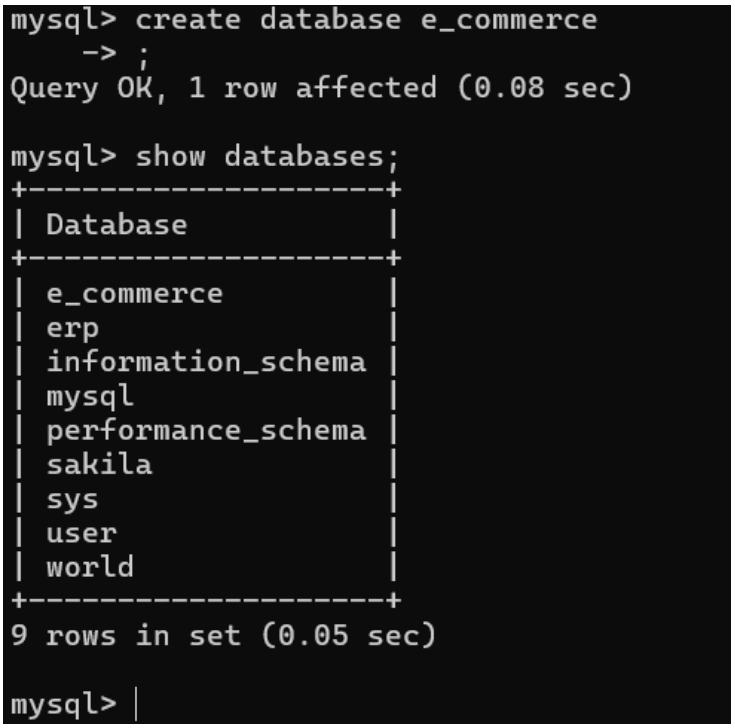
MySQL Assignment:

1. Create Database e_commerce



2. Create following Tables:

Customers:

a. customer_id - int auto-increment primary key
b. name - varchar(50)
c. email - varchar(50)
d. mobile - varchar(15)

```
mysql> use e_commerce;
Database changed
mysql> create table customers (
    ->   customer_id int auto_increment primary key,
    ->   name varchar(50),
    ->   email varchar(50),
    ->   mobile varchar(15)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

Products:

a. id - int
b. name - varchar(50) not null
c. description - varchar(200)
d. price - decimal(10, 2) not null
e. category - varchar(50)

```
mysql> create table products (
    ->   id int,
    ->   name varchar(50) not null,
    ->   description varchar(200),
    ->   price decimal(10, 2) not null,
    ->   category varchar(50)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

3. Modify Tables(using Alter keyword):

a. Add not null on name and email in the Customers table

```
mysql> alter table customers modify name varchar(50) not null,
    -> modify email varchar(50) not null;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

b. Add unique key on email in the Customers table

```
mysql> alter table customer add unique(email);
ERROR 1146 (42S02): Table 'e_commerce.customer' doesn't exist
mysql> alter table customers add unique(email);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

c. Add column age in the Customers table

```
mysql> alter table customers add age int;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

d. Change column name from id to product_id in the Products table;

```
mysql> alter table products rename column id to product_id;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

e. Add primary key and auto increment on product_id in the Products table

```
mysql> alter table products modify product_id int auto_increment primary key;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

f. Change datatype of description from varchar to text in the Products table

```
mysql> alter table products modify description text;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

4. **Create table Order:**
    a. order_id - int auto-increment primary key
    b. customer_id - int -foreign key
    c. product_id - int
    d. quantity - int not null,
    e. order_date - date not null,
    f. status - enum(Pending, Success, Cancel),
    g. payment_method - enum(Credit, Debit, UPI),
    h. total_amount - decimal(10, 2) not null

```
mysql> create table orders (
    -> order_id int auto_increment primary key,
    -> customer_id int,
    -> product_id int,
    -> quantity int not null,
    -> order_date date not null,
    -> status enum('pending', 'success', 'cancel'),
    -> payment_method enum('credit', 'debit', 'upi'),
    -> total_amount decimal(10, 2) not null,
    -> foreign key (customer_id) references customers,
    -> foreign key (product_id) references products
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql>
```

5. Modify Orders Table(using Alter keyword):

a. Change table name Order -> Orders

b. Set default value pending in status.

```
mysql> alter table orders modify status enum('pending', 'success', 'cancel') default 'pending';
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

c. Modify payment_method ENUM to add one more value: 'COD'

```
mysql> alter table orders modify status enum('credit', 'debit', 'upi', 'COD');
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

d. Make product id as foreign key

```
mysql> alter table orders add foreign key (product_id) references products(product_id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

6. Insert 20 sample records in all the tables. (taking the sample data from Chat GPT)

Customers table:

```
mysql> insert into customers (name, email, mobile, age) values
    -> ('John Doe', 'john@example.com', '1234567890', 30),
    -> ('Jane Smith', 'jane@example.com', '1234567891', 25),
    -> ('Mike Johnson', 'mike@example.com', '1234567892', 28),
    -> ('Emily Davis', 'emily@example.com', '1234567893', 35),
    -> ('Chris Brown', 'chris@example.com', '1234567894', 40),
    -> ('Sarah Wilson', 'sarah@example.com', '1234567895', 22),
    -> ('David Lee', 'david@example.com', '1234567896', 29),
    -> ('Laura Adams', 'laura@example.com', '1234567897', 32),
    -> ('Kevin White', 'kevin@example.com', '1234567898', 33),
    -> ('Olivia Taylor', 'olivia@example.com', '1234567899', 27),
    -> ('Sophia Anderson', 'sophia@example.com', '1234567800', 24),
    -> ('James Clark', 'james@example.com', '1234567801', 26),
    -> ('Isabella Walker', 'isabella@example.com', '1234567802', 31),
    -> ('Ethan Martinez', 'ethan@example.com', '1234567803', 38),
    -> ('Madison Scott', 'madison@example.com', '1234567804', 34),
    -> ('Alexander King', 'alexander@example.com', '1234567805', 23),
    -> ('Mia Lee', 'mia@example.com', '1234567806', 30),
    -> ('Jack Harris', 'jack@example.com', '1234567807', 41),
    -> ('Ava Robinson', 'ava@example.com', '1234567808', 29);
Query OK, 19 rows affected (0.02 sec)
Records: 19  Duplicates: 0  Warnings: 0
```

Products table

```
mysql> insert into products (product_id, name, description, price, category) values
    -> (1, 'Laptop', 'High performance laptop', 799.99, 'electronics'),
    -> (2, 'Smartphone', 'Latest model smartphone', 499.99, 'electronics'),
    -> (3, 'Headphones', 'Noise-cancelling headphones', 120.50, 'electronics'),
    -> (4, 'TV', '4K UHD Television', 599.99, 'electronics'),
    -> (5, 'Refrigerator', 'Energy efficient fridge', 349.00, 'appliances'),
    -> (6, 'Washing Machine', 'Top-load washing machine', 249.99, 'appliances'),
    -> (7, 'Microwave', 'Compact microwave oven', 89.99, 'appliances'),
    -> (8, 'Blender', 'Powerful kitchen blender', 49.99, 'appliances'),
    -> (9, 'Shoes', 'Running shoes', 69.99, 'fashion'),
    -> (10, 'T-shirt', 'Cotton T-shirt', 15.99, 'fashion'),
    -> (11, 'Jeans', 'Denim jeans', 39.99, 'fashion'),
    -> (12, 'Jacket', 'Leather jacket', 129.99, 'fashion'),
    -> (13, 'Coffee Maker', 'Automatic coffee maker', 99.99, 'home & kitchen'),
    -> (14, 'Toaster', 'Two-slice toaster', 29.99, 'home & kitchen'),
    -> (15, 'Cookware Set', 'Non-stick cookware set', 129.99, 'home & kitchen'),
    -> (16, 'Microwave Oven', 'Multi-functional microwave oven', 199.99, 'home & kitchen'),
    -> (17, 'Bicycle', 'Mountain bicycle', 350.00, 'sports'),
    -> (18, 'Yoga Mat', 'Comfortable yoga mat', 19.99, 'sports'),
    -> (19, 'Tennis Racket', 'Professional tennis racket', 150.00, 'sports'),
    -> (20, 'Dumbbells', 'Pair of 5kg dumbbells', 29.99, 'sports');
Query OK, 20 rows affected (0.01 sec)
Records: 20  Duplicates: 0  Warnings: 0
```

Orders table:

```
mysql> select * from orders;
+----------+-------------+------------+----------+------------+---------+----------------+--------------+
| order_id | customer_id | product_id | quantity | order_date | status  | payment_method | total_amount |
+----------+-------------+------------+----------+------------+---------+----------------+--------------+
|       81 |           1 |          1 |        1 | 2025-02-01 | pending | credit         |       799.99 |
|      101 |           1 |          1 |        1 | 2025-02-01 | pending | credit         |       799.99 |
|      102 |           2 |          2 |        1 | 2025-02-02 | success | debit          |       499.99 |
|      121 |           3 |          3 |        1 | 2025-02-03 | cancel  | upi            |       120.50 |
|      122 |           4 |          4 |        1 | 2025-02-04 | pending | credit         |       599.99 |
|      123 |           5 |          5 |        1 | 2025-02-05 | success | debit          |       349.00 |
|      124 |           6 |          6 |        1 | 2025-02-06 | success | upi            |       249.99 |
|      125 |           7 |          7 |        1 | 2025-02-07 | pending | credit         |        89.99 |
|      126 |           8 |          8 |        1 | 2025-02-08 | success | debit          |        49.99 |
|      127 |           9 |          9 |        1 | 2025-02-09 | cancel  | upi            |        69.99 |
|      128 |          10 |         10 |        1 | 2025-02-10 | pending | credit         |        15.99 |
|      129 |          11 |         11 |        1 | 2025-02-11 | success | debit          |        39.99 |
|      130 |          12 |         12 |        1 | 2025-02-12 | cancel  | upi            |       129.99 |
|      131 |          13 |         13 |        1 | 2025-02-13 | pending | credit         |        99.99 |
|      132 |          14 |         14 |        1 | 2025-02-14 | success | debit          |        29.99 |
|      133 |          15 |         15 |        1 | 2025-02-15 | pending | upi            |       129.99 |
|      134 |          16 |         16 |        1 | 2025-02-16 | success | credit         |       199.99 |
|      135 |          17 |         17 |        1 | 2025-02-17 | pending | debit          |       350.00 |
|      136 |          18 |         18 |        1 | 2025-02-18 | success | upi            |        19.99 |
|      137 |          19 |         19 |        1 | 2025-02-19 | cancel  | credit         |       150.00 |
+----------+-------------+------------+----------+------------+---------+----------------+--------------+
20 rows in set (0.00 sec)

mysql>
```

7. Perform following queries:

a. Count the number of products as product_count in each category.

```
mysql> select category, count(*) as product_count
    -> from products
    -> group by category;
+-----------------+---------------+
| category        | product_count |
+-----------------+---------------+
| electronics     |             4 |
| appliances      |             4 |
| fashion         |             4 |
| home & kitchen  |             4 |
| sports          |             4 |
+-----------------+---------------+
5 rows in set (0.02 sec)
```

b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.

```
mysql> select * from products where category='electronics' and price between 50 and 500 and name like '%a%';
+------------+------------+------------------------------+--------+-------------+
| product_id | name       | description                  | price  | category    |
+------------+------------+------------------------------+--------+-------------+
|          2 | Smartphone | Latest model smartphone      | 499.99 | electronics |
|          3 | Headphones | Noise-cancelling headphones  | 120.50 | electronics |
+------------+------------+------------------------------+--------+-------------+
2 rows in set (0.01 sec)
```

c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2. (limit/offset)

```
mysql> select * from products where category='electronics' order by price desc limit 3 offset 2;
+------------+------------+------------------------------+--------+-------------+
| product_id | name       | description                  | price  | category    |
+------------+------------+------------------------------+--------+-------------+
|          2 | Smartphone | Latest model smartphone      | 499.99 | electronics |
|          3 | Headphones | Noise-cancelling headphones  | 120.50 | electronics |
+------------+------------+------------------------------+--------+-------------+
2 rows in set (0.00 sec)
```

d. Retrieve customers who have not placed any orders. (sub query approach)

```
mysql> select * from customers where customer_id not in (select customer_id from orders);
Empty set (0.02 sec)
```

e. Find the average total amount spent by each customer.

```
mysql> select customer_id,avg(total_amount) as avgerage from orders group by customer_id;
+-------------+------------+
| customer_id | avgerage   |
+-------------+------------+
|           1 | 799.990000 |
|           2 | 499.990000 |
|           3 | 120.500000 |
|           4 | 599.990000 |
|           5 | 349.000000 |
|           6 | 249.990000 |
|           7 |  89.990000 |
|           8 |  49.990000 |
|           9 |  69.990000 |
|          10 |  15.990000 |
|          11 |  39.990000 |
|          12 | 129.990000 |
|          13 |  99.990000 |
|          14 |  29.990000 |
|          15 | 129.990000 |
|          16 | 199.990000 |
|          17 | 350.000000 |
|          18 |  19.990000 |
|          19 | 150.000000 |
+-------------+------------+
19 rows in set (0.00 sec)
```

f. Get the products that have a price less than the average price of all products.

```
mysql> select * from products where price < (select avg(price) from products);
+------------+----------------+---------------------------------+--------+----------------+
| product_id | name           | description                     | price  | category       |
+------------+----------------+---------------------------------+--------+----------------+
|          3 | Headphones     | Noise-cancelling headphones     | 120.50 | electronics    |
|          7 | Microwave      | Compact microwave oven          |  89.99 | appliances     |
|          8 | Blender        | Powerful kitchen blender        |  49.99 | appliances     |
|          9 | Shoes          | Running shoes                   |  69.99 | fashion        |
|         10 | T-shirt        | Cotton T-shirt                  |  15.99 | fashion        |
|         11 | Jeans          | Denim jeans                     |  39.99 | fashion        |
|         12 | Jacket         | Leather jacket                  | 129.99 | fashion        |
|         13 | Coffee Maker   | Automatic coffee maker          |  99.99 | home & kitchen |
|         14 | Toaster        | Two-slice toaster               |  29.99 | home & kitchen |
|         15 | Cookware Set   | Non-stick cookware set          | 129.99 | home & kitchen |
|         16 | Microwave Oven | Multi-functional microwave oven | 199.99 | home & kitchen |
|         18 | Yoga Mat       | Comfortable yoga mat            |  19.99 | sports         |
|         19 | Tennis Racket  | Professional tennis racket      | 150.00 | sports         |
|         20 | Dumbbells      | Pair of 5kg dumbbells           |  29.99 | sports         |
+------------+----------------+---------------------------------+--------+----------------+
14 rows in set (0.01 sec)
```

g. Calculate the total quantity of products ordered by each customer:

```
mysql> select customer_id,sum(quantity) as total from orders group by customer_id;
+-------------+-------+
| customer_id | total |
+-------------+-------+
|           1 |     2 |
|           2 |     1 |
|           3 |     1 |
|           4 |     1 |
|           5 |     1 |
|           6 |     1 |
|           7 |     1 |
|           8 |     1 |
|           9 |     1 |
|          10 |     1 |
|          11 |     1 |
|          12 |     1 |
|          13 |     1 |
|          14 |     1 |
|          15 |     1 |
|          16 |     1 |
|          17 |     1 |
|          18 |     1 |
|          19 |     1 |
+-------------+-------+
19 rows in set (0.00 sec)
```

h. List all orders along with customer name and product name.

```
mysql> select o.order_id,c.name as customer_name,p.name as product_name,o.quantity,o.order_date,o.status,o.payment_method,o.total_amount from orders o join
customers c on o.customer_id=c.customer_id join products p on o.product_id=p.product_id;
+----------+---------------+----------------+----------+------------+---------+----------------+--------------+
| order_id | customer_name | product_name   | quantity | order_date | status  | payment_method | total_amount |
+----------+---------------+----------------+----------+------------+---------+----------------+--------------+
|       81 | John Doe      | Laptop         |        1 | 2025-02-01 | pending | credit         |       799.99 |
|      101 | John Doe      | Laptop         |        1 | 2025-02-01 | pending | credit         |       799.99 |
|      102 | Jane Smith    | Smartphone     |        1 | 2025-02-02 | success | debit          |       499.99 |
|      121 | Mike Johnson  | Headphones     |        1 | 2025-02-03 | cancel  | upi            |       120.50 |
|      122 | Emily Davis   | TV             |        1 | 2025-02-04 | pending | credit         |       599.99 |
|      123 | Chris Brown   | Refrigerator   |        1 | 2025-02-05 | success | debit          |       349.00 |
|      124 | Sarah Wilson  | Washing Machine |        1 | 2025-02-06 | success | upi            |       249.99 |
|      125 | David Lee     | Microwave      |        1 | 2025-02-07 | pending | credit         |        89.99 |
|      126 | Laura Adams   | Blender        |        1 | 2025-02-08 | success | debit          |        49.99 |
|      127 | Kevin White   | Shoes          |        1 | 2025-02-09 | cancel  | upi            |        69.99 |
|      128 | Olivia Taylor | T-shirt        |        1 | 2025-02-10 | pending | credit         |        15.99 |
|      129 | Sophia Anderson | Jeans          |        1 | 2025-02-11 | success | debit          |        39.99 |
|      130 | James Clark   | Jacket         |        1 | 2025-02-12 | cancel  | upi            |       129.99 |
|      131 | Isabella Walker | Coffee Maker   |        1 | 2025-02-13 | pending | credit         |        99.99 |
|      132 | Ethan Martinez | Toaster        |        1 | 2025-02-14 | success | debit          |        29.99 |
|      133 | Madison Scott | Cookware Set   |        1 | 2025-02-15 | pending | upi            |       129.99 |
|      134 | Alexander King | Microwave Oven |        1 | 2025-02-16 | success | credit         |       199.99 |
|      135 | Mia Lee       | Bicycle        |        1 | 2025-02-17 | pending | debit          |       350.00 |
|      136 | Jack Harris   | Yoga Mat       |        1 | 2025-02-18 | success | upi            |        19.99 |
|      137 | Ava Robinson  | Tennis Racket  |        1 | 2025-02-19 | cancel  | credit         |       150.00 |
+----------+---------------+----------------+----------+------------+---------+----------------+--------------+
20 rows in set (0.00 sec)
```

i. Find products that have never been ordered.

```
mysql> select * from products where product_id not in (select product_id from orders);
+-------------+-----------+------------------------+-------+----------+
| product_id | name      | description            | price | category |
+-------------+-----------+------------------------+-------+----------+
|          20 | Dumbbells | Pair of 5kg dumbbells | 29.99 | sports   |
+-------------+-----------+------------------------+-------+----------+
1 row in set (0.00 sec)

mysql>
```