**LAB 1: Understanding ORM with a Retail Inventory System**

1. **What is ORM?**  
   Solution:  
     
   ORM (Object-Relational Mapping) is a technique that allows developers to interact with a relational database using object-oriented programming concepts. Instead of writing raw SQL queries, developers can work with C# classes and objects, and the ORM automatically handles translating these operations to SQL.

ORM Maps C# Classes to Database Tables:  
  
Each C# class corresponds to a table in the database.

Each property of the class corresponds to a column in that table.

Each instance of the class represents a row in the table.

Benefits of ORM:

Productivity: Reduces the need to write boilerplate SQL queries.

Maintainability: Changes in the database schema can be easily managed through C# code.

Abstraction: Developers can work with C# objects and rely on the ORM to handle SQL under the hood.

1. **EF Core vs EF Framework:**Solution:

| **Feature** | **EF Core** | **EF Framework (EF6)** |
| --- | --- | --- |
| Platform | Cross-platform (.NET Core/.NET) | Windows-only (.NET Framework) |
| Lightweight | Yes | No |
| Modern Features | LINQ, async queries, compiled queries | Fewer modern features |
| Performance | Better due to optimizations | Slower |
| Flexibility | More flexible and extensible | Less flexible |
| Maturity | Newer, still evolving | More mature, stable |

1. **EF Core 8.0 Features:**

Solution:  
EF Core 8.0 introduces several powerful features:

JSON Column Mapping: Store and query JSON data directly within relational columns.

Compiled Models: Improve performance by precompiling the model at build time.

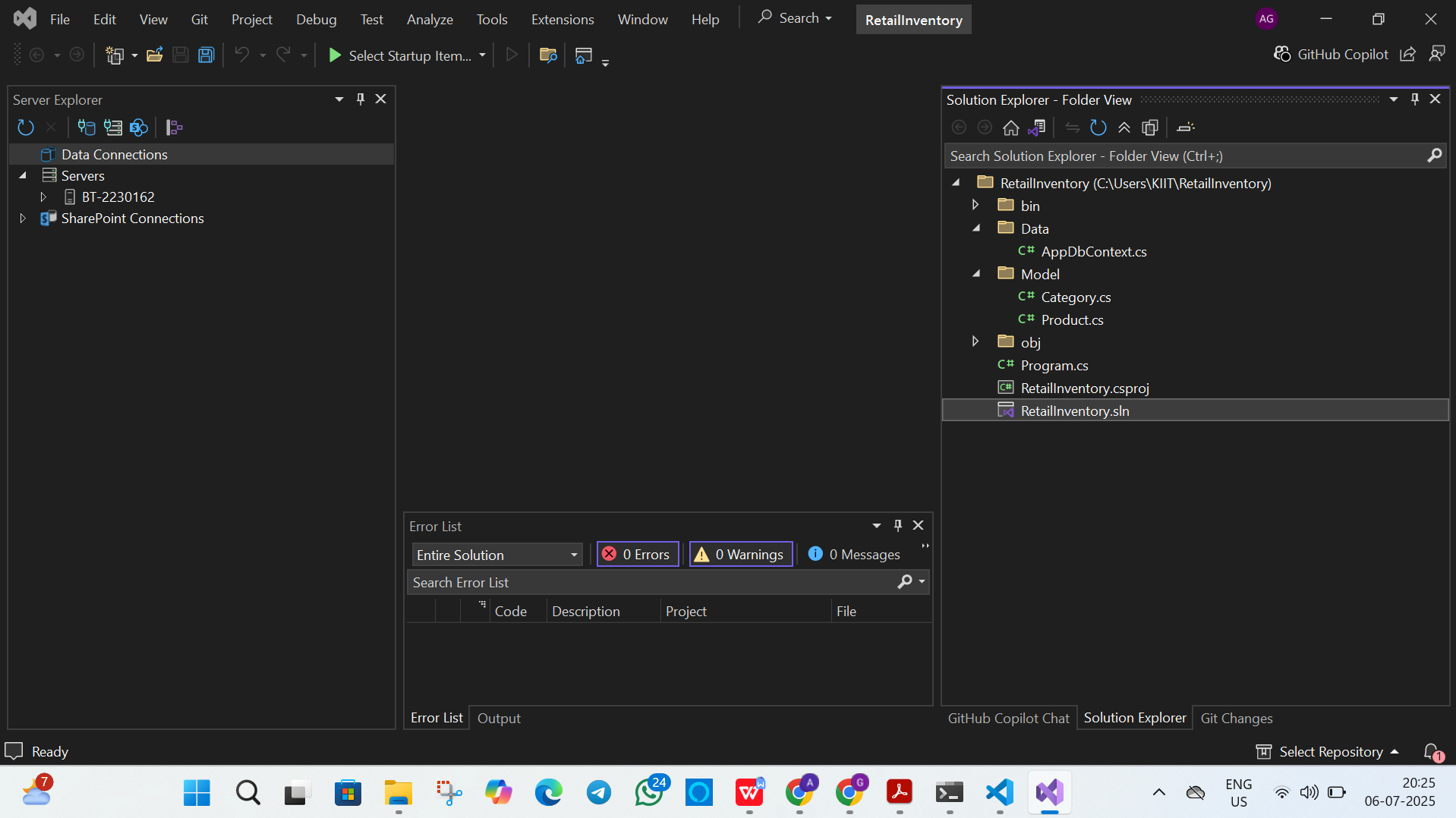
Interceptors: Customize and intercept EF Core operations like queries or commands.

Improved Bulk Operations: Better handling of batch inserts, updates, and deletes.

These features make EF Core 8.0 highly efficient and flexible for enterprise-level applications.

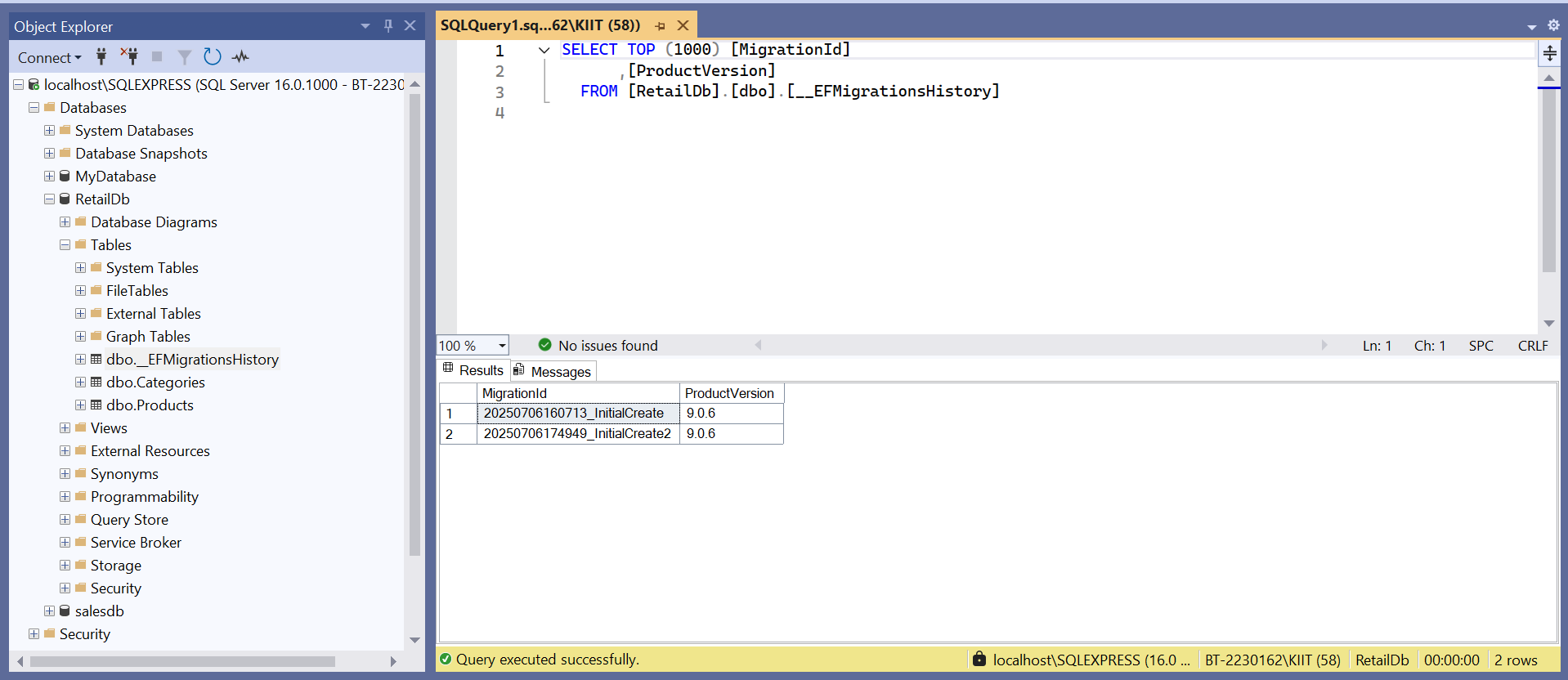
**LAB 2: Setting Up the Database Context for a Retail Store**

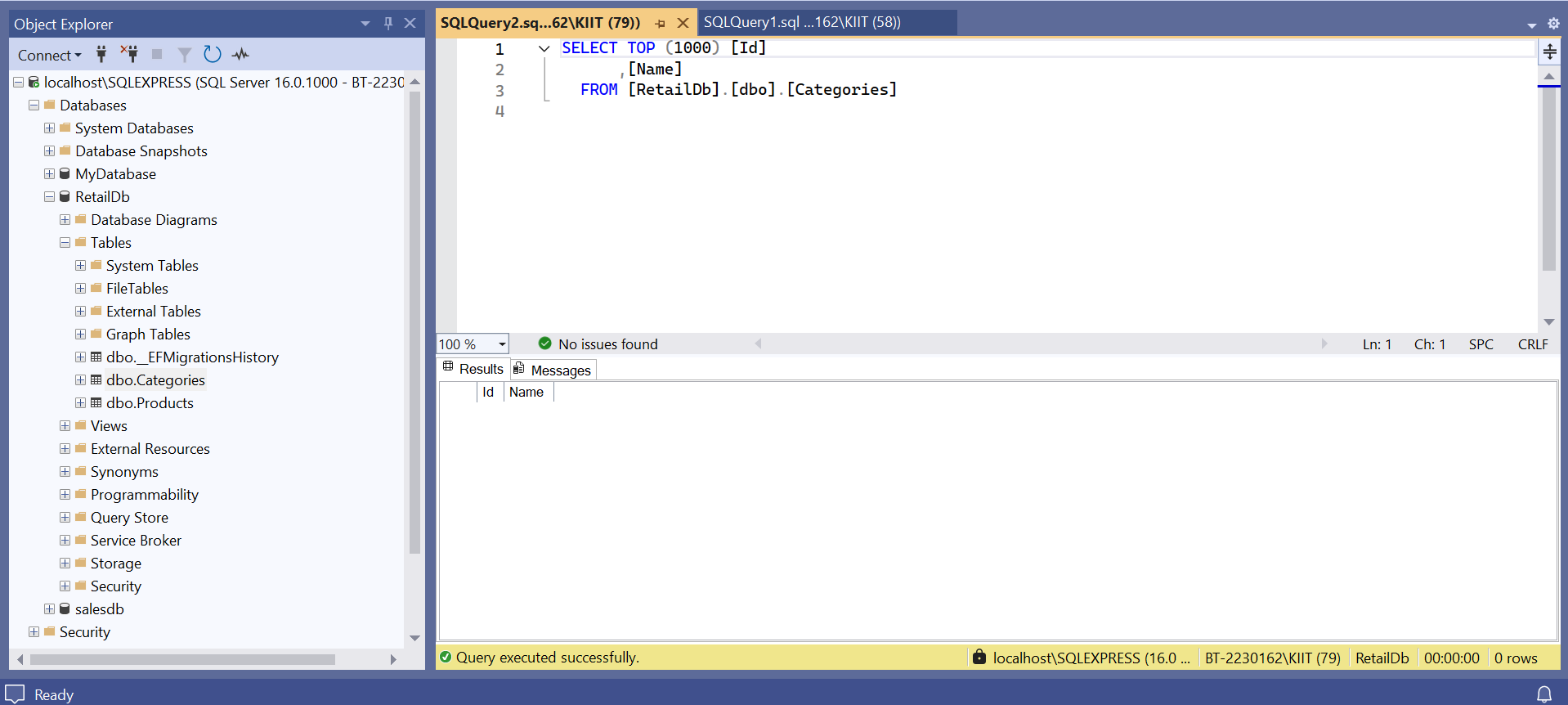
Scenario: The retail store wants to store product and category data in SQL Server.

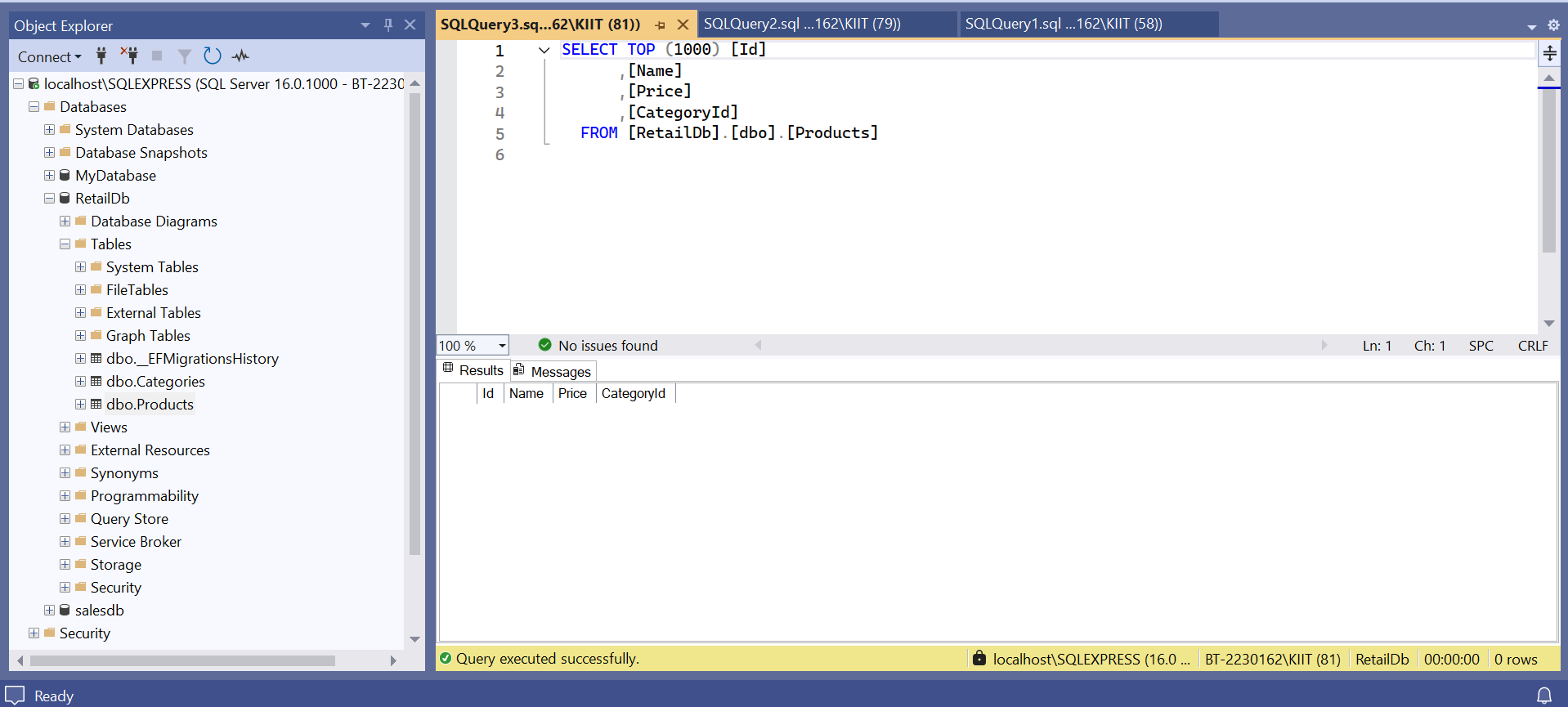
Objective: Configure DbContext and connect to SQL Server.  
  
Solution:  
This page was made in Visual Studio 2022  


**LAB 3: Using EF Core CLI to Create and Apply Migrations**  
  
Scenario: The retail store's database needs to be created based on the models you've defined. You’ll use EF Core CLI to generate and apply migrations.

Objective: Learn how to use EF Core CLI to manage database schema changes.

Solution:  
Database with tables created:  


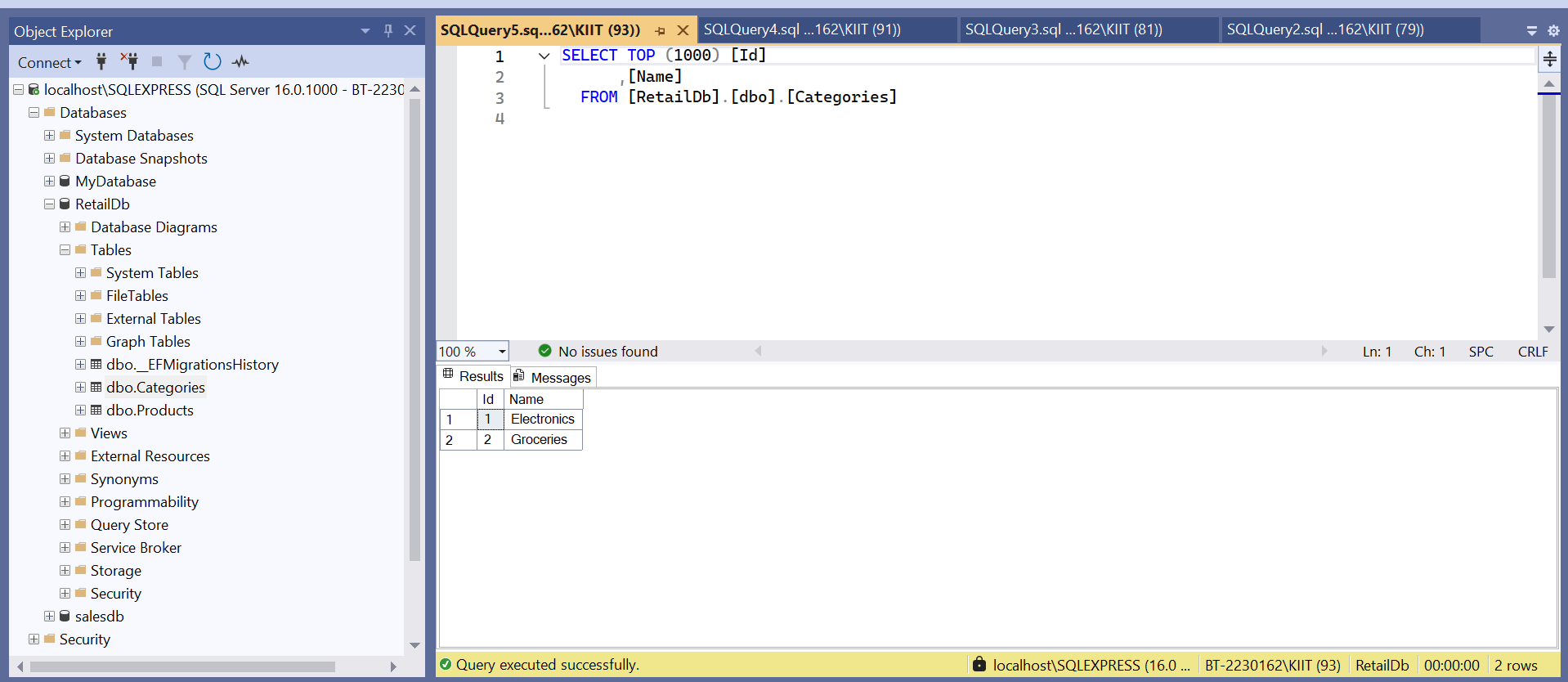


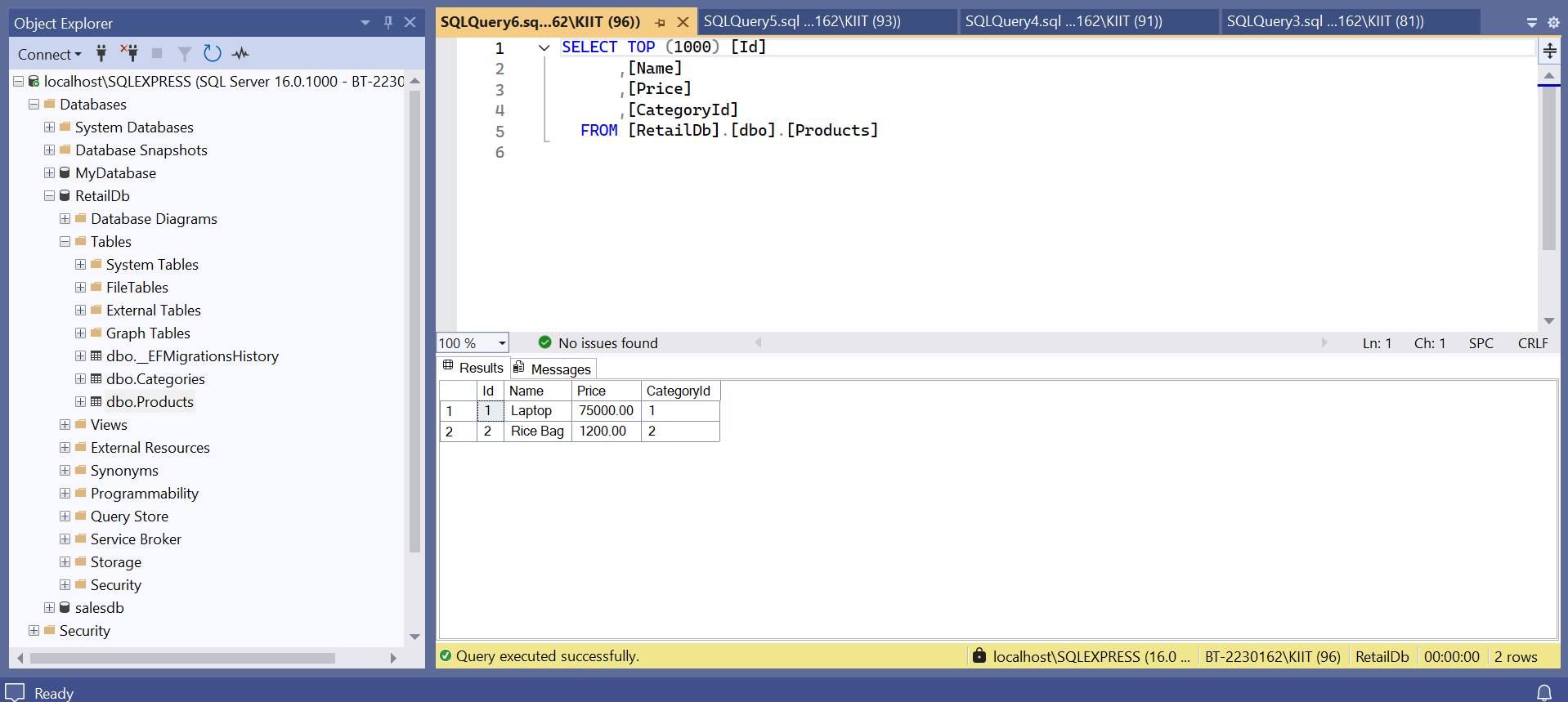


**LAB 4: Inserting Initial Data into the Database**  
Scenario: The store manager wants to add initial product categories and products to the system.

Objective: Use EF Core to insert records using AddAsync and SaveChangesAsync.

Solution:  
Updated databses





**LAB 5: Retrieving Data from the Database**

Scenario: The store wants to display product details on the dashboard.

Objective: Use Find, FirstOrDefault, and ToListAsync to retrieve data.

Solution:  
