```
##  Create the customers table
mysql>desc customers;
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| id      | int         | NO   | PRI | NULL    | auto_increment |
| name    | varchar(30) | NO   |     | NULL    |                |
| email   | varchar(100)| NO   |     | NULL    |                |
| address | varchar(200)| NO   |     | NULL    |                |
+---------+-------------+------+-----+---------+----------------+

## Create the orders table
mysql> desc orders;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| id           | int           | NO   | PRI | NULL    | auto_increment |
| customer_id  | int           | YES  | MUL | NULL    |                |
| order_date   | date          | NO   |     | NULL    |                |
| total_amount | decimal(10,2) | NO   |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+


## Create the products table

mysql> desc products;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | int           | NO   | PRI | NULL    | auto_increment |
| name        | varchar(30)   | NO   |     | NULL    |                |
| price       | decimal(10,2) | NO   |     | NULL    |                |
| description | text          | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+


## Insert sample data into customers table

mysql> select * from customers;
```

```
+----+---------------+---------------------+-----------------+
| id | name          | email               | address         |
+----+---------------+---------------------+-----------------+
|  1 | Alice Smith   | alice@example.com   | 123Aapple st    |
|  2 | Bob  Brown    | bob@example.com     | 456 Banana st   |
|  3 | Charlie Davis | charlie@example.com | 789 Cherry Blvd |
+----+---------------+---------------------+-----------------+
```

Insert sample data into orders table

```
mysql> select * from orders;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  1 |           1 | 2023-10-01 |       120.50 |
|  2 |           2 | 2023-11-01 |       200.00 |
|  3 |           1 | 2023-10-15 |        80.00 |
|  4 |           3 | 2023-09-30 |       150.75 |
+----+-------------+------------+--------------+
```

Insert sample data into products table

```
mysql> select * from products;
+----+-----------+-------+--------------------------+
| id | name      | price | description              |
+----+-----------+-------+--------------------------+
|  1 | Product A | 30.00 | Description of Product A |
|  2 | Product B | 25.00 | Description of Product B |
|  3 | Product C | 45.00 | Description of Product C |
+----+-----------+-------+--------------------------+
```

Queries

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

```
mysql> select distinct customers.name, customers.email from customers
    -> join orders on
    -> customers.id = orders.customer_id
    -> where orders.order_date >= curdate()-interval 30 day;
```

```
Empty set (0.01 sec)
```

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

```
mysql> select customers.name, sum(orders.total_amount)as total_spent
    -> from customers
    -> join orders on customers.id = orders.customer_id
    -> group by customers.id;
+---------------+-------------+
| name          | total_spent |
+---------------+-------------+
| Alice Smith   |      200.50 |
| Bob  Brown    |      200.00 |
| Charlie Davis |      150.75 |
+---------------+-------------+
```

3. Update the price of Product C to 45.00.

```
mysql> update products
    -> set price = 45.00
    -> where name = 'Product C';
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select * from products;
+----+-----------+-------+--------------------------+
| id | name      | price | description              |
+----+-----------+-------+--------------------------+
|  1 | Product A | 30.00 | Description of Product A |
|  2 | Product B | 25.00 | Description of Product B |
|  3 | Product C | 45.00 | Description of Product C |
+----+-----------+-------+--------------------------+
```

4. Add a new column discount to the products table

```
mysql> alter table products
    -> add column discount decimal(5,2) default 0.00;
Query OK, 0 rows affected (0.09 sec)
```

```
Records: 0  Duplicates: 0  Warnings: 0

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

mysql> select name, price
    -> from products
    -> order by price desc
    -> limit 3;
+-----------+-------+
| name      | price |
+-----------+-------+
| Product C | 45.00 |
| Product A | 30.00 |
| Product B | 25.00 |
+-----------+-------+
3 rows in set (0.00 sec)



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

mysql> select distinct customers.name
    -> from customers
    -> join orders on customers.id = orders.customer_id
    -> join order_items on orders.id = order_items.order_id
    -> join products on order_items.product_id = products.id
    -> where products.name = 'Product A';
+-------------+
| name        |
+-------------+
| Alice Smith |
+-------------+
1 row in set (0.01 sec)

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

mysql> select customers.name, orders.order_date
    -> from orders
    -> join customers on orders.customer_id = customers.id;
```

```
+---------------+------------+
| name          | order_date |
+---------------+------------+
| Alice Smith   | 2023-10-01 |
| Alice Smith   | 2023-10-15 |
| Bob  Brown    | 2023-11-01 |
| Charlie Davis | 2023-09-30 |
+---------------+------------+
```

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

```
mysql> select * from orders
    -> where total_amount>150.00
    -> ;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  2 |           2 | 2023-11-01 |       200.00 |
|  4 |           3 | 2023-09-30 |       150.75 |
+----+-------------+------------+--------------+
```

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

```
// Create the order_items table

mysql>CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(id),
    FOREIGN KEY (product_id) REFERENCES products(id)
);

mysql> select * from order_items;
+----+----------+------------+----------+-------+
| id | order_id | product_id | quantity | price |
+----+----------+------------+----------+-------+
|  1 |        1 |          1 |        2 | 30.00 |
```

```
|  2 |         1 |          2 |         1 | 25.00 |
|  3 |         2 |          3 |         3 | 45.00 |
|  4 |         3 |          1 |         1 | 40.00 |
|  5 |         3 |          2 |         1 | 20.00 |
+----+-----------+------------+-----------+-------+


# Remove the total_amount column from orders
mysql> alter table orders
    -> drop column total_amount;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from orders;
+----+-------------+------------+
| id | customer_id | order_date |
+----+-------------+------------+
|  1 |           1 | 2023-10-01 |
|  2 |           2 | 2023-11-01 |
|  3 |           1 | 2023-10-15 |
|  4 |           3 | 2023-09-30 |
+----+-------------+------------+


## 10. Retrieve the average total of all orders.
mysql> select avg(order_total)as average_order_total
    -> from(
    -> select order_id, sum(price*quantity)as order_total
    -> from order_items
    -> group by order_id
    -> )as order_totals;
+---------------------+
| average_order_total |
+---------------------+
|           93.333333 |
+---------------------+
1 row in set (0.01 sec)
```