



Universidad Nacional del Nordeste



**Facultad de Ciencias Exactas Naturales y
Agrimensura**

Carrera: Licenciatura en Sistemas de Información

Cátedra: Bases de Datos I

PROYECTO

**“Manejo de transacciones y transacciones
anidadas”**

Alumnos:

- Gonzalez Pasi, Farid José
- Frias, Javier Antonio
- Gomez, Mauricio Agustín
- Garcia, Brenda

Profesor: Darío Villegas

Año: 2023

Índice de contenido

Índice de contenido.....	2
1 Introducción	3
1.1 Manejo de transacciones y transacciones anidadas.....	3
1.2 Problema de investigación.....	3
1.3 Objetivo del proyecto	3
1.3.1 Objetivos Generales.....	3
1.3.2 Objetivos Específicos	3
2 Marco conceptual o referencial.....	4
3 Metodología seguida.....	6
4 Desarrollo del tema	6
5 Conclusiones.....	19
6 Bibliografía	19

1 Introducción

1.1 Manejo de transacciones y transacciones anidadas

El tema de este proyecto se centra en el manejo de transacciones y transacciones anidadas en el contexto de bases de datos relacionales, con un enfoque en la aplicación de operaciones SQL para garantizar la consistencia de los datos. Para mostrar este proceso, se realizará una serie de operaciones que involucren la inserción de registros en diferentes tablas de una base de datos, así como una actualización condicional de los mismos, con el propósito de comprender cómo se gestionan las transacciones en las bases de datos.

1.2 Problema de investigación

El problema que se aborda en este informe radica en la necesidad de comprender y aplicar el concepto de transacciones en el contexto de las bases de datos. Las transacciones son fundamentales para garantizar la integridad de los datos y para asegurarse de que las operaciones se completen con éxito o se deshagan por completo en caso de errores. Por lo tanto, es crucial entender cómo se pueden llevar a cabo transacciones anidadas y cómo se pueden manejar situaciones de error de manera consistente.

1.3 Objetivo del proyecto

1.3.1 Objetivos Generales

El objetivo general de este Trabajo Práctico es adquirir conocimiento sobre el manejo de transacciones en bases de datos relacionales, centrándonos en la consistencia de los datos y en la importancia de asegurar que las operaciones se realicen en su totalidad o se deshagan si se producen errores, con la finalidad de proporcionar una comprensión práctica de cómo las transacciones y las transacciones anidadas funcionan en el entorno de las bases de datos, y cómo se pueden aplicar de manera efectiva para mantener la integridad de los datos.

1.3.2 Objetivos Específicos

Los objetivos específicos que se abordarán en este trabajo son los siguientes:

- Comprender los conceptos fundamentales de las transacciones en bases de datos y su importancia en la integridad de los datos.
- Diseñar un conjunto de operaciones SQL que incluyan la inserción de registros en las tablas de una base de datos.
- Implementar una transacción que garantice que las operaciones de inserción se completen con éxito o se deshagan por completo si se produce un error.
- Introducir intencionalmente un error después de la inserción en una tabla y observar cómo se gestionan las transacciones para mantener la consistencia de los datos.
- Analizar y expresar conclusiones basadas en las pruebas realizadas, destacando la importancia de las transacciones en la gestión de datos y la capacidad de recuperación en situaciones de error.

2 Marco conceptual o referencial

Una transacción es una interacción con una estructura de datos compleja, compuesta por varios procesos que se han de aplicar uno después del otro. La transacción debe realizarse de una sola vez y sin que la estructura a medio manipular pueda ser alcanzada por el resto del sistema hasta que se hayan finalizado todos sus procesos.

El objetivo principal de una transacción es asegurarse de que todas las operaciones se completen con éxito o, en caso de que ocurra un error, se deshagan por completo, de modo que la base de datos se mantenga en un estado coherente. Las transacciones permiten a los usuarios y aplicaciones trabajar con los datos de manera segura, evitando situaciones donde los datos quedan en un estado intermedio o inconsistente debido a fallos o errores.

Las transacciones anidadas son una extensión del concepto de transacciones. En una transacción anidada, una transacción principal puede contener sub transacciones, y estas sub transacciones pueden contener a su vez otras sub transacciones. Esto permite una mayor granularidad y flexibilidad en el manejo de operaciones complejas que involucran múltiples pasos. Sin embargo, también introduce desafíos adicionales en cuanto a la gestión de la consistencia y la recuperación en caso de errores.

Una transacción en un Sistema de Gestión de Bases de Datos (SGBD), es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

Un SGBD se dice transaccional, si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

Las transacciones deben cumplir cuatro propiedades denominada ACID por sus iniciales en inglés:

- **Atomicidad**(Atomicity): es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia** (Consistency): es la propiedad que asegura que sólo se empiece aquello que se puede acabar. Por lo tanto, se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento**(Isolation): es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información nunca generará ningún tipo de error.
- **Durabilidad**(Durability): es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

El lenguaje de consulta de datos SQL (*Structured Query Language*), provee los mecanismos para especificar que un conjunto de acciones constituyen una transacción.

- **BEGIN TRANSACTION:** Especifica que va a empezar una transacción, representa un punto en el que los datos a los que hace referencia una conexión son lógicos y físicamente coherentes. Si se producen errores, se pueden revertir todas las modificaciones realizadas en los datos después de **BEGIN TRANSACTION** para devolver los datos al estado conocido de coherencia.
- **COMMIT TRANSACTION:** Le indica al motor que puede considerar la transacción completada con éxito y hace que las modificaciones de datos desde el inicio de la transacción sean una parte permanente de las bases de datos.
- **ROLLBACK TRANSACTION:** Revierte una transacción explícita o implícita hasta el inicio de la transacción o hasta un punto de retorno dentro de la transacción. Puede usar **ROLLBACK TRANSACTION** para borrar todas las modificaciones de datos realizadas desde el inicio de la transacción o hasta un punto de retorno. También libera los recursos que mantiene la transacción. Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

3 Metodología seguida

Los integrantes del grupo nos reunimos virtualmente a través de Discord, aprovechando las características de la plataforma para debatir sobre el tema y llevar a cabo el ejercicio práctico. Utilizamos la plataforma para compartir documentos y recursos.

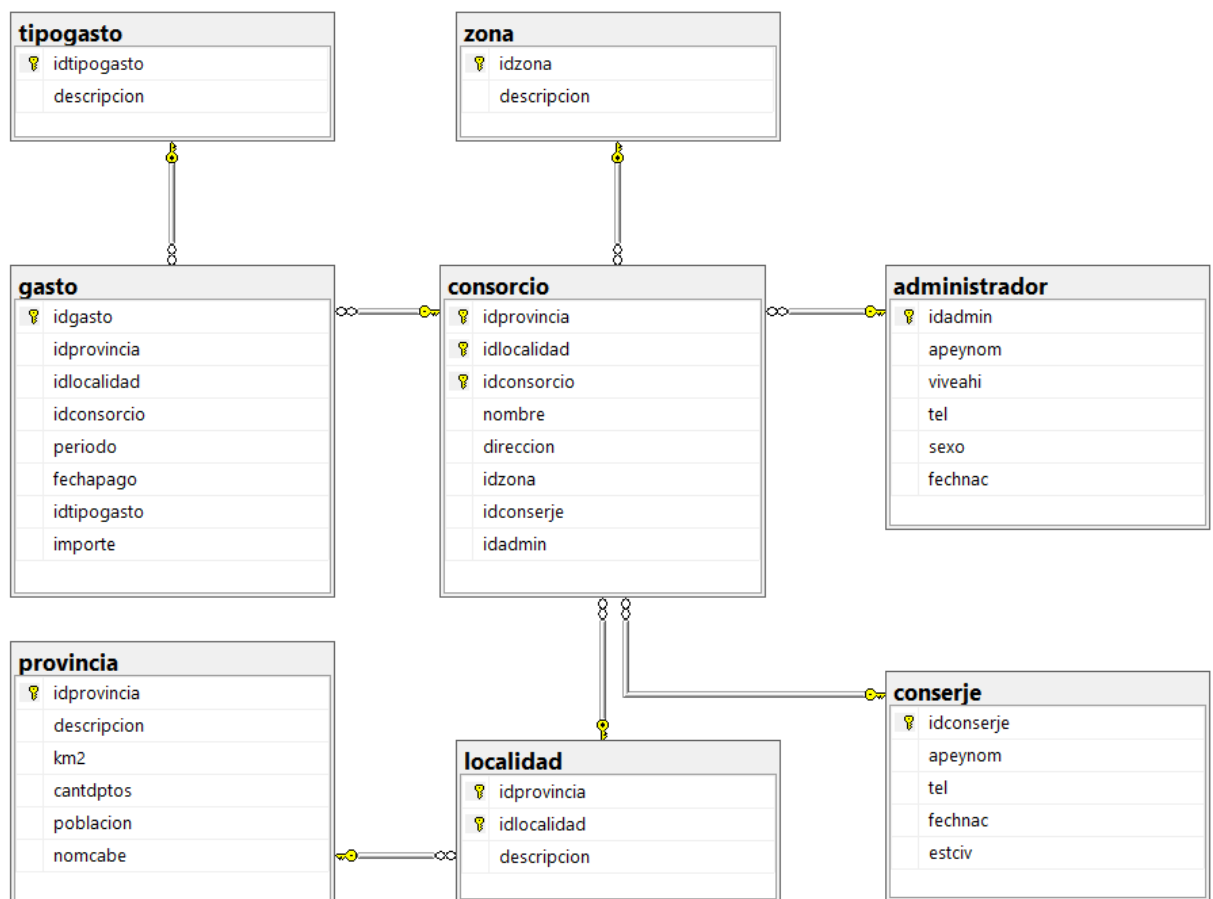
Se elaboró este documento del proyecto en Google Docs en el cual colaboramos en línea.

Para el ejercicio práctico utilizamos el motor de bases de datos SQL server 2022 y la gestión de la base de datos se trabajó en Microsoft SQL Server Management Studio.

4 Desarrollo del tema

Trabajaremos en un ejercicio práctico de transacciones sobre la siguiente base de datos:

Modelo de datos



Diccionario de datos

Características de la tabla			
Nombre	Provincia		
Descripción	Esta tabla almacena información de provincias		
Características de los datos			
campo	tipo	long	significado
idprovincia	int	4	Identificación única para una provincia
descripcion	varchar	50	Nombre de la provincia
km2	int	4	Superficie de la provincia
cantdptos	int	4	Cantidad de departamentos de la provincia
poblacion	int	4	Cantidad de población de la provincia
nomcabe	varchar	50	Nombre de la capital de la provincia
Restricciones			
campo		tipo de restricción	
idprovincia		Clave primaria	

Características de la tabla			
Nombre	localidad		
Descripción	Esta tabla almacena información de las localidades de provincias		
Características de los datos			
campo	tipo	long	significado
idprovincia	int	4	identificación única de la provincia a la que pertenece esta localidad
idlocalidad	int	4	identificación única para una localidad
descripcion	varchar	50	nombre de la localidad
Restricciones			
campo		tipo de restricción	
idlocalidad		clave primaria	
Claves foráneas			
campo		entidad asociada	
idprovincia		provincia	

Características de la tabla			
Nombre	conserje		
Descripción	Esta tabla almacena información de conserjes de consorcios		
Características de los datos			
campo	tipo	long	significado
idconserje	int	4	Identificación única para un conserje
apeynom	varchar	50	Apellido y nombre del conserje
tel	varchar	20	Numero de telefono del conserje
fechnac	datetime	8	Fecha de nacimiento del conserje
estciv	varchar	1	Estado civil del conserje
Restricciones			
campo		tipo de restricción	
idconserje		clave primaria	

Características de la tabla			
Nombre	administrador		
Descripción	Esta tabla almacena información de administradores de consorcios		
Características de los datos			
campo	tipo	long	significado
idadmin	int	4	identificación única para un administrador
apecynom	varchar	50	apellido y nombre de un administrador
veveahi	varchar	1	indica si el administrador vive en el consorcio
tel	varchar	20	teléfono del administrador
sexo	varchar	1	indica el sexo del administrador
fechnac	datetime	8	fecha de nacimiento del administrador
Restricciones			
campo		tipo de restricción	
idadmin		clave primaria	

Características de la tabla			
Nombre	zona		
Descripción	Esta tabla almacena información de la zona en donde se encuentra el consorcio		
Características de los datos			
campo	tipo	long	significado
idzona	int	4	identificación única para una zona
descripcion	varchar	50	nombre de la zona
Restricciones			
campo		tipo de restricción	
idzona		clave primaria	

Características de la tabla			
Nombre	tipogasto		
Descripción	Esta tabla almacena información sobre un tipo de gasto		
Características de los datos			
campo	tipo	long	significado
idtipogasto	int	4	identificación única para un tipo de gasto
descripcion	varchar	50	nombre del tipo de gasto
Restricciones			
campo		tipo de restricción	
idtipogasto		clave primaria	

Características de la tabla			
Nombre	gasto		
Descripción	Esta tabla almacena información de un gasto		
Características de los datos			
campo	tipo	long	significado
idgasto	int	4	identificación única para un gasto
idprovincia	int	4	identificación única de la provincia donde se registra el gasto
idlocalidad	int	4	identificación única de la localidad donde se registra el gasto
idconsorcio	int	4	identificación única del consorcio donde se registra el gasto
periodo	int	4	periodo en el que se registra el gasto
idtipogasto	int	4	identificación única del tipo de gasto
fechapago	datetime	8	fecha en la que se registra el gasto
importe	decimal	5	importe por el cual se realiza el gasto
Restricciones			
campo		tipo de restricción	
idgasto		clave primaria	
Claves foráneas			
campo		entidad asociada	
idprovincia		provincia	

Bases de Datos I
Transacciones y Transacciones anidadas

idlocalidad	localidad
idconsorcio	consorcio
idtipogasto	tipogasto

Características de la tabla			
Nombre	consorcio		
Descripción	Esta tabla almacena información de un consorcio		
Características de los datos			
campo	tipo	long	significado
idconsorcio	int	4	identificación única para un consorcio
idlocalidad	int	4	identificación de la localidad en la cual se encuentra el consorcio
idprovincia	int	4	identificación de la provincia en la cual se encuentra el consorcio
nombre	varchar	50	nombre del consorcio
direccion	varchar	250	dirección del consorcio
idzona	int	4	identificación única de la zona en la cual se encuentra el consorcio
idconserje	int	4	identificación única del conserje del consorcio
idadmin	int	4	identificación única del administrador del consorcio
Restricciones			

Bases de Datos I
Transacciones y Transacciones anidadas

campo	tipo de restricción
idprovincia	clave primaria compuesta
idlocalidad	clave primaria compuesta
idconsorcio	clave primaria compuesta
Claves foráneas	
campo	entidad asociada
idzona	zona
idconserje	conserje
idadmin	admin
idprovincia	provincia
idlocalidad	localidad

Restricciones

Entidad	Tipo de restricción	Nombre de la restricción	Columna
localidad	PRIMARY KEY(compuesta)	PK_localidad	idprovincia, idlocalidad
localidad	FOREIGN KEY	FK_localidad_pcia	idprovincia
provincia	PRIMARY KEY	PK__provinci__1AF96EE8B68C70C1	idprovincia
conserje	CHECK	CK_estadocivil	estciv
conserje	PRIMARY KEY	PK__conserje__095979F6C851AAB7	idconserje
administrador	PRIMARY KEY	PK__administ__5B93BD6EBC8961F6	idadministrador
administrador	CHECK	CK_habitante_viveahi	viveahi

Bases de Datos I
Transacciones y Transacciones anidadas

administrador	CHECK	CK_sexo	sexo
zona	PRIMARY KEY	PK_zona_D75A19E2CDF8BE50	idzona
tipogasto	PRIMARY KEY	PK_tipogast_8CBCB4216161C3DD	idtipogasto
consorcio	PRIMARY KEY(compuesta)	PK_consorcio	idprovincia, idlocalidad,idconsorcio
consorcio	FOREIGN KEY	FK_consorcio_pcia	idprovincia,idlocalidad
consorcio	FOREIGN KEY	FK_consorcio_zona	idzona
consorcio	FOREIGN KEY	FK_consorcio_conserje	idconserje
consorcio	FOREIGN KEY	FK_consorcio_admin	idadmin
gasto	PRIMARY KEY	PK_gasto	idgasto
gasto	FOREIGN KEY	FK_gasto_consorcio	idprovincia,idlocalidad,idconsorcio
gasto	FOREIGN KEY	FK_gasto_tipo	idtipogasto

Transacción

- El siguiente script inserta un registro en administrador, luego otro registro en consorcio y por último 3 registros en gasto, correspondiente a ese nuevo consorcio. Los datos se actualizan solamente si toda la operación es completada con éxito.

```
BEGIN TRANSACTION TranExterna;
```

```
-- Variables para almacenar IDs
```

```
DECLARE @AdminID INT, @ConsortioID INT;
```

```
-- Inserción en la tabla "administrador"
```

```
INSERT INTO administrador (apeynom, viveahi, tel, sexo, fechnac)
```

```
VALUES ('JAVIER FRIAS', 'N', '370816973', 'M', 24-02-1997);
```

```
-- Obtener el ID del administrador recién insertado
```

```
SET @AdminID = SCOPE_IDENTITY();
```

```
-- Inserción en la tabla "consorcio" con referencia al administrador
```

```
INSERT INTO consorcio (idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin)
```

```
VALUES (1, 1, 5, 'Edificio-9999', 'Remedios Escalada 5407', 1, 1, @AdminID);
```

```
-- Obtener el ID del consorcio recién insertado
```

```
SET @ConsortioID = 5;
```

```
-- Inserción de 3 registros en la tabla "gasto" correspondientes al nuevo consorcio
```

```
INSERT INTO gasto (idprovincia, idlocalidad, idconsorcio, periodo, fechapago, idtipogasto, importe)
```

```
VALUES
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 1, 100.00),
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 2, 200.00),
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 3, 300.00);
```


- Consultamos las tablas para ver cómo fueron insertados los datos:

```
SELECT * FROM gasto WHERE periodo=2023;
go
SELECT * FROM administrador WHERE apeynom= 'JAVIER FRIAS';
go
SELECT * FROM consorcio WHERE idconsorcio = 5;
```

110 %

Results Messages

	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8004	1	1	5	2023	2023-10-31 21:59:42.243	1	100.00
2	8005	1	1	5	2023	2023-10-31 21:59:42.243	2	200.00
3	8006	1	1	5	2023	2023-10-31 21:59:42.243	3	300.00

	idadmin	apeynom	viveahi	tel	sexo	fechnac
1	176	JAVIER FRIAS	N	370816973	M	1997-02-24 00:00:00.000

	idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin
1	1	1	5	Edificio-9999	Remedios Escalada 5407	1	1	176
2	3	29	5	EDIFICIO-3295	MZ1 PC6 C26 B° SANTA BARBARA	5	92	9
3	5	14	5	EDIFICIO-5145	ESPA&AMP;AMP;NTILDE;A...	5	84	17

Query executed successfully.

- Si todo se ha completado con éxito, confirmamos la transacción externa, se harán las modificaciones de datos de manera permanente en la base de datos:

COMMIT TRANSACTION TranExterna;

Si alguna operación sale mal en algún punto, revertimos la transacción:

ROLLBACK TRANSACTION TranExterna;

- Ahora modificaremos el script anterior y provocaremos intencionalmente un error luego del insert en consorcio y verificaremos que los datos queden consistentes:

BEGIN TRANSACTION TranExterna;

-- Variables para almacenar IDs

DECLARE @AdminID INT, @ConsortioID INT;

-- Inserción en la tabla "administrador"

INSERT INTO administrador (apeynom, viveahi, tel, sexo, fechnac)

VALUES ('JAVIER FRIAS', 'N', '370816973', 'M', '1997-02-24');

-- Obtener el ID del administrador recién insertado

SET @AdminID = SCOPE_IDENTITY();

-- Inserción en la tabla "consorcio" con referencia al administrador

INSERT INTO consorcio (idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin)

VALUES (1, 1, 3, 'Edificio-9999', 'Remedios Escalada 5407', 1, 1, @AdminID);

-- Simulamos un error intencional

```
-- Cambiamos el valor de idconsorcio a un valor que no existe en la tabla "consorcio"  
SET @ConsortioID = 999; -- Valor que no existe en la tabla "consorcio"
```

```
-- Intento de inserción de 3 registros en la tabla "gasto" correspondientes al nuevo  
consorcio
```

```
-- Esto debe fallar debido al valor inexistente de @ConsortioID
```

```
INSERT INTO gasto (idprovincia, idlocalidad, idconsorcio, periodo, fechapago,  
idtipogasto, importe)
```

```
VALUES
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 1, 100.00),
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 2, 200.00),
```

```
(1, 1, @ConsortioID, 2023, GETDATE(), 3, 300.00);
```

```
-- Si algo sale mal en cualquier punto, revertimos la transacción
```

```
IF @@ERROR <> 0
```

```
BEGIN
```

```
    ROLLBACK TRAN TranExterna;
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
    COMMIT TRAN TranExterna;
```

```
END;
```

- Al ejecutar la consulta, El SGBD nos da el siguiente mensaje:

```
The INSERT statement conflicted with the FOREIGN KEY constraint  
"FK_gasto_consorcio". The conflict occurred in database "base_consorcio", table  
"dbo.consorcio".
```

La instrucción INSERT entró en conflicto con la restricción FOREIGN KEY "FK_gasto_consorcio". El conflicto ocurrió en la base de datos "base_consorcio", tabla "consorcio".

- La evaluación condicional del error se realiza de la siguiente manera:

Si el valor de @@ERROR es diferente de 0, indica que se generó un error durante la ejecución de la última instrucción SQL. En este caso, se ejecuta la instrucción ROLLBACK TRAN TranExterna;, lo que revierte todos los cambios realizados en la transacción TranExterna. Esto garantiza que la base de datos vuelva a su estado original antes de que se produzca el error.

Si el valor de @@ERROR es igual a 0, significa que la última instrucción SQL se ejecutó sin errores. En este caso, se ejecuta la instrucción COMMIT TRAN TranExterna;, lo que confirma la transacción TranExterna. Esto indica que la operación se completó con éxito y se confirman los cambios en la base de datos.

5 Conclusiones

Durante el transcurso de este trabajo práctico investigamos y profundizamos el entendimiento del manejo de transacciones y transacciones anidadas, siendo éste un concepto fundamental en SQL en el contexto de base de datos. Esto se llevó a cabo gracias al aprovechamiento de diferentes fuentes de información o bibliografías que, en combinación con una aplicación práctica se logró sin dificultades el objetivo de comprender la integridad y la gestión de transacciones.

En el desarrollo del trabajo, hemos logrado los siguientes resultados y conclusiones:

1. **Transacciones:** Se ha demostrado cómo usar transacciones para garantizar la atomicidad y consistencia en operaciones de inserción y actualización de datos. Hemos aplicado una transacción que involucra múltiples inserciones de registros, aplicarlos permanentemente con COMMIT TRANSACTION, y adicionalmente a esto, vimos cómo se pueden revertir las transacciones en caso de un error, usando ROLLBACK TRANSACTION, lo que impide cambios parciales en la base de datos.
2. **Simulación de errores:** Se demostró la importancia de las transacciones al simular un error intencional en una inserción de datos, esto nos permitió comprobar que las transacciones funcionan de manera eficaz para mantener la consistencia de la base de datos en caso de fallos.

Se podría decir con toda seguridad que gracias a este trabajo investigativo adquirimos los conocimientos necesarios para gestionar transacciones de manera efectiva, conocimientos que son esenciales para facilitar la toma de decisiones basadas en datos en diversas aplicaciones y entornos.

6 Bibliografía

<https://topicdb.wordpress.com/1-5-manejo-de-transacciones/>

https://programacion.net/articulo/transacciones_en_sql_server_299

<https://learn.microsoft.com/es-es/sql/t-sql/language-elements/transactions-transact-sql?view=sql-server-ver16>

<https://learn.microsoft.com/es-es/office/client-developer/access/desktop-database-reference/transaction-processing#nested-transactions>