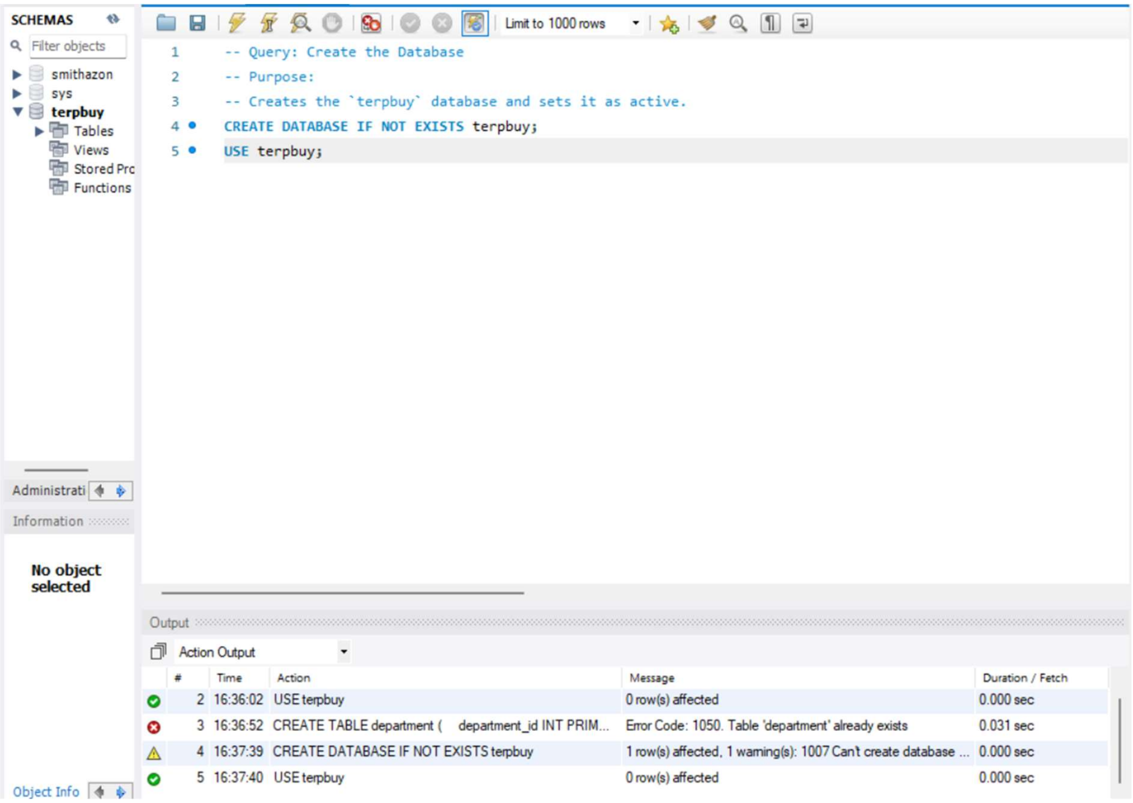
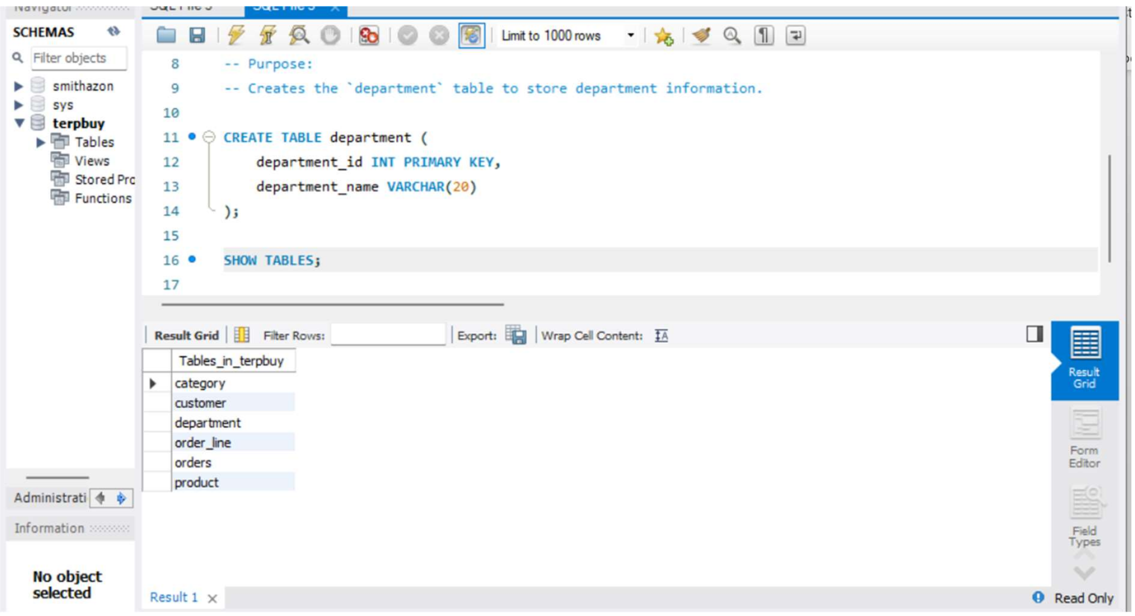


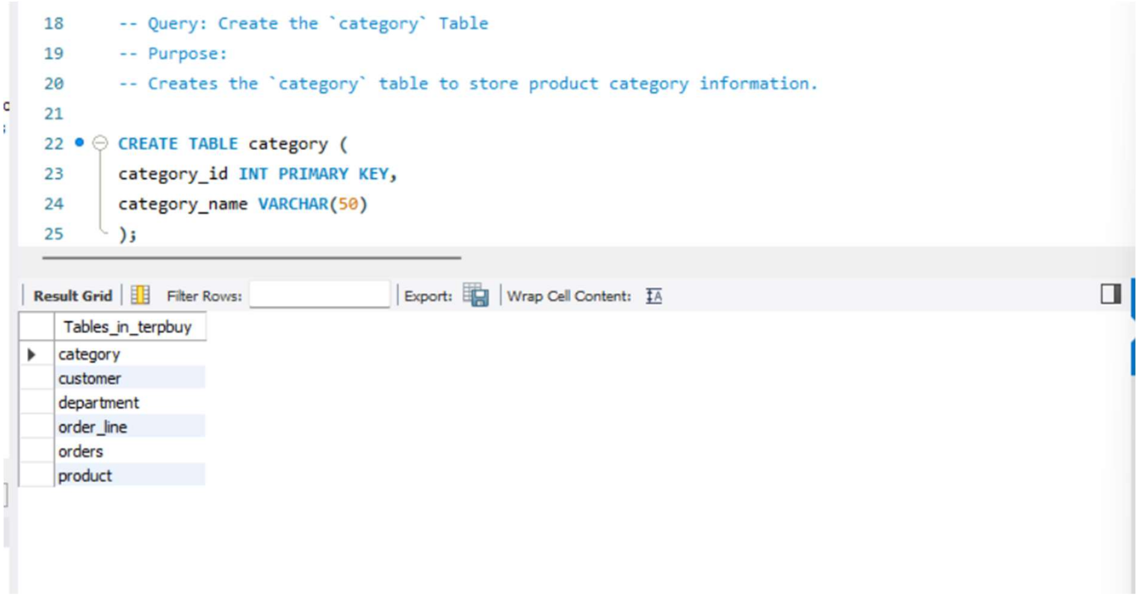
```sql
-- Query: Create the Database
-- Purpose:
-- Creates the `terpbuy` database and sets it as active.
CREATE DATABASE IF NOT EXISTS terpbuy;
USE terpbuy;
```

#	Time	Action	Message	Duration / Fetch
2	16:36:02	USE terpbuy	0 row(s) affected	0.000 sec
3	16:36:52	CREATE TABLE department (department_id INT PRIM...	Error Code: 1050. Table 'department' already exists	0.031 sec
4	16:37:39	CREATE DATABASE IF NOT EXISTS terpbuy	1 row(s) affected, 1 warning(s): 1007 Can't create database ...	0.000 sec
5	16:37:40	USE terpbuy	0 row(s) affected	0.000 sec



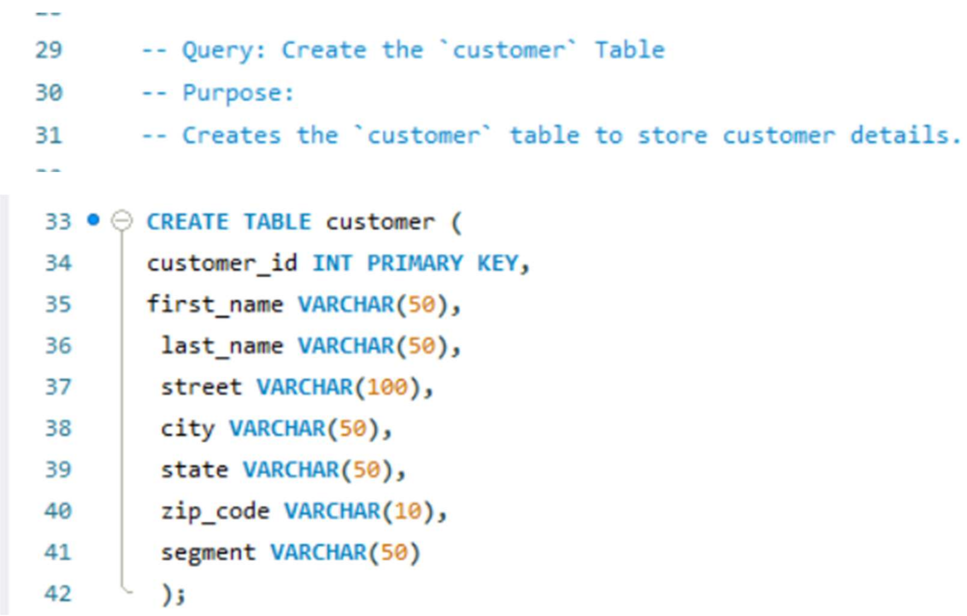
```sql
-- Purpose:
-- Creates the `department` table to store department information.

CREATE TABLE department (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(20)
);

SHOW TABLES;

```

Tables_in_terpbuy
category
customer
department
order_line
orders
product

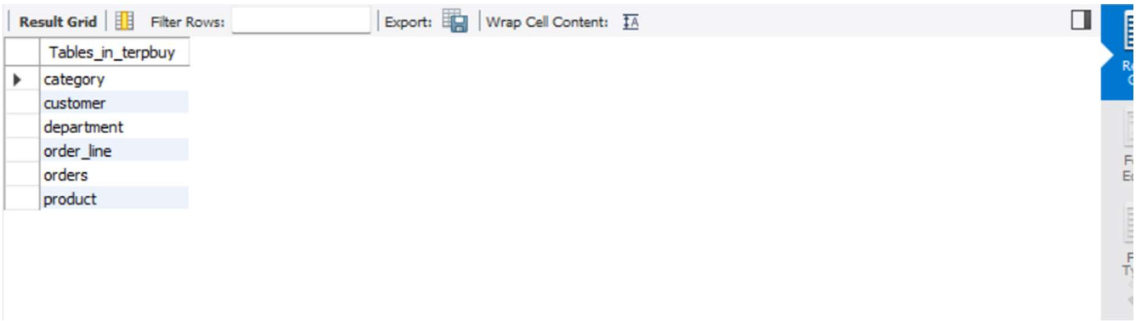


```sql
-- Query: Create the `category` Table
-- Purpose:
-- Creates the `category` table to store product category information.

CREATE TABLE category (
category_id INT PRIMARY KEY,
category_name VARCHAR(50)
);
```

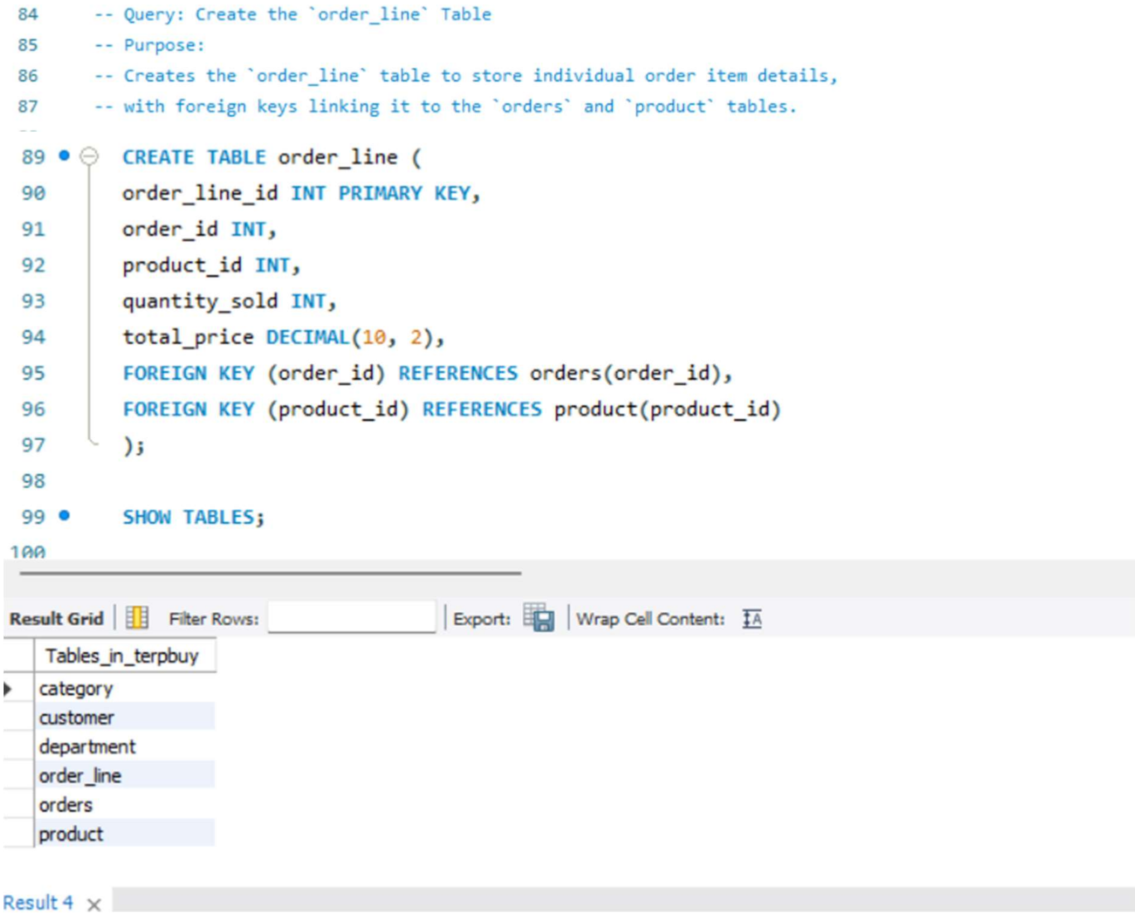
Tables_in_terpbuy
category
customer
department
order_line
orders
product



```sql
--
-- Query: Create the `customer` Table
-- Purpose:
-- Creates the `customer` table to store customer details.
--
CREATE TABLE customer (
customer_id INT PRIMARY KEY,
first_name VARCHAR(50),
 last_name VARCHAR(50),
 street VARCHAR(100),
 city VARCHAR(50),
 state VARCHAR(50),
 zip_code VARCHAR(10),
 segment VARCHAR(50)
 );
```

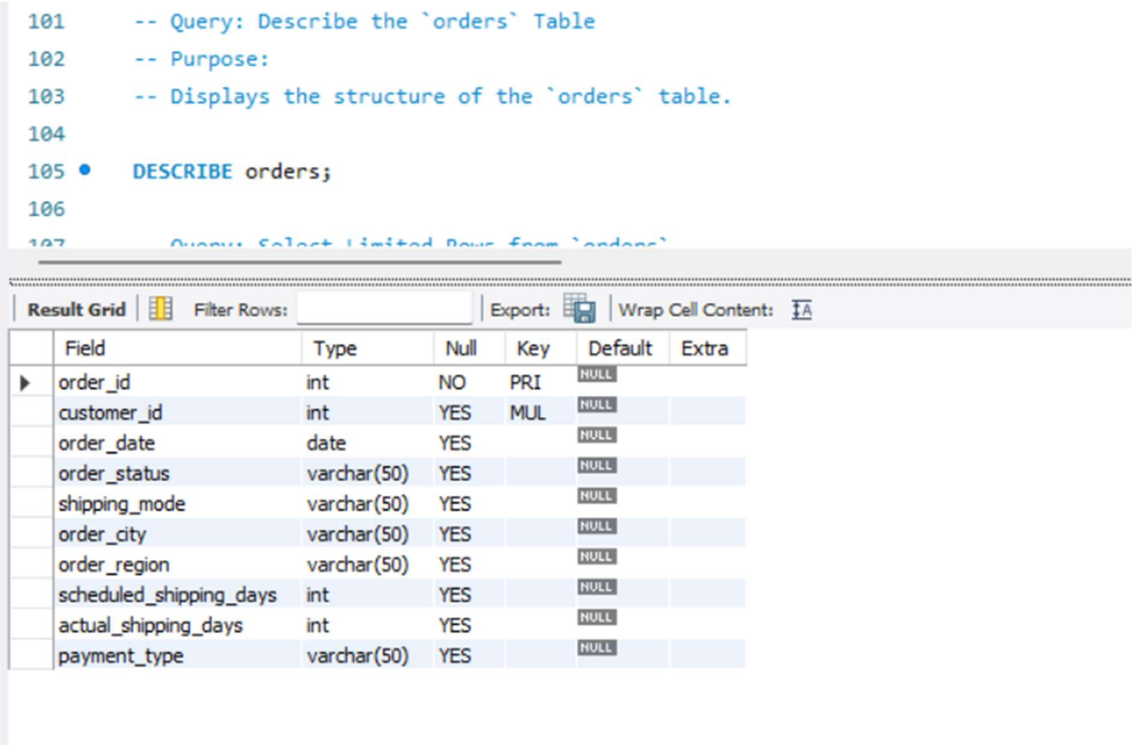


Tables_in_terpbuy
category
customer
department
order_line
orders
product



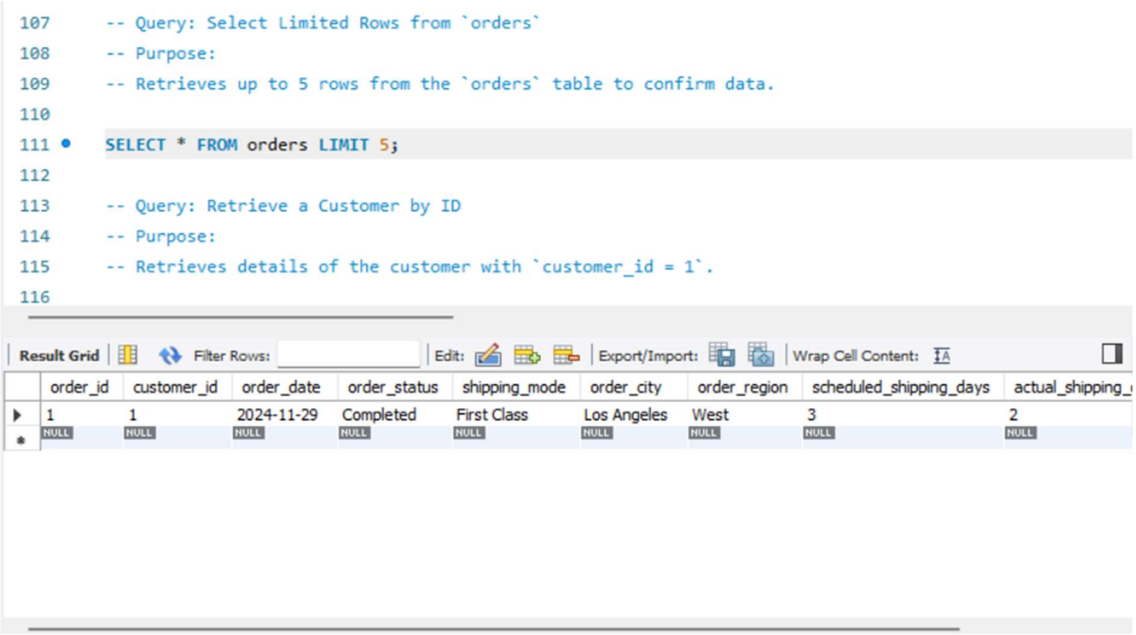
```sql
-- Query: Create the `order_line` Table
-- Purpose:
-- Creates the `order_line` table to store individual order item details,
-- with foreign keys linking it to the `orders` and `product` tables.

CREATE TABLE order_line (
order_line_id INT PRIMARY KEY,
order_id INT,
product_id INT,
quantity_sold INT,
total_price DECIMAL(10, 2),
FOREIGN KEY (order_id) REFERENCES orders(order_id),
FOREIGN KEY (product_id) REFERENCES product(product_id)
);

SHOW TABLES;
```

Tables_in_terpbuy
category
customer
department
order_line
orders
product



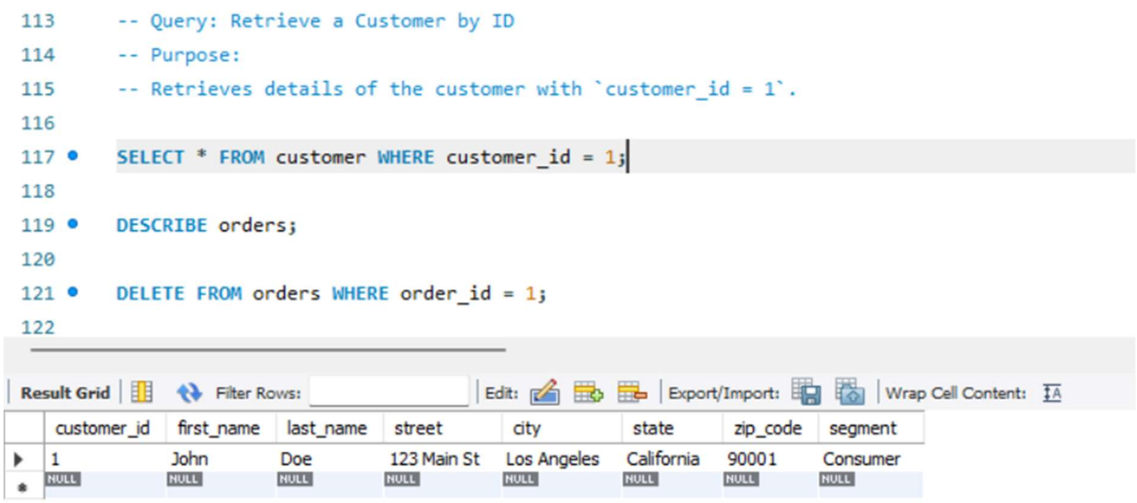
```sql
-- Query: Describe the `orders` Table
-- Purpose:
-- Displays the structure of the `orders` table.

DESCRIBE orders;

-- Query: Select Limited Rows from `orders`
```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	
order_status	varchar(50)	YES		NULL	
shipping_mode	varchar(50)	YES		NULL	
order_city	varchar(50)	YES		NULL	
order_region	varchar(50)	YES		NULL	
scheduled_shipping_days	int	YES		NULL	
actual_shipping_days	int	YES		NULL	
payment_type	varchar(50)	YES		NULL	



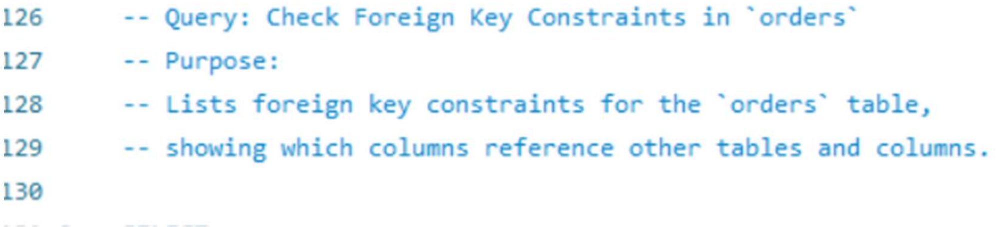
```sql
-- Query: Select Limited Rows from `orders`
-- Purpose:
-- Retrieves up to 5 rows from the `orders` table to confirm data.

SELECT * FROM orders LIMIT 5;

-- Query: Retrieve a Customer by ID
-- Purpose:
-- Retrieves details of the customer with `customer_id = 1`.

```

order_id	customer_id	order_date	order_status	shipping_mode	order_city	order_region	scheduled_shipping_days	actual_shipping_
1	1	2024-11-29	Completed	First Class	Los Angeles	West	3	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

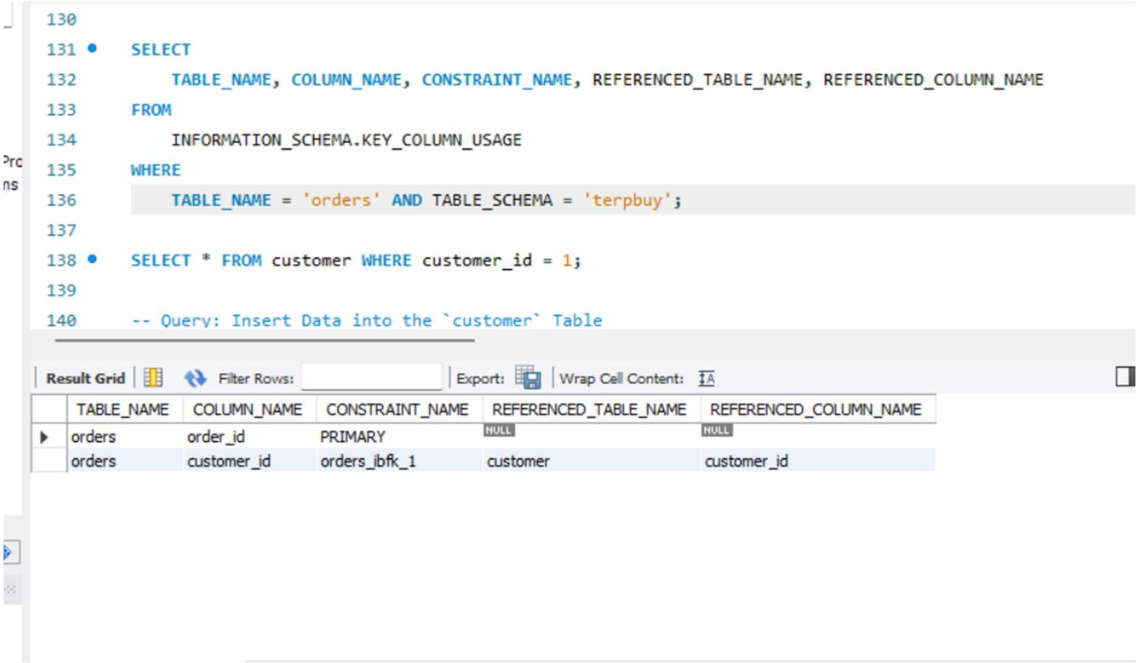


```sql
-- Query: Retrieve a Customer by ID
-- Purpose:
-- Retrieves details of the customer with `customer_id = 1`.

SELECT * FROM customer WHERE customer_id = 1;

DESCRIBE orders;

DELETE FROM orders WHERE order_id = 1;

```

customer_id	first_name	last_name	street	city	state	zip_code	segment
1	John	Doe	123 Main St	Los Angeles	California	90001	Consumer
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

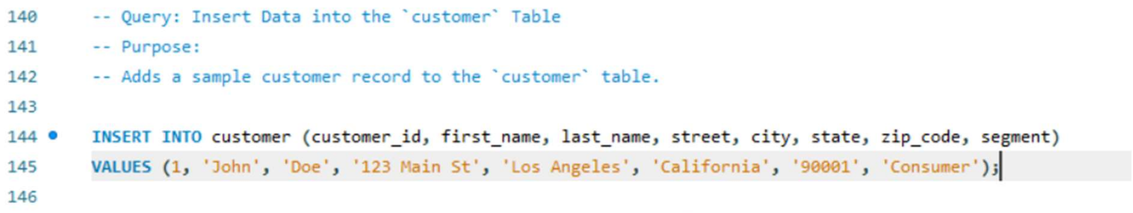


```sql
-- Query: Check Foreign Key Constraints in `orders`
-- Purpose:
-- Lists foreign key constraints for the `orders` table,
-- showing which columns reference other tables and columns.

```

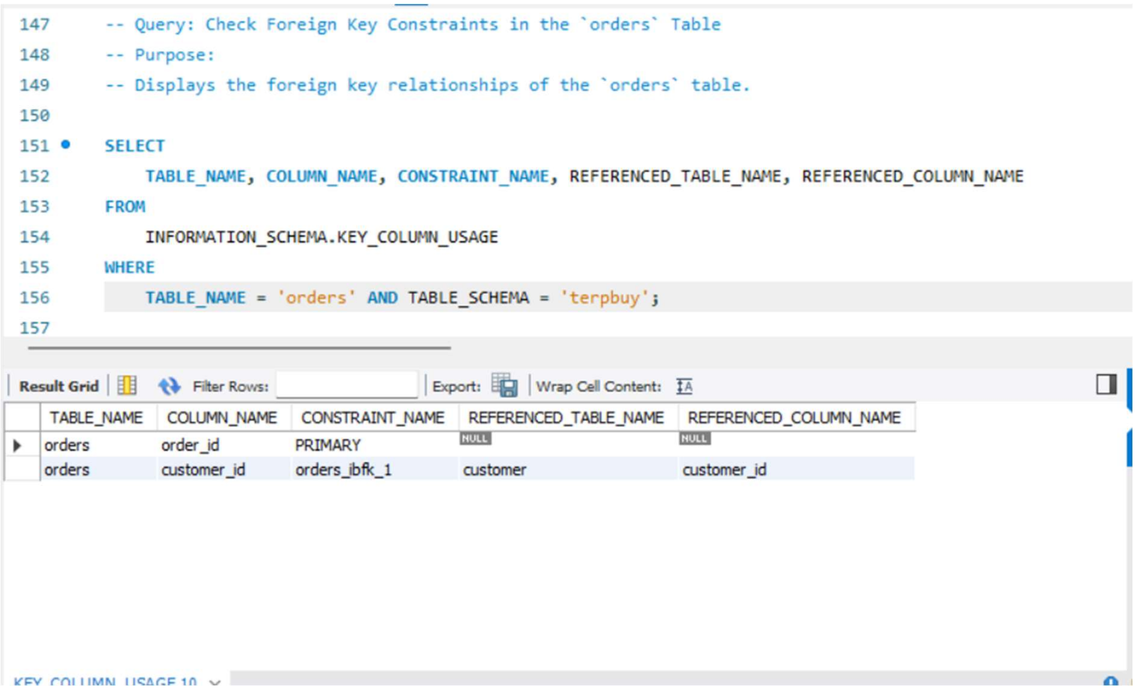


```sql

SELECT
    TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
FROM
    INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE
    TABLE_NAME = 'orders' AND TABLE_SCHEMA = 'terpbuy';

SELECT * FROM customer WHERE customer_id = 1;

-- Query: Insert Data into the `customer` Table
```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME	REFERENCED_COLUMN_NAME
orders	order_id	PRIMARY	NULL	NULL
orders	customer_id	orders_ibfk_1	customer	customer_id

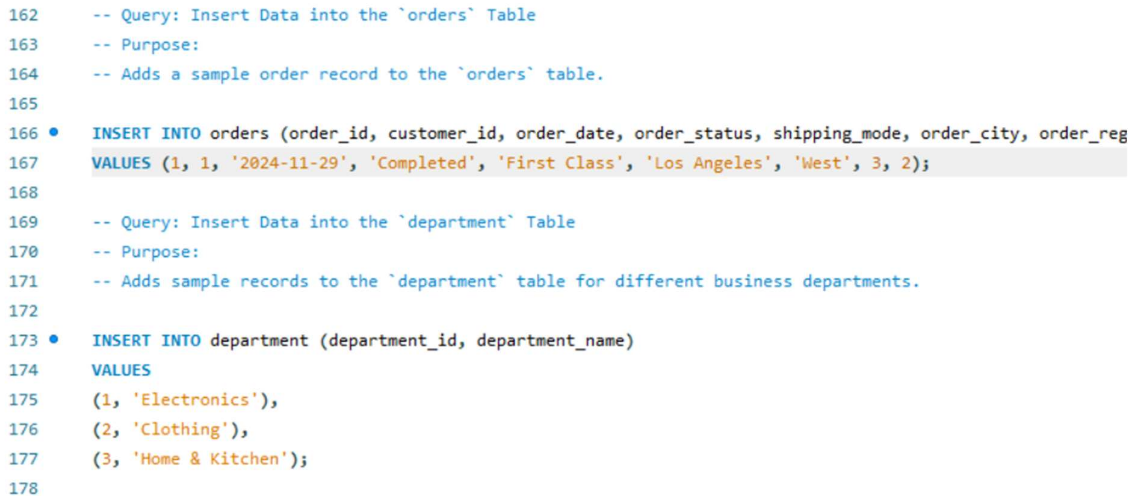


```sql
-- Query: Insert Data into the `customer` Table
-- Purpose:
-- Adds a sample customer record to the `customer` table.

INSERT INTO customer (customer_id, first_name, last_name, street, city, state, zip_code, segment)
VALUES (1, 'John', 'Doe', '123 Main St', 'Los Angeles', 'California', '90001', 'Consumer');

```

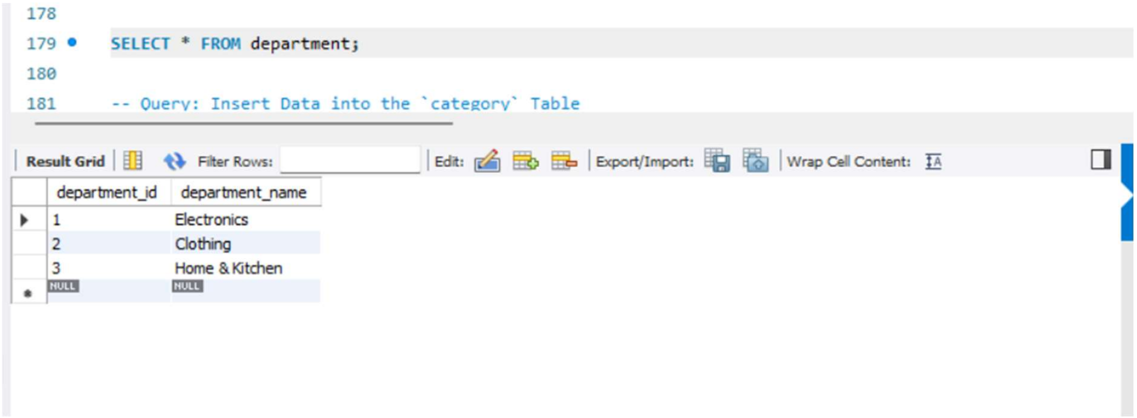


```sql
-- Query: Check Foreign Key Constraints in the `orders` Table
-- Purpose:
-- Displays the foreign key relationships of the `orders` table.

SELECT
    TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
FROM
    INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE
    TABLE_NAME = 'orders' AND TABLE_SCHEMA = 'terpbuy';

```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME	REFERENCED_COLUMN_NAME
orders	order_id	PRIMARY	NULL	NULL
orders	customer_id	orders_ibfk_1	customer	customer_id

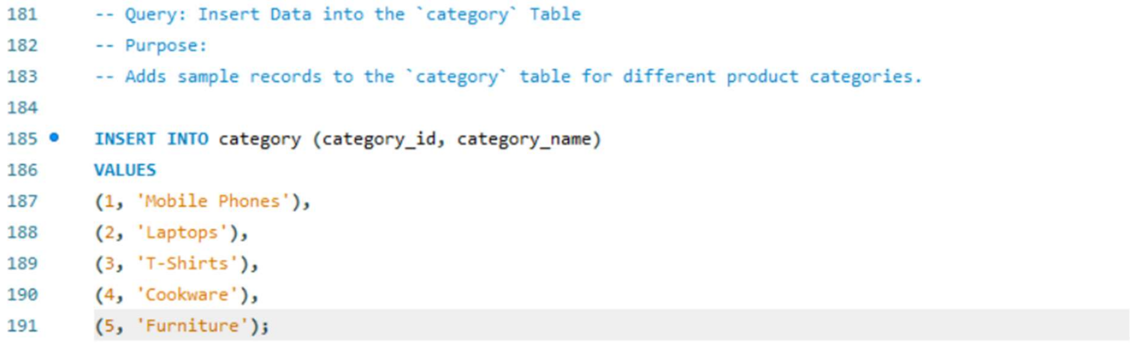


```sql
-- Query: Insert Data into the `orders` Table
-- Purpose:
-- Adds a sample order record to the `orders` table.

INSERT INTO orders (order_id, customer_id, order_date, order_status, shipping_mode, order_city, order_reg
VALUES (1, 1, '2024-11-29', 'Completed', 'First Class', 'Los Angeles', 'West', 3, 2);

-- Query: Insert Data into the `department` Table
-- Purpose:
-- Adds sample records to the `department` table for different business departments.

INSERT INTO department (department_id, department_name)
VALUES
(1, 'Electronics'),
(2, 'Clothing'),
(3, 'Home & Kitchen');

```

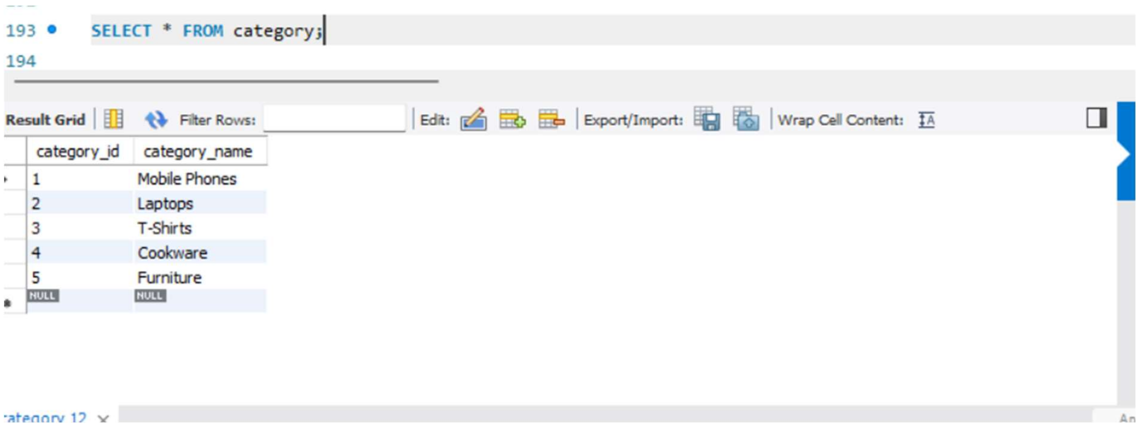


```sql

SELECT * FROM department;

-- Query: Insert Data into the `category` Table
```

department_id	department_name
1	Electronics
2	Clothing
3	Home & Kitchen
NULL	NULL

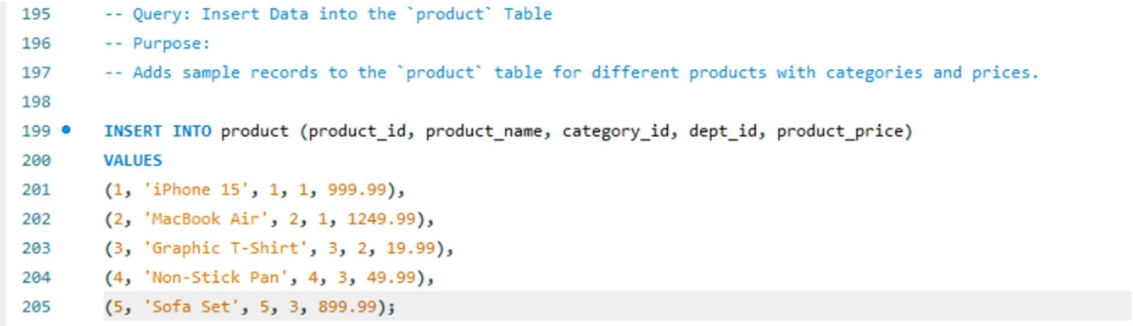


```sql
-- Query: Insert Data into the `category` Table
-- Purpose:
-- Adds sample records to the `category` table for different product categories.

INSERT INTO category (category_id, category_name)
VALUES
(1, 'Mobile Phones'),
(2, 'Laptops'),
(3, 'T-Shirts'),
(4, 'Cookware'),
(5, 'Furniture');
```

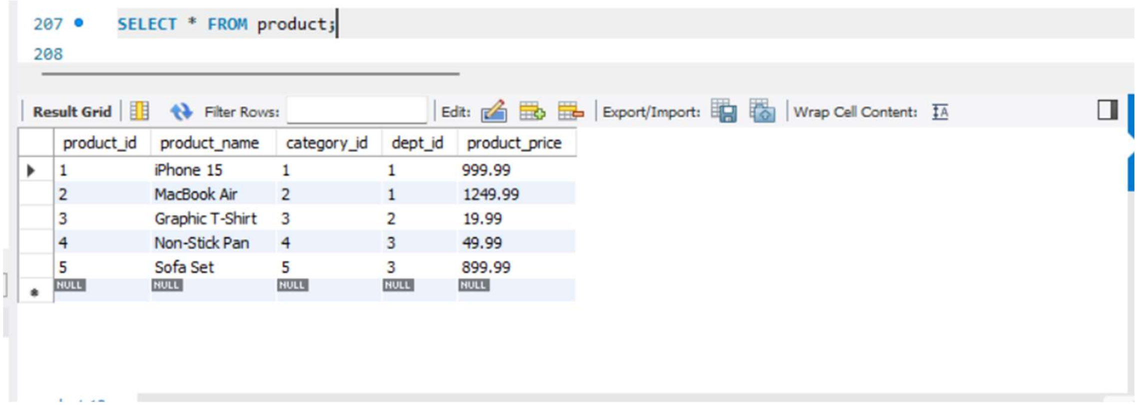


```sql
SELECT * FROM category;

```

category_id	category_name
1	Mobile Phones
2	Laptops
3	T-Shirts
4	Cookware
5	Furniture
NULL	NULL

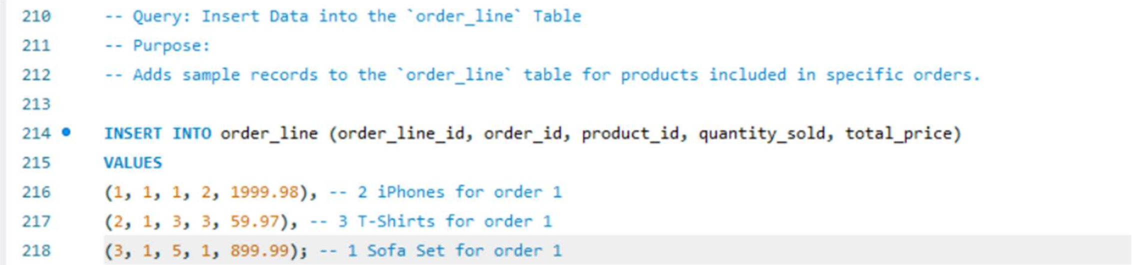


```sql
-- Query: Insert Data into the `product` Table
-- Purpose:
-- Adds sample records to the `product` table for different products with categories and prices.

INSERT INTO product (product_id, product_name, category_id, dept_id, product_price)
VALUES
(1, 'iPhone 15', 1, 1, 999.99),
(2, 'MacBook Air', 2, 1, 1249.99),
(3, 'Graphic T-Shirt', 3, 2, 19.99),
(4, 'Non-Stick Pan', 4, 3, 49.99),
(5, 'Sofa Set', 5, 3, 899.99);
```

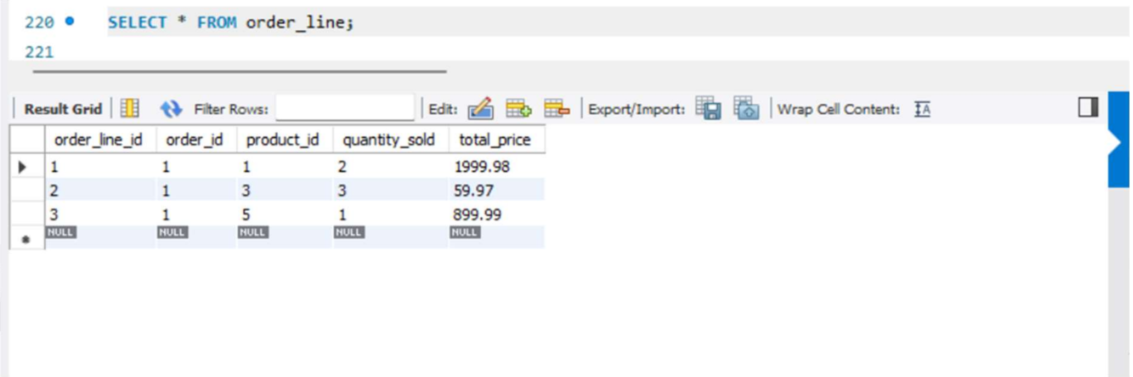


```sql
SELECT * FROM product;
```

product_id	product_name	category_id	dept_id	product_price
1	iPhone 15	1	1	999.99
2	MacBook Air	2	1	1249.99
3	Graphic T-Shirt	3	2	19.99
4	Non-Stick Pan	4	3	49.99
5	Sofa Set	5	3	899.99
NULL	NULL	NULL	NULL	NULL

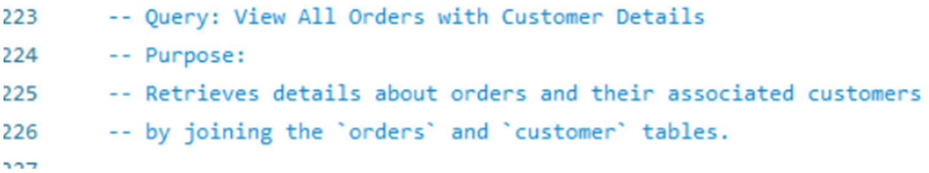


```sql
-- Query: Insert Data into the `order_line` Table
-- Purpose:
-- Adds sample records to the `order_line` table for products included in specific orders.

INSERT INTO order_line (order_line_id, order_id, product_id, quantity_sold, total_price)
VALUES
(1, 1, 1, 2, 1999.98), -- 2 iPhones for order 1
(2, 1, 3, 3, 59.97), -- 3 T-Shirts for order 1
(3, 1, 5, 1, 899.99); -- 1 Sofa Set for order 1
```

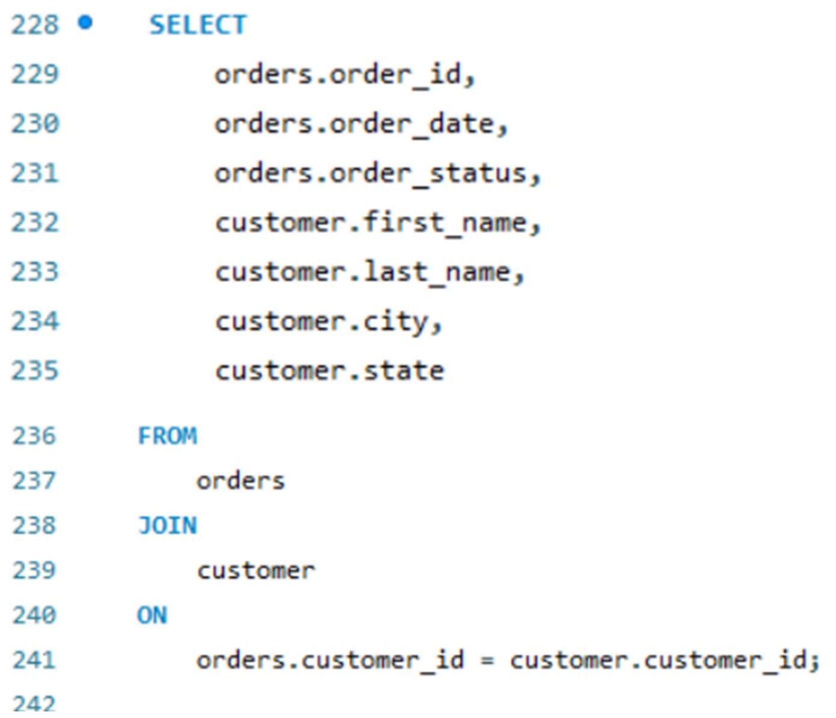


```sql
SELECT * FROM order_line;
```

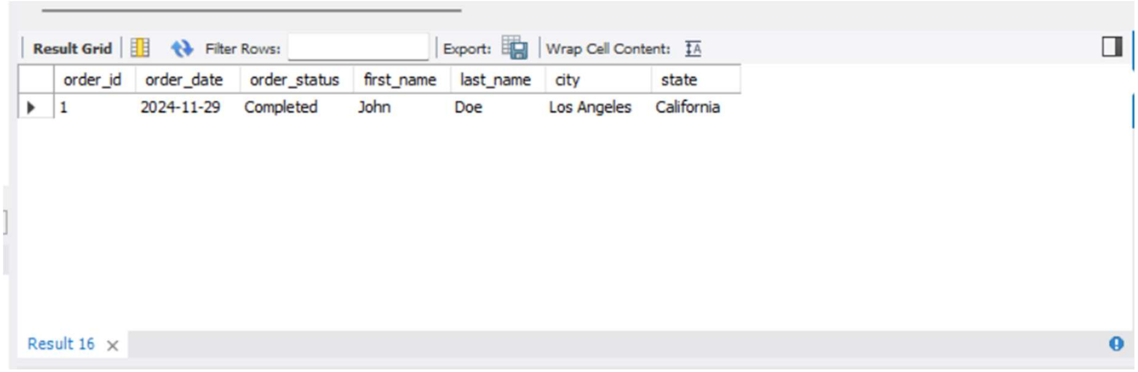
order_line_id	order_id	product_id	quantity_sold	total_price
1	1	1	2	1999.98
2	1	3	3	59.97
3	1	5	1	899.99
NULL	NULL	NULL	NULL	NULL



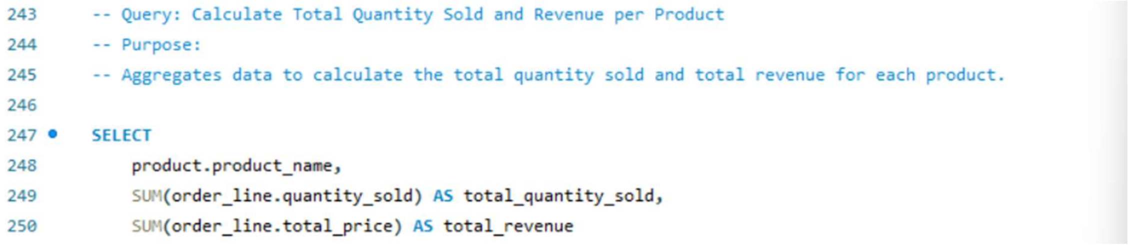
```sql
-- Query: View All Orders with Customer Details
-- Purpose:
-- Retrieves details about orders and their associated customers
-- by joining the `orders` and `customer` tables.
```



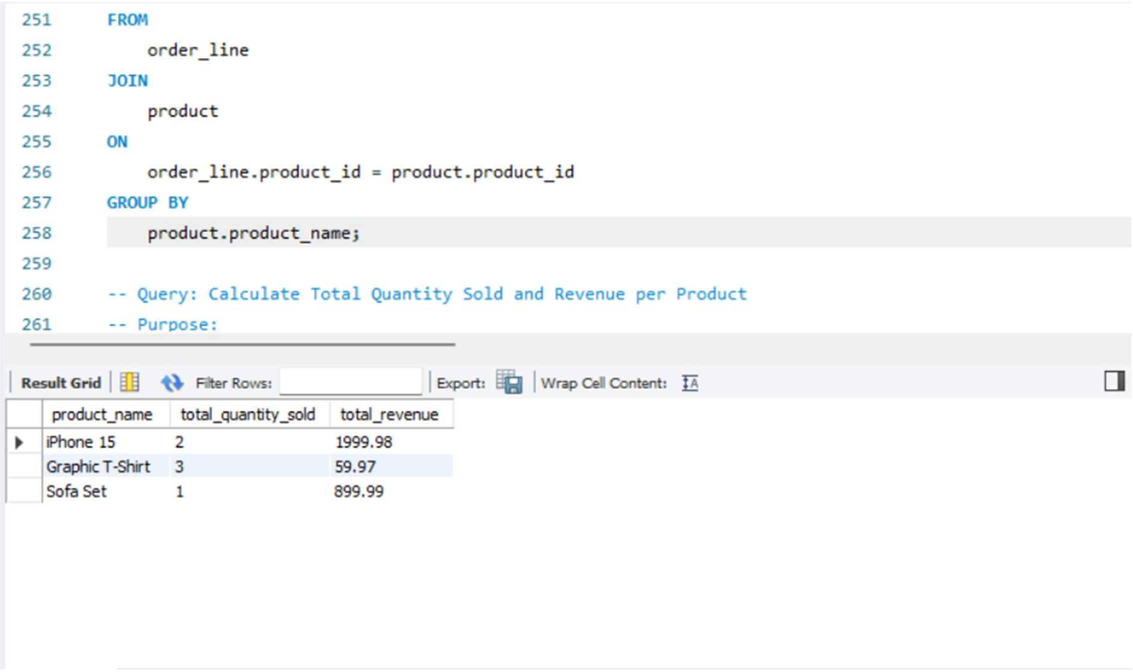
```sql
SELECT
    orders.order_id,
    orders.order_date,
    orders.order_status,
    customer.first_name,
    customer.last_name,
    customer.city,
    customer.state
FROM
    orders
JOIN
    customer
ON
    orders.customer_id = customer.customer_id;
```



order_id	order_date	order_status	first_name	last_name	city	state
1	2024-11-29	Completed	John	Doe	Los Angeles	California

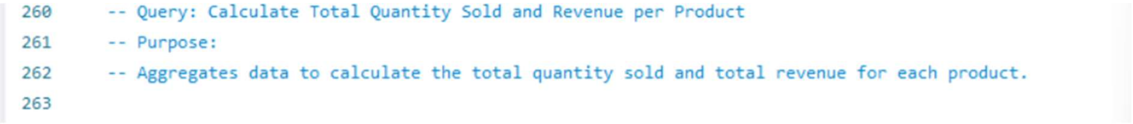


```sql
-- Query: Calculate Total Quantity Sold and Revenue per Product
-- Purpose:
-- Aggregates data to calculate the total quantity sold and total revenue for each product.

SELECT
    product.product_name,
    SUM(order_line.quantity_sold) AS total_quantity_sold,
    SUM(order_line.total_price) AS total_revenue
```

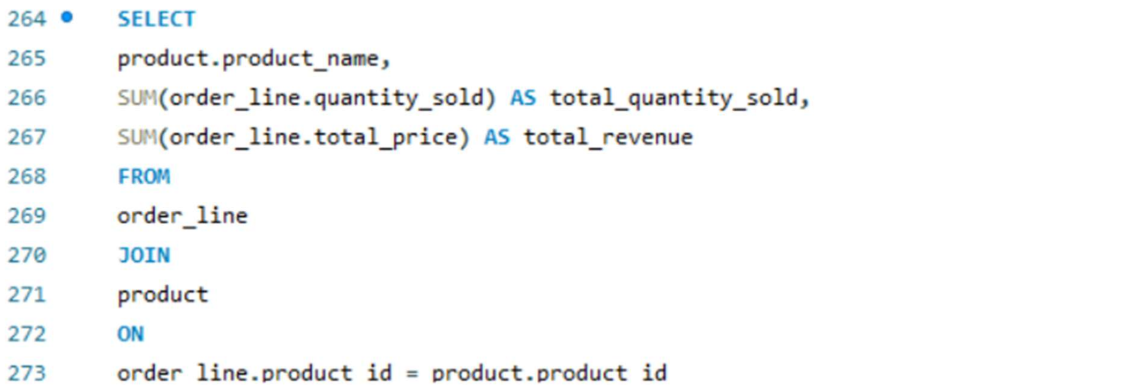


```sql
FROM
    order_line
JOIN
    product
ON
    order_line.product_id = product.product_id
GROUP BY
    product.product_name;

-- Query: Calculate Total Quantity Sold and Revenue per Product
-- Purpose:
```

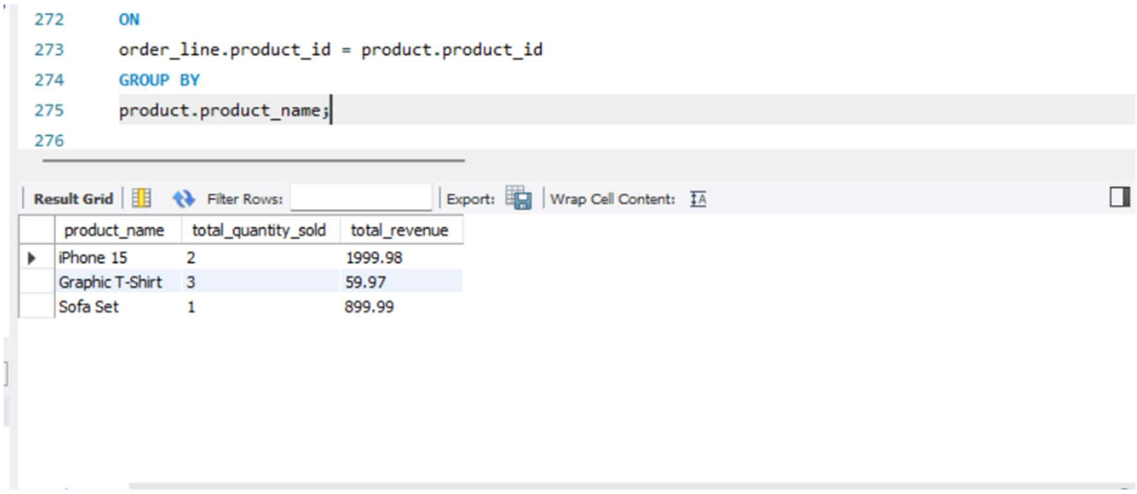
product_name	total_quantity_sold	total_revenue
iPhone 15	2	1999.98
Graphic T-Shirt	3	59.97
Sofa Set	1	899.99



```sql
-- Query: Calculate Total Quantity Sold and Revenue per Product
-- Purpose:
-- Aggregates data to calculate the total quantity sold and total revenue for each product.
```

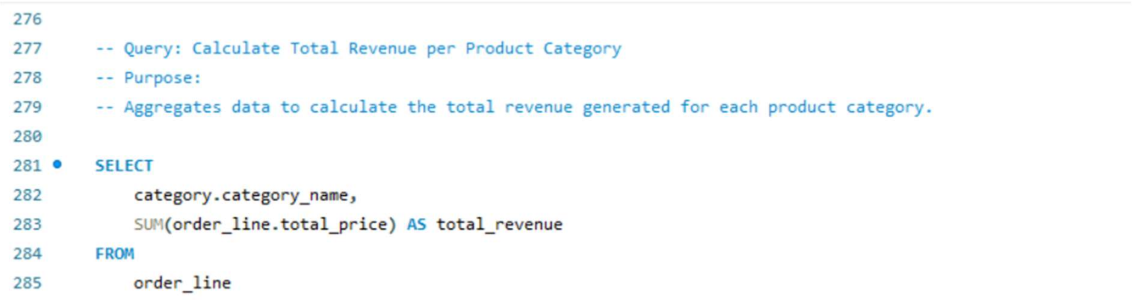


```sql
SELECT
product.product_name,
SUM(order_line.quantity_sold) AS total_quantity_sold,
SUM(order_line.total_price) AS total_revenue
FROM
order_line
JOIN
product
ON
order line.product id = product.product id
```

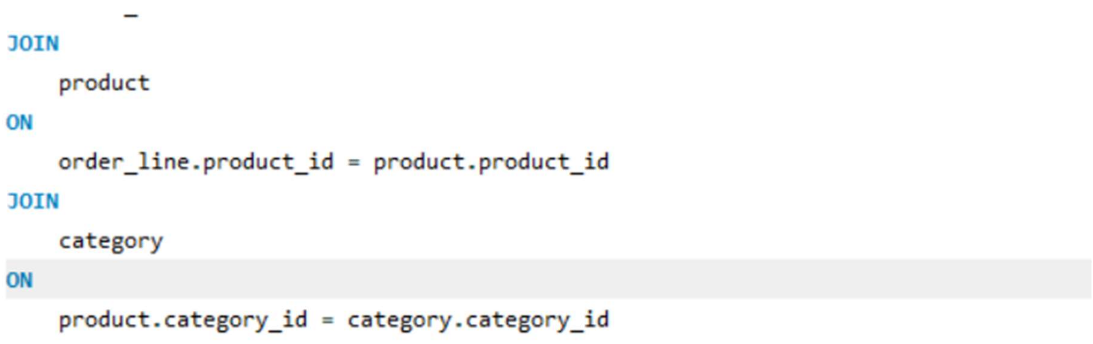


```sql
ON
order_line.product_id = product.product_id
GROUP BY
product.product_name;
```

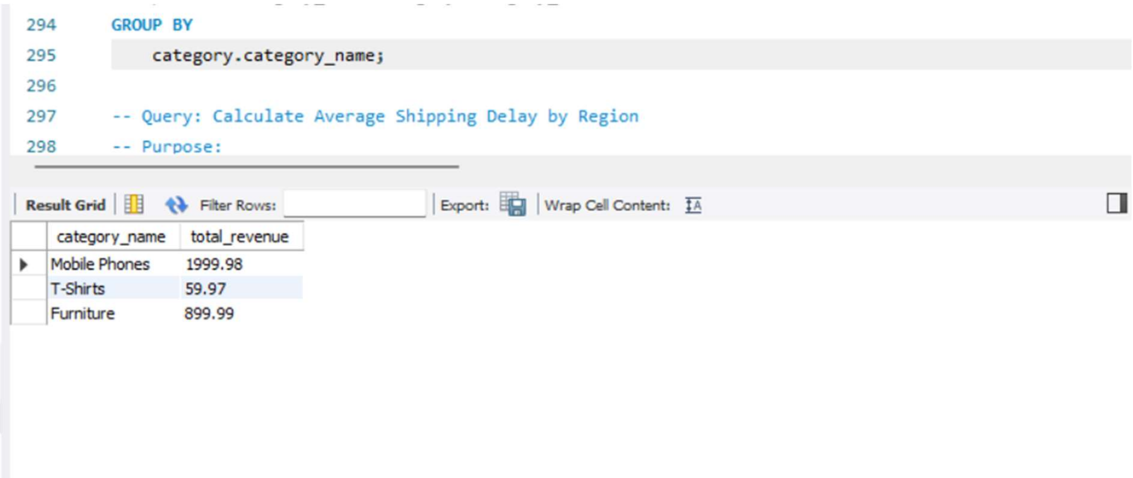
product_name	total_quantity_sold	total_revenue
iPhone 15	2	1999.98
Graphic T-Shirt	3	59.97
Sofa Set	1	899.99



```
-- Query: Calculate Total Revenue per Product Category
-- Purpose:
-- Aggregates data to calculate the total revenue generated for each product category.

SELECT
    category.category_name,
    SUM(order_line.total_price) AS total_revenue
FROM
    order_line
```

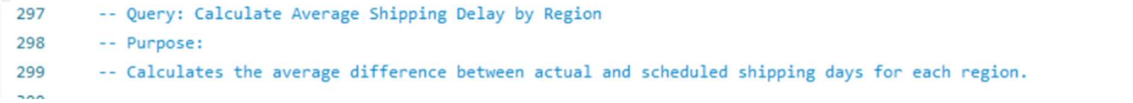


```
JOIN
    product
ON
    order_line.product_id = product.product_id
JOIN
    category
ON
    product.category_id = category.category_id
```

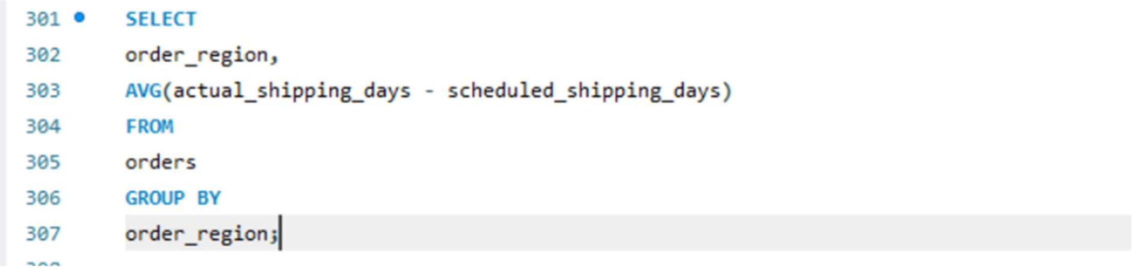


```
GROUP BY
    category.category_name;

-- Query: Calculate Average Shipping Delay by Region
-- Purpose:
```

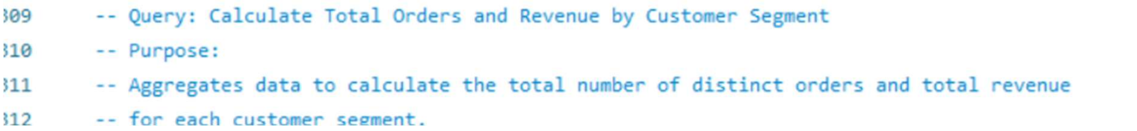
category_name	total_revenue
Mobile Phones	1999.98
T-Shirts	59.97
Furniture	899.99



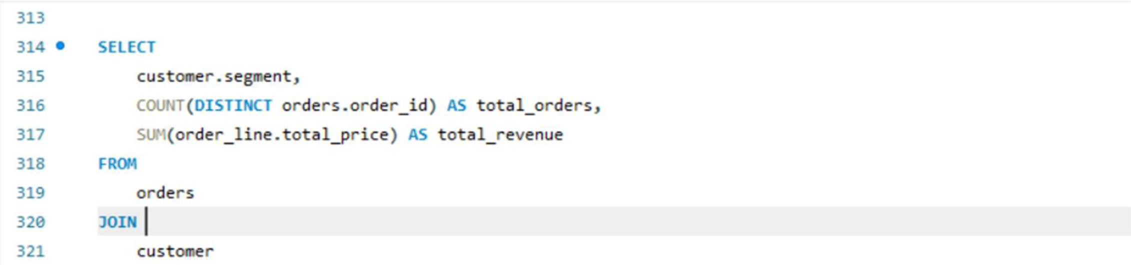
```
-- Query: Calculate Average Shipping Delay by Region
-- Purpose:
-- Calculates the average difference between actual and scheduled shipping days for each region.
```



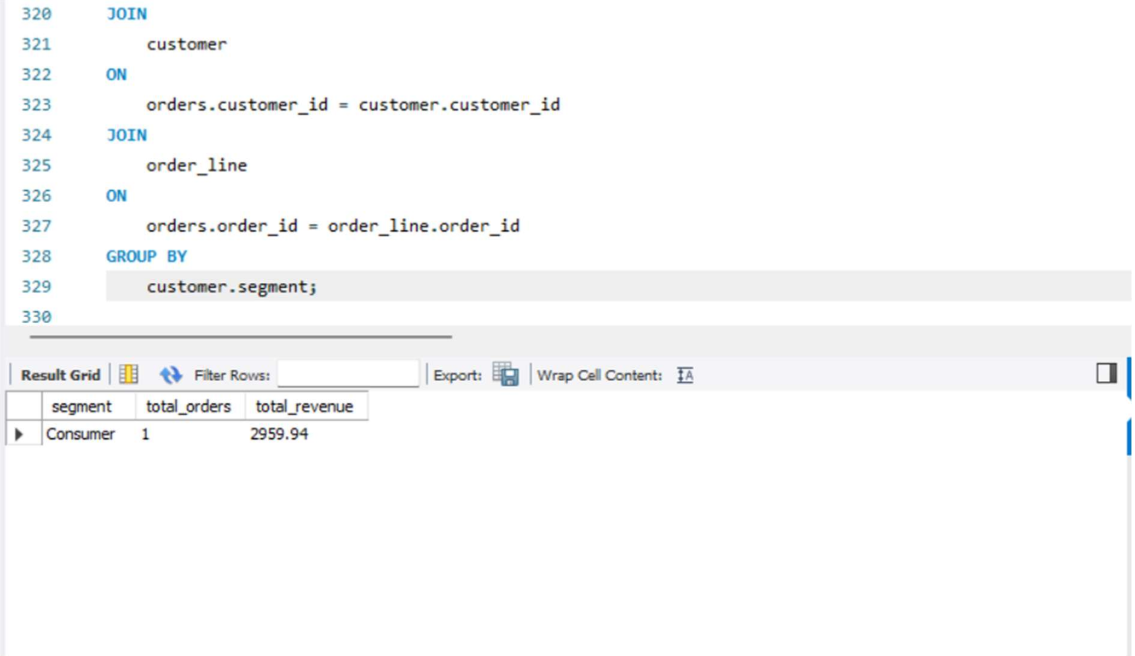
```
SELECT
order_region,
AVG(actual_shipping_days - scheduled_shipping_days)
FROM
orders
GROUP BY
order_region;
```



```
-- Query: Calculate Total Orders and Revenue by Customer Segment
-- Purpose:
-- Aggregates data to calculate the total number of distinct orders and total revenue
-- for each customer segment.
```



```
SELECT
    customer.segment,
    COUNT(DISTINCT orders.order_id) AS total_orders,
    SUM(order_line.total_price) AS total_revenue
FROM
    orders
JOIN
    customer
```



```
JOIN
    customer
ON
    orders.customer_id = customer.customer_id
JOIN
    order_line
ON
    orders.order_id = order_line.order_id
GROUP BY
    customer.segment;
```

segment	total_orders	total_revenue
Consumer	1	2959.94