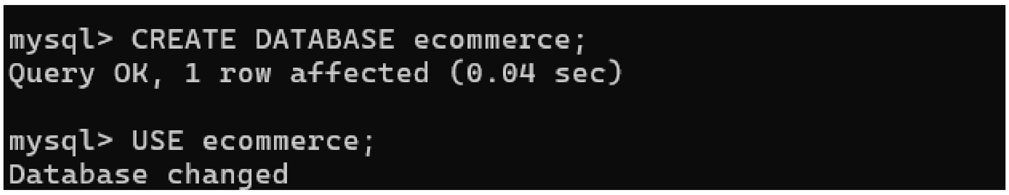
#MySQL Task

-- To create the ecommerce database:

```
CREATE DATABASE ecommerce;
USE ecommerce;
```



```
mysql> CREATE DATABASE ecommerce;
Query OK, 1 row affected (0.04 sec)

mysql> USE ecommerce;
Database changed
```

-- To create the customers table with id, name, email, and address:

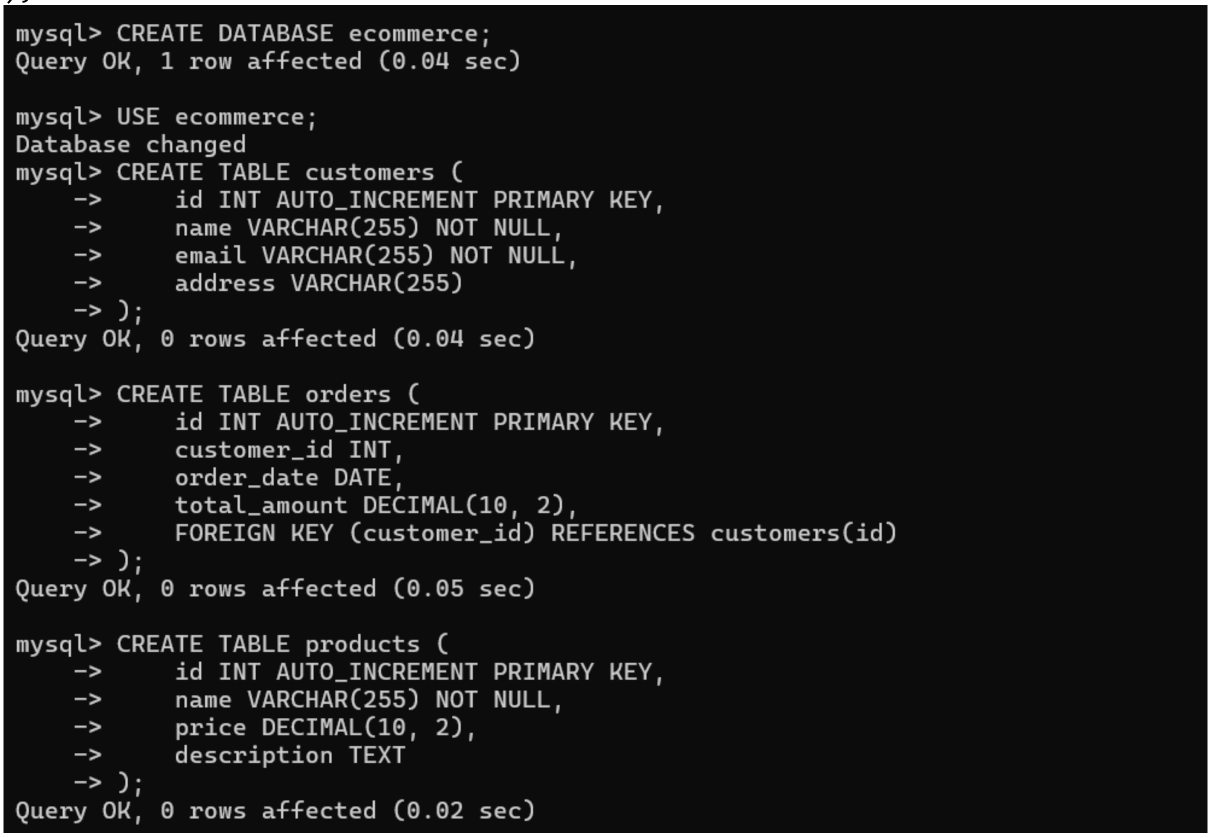
```
CREATE TABLE customers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    address VARCHAR(255)
);
```

-- Create orders table

```
CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10, 2),
    FOREIGN KEY (customer_id) REFERENCES customers(id)
);
```

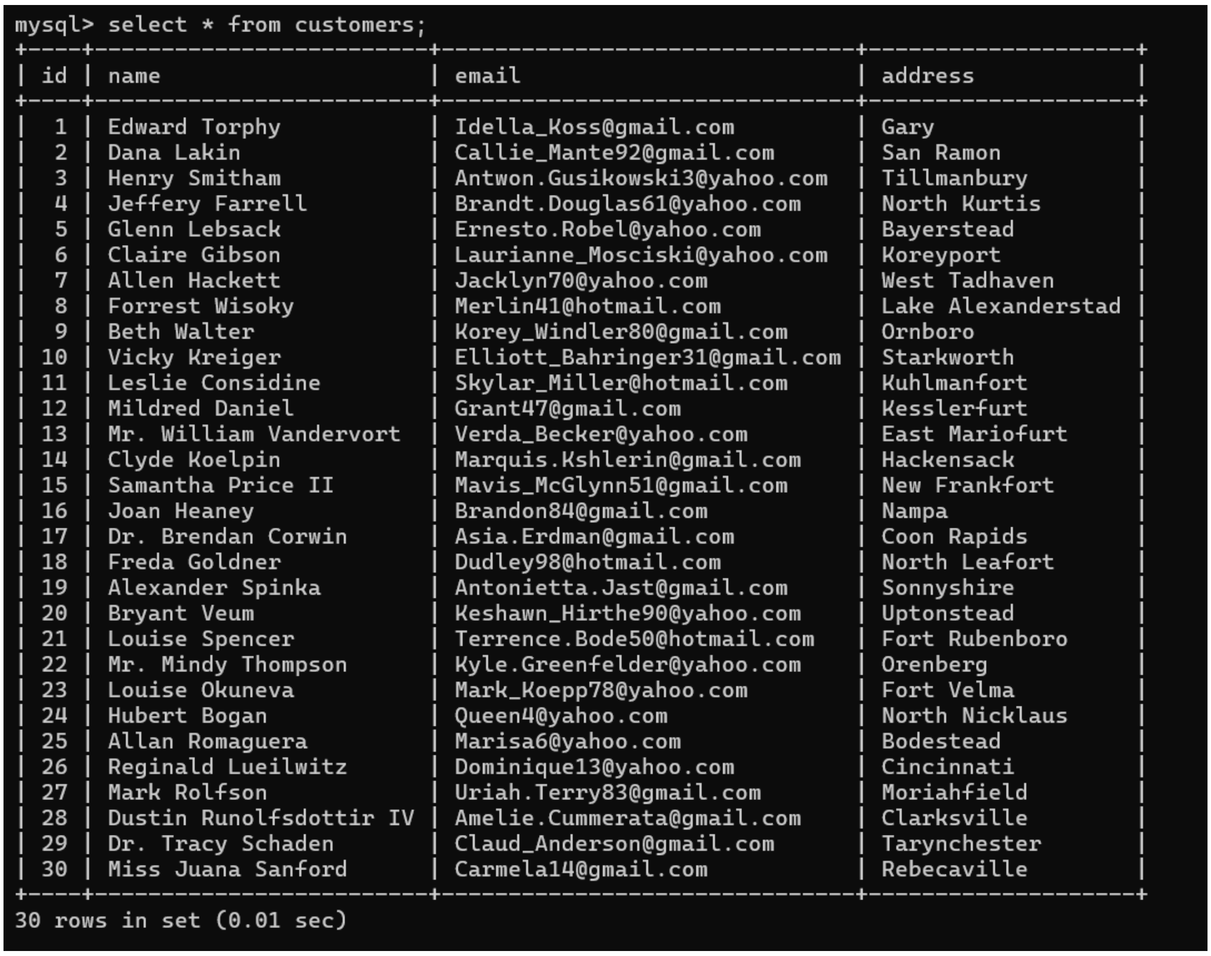
-- Create products table

```
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    price DECIMAL(10, 2),
    description TEXT
);
```



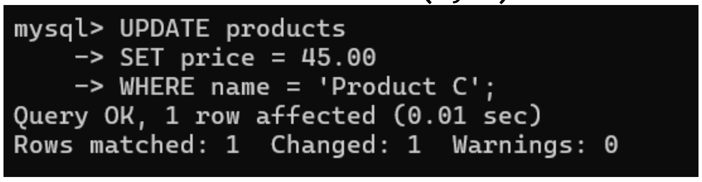
```
mysql> CREATE DATABASE ecommerce;
Query OK, 1 row affected (0.04 sec)

mysql> USE ecommerce;
Database changed
mysql> CREATE TABLE customers (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     email VARCHAR(255) NOT NULL,
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE orders (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     customer_id INT,
    ->     order_date DATE,
    ->     total_amount DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES customers(id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE products (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     price DECIMAL(10, 2),
    ->     description TEXT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
-- Insert sample data into customers table
INSERT INTO customers (name, email, address) VALUES
('Edward Torphy', 'Idella_Koss@gmail.com', 'Gary'),
('Dana Lakin', 'Callie_Mante92@gmail.com', 'San Ramon'),
('Henry Smitham', 'Antwon.Gusikowski3@yahoo.com', 'Tillmanbury'),
('Jeffery Farrell', 'Brandt.Douglas61@yahoo.com', 'North Kurtis'),
('Glenn Lebsack', 'Ernesto.Robel@yahoo.com', 'Bayerstead'),
('Claire Gibson', 'Laurianne_Mosciski@yahoo.com', 'Koreyport'),
('Allen Hackett', 'Jacklyn70@yahoo.com', 'West Tadhaven'),
('Forrest Wisoky', 'Merlin41@hotmail.com', 'Lake Alexanderstad'),
('Beth Walter', 'Korey_Windler80@gmail.com', 'Ornboro'),
('Vicky Kreiger', 'Elliott_Bahringer31@gmail.com', 'Starkworth'),
('Leslie Considine', 'Skylar_Miller@hotmail.com', 'Kuhlmanfort'),
('Mildred Daniel', 'Grant47@gmail.com', 'Kesslerfurt'),
('Mr. William Vandervort', 'Verda_Becker@yahoo.com', 'East Mariofurt'),
('Clyde Koelpin', 'Marquis.Kshlerin@gmail.com', 'Hackensack'),
('Samantha Price II', 'Mavis_McGlynn51@gmail.com', 'New Frankfort'),
('Joan Heaney', 'Brandon84@gmail.com', 'Nampa'),
('Dr. Brendan Corwin', 'Asia.Erdman@gmail.com', 'Coon Rapids'),
('Freda Goldner', 'Dudley98@hotmail.com', 'North Leafort'),
('Alexander Spinka', 'Antonietta.Jast@gmail.com', 'Sonnyshire'),
('Bryant Veum', 'Keshawn_Hirthe90@yahoo.com', 'Uptonstead'),
('Louise Spencer', 'Terrence.Bode50@hotmail.com', 'Fort Rubenboro'),
('Mr. Mindy Thompson', 'Kyle.Greenfelder@yahoo.com', 'Orenberg'),
('Louise Okuneva', 'Mark_Koepp78@yahoo.com', 'Fort Velma'),
('Hubert Bogan', 'Queen4@yahoo.com', 'North Nicklaus'),
('Allan Romaguera', 'Marisa6@yahoo.com', 'Bodestead'),
('Reginald Lueilwitz', 'Dominique13@yahoo.com', 'Cincinnati'),
('Mark Rolfson', 'Uriah.Terry83@gmail.com', 'Moriahfield'),
('Dustin Runolfsdottir IV', 'Amelie.Cummerata@gmail.com', 'Clarksville'),
('Dr. Tracy Schaden', 'Claud_Anderson@gmail.com', 'Tarynchester'),
('Miss Juana Sanford', 'Carmela14@gmail.com', 'Rebecaville');
Select * from customers
```



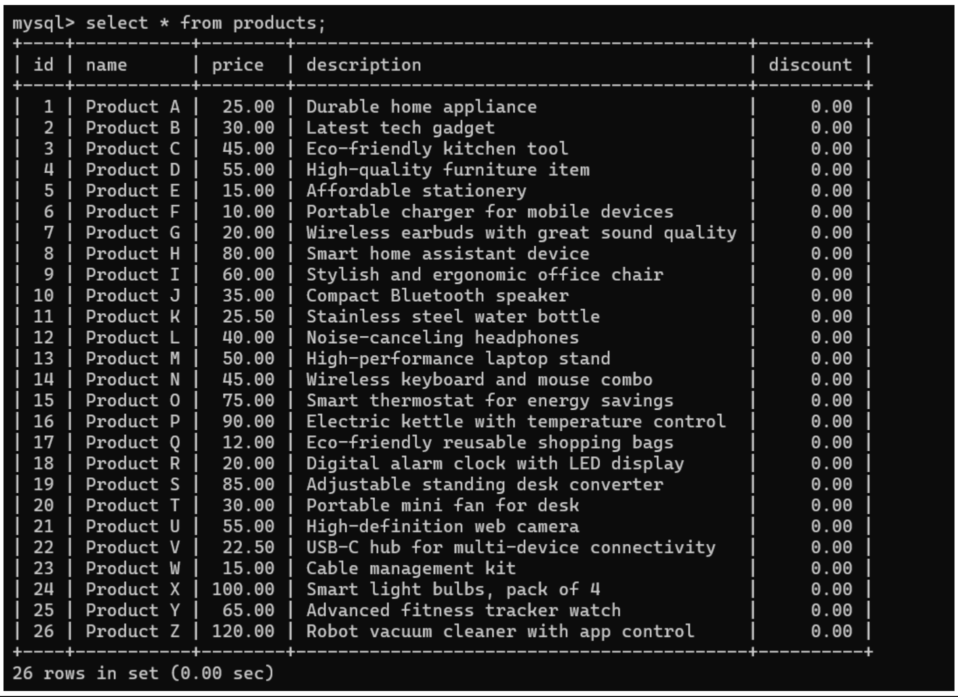
```
mysql> select * from customers;
+----+-------------------------+-------------------------------+--------------------+
| id | name                    | email                         | address            |
+----+-------------------------+-------------------------------+--------------------+
|  1 | Edward Torphy           | Idella_Koss@gmail.com         | Gary               |
|  2 | Dana Lakin              | Callie_Mante92@gmail.com      | San Ramon          |
|  3 | Henry Smitham           | Antwon.Gusikowski3@yahoo.com  | Tillmanbury        |
|  4 | Jeffery Farrell         | Brandt.Douglas61@yahoo.com    | North Kurtis       |
|  5 | Glenn Lebsack           | Ernesto.Robel@yahoo.com       | Bayerstead         |
|  6 | Claire Gibson           | Laurianne_Mosciski@yahoo.com  | Koreyport          |
|  7 | Allen Hackett           | Jacklyn70@yahoo.com           | West Tadhaven      |
|  8 | Forrest Wisoky          | Merlin41@hotmail.com          | Lake Alexanderstad |
|  9 | Beth Walter             | Korey_Windler80@gmail.com     | Ornboro            |
| 10 | Vicky Kreiger           | Elliott_Bahringer31@gmail.com | Starkworth         |
| 11 | Leslie Considine        | Skylar_Miller@hotmail.com     | Kuhlmanfort        |
| 12 | Mildred Daniel          | Grant47@gmail.com             | Kesslerfurt        |
| 13 | Mr. William Vandervort  | Verda_Becker@yahoo.com        | East Mariofurt     |
| 14 | Clyde Koelpin           | Marquis.Kshlerin@gmail.com    | Hackensack         |
| 15 | Samantha Price II       | Mavis_McGlynn51@gmail.com     | New Frankfort      |
| 16 | Joan Heaney             | Brandon84@gmail.com           | Nampa              |
| 17 | Dr. Brendan Corwin      | Asia.Erdman@gmail.com         | Coon Rapids        |
| 18 | Freda Goldner           | Dudley98@hotmail.com          | North Leafort      |
| 19 | Alexander Spinka        | Antonietta.Jast@gmail.com     | Sonnyshire         |
| 20 | Bryant Veum             | Keshawn_Hirthe90@yahoo.com    | Uptonstead         |
| 21 | Louise Spencer          | Terrence.Bode50@hotmail.com   | Fort Rubenboro     |
| 22 | Mr. Mindy Thompson      | Kyle.Greenfelder@yahoo.com    | Orenberg           |
| 23 | Louise Okuneva          | Mark_Koepp78@yahoo.com        | Fort Velma         |
| 24 | Hubert Bogan            | Queen4@yahoo.com              | North Nicklaus     |
| 25 | Allan Romaguera         | Marisa6@yahoo.com             | Bodestead          |
| 26 | Reginald Lueilwitz      | Dominique13@yahoo.com         | Cincinnati         |
| 27 | Mark Rolfson            | Uriah.Terry83@gmail.com       | Moriahfield        |
| 28 | Dustin Runolfsdottir IV | Amelie.Cummerata@gmail.com    | Clarksville        |
| 29 | Dr. Tracy Schaden       | Claud_Anderson@gmail.com      | Tarynchester       |
| 30 | Miss Juana Sanford      | Carmela14@gmail.com           | Rebecaville        |
+----+-------------------------+-------------------------------+--------------------+
30 rows in set (0.01 sec)
```

```
-- Insert sample data into products table (30 products)
INSERT INTO products (name, price, description) VALUES
INSERT INTO products (name, price, description) VALUES
('Product A', 25.00, 'Durable home appliance'),
('Product B', 30.00, 'Latest tech gadget'),
('Product C', 45.50, 'Eco-friendly kitchen tool'),
('Product D', 55.00, 'High-quality furniture item'),
('Product E', 15.00, 'Affordable stationery'),
('Product F', 10.00, 'Portable charger for mobile devices'),
('Product G', 20.00, 'Wireless earbuds with great sound quality'),
('Product H', 80.00, 'Smart home assistant device');
('Product I', 60.00, 'Stylish and ergonomic office chair'),
('Product J', 35.00, 'Compact Bluetooth speaker'),
('Product K', 25.50, 'Stainless steel water bottle'),
('Product L', 40.00, 'Noise-canceling headphones'),
('Product M', 50.00, 'High-performance laptop stand'),
('Product N', 45.00, 'Wireless keyboard and mouse combo'),
('Product O', 75.00, 'Smart thermostat for energy savings'),
('Product P', 90.00, 'Electric kettle with temperature control'),
('Product Q', 12.00, 'Eco-friendly reusable shopping bags'),
('Product R', 20.00, 'Digital alarm clock with LED display'),
('Product S', 85.00, 'Adjustable standing desk converter'),
('Product T', 30.00, 'Portable mini fan for desk'),
('Product U', 55.00, 'High-definition web camera'),
('Product V', 22.50, 'USB-C hub for multi-device connectivity'),
('Product W', 15.00, 'Cable management kit'),
('Product X', 100.00, 'Smart light bulbs, pack of 4'),
('Product Y', 65.00, 'Advanced fitness tracker watch'),
('Product Z', 120.00, 'Robot vacuum cleaner with app control');
-- 3. Update the price of Product C to 45.00.
UPDATE products
SET price = 45.00
WHERE name = 'Product C';
-- 4. Add a new column `discount` to the products table.
ALTER TABLE products
ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00;
```



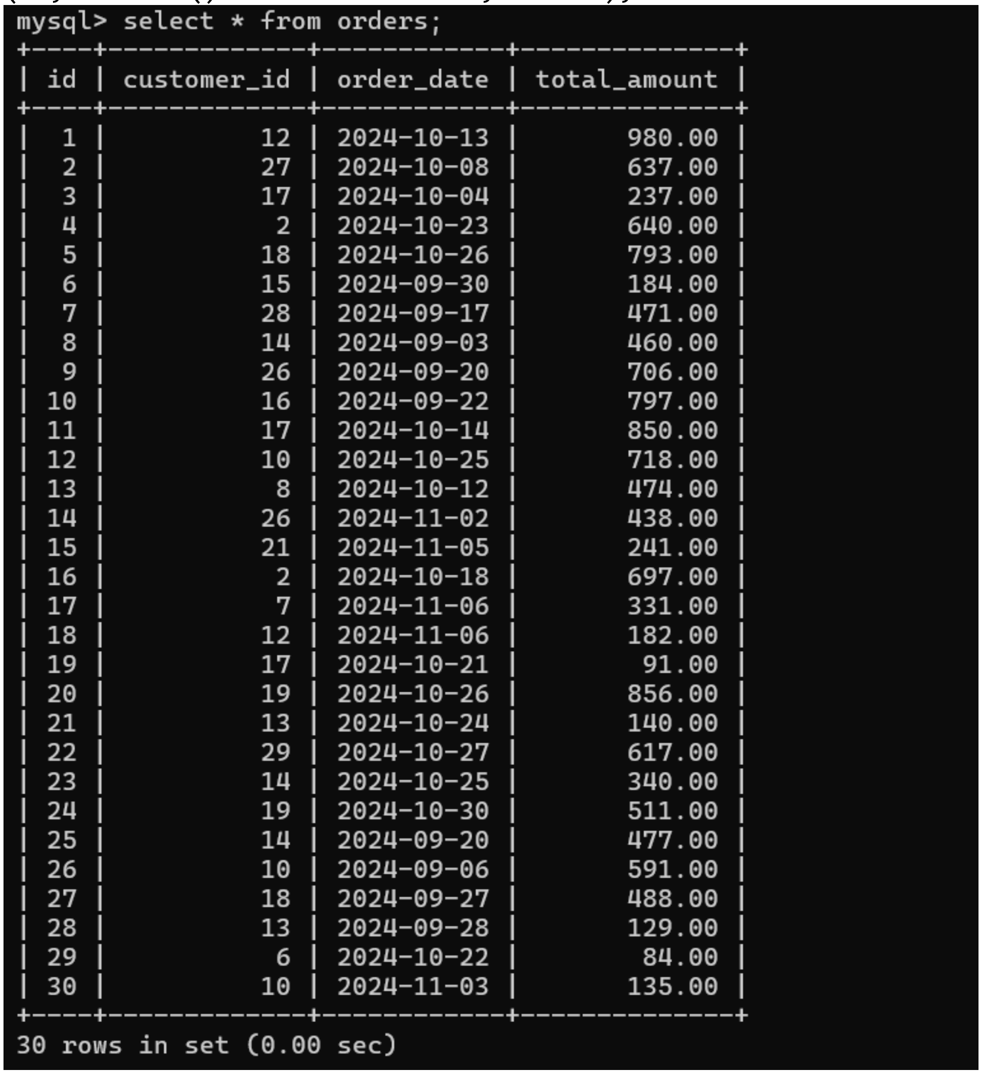
```
mysql> UPDATE products
    -> SET price = 45.00
    -> WHERE name = 'Product C';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
Select * from products;
```



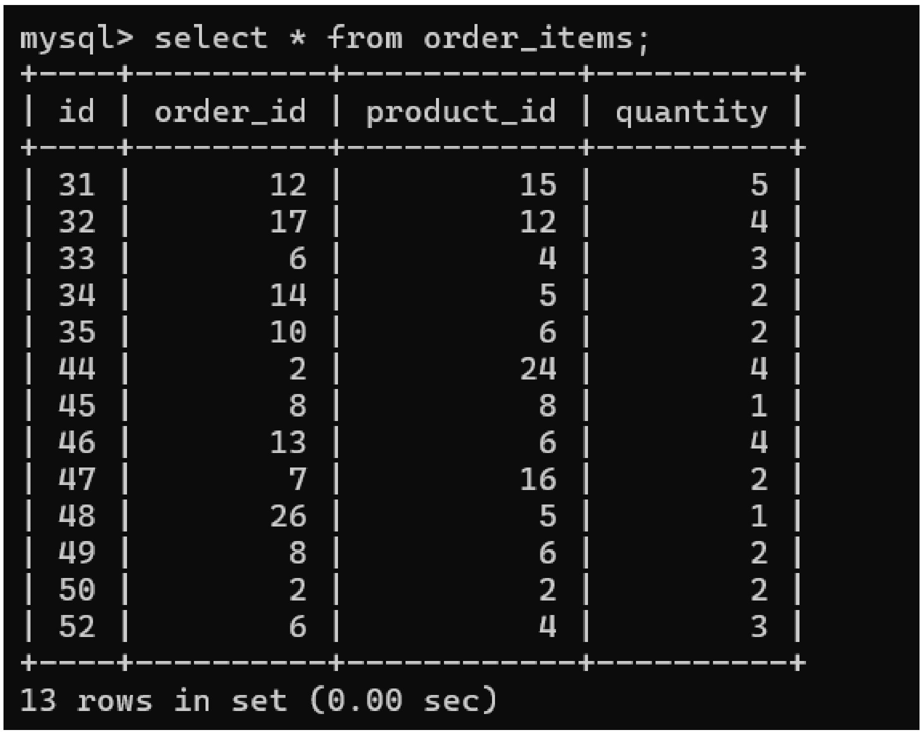
```
mysql> select * from products;
+----+-----------+--------+-------------------------------------------+----------+
| id | name      | price  | description                               | discount |
+----+-----------+--------+-------------------------------------------+----------+
|  1 | Product A |  25.00 | Durable home appliance                    |     0.00 |
|  2 | Product B |  30.00 | Latest tech gadget                        |     0.00 |
|  3 | Product C |  45.00 | Eco-friendly kitchen tool                 |     0.00 |
|  4 | Product D |  55.00 | High-quality furniture item               |     0.00 |
|  5 | Product E |  15.00 | Affordable stationery                     |     0.00 |
|  6 | Product F |  10.00 | Portable charger for mobile devices       |     0.00 |
|  7 | Product G |  20.00 | Wireless earbuds with great sound quality |     0.00 |
|  8 | Product H |  80.00 | Smart home assistant device               |     0.00 |
|  9 | Product I |  60.00 | Stylish and ergonomic office chair        |     0.00 |
| 10 | Product J |  35.00 | Compact Bluetooth speaker                 |     0.00 |
| 11 | Product K |  25.50 | Stainless steel water bottle              |     0.00 |
| 12 | Product L |  40.00 | Noise-canceling headphones                |     0.00 |
| 13 | Product M |  50.00 | High-performance laptop stand             |     0.00 |
| 14 | Product N |  45.00 | Wireless keyboard and mouse combo         |     0.00 |
| 15 | Product O |  75.00 | Smart thermostat for energy savings       |     0.00 |
| 16 | Product P |  90.00 | Electric kettle with temperature control  |     0.00 |
| 17 | Product Q |  12.00 | Eco-friendly reusable shopping bags       |     0.00 |
| 18 | Product R |  20.00 | Digital alarm clock with LED display      |     0.00 |
| 19 | Product S |  85.00 | Adjustable standing desk converter        |     0.00 |
| 20 | Product T |  30.00 | Portable mini fan for desk                |     0.00 |
| 21 | Product U |  55.00 | High-definition web camera                |     0.00 |
| 22 | Product V |  22.50 | USB-C hub for multi-device connectivity   |     0.00 |
| 23 | Product W |  15.00 | Cable management kit                      |     0.00 |
| 24 | Product X | 100.00 | Smart light bulbs, pack of 4              |     0.00 |
| 25 | Product Y |  65.00 | Advanced fitness tracker watch            |     0.00 |
| 26 | Product Z | 120.00 | Robot vacuum cleaner with app control     |     0.00 |
+----+-----------+--------+-------------------------------------------+----------+
26 rows in set (0.00 sec)
```

```
INSERT INTO orders (customer_id, order_date, total_amount) VALUES
 (12, CURDATE() - INTERVAL 25 DAY, 980.00),
(27, CURDATE() - INTERVAL 30 DAY, 637.00),
(17, CURDATE() - INTERVAL 34 DAY, 237.00),
(2,CURDATE() - INTERVAL 15 DAY, 640.00),
(18, CURDATE() - INTERVAL 12 DAY, 793.00),
(15, CURDATE() - INTERVAL 38 DAY, 184.00),
(28, CURDATE() - INTERVAL 51 DAY, 471.00),
(14, CURDATE() - INTERVAL 65 DAY, 460.00),
(26, CURDATE() - INTERVAL 48 DAY, 706.00),
(16, CURDATE() - INTERVAL 46 DAY, 797.00),
(17, CURDATE() - INTERVAL 24 DAY, 850.00),
(10, CURDATE() - INTERVAL 13 DAY, 718.00),
(8,CURDATE() - INTERVAL 26 DAY, 474.00),
(26, CURDATE() - INTERVAL 5 DAY, 438.00),
(21, CURDATE() - INTERVAL 2 DAY, 241.00),
(2,CURDATE() - INTERVAL 20 DAY, 697.00),
(7, CURDATE() - INTERVAL 1 DAY, 331.00),
(12, CURDATE() - INTERVAL 1 DAY, 182.00),
(17, CURDATE() - INTERVAL 17 DAY, 91.00),
(19,CURDATE() - INTERVAL 12 DAY, 856.00),
(13, CURDATE() - INTERVAL 14 DAY, 140.00),
(29, CURDATE() - INTERVAL 11 DAY, 617.00),
(14, CURDATE() - INTERVAL 13 DAY, 340.00),
(19,CURDATE() - INTERVAL 8 DAY, 511.00),
(14, CURDATE() - INTERVAL 48 DAY, 477.00),
(10, CURDATE() - INTERVAL 62 DAY, 591.00),
(18, CURDATE() - INTERVAL 41 DAY, 488.00),
(13, CURDATE() - INTERVAL 40 DAY, 129.00),
(6, CURDATE() - INTERVAL 16 DAY, 84.00),
(10, CURDATE() - INTERVAL 4 DAY, 135.00);
```



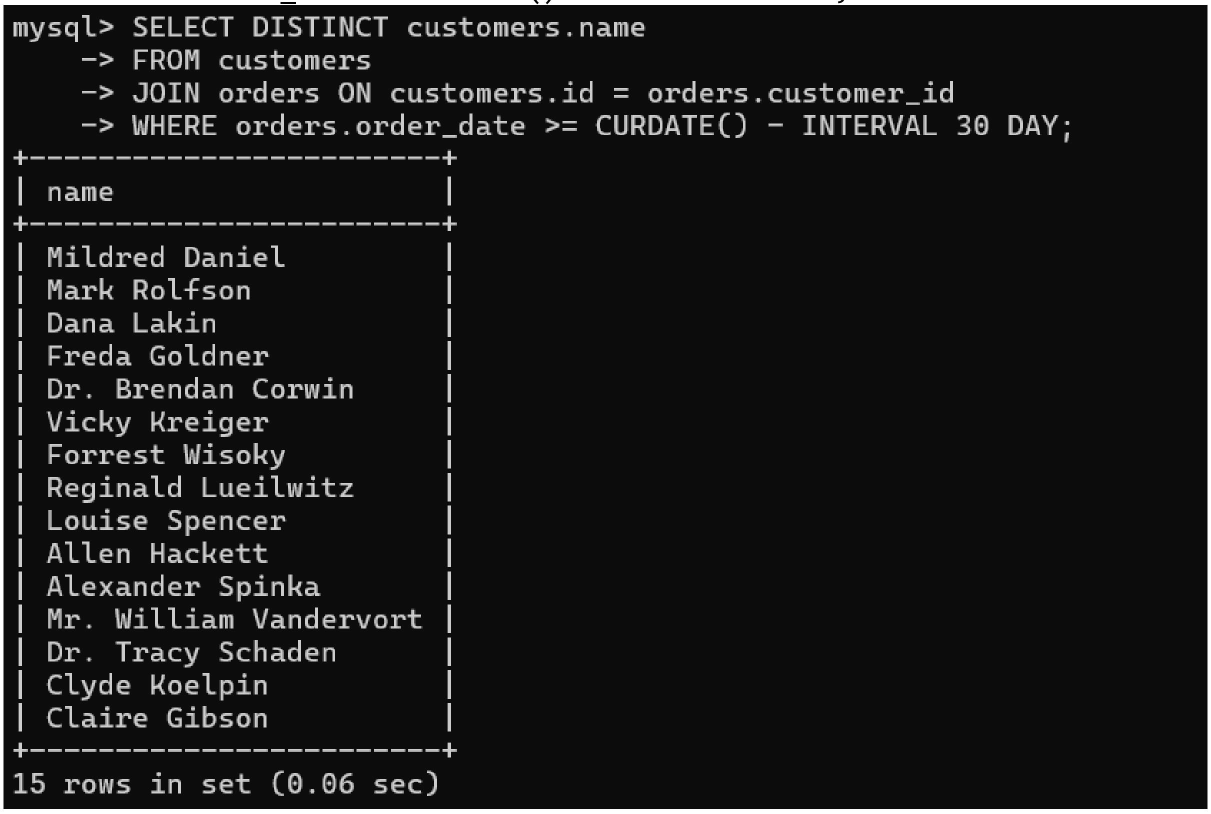
```
mysql> select * from orders;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  1 |          12 | 2024-10-13 |       980.00 |
|  2 |          27 | 2024-10-08 |       637.00 |
|  3 |          17 | 2024-10-04 |       237.00 |
|  4 |           2 | 2024-10-23 |       640.00 |
|  5 |          18 | 2024-10-26 |       793.00 |
|  6 |          15 | 2024-09-30 |       184.00 |
|  7 |          28 | 2024-09-17 |       471.00 |
|  8 |          14 | 2024-09-03 |       460.00 |
|  9 |          26 | 2024-09-20 |       706.00 |
| 10 |          16 | 2024-09-22 |       797.00 |
| 11 |          17 | 2024-10-14 |       850.00 |
| 12 |          10 | 2024-10-25 |       718.00 |
| 13 |           8 | 2024-10-12 |       474.00 |
| 14 |          26 | 2024-11-02 |       438.00 |
| 15 |          21 | 2024-11-05 |       241.00 |
| 16 |           2 | 2024-10-18 |       697.00 |
| 17 |           7 | 2024-11-06 |       331.00 |
| 18 |          12 | 2024-11-06 |       182.00 |
| 19 |          17 | 2024-10-21 |        91.00 |
| 20 |          19 | 2024-10-26 |       856.00 |
| 21 |          13 | 2024-10-24 |       140.00 |
| 22 |          29 | 2024-10-27 |       617.00 |
| 23 |          14 | 2024-10-25 |       340.00 |
| 24 |          19 | 2024-10-30 |       511.00 |
| 25 |          14 | 2024-09-20 |       477.00 |
| 26 |          10 | 2024-09-06 |       591.00 |
| 27 |          18 | 2024-09-27 |       488.00 |
| 28 |          13 | 2024-09-28 |       129.00 |
| 29 |           6 | 2024-10-22 |        84.00 |
| 30 |          10 | 2024-11-03 |       135.00 |
+----+-------------+------------+--------------+
30 rows in set (0.00 sec)
```

```
-- 9. The `order_items` table was created earlier to normalize the data by
storing items for each order, allowing a many-to-many relationship between
`orders` and `products`.

-- Insert sample data into order_items table (normalized item-level details
per order)
CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    FOREIGN KEY (order_id) REFERENCES orders(id),
    FOREIGN KEY (product_id) REFERENCES products(id)
);

-- Insert data into order_items to reference items in each order
INSERT INTO order_items (order_id, product_id, quantity) VALUES
(12,15,5),
(17,12,4),
(6,4,3),
(14,5,2),
(10,6,2),
(2,24,4),
(8,8,1),
(13,6,4),
(7,16,2),
(26,5,1),
(8,6,2),
(2,2,2),
(6,4,3);
```



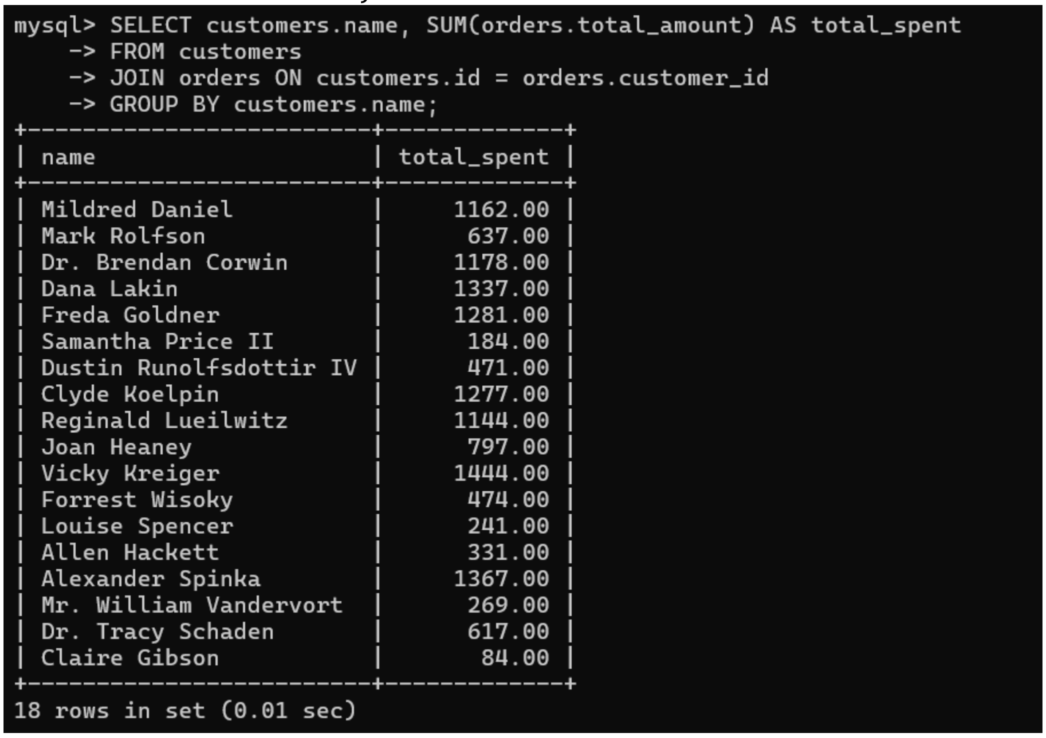
```
mysql> select * from order_items;
+----+----------+------------+----------+
| id | order_id | product_id | quantity |
+----+----------+------------+----------+
| 31 |       12 |         15 |        5 |
| 32 |       17 |         12 |        4 |
| 33 |        6 |          4 |        3 |
| 34 |       14 |          5 |        2 |
| 35 |       10 |          6 |        2 |
| 44 |        2 |         24 |        4 |
| 45 |        8 |          8 |        1 |
| 46 |       13 |          6 |        4 |
| 47 |        7 |         16 |        2 |
| 48 |       26 |          5 |        1 |
| 49 |        8 |          6 |        2 |
| 50 |        2 |          2 |        2 |
| 52 |        6 |          4 |        3 |
+----+----------+------------+----------+
13 rows in set (0.00 sec)
```

```
-- Queries for the e-commerce system:

-- 1. Retrieve all customers who have placed an order in the last 30 days.
SELECT DISTINCT customers.name
FROM customers
JOIN orders ON customers.id = orders.customer_id
WHERE orders.order_date >= CURDATE() - INTERVAL 30 DAY;
```



```
mysql> SELECT DISTINCT customers.name
    -> FROM customers
    -> JOIN orders ON customers.id = orders.customer_id
    -> WHERE orders.order_date >= CURDATE() - INTERVAL 30 DAY;
+------------------------+
| name                   |
+------------------------+
| Mildred Daniel         |
| Mark Rolfson           |
| Dana Lakin             |
| Freda Goldner          |
| Dr. Brendan Corwin     |
| Vicky Kreiger          |
| Forrest Wisoky         |
| Reginald Lueilwitz     |
| Louise Spencer         |
| Allen Hackett          |
| Alexander Spinka       |
| Mr. William Vandervort |
| Dr. Tracy Schaden      |
| Clyde Koelpin          |
| Claire Gibson          |
+------------------------+
15 rows in set (0.06 sec)
```

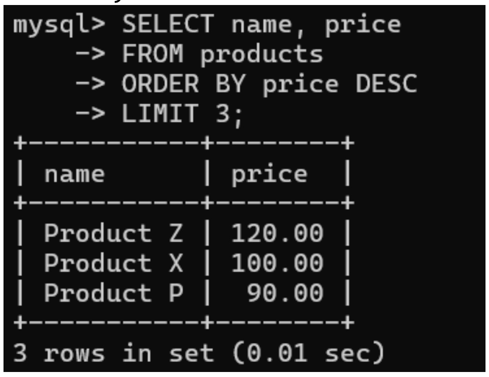
```
-- 2. Get the total amount of all orders placed by each customer.
SELECT customers.name, SUM(orders.total_amount) AS total_spent
FROM customers
JOIN orders ON customers.id = orders.customer_id
GROUP BY customers.name;
```



```
mysql> SELECT customers.name, SUM(orders.total_amount) AS total_spent
    -> FROM customers
    -> JOIN orders ON customers.id = orders.customer_id
    -> GROUP BY customers.name;
+------------------------+-------------+
| name                   | total_spent |
+------------------------+-------------+
| Mildred Daniel         |     1162.00 |
| Mark Rolfson           |      637.00 |
| Dr. Brendan Corwin     |     1178.00 |
| Dana Lakin             |     1337.00 |
| Freda Goldner          |     1281.00 |
| Samantha Price II      |      184.00 |
| Dustin Runolfsdottir IV |      471.00 |
| Clyde Koelpin          |     1277.00 |
| Reginald Lueilwitz     |     1144.00 |
| Joan Heaney            |      797.00 |
| Vicky Kreiger          |     1444.00 |
| Forrest Wisoky         |      474.00 |
| Louise Spencer         |      241.00 |
| Allen Hackett          |      331.00 |
| Alexander Spinka       |     1367.00 |
| Mr. William Vandervort |      269.00 |
| Dr. Tracy Schaden      |      617.00 |
| Claire Gibson          |       84.00 |
+------------------------+-------------+
18 rows in set (0.01 sec)
```

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

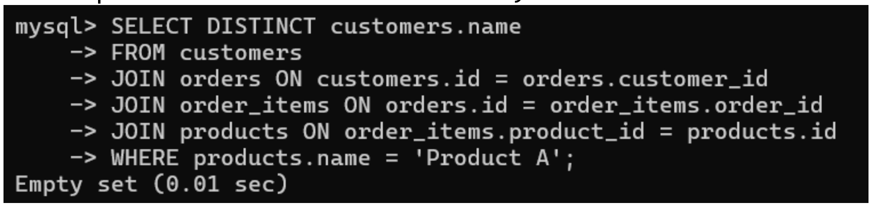
```sql
SELECT name, price
FROM products
ORDER BY price DESC
LIMIT 3;
```



```
mysql> SELECT name, price
    -> FROM products
    -> ORDER BY price DESC
    -> LIMIT 3;
+-----------+--------+
| name      | price  |
+-----------+--------+
| Product Z | 120.00 |
| Product X | 100.00 |
| Product P |  90.00 |
+-----------+--------+
3 rows in set (0.01 sec)
```

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

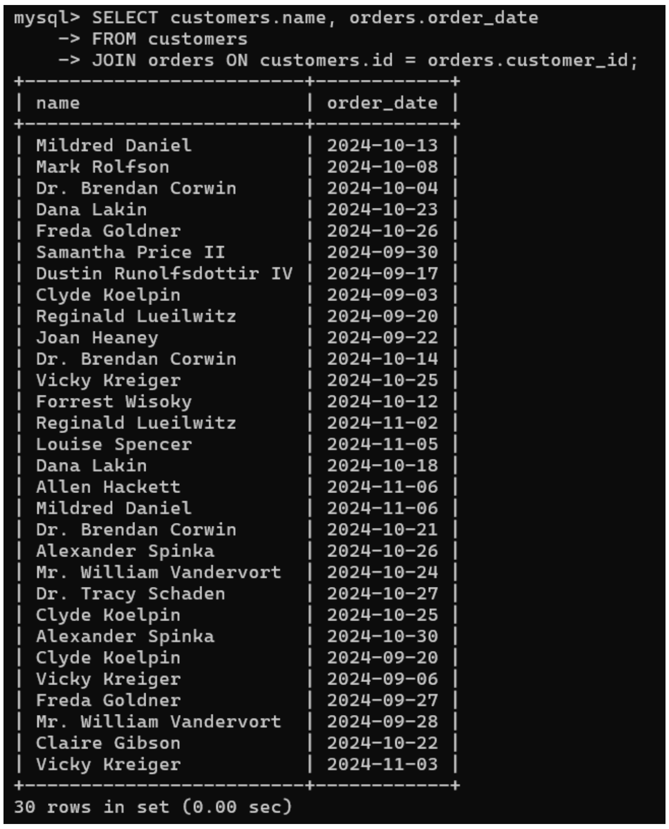
```sql
SELECT DISTINCT customers.name
FROM customers
JOIN orders ON customers.id = orders.customer_id
JOIN order_items ON orders.id = order_items.order_id
JOIN products ON order_items.product_id = products.id
WHERE products.name = 'Product A';
```



```
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';
Empty set (0.01 sec)
```

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

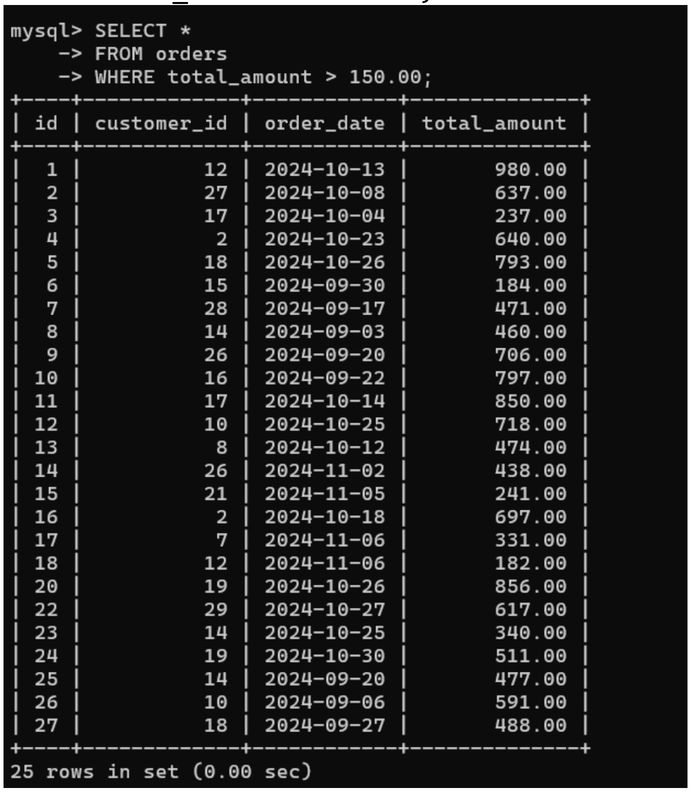
```sql
SELECT customers.name, orders.order_date
FROM customers
JOIN orders ON customers.id = orders.customer_id;
```



```
mysql> SELECT customers.name, orders.order_date
    -> FROM customers
    -> JOIN orders ON customers.id = orders.customer_id;
+-------------------------+------------+
| name                    | order_date |
+-------------------------+------------+
| Mildred Daniel          | 2024-10-13 |
| Mark Rolfson            | 2024-10-08 |
| Dr. Brendan Corwin      | 2024-10-04 |
| Dana Lakin              | 2024-10-23 |
| Freda Goldner           | 2024-10-26 |
| Samantha Price II       | 2024-09-30 |
| Dustin Runolfsdottir IV | 2024-09-17 |
| Clyde Koelpin           | 2024-09-03 |
| Reginald Lueilwitz      | 2024-09-20 |
| Joan Heaney             | 2024-09-22 |
| Dr. Brendan Corwin      | 2024-10-14 |
| Vicky Kreiger           | 2024-10-25 |
| Forrest Wisoky          | 2024-10-12 |
| Reginald Lueilwitz      | 2024-11-02 |
| Louise Spencer          | 2024-11-05 |
| Dana Lakin              | 2024-10-18 |
| Allen Hackett           | 2024-11-06 |
| Mildred Daniel          | 2024-11-06 |
| Dr. Brendan Corwin      | 2024-10-21 |
| Alexander Spinka        | 2024-10-26 |
| Mr. William Vandervort  | 2024-10-24 |
| Dr. Tracy Schaden       | 2024-10-27 |
| Clyde Koelpin           | 2024-10-25 |
| Alexander Spinka        | 2024-10-30 |
| Clyde Koelpin           | 2024-09-20 |
| Vicky Kreiger           | 2024-09-06 |
| Freda Goldner           | 2024-09-27 |
| Mr. William Vandervort  | 2024-09-28 |
| Claire Gibson           | 2024-10-22 |
| Vicky Kreiger           | 2024-11-03 |
+-------------------------+------------+
30 rows in set (0.00 sec)
```

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

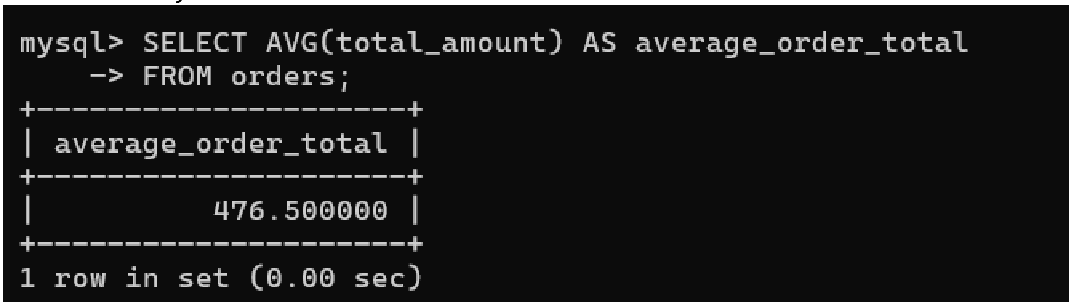
```
SELECT *
FROM orders
WHERE total_amount > 150.00;
```



```
mysql> SELECT *
    -> FROM orders
    -> WHERE total_amount > 150.00;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  1 |          12 | 2024-10-13 |       980.00 |
|  2 |          27 | 2024-10-08 |       637.00 |
|  3 |          17 | 2024-10-04 |       237.00 |
|  4 |           2 | 2024-10-23 |       640.00 |
|  5 |          18 | 2024-10-26 |       793.00 |
|  6 |          15 | 2024-09-30 |       184.00 |
|  7 |          28 | 2024-09-17 |       471.00 |
|  8 |          14 | 2024-09-03 |       460.00 |
|  9 |          26 | 2024-09-20 |       706.00 |
| 10 |          16 | 2024-09-22 |       797.00 |
| 11 |          17 | 2024-10-14 |       850.00 |
| 12 |          10 | 2024-10-25 |       718.00 |
| 13 |           8 | 2024-10-12 |       474.00 |
| 14 |          26 | 2024-11-02 |       438.00 |
| 15 |          21 | 2024-11-05 |       241.00 |
| 16 |           2 | 2024-10-18 |       697.00 |
| 17 |           7 | 2024-11-06 |       331.00 |
| 18 |          12 | 2024-11-06 |       182.00 |
| 20 |          19 | 2024-10-26 |       856.00 |
| 22 |          29 | 2024-10-27 |       617.00 |
| 23 |          14 | 2024-10-25 |       340.00 |
| 24 |          19 | 2024-10-30 |       511.00 |
| 25 |          14 | 2024-09-20 |       477.00 |
| 26 |          10 | 2024-09-06 |       591.00 |
| 27 |          18 | 2024-09-27 |       488.00 |
+----+-------------+------------+--------------+
25 rows in set (0.00 sec)
```

-- 10. Retrieve the average total of all orders.

```
SELECT AVG(total_amount) AS average_order_total
FROM orders;
```



```
mysql> SELECT AVG(total_amount) AS average_order_total
    -> FROM orders;
+---------------------+
| average_order_total |
+---------------------+
|          476.500000 |
+---------------------+
1 row in set (0.00 sec)
```