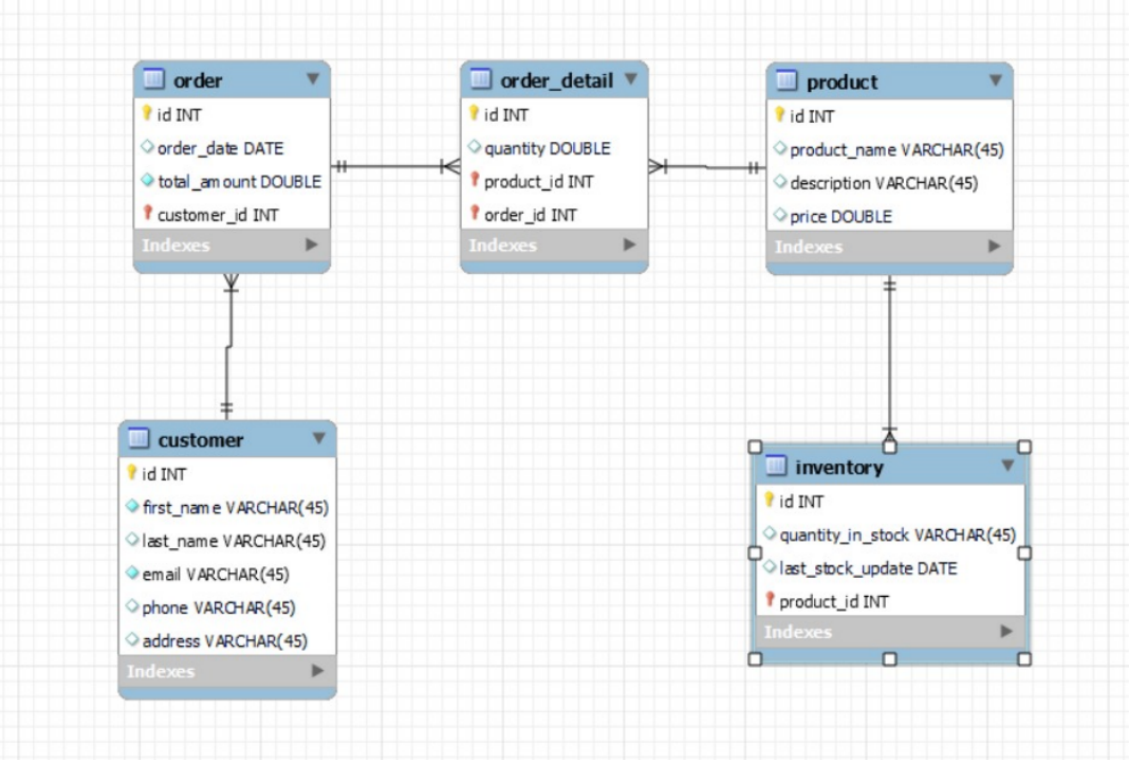
Project- Electronic Gadgets

ER Diagram:



Code:

```
-- Name of my Created Database
use learnings;

-- Create Customers table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(255)
);

-- Create Products table
```

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10, 2)
);

-- Create Orders table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- Create OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

-- Create Inventory table
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY AUTO_INCREMENT,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

-- Sample entries for Customers table
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address) VALUES
    ('Arjun', 'Kumar', 'arjun@example.com', '123-456-7890', '123 Main St, Bangalore'),
    ('Aishwarya', 'Naidu', 'aishwarya@example.com', '456-789-0123', '456 Park Ave, Chennai'),
    ('Ganesh', 'Menon', 'ganesh@example.com', '789-012-3456', '789 Elm St, Hyderabad'),
    ('Divya', 'Sharma', 'divya@example.com', '987-654-3210', '987 Maple Ave, Coimbatore'),
    ('Krishna', 'Iyer', 'krishna@example.com', '654-321-0987', '654 Oak St, Mysore'),
    ('Meera', 'Raj', 'meera@example.com', '321-098-7654', '321 Pine St, Kochi'),
    ('Neha', 'Pillai', 'neha@example.com', '210-987-6543', '210 Cedar St, Trivandrum'),
    ('Rajesh', 'Kumar', 'rajesh@example.com', '543-210-9876', '543 Birch St, Mangalore'),
    ('Saravanan', 'Rao', 'saravanan@example.com', '876-543-2109', '876 Walnut St,
Pondicherry'),
```

```
    ('Shalini', 'Menon', 'shalini@example.com', '234-567-8901', '234 Pineapple St, Vijayawada');

-- Sample entries for Products table
INSERT INTO Products (ProductName, Description, Price) VALUES
    ('Smartphone', '5.5-inch display, 64GB storage, 12MP camera', 499.99),
    ('Laptop', '15.6-inch screen, 8GB RAM, 512GB SSD', 899.99),
    ('Tablet', '10-inch display, 128GB storage, quad-core processor', 349.99),
    ('Smartwatch', 'Water-resistant, heart rate monitor, fitness tracker', 199.99),
    ('Headphones', 'Wireless, noise-canceling, over-ear design', 149.99),
    ('Bluetooth Speaker', 'Portable, 360-degree sound, built-in microphone', 79.99),
    ('Digital Camera', '20MP sensor, 4K video recording, 3-inch LCD screen', 449.99),
    ('Gaming Console', '4K gaming, 1TB storage, HDR support', 399.99),
    ('Wireless Earbuds', 'True wireless, sweat-resistant, up to 8 hours battery life', 129.99),
    ('Fitness Tracker', 'Step counter, sleep monitor, heart rate sensor', 79.99);

-- Sample entries for Orders table
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
    (1, '2024-03-01', 699.98),
    (2, '2024-03-02', 849.99),
    (3, '2024-03-03', 449.99),
    (4, '2024-03-04', 599.98),
    (5, '2024-03-05', 249.99),
    (6, '2024-03-06', 149.99),
    (7, '2024-03-07', 129.99),
    (8, '2024-03-08', 399.99),
    (9, '2024-03-09', 79.99),
    (10, '2024-03-10', 299.99);

-- Sample entries for OrderDetails table
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
    (1, 1, 1),
    (2, 2, 1),
    (3, 3, 1),
    (4, 4, 1),
    (5, 5, 1),
    (6, 6, 1),
    (7, 7, 1),
    (8, 8, 1),
    (9, 9, 1),
    (10, 10, 1);

-- Sample entries for Inventory table
INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate) VALUES
    (1, 100, '2024-03-01'),
    (2, 50, '2024-03-02'),
    (3, 75, '2024-03-03'),
    (4, 200, '2024-03-04'),
```

```
   (5, 150, '2024-03-05'),
   (6, 100, '2024-03-06'),
   (7, 120, '2024-03-07'),
   (8, 80, '2024-03-08'),
   (9, 90, '2024-03-09'),
   (10, 110, '2024-03-10');


-- Working on select, where, betwen, and, like commands

-- Retrieving the names and emails of all customers
SELECT FirstName, LastName, Email
FROM Customers;

-- List all orders with their order dates and corresponding customer names
SELECT o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID;

-- Insert a new customer record into the "Customers" table
INSERT INTO Customers (FirstName, LastName, Email, Address)
VALUES ('John', 'Doe', 'john.doe@example.com', '123 New Street, City, Country');

-- Recalculate and update the total cost of each order in the "Orders" table
UPDATE Orders AS o
SET TotalAmount = (
   SELECT SUM(p.Price * od.Quantity)
   FROM OrderDetails AS od
   JOIN Products AS p ON od.ProductID = p.ProductID
   WHERE od.OrderID = o.OrderID
);


-- Insert a new electronic gadget product into the "Products" table:
INSERT INTO Products (ProductName, Category, Price)
VALUES ('New Gadget', 'Electronic', 199.99);


-- Calculate and update the number of orders placed by each customer in the "Customers"
table
UPDATE Customers AS c
SET TotalOrders = (
   SELECT COUNT(*)
   FROM Orders AS o
   WHERE o.CustomerID = c.CustomerID
);
```

-- Using Order by, group by, having and joins

-- List all orders with their order dates and corresponding customer names
SELECT o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID;

-- total revenue generated by each product
SELECT p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
FROM OrderDetails od
INNER JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName;

-- List the first name, last name, email, and phone number of customers who have placed orders, with each customer listed only once
SELECT c.FirstName, c.LastName, c.Email, c.Phone
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;

-- What is the product with the highest total quantity ordered
SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
FROM OrderDetails od
INNER JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductID
ORDER BY TotalQuantityOrdered DESC
LIMIT 1;

-- What is the average order value for each customer
SELECT CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, AVG(o.TotalAmount) AS AvgOrderValue
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;

-- Which order has the highest total revenue along with the customer name associated with each order
SELECT o.OrderID, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, MAX(o.TotalAmount) AS TotalRevenue
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY o.OrderID;

-- How many times each product has been ordered
SELECT p.ProductName, COUNT(od.OrderDetailID) AS OrderCount

```
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID;

-- What is the total revenue generated by all orders placed within the time period from March 1, 2024, to March 6, 2024
SELECT SUM(o.TotalAmount) AS TotalRevenue
FROM Orders o
WHERE o.OrderDate BETWEEN '2024-03-01' AND '2024-03-06';


-- Sub Queries

-- List the first name, last name, and email of customers who have not placed any orders.
SELECT FirstName, LastName, Email
FROM Customers c
WHERE NOT EXISTS (
    SELECT 1
    FROM Orders o
    WHERE o.CustomerID = c.CustomerID
);

-- What is the total number of products available for sale
SELECT COUNT(*) AS TotalProducts
FROM Products;

-- What is the total revenue generated by all orders
SELECT (
    SELECT SUM(TotalAmount)
    FROM Orders
) AS TotalRevenue;

-- Which customer has placed the most orders, and how many orders have they placed
SELECT FirstName, LastName, OrderCount
FROM (
    SELECT c.FirstName, c.LastName, COUNT(*) AS OrderCount
    FROM Customers c
    JOIN Orders o ON c.CustomerID = o.CustomerID
    GROUP BY c.CustomerID
    ORDER BY OrderCount DESC
) AS CustomerOrders
LIMIT 1;

-- What is the average order value across all orders
SELECT (
    SELECT AVG(TotalAmount)
    FROM Orders
```

```
) AS AvgOrderValue;

-- What is the total number of orders placed by each customer, along with their first name and last name
SELECT c.FirstName, c.LastName, OrderCount
FROM Customers c
JOIN (
   SELECT CustomerID, COUNT(*) AS OrderCount
   FROM Orders
   GROUP BY CustomerID
) AS CustomerOrders ON c.CustomerID = CustomerOrders.CustomerID;
```