Advanced SQL Exercise – Ranking and Window Functions

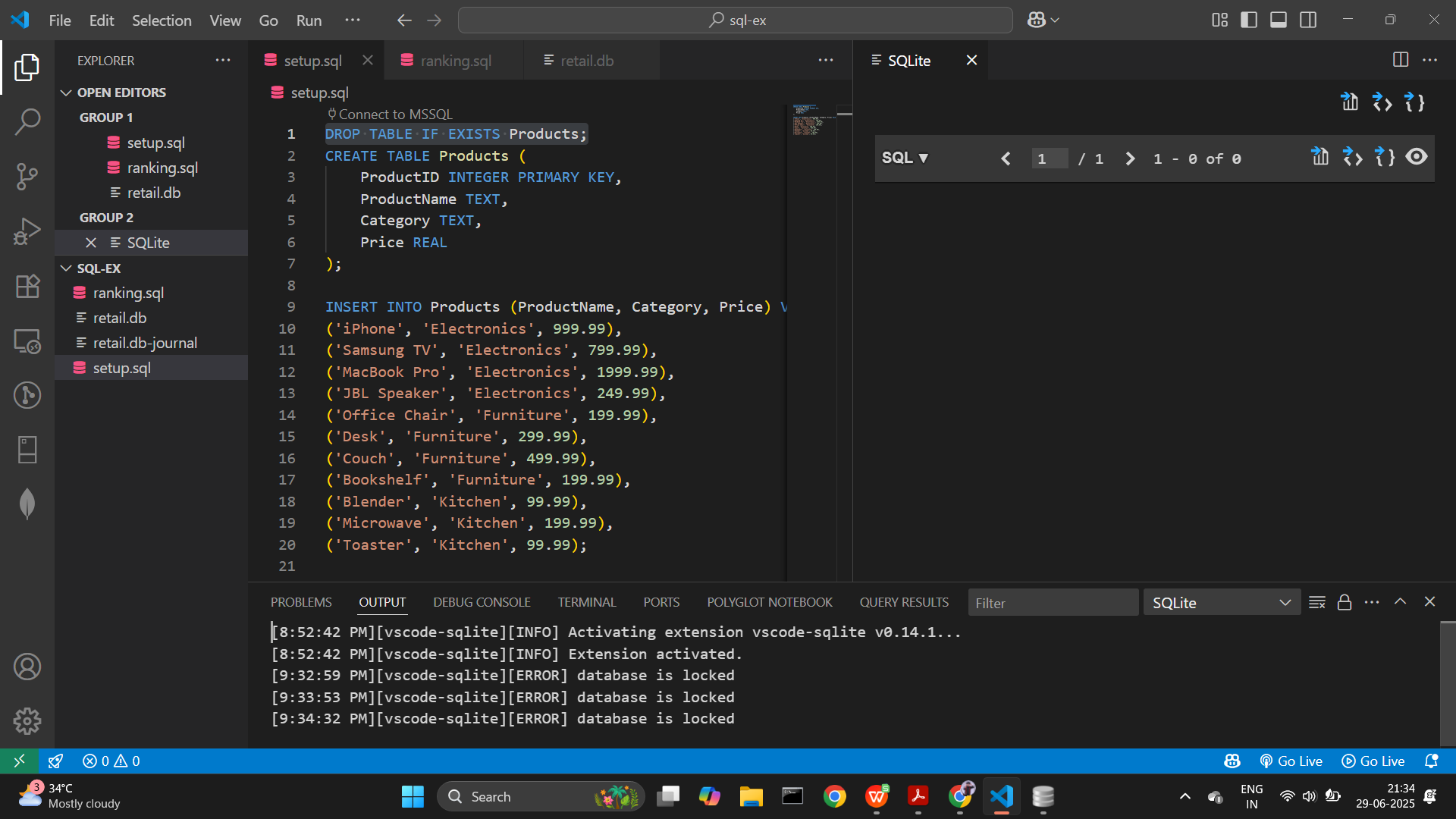
# What I Learned

This exercise focused on mastering SQL Window Functions using SQLite inside Visual Studio Code.   
I learned to use ROW\_NUMBER(), RANK(), and DENSE\_RANK() with PARTITION BY and ORDER BY clauses to rank products within categories.  
I also learned how to configure VS Code to work with SQLite effectively, including resolving locking issues and extension setup.

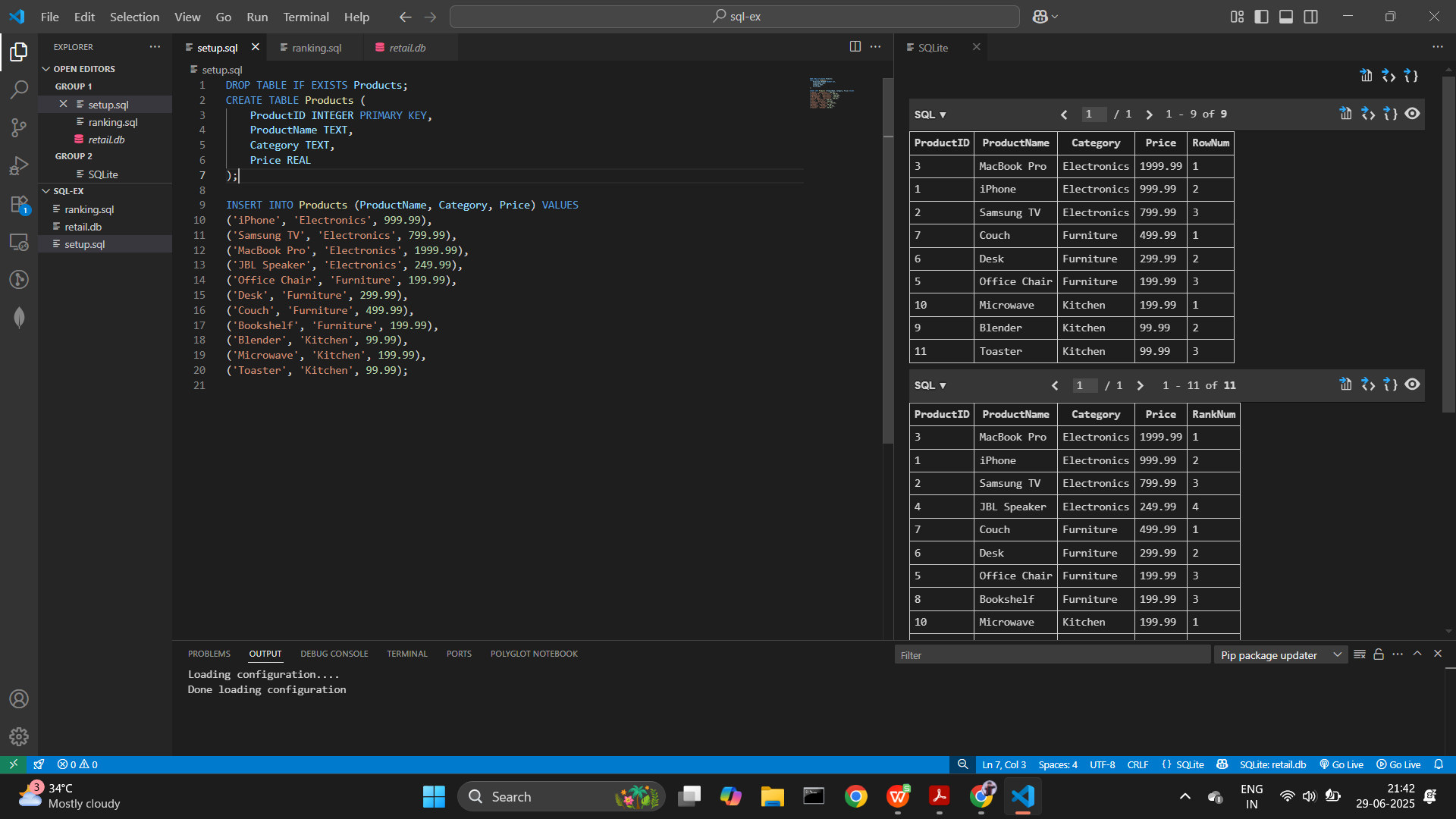
# Tools Used

- Visual Studio Code (VS Code)  
- SQLite Extension (Alex Covizzi)  
- retail.db SQLite database  
- setup.sql and ranking.sql scripts

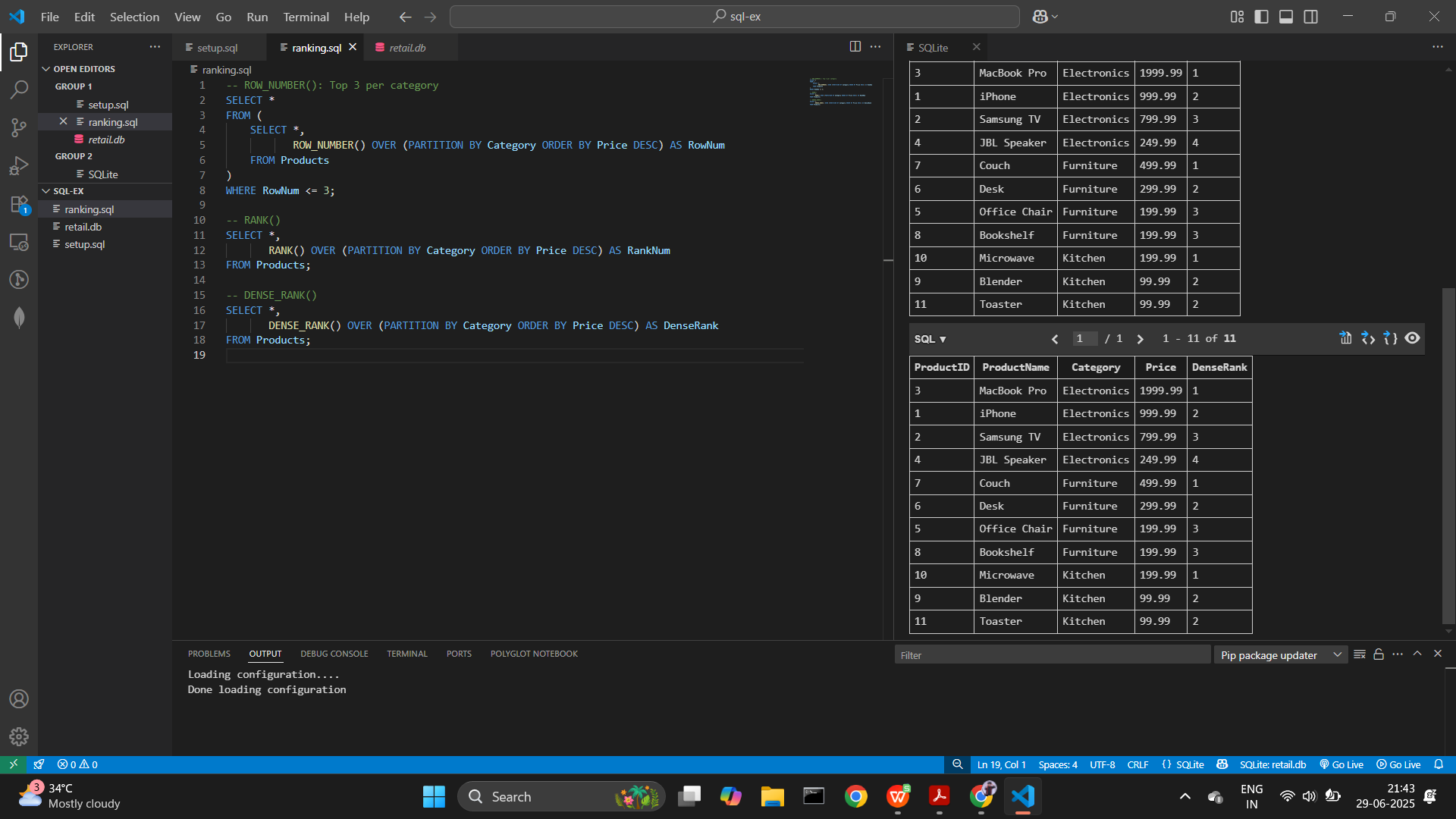
## Screenshot: setup.sql – Table Creation & Data Insertion



## Screenshot: ranking.sql – Query with ROW\_NUMBER, RANK, DENSE\_RANK



## Screenshot: Query Outputs in SQLite Explorer



# SQL Queries Used

-- Create table and insert products  
DROP TABLE IF EXISTS Products;  
CREATE TABLE Products (  
 ProductID INTEGER PRIMARY KEY,  
 ProductName TEXT,  
 Category TEXT,  
 Price REAL  
);  
  
INSERT INTO Products (ProductName, Category, Price) VALUES  
('iPhone', 'Electronics', 999.99),  
('Samsung TV', 'Electronics', 799.99),  
('MacBook Pro', 'Electronics', 1999.99),  
('JBL Speaker', 'Electronics', 249.99),  
('Office Chair', 'Furniture', 199.99),  
('Desk', 'Furniture', 299.99),  
('Couch', 'Furniture', 499.99),  
('Bookshelf', 'Furniture', 199.99),  
('Blender', 'Kitchen', 99.99),  
('Microwave', 'Kitchen', 199.99),  
('Toaster', 'Kitchen', 99.99);

-- ROW\_NUMBER(): Top 3 per category  
SELECT \*  
FROM (  
 SELECT \*,  
 ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Price DESC) AS RowNum  
 FROM Products  
)  
WHERE RowNum <= 3;

-- RANK()  
SELECT \*,  
 RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS RankNum  
FROM Products;

-- DENSE\_RANK()  
SELECT \*,  
 DENSE\_RANK() OVER (PARTITION BY Category ORDER BY Price DESC) AS DenseRank  
FROM Products;