



## Ejercicios de Base de Datos

### Ejercicio 1 — Gestión de Biblioteca

Crea una base de datos llamada 'biblioteca' con las tablas 'autores' y 'libros', incluyendo claves primarias y foráneas, inserciones y consultas.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the 'biblioteca' database structure, including tables 'dbo.autores' and 'dbo.libros'. The main window shows a SQL query window with the following code:

```
1  
2  
3  
4  
5  
6  
7  
8  
USE biblioteca;  
SELECT libros.titulo, autores.nombre  
FROM libros  
INNER JOIN autores ON libros.id_autor = autores.id_autor;
```

Below the query window, the Results pane shows the output of the query:

	titulo	nombre
1	Cien años de soledad	Gabriel García Márquez
2	La ciudad y los perros	Mario Vargas Llosa

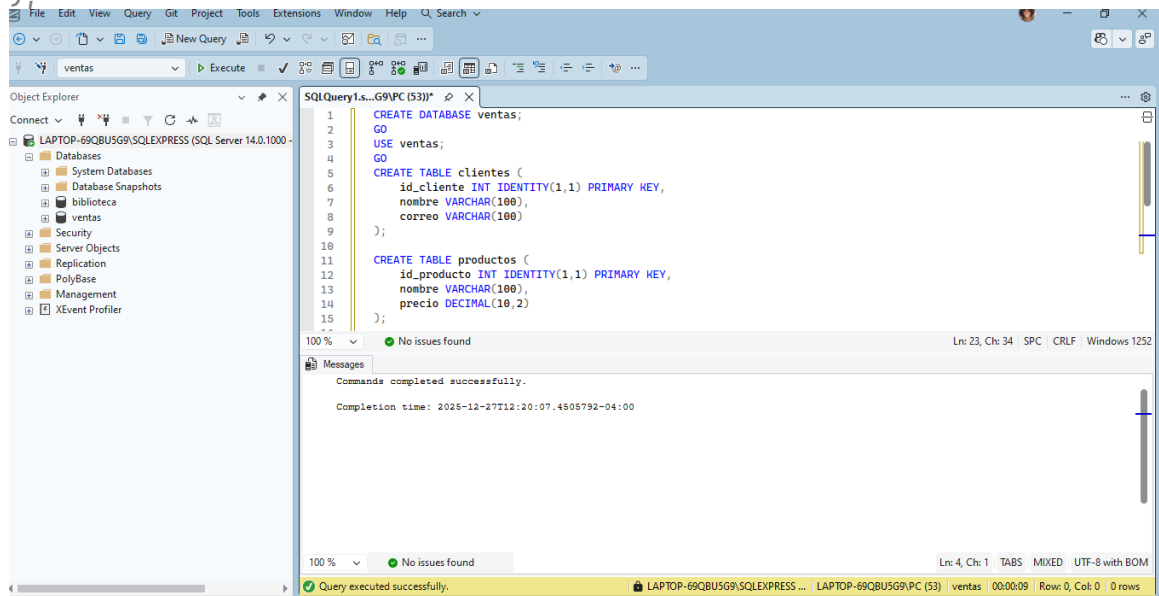
### Ejercicio 2 — Sistema de Ventas

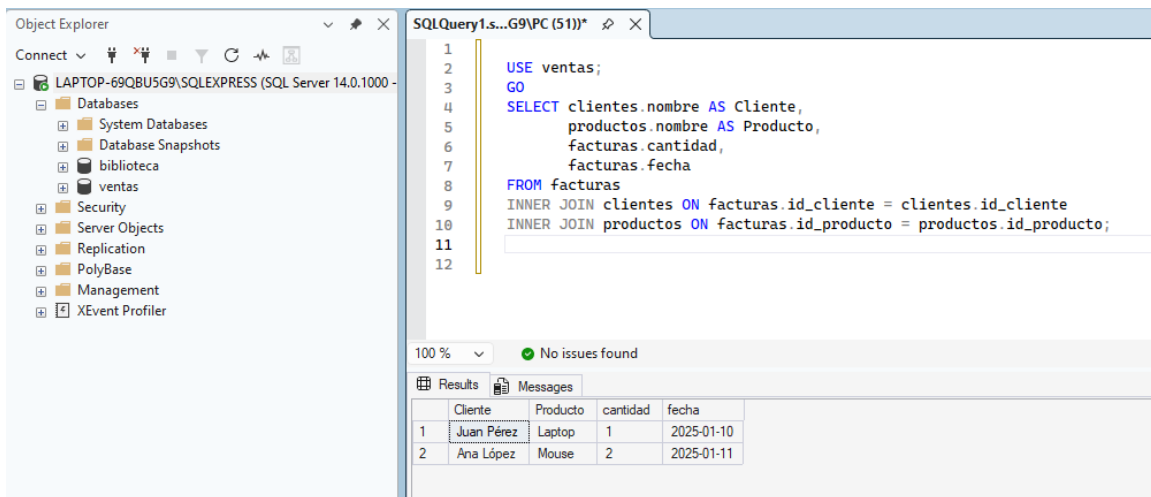
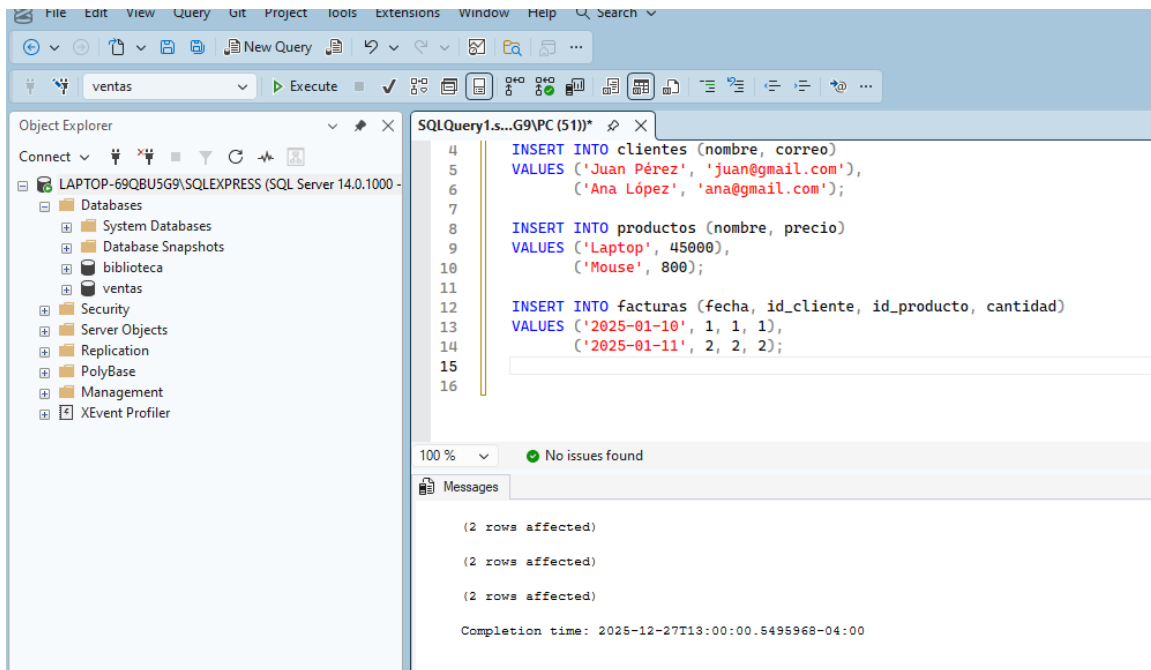
Crea una base de datos llamada 'ventas' con las tablas 'clientes', 'productos' y 'facturas'. Define las relaciones, inserta datos y realiza consultas.

```
CREATE DATABASE ventas;  
GO  
USE ventas;  
GO  
CREATE TABLE clientes (  
    id_cliente INT IDENTITY(1,1) PRIMARY KEY,  
    nombre VARCHAR(100),  
    correo VARCHAR(100)  
);
```

```
CREATE TABLE productos (  
    id_producto INT IDENTITY(1,1) PRIMARY KEY,  
    nombre VARCHAR(100),  
    precio DECIMAL(10,2)  
);
```

```
CREATE TABLE facturas (  
    id_factura INT IDENTITY(1,1) PRIMARY KEY,  
    fecha DATE,  
    id_cliente INT,  
    id_producto INT,  
    cantidad INT,  
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),  
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)  
);
```





### Ejercicio 3 — Control de Estudiantes

Crea una base de datos llamada 'colegio' con las tablas 'estudiantes', 'cursos' y 'matriculas'. Incluye claves primarias, foráneas, inserciones y consultas.

```

CREATE DATABASE colegio;
GO
USE colegio;
GO
CREATE TABLE estudiantes (
    id_estudiante INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(100),
    edad INT
);

```

```
CREATE TABLE cursos (
    id_curso INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(100),
    credits INT
);
```

```
CREATE TABLE matriculas (
    id_matricula INT IDENTITY(1,1) PRIMARY KEY,
    id_estudiante INT,
    id_curso INT,
    fecha DATE,
    FOREIGN KEY (id_estudiante) REFERENCES estudiantes(id_estudiante),
    FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)
);
```

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the server structure for 'LAPTOP-69QBU5G9\SQLEXPRESS (SQL Server 14.0.1000)'. The central pane shows the execution of a SQL script named 'SQLQuery1.sql...5G9\PC (51)'. The script content is as follows:

```
16
17 CREATE TABLE matriculas (
18     id_matricula INT IDENTITY(1,1) PRIMARY KEY,
19     id_estudiante INT,
20     id_curso INT,
21     fecha DATE,
22     FOREIGN KEY (id_estudiante) REFERENCES estudiantes(id_estudiante),
23     FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)
24 );
25
```

The status bar indicates '100 %' zoom and 'No issues found'. The Messages pane at the bottom shows the following output:

```
Commands completed successfully.
Completion time: 2025-12-27T13:04:05.5530906-04:00
```

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the server structure for 'LAPTOP-69QBU5G9\SQLEXPRESS (SQL Server 14.0.1000)'. The central pane shows the execution of a SQL script named 'SQLQuery1.s...G9\PC (51)'. The script content is as follows:

```
1 USE colegio;
2 GO
3 SELECT estudiantes.nombre AS Estudiante,
4        cursos.nombre AS Curso
5 FROM matriculas
6 INNER JOIN estudiantes ON matriculas.id_estudiante = estudiantes.id_estudiante
7 INNER JOIN cursos ON matriculas.id_curso = cursos.id_curso;
8
```

The status bar indicates '100 %' zoom and '3' errors/warnings. The Results pane at the bottom shows the following data:

	Estudiante	Curso
1	Carlos Gómez	Base de Datos
2	Maria Rodríguez	Programación