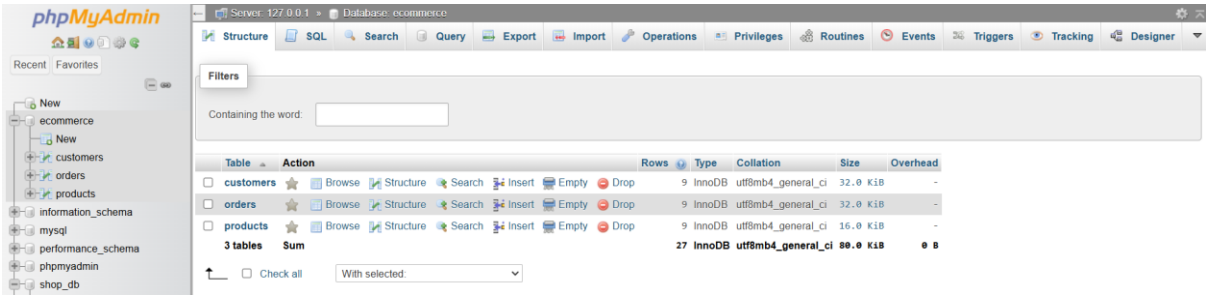
Database name – Ecommerce -> table name : Customers,orders,products



1. Retrieve all customers who have placed an order in the last 30 days.

```
SELECT name, email, address

FROM customers

WHERE id IN (

   SELECT customer_id

   FROM orders

   WHERE order_date >= CURDATE() - INTERVAL 30 DAY

);
```

	name	email	address
Edit Copy Delete	Rajesh Kumar	rajesh.kumar@example.com	Chennai, Tamil Nadu
Edit Copy Delete	Anitha Reddy	anitha.reddy@example.com	Coimbatore, Tamil Nadu
Edit Copy Delete	Vikram Sharma	vikram.sharma@example.com	Madurai, Tamil Nadu
Edit Copy Delete	Priya Subramanian	priya.subramanian@example.com	Tiruchirappalli, Tamil Nadu
Edit Copy Delete	Arun Prakash	arun.prakash@example.com	Salem, Tamil Nadu
Edit Copy Delete	Deepa Ramachandran	deepa.ramachandran@example.com	Erode, Tamil Nadu
Edit Copy Delete	Karthik Natarajan	karthik.natarajan@example.com	Vellore, Tamil Nadu
Edit Copy Delete	Lakshmi Narayan	lakshmi.narayan@example.com	Tirunelveli, Tamil Nadu
Edit Copy Delete	Suresh Babu	suresh.babu@example.com	Thanjavur, Tamil Nadu

2. Get the total amount of all orders placed by each customer.

```
SELECT customer_id, SUM(total_amount) AS total_amount

FROM orders

GROUP BY customer_id;
```

	customer_id	total_amount
Edit Copy Delete	1	15000.00
Edit Copy Delete	2	45000.00
Edit Copy Delete	3	20000.00
Edit Copy Delete	4	30000.00
Edit Copy Delete	5	12000.00
Edit Copy Delete	6	8000.00
Edit Copy Delete	7	40000.00
Edit Copy Delete	8	35000.00
Edit Copy Delete	9	10000.00

3. Update the price of Product C to 45.00.

(Product C in my table Washing Machine)

UPDATE products SET price = 45.00 WHERE name = 'Washing Machine;

	id	name	price	description
Edit Copy Delete	1	Mobile Phone	15000.00	A high-performance smartphone with 128GB storage.
Edit Copy Delete	2	Laptop	45000.00	A lightweight laptop with 8GB RAM and 256GB SSD.
Edit Copy Delete	3	Washing Machine	45.00	A fully automatic washing machine with 6kg capacit...
Edit Copy Delete	4	Refrigerator	30000.00	A double-door refrigerator with 300L capacity.
Edit Copy Delete	5	Microwave Oven	12000.00	A convection microwave oven with smart features.
Edit Copy Delete	6	Smartwatch	8000.00	A smartwatch with fitness tracking and notificatio...
Edit Copy Delete	7	Television	40000.00	A 50-inch 4K Ultra HD Smart TV.
Edit Copy Delete	8	Air Conditioner	35000.00	A 1.5-ton split AC with inverter technology.
Edit Copy Delete	9	Vacuum Cleaner	10000.00	A high-power vacuum cleaner with HEPA filter.

4. Add a new column discount to the products table

ALTER TABLE products

ADD discount DECIMAL(5, 2);

	id	name	price	description	discount
Edit Copy Delete	1	Mobile Phone	15000.00	A high-performance smartphone with 128GB storage.	NULL
Edit Copy Delete	2	Laptop	45000.00	A lightweight laptop with 8GB RAM and 256GB SSD.	NULL
Edit Copy Delete	3	Washing Machine	45.00	A fully automatic washing machine with 6kg capacit...	NULL
Edit Copy Delete	4	Refrigerator	30000.00	A double-door refrigerator with 300L capacity.	NULL
Edit Copy Delete	5	Microwave Oven	12000.00	A convection microwave oven with smart features.	NULL
Edit Copy Delete	6	Smartwatch	8000.00	A smartwatch with fitness tracking and notificatio...	NULL
Edit Copy Delete	7	Television	40000.00	A 50-inch 4K Ultra HD Smart TV.	NULL
Edit Copy Delete	8	Air Conditioner	35000.00	A 1.5-ton split AC with inverter technology.	NULL
Edit Copy Delete	9	Vacuum Cleaner	10000.00	A high-power vacuum cleaner with HEPA filter.	NULL

5. Retrieve the top 3 products with the highest price.

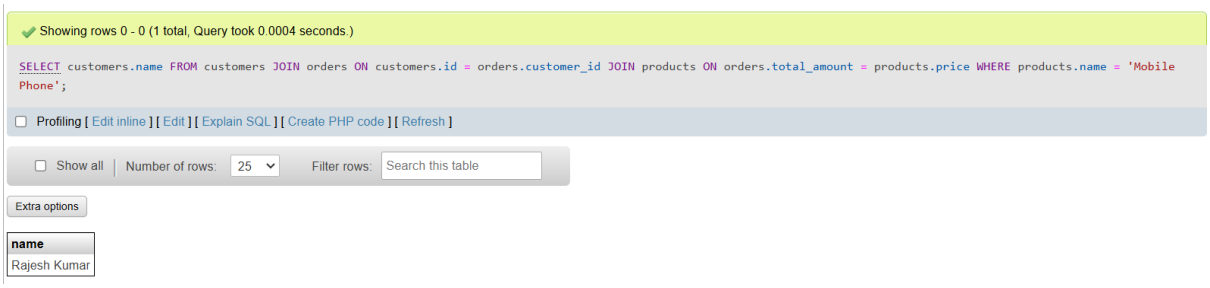
SELECT name, price FROM products ORDER BY price DESC LIMIT 3;

	name	price 1
Edit Copy Delete	Laptop	45000.00
Edit Copy Delete	Television	40000.00
Edit Copy Delete	Air Conditioner	35000.00

6. Get the names of customers who have ordered Product A.

(in my table product A is Mobile phone)

SELECT customers.name FROM customers JOIN orders ON customers.id = orders.customer_id JOIN products ON orders.total_amount = products.price WHERE products.name = 'Mobile Phone';



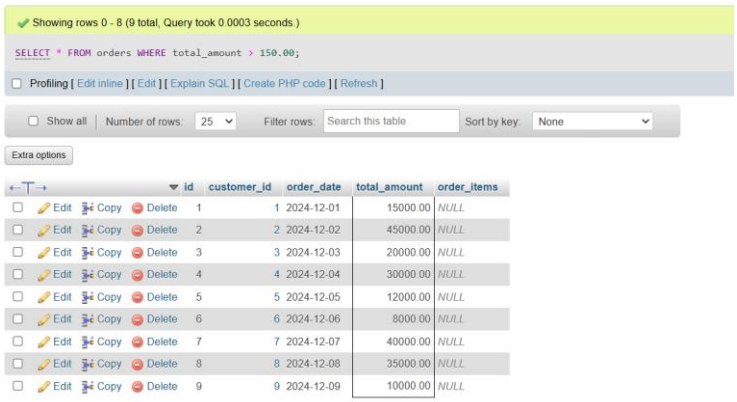
7. Join the orders and customers tables to retrieve the customer's name and order date for each order.

SELECT c.name, o.order_date FROM customers c JOIN orders o ON c.id = o.customer_id;

name	order_date
Rajesh Kumar	2024-12-01
Anitha Reddy	2024-12-02
Vikram Sharma	2024-12-03
Priya Subramanian	2024-12-04
Arun Prakash	2024-12-05
Deepa Ramachandran	2024-12-06
Karthik Natarajan	2024-12-07
Lakshmi Narayan	2024-12-08
Suresh Babu	2024-12-09

8. Retrieve the orders with a total amount greater than 150.00.

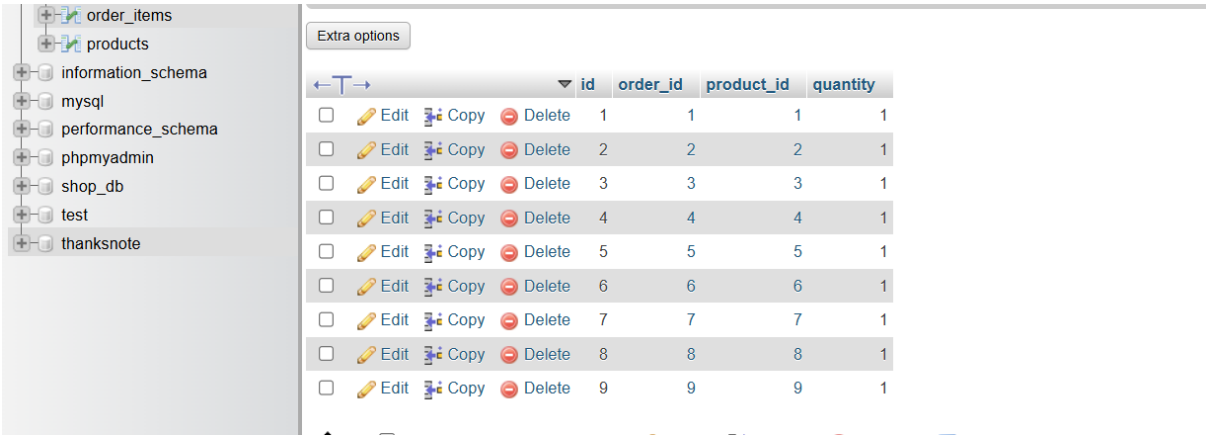
SELECT * FROM orders WHERE total_amount > 150.00;



	id	customer_id	order_date	total_amount	order_items
Edit Copy Delete	1	1	2024-12-01	15000.00	NULL
Edit Copy Delete	2	2	2024-12-02	45000.00	NULL
Edit Copy Delete	3	3	2024-12-03	20000.00	NULL
Edit Copy Delete	4	4	2024-12-04	30000.00	NULL
Edit Copy Delete	5	5	2024-12-05	12000.00	NULL
Edit Copy Delete	6	6	2024-12-06	8000.00	NULL
Edit Copy Delete	7	7	2024-12-07	40000.00	NULL
Edit Copy Delete	8	8	2024-12-08	35000.00	NULL
Edit Copy Delete	9	9	2024-12-09	10000.00	NULL

9. Normalize the database by creating a separate table for order items and updating the orders table to reference the order_items table.

```
INSERT INTO order_items (order_id, product_id, quantity)

SELECT o.id AS order_id,

    p.id AS product_id,

    1 AS quantity

FROM orders o

JOIN customers c ON o.customer_id = c.id

JOIN products p ON p.price = o.total_amount;
```



10. Retrieve the average total of all orders.

```
SELECT AVG(total) AS average_total
FROM (
    SELECT order_id, SUM(p.price * oi.quantity) AS total
    FROM order_items oi
    JOIN products p ON oi.product_id = p.id
    GROUP BY order_id
) AS order_totals;
```