

SQL Assignment-1

1.Create Database e_commerce

```
mysql> create database e_commerce;  
Query OK, 1 row affected (0.23 sec)
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| data |  
| e_commerce |  
| library |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
7 rows in set (0.44 sec)
```

2.Create following Tables:

Customers:

- customer_id - int auto-increment primary key
- name - varchar(50)
- email - varchar(50)
- mobile - varchar(15)

```
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 (0.09 sec)
```

Products:

- id - int
- name - varchar(50) not null
- description - varchar(200)
- price - decimal(10, 2) not null
- category - varchar(50)

```
mysql> create table Products(  
  -> id int Primary Key,  
  -> name varchar(50) not null,  
  -> description varchar(200),  
  -> price decimal(10,2) not null,  
  -> category varchar(50)  
  -> );  
Query OK, 0 rows affected (0.80 sec)
```

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 (1.31 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

- b. Add unique key on email in the Customers table

```
mysql> alter table Customers  
  -> add unique(email);  
Query OK, 0 rows affected (0.41 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

- c. Add column age in the Customers table

```
mysql> alter table Customers  
  -> add column age int;  
Query OK, 0 rows affected (0.25 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

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

```
mysql> alter table Products  
  -> change column id product_id int;  
Query OK, 0 rows affected (0.05 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

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

```
mysql> alter table Products
    -> modify column product_id int auto_increment ;
Query OK, 0 rows affected (0.08 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 column description text;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

4. Create table Order:

- order_id - int auto-increment primary key
- customer_id - int -foreign key
- 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

```
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') not null,
    -> payment_method ENUM('Credit','Debit','UPI') not null,
    -> total_amount decimal(10,2) not null,
    -> foreign key (customer_id) references Customers(customer_id),
    -> foreign key (product_id) references Products(product_id)
    -> );
Query OK, 0 rows affected (0.67 sec)
```

5. Modify Orders Table(using Alter keyword):

- Change table name Order -> Orders
- Set default value pending in status.

```
mysql> alter table Orders
  -> modify column status ENUM('Pending','Success','Cancel') not null default 'Pending';
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

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

```
mysql> alter table Orders
  -> modify column payment_method enum('Credit','Debit','UPI','COD') not null;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- d. Make product id as foreign key

```
mysql> ALTER TABLE Orders
  -> ADD CONSTRAINT fk_product
  -> FOREIGN KEY (product_id) REFERENCES Products(product_id);
Query OK, 20 rows affected (0.05 sec)
Records: 20 Duplicates: 0 Warnings: 0
mysql>
```

6. 20 sample records in all the tables.

```
mysql> desc customers;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| customer_id | int(11)       | NO   | PRI | NULL    | auto_increment |
| name       | varchar(50)   | NO   |     | NULL    |                |
| email      | varchar(50)   | NO   | UNI | NULL    |                |
| mobile     | varchar(15)   | YES  |     | NULL    |                |
| age        | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> desc products;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| product_id    | int(11)       | NO   | PRI | NULL    | auto_increment |
| name          | varchar(50)   | NO   |     | NULL    |                |
| description    | text          | YES  |     | NULL    |                |
| price         | decimal(10,2) | NO   |     | NULL    |                |
| category      | varchar(50)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

```
mysql> desc Orders;
```

Field	Type	Null	Key	Default	Extra
order_id	int(11)	NO	PRI	NULL	auto_increment
customer_id	int(11)	YES	MUL	NULL	
product_id	int(11)	YES	MUL	NULL	
quantity	int(11)	NO		NULL	
status	enum('Pending','Success','Cancel')	NO		Pending	
payment_method	enum('Credit','Debit','UPI','COD')	NO		NULL	
total_amount	decimal(10,2)	NO		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Customers (name, email, mobile, age) VALUES
```

```
-> ('John Doe', 'johndoe@example.com', '9876543210', 30),
-> ('Alice Smith', 'alice.smith@example.com', '9123456789', 28),
-> ('Bob Johnson', 'bob.johnson@example.com', '9234567890', 35),
-> ('Charlie Brown', 'charlie.brown@example.com', '9345678901', 40),
-> ('David Wilson', 'david.wilson@example.com', '9456789012', 32),
-> ('Emma Davis', 'emma.davis@example.com', '9567890123', 27),
-> ('Frank Miller', 'frank.miller@example.com', '9678901234', 29),
-> ('Grace Lee', 'grace.lee@example.com', '9789012345', 26),
-> ('Hannah Scott', 'hannah.scott@example.com', '9890123456', 31),
-> ('Ian Clark', 'ian.clark@example.com', '9901234567', 33),
-> ('Jack White', 'jack.white@example.com', '9012345678', 45),
-> ('Kelly Green', 'kelly.green@example.com', '9123456780', 29),
-> ('Liam Harris', 'liam.harris@example.com', '9234567891', 36),
-> ('Mia Turner', 'mia.turner@example.com', '9345678902', 25),
-> ('Nathan Lewis', 'nathan.lewis@example.com', '9456789013', 38),
-> ('Olivia Walker', 'olivia.walker@example.com', '9567890124', 34),
-> ('Paul Hall', 'paul.hall@example.com', '9678901235', 39),
-> ('Quinn Allen', 'quinn.allen@example.com', '9789012346', 28),
-> ('Rachel Adams', 'rachel.adams@example.com', '9890123457', 30),
-> ('Steve Baker', 'steve.baker@example.com', '9901234568', 37);
```

```
Query OK, 20 rows affected (0.30 sec)
```

```
Records: 20 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Products (name, description, price, category)
-> VALUES
-> ('Laptop', 'High-performance laptop with 16GB RAM and 512GB SSD.', 1200.99, 'Electronics'),
-> ('Smartphone', 'Latest 5G smartphone with OLED display.', 899.50, 'Electronics'),
-> ('Headphones', 'Noise-cancelling wireless headphones.', 199.99, 'Accessories'),
-> ('Smartwatch', 'Fitness tracking smartwatch with heart rate monitor.', 150.75, 'Wearables'),
-> ('Gaming Mouse', 'Ergonomic gaming mouse with RGB lighting.', 59.99, 'Accessories'),
-> ('Mechanical Keyboard', 'RGB mechanical keyboard with customizable keys.', 99.99, 'Accessories'),
-> ('4K TV', '55-inch Smart TV with Ultra HD resolution.', 799.99, 'Electronics'),
-> ('Bluetooth Speaker', 'Portable Bluetooth speaker with deep bass.', 129.99, 'Audio'),
-> ('Wireless Earbuds', 'True wireless earbuds with active noise cancellation.', 159.99, 'Audio'),
-> ('Tablet', '10-inch tablet with stylus support.', 499.99, 'Electronics'),
-> ('Backpack', 'Water-resistant laptop backpack with multiple compartments.', 49.99, 'Accessories'),
-> ('Gaming Console', 'Latest-gen gaming console with 4K support.', 499.99, 'Gaming'),
-> ('External Hard Drive', '2TB external HDD for data storage.', 89.99, 'Storage'),
-> ('Monitor', '27-inch IPS monitor with 144Hz refresh rate.', 299.99, 'Electronics'),
-> ('Wireless Charger', 'Fast wireless charger for smartphones.', 39.99, 'Accessories'),
-> ('Action Camera', 'Waterproof action camera with 4K recording.', 249.99, 'Cameras'),
-> ('Smart Home Hub', 'Voice-controlled smart home hub.', 179.99, 'Smart Home'),
-> ('Fitness Tracker', 'Smart fitness band with step counter and sleep monitor.', 79.99, 'Wearables'),
-> ('VR Headset', 'Virtual reality headset with motion controllers.', 399.99, 'Gaming'),
-> ('Coffee Maker', 'Automatic coffee machine with programmable settings.', 129.99, 'Appliances');
Query OK, 20 rows affected (0.00 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Orders (customer_id, product_id, quantity, status, payment_method, total_amount)
-> VALUES
-> (1, 2, 3, 'Pending', 'Credit', 150.00),
-> (2, 5, 1, 'Success', 'Debit', 200.50),
-> (3, 3, 2, 'Pending', 'UPI', 75.99),
-> (4, 1, 5, 'Success', 'Credit', 500.00),
-> (5, 6, 2, 'Cancel', 'UPI', 120.75),
-> (6, 4, 1, 'Pending', 'COD', 180.00),
-> (7, 2, 3, 'Success', 'Debit', 250.60),
-> (8, 7, 1, 'Cancel', 'Credit', 99.99),
-> (9, 9, 4, 'Success', 'UPI', 330.00),
-> (10, 10, 2, 'Pending', 'COD', 450.99),
-> (11, 3, 3, 'Success', 'Credit', 199.99),
-> (12, 8, 1, 'Cancel', 'Debit', 89.50),
-> (13, 6, 5, 'Pending', 'UPI', 600.00),
-> (14, 5, 2, 'Success', 'COD', 220.80),
-> (15, 2, 4, 'Pending', 'Credit', 310.30),
-> (16, 9, 1, 'Success', 'Debit', 150.00),
-> (17, 7, 3, 'Cancel', 'UPI', 140.40),
-> (18, 4, 2, 'Pending', 'COD', 280.70),
-> (19, 1, 5, 'Success', 'Credit', 510.50),
-> (20, 8, 1, 'Pending', 'Debit', 99.99);
Query OK, 20 rows affected (0.48 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

7. Perform following queries:

- 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
Accessories	5
Appliances	1
Audio	2
Cameras	1
Electronics	5
Gaming	2
Smart Home	1
Storage	1
Wearables	2

9 rows in set (0.39 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
10	Tablet	10-inch tablet with stylus support.	499.99	Electronics

1 row in set (0.49 sec)

- c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.

```
mysql> SELECT *
-> FROM Products
-> WHERE category = 'Electronics'
-> ORDER BY price DESC
-> LIMIT 5 OFFSET 2;
```

product_id	name	description	price	category
7	4K TV	55-inch Smart TV with Ultra HD resolution.	799.99	Electronics
10	Tablet	10-inch tablet with stylus support.	499.99	Electronics
14	Monitor	27-inch IPS monitor with 144Hz refresh rate.	299.99	Electronics

3 rows in set (0.11 sec)

- d. Retrieve customers who have not placed any orders.

```
mysql> SELECT c.*
-> FROM Customers c
-> LEFT JOIN Orders o ON c.customer_id = o.customer_id
-> WHERE o.order_id IS NULL;
Empty set (0.58 sec)
```

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

```
mysql> SELECT o.customer_id, c.name, AVG(o.total_amount) AS avg_spent
-> FROM Orders o
-> JOIN Customers c ON o.customer_id = c.customer_id
-> GROUP BY o.customer_id, c.name;
+-----+-----+-----+
| customer_id | name          | avg_spent |
+-----+-----+-----+
| 1           | John Doe     | 150.000000 |
| 2           | Alice Smith  | 200.500000 |
| 3           | Bob Johnson  | 75.990000  |
| 4           | Charlie Brown| 500.000000 |
| 5           | David Wilson | 120.750000 |
| 6           | Emma Davis   | 180.000000 |
| 7           | Frank Miller | 250.600000 |
| 8           | Grace Lee    | 99.990000  |
| 9           | Hannah Scott | 330.000000 |
| 10          | Ian Clark    | 450.990000 |
| 11          | Jack White   | 199.990000 |
| 12          | Kelly Green  | 89.500000  |
| 13          | Liam Harris  | 600.000000 |
| 14          | Mia Turner   | 220.800000 |
| 15          | Nathan Lewis | 310.300000 |
| 16          | Olivia Walker| 150.000000 |
| 17          | Paul Hall    | 140.400000 |
| 18          | Quinn Allen  | 280.700000 |
| 19          | Rachel Adams | 510.500000 |
| 20          | Steve Baker  | 99.990000  |
+-----+-----+-----+
20 rows in set (0.68 sec)

mysql> SELECT *
```

- 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 wireless headphones.	199.99	Accessories
4	Smartwatch	Fitness tracking smartwatch with heart rate monitor.	150.75	Wearables
5	Gaming Mouse	Ergonomic gaming mouse with RGB lighting.	59.99	Accessories
6	Mechanical Keyboard	RGB mechanical keyboard with customizable keys.	99.99	Accessories
8	Bluetooth Speaker	Portable Bluetooth speaker with deep bass.	129.99	Audio
9	Wireless Earbuds	True wireless earbuds with active noise cancellation.	159.99	Audio
11	Backpack	Water-resistant laptop backpack with multiple compartments.	49.99	Accessories
13	External Hard Drive	2TB external HDD for data storage.	89.99	Storage
14	Monitor	27-inch IPS monitor with 144Hz refresh rate.	299.99	Electronics
15	Wireless Charger	Fast wireless charger for smartphones.	39.99	Accessories
16	Action Camera	Waterproof action camera with 4K recording.	249.99	Cameras
17	Smart Home Hub	Voice-controlled smart home hub.	179.99	Smart Home
18	Fitness Tracker	Smart fitness band with step counter and sleep monitor.	79.99	Wearables
20	Coffee Maker	Automatic coffee machine with programmable settings.	129.99	Appliances

14 rows in set (0.16 sec)

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

```
mysql> SELECT o.customer_id, c.name, SUM(o.quantity) AS total_quantity
-> FROM Orders o
-> JOIN Customers c ON o.customer_id = c.customer_id
-> GROUP BY o.customer_id, c.name;
```

customer_id	name	total_quantity
1	John Doe	3
2	Alice Smith	1
3	Bob Johnson	2
4	Charlie Brown	5
5	David Wilson	2
6	Emma Davis	1
7	Frank Miller	3
8	Grace Lee	1
9	Hannah Scott	4
10	Ian Clark	2
11	Jack White	3
12	Kelly Green	1
13	Liam Harris	5
14	Mia Turner	2
15	Nathan Lewis	4
16	Olivia Walker	1
17	Paul Hall	3
18	Quinn Allen	2
19	Rachel Adams	5
20	Steve Baker	1

20 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.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	total_amount
1	John Doe	Smartphone	3	150.00
2	Alice Smith	Gaming Mouse	1	200.50
3	Bob Johnson	Headphones	2	75.99
4	Charlie Brown	Laptop	5	500.00
5	David Wilson	Mechanical Keyboard	2	120.75
6	Emma Davis	Smartwatch	1	180.00
7	Frank Miller	Smartphone	3	250.60
8	Grace Lee	4K TV	1	99.99
9	Hannah Scott	Wireless Earbuds	4	330.00
10	Ian Clark	Tablet	2	450.99
11	Jack White	Headphones	3	199.99
12	Kelly Green	Bluetooth Speaker	1	89.50
13	Liam Harris	Mechanical Keyboard	5	600.00
14	Mia Turner	Gaming Mouse	2	220.80
15	Nathan Lewis	Smartphone	4	310.30
16	Olivia Walker	Wireless Earbuds	1	150.00
17	Paul Hall	4K TV	3	140.40
18	Quinn Allen	Smartwatch	2	280.70
19	Rachel Adams	Laptop	5	510.50
20	Steve Baker	Bluetooth Speaker	1	99.99

```
20 rows in set (0.00 sec)
```

- i. Find products that have never been ordered.

```
mysql> SELECT p.*
-> FROM Products p
-> WHERE NOT EXISTS (
-> SELECT 1 FROM Orders o WHERE o.product_id = p.product_id
-> );
```

product_id	name	description	price	category
11	Backpack	Water-resistant laptop backpack with multiple compartments.	49.99	Accessories
12	Gaming Console	Latest-gen gaming console with 4K support.	499.99	Gaming
13	External Hard Drive	2TB external HDD for data storage.	89.99	Storage
14	Monitor	27-inch IPS monitor with 144Hz refresh rate.	299.99	Electronics
15	Wireless Charger	Fast wireless charger for smartphones.	39.99	Accessories
16	Action Camera	Waterproof action camera with 4K recording.	249.99	Cameras
17	Smart Home Hub	Voice-controlled smart home hub.	179.99	Smart Home
18	Fitness Tracker	Smart fitness band with step counter and sleep monitor.	79.99	Wearables
19	VR Headset	Virtual reality headset with motion controllers.	399.99	Gaming
20	Coffee Maker	Automatic coffee machine with programmable settings.	129.99	Appliances

```
10 rows in set (0.00 sec)
```