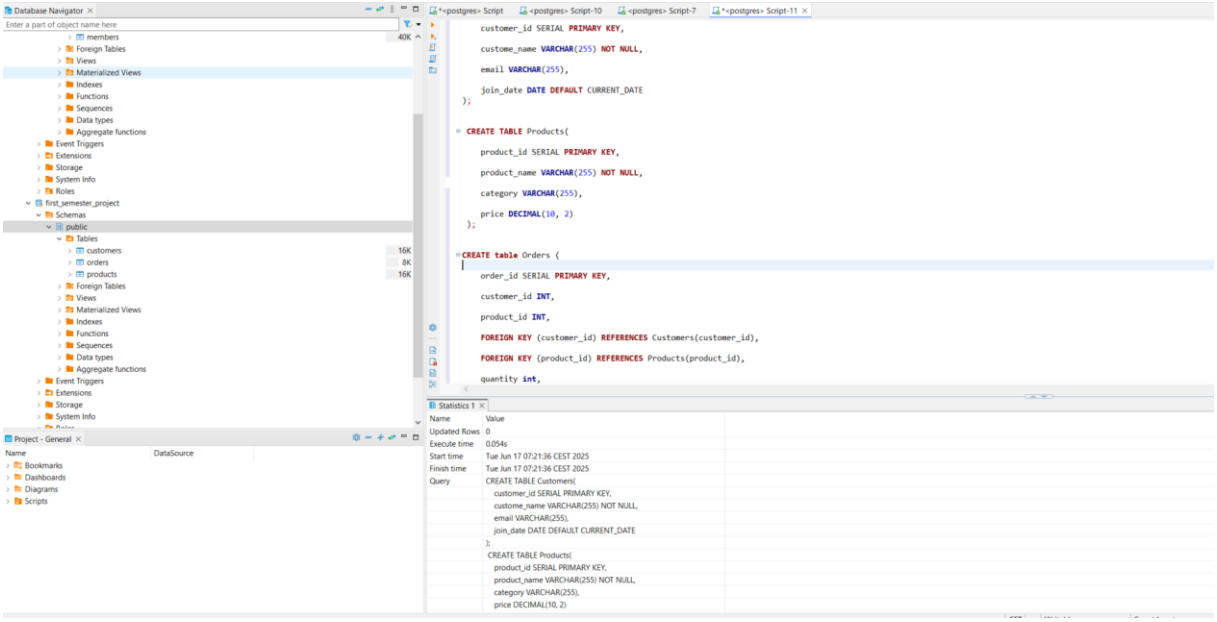
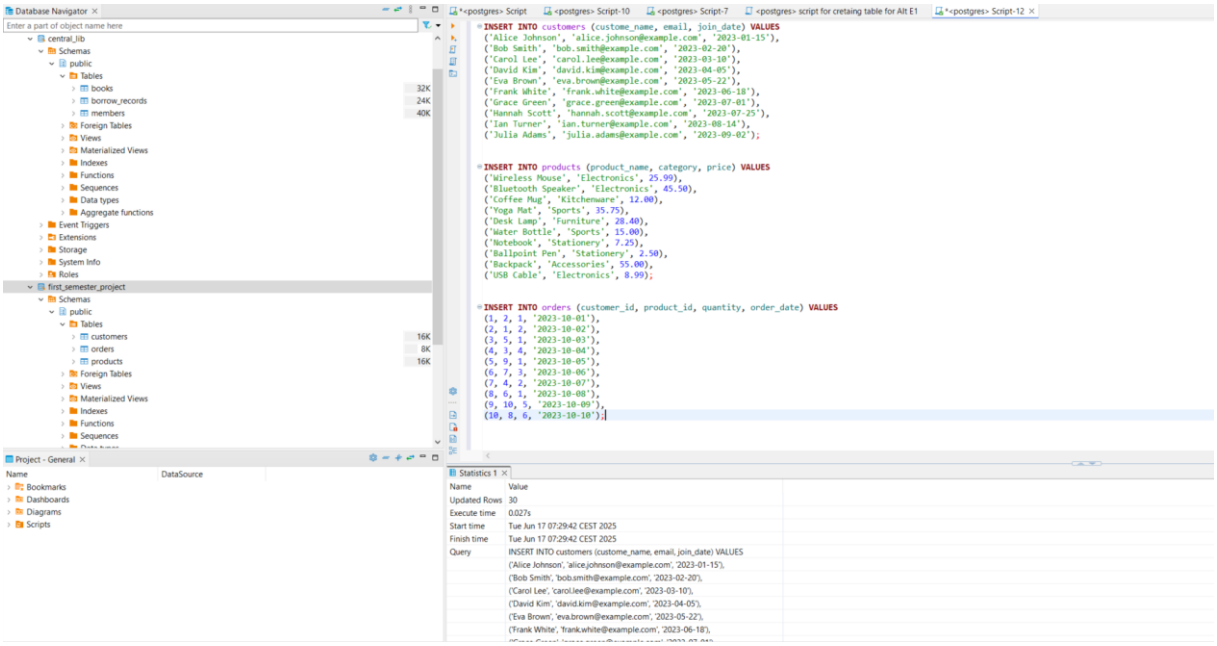
Created database and tables



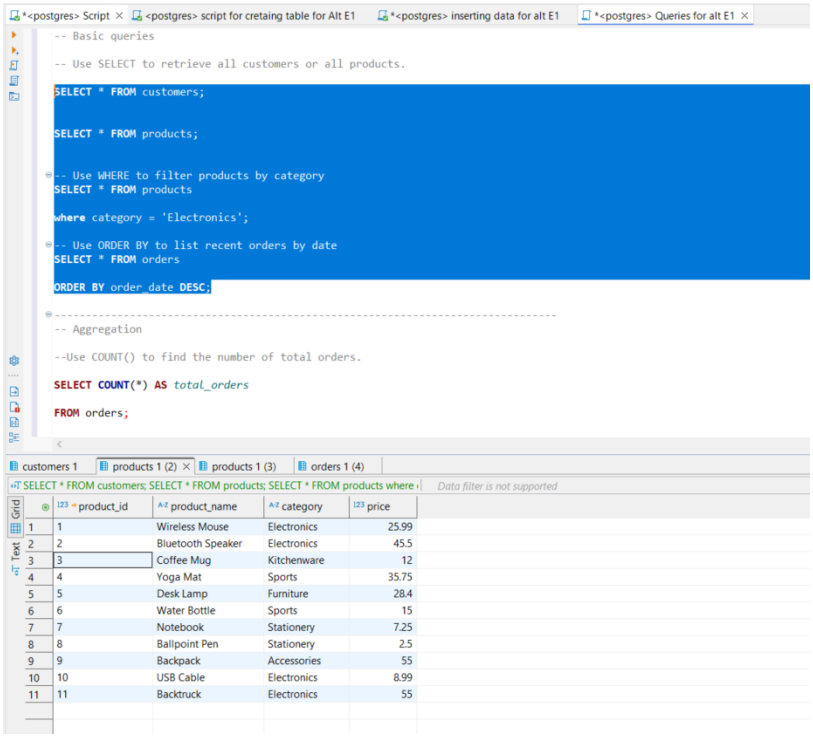
```
customer_id SERIAL PRIMARY KEY,
custome_name VARCHAR(255) NOT NULL,
email VARCHAR(255),
join_date DATE DEFAULT CURRENT_DATE
);

CREATE TABLE Products(
product_id SERIAL PRIMARY KEY,
product_name VARCHAR(255) NOT NULL,
category VARCHAR(255),
price DECIMAL(10, 2)
);

CREATE table Orders (
order_id SERIAL PRIMARY KEY,
customer_id INT,
product_id INT,
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
FOREIGN KEY (product_id) REFERENCES Products(product_id),
quantity int,
```

Inserting data into table

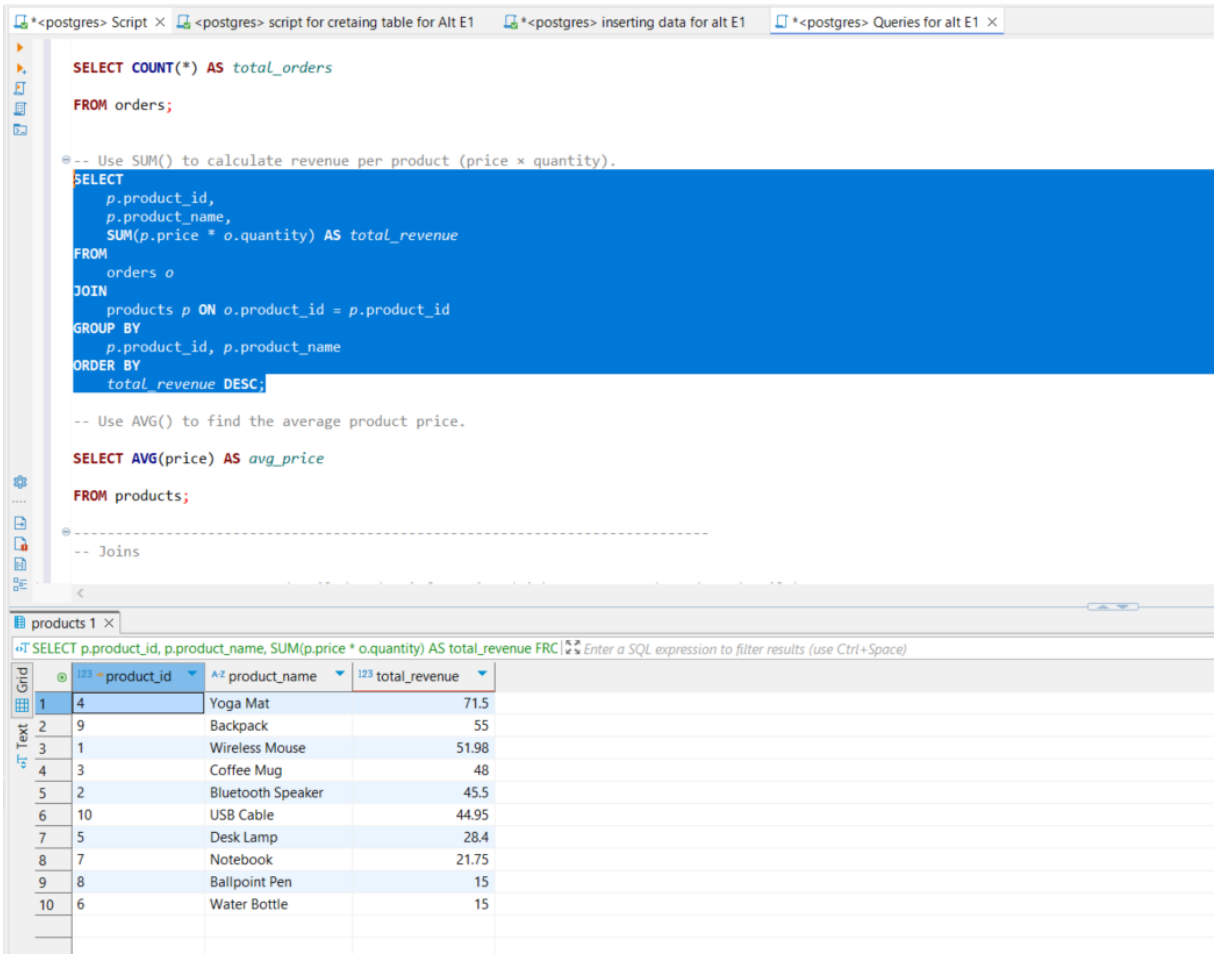


```
INSERT INTO customers (custome_name, email, join_date) VALUES
('Alice Johnson', 'alice.johnson@example.com', '2023-01-15'),
('Bob Smith', 'bob.smith@example.com', '2023-02-20'),
('Carol Lee', 'carol.lee@example.com', '2023-03-10'),
('David Kim', 'david.kim@example.com', '2023-04-05'),
('Eva Brown', 'eva.brown@example.com', '2023-05-22'),
('Frank White', 'frank.white@example.com', '2023-06-18'),
('Grace Green', 'grace.green@example.com', '2023-07-01'),
('Hannah Scott', 'hannah.scott@example.com', '2023-07-25'),
('Ian Turner', 'ian.turner@example.com', '2023-08-14'),
('Julia Adams', 'julia.adams@example.com', '2023-09-02');


INSERT INTO products (product_name, category, price) VALUES
('Wireless Mouse', 'Electronics', 25.99),
('Bluetooth Speaker', 'Electronics', 45.50),
('Coffee Mug', 'Kitchenware', 12.00),
('Yoga Mat', 'Sports', 35.75),
('Desk Lamp', 'Furniture', 28.40),
('Water Bottle', 'Sports', 15.00),
('Notebook', 'Stationery', 7.25),
('Ballpoint Pen', 'Stationery', 2.50),
('Backpack', 'Accessories', 55.00),
('USB Cable', 'Electronics', 8.99);


INSERT INTO orders (customer_id, product_id, quantity, order_date) VALUES
(1, 2, 1, '2023-10-01'),
(2, 1, 2, '2023-10-02'),
(3, 5, 1, '2023-10-03'),
(4, 3, 4, '2023-10-04'),
(5, 9, 1, '2023-10-05'),
(6, 7, 3, '2023-10-06'),
(7, 4, 2, '2023-10-07'),
(8, 6, 1, '2023-10-08'),
(9, 10, 5, '2023-10-09'),
(10, 8, 6, '2023-10-10');
```

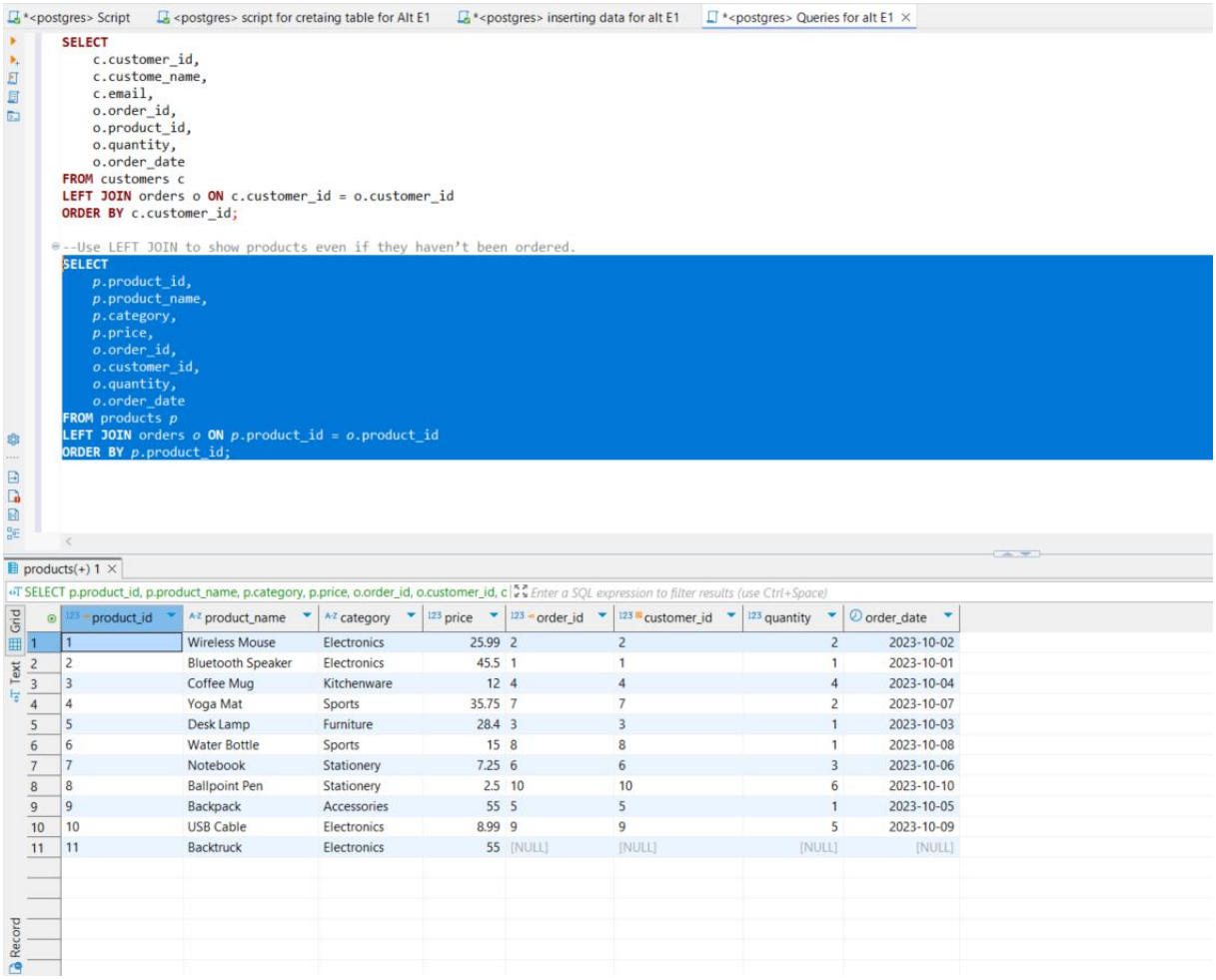
Query 1


*<postgres> Script
<postgres> script for cretaing table for Alt E1
*<postgres> inserting data for alt E1
*<postgres> Queries for alt E1
-- Basic queries
-- Use SELECT to retrieve all customers or all products.
SELECT * FROM customers;
SELECT * FROM products;
-- Use WHERE to filter products by category
SELECT * FROM products
where category = 'Electronics';
-- Use ORDER BY to list recent orders by date
SELECT * FROM orders
ORDER BY order_date DESC;
-- Aggregation
--Use COUNT() to find the number of total orders.
SELECT COUNT(*) AS total_orders
FROM orders;
customers 1
products 1 (2)
products 1 (3)
orders 1 (4)
SELECT * FROM customers; SELECT * FROM products; SELECT * FROM products where
Data filter is not supported
product_id
product_name
category
price
1 1 Wireless Mouse Electronics 25.99
2 2 Bluetooth Speaker Electronics 45.5
3 3 Coffee Mug Kitchenware 12
4 4 Yoga Mat Sports 35.75
5 5 Desk Lamp Furniture 28.4
6 6 Water Bottle Sports 15
7 7 Notebook Stationery 7.25
8 8 Ballpoint Pen Stationery 2.5
9 9 Backpack Accessories 55
10 10 USB Cable Electronics 8.99
11 11 Backtruck Electronics 55

Query 2


*<postgres> Script
<postgres> script for cretaing table for Alt E1
*<postgres> inserting data for alt E1
*<postgres> Queries for alt E1
SELECT COUNT(*) AS total_orders
FROM orders;
-- Use SUM() to calculate revenue per product (price × quantity).
SELECT
p.product_id,
p.product_name,
SUM(p.price * o.quantity) AS total_revenue
FROM
orders o
JOIN
products p ON o.product_id = p.product_id
GROUP BY
p.product_id, p.product_name
ORDER BY
total_revenue DESC;
-- Use AVG() to find the average product price.
SELECT AVG(price) AS avg_price
FROM products;
-- Joins
products 1
SELECT p.product_id, p.product_name, SUM(p.price * o.quantity) AS total_revenue FRC
Enter a SQL expression to filter results (use Ctrl+Space)
product_id
product_name
total_revenue
1 4 Yoga Mat 71.5
2 9 Backpack 55
3 1 Wireless Mouse 51.98
4 3 Coffee Mug 48
5 2 Bluetooth Speaker 45.5
6 10 USB Cable 44.95
7 5 Desk Lamp 28.4
8 7 Notebook 21.75
9 8 Ballpoint Pen 15
10 6 Water Bottle 15

Query 3


*<postgres> Script
<postgres> script for cretaing table for Alt E1
*<postgres> inserting data for alt E1
*<postgres> Queries for alt E1
SELECT
c.customer_id,
c.custome_name,
c.email,
o.order_id,
o.product_id,
o.quantity,
o.order_date
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
ORDER BY c.customer_id;
--Use LEFT JOIN to show products even if they haven't been ordered.
SELECT
p.product_id,
p.product_name,
p.category,
p.price,
o.order_id,
o.customer_id,
o.quantity,
o.order_date
FROM products p
LEFT JOIN orders o ON p.product_id = o.product_id
ORDER BY p.product_id;
products(+) 1
SELECT p.product_id, p.product_name, p.category, p.price, o.order_id, o.customer_id, c
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
product_id
product_name
category
price
order_id
customer_id
quantity
order_date
1 1 Wireless Mouse Electronics 25.99 2 2 2 2023-10-02
2 2 Bluetooth Speaker Electronics 45.5 1 1 1 2023-10-01
3 3 Coffee Mug Kitchenware 12 4 4 4 2023-10-04
4 4 Yoga Mat Sports 35.75 7 7 2 2023-10-07
5 5 Desk Lamp Furniture 28.4 3 3 1 2023-10-03
6 6 Water Bottle Sports 15 8 8 1 2023-10-08
7 7 Notebook Stationery 7.25 6 6 3 2023-10-06
8 8 Ballpoint Pen Stationery 2.5 10 10 6 2023-10-10
9 9 Backpack Accessories 55 5 5 1 2023-10-05
10 10 USB Cable Electronics 8.99 9 9 5 2023-10-09
11 11 Backtruck Electronics 55 [NULL] [NULL] [NULL] [NULL]