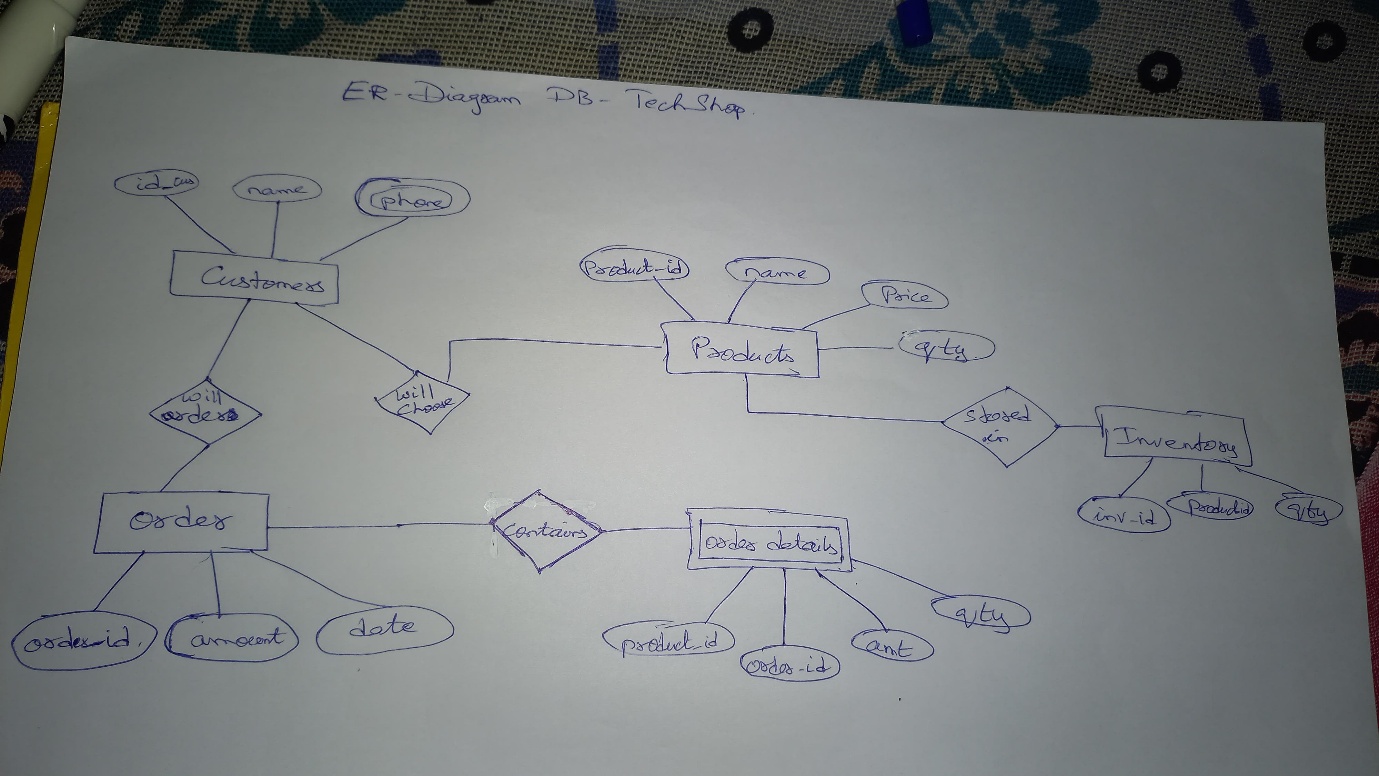
**ELECTRONIC GADGETS**

TASK 1

--Create database

CREATE DATABASE TechShop;

ER Diagram



-- Create the customers table

CREATE TABLE customers (

customerid INT PRIMARY KEY,

firstname VARCHAR(50),

lastname VARCHAR(50),

email VARCHAR(100),

phone VARCHAR(20),

address VARCHAR(255)

);

-- Insert values into the customers table

INSERT INTO customers (customerid, firstname, lastname, email, phone, address) VALUES

(2, 'Jane', 'Smith', 'janesmith@example.com', '9876543210', '456 Elm St'),

(3, 'Michael', 'Johnson', 'michaelj@example.com', '5551234567', '789 Pine St'),

(4, 'Emily', 'Brown', 'emilyb@example.com', '4449876543', '321 Oak St'),

(5, 'Daniel', 'Wilson', 'danielw@example.com', '7775554321', '654 Birch St'),

(6, 'Sarah', 'Taylor', 'saraht@example.com', '9991112233', '987 Cedar St'),

(7, 'David', 'Anderson', 'davidand@example.com', '3332221111', '741 Maple St'),

(8, 'Laura', 'Martinez', 'lauram@example.com', '6667778888', '852 Walnut St'),

(9, 'James', 'Hernandez', 'jamesh@example.com', '1112223333', '963 Cherry St'),

(10, 'Olivia', 'Clark', 'oliviac@example.com', '4445556666', '159 Ash St'),

(21, 'Helen', 'Keller', 'helen@gmail.com', '75555677', '324 Oakland'),

(1, 'John', 'Doe', 'johny@example.com', '2134567890', '123 Main St');

-- Create the products table

CREATE TABLE products (

productid INT PRIMARY KEY,

productname VARCHAR(100),

description TEXT,

price INT,

category VARCHAR(50)

);

-- Insert values into the products table

INSERT INTO products (productid, productname, description, price, category) VALUES

(11, 'USB Flash Drive', '64GB USB flash drive', 275, 'Storage'),

(1, 'Laptop', 'High-performance laptop', 8800, 'Laptops'),

(2, 'Smartphone', 'Latest model smartphone', 6600, 'Smartphones'),

(3, 'Tablet', 'Lightweight tablet with high resolution', 4400, 'Tablets'),

(4, 'Smartwatch', 'Feature-rich smartwatch', 2200, 'Wearables'),

(5, 'Wireless Headphones', 'Noise-canceling headphones', 1650, 'Audio'),

(6, 'Bluetooth Speaker', 'Portable Bluetooth speaker', 1100, 'Audio'),

(7, 'Monitor', '27-inch HD monitor', 2750, 'Monitors'),

(8, 'Mechanical Keyboard', 'RGB mechanical keyboard', 880, 'Accessories'),

(9, 'Gaming Mouse', 'Ergonomic gaming mouse', 550, 'Accessories'),

(10, 'External Hard Drive', '1TB external hard drive', 1320, 'Storage');

-- Create the orders table

CREATE TABLE orders (

orderid INT PRIMARY KEY,

customerid INT,

orderdate DATE,

totalamount INT,

status VARCHAR(20),

FOREIGN KEY (customerid) REFERENCES customers(customerid)

);

-- Insert values into the orders table

INSERT INTO orders (orderid, customerid, orderdate, totalamount, status) VALUES

(4, 4, '2025-03-04', 2200, 'Completed'),

(5, 5, '2025-03-05', 1650, 'Completed'),

(6, 6, '2025-03-06', 1100, 'Dispatched'),

(7, 7, '2025-03-07', 2750, 'Dispatched'),

(8, 8, '2025-03-08', 880, 'Cancelled'),

(9, 9, '2025-03-09', 550, 'Cancelled'),

(10, 10, '2025-03-10', 1320, 'Cancelled'),

(2, 2, '2025-03-02', 6600, 'Shipped'),

(3, 3, '2025-03-03', 4400, 'Shipped'),

(21, 21, '2025-04-01', NULL, 'Pending');

-- Create the orderdetails table

CREATE TABLE orderdetails (

orderdetailid INT PRIMARY KEY,

orderid INT,

productid INT,

quantity INT,

FOREIGN KEY (orderid) REFERENCES orders(orderid),

FOREIGN KEY (productid) REFERENCES products(productid)

);

-- Insert values into the orderdetails table

INSERT INTO orderdetails (orderdetailid, orderid, productid, quantity) VALUES

(2, 2, 2, 1),

(3, 3, 3, 1),

(4, 4, 4, 1),

(5, 5, 5, 1),

(6, 6, 6, 1),

(7, 7, 7, 1),

(8, 8, 8, 1),

(9, 9, 9, 1),

(10, 10, 10, 1);

-- Create the inventory table

CREATE TABLE inventory (

inventoryid INT PRIMARY KEY,

productid INT,

quantityinstock INT,

laststockupdate DATE,

FOREIGN KEY (productid) REFERENCES products(productid)

);

-- Insert values into the inventory table

INSERT INTO inventory (inventoryid, productid, quantityinstock, laststockupdate) VALUES

(1, 1, 50, '2025-02-28'),

(2, 2, 30, '2025-02-27'),

(3, 3, 40, '2025-02-26'),

(4, 4, 20, '2025-02-25'),

(5, 5, 60, '2025-02-24'),

(6, 6, 70, '2025-02-23'),

(7, 7, 15, '2025-02-22'),

(8, 8, 25, '2025-02-21'),

(9, 9, 35, '2025-02-20'),

(10, 10, 45, '2025-02-19');

TASK 2

-- 1. Retrieve names and emails of all customers

select concat(firstname, ' ', lastname) as name, email from customers;

-- 2. List all orders with order dates and corresponding customer names

select o.orderid, o.orderdate, c.firstname, c.lastname

from orders o

join customers c on o.customerid = c.customerid;

-- 3. Insert a new customer into the "Customers" table

insert into customers (customerid, firstname, lastname, email, phone, address)

values (21, 'Helen', 'Keller', 'helen@gmail.com', '75555677', '324 Oakland');

-- 4. Increase all product prices by 10%

update products set price = price + (price \* 0.10);

-- 5. Delete a specific order and its associated order details

delete from orderdetails where orderid = ?;

delete from orders where orderid = ?;

-- 6. Insert a new order into the "Orders" table

insert into orders (orderid, customerid, orderdate, totalamount)

values (21, 21, '2025-04-01', 450);

-- 7. Update customer contact information

update customers set email='johny@example.com', phone='2134567890' where customerid=1;

-- 8. Recalculate and update total cost of each order

update orders o

set totalamount = (

select sum(od.quantity \* p.price)

from orderdetails od

join products p on od.productid = p.productid

where od.orderid = o.orderid

);

-- 9. Delete all orders and associated order details for a specific customer

delete from orderdetails where orderid in (select orderid from orders where customerid = ?);

delete from orders where customerid = ?;

-- 10. Insert a new electronic gadget into the "Products" table

insert into products (productid, productname, description, price, category)

values (11, 'USB Flash Drive', '64GB USB flash drive', 275, 'Storage');

-- 11. Update order status from "Pending" to "Shipped"

update orders set status='Shipped' where status='Pending';

-- 12. Update customer order count based on the number of orders

update customers c

set order\_count = coalesce((

select count(o.orderid)

from orders o

where o.customerid = c.customerid

), 0);

TASK 3

-- 1. Retrieve a list of all orders along with customer information

select o.orderid, c.customerid, c.firstname, c.lastname, c.email, o.orderdate, o.totalamount

from orders o

join customers c on o.customerid = c.customerid;

-- 2. Find the total revenue generated by each electronic gadget product

select p.productname, sum(od.quantity \* p.price) as total\_revenue

from orderdetails od

join products p on od.productid = p.productid

group by p.productname;

-- 3. List all customers who have made at least one purchase

select distinct c.customerid, c.firstname, c.lastname, c.email, c.phone, c.address

from customers c

join orders o on c.customerid = o.customerid;

-- 4. Find the most popular electronic gadget (highest total quantity ordered)

select p.productname, sum(od.quantity) as total\_quantity\_ordered

from orderdetails od

join products p on od.productid = p.productid

group by p.productname

order by total\_quantity\_ordered desc

limit 1;

-- 5. Retrieve a list of electronic gadgets along with their corresponding categories

select productname, category from products;

-- 6. Calculate the average order value for each customer

select concat(c.firstname, ' ', c.lastname) as name, avg(o.totalamount) as average\_order\_value

from customers c

join orders o on c.customerid = o.customerid

group by c.customerid, c.firstname, c.lastname;

-- 7. Find the order with the highest total revenue

select o.orderid, c.customerid, c.firstname, c.lastname, c.email, o.totalamount as total\_revenue

from orders o

join customers c on o.customerid = c.customerid

order by o.totalamount desc

limit 1;

-- 8. List electronic gadgets and the number of times each product has been ordered

select p.productname, count(od.productid) as order\_count

from products p

join orderdetails od on p.productid = od.productid

group by p.productid, p.productname;

-- 9. Find customers who have purchased a specific electronic gadget product

select distinct c.customerid, c.firstname, c.lastname, c.email

from customers c

join orders o on c.customerid = o.customerid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

where p.productname = 'Wireless Charger';

-- 10. Calculate the total revenue generated by all orders placed within a specific time period

select sum(totalamount) as total\_revenue

from orders

where orderdate between '2025-03-15' and '2025-03-17';

TASK 4

-- 1. Find customers who have not placed any orders

select c.customerid, c.firstname, c.lastname, c.email

from customers c

left join orders o on c.customerid = o.customerid

where o.orderid is null;

-- 2. Find the total number of products available for sale

select sum(quantityinstock) as total\_products\_available from inventory;

-- 3. Calculate the total revenue generated by TechShop

select sum(totalamount) as total\_revenue from orders;

-- 4. Calculate the average quantity ordered for products in a specific category

select avg(od.quantity) as average\_quantity

from orderdetails od

join products p on od.productid = p.productid

where p.category = 'Components';

-- 5. Calculate the total revenue generated by a specific customer

select sum(o.totalamount) as total\_revenue

from orders o

join customers c on o.customerid = c.customerid

where c.customerid = 20;

-- 6. Find the customers who have placed the most orders

select c.firstname || ' ' || c.lastname as name, count(o.orderid) as order\_count

from customers c

join orders o on c.customerid = o.customerid

group by c.customerid, c.firstname, c.lastname

order by order\_count desc

limit 1;

-- 7. Find the most popular product category (highest total quantity ordered)

select p.category, sum(od.quantity) as total\_quantity\_ordered

from orderdetails od

join products p on od.productid = p.productid

group by p.category

order by total\_quantity\_ordered desc

limit 1;

-- 8. Find the customer who has spent the most money on electronic gadgets

select c.customerid, c.firstname, c.lastname, sum(p.price \* od.quantity) as total\_spent

from customers c

join orders o on c.customerid = o.customerid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

group by c.customerid, c.firstname, c.lastname

order by total\_spent desc

limit 1;

-- 9. Calculate the average order value (total revenue divided by number of orders)

select avg(o.totalamount) as average\_order\_value

from orders o;

-- 10. Find the total number of orders placed by each customer

select c.customerid, c.firstname, c.lastname, count(o.orderid) as order\_count

from customers c

join orders o on c.customerid = o.customerid

group by c.customerid, c.firstname, c.lastname

order by order\_count desc;