UNNE
Universidad Nacional del Nordeste

FaCENA - UNNE

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

Bases de datos I

PROYECTO DE ESTUDIO E INVESTIGACIÓN

Tema: Manejo de transacciones y transacciones anidadas

Profesores:

- Cuzziol, Juan José
- Sambrana, Ivan

Grupo n° 4:

Integrantes:

- Duete, Juan Pablo
- Bartolucci, Gastón

UNNE
Universidad Nacional
del Nordeste

FaCENA - UNNE

Contenido

CAPÍTULO I: INTRODUCCIÓN

El presente trabajo de investigación tiene como objetivo entender el funcionamiento de las transacciones en SQL Server, así como los distintos tipos de opciones que tenemos en su aplicación. La elección del tipo de transacción que escogemos para realizar depende concretamente del dominio del problema a resolver.

Para empezar a trabajar con transacciones debemos entender que una transacción es una unidad única de trabajo. Si una transacción tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y se convierten en una parte permanente de la base de datos. Si una transacción encuentra errores y debe cancelarse o revertirse, se borran todas las modificaciones de los datos.

Cuando enviamos una sentencia al SQL Server se escribe en el fichero de transacciones lo que va a ocurrir y a continuación realiza los cambios necesarios en la base de datos. Si hay algún tipo de problema al hacer esta operación el SQL Server puede leer en el fichero de transacciones lo que se estaba haciendo y si es necesario puede devolver la base de datos al estado en el que se encontraba antes de recibir la sentencia.

Por supuesto este tipo de transacciones no requieren de nuestra intervención puesto que el sistema se encarga de todo. Sin embargo, si hay que realizar varias operaciones y queremos que sean tratadas como una unidad tenemos que crear esas transacciones de manera explícita.

En un conjunto de operaciones marcadas como transacción, todas las operaciones que la conforman tienen éxito o todas fracasan.

En ciertas ocasiones también necesitamos llamar a un procedimiento dentro de una transacción, y que éste a su vez tenga otra transacción. Con lo cual tendríamos una transacción dentro de otra. Esto es lo que se conoce como transacciones anidadas.

CAPITULO II: MARCO CONCEPTUAL O REFERENCIAL

Transacciones

Una transacción es un conjunto de operaciones que van a ser tratadas como una única unidad.

Estas transacciones deben cumplir 4 propiedades fundamentales comúnmente conocidas como ACID (atomicidad, consistencia, asilamiento y durabilidad):

1. **Atomicidad:** Es la propiedad de las transacciones que permite observarlas como operaciones atómicas: ocurren totalmente o no ocurren.
2. **Consistencia:** La ejecución aislada de la transacción conserva la consistencia de la base de datos.
3. **Aislamiento:** Las transacciones son independientes entre sí.
4. **Durabilidad:** El sistema gestor de bases de datos asegura que perduren los cambios realizados por una transacción que termina con éxito.

Una transacción puede estar en distintos estados:

• Activa: Durante su ejecución.

• Parcialmente comprometida: Después de ejecutar su última instrucción.

• Fallida: Imposible de continuar su ejecución normal.

• Abortada: Transacción retrocedida y base de datos restaurada al estado anterior a su ejecución. Se puede reiniciar o cancelar.

La transacción más simple en SQL Server es una única sentencia SQL.

La sentencia que se utiliza para indicar el comienzo de una transacción es 'BEGIN TRAN'. Si alguna de las operaciones de una transacción falla hay que deshacer la transacción en su totalidad para volver al estado inicial en el que estaba la base de datos antes de empezar. Esto se consigue con la sentencia 'ROLLBACK TRAN'.

Si todas las operaciones de una transacción se completan con éxito hay que marcar el fin de una transacción para que la base de datos vuelva a estar en un estado consistente con la sentencia 'COMMIT TRAN'.

Transacciones anidadas

Otra de las posibilidades que nos ofrece el SQL Server es utilizar transacciones anidadas. Esto quiere decir que podemos tener transacciones dentro de transacciones, es decir, podemos empezar una nueva transacción sin haber terminado la anterior.

Asociada a esta idea de anidamiento existe una variable global @@TRANCOUNT que tiene valor 0 si no existe ningún nivel de anidamiento, 1 si hay una transacción anidada, 2 si estamos en el segundo nivel de anidamiento. y así sucesivamente.

La dificultad de trabajar con transacciones anidadas está en el comportamiento que tienen ahora las sentencias 'COMMIT TRAN' y 'ROLLBACK TRAN'

- ROLLBACK TRAN: Dentro de una transacción anidada esta sentencia deshace todas las transacciones internas hasta la instrucción BEGIN TRANSACTION más externa.
- COMMIT TRAN: Dentro de una transacción anidada esta sentencia únicamente reduce en 1 el valor de @@TRANCOUNT, pero no "finaliza" ninguna transacción ni "guarda" los cambios. En el caso en el que @@TRANCOUNT=1 (cuando estamos en la última transacción) COMMIT TRAN hace que todas las modificaciones efectuadas sobre los datos desde el inicio de la transacción sean parte permanente de la base de datos, libera los recursos mantenidos por la conexión y reduce @@TRANCOUNT a 0.

CAPÍTULO III: METODOLOGÍA SEGUIDA

Para la realización del presente trabajo se optó por trabajar con SQL Server a través de la aplicación SQL Management Studio, en su versión *Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64)*.

Para demostrar el funcionamiento del uso de las transacciones y transacciones anidadas decidimos introducir valores erróneos, de manera intencional, en las inserciones de datos a las tablas.

Entre las distintas opciones elegimos hacer referencia a través de una foreign key (desde la tabla gasto), a un id que no se encuentra entre los datos de la tabla provincia. Esto genera un error, puesto que el valor que se intenta referenciar es inexistente.

Normalmente la ejecución de las sentencias anteriores a ésta, deberían realizarse correctamente y realizar las inserciones correspondientes. Pero al estar todas las sentencias dentro de una misma transacción, ninguna se ejecuta, ya que todas las operaciones dentro de las transacciones funcionan como una única unidad.

Verificamos que las transacciones se ejecuten correctamente y no se realicen las modificaciones de las sentencias anteriores a la manifestación del error.

Para el manejo de transacciones anidadas, también se optó por ingresar valores erróneos en las inserciones de datos de determinadas tablas.

Las anidaciones fueron realizadas en distintos niveles de anidamiento. Para demostrar el anidamiento de transacciones se decidió añadir un SELECT al principio de cada transacción que devuelva el numero de transacciones pendientes a través de la sentencia “@@trancount”.

Nuevamente, al finalizar la ejecución de las transacciones correspondientes, se verificó que los datos no fueron modificados cuando ocurrió un error dentro de la transacción.

CAPÍTULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DE RESULTADOS

A continuación, se presentan los scripts utilizados para realizar las pruebas con sus correspondientes resultados:

Prueba de transacción con error en la ejecución: Script

```
SET LANGUAGE Spanish;

BEGIN TRY
	BEGIN TRAN

	--Inserción registro administrador
		INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac)
	values ('Juan Pablo Duete', 'S', '3794000102', 'M', '19890625')

	--Inserción registro consorcio
		INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,
	Nombre,direccion,idzona,idconserje,idadmin)
		VALUES (7, 7, 1, 'EDIFICIO-24500', '9 de julio Nº 1650', 2,
	Null, (select top 1 idadmin from administrador order by idadmin desc))
	--id del último administrador

	--Inserción 3 registros gasto
		INSERT INTO gasto
		(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipoga
		sto,importe)
		VALUES (30, 7, 1,1,'20231005',2,5000.00)

		INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
		VALUES (7, 7, 1,1,'20231015',2,20000.00)

		INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
		VALUES (7, 7, 1,1,'20231028',2,10000.00)

	COMMIT TRAN
END TRY

BEGIN CATCH
	SELECT ERROR_MESSAGE() As Error-- Se captura el error y lo muestra.
	ROLLBACK TRAN
END CATCH
```

UNNE
Universidad Nacional
del Nordeste

FaCENA - UNNE

Prueba de transacción con error en la ejecución: Resultados

```
    VALUES (7, 7, 1,1,'20231015',2,20000.00)

    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231028',2,10000.00)

    COMMIT TRAN
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() As Error-- Se captura el error y lo muestra.
    ROLLBACK TRAN
END CATCH

select * from administrador where apeynom = 'Juan Pablo Duete'
select * from consorcio where idprovincia = 7 --No existen en principio consorcios en la provincia de Corrientes
select * from gasto where idprovincia = 7
```

Results | Messages

Error

Instrucción INSERT en conflicto con la restricción FOREIGN KEY 'FK_gasto_consorcio'. El conflicto ha aparecido en la base de datos 'base_consorcio', tabla 'dbo.consorcio'.

idadmin	apeynom	viveahi	tel	sexo	fechnac

idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin

idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe

Se observa que se muestra el error ocurrido y que los SELECT utilizados para el control no devuelven un valor por lo tanto no se ejecutaron la totalidad de las instrucciones dentro de la transacción, más allá de que el error se presente luego de instrucciones que son válidas.

UNNE
Universidad Nacional
del Nordeste

FaCENA - UNNE

Prueba de transacción sin error en la ejecución: Script

```
SET LANGUAGE Spanish;

BEGIN TRY
	BEGIN TRAN

	--Inserción registro administrador
		INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac)
	values ('Juan Pablo Duete', 'S', '3794000102', 'M', '19890625')

	--Inserción registro consorcio
		INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,
	Nombre,direccion,idzona,idconserje,idadmin)
		VALUES (7, 7, 1, 'EDIFICIO-24500', '9 de julio Nº 1650', 2,
	Null, (select top 1 idadmin from administrador order by idadmin desc))
	--id del último administrador

	--Inserción 3 registros gasto
		INSERT INTO gasto
		(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipoga
		sto,importe)
		VALUES (7, 7, 1,1,'20231005',2,5000.00)

		INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
		VALUES (7, 7, 1,1,'20231015',2,20000.00)

		INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
		VALUES (7, 7, 1,1,'20231028',2,10000.00)

	COMMIT TRAN
END TRY

BEGIN CATCH
	SELECT ERROR_MESSAGE() As Error-- Se captura el error y lo muestra.
	ROLLBACK TRAN
END CATCH
```

Prueba de transacción con error en la ejecución: Resultados

```
    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231015',2,20000.00)

    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231028',2,10000.00)

    COMMIT TRAN
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() As Error-- Se captura el error y lo muestra.
    ROLLBACK TRAN
END CATCH

select * from administrador where apeynom = 'Juan Pablo Duete'
select * from consorcio where idprovincia = 7 --No existen en principio consorcios en la provincia de Corrientes
select * from gasto where idprovincia = 7
```

100 %

Results | Messages

	idadmin	apeynom	viveahi	tel	sexo	fechnac
1	178	Juan Pablo Duete	S	3794000102	M	1989-06-25 00:00:00.000

	idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin
1	7	7	1	EDIFICIO-24500	9 de julio Nº 1650	2	NULL	178

	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8004	7	7	1	1	2023-10-05 00:00:00.000	2	5000.00
2	8005	7	7	1	1	2023-10-15 00:00:00.000	2	20000.00
3	8006	7	7	1	1	2023-10-28 00:00:00.000	2	10000.00

Se observa que se muestra ningún mensaje de error y que los SELECT utilizados para el control devuelven los valores insertados, por lo tanto, se ejecutaron la totalidad de las instrucciones dentro de la transacción.

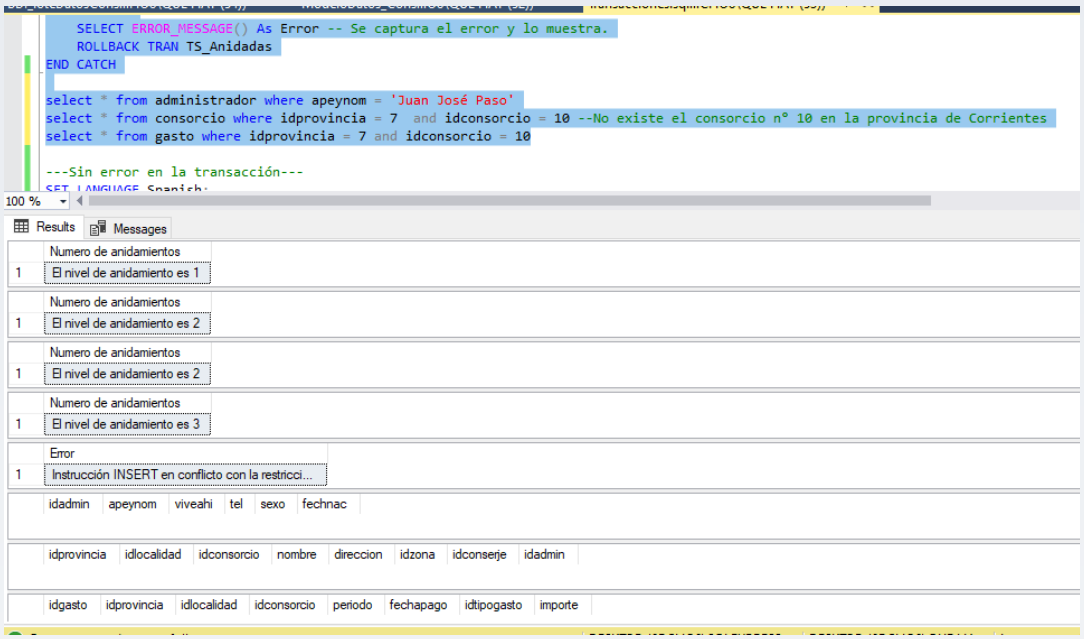
Prueba de transacciones anidadas con error en la ejecución: Script

```
SET LANGUAGE Spanish;

BEGIN TRY
	BEGIN TRAN TS_Anidadas
	SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de
anidamientos' --Esta sentencia cuenta el número de transacciones anidadas
		BEGIN TRAN TS_InsertarAdmin
			--Inserción registro administrador
			SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As
'Numero de anidamientos'
			INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac)
values ('Juan José Paso', 'S', '3794000102', 'M', '19890625')
		COMMIT TRAN TS_InsertarAdmin
		BEGIN TRAN TS_InsertarConsorcio
			--Inserción registro consorcio
			SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As
'Numero de anidamientos'
			INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,
Nombre,direccion,idzona,idconserje,idadmin)
			VALUES (7, 7, 2, 'EDIFICIO-24500', '9 de julio Nº 1650', 2,
Null, (select top 1 idadmin from administrador order by idadmin desc))
			--Para insertar el id del último administrador se hace un
select del ultimo id de administrador insertado.

			BEGIN TRAN TS_InsertarGastos
				SELECT CONCAT('El nivel de anidamiento es ',
@@TRANCOUNT) As 'Numero de anidamientos'
				--Inserción 3 registros de gastos
				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 2,1,'20231005',2,5000.00)

				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 2,1,'20231015',2,20000.00)

				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 10,1,'20231028',2,10000.00)
			COMMIT TRAN TS_InsertarGastos
		COMMIT TRAN TS_InsertarConsorcio


	COMMIT TRAN TS_Anidadas
END TRY

BEGIN CATCH
	SELECT ERROR_MESSAGE() As Error -- Se captura el error y lo muestra.
	ROLLBACK TRAN TS_Anidadas
END CATCH
```

UNNE
Universidad Nacional
del Nordeste

FaCENA - UNNE

Prueba de transacciones anidadas con error en la ejecución: Resultados



Se observa que se muestra el nivel de anidamiento de cada transacción a través del uso de la sentencia “@@TRANCOUNT”, así como también el error ocurrido y que los SELECT utilizados para el control no devuelven un valor por lo tanto no se ejecutaron la totalidad de las instrucciones dentro de la transacción, más allá de que el error se presente luego de instrucciones que son válidas.

Prueba de transacciones anidadas sin error en la ejecución: Script

```
SET LANGUAGE Spanish;

BEGIN TRY
	BEGIN TRAN TS_Anidadas
	SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de
anidamientos' --Esta sentencia cuenta el número de transacciones anidadas
		BEGIN TRAN TS_InsertarAdmin
			--Inserción registro administrador
			SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As
'Numero de anidamientos'
			INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac)
values ('Juan José Paso', 'S', '3794000102', 'M', '19890625')
		COMMIT TRAN TS_InsertarAdmin
		BEGIN TRAN TS_InsertarConsorcio
			--Inserción registro consorcio
			SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As
'Numero de anidamientos'
			INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,
Nombre,direccion,idzona,idconserje,idadmin)
			VALUES (7, 7, 2, 'EDIFICIO-24500', '9 de julio Nº 1650', 2,
Null, (select top 1 idadmin from administrador order by idadmin desc))
			--Para insertar el id del último administrador se hace un
select del ultimo id de administrador insertado.

			BEGIN TRAN TS_InsertarGastos
				SELECT CONCAT('El nivel de anidamiento es ',
@@TRANCOUNT) As 'Numero de anidamientos'
				--Inserción 3 registros de gastos
				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 2,1,'20231005',2,5000.00)

				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 2,1,'20231015',2,20000.00)

				INSERT INTO gasto
(idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
				VALUES (7, 7, 2,1,'20231028',2,10000.00)
			COMMIT TRAN TS_InsertarGastos
		COMMIT TRAN TS_InsertarConsorcio


	COMMIT TRAN TS_Anidadas
END TRY

BEGIN CATCH
	SELECT ERROR_MESSAGE() As Error -- Se captura el error y lo muestra.
	ROLLBACK TRAN TS_Anidadas
END CATCH
```

UNNE
Universidad Nacional del Nordeste

FaCENA - UNNE

Prueba de transacción sin error en la ejecución: Resultados

```
    SELECT ERROR_MESSAGE() As Error -- Se captura el error y lo muestra.
    ROLLBACK TRAN TS_Anidadas
END CATCH

select * from administrador where apeynom = 'Juan José Paso'
select * from consorcio where idprovincia = 7  and idconsorcio = 2 --No existe el consorcio n° 10 en la provincia de Corrientes
select * from gasto where idprovincia = 7 and idconsorcio = 2

---Sin error en la transacción---
```

Results | Messages

	Numero de anidamientos
1	El nivel de anidamiento es 1

	Numero de anidamientos
1	El nivel de anidamiento es 2

	Numero de anidamientos
1	El nivel de anidamiento es 2

	Numero de anidamientos
1	El nivel de anidamiento es 3

	idadmin	apeynom	viveahi	tel	sexo	fechnac
1	184	Juan José Paso	S	3794000102	M	1989-06-25 00:00:00.000

	idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin
1	7	7	2	EDIFICIO-24500	9 de julio Nº 1650	2	NULL	184

	idgasto	idprovincia	idlocalidad	idconsorcio	periodo	fechapago	idtipogasto	importe
1	8018	7	7	2	1	2023-10-05 00:00:00.000	2	5000.00
2	8019	7	7	2	1	2023-10-15 00:00:00.000	2	20000.00
3	8020	7	7	2	1	2023-10-28 00:00:00.000	2	10000.00

Se observa, además del nivel de anidamiento de cada transacción a través del uso de la sentencia “@@TRANCOUNT”, que no se muestra ningún mensaje de error y que los SELECT utilizados para el control devuelven los valores insertados, por lo tanto, se ejecutaron la totalidad de las instrucciones dentro de la transacción.

CAPÍTULO V: CONCLUSIONES

El manejo de transacciones es una poderosa herramienta para el manejo de bases de datos. El hecho de poder agrupar sentencias y poder ejecutarlas como un todo, sin que parte del trabajo quede inconcluso, es una grandísima ventaja al momento de manejar grandes volúmenes de datos y de operaciones que necesariamente deben realizarse en conjunto.

Posee sentencias propias de las transacciones como por ejemplo “@@TRANCOUNT” que nos permite llevar un mejor control de numero de transacciones que estamos manejando.

El anidamiento de transacciones nos da la posibilidad de dividir las transacciones en distintos tipos de operaciones, para entender de mejor manera el trabajo que realiza el grupo de sentencias que estamos utilizando como una sola transacción.

Las transacciones (junto con otras herramientas como los índices, procedimientos almacenados, entre otros), nos permiten tener un mayor y mas eficiente manejo de los datos con los que trabajamos.

UNNE
Universidad Nacional del Nordeste

FaCENA - UNNE

BIBLIOGRAFÍA

rwestMSFT. (s/f). Transacciones (Transact-SQL). Microsoft.com. Recuperado el 1 de noviembre de 2023, de https://learn.microsoft.com/es-es/sql/t-sql/language-elements/transactions-transact-sql?view=sql-server-ver15

Transacciones y control de la concurrencia. Ucm.es. Recuperado el 1 de noviembre de 2023, de https://www.fdi.ucm.es/profesor/fernan/DBD/apuntestema07.pdf

El-jamesaranda [@el-jamesaranda]. (2021, noviembre 14). Transactions en SQL Server - Rollback / Commit. Youtube. https://www.youtube.com/watch?v=xiTfest5ApU

Hdeleon.net [@hdeleonnet]. (2020, agosto 21). ✨ La magia de las transacciones SQL | Ejemplo en Sql Server. Youtube. https://www.youtube.com/watch?v=keL9-EtE-zE

Peña, A. (2022, abril 2). Transacciones SQL: Confirmar, Revertir, Guardar. Sqlserverdb. https://sqlserverdb.com/transacciones-sql/

Programacion en Castellano, S. L. (s/f). Transacciones en SQL Server. Recuperado el 1 de noviembre de 2023, de https://programacion.net/articulo/transacciones_en_sql_server_299

Sergio, P. P. (s/f). sViudes Blog. Blogspot.com. Recuperado el 1 de noviembre de 2023, de https://sviudes.blogspot.com/2009/12/transacciones-anidadas-nested.html