HEXAWARE TECHNICAL TRAINING

MySQL Coding Challenge

RUTH ROSHINI S

Creating Database:

create database Ecom;

```
mysql> create database Ecom;
Query OK, 1 row affected (0.04 sec)
```

Selecting Database:

Use Ecom;

```
mysql> use Ecom;
Database changed
```

CREATING TABLES:

Customer table:

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    password VARCHAR(100));
```

```
mysql> CREATE TABLE customers (
    ->     customer_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     email VARCHAR(100),
    ->     password VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.12 sec)
```

Product Table:

```
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    name VARCHAR(100),
    description VARCHAR(255),
```

price DECIMAL(10, 2),

stockQuantity INT);

```
mysql> CREATE TABLE products (
    ->     product_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     description VARCHAR(255),
    ->     price DECIMAL(10, 2),
    ->     stockQuantity INT
    -> );
Query OK, 0 rows affected (0.07 sec)
```

Cart Table:

CREATE TABLE cart (

cart_id INT PRIMARY KEY,

customer_id INT,

product_id INT,

quantity INT,

FOREIGN KEY (customer_id) REFERENCES customers(customer_id),

FOREIGN KEY (product_id) REFERENCES products(product_id));

```
mysql> CREATE TABLE cart (
    ->     cart_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     product_id INT,
    ->     quantity INT,
    ->     FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    ->     FOREIGN KEY (product_id) REFERENCES products(product_id)
    -> );
Query OK, 0 rows affected (0.17 sec)
```

Order Table:

CREATE TABLE orders (

order_id INT PRIMARY KEY,

customer_id INT,

order_date DATE,

total_price DECIMAL(10, 2),

shipping_address VARCHAR(255),

FOREIGN KEY (customer_id) REFERENCES customers(customer_id));

```
mysql> CREATE TABLE orders (
    ->     order_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     order_date DATE,
    ->     total_price DECIMAL(10, 2),
    ->     shipping_address VARCHAR(255),
    ->     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
    -> );
Query OK, 0 rows affected (0.12 sec)
```

Order Items Table:

CREATE TABLE order_items (

order_item_id INT PRIMARY KEY,

order_id INT,

product_id INT,

quantity INT,

item_amount DECIMAL(10, 2),

FOREIGN KEY (order_id) REFERENCES orders(order_id),

FOREIGN KEY (product_id) REFERENCES products(product_id));

```
mysql> CREATE TABLE order_items (
    ->     order_item_id INT PRIMARY KEY,
    ->     order_id INT,
    ->     product_id INT,
    ->     quantity INT,
    ->     item_amount DECIMAL(10, 2),
    ->     FOREIGN KEY (order_id) REFERENCES orders(order_id),
    ->     FOREIGN KEY (product_id) REFERENCES products(product_id)
    -> );
Query OK, 0 rows affected (0.16 sec)
```

INSERTING VALUES :

Customer table:

INSERT INTO customers (customer_id, name, email, password)

VALUES

(1, 'John Doe', 'johndoe@example.com', 'password1'),

(2, 'Jane Smith', 'janesmith@example.com', 'password2'),

(3, 'Robert Johnson', 'robert@example.com', 'password3'),

(4, 'Sarah Brown', 'sarah@example.com', 'password4'),

(5, 'David Lee', 'david@example.com', 'password5'),

(6, 'Laura Hall', 'laura@example.com', 'password6'),

(7, 'Michael Davis', 'michael@example.com', 'password7'),

(8, 'Emma Wilson', 'emma@example.com', 'password8'),

(9, 'William Taylor', 'william@example.com', 'password9'),

(10, 'Olivia Adams', 'olivia@example.com', 'password10');

```
mysql> INSERT INTO customers (customer_id, name, email, password)
    -> VALUES
    ->     (1, 'John Doe', 'johndoe@example.com', 'password1'),
    ->     (2, 'Jane Smith', 'janesmith@example.com', 'password2'),
    ->     (3, 'Robert Johnson', 'robert@example.com', 'password3'),
    ->     (4, 'Sarah Brown', 'sarah@example.com', 'password4'),
    ->     (5, 'David Lee', 'david@example.com', 'password5'),
    ->     (6, 'Laura Hall', 'laura@example.com', 'password6'),
    ->     (7, 'Michael Davis', 'michael@example.com', 'password7'),
    ->     (8, 'Emma Wilson', 'emma@example.com', 'password8'),
    ->     (9, 'William Taylor', 'william@example.com', 'password9'),
    ->     (10, 'Olivia Adams', 'olivia@example.com', 'password10');
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Customers;
+-------------+---------------+----------------------+-----------+
| customer_id | name          | email                | password  |
+-------------+---------------+----------------------+-----------+
|           1 | John Doe      | johndoe@example.com  | password1 |
|           2 | Jane Smith    | janesmith@example.com | password2 |
|           3 | Robert Johnson | robert@example.com   | password3 |
|           4 | Sarah Brown   | sarah@example.com    | password4 |
|           5 | David Lee     | david@example.com    | password5 |
|           6 | Laura Hall    | laura@example.com    | password6 |
|           7 | Michael Davis | michael@example.com  | password7 |
|           8 | Emma Wilson   | emma@example.com     | password8 |
|           9 | William Taylor | william@example.com | password9 |
|          10 | Olivia Adams  | olivia@example.com   | password10 |
+-------------+---------------+----------------------+-----------+
10 rows in set (0.01 sec)
```

Product Table:

```
INSERT INTO products (product_id, name, description, price, stockQuantity)
    VALUES
     (1, 'Laptop', 'High-performance laptop', 800.00, 10),
      (2, 'Smartphone', 'Latest smartphone', 600.00, 15),
      (3, 'Tablet', 'Portable tablet', 300.00, 20),
      (4, 'Headphones', 'Noise-canceling', 150.00, 30),
      (5, 'TV', '4K Smart TV', 900.00, 5),
       (6, 'Coffee Maker', 'Automatic coffee maker', 50.00, 25),
       (7, 'Refrigerator', 'Energy-efficient', 700.00, 10),
       (8, 'Microwave Oven', 'Countertop microwave', 80.00, 15),
       (9, 'Blender', 'High-speed blender', 70.00, 20),
       (10, 'Vacuum Cleaner', 'Bagless vacuum cleaner', 120.00, 10);
```

```
mysql> INSERT INTO products (product_id, name, description, price, stockQuantity)
    -> VALUES
    ->     (1, 'Laptop', 'High-performance laptop', 800.00, 10),
    ->     (2, 'Smartphone', 'Latest smartphone', 600.00, 15),
    ->     (3, 'Tablet', 'Portable tablet', 300.00, 20),
    ->     (4, 'Headphones', 'Noise-canceling', 150.00, 30),
    ->     (5, 'TV', '4K Smart TV', 900.00, 5),
    ->     (6, 'Coffee Maker', 'Automatic coffee maker', 50.00, 25),
    ->     (7, 'Refrigerator', 'Energy-efficient', 700.00, 10),
    ->     (8, 'Microwave Oven', 'Countertop microwave', 80.00, 15),
    ->     (9, 'Blender', 'High-speed blender', 70.00, 20),
    ->     (10, 'Vacuum Cleaner', 'Bagless vacuum cleaner', 120.00, 10);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Products;
+------------+---------------+------------------------+--------+---------------+
| product_id | name          | description            | price  | stockQuantity |
+------------+---------------+------------------------+--------+---------------+
|          1 | Laptop        | High-performance laptop | 800.00 |            10 |
|          2 | Smartphone    | Latest smartphone      | 600.00 |            15 |
|          3 | Tablet        | Portable tablet        | 300.00 |            20 |
|          4 | Headphones    | Noise-canceling        | 150.00 |            30 |
|          5 | TV            | 4K Smart TV            | 900.00 |             5 |
|          6 | Coffee Maker  | Automatic coffee maker |  50.00 |            25 |
|          7 | Refrigerator  | Energy-efficient       | 700.00 |            10 |
|          8 | Microwave Oven | Countertop microwave  |  80.00 |            15 |
|          9 | Blender       | High-speed blender     |  70.00 |            20 |
|         10 | Vacuum Cleaner | Bagless vacuum cleaner | 120.00 |            10 |
+------------+---------------+------------------------+--------+---------------+
10 rows in set (0.01 sec)
```

Cart Table :

INSERT INTO cart (cart_id, customer_id, product_id, quantity)

VALUES

(1, 1, 1, 2), (2, 1, 3, 1),

(3, 2, 2, 3), (4, 3, 4, 4),

(5, 3, 5, 2),(6, 4, 6, 1),

(7, 5, 1, 1), (8, 6, 10, 2),

(9, 6, 9, 3), (10, 7, 7, 2);

```
mysql> INSERT INTO cart (cart_id, customer_id, product_id, quantity)
    -> VALUES
    ->     (1, 1, 1, 2),
    ->     (2, 1, 3, 1),
    ->     (3, 2, 2, 3),
    ->     (4, 3, 4, 4),
    ->     (5, 3, 5, 2),
    ->     (6, 4, 6, 1),
    ->     (7, 5, 1, 1),
    ->     (8, 6, 10, 2),
    ->     (9, 6, 9, 3),
    ->     (10, 7, 7, 2);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Cart;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       1 |           1 |          1 |        2 |
|       2 |           1 |          3 |        1 |
|       3 |           2 |          2 |        3 |
|       4 |           3 |          4 |        4 |
|       5 |           3 |          5 |        2 |
|       6 |           4 |          6 |        1 |
|       7 |           5 |          1 |        1 |
|       8 |           6 |         10 |        2 |
|       9 |           6 |          9 |        3 |
|      10 |           7 |          7 |        2 |
+---------+-------------+------------+----------+
10 rows in set (0.01 sec)
```

Order Table:

```
INSERT INTO orders (order_id, customer_id, order_date, total_price, shipping_address)
VALUES
    (1, 1, '2023-01-05', 1200.00, '123 Main St, City'),
    (2, 2, '2023-02-10', 900.00, '456 Elm St, Town'),
    (3, 3, '2023-03-15', 300.00, '789 Oak St, Village'),
    (4, 4, '2023-04-20', 150.00, '101 Pine St, Suburb'),
    (5, 5, '2023-05-25', 1800.00, '234 Cedar St, District'),
    (6, 6, '2023-06-30', 400.00, '567 Birch St, County'),
    (7, 7, '2023-07-05', 700.00, '890 Maple St, State'),
    (8, 8, '2023-08-10', 160.00, '321 Redwood St, Country'),
    (9, 9, '2023-09-15', 140.00, '432 Spruce St, Province'),
    (10, 10, '2023-10-20', 1400.00, '765 Fir St, Territory');
```

```
mysql> INSERT INTO orders (order_id, customer_id, order_date, total_price, shipping_address)
    -> VALUES
    ->     (1, 1, '2023-01-05', 1200.00, '123 Main St, City'),
    ->     (2, 2, '2023-02-10', 900.00, '456 Elm St, Town'),
    ->     (3, 3, '2023-03-15', 300.00, '789 Oak St, Village'),
    ->     (4, 4, '2023-04-20', 150.00, '101 Pine St, Suburb'),
    ->     (5, 5, '2023-05-25', 1800.00, '234 Cedar St, District'),
    ->     (6, 6, '2023-06-30', 400.00, '567 Birch St, County'),
    ->     (7, 7, '2023-07-05', 700.00, '890 Maple St, State'),
    ->     (8, 8, '2023-08-10', 160.00, '321 Redwood St, Country'),
    ->     (9, 9, '2023-09-15', 140.00, '432 Spruce St, Province'),
    ->     (10, 10, '2023-10-20', 1400.00, '765 Fir St, Territory');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Orders;
+----------+-------------+------------+-------------+------------------------+
| order_id | customer_id | order_date | total_price | shipping_address       |
+----------+-------------+------------+-------------+------------------------+
|        1 |           1 | 2023-01-05 |     1200.00 | 123 Main St, City      |
|        2 |           2 | 2023-02-10 |      900.00 | 456 Elm St, Town       |
|        3 |           3 | 2023-03-15 |      300.00 | 789 Oak St, Village    |
|        4 |           4 | 2023-04-20 |      150.00 | 101 Pine St, Suburb    |
|        5 |           5 | 2023-05-25 |     1800.00 | 234 Cedar St, District |
|        6 |           6 | 2023-06-30 |      400.00 | 567 Birch St, County   |
|        7 |           7 | 2023-07-05 |      700.00 | 890 Maple St, State    |
|        8 |           8 | 2023-08-10 |      160.00 | 321 Redwood St, Country |
|        9 |           9 | 2023-09-15 |      140.00 | 432 Spruce St, Province |
|       10 |          10 | 2023-10-20 |     1400.00 | 765 Fir St, Territory  |
+----------+-------------+------------+-------------+------------------------+
10 rows in set (0.02 sec)
```

Order Item Table:

INSERT INTO order_items (order_item_id, order_id, product_id, quantity, item_amount) VALUES

(1, 1, 1, 2, 1600.00),

(2, 1, 3, 1, 300.00),

(3, 2, 2, 3, 1800.00),

(4, 3, 5, 2, 1800.00),

(5, 4, 4, 4, 600.00),

(6, 4, 6, 1, 50.00),

(7, 5, 1, 1, 800.00),

(8, 5, 2, 2, 1200.00),

(9, 6, 10, 2, 240.00),

(10, 6, 9, 3, 210.00);

```
mysql> INSERT INTO order_items (order_item_id, order_id, product_id, quantity, item_amount)
    -> VALUES
    ->     (1, 1, 1, 2, 1600.00),
    ->     (2, 1, 3, 1, 300.00),
    ->     (3, 2, 2, 3, 1800.00),
    ->     (4, 3, 5, 2, 1800.00),
    ->     (5, 4, 4, 4, 600.00),
    ->     (6, 4, 6, 1, 50.00),
    ->     (7, 5, 1, 1, 800.00),
    ->     (8, 5, 2, 2, 1200.00),
    ->     (9, 6, 10, 2, 240.00),
    ->     (10, 6, 9, 3, 210.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Order_items;
+---------------+----------+------------+----------+-------------+
| order_item_id | order_id | product_id | quantity | item_amount |
+---------------+----------+------------+----------+-------------+
|             1 |        1 |          1 |        2 |     1600.00 |
|             2 |        1 |          3 |        1 |      300.00 |
|             3 |        2 |          2 |        3 |     1800.00 |
|             4 |        3 |          5 |        2 |     1800.00 |
|             5 |        4 |          4 |        4 |      600.00 |
|             6 |        4 |          6 |        1 |       50.00 |
|             7 |        5 |          1 |        1 |      800.00 |
|             8 |        5 |          2 |        2 |     1200.00 |
|             9 |        6 |         10 |        2 |      240.00 |
|            10 |        6 |          9 |        3 |      210.00 |
+---------------+----------+------------+----------+-------------+
10 rows in set (0.00 sec)
```

UPDATE:

```
UPDATE customers
   SET address = CASE customer_id
     WHEN 1 THEN '123 Main St, City'
     WHEN 2 THEN '456 Elm St, Town'
     WHEN 3 THEN '789 Oak St, Village'
     WHEN 4 THEN '101 Pine St, Suburb'
     WHEN 5 THEN '234 Cedar St, District'
     WHEN 6 THEN '567 Birch St, County'
      WHEN 7 THEN '890 Maple St, State'
      WHEN 8 THEN '321 Redwood St, Country'
      WHEN 9 THEN '432 Spruce St, Province'
      WHEN 10 THEN '765 Fir St, Territory'
   END
  WHERE customer_id IN (1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
```

```
mysql> select * from Customers;
+-------------+---------------+----------------------+-----------+-------------------------+
| customer_id | name          | email                | password  | address                 |
+-------------+---------------+----------------------+-----------+-------------------------+
|           1 | John Doe      | johndoe@example.com  | password1 | 123 Main St, City       |
|           2 | Jane Smith    | janesmith@example.com| password2 | 456 Elm St, Town        |
|           3 | Robert Johnson| robert@example.com   | password3 | 789 Oak St, Village     |
|           4 | Sarah Brown   | sarah@example.com    | password4 | 101 Pine St, Suburb     |
|           5 | David Lee     | david@example.com    | password5 | 234 Cedar St, District  |
|           6 | Laura Hall    | laura@example.com    | password6 | 567 Birch St, County    |
|           7 | Michael Davis | michael@example.com  | password7 | 890 Maple St, State     |
|           8 | Emma Wilson   | emma@example.com     | password8 | 321 Redwood St, Country |
|           9 | William Taylor| william@example.com  | password9 | 432 Spruce St, Province |
|          10 | Olivia Adams  | olivia@example.com   | password10| 765 Fir St, Territory   |
+-------------+---------------+----------------------+-----------+-------------------------+
10 rows in set (0.00 sec)
```

QUERIES:

1. Update refrigerator product price to 800.

```
UPDATE products
SET price = 800
WHERE product_id = 7;
```

```
mysql> UPDATE products
    -> SET price = 800
    -> WHERE product_id = 7;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Products;
+------------+---------------+-------------------------+--------+---------------+
| product_id | name          | description             | price  | stockQuantity |
+------------+---------------+-------------------------+--------+---------------+
|          1 | Laptop        | High-performance laptop | 800.00 |            10 |
|          2 | Smartphone    | Latest smartphone       | 600.00 |            15 |
|          3 | Tablet        | Portable tablet         | 300.00 |            20 |
|          4 | Headphones    | Noise-canceling         | 150.00 |            30 |
|          5 | TV            | 4K Smart TV             | 900.00 |             5 |
|          6 | Coffee Maker  | Automatic coffee maker  |  50.00 |            25 |
|          7 | Refrigerator  | Energy-efficient        | 800.00 |            10 |
|          8 | Microwave Oven| Countertop microwave    |  80.00 |            15 |
|          9 | Blender       | High-speed blender      |  70.00 |            20 |
|         10 | Vacuum Cleaner| Bagless vacuum cleaner  | 120.00 |            10 |
+------------+---------------+-------------------------+--------+---------------+
10 rows in set (0.00 sec)
```

2. Remove all cart items for a specific customer.

DELETE FROM cart
WHERE customer_id = 1;

```
mysql> DELETE FROM cart
    -> WHERE customer_id = 1;
Query OK, 2 rows affected (0.01 sec)

mysql> select * from Cart;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       3 |           2 |          2 |        3 |
|       4 |           3 |          4 |        4 |
|       5 |           3 |          5 |        2 |
|       6 |           4 |          6 |        1 |
|       7 |           5 |          1 |        1 |
|       8 |           6 |         10 |        2 |
|       9 |           6 |          9 |        3 |
|      10 |           7 |          7 |        2 |
+---------+-------------+------------+----------+
8 rows in set (0.00 sec)
```

3. Retrieve Products Priced Below $100.

SELECT * FROM Products WHERE Price <100;

```
mysql> SELECT * FROM Products WHERE Price <100;
+------------+---------------+-----------------------+-------+---------------+
| product_id | name          | description           | price | stockQuantity |
+------------+---------------+-----------------------+-------+---------------+
|          6 | Coffee Maker  | Automatic coffee maker| 50.00 |            25 |
|          8 | Microwave Oven| Countertop microwave  | 80.00 |            15 |
|          9 | Blender       | High-speed blender    | 70.00 |            20 |
+------------+---------------+-----------------------+-------+---------------+
3 rows in set (0.01 sec)
```

4. Find Products with Stock Quantity Greater Than 5.

SELECT * FROM Products WHERE StockQuantity >5;

```
mysql> SELECT * FROM Products WHERE StockQuantity >5;
+------------+---------------+-----------------------+--------+---------------+
| product_id | name          | description           | price  | stockQuantity |
+------------+---------------+-----------------------+--------+---------------+
|          1 | Laptop        | High-performance laptop| 800.00 |            10 |
|          2 | Smartphone    | Latest smartphone     | 600.00 |            15 |
|          3 | Tablet        | Portable tablet       | 300.00 |            20 |
|          4 | Headphones    | Noise-canceling       | 150.00 |            30 |
|          6 | Coffee Maker  | Automatic coffee maker|  50.00 |            25 |
|          7 | Refrigerator  | Energy-efficient      | 800.00 |            10 |
|          8 | Microwave Oven| Countertop microwave  |  80.00 |            15 |
|          9 | Blender       | High-speed blender    |  70.00 |            20 |
|         10 | Vacuum Cleaner| Bagless vacuum cleaner| 120.00 |            10 |
+------------+---------------+-----------------------+--------+---------------+
9 rows in set (0.00 sec)
```

5. Retrieve Orders with Total Amount Between $500 and $1000.

SELECT * FROM Orders WHERE total_price BETWEEN 500 AND 1000;

```
mysql> SELECT * FROM Orders WHERE total_price BETWEEN 500 AND 1000;
+----------+-------------+------------+-------------+--------------------+
| order_id | customer_id | order_date | total_price | shipping_address   |
+----------+-------------+------------+-------------+--------------------+
|        2 |           2 | 2023-02-10 |      900.00 | 456 Elm St, Town   |
|        7 |           7 | 2023-07-05 |      700.00 | 890 Maple St, State|
+----------+-------------+------------+-------------+--------------------+
2 rows in set (0.00 sec)
```

6. Find Products which name end with letter ‘r’.

SELECT * FROM Products WHERE name LIKE ’%r’;

```
mysql> SELECT * FROM Products WHERE name LIKE '%r';
+------------+----------------+-------------------------+--------+---------------+
| product_id | name           | description             | price  | stockQuantity |
+------------+----------------+-------------------------+--------+---------------+
|          6 | Coffee Maker   | Automatic coffee maker |  50.00 |            25 |
|          7 | Refrigerator   | Energy-efficient        | 800.00 |            10 |
|          9 | Blender        | High-speed blender      |  70.00 |            20 |
|         10 | Vacuum Cleaner | Bagless vacuum cleaner | 120.00 |            10 |
+------------+----------------+-------------------------+--------+---------------+
4 rows in set (0.00 sec)
```

7. Retrieve Cart Items for Customer 5.

SELECT * FROM Cart WHERE Customer_id=5;

```
mysql> SELECT * FROM Cart WHERE Customer_id=5;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       7 |           5 |          1 |        1 |
+---------+-------------+------------+----------+
1 row in set (0.00 sec)
```

8. Find Customers Who Placed Orders in 2023.

SELECT * FROM customers WHERE customer_id IN (SELECT customer_id FROM orders WHERE EXTRACT(YEAR FROM order_date) = 2023);

```
mysql>  SELECT * FROM customers
    ->  WHERE customer_id IN (SELECT customer_id FROM orders WHERE EXTRACT(YEAR FROM order_date) = 2023);
+-------------+---------------+----------------------+-----------+-------------------------+
| customer_id | name          | email                | password  | address                 |
+-------------+---------------+----------------------+-----------+-------------------------+
|           1 | John Doe      | johndoe@example.com  | password1 | 123 Main St, City       |
|           2 | Jane Smith    | janesmith@example.com| password2 | 456 Elm St, Town        |
|           3 | Robert Johnson| robert@example.com   | password3 | 789 Oak St, Village     |
|           4 | Sarah Brown   | sarah@example.com    | password4 | 101 Pine St, Suburb     |
|           5 | David Lee     | david@example.com    | password5 | 234 Cedar St, District  |
|           6 | Laura Hall    | laura@example.com    | password6 | 567 Birch St, County    |
|           7 | Michael Davis | michael@example.com  | password7 | 890 Maple St, State     |
|           8 | Emma Wilson   | emma@example.com     | password8 | 321 Redwood St, Country |
|           9 | William Taylor| william@example.com  | password9 | 432 Spruce St, Province |
|          10 | Olivia Adams  | olivia@example.com   | password10| 765 Fir St, Territory   |
+-------------+---------------+----------------------+-----------+-------------------------+
10 rows in set (0.01 sec)
```

9. Determine the Minimum Stock Quantity for Each Product Category.

SELECT MIN(stockQuantity) AS min_stock,name FROM Products
GROUP BY name;

```
mysql> SELECT MIN(stockQuantity) AS min_stock,name FROM Products GROUP BY name;
+-----------+----------------+
| min_stock | name           |
+-----------+----------------+
|        10 | Laptop         |
|        15 | Smartphone     |
|        20 | Tablet         |
|        30 | Headphones     |
|         5 | TV             |
|        25 | Coffee Maker   |
|        10 | Refrigerator   |
|        15 | Microwave Oven |
|        20 | Blender        |
|        10 | Vacuum Cleaner |
+-----------+----------------+
10 rows in set (0.01 sec)
```

10. Calculate the Total Amount Spent by Each Customer.

SELECT c.customer_id, c.name, SUM(o.total_price) AS total_spent
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;

```
mysql> SELECT c.customer_id, c.name, SUM(o.total_price) AS total_spent
    -> FROM customers c
    -> JOIN orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.customer_id, c.name;
+-------------+----------------+-------------+
| customer_id | name           | total_spent |
+-------------+----------------+-------------+
|           1 | John Doe       |     1200.00 |
|           2 | Jane Smith     |      900.00 |
|           3 | Robert Johnson |      300.00 |
|           4 | Sarah Brown    |      150.00 |
|           5 | David Lee      |     1800.00 |
|           6 | Laura Hall     |      400.00 |
|           7 | Michael Davis  |      700.00 |
|           8 | Emma Wilson    |      160.00 |
|           9 | William Taylor |      140.00 |
|          10 | Olivia Adams   |     1400.00 |
+-------------+----------------+-------------+
10 rows in set (0.00 sec)
```

11. Find the Average Order Amount for Each Customer.

SELECT customer_id, AVG(total_price) AS avg_order_amount
FROM orders

GROUP BY customer_id;

```
mysql> SELECT customer_id, AVG(total_price) AS avg_order_amount
    -> FROM orders
    -> GROUP BY customer_id;
+-------------+------------------+
| customer_id | avg_order_amount |
+-------------+------------------+
|           1 |      1200.000000 |
|           2 |       900.000000 |
|           3 |       300.000000 |
|           4 |       150.000000 |
|           5 |      1800.000000 |
|           6 |       400.000000 |
|           7 |       700.000000 |
|           8 |       160.000000 |
|           9 |       140.000000 |
|          10 |      1400.000000 |
+-------------+------------------+
10 rows in set (0.00 sec)
```

12. Count the Number of Orders Placed by Each Customer.

SELECT customer_id, COUNT(order_id) AS num_orders
FROM orders
GROUP BY customer_id;

```
mysql> SELECT customer_id, COUNT(order_id) AS num_orders
    -> FROM orders
    -> GROUP BY customer_id;
+-------------+------------+
| customer_id | num_orders |
+-------------+------------+
|           1 |          1 |
|           2 |          1 |
|           3 |          1 |
|           4 |          1 |
|           5 |          1 |
|           6 |          1 |
|           7 |          1 |
|           8 |          1 |
|           9 |          1 |
|          10 |          1 |
+-------------+------------+
10 rows in set (0.00 sec)
```

13. Find the Maximum Order Amount for Each Customer.

SELECT customer_id, MAX(total_price) AS max_order_amount FROM orders
GROUP BY customer_id;

```
mysql> SELECT customer_id, MAX(total_price) AS max_order_amount
    -> FROM orders
    -> GROUP BY customer_id;
+-------------+------------------+
| customer_id | max_order_amount |
+-------------+------------------+
|           1 |          1200.00 |
|           2 |           900.00 |
|           3 |           300.00 |
|           4 |           150.00 |
|           5 |          1800.00 |
|           6 |           400.00 |
|           7 |           700.00 |
|           8 |           160.00 |
|           9 |           140.00 |
|          10 |          1400.00 |
+-------------+------------------+
10 rows in set (0.00 sec)
```

14. Get Customers Who Placed Orders Totaling Over $1000.

```
SELECT c.*
FROM customers c
JOIN ( SELECT customer_id, SUM(total_price) AS total_spent
   FROM orders
   GROUP BY customer_id
   HAVING SUM(total_price) > 1000) o ON c.customer_id = o.customer_id;
```

```
mysql> SELECT *
    -> FROM customers c
    -> JOIN (
    ->     SELECT customer_id, SUM(total_price) AS total_spent
    ->     FROM orders
    ->     GROUP BY customer_id
    ->     HAVING SUM(total_price) > 1000
    -> ) o ON c.customer_id = o.customer_id;
+-------------+-------------+--------------------+-----------+-----------------------+-------------+-------------+
| customer_id | name        | email              | password  | address               | customer_id | total_spent |
+-------------+-------------+--------------------+-----------+-----------------------+-------------+-------------+
|           1 | John Doe    | johndoe@example.com | password1  | 123 Main St, City     |           1 |     1200.00 |
|           5 | David Lee   | david@example.com   | password5  | 234 Cedar St, District |           5 |     1800.00 |
|          10 | Olivia Adams | olivia@example.com  | password10 | 765 Fir St, Territory |          10 |     1400.00 |
+-------------+-------------+--------------------+-----------+-----------------------+-------------+-------------+
3 rows in set (0.00 sec)
```

15. Subquery to Find Products Not in the Cart**.**

```
SELECT *FROM products
WHERE product_id NOT IN (SELECT product_id FROM cart);
```

```
mysql> SELECT *FROM products
    -> WHERE product_id NOT IN (SELECT product_id FROM cart);
+------------+---------------+----------------------+--------+---------------+
| product_id | name          | description          | price  | stockQuantity |
+------------+---------------+----------------------+--------+---------------+
|          3 | Tablet        | Portable tablet      | 300.00 |            20 |
|          8 | Microwave Oven | Countertop microwave |  80.00 |            15 |
+------------+---------------+----------------------+--------+---------------+
2 rows in set (0.00 sec)
```

16. Subquery to Find Customers Who Haven't Placed Orders.

```
SELECT * FROM customers
WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM orders);
```

```
mysql> SELECT * FROM customers
    -> WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM orders);
Empty set (0.01 sec)
```

17. Subquery to Calculate the Percentage of Total Revenue for a Product.

```
SELECT p.product_id, p.name, (SUM(oi.item_amount) / (SELECT SUM(total_price) FROM orders)) * 100 AS revenue_percentage
FROM products p
JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY p.product_id, p.name;
```

```
mysql> SELECT p.product_id, p.name, (SUM(oi.item_amount) / (SELECT SUM(total_price) FROM orders)) * 100 AS revenue_percentage
    -> FROM products p
    -> JOIN order_items oi ON p.product_id = oi.product_id
    -> GROUP BY p.product_id, p.name;
+------------+---------------+--------------------+
| product_id | name          | revenue_percentage |
+------------+---------------+--------------------+
|          1 | Laptop        |          33.566434 |
|          2 | Smartphone    |          41.958042 |
|          3 | Tablet        |           4.195804 |
|          4 | Headphones    |           8.391608 |
|          5 | TV            |          25.174825 |
|          6 | Coffee Maker  |           0.699301 |
|          9 | Blender       |           2.937063 |
|         10 | Vacuum Cleaner |           3.356643 |
+------------+---------------+--------------------+
8 rows in set (0.01 sec)
```

18. Subquery to Find Products with Low Stock.

```
SELECT *
FROM products
WHERE stockQuantity < (SELECT AVG(stockQuantity) FROM products);
```

```
mysql> SELECT *
    -> FROM products
    -> WHERE stockQuantity < (SELECT AVG(stockQuantity) FROM products);
+------------+---------------+--------------------------+--------+---------------+
| product_id | name          | description              | price  | stockQuantity |
+------------+---------------+--------------------------+--------+---------------+
|          1 | Laptop        | High-performance laptop | 800.00 |            10 |
|          2 | Smartphone    | Latest smartphone        | 600.00 |            15 |
|          5 | TV            | 4K Smart TV              | 900.00 |             5 |
|          7 | Refrigerator  | Energy-efficient         | 800.00 |            10 |
|          8 | Microwave Oven | Countertop microwave     |  80.00 |            15 |
|         10 | Vacuum Cleaner | Bagless vacuum cleaner   | 120.00 |            10 |
+------------+---------------+--------------------------+--------+---------------+
6 rows in set (0.00 sec)
```

19. Subquery to Find Customers Who Placed High-Value Orders.

```
SELECT * FROM customers
WHERE customer_id IN (
   SELECT customer_id
   FROM orders
   GROUP BY customer_id
   HAVING SUM(total_price) > 1000 );
```

```
mysql> SELECT * FROM customers
    -> WHERE customer_id IN (
    ->     SELECT customer_id
    ->     FROM orders
    ->     GROUP BY customer_id
    ->     HAVING SUM(total_price) > 1000 );
+-------------+-------------+--------------------+-----------+-----------------------+
| customer_id | name        | email              | password  | address               |
+-------------+-------------+--------------------+-----------+-----------------------+
|           1 | John Doe    | johndoe@example.com | password1  | 123 Main St, City      |
|           5 | David Lee   | david@example.com   | password5  | 234 Cedar St, District |
|          10 | Olivia Adams | olivia@example.com  | password10 | 765 Fir St, Territory  |
+-------------+-------------+--------------------+-----------+-----------------------+
3 rows in set (0.00 sec)
```