GROUP BY l.productid;

/\*Updating New lasteststock values\*/

UPDATE inventory i

JOIN

tempquantity tq

ON i.productid=tq.productid

SET quantityinstock=qanis;

/\*Dropping Temporary table\*/

DROP TABLE tempquantity;

/\*Updating date in Inventory table (laststockupdate) with today date\*/

UPDATE inventory SET laststockupdate = '2023-12-11';

/\*Total quantity of available products for sale\*/

SELECT SUM(quantityinstock) 'Available\_products'

FROM inventory

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

Ans.

SELECT SUM(totalamount) 'totalrevenue' FROM orders

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.

SET @categry = 'MobileAccessories';

SELECT AVG(totalquantity) 'average\_quantity' FROM

(SELECT p.productname,od.productid,SUM(quantity) totalquantity

FROM products p

JOIN

orderdetails od

ON od.productid = p.productid

WHERE category=@categry

GROUP BY od.productid) AS s

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.

SET @custid=8;

SELECT Customername,totalrevenue FROM

(SELECT o.customerid 'custmrid',CONCAT(firstname,' ',lastname) 'Customername',SUM(totalamount) 'totalrevenue'

FROM orders o

JOIN

customers c

ON o.customerid = c.customerid

GROUP BY o.customerid) AS t

WHERE custmrid = @custid;

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 firstname,cnt FROM customers

JOIN

(SELECT customerid,COUNT(orderid) 'cnt' FROM orders GROUP BY orders.customerid) AS t

ON t.customerid=customers.customerid

WHERE cnt = (SELECT MAX(cnt) FROM (SELECT COUNT(orderid) 'cnt' FROM orders GROUP BY orders.customerid) AS c)

GROUP BY t.customerid;

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 category,sum(quantity) 'Totalquantityordered' FROM orderdetails od

JOIN

products p

ON p.productid=od.productid

GROUP BY p.category

ORDER BY Totalquantityordered 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 CONCAT(firstname,' ',lastname) 'Customername', SUM(totalamount) 'totalspending' FROM orders o

JOIN customers c

ON o.customerid=c.customerid

GROUP BY o.customerid

ORDER BY totalspending 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 SUM(totalamount)/COUNT(orderid) FROM orders;

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 CONCAT(firstname,' ',lastname) 'Customername',COUNT(orderid) 'totalorders' FROM orders o,customers c

WHERE o.customerid = c.customerid

GROUP BY o.customerid