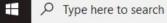```
MySQL 8.0 Command Line Client   —   □   ×

mysql>
mysql>
mysql> /*CODING CHALLENGE-6 Ecommerce - SQL*/
mysql>
mysql> CREATE DATABASE IF NOT EXISTS Ecommerce;
Query OK, 1 row affected (0.01 sec)

mysql> USE ecommerce;
Database changed
mysql> CREATE TABLE customers (
    ->   customer_id INT PRIMARY KEY,
    ->   name VARCHAR(255),
    ->   email VARCHAR(255),
    ->   password VARCHAR(255)
    ->);
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE products (
    ->   product_id INT PRIMARY KEY,
    ->   name VARCHAR(255),
    ->   price DECIMAL(10, 2),
    ->   description TEXT,
    ->   stockQuantity INT
    ->);
Query OK, 0 rows affected (0.05 sec)

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 (0.07 sec)

mysql> CREATE TABLE orders (
    ->   order_id INT PRIMARY KEY,
    ->   customer_id INT,
    ->   order_date DATE,
    ->   total_price DECIMAL(10, 2),
    ->   shipping_address TEXT,
    ->   FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
    ->);
```

```
mysql> CREATE TABLE order_items (
    ->     order_item_id INT PRIMARY KEY,
    ->     order_id INT,
    ->     product_id INT,
    ->     quantity INT,
    ->     FOREIGN KEY (order_id) REFERENCES orders(order_id),
    ->     FOREIGN KEY (product_id) REFERENCES products(product_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

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 (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO customers (customer_id, name, email, password)
    -> VALUES
    -> (1, 'John Doe', 'johndoe@example.com', '123 Main St, City'),
    -> (2, 'Jane Smith', 'janesmith@example.com', '456 Elm St, Town'),
    -> (3, 'Robert Johnson', 'robert@example.com', '789 Oak St, Village'),
    -> (4, 'Sarah Brown', 'sarah@example.com', '101 Pine St, Suburb'),
    -> (5, 'David Lee', 'david@example.com', '234 Cedar St, District'),
    -> (6, 'Laura Hall', 'laura@example.com', '567 Birch St, County'),
    -> (7, 'Michael Davis', 'michael@example.com', '890 Maple St, State'),
    -> (8, 'Emma Wilson', 'emma@example.com', '321 Redwood St, Country'),
    -> (9, 'William Taylor', 'william@example.com', '432 Spruce St, Province'),
    -> (10, 'Olivia Adams', 'olivia@example.com', '765 Fir St, Territory');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

```
MySQL 8.0 Command Line Client                                                    —   □   ×

Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO orders (order_id, customer_id, order_date, total_price, shipping_address)
    -> VALUES
    -> (1, 1, '2023-01-05', 1200.00, 'Shipping Address 1'),
    -> (2, 2, '2023-02-10', 900.00, 'Shipping Address 2'),
    -> (3, 3, '2023-03-15', 300.00, 'Shipping Address 3'),
    -> (4, 4, '2023-04-20', 150.00, 'Shipping Address 4'),
    -> (5, 5, '2023-05-25', 1800.00, 'Shipping Address 5'),
    -> (6, 6, '2023-06-30', 400.00, 'Shipping Address 6'),
    -> (7, 7, '2023-07-05', 700.00, 'Shipping Address 7'),
    -> (8, 8, '2023-08-10', 160.00, 'Shipping Address 8'),
    -> (9, 9, '2023-09-15', 140.00, 'Shipping Address 9'),
    -> (10, 10, '2023-10-20', 1400.00, 'Shipping Address 10');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO order_items (order_item_id, order_id, product_id, quantity, itemAmount)
    -> 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);
ERROR 1054 (42S22): Unknown column 'itemAmount' in 'field list'
mysql> ALTER TABLE
    -> ADD COLUMN itemAmount int;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'ADD COLUMN it
emAmount int' at line 2
mysql> ALTER TABLE order_items
    -> ADD COLUMN itemAmount DECIMAL(10, 2);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

```
MySQL 8.0 Command Line Client                                                    —    ⟋    ✕

mysql> INSERT INTO order_items (order_item_id, order_id, product_id, quantity, itemAmount)
    -> 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 (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

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> /*1. Update refrigerator product price to 800*/
mysql> UPDATE products
    -> SET price = 800
    -> WHERE product_id = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> /*2. Remove all cart items for a specific customer*/
mysql> DELETE FROM cart
    -> WHERE customer_id = 2;
Query OK, 1 row affected (0.01 sec)

mysql>
```

```
mysql> /*3. Retrieve Products Priced Below $100*/
mysql> SELECT *
    -> FROM products
    -> WHERE price < 100;
+------------+---------------+-------+-----------------------+---------------+
| product_id | name          | price | description           | stockQuantity |
+------------+---------------+-------+-----------------------+---------------+
|          6 | Coffee Maker  | 50.00 | Automatic coffee maker |           25 |
|          8 | Microwave Oven | 80.00 | Countertop microwave  |           15 |
|          9 | Blender       | 70.00 | High-speed blender    |           20 |
+------------+---------------+-------+-----------------------+---------------+
3 rows in set (0.00 sec)

mysql> /*4. Find Products with Stock Quantity Greater Than 5*/
mysql> SELECT *
    -> FROM products
    -> WHERE stockQuantity > 5;
+------------+---------------+--------+-----------------------+---------------+
| product_id | name          | price  | description           | stockQuantity |
+------------+---------------+--------+-----------------------+---------------+
|          1 | Laptop        | 800.00 | High-performance laptop |         10 |
|          2 | Smartphone    | 800.00 | Latest smartphone     |           15 |
|          3 | Tablet        | 300.00 | Portable tablet       |           20 |
|          4 | Headphones    | 150.00 | Noise-canceling       |           30 |
|          6 | Coffee Maker  |  50.00 | Automatic coffee maker |          25 |
|          7 | Refrigerator  | 700.00 | Energy-efficient      |           10 |
|          8 | Microwave Oven |  80.00 | Countertop microwave  |          15 |
|          9 | Blender       |  70.00 | High-speed blender    |           20 |
|         10 | Vacuum Cleaner | 120.00 | Bagless vacuum cleaner |          10 |
+------------+---------------+--------+-----------------------+---------------+
9 rows in set (0.00 sec)

mysql> /*5. Retrieve Orders with Total Amount 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 | Shipping Address 2 |
|        7 |           7 | 2023-07-05 |      700.00 | Shipping Address 7 |
+----------+-------------+------------+-------------+--------------------+
```

```
mysql> /*6. Find Products which name end with letter 'r'*/
mysql> SELECT *
    -> FROM products
    -> WHERE name LIKE '%r';
+------------+-----------------+--------+------------------------+---------------+
| product_id | name            | price  | description            | stockQuantity |
+------------+-----------------+--------+------------------------+---------------+
|          6 | Coffee Maker    |  50.00 | Automatic coffee maker |            25 |
|          7 | Refrigerator    | 700.00 | Energy-efficient       |            10 |
|          9 | Blender         |  70.00 | High-speed blender     |            20 |
|         10 | Vacuum Cleaner  | 120.00 | Bagless vacuum cleaner |            10 |
+------------+-----------------+--------+------------------------+---------------+
4 rows in set (0.00 sec)

mysql> /*7. Retrieve Cart Items for Customer 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)

mysql> /*8. Find Customers Who Placed Orders in 2023*/
mysql> SELECT DISTINCT customers.*
    -> FROM customers
    -> JOIN orders ON customers.customer_id = orders.customer_id
    -> WHERE YEAR(orders.order_date) = 2023;
+-------------+----------------+-----------------------+--------------------------+
| customer_id | name           | email                 | password                 |
+-------------+----------------+-----------------------+--------------------------+
|           1 | John Doe       | johndoe@example.com   | 123 Main St, City        |
|           2 | Jane Smith     | janesmith@example.com | 456 Elm St, Town         |
|           3 | Robert Johnson | robert@example.com    | 789 Oak St, Village      |
|           4 | Sarah Brown    | sarah@example.com     | 101 Pine St, Suburb      |
|           5 | David Lee      | david@example.com     | 234 Cedar St, District   |
|           6 | Laura Hall     | laura@example.com     | 567 Birch St, County     |
|           7 | Michael Davis  | michael@example.com   | 890 Maple St, State      |
|           8 | Emma Wilson    | emma@example.com      | 321 Redwood St, Country  |
|           9 | William Taylor | william@example.com   | 432 Spruce St, Province  |
|          10 | Olivia Adams   | olivia@example.com    | 765 Fir St, Territory    |
+-------------+----------------+-----------------------+--------------------------+
```

```
10 rows in set (0.00 sec)

mysql> /*9. Determine the Minimum Stock Quantity for Each Product Category.*/
mysql> SELECT
    ->      categories.category_id,
    ->      categories.name AS category_name,
    ->      MIN(products.stockQuantity) AS min_stock_quantity
    -> FROM categories
    -> JOIN products ON categories.category_id = products.category_id
    -> GROUP BY categories.category_id, categories.name;
ERROR 1146 (42S02): Table 'ecommerce.categories' doesn't exist
mysql>
mysql>
mysql> /*10. Calculate the Total Amount Spent by Each Customer.*/
mysql> SELECT
    ->      customers.customer_id,
    ->      customers.name AS customer_name,
    ->      SUM(orders.total_price) AS total_amount_spent
    -> FROM customers
    -> LEFT JOIN orders ON customers.customer_id = orders.customer_id
    -> GROUP BY customers.customer_id, customers.name;
+-------------+----------------+--------------------+
| customer_id | customer_name  | total_amount_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)

mysql> /*11. Find the Average Order Amount for Each Customer*/
mysql> SELECT
    ->      customers.customer_id,
    ->      customers.name AS customer_name,
    ->      AVG(orders.total_price) AS average_order_amount
    -> FROM customers
    -> LEFT JOIN orders ON customers.customer_id = orders.customer_id
```

```
    ->       customers.customer_id,
    ->       customers.name AS customer_name,
    ->       AVG(orders.total_price) AS average_order_amount
    -> FROM customers
    -> LEFT JOIN orders ON customers.customer_id = orders.customer_id
    -> GROUP BY customers.customer_id, customers.name;
+-------------+-----------------+----------------------+
| customer_id | customer_name   | average_order_amount |
+-------------+-----------------+----------------------+
|           1 | John Doe        |          1200.000000 |
|           2 | Jane Smith      |           900.000000 |
|           3 | Robert Johnson  |           300.000000 |
|           4 | Sarah Brown     |           150.000000 |
|           5 | David Lee       |          1800.000000 |
|           6 | Laura Hall      |           400.000000 |
|           7 | Michael Davis   |           700.000000 |
|           8 | Emma Wilson     |           160.000000 |
|           9 | William Taylor  |           140.000000 |
|          10 | Olivia Adams    |          1400.000000 |
+-------------+-----------------+----------------------+
10 rows in set (0.00 sec)

mysql> /*12. Count the Number of Orders Placed by Each Customer*/
mysql> SELECT
    ->       customers.customer_id,
    ->       customers.name AS customer_name,
    ->       COUNT(orders.order_id) AS order_count
    -> FROM customers
    -> LEFT JOIN orders ON customers.customer_id = orders.customer_id
    -> GROUP BY customers.customer_id, customers.name;
+-------------+-----------------+-------------+
| customer_id | customer_name   | order_count |
+-------------+-----------------+-------------+
|           1 | John Doe        |           1 |
|           2 | Jane Smith      |           1 |
|           3 | Robert Johnson  |           1 |
|           4 | Sarah Brown     |           1 |
|           5 | David Lee       |           1 |
|           6 | Laura Hall      |           1 |
|           7 | Michael Davis   |           1 |
|           8 | Emma Wilson     |           1 |
|           9 | William Taylor  |           1 |
|          10 | Olivia Adams    |           1 |
+-------------+-----------------+-------------+
```

```
mysql> /*13. Find the Maximum Order Amount for Each Customer.*/
mysql> SELECT
    ->     customers.customer_id,
    ->     customers.name AS customer_name,
    ->     MAX(orders.total_price) AS max_order_amount
    -> FROM customers
    -> LEFT JOIN orders ON customers.customer_id = orders.customer_id
    -> GROUP BY customers.customer_id, customers.name;
+-------------+----------------+------------------+
| customer_id | customer_name  | max_order_amount |
+-------------+----------------+------------------+
|           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)

mysql> /*14. Get Customers Who Placed Orders Totaling Over $1000.*/
mysql> SELECT DISTINCT
    ->     customers.customer_id,
    ->     customers.name AS customer_name
    -> FROM customers
    -> JOIN orders ON customers.customer_id = orders.customer_id
    -> WHERE orders.total_price > 1000;
+-------------+----------------+
| customer_id | customer_name  |
+-------------+----------------+
|           1 | John Doe       |
|           5 | David Lee      |
|          10 | Olivia Adams   |
+-------------+----------------+
3 rows in set (0.00 sec)

mysql>
```

```
|           5 | David Lee      |
|          10 | Olivia Adams   |
+-------------+----------------+
3 rows in set (0.00 sec)

mysql> /*15. Subquery to Find Products Not in the Cart*/
mysql> SELECT *
    -> FROM products
    -> WHERE product_id NOT IN (
    ->     SELECT product_id
    ->     FROM cart
    -> );
+------------+---------------+--------+----------------------+---------------+
| product_id | name          | price  | description          | stockQuantity |
+------------+---------------+--------+----------------------+---------------+
|          2 | Smartphone    | 800.00 | Latest smartphone    |            15 |
|          8 | Microwave Oven |  80.00 | Countertop microwave |            15 |
+------------+---------------+--------+----------------------+---------------+
2 rows in set (0.00 sec)

mysql> /*16. Subquery to Find Customers Who Haven't Placed Orders.*/
mysql> SELECT *
    -> FROM customers
    -> WHERE customer_id NOT IN (
    ->     SELECT DISTINCT customer_id
    ->     FROM orders
    -> );
Empty set (0.00 sec)
```

```
mysql>
mysql> /*17. Subquery to Calculate the Percentage of Total Revenue for a Product*/
mysql> SELECT
    ->     products.product_id,
    ->     products.name AS product_name,
    ->     products.price,
    ->     (products.price * order_items.quantity) / (SELECT SUM(total_price) FROM orders) * 100 AS revenue_percentage
    -> FROM order_items
    -> JOIN products ON order_items.product_id = products.product_id;
+------------+---------------+--------+--------------------+
| product_id | product_name  | price  | revenue_percentage |
+------------+---------------+--------+--------------------+
|          1 | Laptop        | 800.00 |          22.377622 |
|          3 | Tablet        | 300.00 |           4.195804 |
|          2 | Smartphone    | 800.00 |          33.566434 |
|          5 | TV            | 900.00 |          25.174825 |
|          4 | Headphones    | 150.00 |           8.391608 |
|          6 | Coffee Maker  |  50.00 |           0.699301 |
|          1 | Laptop        | 800.00 |          11.188811 |
|          2 | Smartphone    | 800.00 |          22.377622 |
|         10 | Vacuum Cleaner| 120.00 |           3.356643 |
|          9 | Blender       |  70.00 |           2.937063 |
+------------+---------------+--------+--------------------+
10 rows in set (0.00 sec)

mysql> /*18. Subquery to Find Products with Low Stock*/
mysql> SELECT *
    -> FROM products
    -> WHERE stockQuantity < (SELECT AVG(stockQuantity) FROM products);
+------------+---------------+--------+-----------------------+---------------+
| product_id | name          | price  | description           | stockQuantity |
+------------+---------------+--------+-----------------------+---------------+
|          1 | Laptop        | 800.00 | High-performance laptop |          10 |
|          2 | Smartphone    | 800.00 | Latest smartphone     |            15 |
|          5 | TV            | 900.00 | 4K Smart TV           |             5 |
|          7 | Refrigerator  | 700.00 | Energy-efficient      |            10 |
|          8 | Microwave Oven|  80.00 | Countertop microwave  |            15 |
|         10 | Vacuum Cleaner| 120.00 | Bagless vacuum cleaner |           10 |
+------------+---------------+--------+-----------------------+---------------+
6 rows in set (0.00 sec)

mysql>
```

```
|     2 | Smartphone     | 800.00 | Latest smartphone        |             15 |
|     5 | TV             | 900.00 | 4K Smart TV              |              5 |
|     7 | Refrigerator   | 700.00 | Energy-efficient         |             10 |
|     8 | Microwave Oven |  80.00 | Countertop microwave     |             15 |
|    10 | Vacuum Cleaner | 120.00 | Bagless vacuum cleaner   |             10 |
+-------------+----------------+--------+--------------------------+----------------+
6 rows in set (0.00 sec)

mysql> /*19. Subquery to Find Customers Who Placed High-Value Orders.*/
mysql>
mysql> SELECT DISTINCT
    ->     customers.customer_id,
    ->     customers.name AS customer_name
    -> FROM customers
    -> JOIN orders ON customers.customer_id = orders.customer_id
    -> WHERE orders.total_price > (SELECT AVG(total_price) FROM orders);
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|           1 | John Doe      |
|           2 | Jane Smith    |
|           5 | David Lee     |
|          10 | Olivia Adams  |
+-------------+---------------+
4 rows in set (0.00 sec)

mysql>
```