

|  |
| --- |
|  |

Retail Database Project Report

**Student: Seif Ahmed Hosny**

**Student ID: 1110148878**

Table of Contents

### Introduction

### Database Design

### Tables and Relationships

### Design Choices

### Sample Data

### Complex Queries

### Inner Join

### Left Join

### Right Join

### Full Outer Join

### Subquery

### Common Table Expressions (CTEs)

### Window Functions

### Views and Stored Procedures

### Views

### Stored Procedures

### Conclusion

1. Introduction

**Purpose: This report provides a comprehensive analysis of the retail database design, usage, and query capabilities.**

**Scope: It includes details on database design, sample data, complex queries, and advanced features such as views and stored procedures.**

2. Database Design

# 2.1.Tables and Relationships

### 1.Product Table

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **ProductID** | **INT** | **Unique identifier for each product** |
| **ProductName** | **VARCHAR** | **Name of the product** |
| **Category** | **VARCHAR** | **Product category** |
| **Price** | **DECIMAL** | **Price of the product** |
| **StockQuantity** | **INT** | **Quantity in stock** |

### 2.Customer Table

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **CustomerID** | **INT** | **Unique identifier for each customer** |
| **FirstName** | **VARCHAR** | **Customer's first name** |
| **LastName** | **VARCHAR** | **Customer's last name** |
| **Email** | **VARCHAR** | **Customer's email(unique)** |
| **Phone** | **VARCHAR** | **Customer's phone** |
| **Address** | **VARCHAR** | **Customer's address** |

### 3.Order Table

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **OrderID** | **INT** | **Unique identifier for each order** |
| **CustomerID** | **INT** | **Links to the Customers table** |
| **OrderDate** | **DATE** | **Date when the order was placed** |
| **TotalAmount** | **DECEMAL** | **Total amount of the order** |
| **OrderStatus** | **VARCHAR** | **Status of the order** |

### 4.Order Details Table

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| OrderDetailID | INT | Unique identifier for each order details |
| OrderID | INT | Links to the Orders table |
| ProductID | INT | Links to the Product table |
| Quantity | INT | Quantity of the product ordered |
| UnitPrice | DECEMAL | Price of the product at the time of order |

### 

### 5.Suppliers Table

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| SupplierID | INT | Unique identifier for each suppliers |
| SupplierName | VARCHAR | Name of the supplier |
| ContactName | VARCHAR | Contact person at the supplier |
| Phone | VARCHAR | Supplier's phone number |
| Email | VARCHAR | Supplier's email (unique) |

2.2Design Choices

### **Normalization:** The design is normalized to minimize redundancy and ensure data integrity.

### **Referential Integrity:** Foreign key constraints are used to maintain relationships between tables.

### **Constraints:** Constraints such as CHECK and UNIQUE enforce data validity.

3. Sample Data

### Products: 1000 products with various attributes.

### Customers: 1000 customers with diverse details.

### Orders: 1000 orders with different statuses and amounts.

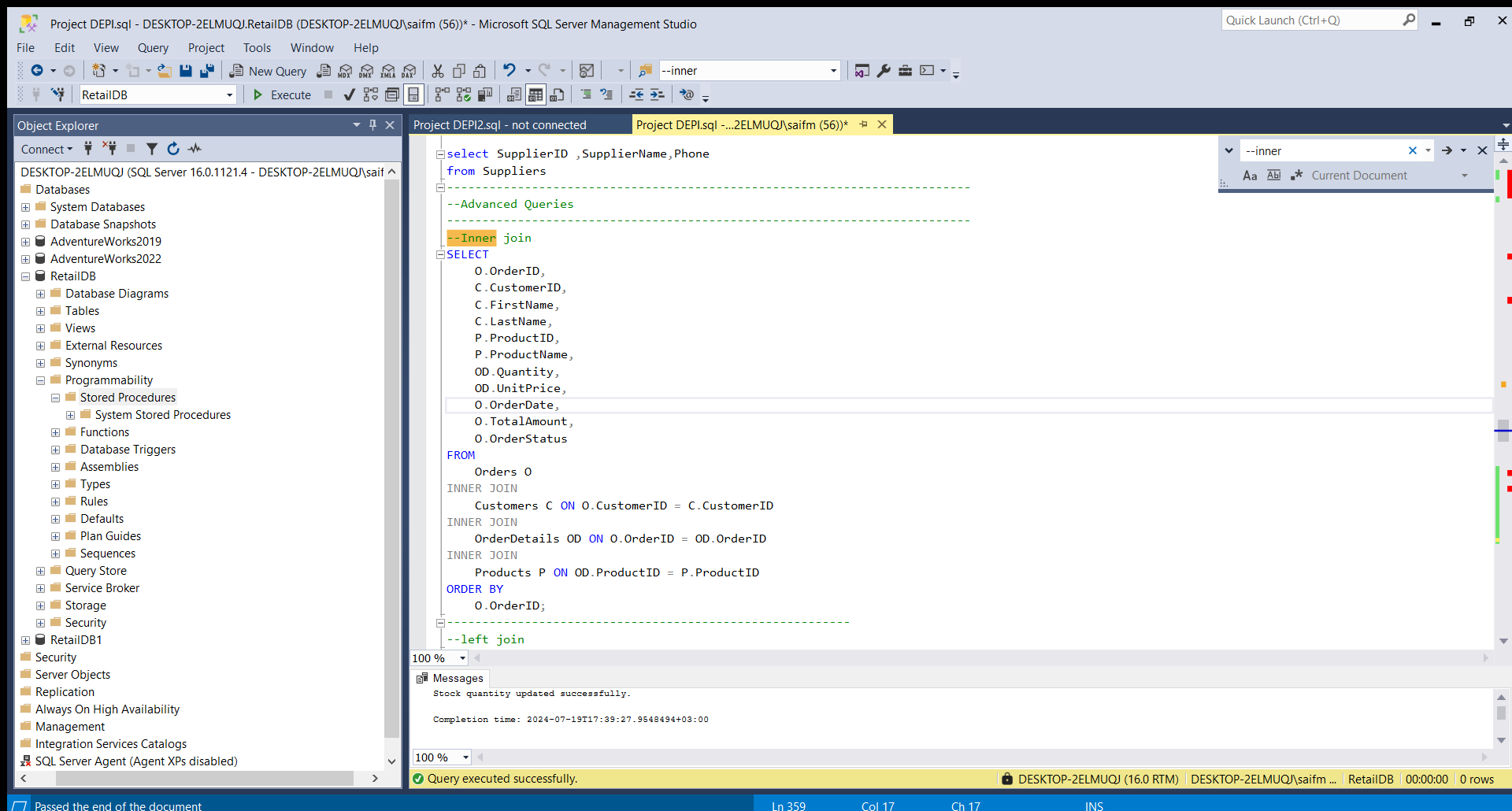
### OrderDetails: 1000 details linking products with orders.

### Suppliers: 10000 suppliers with contact information.

### Purpose: The sample data facilitates testing and validation of database operations.

### 4. Complex Queries

#### **4.1Inner Join**



***Explanation:* Retrieves detailed order information including customer and product details.**

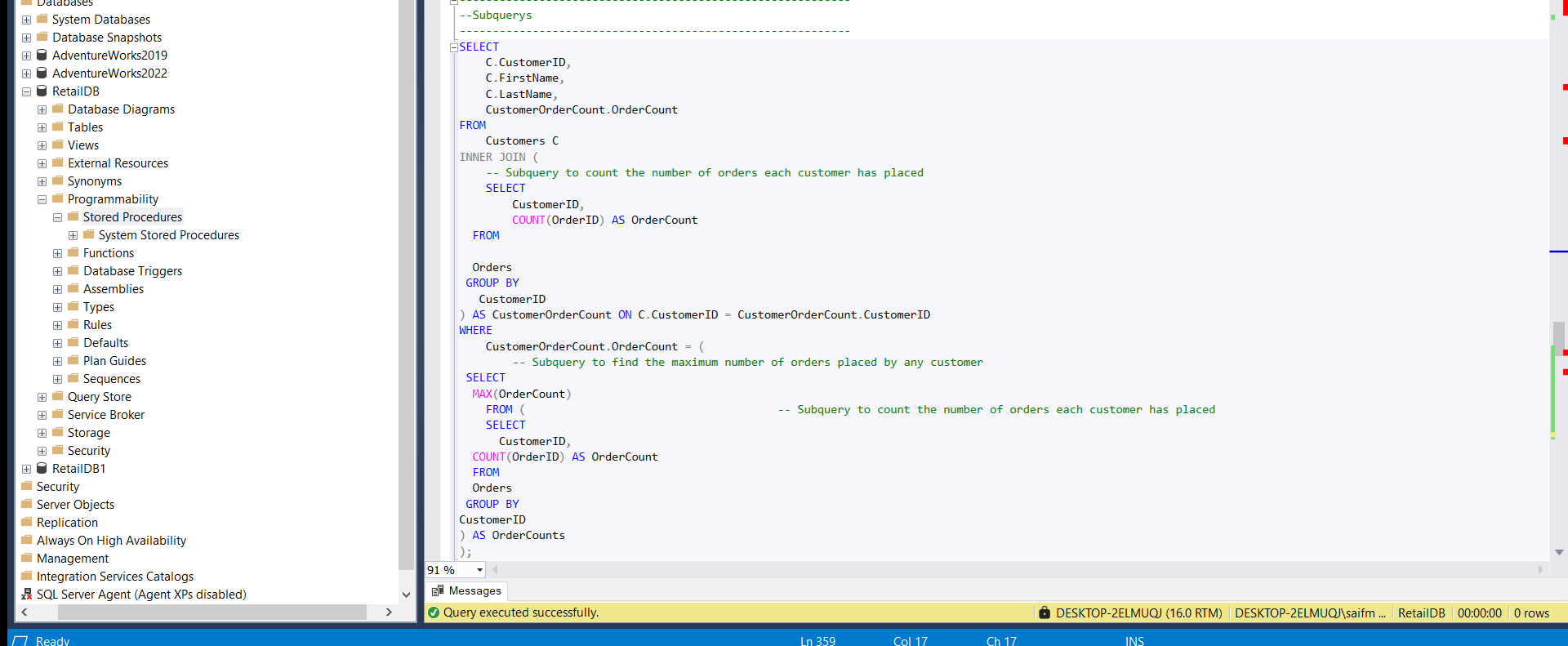
#### **4.2 Left Join,Right Join,Full Outer Join**

***Left Join Explanation:* Retrieves all products and their associated suppliers if any.**

***Right Join Explanation:* Retrieves all suppliers and their associated products if any.**

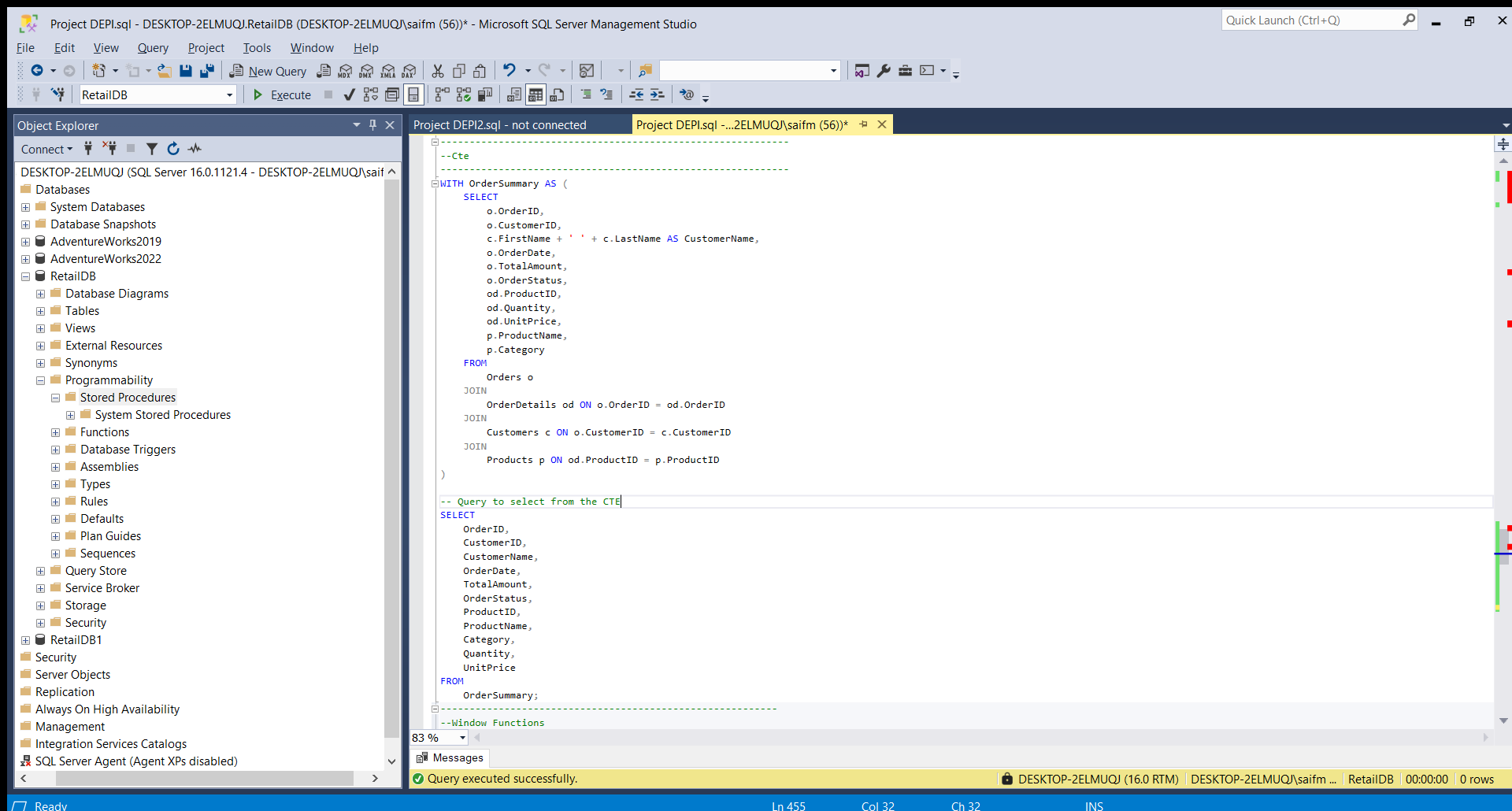
***Full Outer Join Explanation:* Retrieves all customers and all orders, including those that don't have matching records in the other table.**

#### **4.3 Subquery**



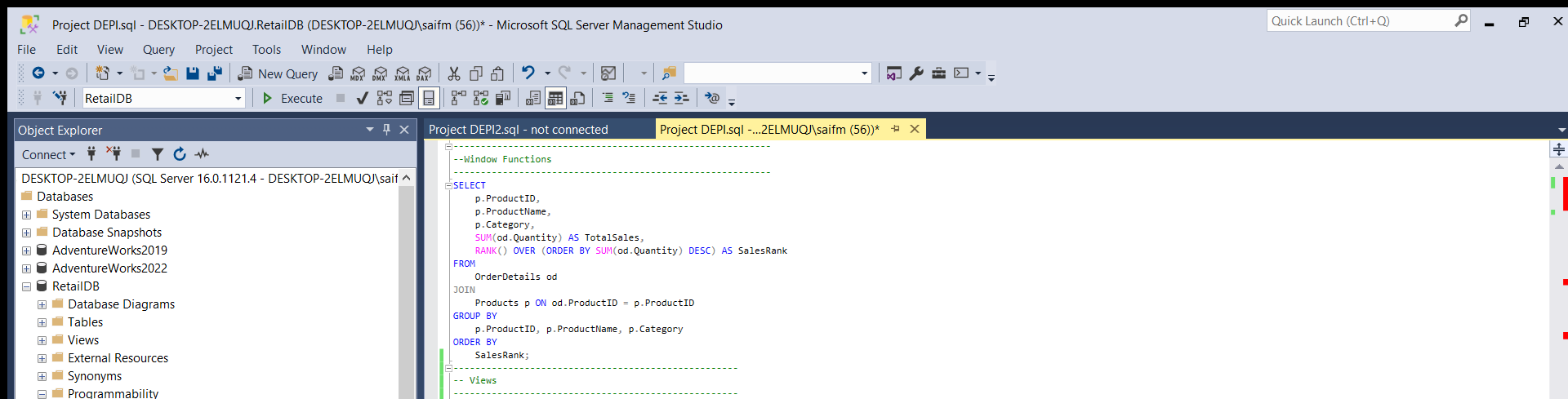
***Explanation:* Finds customers with the highest number of orders.**

#### **4.4 Common Table Expressions (CTEs)**



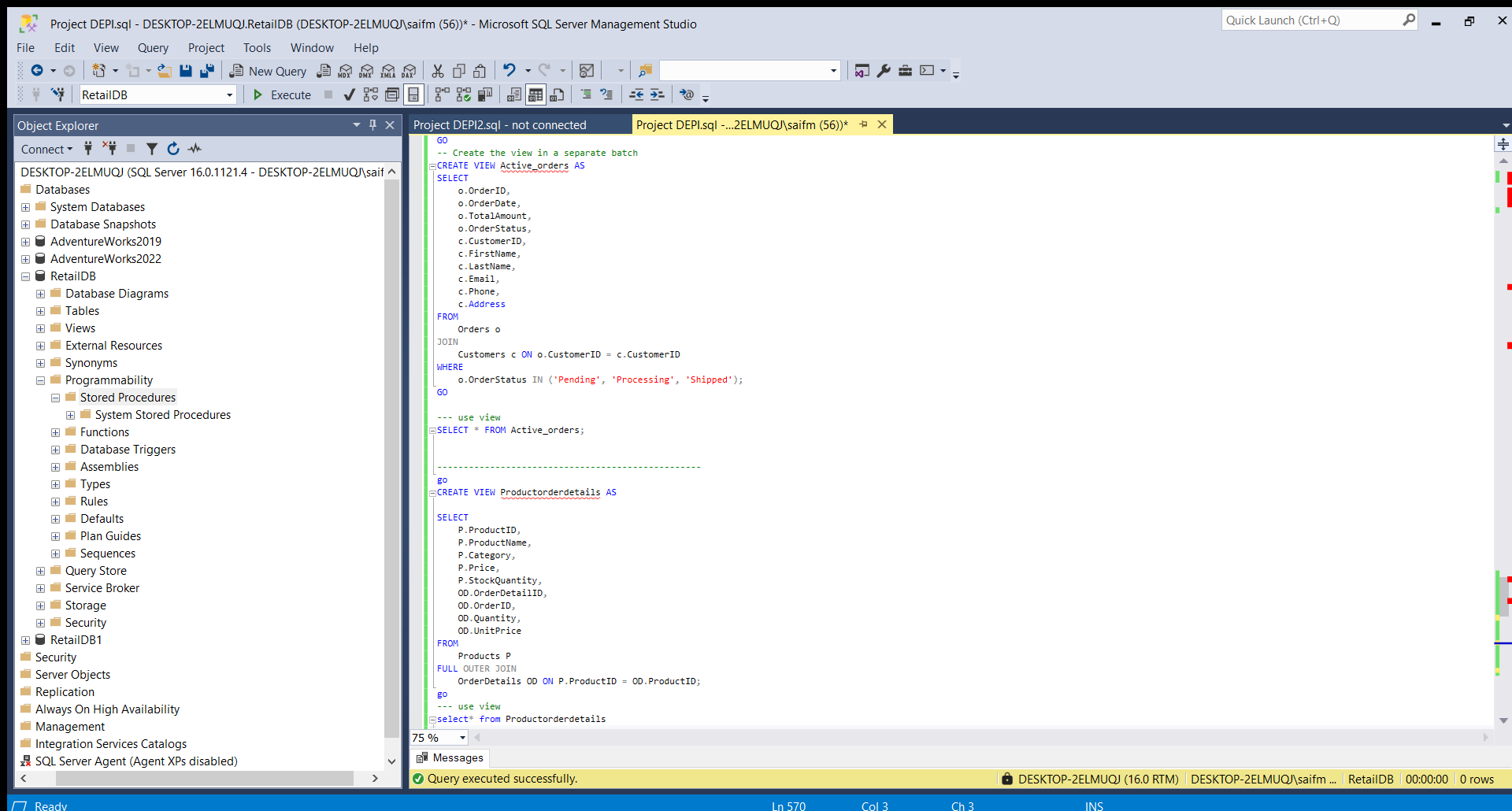
**Explanation: Provides a summary of orders, including customer and product details.**

#### **4.5 Window Functions**



***Explanation:* Ranks products based on total sales.**

### 4.6 Views



**Active\_Orders**

**Explanation: The view is created to provide a consolidated view of orders that are currently active. Active orders include those that are in 'Pending', 'Processing', or 'Shipped' statuses.**

##### Product Order Details

**Explanation: The view is designed to provide comprehensive details about products and their corresponding order details. This includes all products and order details, even if there are no matching entries between the Products and OrderDetails tables.**

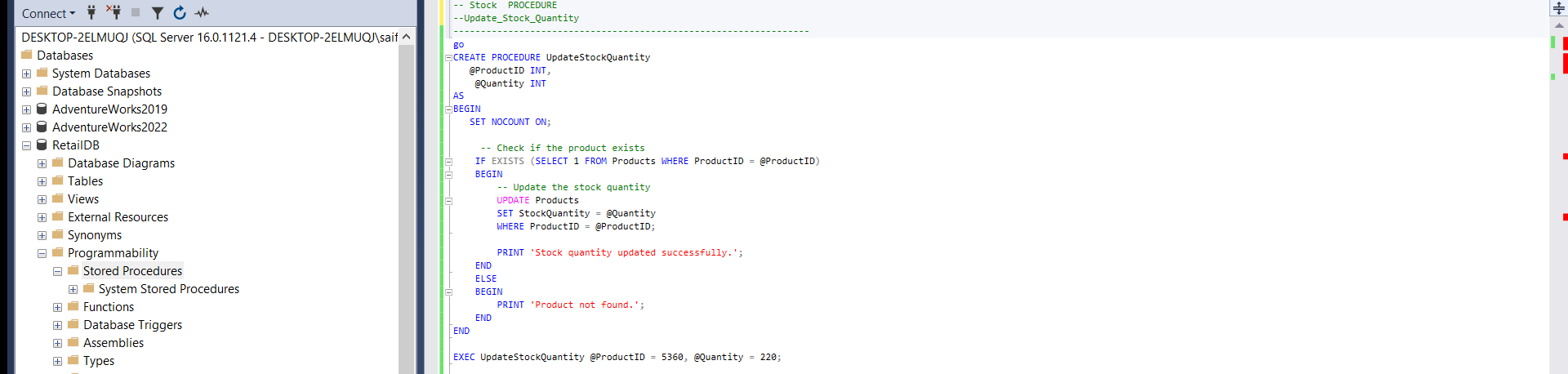
#### **4.7 Stored Procedures**

**Update\_Stock\_Quantity:  
  
This stored procedure is designed to update the stock quantity of a specific product in the database.**

**Parameters:**

**@ProductID: The ID of the product whose stock quantity needs to be updated.**

**@Quantity: The new stock quantity for the product.**

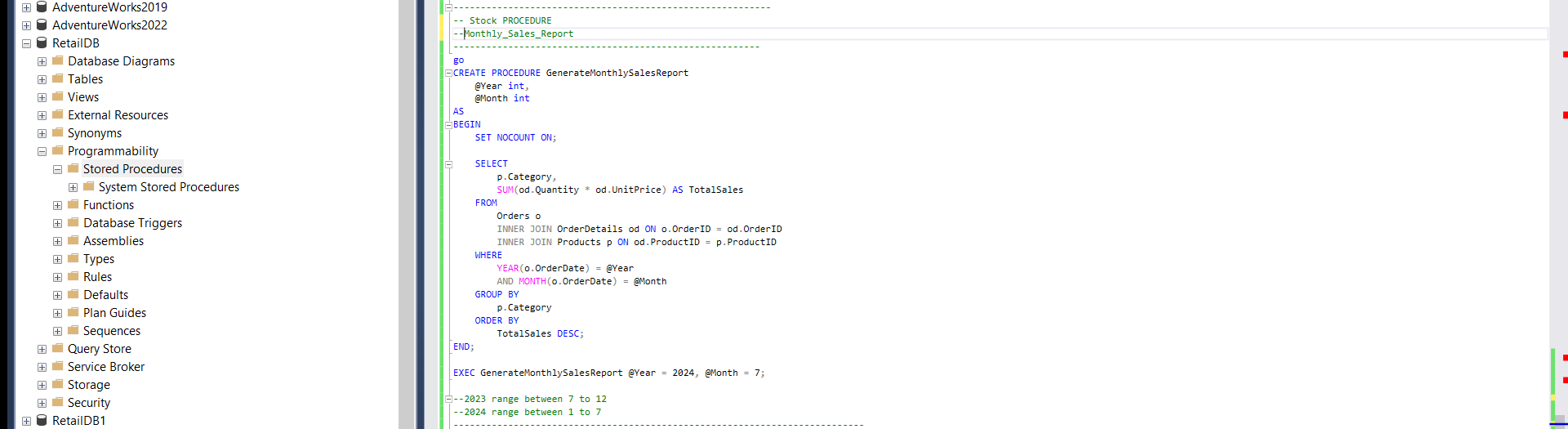


**Monthly\_Sales\_Report  
The GenerateMonthlySalesReport stored procedure is designed to generate a summary of sales for a given month and year.**

**Parameters:**

**@Year int: The year for which the sales report is generated.**

**@Month int: The month for which the sales report is generated.**



8. Conclusion

### **Summary:** This report provides an in-depth overview of the database design, sample data, complex queries, and advanced features. The design is robust, and the queries and procedures are crafted to handle various data retrieval and manipulation needs effectively.