My SQL Task

Create the ecommerce database

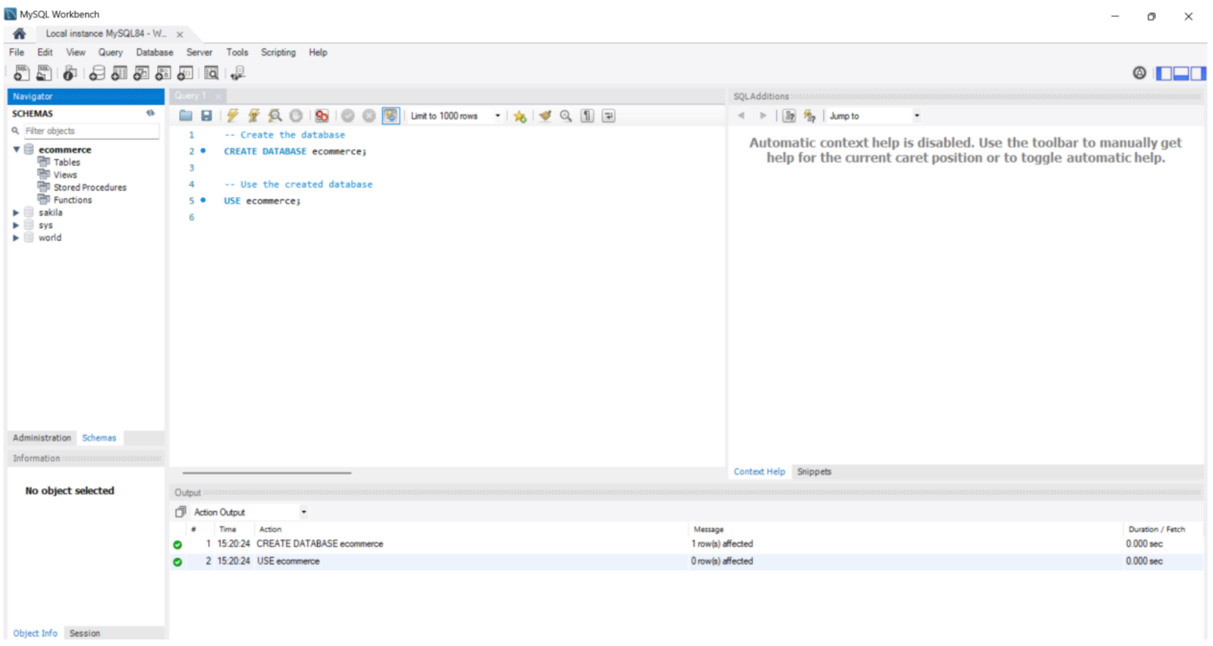
-- Create the database

CREATE DATABASE ecommerce;

-- Use the created database

USE ecommerce;

Output:



```
-- Create the database
CREATE DATABASE ecommerce;

-- Use the created database
USE ecommerce;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#   Time      Action                     Message            Duration / Fetch
1   15:20:24  CREATE DATABASE ecommerce  1 row(s) affected  0.000 sec
2   15:20:24  USE ecommerce              0 row(s) affected  0.000 sec
```

Create the customers, orders, and products tables

-- Create the 'customers' table

CREATE TABLE customers (

  id INT AUTO_INCREMENT PRIMARY KEY,

  name VARCHAR(100) NOT NULL,

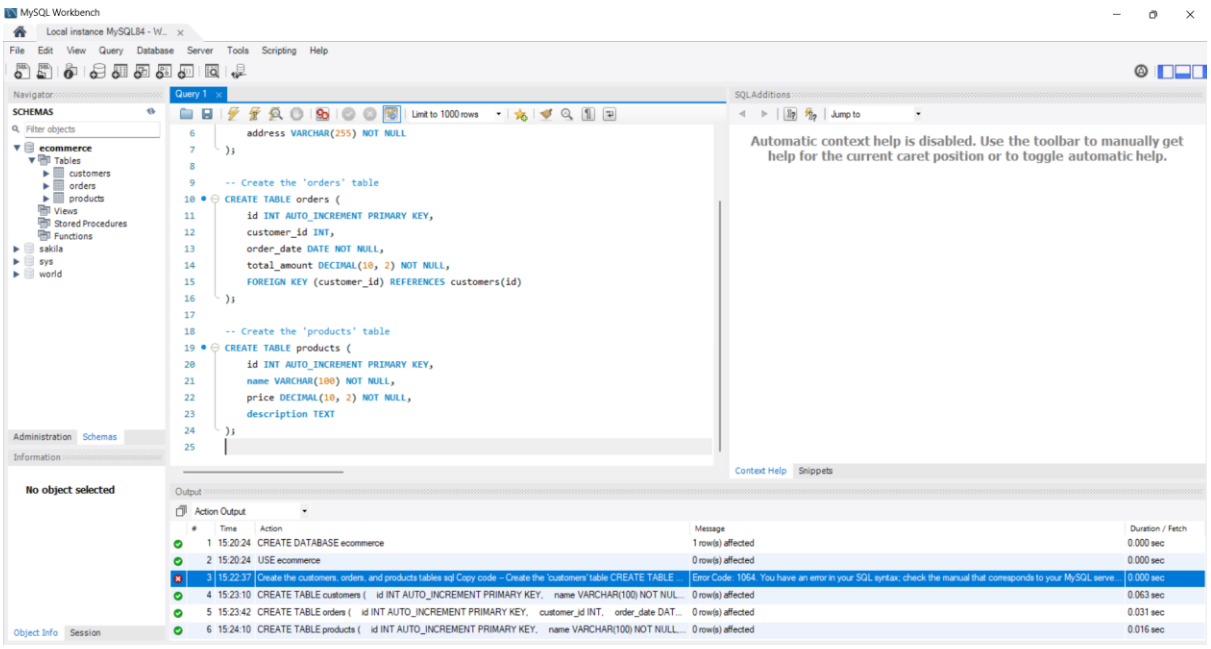
  email VARCHAR(100) UNIQUE NOT NULL,

  address VARCHAR(255) NOT NULL

);

```
-- Create the 'orders' table

CREATE TABLE orders (

    id INT AUTO_INCREMENT PRIMARY KEY,

    customer_id INT,

    order_date DATE NOT NULL,

    total_amount DECIMAL(10, 2) NOT NULL,

    FOREIGN KEY (customer_id) REFERENCES customers(id)

);


-- Create the 'products' table

CREATE TABLE products (

    id INT AUTO_INCREMENT PRIMARY KEY,

    name VARCHAR(100) NOT NULL,

    price DECIMAL(10, 2) NOT NULL,

    description TEXT

);
```

Output:



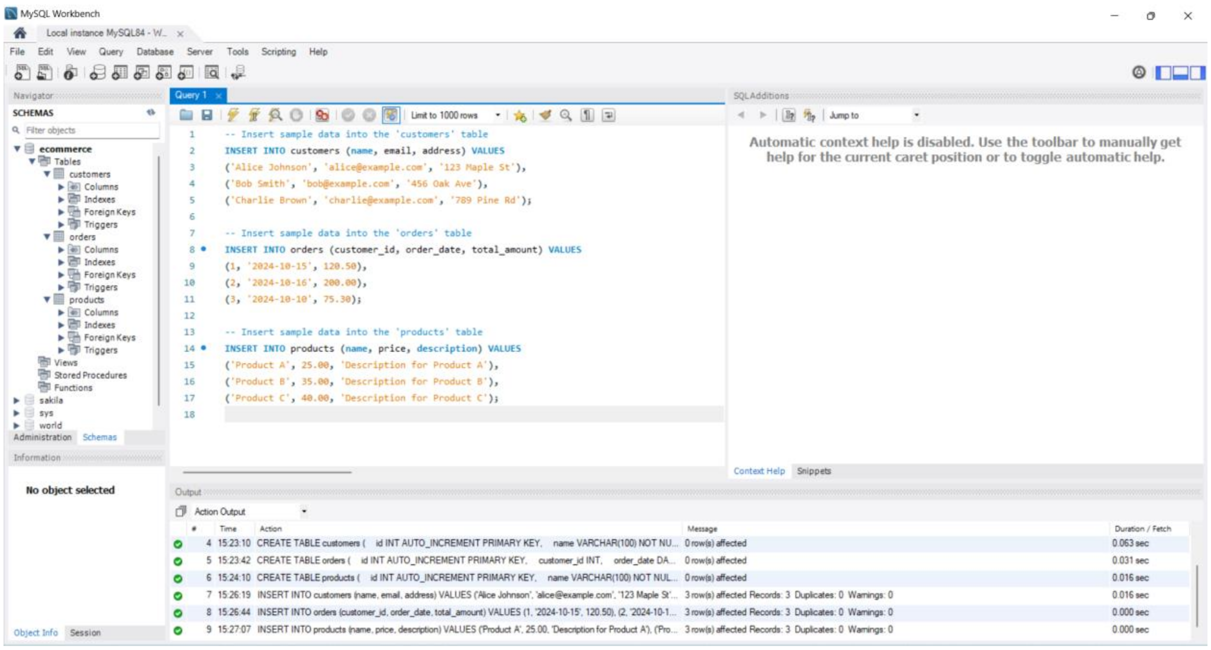
```
    address VARCHAR(255) NOT NULL
);

-- Create the 'orders' table
CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATE NOT NULL,
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(id)
);

-- Create the 'products' table
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    description TEXT
);
```

	#	Time	Action	Message	Duration / Fetch
✓	1	15:20:24	CREATE DATABASE ecommerce	1 row(s) affected	0.000 sec
✓	2	15:20:24	USE ecommerce	0 row(s) affected	0.000 sec
✗	3	15:22:37	Create the customers, orders, and products tables sql Copy code -- Create the 'customers' table CREATE TABLE ...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serve...	0.000 sec
✓	4	15:23:10	CREATE TABLE customers (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NUL...	0 row(s) affected	0.063 sec
✓	5	15:23:42	CREATE TABLE orders (id INT AUTO_INCREMENT PRIMARY KEY, customer_id INT, order_date DAT...	0 row(s) affected	0.031 sec
✓	6	15:24:10	CREATE TABLE products (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL,...	0 row(s) affected	0.016 sec

Insert Sample Data into the Tables

```
-- Insert sample data into the 'customers' table

INSERT INTO customers (name, email, address) VALUES

('Alice Johnson', 'alice@example.com', '123 Maple St'),

('Bob Smith', 'bob@example.com', '456 Oak Ave'),

('Charlie Brown', 'charlie@example.com', '789 Pine Rd');


-- Insert sample data into the 'orders' table

INSERT INTO orders (customer_id, order_date, total_amount) VALUES

(1, '2024-10-15', 120.50),

(2, '2024-10-16', 200.00),

(3, '2024-10-10', 75.30);


-- Insert sample data into the 'products' table

INSERT INTO products (name, price, description) VALUES

('Product A', 25.00, 'Description for Product A'),

('Product B', 35.00, 'Description for Product B'),

('Product C', 40.00, 'Description for Product C');
```

Output:



```sql
-- Insert sample data into the 'customers' table
INSERT INTO customers (name, email, address) VALUES
('Alice Johnson', 'alice@example.com', '123 Maple St'),
('Bob Smith', 'bob@example.com', '456 Oak Ave'),
('Charlie Brown', 'charlie@example.com', '789 Pine Rd');

-- Insert sample data into the 'orders' table
INSERT INTO orders (customer_id, order_date, total_amount) VALUES
(1, '2024-10-15', 120.50),
(2, '2024-10-16', 200.00),
(3, '2024-10-10', 75.30);

-- Insert sample data into the 'products' table
INSERT INTO products (name, price, description) VALUES
('Product A', 25.00, 'Description for Product A'),
('Product B', 35.00, 'Description for Product B'),
('Product C', 40.00, 'Description for Product C');
```

```
#  Time      Action                                                                                         Message                                                Duration / Fetch
4  15:23:10  CREATE TABLE customers (   id INT AUTO_INCREMENT PRIMARY KEY,    name VARCHAR(100) NOT NU...   0 row(s) affected                                      0.063 sec
5  15:23:42  CREATE TABLE orders (   id INT AUTO_INCREMENT PRIMARY KEY,    customer_id INT,    order_date DA...   0 row(s) affected                                   0.031 sec
6  15:24:10  CREATE TABLE products (   id INT AUTO_INCREMENT PRIMARY KEY,    name VARCHAR(100) NOT NUL...   0 row(s) affected                                      0.016 sec
7  15:26:19  INSERT INTO customers (name, email, address) VALUES ('Alice Johnson', 'alice@example.com', '123 Maple St'...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.016 sec
8  15:26:44  INSERT INTO orders (customer_id, order_date, total_amount) VALUES (1, '2024-10-15', 120.50), (2, '2024-10-1...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
9  15:27:07  INSERT INTO products (name, price, description) VALUES ('Product A', 25.00, 'Description for Product A'), ('Pro...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
```

Queries

Retrieve all customers who have placed an order in the last 30 days

-- Retrieve customers who placed an order in the last 30 days

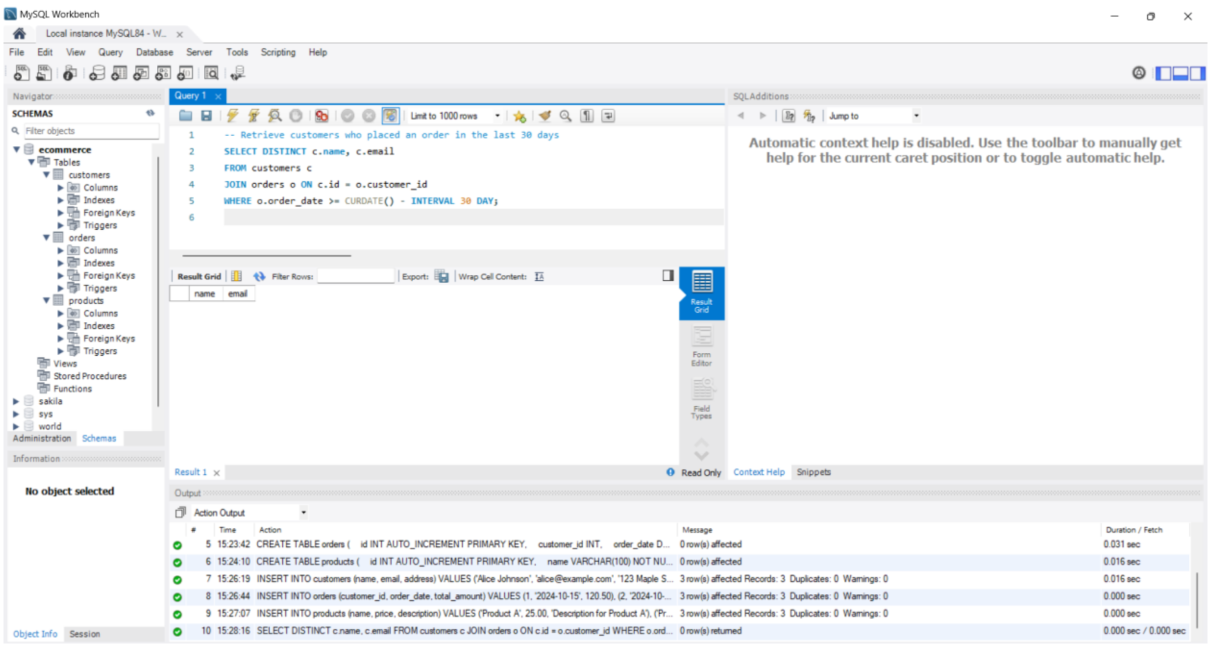
SELECT DISTINCT c.name, c.email

FROM customers c

JOIN orders o ON c.id = o.customer_id

WHERE o.order_date >= CURDATE() - INTERVAL 30 DAY;

Output:



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

	name	email

```
#   Time      Action                                                                                           Message                                                       Duration / Fetch
5   15:23:42  CREATE TABLE orders (    id INT AUTO_INCREMENT PRIMARY KEY,    customer_id INT,    order_date D...   0 row(s) affected                                             0.031 sec
6   15:24:10  CREATE TABLE products (    id INT AUTO_INCREMENT PRIMARY KEY,    name VARCHAR(100) NOT NU...         0 row(s) affected                                             0.016 sec
7   15:26:19  INSERT INTO customers (name, email, address) VALUES ('Alice Johnson', 'alice@example.com', '123 Maple S...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.016 sec
8   15:26:44  INSERT INTO orders (customer_id, order_date, total_amount) VALUES (1, '2024-10-15', 120.50), (2, '2024-10-...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
9   15:27:07  INSERT INTO products (name, price, description) VALUES ('Product A', 25.00, 'Description for Product A'), ('Pr...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
10  15:28:16  SELECT DISTINCT c.name, c.email FROM customers c JOIN orders o ON c.id = o.customer_id WHERE o.ord...   0 row(s) returned                                             0.000 sec / 0.000 sec
```

Get the total amount of all orders placed by each customer

-- Get total amount of orders placed by each customer

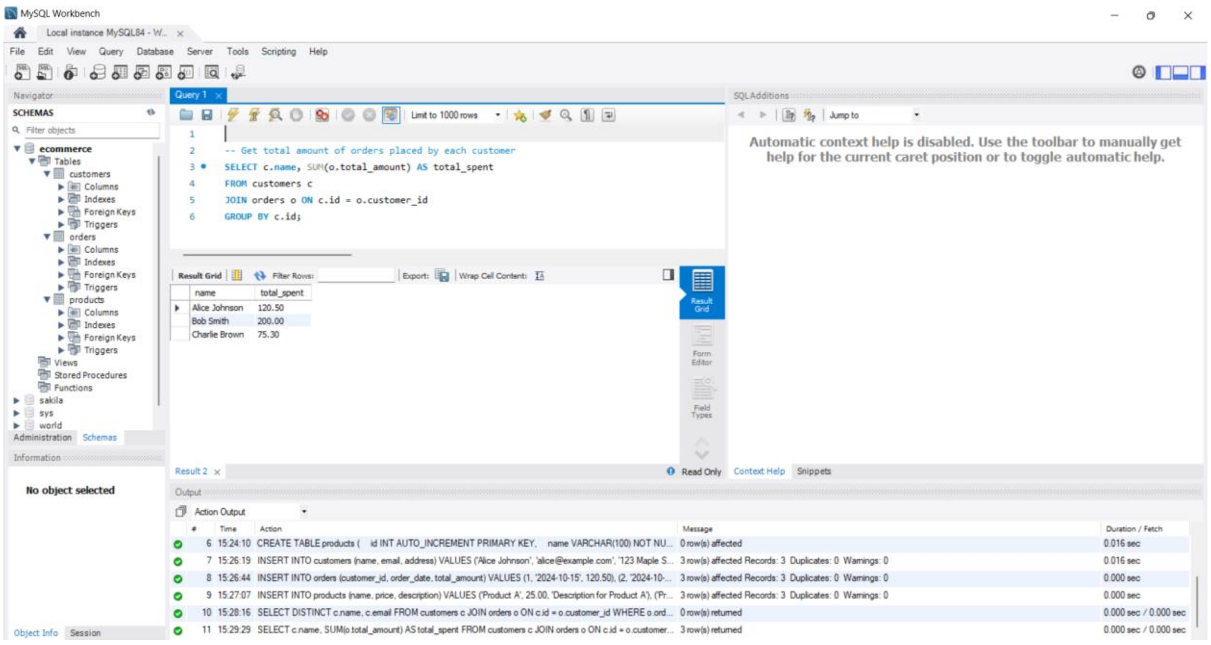
SELECT c.name, SUM(o.total_amount) AS total_spent

FROM customers c

JOIN orders o ON c.id = o.customer_id

GROUP BY c.id;

Output:



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

name	total_spent
Alice Johnson	120.50
Bob Smith	200.00
Charlie Brown	75.30

```
#  Time      Action                                                                                              Message                                                      Duration / Fetch
6  15:24:10  CREATE TABLE products (    id INT AUTO_INCREMENT PRIMARY KEY,    name VARCHAR(100) NOT NU...        0 row(s) affected                                            0.016 sec
7  15:26:19  INSERT INTO customers (name, email, address) VALUES ('Alice Johnson', 'alice@example.com', '123 Maple S...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.016 sec
8  15:26:44  INSERT INTO orders (customer_id, order_date, total_amount) VALUES (1, '2024-10-15', 120.50), (2, '2024-10-...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
9  15:27:07  INSERT INTO products (name, price, description) VALUES ('Product A', 25.00, 'Description for Product A'), ('Pr...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
10 15:28:16  SELECT DISTINCT c.name, c.email FROM customers c JOIN orders o ON c.id = o.customer_id WHERE o.ord...   0 row(s) returned                                            0.000 sec / 0.000 sec
11 15:29:29  SELECT c.name, SUM(o.total_amount) AS total_spent FROM customers c JOIN orders o ON c.id = o.customer...   3 row(s) returned                                            0.000 sec / 0.000 sec
```

Update the price of Product C to 45.00

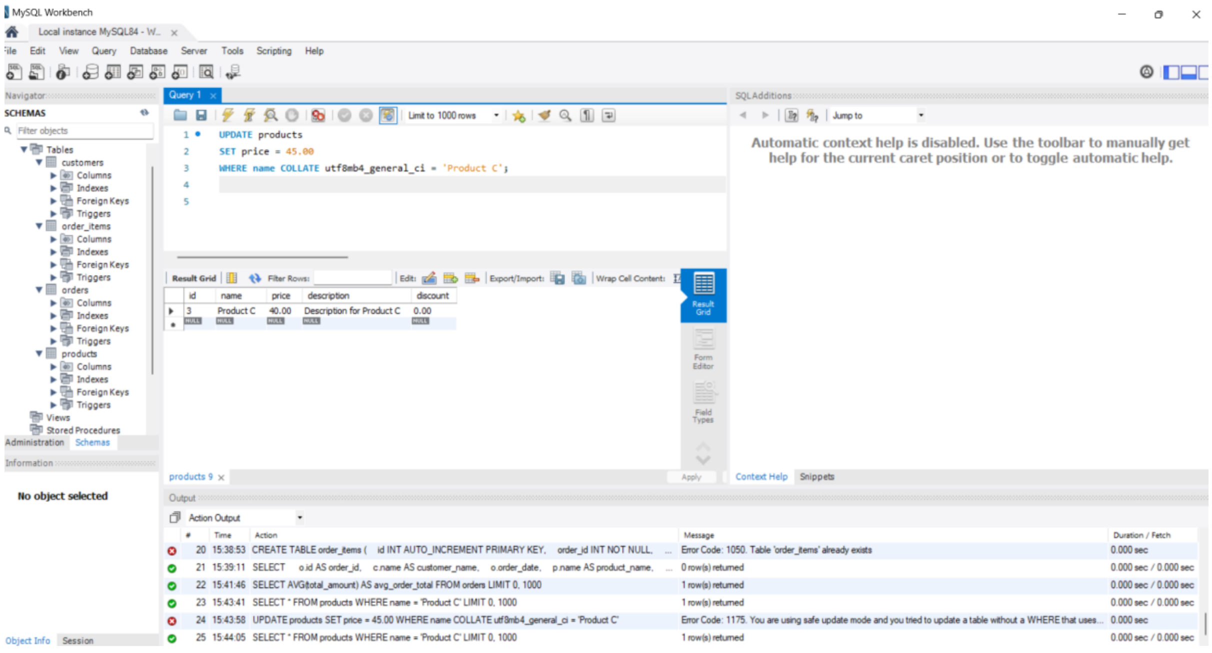
-- Update the price of Product C to 45.00

UPDATE products

SET price = 45.00

WHERE name = 'Product C';

Output:



```
UPDATE products
SET price = 45.00
WHERE name COLLATE utf8mb4_general_ci = 'Product C';
```

id	name	price	description	discount
3	Product C	40.00	Description for Product C	0.00
NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                                                                              Message                                                                                  Duration / Fetch
20  15:38:53  CREATE TABLE order_items (    id INT AUTO_INCREMENT PRIMARY KEY,    order_id INT NOT NULL,  ...  Error Code: 1050. Table 'order_items' already exists                              0.000 sec
21  15:39:11  SELECT     o.id AS order_id,    c.name AS customer_name,    o.order_date,    p.name AS product_name,  ...  0 row(s) returned                                                  0.000 sec / 0.000 sec
22  15:41:46  SELECT AVG(total_amount) AS avg_order_total FROM orders LIMIT 0, 1000                     1 row(s) returned                                                                        0.000 sec / 0.000 sec
23  15:43:41  SELECT * FROM products WHERE name = 'Product C' LIMIT 0, 1000                             1 row(s) returned                                                                        0.000 sec / 0.000 sec
24  15:43:58  UPDATE products SET price = 45.00 WHERE name COLLATE utf8mb4_general_ci = 'Product C'     Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...  0.000 sec
25  15:44:05  SELECT * FROM products WHERE name = 'Product C' LIMIT 0, 1000                             1 row(s) returned                                                                        0.000 sec / 0.000 sec
```

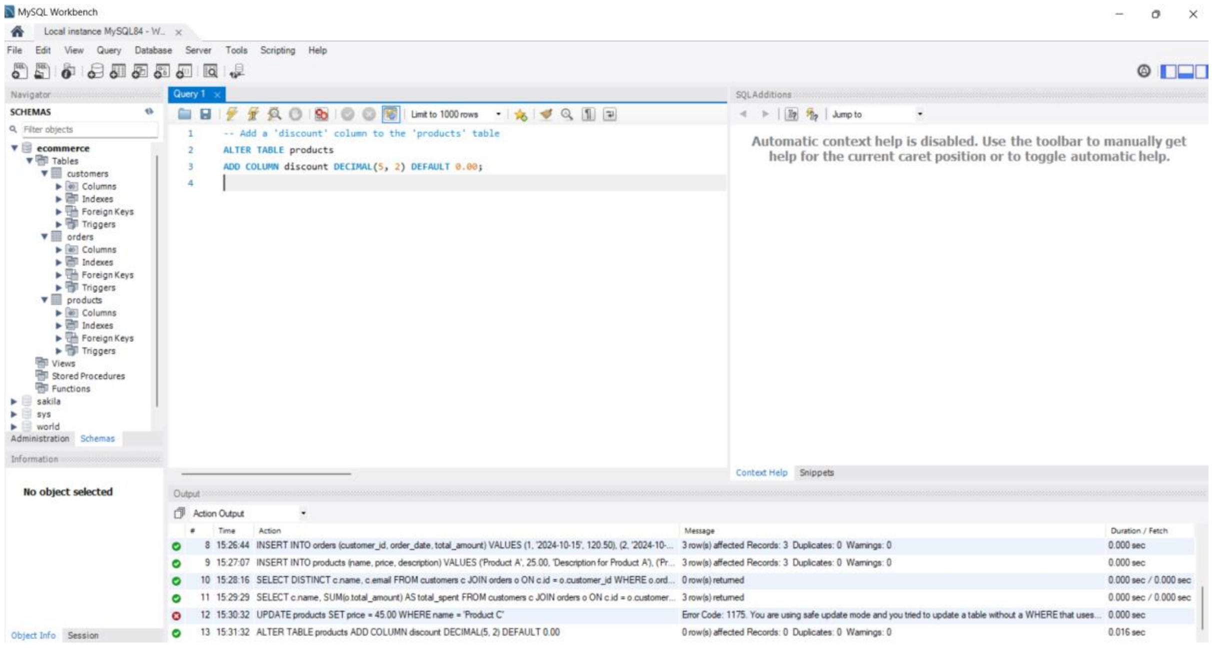
Add a new column 'discount' to the products table

-- Add a 'discount' column to the 'products' table

ALTER TABLE products

ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00;

Output:



```
-- Add a 'discount' column to the 'products' table
ALTER TABLE products
ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00;
```

```
#   Time      Action                                                                                           Message                                                  Duration / Fetch
8   15:26:44  INSERT INTO orders (customer_id, order_date, total_amount) VALUES (1, '2024-10-15', 120.50), (2, '2024-10-...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
9   15:27:07  INSERT INTO products (name, price, description) VALUES ('Product A', 25.00, 'Description for Product A'), ('Pr...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
10  15:28:16  SELECT DISTINCT c.name, c.email FROM customers c JOIN orders o ON c.id = o.customer_id WHERE o.ord...  0 row(s) returned                                        0.000 sec / 0.000 sec
11  15:29:29  SELECT c.name, SUM(o.total_amount) AS total_spent FROM customers c JOIN orders o ON c.id = o.customer...  3 row(s) returned                                   0.000 sec / 0.000 sec
12  15:30:32  UPDATE products SET price = 45.00 WHERE name = 'Product C'                                    Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...  0.000 sec
13  15:31:32  ALTER TABLE products ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00                           0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.016 sec
```

Retrieve the top 3 products with the highest price

-- Retrieve top 3 products with the highest price

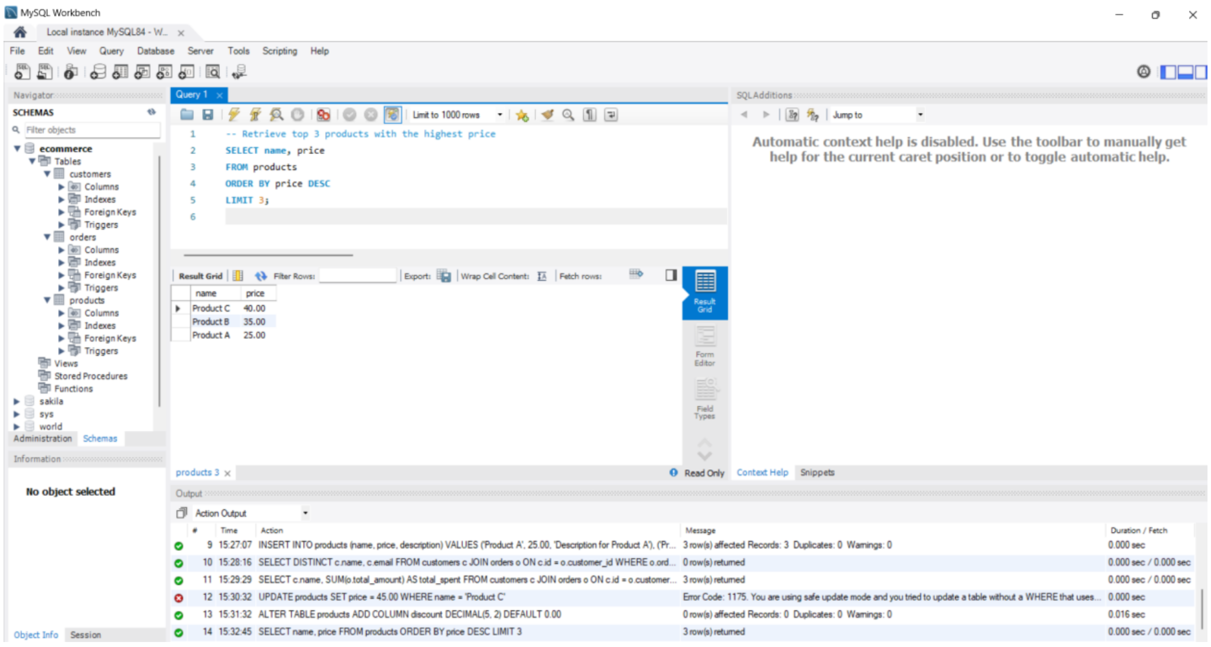
SELECT name, price

FROM products

ORDER BY price DESC

LIMIT 3;

Output:



```
-- Retrieve top 3 products with the highest price
SELECT name, price
FROM products
ORDER BY price DESC
LIMIT 3;
```

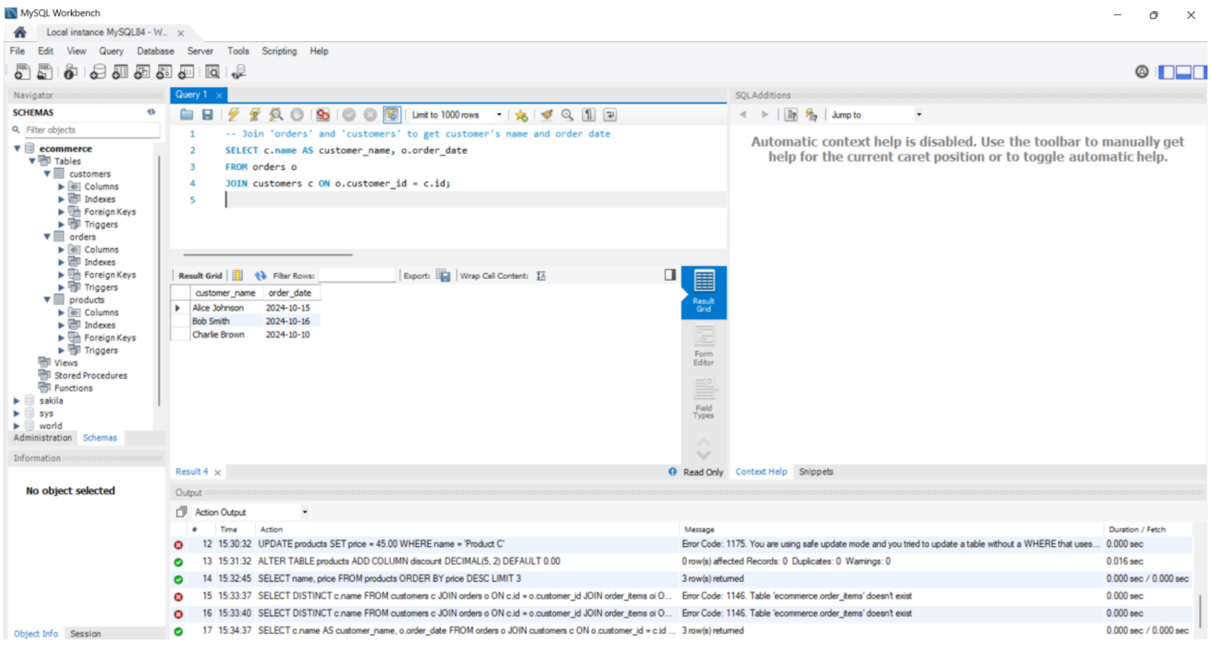
	name	price
▶	Product C	40.00
	Product B	35.00
	Product A	25.00

```
Action Output
 #   Time      Action                                                                                  Message                                                                                    Duration / Fetch
 9   15:27:07  INSERT INTO products (name, price, description) VALUES ('Product A', 25.00, 'Description for Product A'), ('Pr...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0      0.000 sec
 10  15:28:16  SELECT DISTINCT c.name, c.email FROM customers c JOIN orders o ON c.id = o.customer_id WHERE o.ord...   0 row(s) returned      0.000 sec / 0.000 sec
 11  15:29:29  SELECT c.name, SUM(o.total_amount) AS total_spent FROM customers c JOIN orders o ON c.id = o.customer...   3 row(s) returned      0.000 sec / 0.000 sec
 12  15:30:32  UPDATE products SET price = 45.00 WHERE name = 'Product C'      Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...      0.000 sec
 13  15:31:32  ALTER TABLE products ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00      0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0      0.016 sec
 14  15:32:45  SELECT name, price FROM products ORDER BY price DESC LIMIT 3      3 row(s) returned      0.000 sec / 0.000 sec
```

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

```
-- Join 'orders' and 'customers' to get customer's name and order date

SELECT c.name AS customer_name, o.order_date

FROM orders o

JOIN customers c ON o.customer_id = c.id;
```

Output:



```
-- Join 'orders' and 'customers' to get customer's name and order date
SELECT c.name AS customer_name, o.order_date
FROM orders o
JOIN customers c ON o.customer_id = c.id;
```

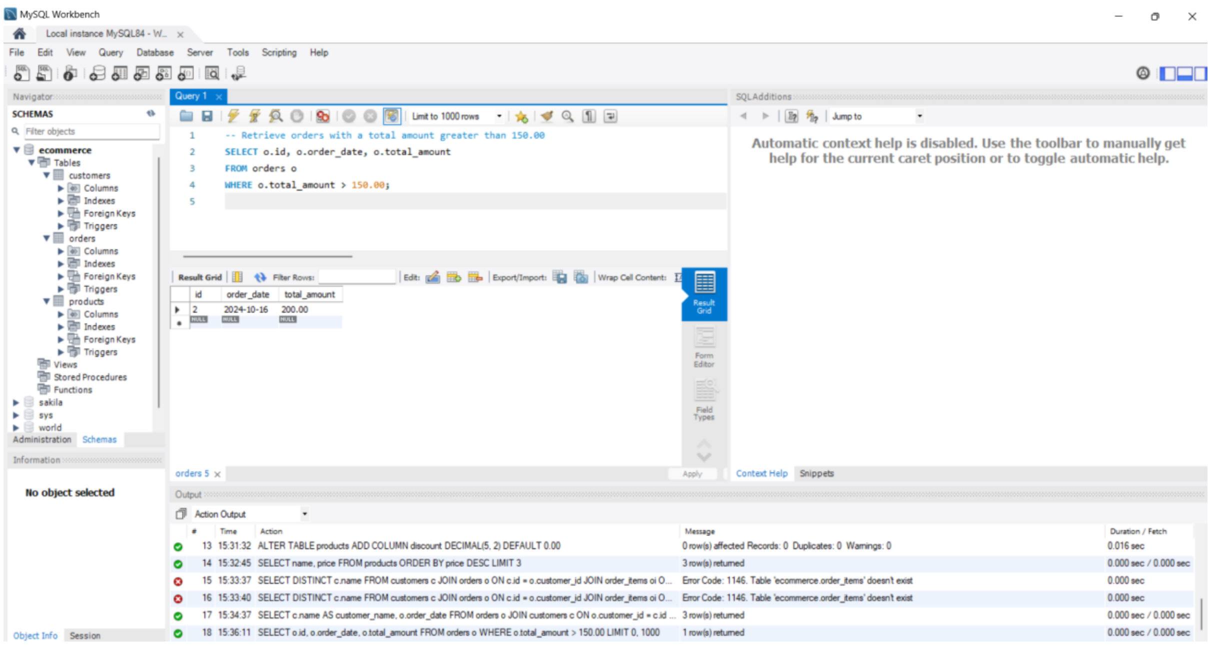
	customer_name	order_date
▶	Alice Johnson	2024-10-15
	Bob Smith	2024-10-16
	Charlie Brown	2024-10-10

	#	Time	Action	Message	Duration / Fetch
✗	12	15:30:32	UPDATE products SET price = 45.00 WHERE name = 'Product C'	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...	0.000 sec
✓	13	15:31:32	ALTER TABLE products ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
✓	14	15:32:45	SELECT name, price FROM products ORDER BY price DESC LIMIT 3	3 row(s) returned	0.000 sec / 0.000 sec
✗	15	15:33:37	SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...	Error Code: 1146. Table 'ecommerce.order_items' doesn't exist	0.000 sec
✗	16	15:33:40	SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...	Error Code: 1146. Table 'ecommerce.order_items' doesn't exist	0.000 sec
✓	17	15:34:37	SELECT c.name AS customer_name, o.order_date FROM orders o JOIN customers c ON o.customer_id = c.id ...	3 row(s) returned	0.000 sec / 0.000 sec

Retrieve the orders with a total amount greater than 150.00

```
-- Retrieve orders with a total amount greater than 150.00

SELECT o.id, o.order_date, o.total_amount

FROM orders o

WHERE o.total_amount > 150.00;
```

Output:



```
-- Retrieve orders with a total amount greater than 150.00
SELECT o.id, o.order_date, o.total_amount
FROM orders o
WHERE o.total_amount > 150.00;
```

id	order_date	total_amount
2	2024-10-16	200.00
NULL	NULL	NULL

```
#   Time      Action                                                                          Message                                                          Duration / Fetch
13  15:31:32  ALTER TABLE products ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00              0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0          0.016 sec
14  15:32:45  SELECT name, price FROM products ORDER BY price DESC LIMIT 3                     3 row(s) returned                                                0.000 sec / 0.000 sec
15  15:33:37  SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...   Error Code: 1146. Table 'ecommerce.order_items' doesn't exist   0.000 sec
16  15:33:40  SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...   Error Code: 1146. Table 'ecommerce.order_items' doesn't exist   0.000 sec
17  15:34:37  SELECT c.name AS customer_name, o.order_date FROM orders o JOIN customers c ON o.customer_id = c.id ...  3 row(s) returned                                0.000 sec / 0.000 sec
18  15:36:11  SELECT o.id, o.order_date, o.total_amount FROM orders o WHERE o.total_amount > 150.00 LIMIT 0, 1000    1 row(s) returned                                0.000 sec / 0.000 sec
```

Normalize the database by creating a separate table for order items and updating the orders table

-- Create the 'order_items' table to normalize the database

CREATE TABLE order_items (

  id INT AUTO_INCREMENT PRIMARY KEY,

  order_id INT,

  product_id INT,

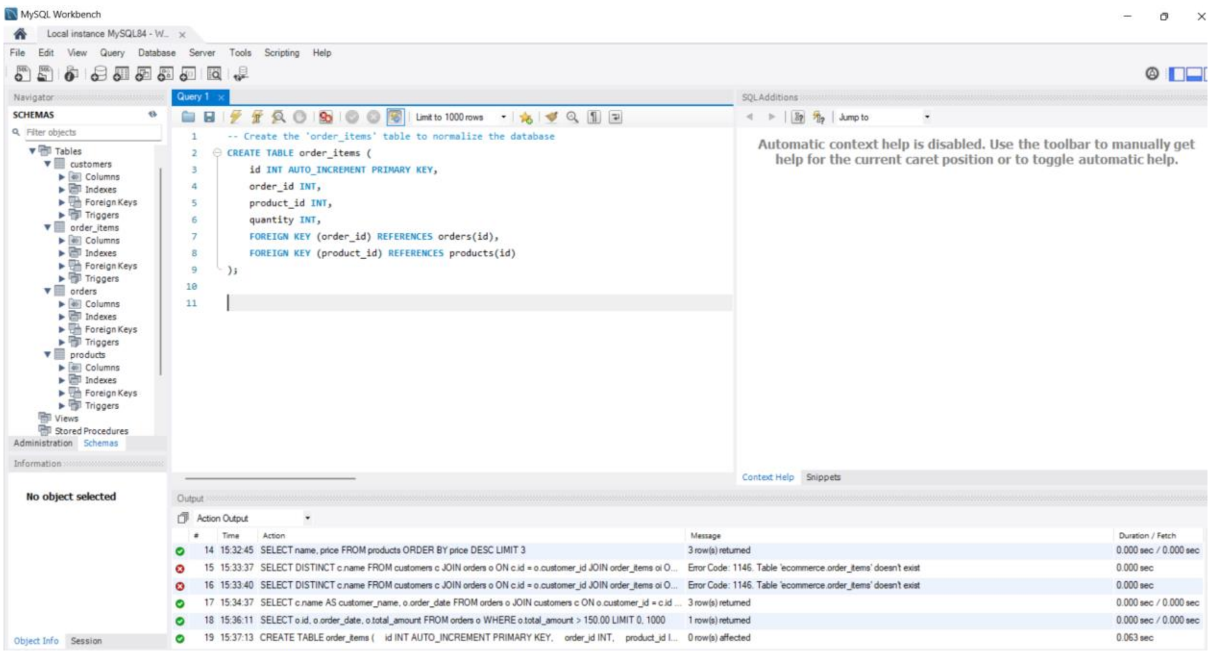
  quantity INT,

  FOREIGN KEY (order_id) REFERENCES orders(id),

  FOREIGN KEY (product_id) REFERENCES products(id)

);

Output:



```
-- Create the 'order_items' table to normalize the database
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)
);
```

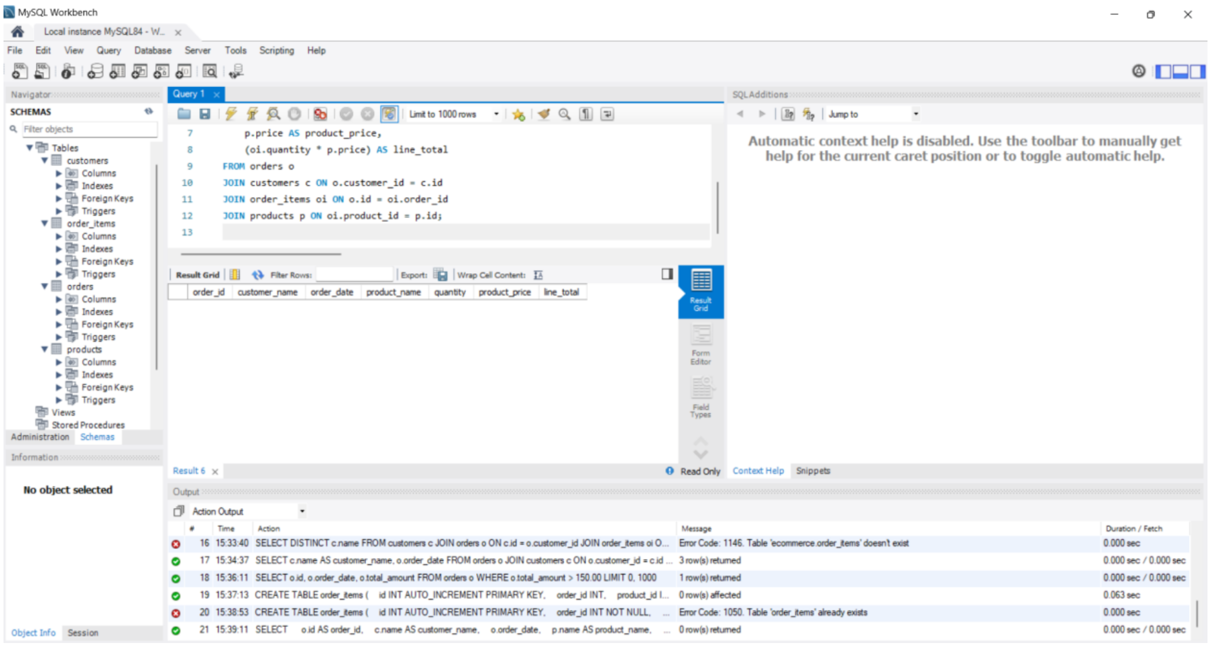
```
#   Time      Action                                                                                   Message                                                     Duration / Fetch
14  15:32:45  SELECT name, price FROM products ORDER BY price DESC LIMIT 3                              3 row(s) returned                                           0.000 sec / 0.000 sec
15  15:33:37  SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...  Error Code: 1146. Table 'ecommerce.order_items' doesn't exist  0.000 sec
16  15:33:40  SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...  Error Code: 1146. Table 'ecommerce.order_items' doesn't exist  0.000 sec
17  15:34:37  SELECT c.name AS customer_name, o.order_date FROM orders o JOIN customers c ON o.customer_id = c.id ...  3 row(s) returned          0.000 sec / 0.000 sec
18  15:36:11  SELECT o.id, o.order_date, o.total_amount FROM orders o WHERE o.total_amount > 150.00 LIMIT 0, 1000  1 row(s) returned            0.000 sec / 0.000 sec
19  15:37:13  CREATE TABLE order_items (    id INT AUTO_INCREMENT PRIMARY KEY,    order_id INT,    product_id I...  0 row(s) affected           0.063 sec
```

Query to Join All Related Tables (Example)

Retrieve full details for each order, including customer name, order date, product name, and quantity:

```
SELECT
    o.id AS order_id,
    c.name AS customer_name,
    o.order_date,
    p.name AS product_name,
    oi.quantity,
    p.price AS product_price,
    (oi.quantity * p.price) AS line_total
FROM orders o
JOIN customers c ON o.customer_id = c.id
JOIN order_items oi ON o.id = oi.order_id
JOIN products p ON oi.product_id = p.id;
```

Output:



```
    p.price AS product_price,
    (oi.quantity * p.price) AS line_total
FROM orders o
JOIN customers c ON o.customer_id = c.id
JOIN order_items oi ON o.id = oi.order_id
JOIN products p ON oi.product_id = p.id;
```

	order_id	customer_name	order_date	product_name	quantity	product_price	line_total

```
#   Time      Action                                                                          Message                                                   Duration / Fetch
16  15:33:40  SELECT DISTINCT c.name FROM customers c JOIN orders o ON c.id = o.customer_id JOIN order_items oi O...  Error Code: 1146. Table 'ecommerce.order_items' doesn't exist  0.000 sec
17  15:34:37  SELECT c.name AS customer_name, o.order_date FROM orders o JOIN customers c ON o.customer_id = c.id ...  3 row(s) returned  0.000 sec / 0.000 sec
18  15:36:11  SELECT o.id, o.order_date, o.total_amount FROM orders o WHERE o.total_amount > 150.00 LIMIT 0, 1000  1 row(s) returned  0.000 sec / 0.000 sec
19  15:37:13  CREATE TABLE order_items (   id INT AUTO_INCREMENT PRIMARY KEY,   order_id INT,   product_id I...  0 row(s) affected  0.063 sec
20  15:38:53  CREATE TABLE order_items (   id INT AUTO_INCREMENT PRIMARY KEY,   order_id INT NOT NULL,   ...  Error Code: 1050. Table 'order_items' already exists  0.000 sec
21  15:39:11  SELECT   o.id AS order_id,   c.name AS customer_name,   o.order_date,   p.name AS product_name,   ...  0 row(s) returned  0.000 sec / 0.000 sec
```

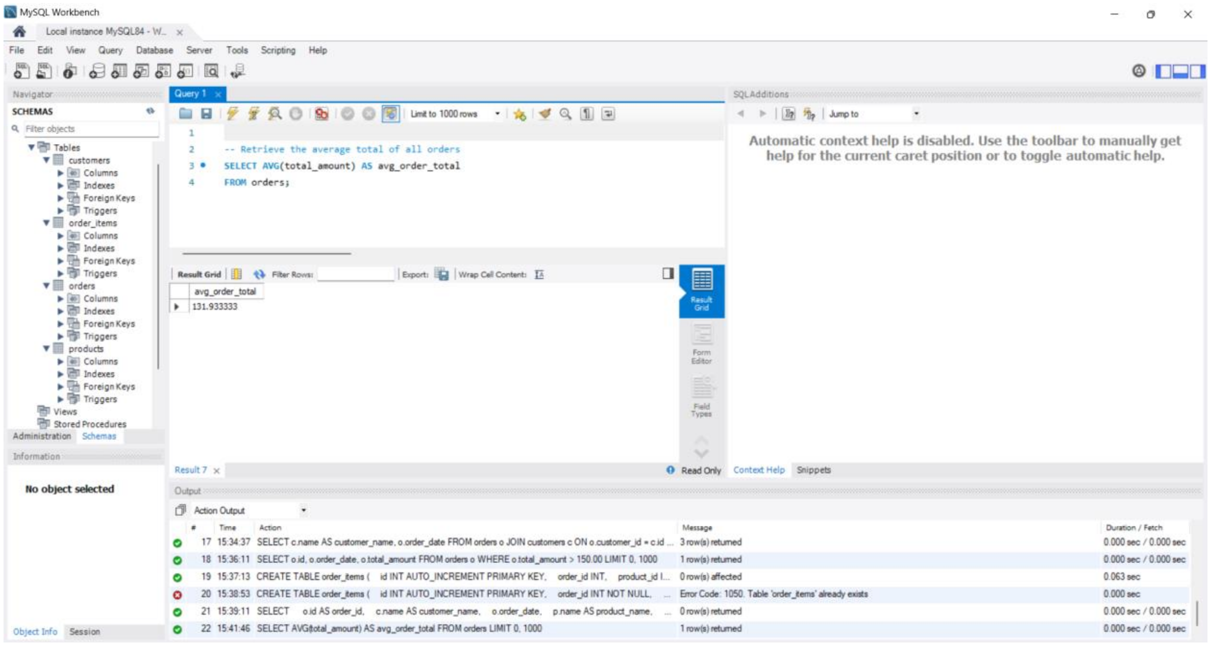
Retrieve the average total of all orders

-- Retrieve the average total of all orders

SELECT AVG(total_amount) AS avg_order_total

FROM orders;

Output:



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```sql
-- Retrieve the average total of all orders
SELECT AVG(total_amount) AS avg_order_total
FROM orders;
```

	avg_order_total
▶	131.933333

	#	Time	Action	Message	Duration / Fetch
✓	17	15:34:37	SELECT c.name AS customer_name, o.order_date FROM orders o JOIN customers c ON o.customer_id = c.id ...	3 row(s) returned	0.000 sec / 0.000 sec
✓	18	15:36:11	SELECT o.id, o.order_date, o.total_amount FROM orders o WHERE o.total_amount > 150.00 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓	19	15:37:13	CREATE TABLE order_items (id INT AUTO_INCREMENT PRIMARY KEY, order_id INT, product_id I...	0 row(s) affected	0.063 sec
✗	20	15:38:53	CREATE TABLE order_items (id INT AUTO_INCREMENT PRIMARY KEY, order_id INT NOT NULL, ...	Error Code: 1050. Table 'order_items' already exists	0.000 sec
✓	21	15:39:11	SELECT o.id AS order_id, c.name AS customer_name, o.order_date, p.name AS product_name, ...	0 row(s) returned	0.000 sec / 0.000 sec
✓	22	15:41:46	SELECT AVG(total_amount) AS avg_order_total FROM orders LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec