**–Create a DataBase**

CREATE DATABASE ecommerce;

**–Switch to the created database**

USE ecommerce;

**Create the customers Table**

CREATE TABLE customers(

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(20) NOT NULL,

email VARCHAR(20) UNIQUE NOT NULL,

address VARCHAR(255) NOT NULL

);

**Create the orders Table with a foreign key as customer id**

CREATE TABLE orders(

id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT NOT NULL,

order\_date DATE NOT NULL,

total\_amount DECIMAL(10,2) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customer(id)

);

**Create the products table**

CREATE TABLE products(

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

price DECIMAL(10,2) NOT NULL,

description text

);

**Insert sample data to customers**

INSERT INTO customers(name,email,address)

VALUES

('Ligin George', 'ligin@gmail.com', 'C1, Gandhi Street'),

('Dhibahar', 'dhiabhar@gmail.com', 'D1 Park Street'),

('Satheesh', 'satheesh@gmail.com', '789 Pine Road');

**Insert sample data to products**

INSERT INTO products(name,price,description)

VALUES

('Product A', 20.00, 'Description of Product A'),

('Product B', 35.00, 'Description of Product B'),

('Product C', 50.00, 'Description of Product C'),

('Product D', 15.00, 'Description of Product D');

**Insert sample data to orders**

INSERT INTO orders(customer\_id,order\_date,total\_amount)

VALUES

(1,CURDATE(),75.00),

(2,CURDATE() - INTERVAL 10 DAY, 45.00),

(3,CURDATE() - INTERVAL 35 DAY, 150.00)

(1,CURDATE() - INTERVAL 40 DAY, 30.00);

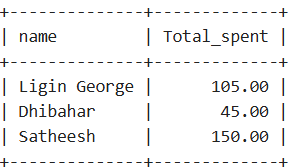
**Fetch customers who placed an order in the last 30 days**

SELECT c.\* FROM customers c JOIN orders o ON c.id = o.customer\_id WHERE o.order\_date>= CURDATE() - INTERVAL 30 DAY;



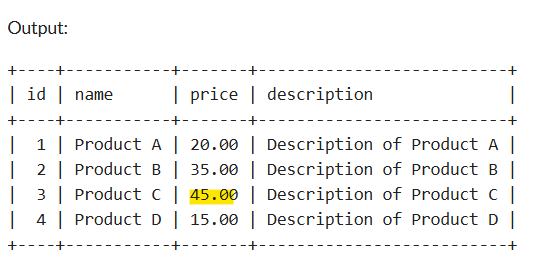
**Calculate total order amount for each customer**

SELECT c.name, SUM(o.total\_amount) AS Total\_spent FROM customers c JOIN orders o ON c.id=o.customer\_id GROUP BY c.name;



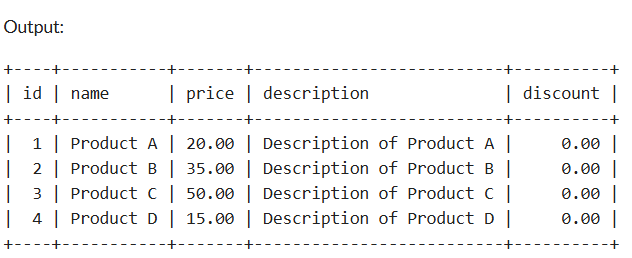
**Update the price of Product C - Initially it was 50.00**

UPDATE products SET price=45.00 WHERE name='Product C';



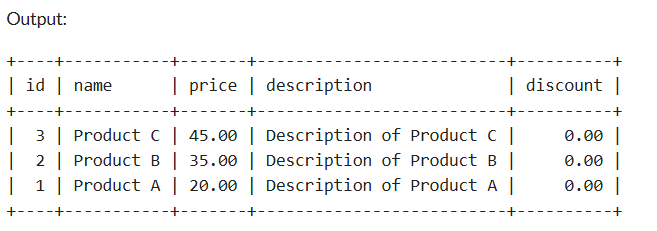
**Add a discount column to the products table - defaulting all the values to zero**

ALTER TABLE products ADD COLUMN discount DECIMAL(10,2) DEFAULT 0.00;



**Get the top 3 most expensive products**

SELECT \* FROM products ORDER BY price DESC LIMIT 3;

****

**Add Normalization step to include an order\_items table**

CREATE TABLE order\_items(

id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT NOT NULL,

product\_id INT NOT NULL,

quantity INT NOT NULL,

FOREIGN KEY (order\_id) REFERENCES orders(id),

FOREIGN KEY (product\_id) REFERENCES product(id)

);

**Inserting sample data to order\_items**

INSERT INTO order\_items(order\_id,product\_id,quantity)

VALUES

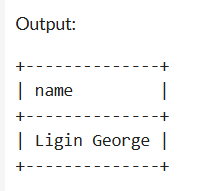
(1,1,2),

(2,2,1),

(3,3,3);

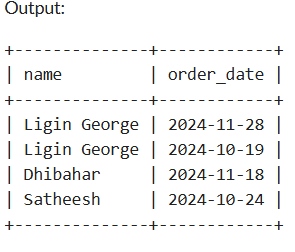
**Fetch the customer names who ordered Product A**

SELECT c.name FROM customers c JOIN orders o ON c.id=o.customer\_id JOIN order\_items oi on o.id = oi.order\_id JOIN products p ON p.id = oi.product\_id WHERE p.name='Product A';



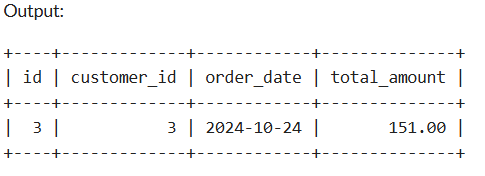
**Fetch customer names and their order dates**

SELECT c.name,o.order\_date FROM customers c JOIN orders o ON c.id = o.customer\_id



**Fetch orders with a total amount exceeding 150.00**

SELECT \* FROM orders WHERE total\_amount > 150;



**Calculate the average total amount of all orders**

SELECT AVG(total\_amount) AS average\_order\_total from orders;

