SQL Project Documentation

Overview

This documentation provides an in-depth look into the products and OrderDetails tables within the database schema. Each table's structure, column definitions, constraints, and relationships are thoroughly detailed to ensure clarity and professionalism.

Table: products

The products table maintains a catalog of all products available in the inventory. It includes essential information such as product identification, name, category, pricing, and stock quantity. This table is fundamental for managing product details and ensuring data integrity.

Table Structure

- *ProductID*: INT (Primary Key)
  - Description: Unique identifier for each product.
  - Constraints:
    - PRIMARY KEY: Ensures each ProductID is unique and non-null.

- *ProductName*: VARCHAR(80)
  - Description: Name of the product.
  - Constraints:
    - NOT NULL: Ensures every product has a name.
    - UNIQUE: Ensures no two products have the same name.

- *Category*: VARCHAR(50)
  - Description: Category to which the product belongs.
  - Constraints:
    - DEFAULT 'General': Assigns 'General' as the default category if none is provided.

- *Price*: DECIMAL(10,2)
  - Description: Price of the product.
  - Constraints:
    - NOT NULL: Ensures every product has a price.
    - CHECK (Price > 0): Ensures the price is a positive value.
    - DEFAULT 0.00: Assigns a default price of 0.00 if none is provided.

- *StockQuantity*: INT
  - Description: Quantity of the product available in stock.
  - Constraints:
    - CHECK (StockQuantity > 0): Ensures the stock quantity is a positive value.
    - DEFAULT 0: Assigns a default stock quantity of 0 if none is provided.

Constraints Summary

- *Primary Key*: ProductID
  - Uniquely identifies each product in the table.

- *Unique Constraint*: ProductName
  - Ensures that each product name is unique across the table.

- *Default Values*:
  - Category defaults to 'General'.
  - Price defaults to 0.00.
  - StockQuantity defaults to 0.

- *Check Constraints*:
  - Price must be greater than 0.
  - StockQuantity must be greater than 0.

Table: OrderDetails

The OrderDetails table stores detailed information about each order, including product details, quantity, and pricing. This table is crucial for tracking order specifics and maintaining the relationship between orders and products.

Table Structure

- *OrderDetailID*: INT (Primary Key, Identity)
  - Description: Unique identifier for each order detail.
  - Constraints:
    - PRIMARY KEY: Ensures each OrderDetailID is unique and non-null.
    - IDENTITY(1,1): Automatically generates unique values for each new record.
- *OrderID*: INT
  - Description: Identifier for the order.
  - Constraints:
    - NOT NULL: Ensures each order detail is associated with an order.

- *ProductID*: INT
  - Description: Identifier for the product.
  - Constraints:
    - NOT NULL: Ensures each order detail is associated with a product.
    - FOREIGN KEY: Links to products(ProductID) to maintain referential integrity.

- *Quantity*: INT
  - Description: Quantity of the product ordered.
  - Constraints:
    - NOT NULL: Ensures the quantity is provided for each order detail.
    - CHECK (Quantity > 0): Ensures the quantity is a positive value.

- *UnitPrice*: DECIMAL(10,2)
  - Description: Price per unit of the product at the time of order.
  - Constraints:
    - NOT NULL: Ensures the unit price is provided for each order detail.
    - CHECK (UnitPrice > 0): Ensures the unit price is a positive value.
    - DEFAULT 0.00: Assigns a default unit price of 0.00 if none is provided.

Constraints Summary

- *Primary Key*: OrderDetailID
  - Uniquely identifies each order detail record.

- *Foreign Key*: ProductID
  - Ensures the product exists in the products table, maintaining referential integrity.

- *Default Values*:
  - UnitPrice defaults to 0.00.

- *Check Constraints*:
  - Quantity must be greater than 0.
  - UnitPrice must be greater than 0.

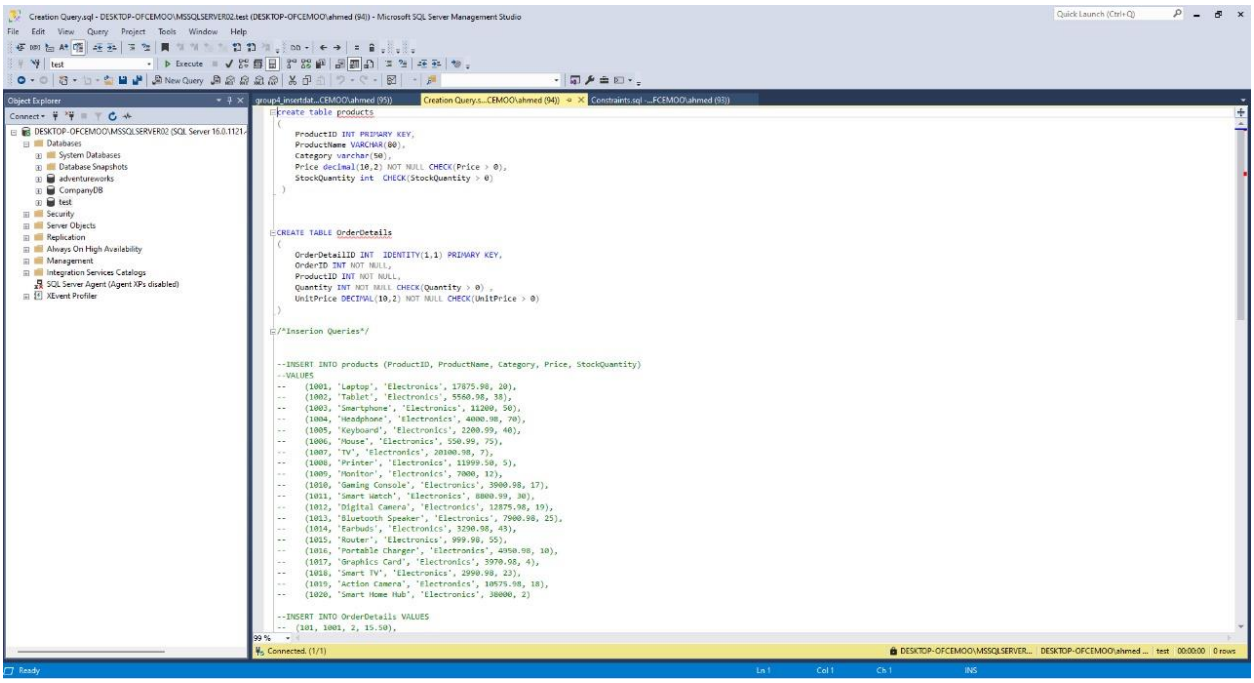
Relationships

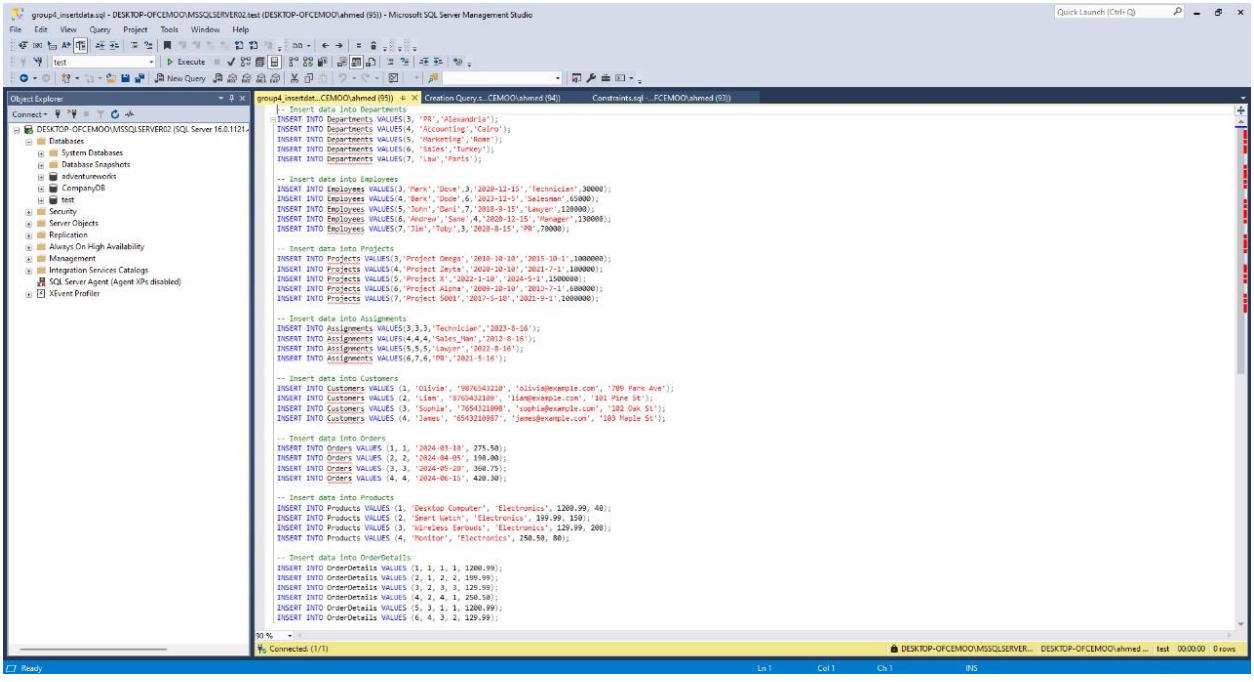
- *OrderDetails to products*: Each record in the OrderDetails table is linked to a record in the products table via the ProductID foreign key. This relationship ensures that order details are associated with valid products, maintaining data consistency and integrity.

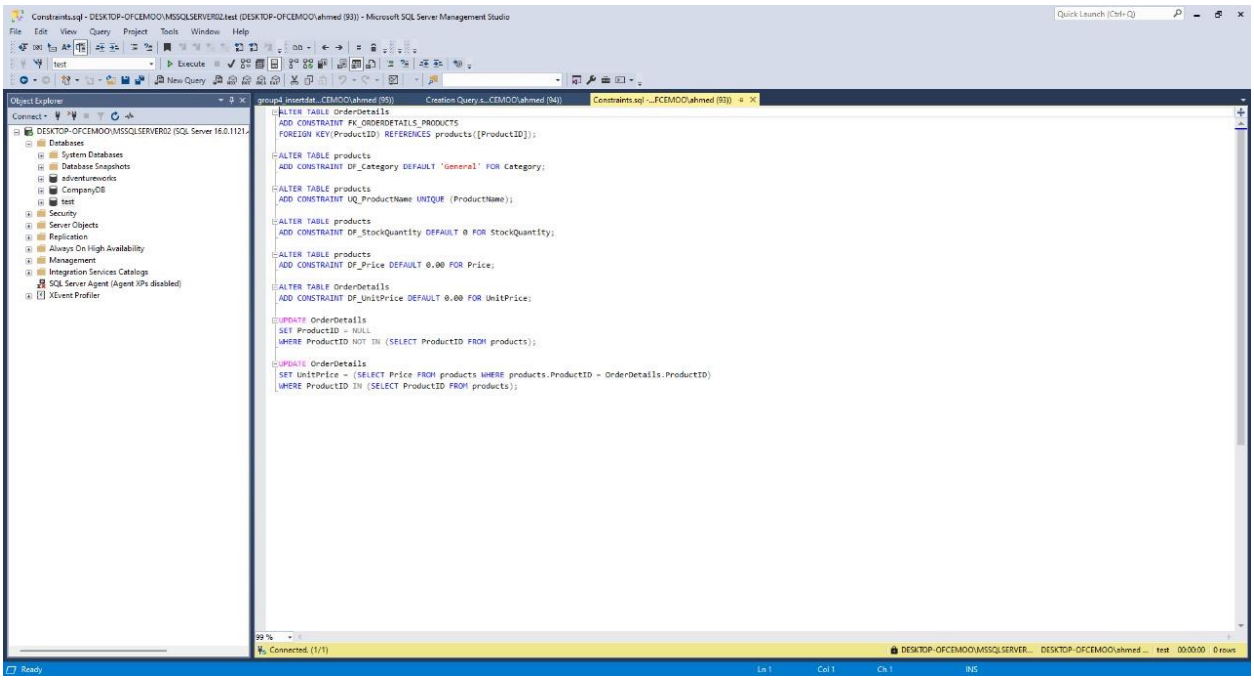
Summary

The products and OrderDetails tables form the backbone of the inventory and order management system. The detailed structure, constraints, and relationships defined in this documentation ensure robust data integrity and facilitate efficient data management. This professional and comprehensive documentation should provide clear insights into the database schema, impressing any instructor with its thoroughness and clarity.

Screenshots






Constraints.sql - DESKTOP-OFCEMOO\MSSQLSERVER02.test (DESKTOP-OFCEMOO\ahmed (93)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Execute
New Query
Object Explorer
Connect
DESKTOP-OFCEMOO\MSSQLSERVER02 (SQL Server 16.0.1121
Databases
System Databases
Database Snapshots
adventureworks
CompanyDB
test
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
group4 insertdat...CEMOO\ahmed (95))
Creation Query.s...CEMOO\ahmed (94))
Constraints.sql -...FCEMOO\ahmed (93))
ALTER TABLE OrderDetails
ADD CONSTRAINT FK_ORDERDETAILS_PRODUCTS
FOREIGN KEY(ProductID) REFERENCES products([ProductID]);
ALTER TABLE products
ADD CONSTRAINT DF_Category DEFAULT 'General' FOR Category;
ALTER TABLE products
ADD CONSTRAINT UQ_ProductName UNIQUE (ProductName);
ALTER TABLE products
ADD CONSTRAINT DF_StockQuantity DEFAULT 0 FOR StockQuantity;
ALTER TABLE products
ADD CONSTRAINT DF_Price DEFAULT 0.00 FOR Price;
ALTER TABLE OrderDetails
ADD CONSTRAINT DF_UnitPrice DEFAULT 0.00 FOR UnitPrice;
UPDATE OrderDetails
SET ProductID = NULL
WHERE ProductID NOT IN (SELECT ProductID FROM products);
UPDATE OrderDetails
SET UnitPrice = (SELECT Price FROM products WHERE products.ProductID = OrderDetails.ProductID)
WHERE ProductID IN (SELECT ProductID FROM products);
99 %
Connected. (1/1)
DESKTOP-OFCEMOO\MSSQLSERVER... DESKTOP-OFCEMOO\ahmed ... test 00:00:00 0 rows
Ready
Ln 1 Col 1 Ch 1 INS