

### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE Elige un elemento.





## II. INFORME

### 2.1 Objetivos

#### General:

Implementar y validar operaciones de bases de datos distribuidas en SQL Server, aplicando transacciones, integridad referencial y manejo de errores para garantizar la consistencia de los datos.

### **Específicos:**

- Crear y gestionar transacciones atómicas para asegurar que operaciones múltiples se completen exitosamente o se reviertan en caso de error.
- Verificar la integridad referencial entre tablas relacionadas mediante claves primarias y foráneas.
- Manejar errores con TRY-CATCH para controlar excepciones durante operaciones críticas.

### 2.2 Modalidad

- Práctica guiada en laboratorio con supervisión del docente.
- Individual o en equipos de 2 personas (según indicaciones del profesor).

## 2.3 Tiempo de duración

- **Presenciales:** 2 horas (tiempo en clase).
- No presenciales: 1 hora adicional para redacción del informe.

### 2.4 Instrucciones

- 1. Conectarse a SQL Server Management Studio (SSMS).
- 2. Ejecutar scripts SQL proporcionados en la guía.
- 3. Documentar resultados y capturas de pantalla.
- 4. Validar cada paso con el docente antes de continuar.

### 2.5 Listado de equipos, materiales y recursos

- Computador con Windows/Linux/macOS.
- SQL Server Management Studio (SSMS) instalado.
- Acceso a la base de datos CentroMedicoDB.

TAC (Tecnol	ogías para	el Apre	ndizaje y (	Conocimiento)	) empleados	s en la gu	ía práctica:
[ D1 . C	1	. •					

- ⊠Plataformas educativas
- ⊠Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ⊠Recursos audiovisuales



### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL







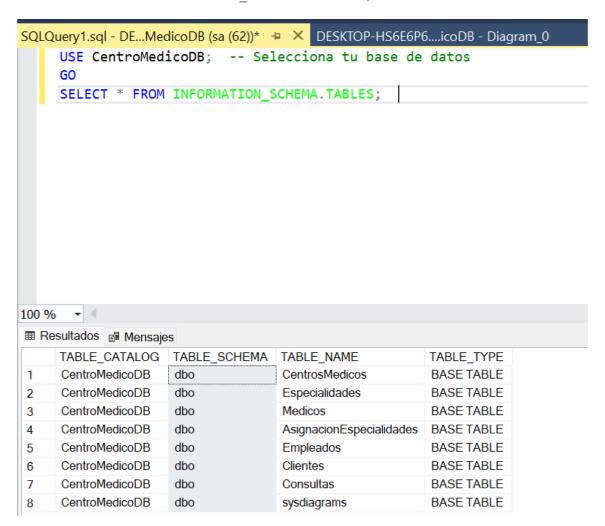
☐ Gamificación	
☐ Inteligencia Artificial	
Otros (Especifique):	

## 2.6 Actividades por desarrollar

### 1. Verificar conexión y estructura de la base de datos

- Abrimos SQL Server Management Studio (SSMS) y nos conectamos al servidor.
- Seleccionamos la base de datos "CentroMedicoDB".
- Verificamos las tablas existentes ejecutando:

SELECT \* FROM INFORMATION\_SCHEMA.TABLES;



## 2. Probar integridad referencial con datos existentes

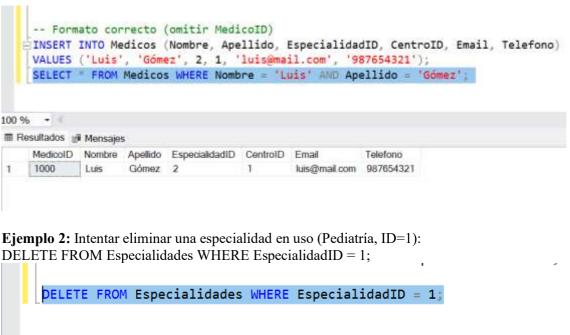
- Ejemplo 1: Insertamos un médico con una especialidad inexistente: INSERT INTO Medicos (Nombre, Apellido, EspecialidadID, CentroID, Email, Telefono) VALUES ('Luis', 'Gómez', 2, 1, 'luis@mail.com', '987654321');



## FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL



CARRERA DE Elige un elemento. CICLO ACADÉMICO: MARZO – JULIO 2025



100 % ▼

Mensajes

Mens. 547, Nivel 16, Estado 0, Línea 15

Instrucción DELETE en conflicto con la restricción REFERENCE 'FK\_Medicos\_Esis Se terminó la instrucción.

Hora de finalización: 2025-04-24T19:52:15.5832676-05:00

#### 2. Transacciones con atomicidad (Commit y Rollback)

**Transacción exitosa:** Crear un centro médico y asignar un médico: BEGIN TRANSACTION;

INSERT INTO CentrosMedicos (Nombre, Ciudad, Direccion, Telefono) VALUES ('Clínica Norte', 'Quito', 'Av. Principal 123', '022222222');

DECLARE @NuevoCentroID INT = SCOPE IDENTITY();

INSERT INTO Medicos (Nombre, Apellido, EspecialidadID, CentroID, Email) VALUES ('Marta', 'Vega', 3, @NuevoCentroID, 'marta@clinicanorte.ec'); COMMIT TRANSACTION;

-- Verificar datos

SELECT \* FROM CentrosMedicos WHERE Nombre = 'Clínica Norte'; SELECT \* FROM Medicos WHERE Nombre = 'Marta';



#### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL







```
BEGIN TRANSACTION;
         INSERT INTO CentrosMedicos (Nombre, Ciudad, Direccion, Telefono)
         VALUES ('Clinica Norte', 'Quito', 'Av. Principal 123', '0222222222');
         DECLARE @NuevoCentroID INT = SCOPE_IDENTITY();
         INSERT INTO Medicos (Nombre, Apellido, EspecialidadID, CentroID, Email)
         VALUES ('Marta', 'Vega', 3, @NuevoCentroID, 'marta@clinicanorte.ec');
    COMMIT TRANSACTION;
     -- Verificar datos
    SELECT * FROM CentrosMedicos WHERE Nombre = 'Clinica Norte';
    SELECT * FROM Medicos WHERE Nombre = 'Marta';
100 %
III Resultados 🛒 Mensajes
    CentrolD Nombre
                     Cludad Direction
                                          Telefono
            Clínica Norte Quito Av. Principal 123 022222222
     MedicolD Nombre Apellido EspecialidadID CentrolD Email
    1003 Marta
                   Vega 3
                                      5
                                               marta@clinicanorte.ec NULL
```

**Transacción fallida:** Intentar asignar un médico a un centro inexistente:

BEGIN TRANSACTION;

DELETE FROM Especialidades WHERE EspecialidadID = 3; -- Especialidad en uso por "Marta Vega"

ROLLBACK TRANSACTION; -- Revierte la operación

#### 3. Pruebas concurrentes

#### Sesión 1 (Actualización):

BEGIN TRANSACTION;

UPDATE Medicos SET Email = 'marta.nuevo@clinicanorte.ec' WHERE Nombre = 'Marta';

-- Mantener la transacción abierta (sin COMMIT)

```
BEGIN TRANSACTION;

UPDATE Medicos SET Email = 'marta.nuevo@clinicanorte.ec' WHERE Nombre = 'Marta';

-- Mantener la transacción abierta (sin COMMIT)

100 % - 4

@ Mensajes

(0 filas afectadas)

Hora de finalización: 2025-04-24720:12:51.2677789-05:00
```



### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL







#### Sesión 2 (Lectura):

SELECT \* FROM Medicos WHERE Nombre = 'Marta'; -- Bloqueado hasta que Sesión 1 finalice



### 4. Manejo de errores con TRY...CATCH

Ejemplo: Insertar una consulta con un ClientelD inexistente.sql

**BEGIN TRY** 

BEGIN TRANSACTION;

INSERT INTO Consultas (MedicoID, ClienteID, FechaConsulta)

VALUES (1, 9999, GETDATE());

COMMIT TRANSACTION;

**END TRY** 

**BEGIN CATCH** 

ROLLBACK TRANSACTION;

PRINT 'Error: ' + ERROR MESSAGE();

## END CATCH;

```
BEGIN TRY
BEGIN TRY
BEGIN TRANSACTION;

INSERT INTO Consultas (MedicoID, ClienteID, FechaConsulta)
VALUES (1, 9999, GETDATE());

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

PRINT 'Error: ' * ERROR_MESSAGE();

END CATCH;

Mercages

(0 files afectades)

Error: Instrucción INSERT en conflicto con la restricción FOREIGN SEY 'FK_Consultas_Clien_4708AE45', El conflicto ha aparecido en la Hora de finalización: 2025-04-24726:32:14.8767943-05:00
```

#### 5. Verificación final

#### **Consultar datos nuevos:**

-- Verificar el nuevo centro médico

SELECT \* FROM CentrosMedicos WHERE Nombre = 'Clínica Norte';

-- Verificar el médico asignado al nuevo centro

SELECT \* FROM Medicos WHERE Nombre = 'Marta' AND Apellido = 'Vega';

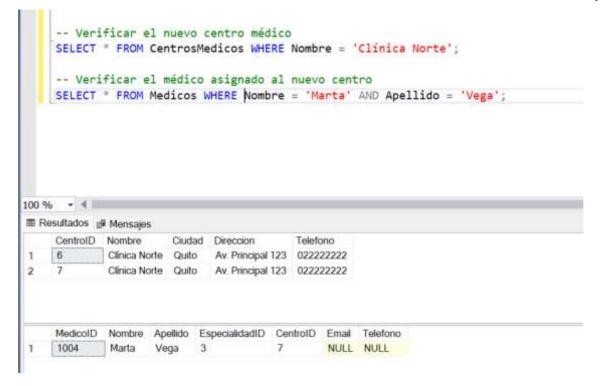


### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL









#### 2.7 Resultados obtenidos

- Transacción exitosa:
  - o Se insertó el centro médico "Clínica Norte" y se asignó la médica "Marta Vega".
  - La atomicidad de la transacción se cumplió: ambas operaciones se ejecutaron como una sola unidad.

## 2.8 Habilidades blandas empleadas en la práctica

	Liderazgo
	Trabajo en equipo
	Comunicación asertiva
	La empatía
$\boxtimes$	Pensamiento crítico
	Flexibilidad
$\boxtimes$	La resolución de conflictos
$\boxtimes$	Adaptabilidad
$\boxtimes$	Responsabilidad

### 2.9 Conclusiones

- Las transacciones garantizan que operaciones múltiples se ejecuten de manera atómica, evitando estados inconsistentes.
- El uso de SCOPE\_IDENTITY() asegura la correcta relación entre tablas con claves foráneas.

#### 2.10 Recomendaciones

- Validar siempre la existencia de claves foráneas antes de insertar datos.
- Usar transacciones en operaciones críticas que involucren múltiples tablas.



#### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE Elige un elemento.





#### 2.11 Referencias bibliográficas

[1] Microsoft Docs, "Claves foráneas en SQL Server", 2023. [En línea]. Disponible: https://docs.microsoft.com/es-es/sql/relational-databases/tables/createforeign-key-relationships.

[2] SQL Server Tutorial, "Manejo de errores con TRY CATCH", 2023. [En línea]. Disponible: <a href="https://www.sqlservertutorial.net/">https://www.sqlservertutorial.net/</a>.

#### 2.12 Anexos

que Sesión 1 finalice

```
Scripts SQL completos usados en la práctica.
   USE CentroMedicoDB; -- Selecciona tu base de datos
   SELECT * FROM INFORMATION SCHEMA.TABLES;
   INSERT INTO Medicos (Nombre, Apellido, EspecialidadID, CentroID,
   Email, Telefono)
   VALUES ('Luis', 'Gómez', 2, 1, 'luis@mail.com', '987654321');
   SET IDENTITY_INSERT Medicos OFF; -- Restaura el autoincremento
   -- Formato correcto (omitir MedicoID)
   INSERT INTO Medicos (Nombre, Apellido, EspecialidadID, CentroID,
   Email, Telefono)
   VALUES ('Luis', 'Gómez', 2, 1, 'luis@mail.com', '987654321');
   SELECT * FROM Medicos WHERE Nombre = 'Luis' AND Apellido =
   'Gómez';
   DELETE FROM Especialidades WHERE EspecialidadID = 1;
   BEGIN TRANSACTION;
       INSERT INTO CentrosMedicos (Nombre, Ciudad, Direccion,
   Telefono)
       VALUES ('Clínica Norte', 'Quito', 'Av. Principal 123',
   '02222222');
       DECLARE @NuevoCentroID INT = SCOPE_IDENTITY();
       INSERT INTO Medicos (Nombre, Apellido, EspecialidadID,
   CentroID, Email)
       VALUES ('Marta', 'Vega', 3, @NuevoCentroID,
   'marta@clinicanorte.ec');
   COMMIT TRANSACTION;
   -- Verificar datos
   SELECT * FROM CentrosMedicos WHERE Nombre = 'Clinica Norte';
   SELECT * FROM Medicos WHERE Nombre = 'Marta';
   BEGIN TRANSACTION;
       DELETE FROM Especialidades WHERE EspecialidadID = 3; --
   Especialidad en uso por "Marta Vega"
   ROLLBACK TRANSACTION; -- Revierte la operación
   BEGIN TRANSACTION:
       UPDATE Medicos SET Email = 'marta.nuevo@clinicanorte.ec' WHERE
   Nombre = 'Marta';
       -- Mantener la transacción abierta (sin COMMIT)
   SELECT * FROM Medicos WHERE Nombre = 'Marta'; -- Bloqueado hasta
```



### FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL







```
BEGIN TRY
       BEGIN TRANSACTION;
           INSERT INTO Consultas (MedicoID, ClienteID, FechaConsulta)
           VALUES (1, 9999, GETDATE());
       COMMIT TRANSACTION;
   END TRY
   BEGIN CATCH
       ROLLBACK TRANSACTION;
       PRINT 'Error: ' + ERROR_MESSAGE();
   END CATCH;
   -- Centros médicos nuevos
   SELECT * FROM CentrosMedicos WHERE CentroID = 10;
   -- Médicos asignados al nuevo centro
   SELECT * FROM Medicos WHERE CentroID = 10;
   DECLARE @NuevoCentroID INT;
   INSERT INTO CentrosMedicos (Nombre, Ciudad, Direccion, Telefono)
   VALUES ('Clínica Norte', 'Quito', 'Av. Principal 123',
   '02222222');
   SET @NuevoCentroID = SCOPE_IDENTITY(); -- Obtiene el ID generado
   BEGIN TRANSACTION;
       INSERT INTO CentrosMedicos (Nombre, Ciudad, Direccion,
   Telefono)
       VALUES ('Clínica Norte', 'Quito', 'Av. Principal 123',
   '02222222');
       DECLARE @NuevoCentroID INT = SCOPE IDENTITY();
       INSERT INTO Medicos (Nombre, Apellido, EspecialidadID,
   CentroID)
       VALUES ('Marta', 'Vega', 3, @NuevoCentroID);
   COMMIT TRANSACTION; -- ¡Confirmar los cambios!
   -- Verificar el nuevo centro médico
   SELECT * FROM CentrosMedicos WHERE Nombre = 'Clinica Norte';
   -- Verificar el médico asignado al nuevo centro
   SELECT * FROM Medicos WHE
Diagrama de relaciones entre tablas
```



## FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE Elige un elemento.

CICLO ACADÉMICO: MARZO – JULIO 2025



