

REPUBLICA DOMINICANA
INSTITUTO TECNOLOGICO DE LAS AMERICAS



INTRODUCCION A LA BASE DE DATOS

SECCION:

2025-C-1-1746-3090-TCD-005

PRACTICA FINAL

ULTIMA PRUEBA EVALUATORIA

Rossy Elania Arvelo Pérez

20240861

Rolando Oscar Garcia Guzman

Ultima Prueba Evaluatoria

Caso de Uso Teórico:

Implementación de un Sistema de Gestión de Inventario en una Empresa de Retail

Objetivo del Caso de Uso

El objetivo de este caso de uso es que los estudiantes de Ciencia de Datos apliquen los conceptos teóricos y prácticos relacionados con los lenguajes SQL DML (Data Manipulation Language), DDL (Data Definition Language) y DCL (Data Control Language) para diseñar, implementar y gestionar un sistema de base de datos que permita optimizar la gestión de inventarios de una empresa de retail.

Los estudiantes deberán demostrar habilidades en la creación, modificación y manipulación de bases de datos, al tiempo que aseguran un manejo seguro y controlado de los datos.

Contexto del Caso de Uso

La empresa de retail necesita un sistema de gestión de inventario que permita:

- Registrar productos y categorías.
- Actualizar y consultar información del inventario en tiempo real.
- Controlar accesos a la base de datos para garantizar la seguridad y la privacidad.

El sistema debe ser diseñado para mantener la integridad de los datos, permitir la gestión eficiente del inventario y proteger la información crítica.

Preguntas y Ejercicios para los Estudiantes

1. Data Definition Language (DDL)

Diseño del esquema de base de datos:

- Diseñe un esquema que incluya tablas para productos, categorías y movimientos de inventario (entradas y salidas).
- Utilice DDL para crear las tablas necesarias, asegurando relaciones adecuadas mediante claves primarias y foráneas.

Ejemplo de pregunta:

Escriba los comandos SQL necesarios para crear la tabla Productos con las siguientes columnas:

- id_producto (clave primaria, entero, autoincremental).
- nombre_producto (cadena, no nulo).
- id_categoria (clave foránea, entero).
- cantidad (entero, no nulo).

Modificación del esquema:

¿Cómo modificaría la tabla Productos para añadir una columna precio? Escriba el comando SQL correspondiente.

2. Data Manipulation Language (DML)

Inserción de datos:

- Inserte datos de ejemplo en las tablas creadas (productos, categorías, movimientos).
- Escriba un comando SQL para añadir un nuevo producto a la base de datos.

Consulta de datos:

Cree una consulta SQL para obtener:

- Todos los productos disponibles con su categoría.
- Productos con menos de 10 unidades en inventario.

Optimice la consulta para mostrar solo los productos ordenados por cantidad en orden ascendente.

Actualización y eliminación de datos:

- Escriba un comando SQL para actualizar la cantidad de un producto específico.
- Escriba un comando SQL para eliminar un producto discontinuado.

3. Data Control Language (DCL)

Control de accesos:

Escriba un comando SQL para crear un nuevo usuario gerente_inventario con permisos para consultar e insertar datos, pero sin la capacidad de eliminar registros.

Gestión de privilegios:

¿Cómo revocaría los permisos de un usuario en la tabla Movimientos? Escriba el comando SQL correspondiente.

Política de acceso seguro:

Proporcione recomendaciones sobre cómo utilizar DCL para implementar políticas de acceso que protejan datos sensibles.

Entrega Esperada

Código SQL:

- Scripts que implementen las estructuras de la base de datos (DDL).
- Comandos de inserción, consulta, actualización y eliminación de datos (DML).
- Comandos para la gestión de accesos y permisos (DCL)

Documentación:

- Una descripción de las decisiones de diseño tomadas para el esquema de la base de datos.
- Explicación del uso de DDL, DML y DCL en el contexto del caso.

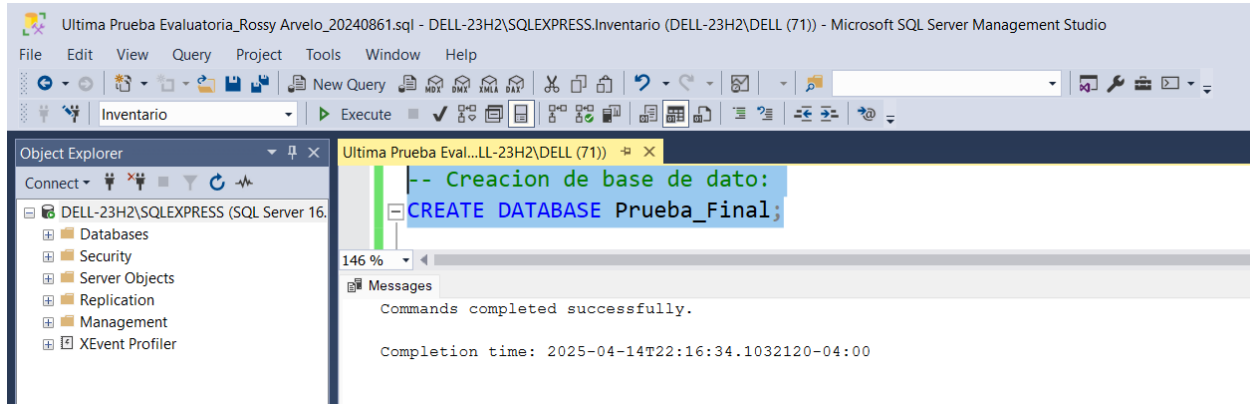
Evaluación práctica:

- Los estudiantes deben implementar y ejecutar el sistema en un entorno de base de datos y presentar los resultados de las consultas y operaciones realizadas.

Código SQL:

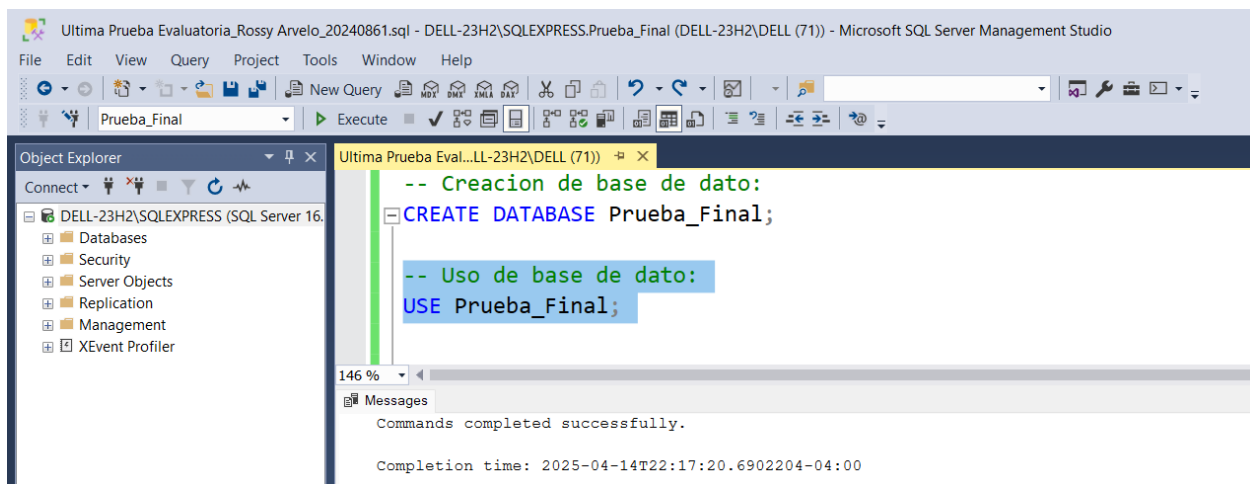
-- Creacion de base de dato:

CREATE DATABASE Inventario;



-- Uso de base de dato:

USE Inventario;



-- 1. CREACIÓN DEL ESQUEMA (DDL) --

-- Crear tabla de Categorías

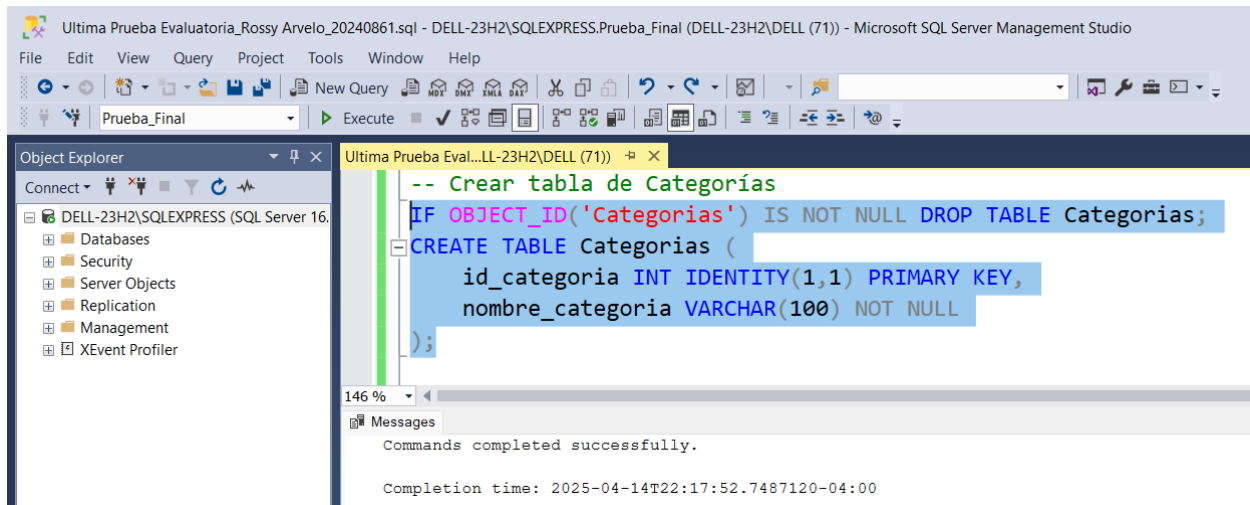
IF OBJECT_ID('Categorias') IS NOT NULL DROP TABLE Categorias;

CREATE TABLE Categorias (

id_categoria INT IDENTITY(1,1) PRIMARY KEY,

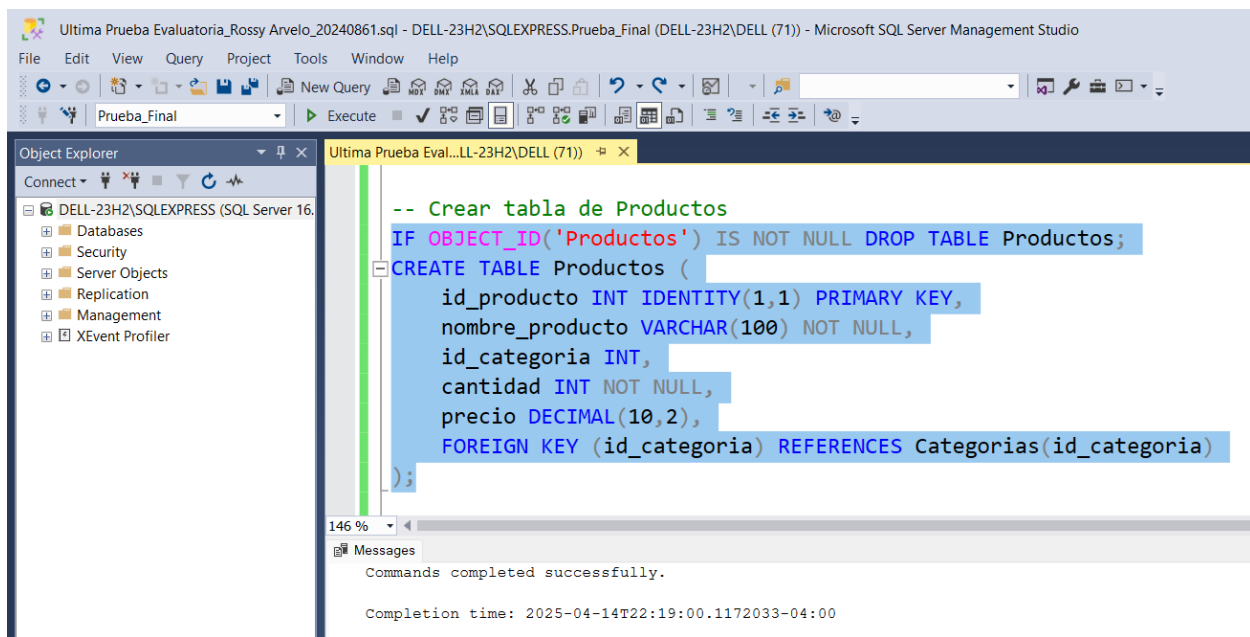
nombre_categoria VARCHAR(100) NOT NULL

);



-- Crear tabla de Productos

```
IF OBJECT_ID('Productos') IS NOT NULL DROP TABLE Productos;
CREATE TABLE Productos (
    id_producto INT IDENTITY(1,1) PRIMARY KEY,
    nombre_producto VARCHAR(100) NOT NULL,
    id_categoria INT,
    cantidad INT NOT NULL,
    precio DECIMAL(10,2),
    FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria)
);
```



-- Crear tabla de Movimientos de Inventario

IF OBJECT_ID('Movimientos') IS NOT NULL DROP TABLE Movimientos;

CREATE TABLE Movimientos (

id_movimiento INT IDENTITY(1,1) PRIMARY KEY,

id_producto INT,

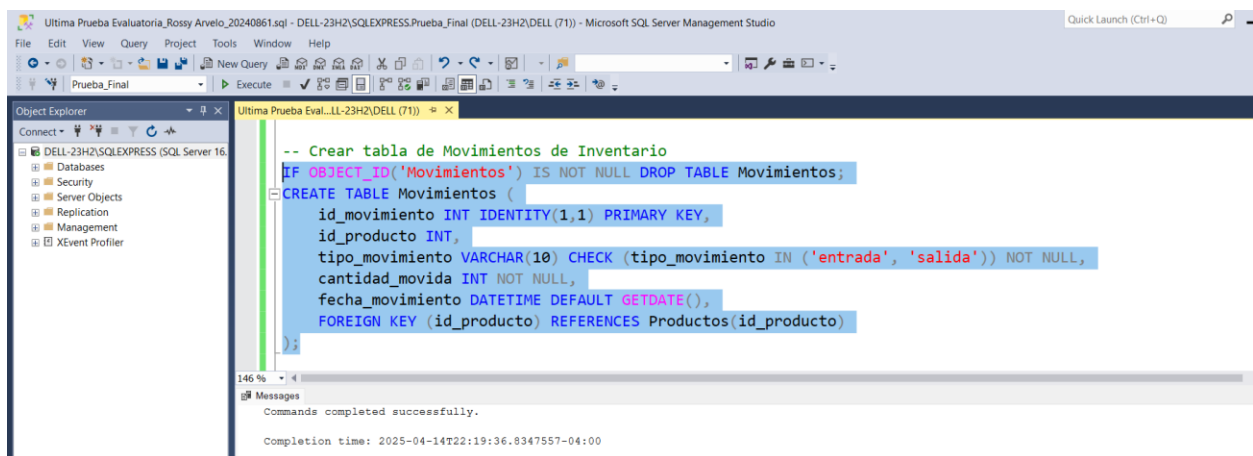
tipo_movimiento VARCHAR(10) CHECK (tipo_movimiento IN ('entrada', 'salida')) NOT NULL,

cantidad_movida INT NOT NULL,

fecha_movimiento DATETIME DEFAULT GETDATE(),

FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)

);



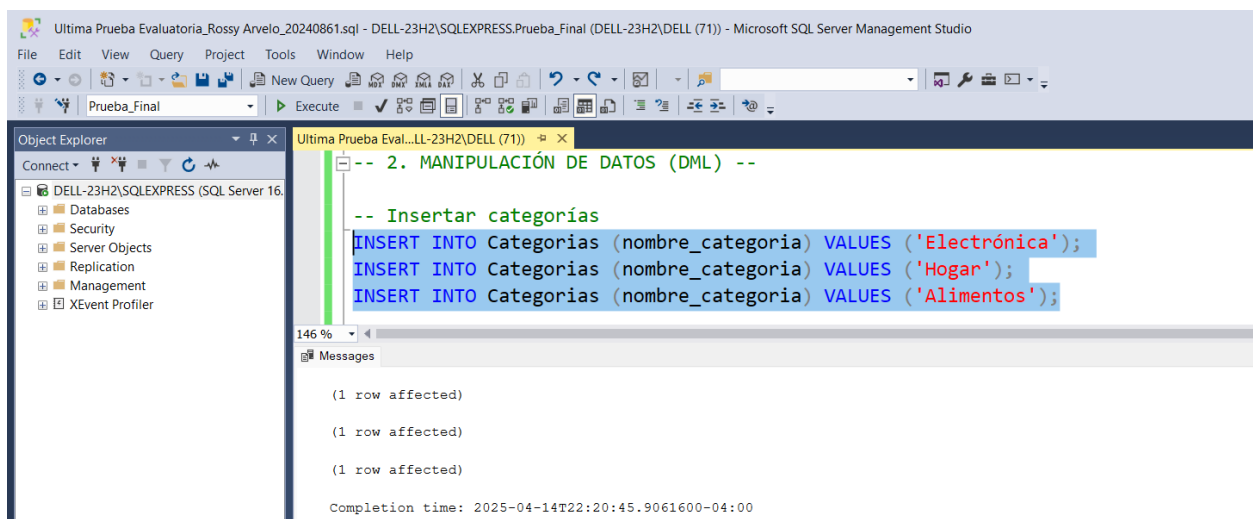
-- 2. MANIPULACIÓN DE DATOS (DML) --

-- Insertar categorías

INSERT INTO Categorías (nombre_categoria) VALUES ('Electrónica');

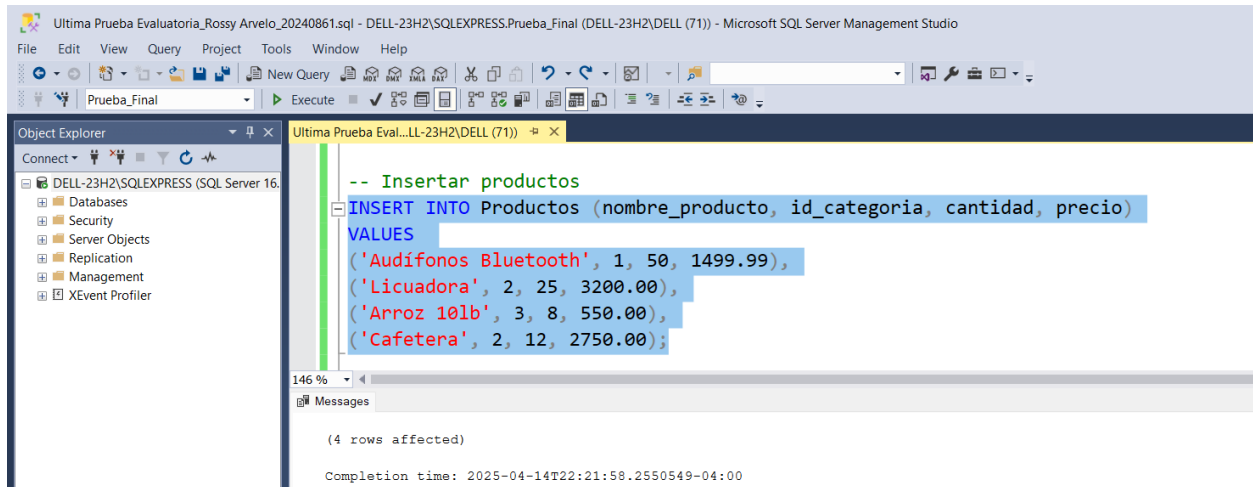
INSERT INTO Categorías (nombre_categoria) VALUES ('Hogar');

INSERT INTO Categorías (nombre_categoria) VALUES ('Alimentos');



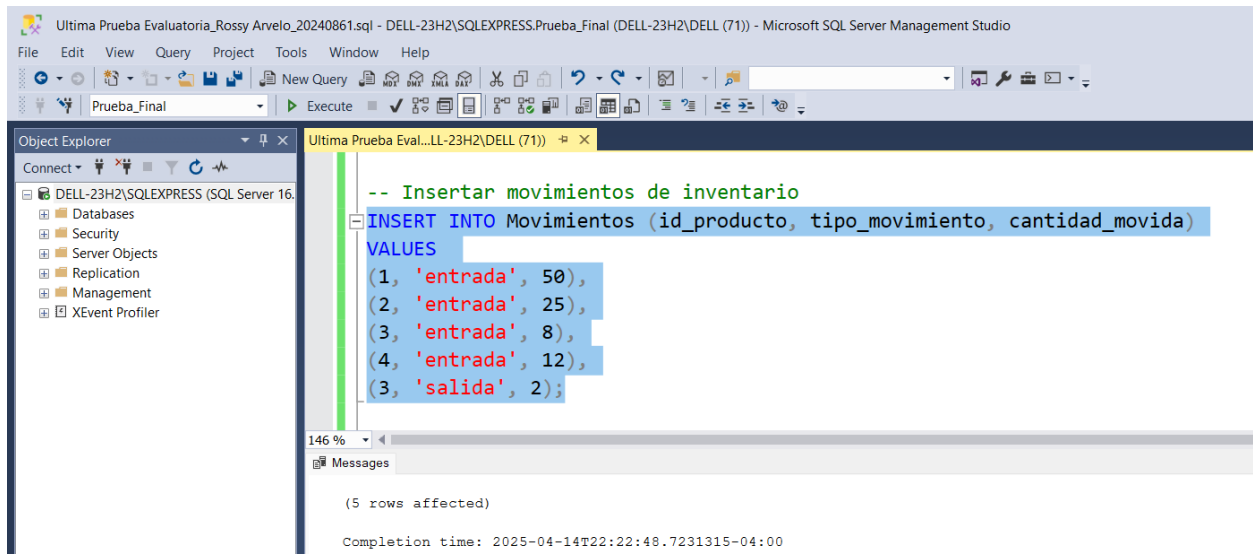
-- Insertar productos

```
INSERT INTO Productos (nombre_producto, id_categoria, cantidad, precio)
VALUES
('Audifonos Bluetooth', 1, 50, 1499.99),
('Licuadora', 2, 25, 3200.00),
('Arroz 10lb', 3, 8, 550.00),
('Cafetera', 2, 12, 2750.00);
```



-- Insertar movimientos de inventario

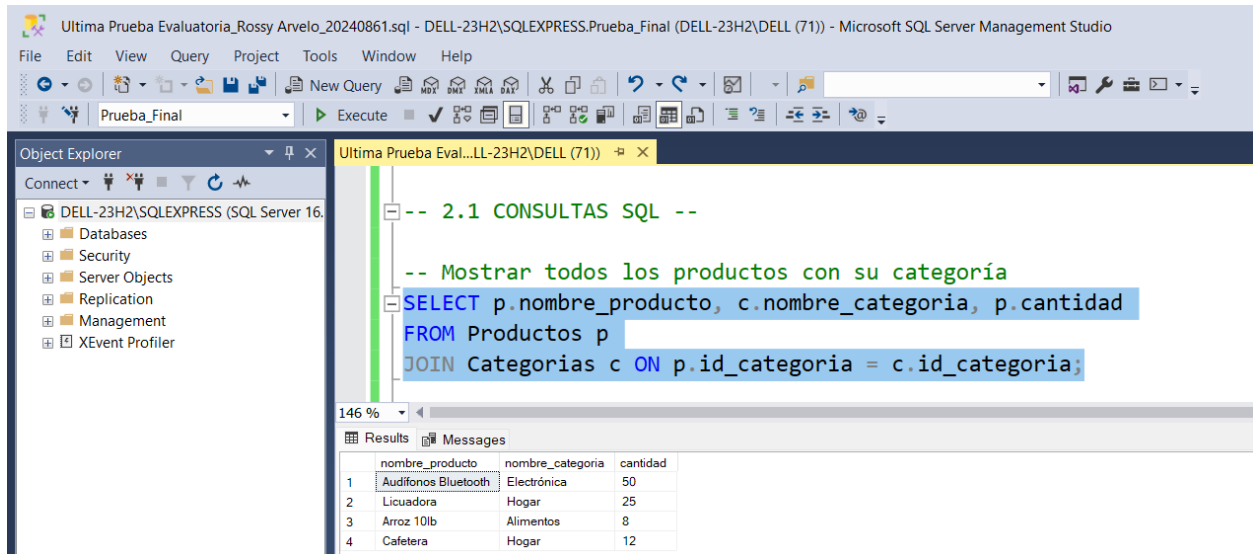
```
INSERT INTO Movimientos (id_producto, tipo_movimiento, cantidad_movida)
VALUES
(1, 'entrada', 50),
(2, 'entrada', 25),
(3, 'entrada', 8),
(4, 'entrada', 12),
(3, 'salida', 2);
```



-- 2.1 CONSULTAS SQL --

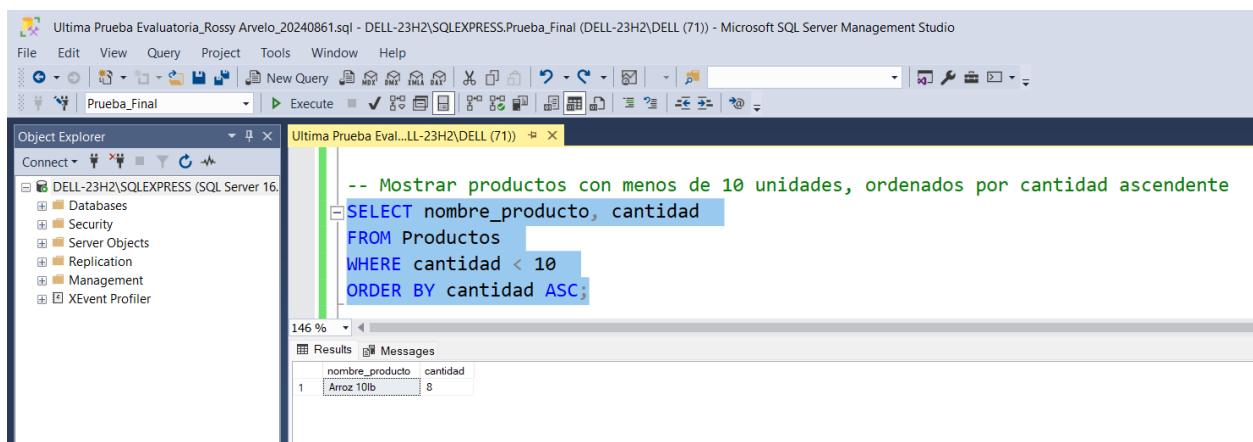
-- Mostrar todos los productos con su categoría

```
SELECT p.nombre_producto, c.nombre_categoria, p.cantidad
FROM Productos p
JOIN Categorías c ON p.id_categoria = c.id_categoria;
```



-- Mostrar productos con menos de 10 unidades, ordenados por cantidad ascendente

```
SELECT nombre_producto, cantidad
FROM Productos
WHERE cantidad < 10
ORDER BY cantidad ASC;
```



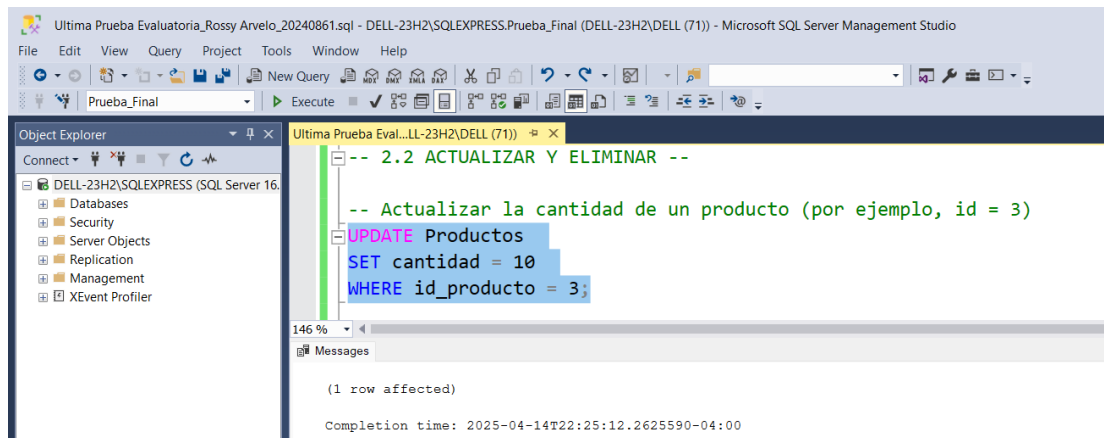
-- 2.2 ACTUALIZAR Y ELIMINAR --

-- Actualizar la cantidad de un producto (por ejemplo, id = 3)

UPDATE Productos

SET cantidad = 10

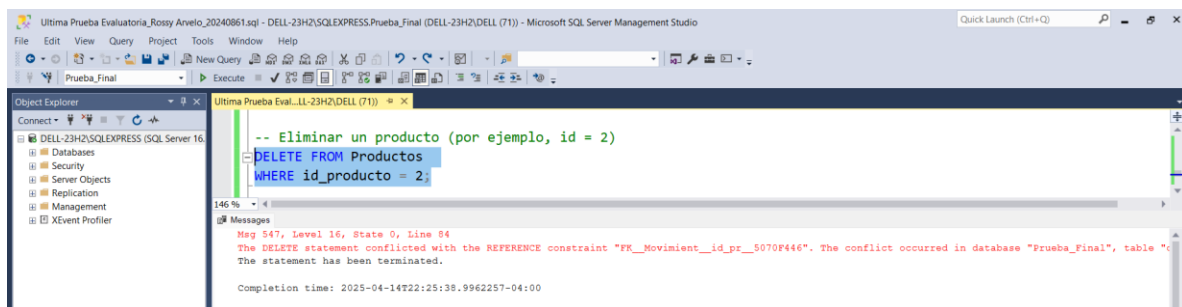
WHERE id_producto = 3;



-- Eliminar un producto (por ejemplo, id = 2)

DELETE FROM Productos

WHERE id_producto = 2;

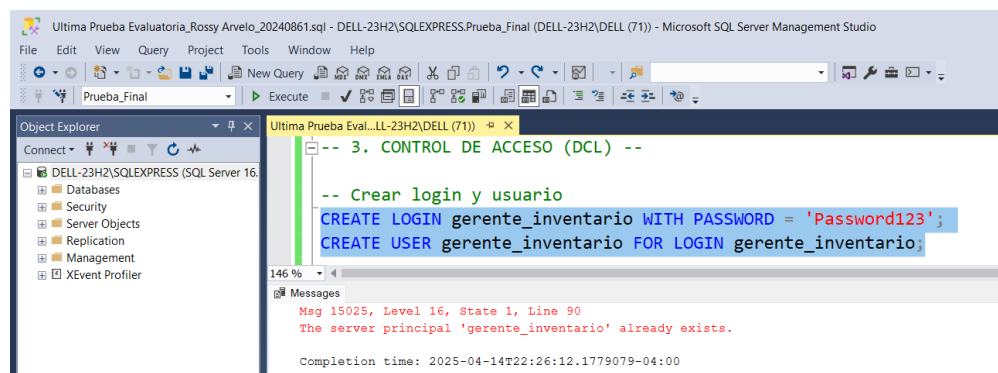


-- 3. CONTROL DE ACCESO (DCL) --

-- Crear login y usuario

CREATE LOGIN gerente_inventario WITH PASSWORD = 'Password123';

CREATE USER gerente_inventario FOR LOGIN gerente_inventario;

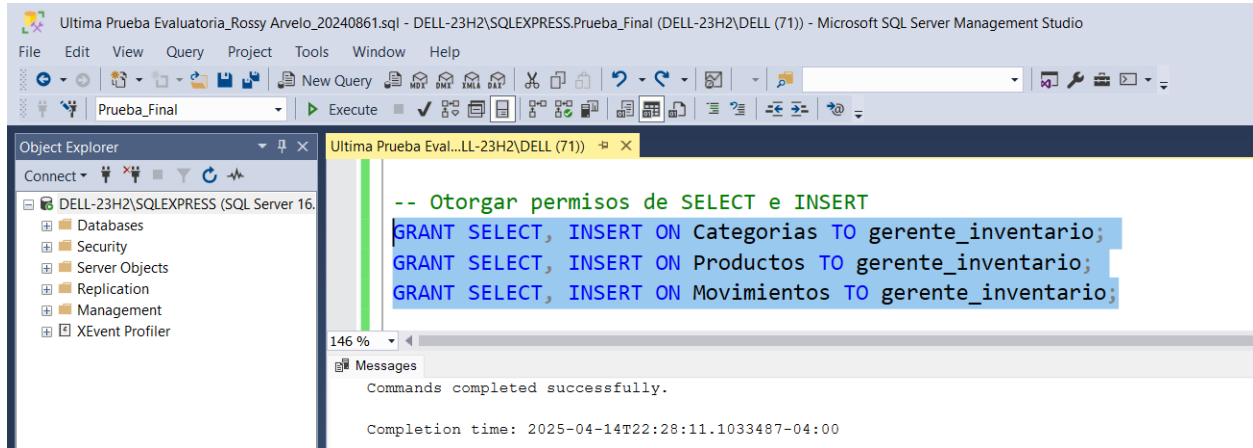


-- Otorgar permisos de SELECT e INSERT

GRANT SELECT, INSERT ON Categorías TO gerente_inventario;

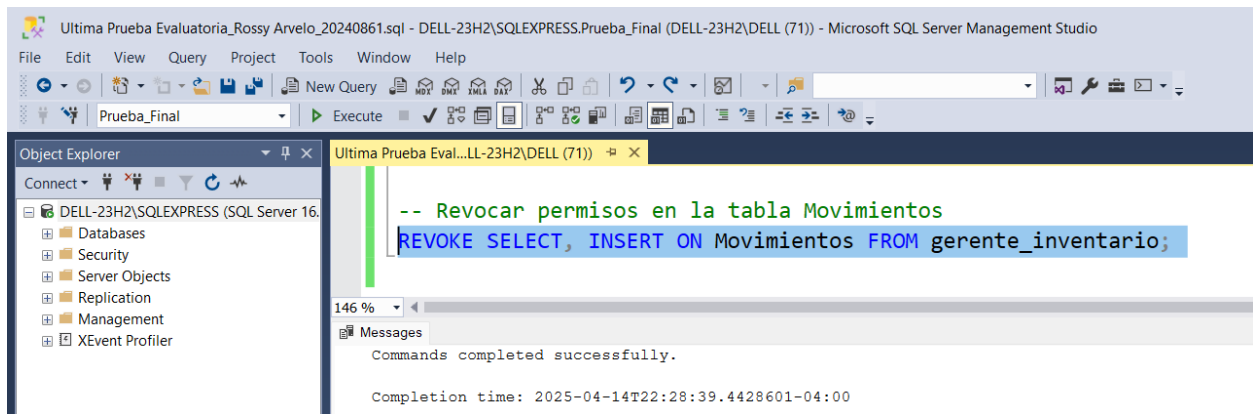
GRANT SELECT, INSERT ON Productos TO gerente_inventario;

GRANT SELECT, INSERT ON Movimientos TO gerente_inventario;



-- Revocar permisos en la tabla Movimientos

REVOKE SELECT, INSERT ON Movimientos FROM gerente_inventario;



Documentación del Proyecto: Sistema de Gestión de Inventario

1. Decisiones de Diseño del Esquema de Base de Datos

Para diseñar el sistema de gestión de inventario de la empresa de retail, se tomaron las siguientes decisiones:

- **Modelo relacional:** Se utilizó un enfoque relacional para mantener la integridad y consistencia de los datos, dividiendo la información en tablas relacionadas por claves primarias y foráneas.
- **Tablas creadas:**
 - **Categorías:** Contiene las distintas categorías a las que puede pertenecer un producto. Esta tabla permite clasificar los productos de manera ordenada.
 - **Productos:** Almacena información básica del inventario como nombre, cantidad disponible, precio y categoría a la que pertenece. Está relacionada con la tabla Categorías mediante una clave foránea.
 - **Movimientos:** Registra las entradas y salidas de inventario, facilitando el seguimiento de cada producto. Se usa un campo tipo ENUM para distinguir entre movimientos de entrada y salida.
- **Tipos de datos:** Se usaron tipos como INT, VARCHAR, DECIMAL, ENUM y DATETIME, buscando un balance entre rendimiento y precisión. Además, se agregó una columna precio a la tabla Productos para incluir información comercial relevante.
- **Restricciones de integridad:**
 - Las claves primarias garantizan que cada fila sea única.
 - Las claves foráneas aseguran la relación entre productos, categorías y movimientos.
 - Las columnas importantes se definieron como NOT NULL para evitar registros incompletos.

2. Uso de SQL en el Contexto del Caso

DDL – Data Definition Language

El DDL se utilizó para definir la estructura de la base de datos. Con él se crearon las tablas necesarias (CREATE TABLE), se establecieron claves primarias y foráneas para mantener relaciones y restricciones, y se modificó una tabla existente con ALTER TABLE para añadir la columna de precio a los productos.

Ejemplos utilizados:

- CREATE TABLE para definir Categorías, Productos y Movimientos.
- ALTER TABLE para añadir la columna precio a la tabla Productos.

DML – Data Manipulation Language

El DML permitió insertar, consultar, actualizar y eliminar información dentro de la base de datos. Estas acciones son fundamentales para mantener el inventario actualizado.

Ejemplos utilizados:

- INSERT INTO para agregar nuevas categorías, productos y movimientos de inventario.
- SELECT para consultar los productos con sus respectivas categorías y filtrar los que tienen pocas unidades.
- UPDATE para modificar la cantidad disponible de un producto.
- DELETE para eliminar productos discontinuados.

DCL – Data Control Language

El DCL se aplicó para controlar el acceso de los usuarios a la base de datos, asegurando que solo personas autorizadas puedan manipular ciertos datos.

Ejemplos utilizados:

- CREATE USER para registrar un nuevo usuario llamado gerente_inventario.
- GRANT para darle acceso solo a lectura e inserción de datos.
- REVOKE para quitarle permisos en la tabla Movimientos, limitando su capacidad de afectar registros sensibles.

Además, se recomendó implementar una política de acceso seguro basada en roles y permisos mínimos para proteger la integridad y privacidad de la información.