Manejo de transacciones y transacciones anidadas

Entender concepto y tipos de transacciones

**Tema: Manejo de transacciones y transacciones anidadas**

**Objetivos:**

* **Entender conceptualmente el significado.**
* **Entender los diferentes tipos de transacciones**

**Tareas:**

* **Crear un documento de acuerdo al modelo existente en el aula virtual.**
* **Escribir el código SQL que permita definir una transacción consistente en: Insertar un registro en Administrador, luego otro registro en consorcio y por último 3 registros en gasto, correspondiente a ese nuevo consorcio. Actualizar los datos solamente si toda la operación es completada con éxito.**
* **Sobre el código escrito anteriormente provocar intencionalmente un error luego del insert en consorcio y verificar que los datos queden consistentes (No se debería realizar ningún insert).**
* **Expresar las conclusiones en base a las pruebas realizadas.**

<https://sqlserverdb.com/transacciones-sql/>

Transacciones (Transact-SQL)

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.

SQL Server funciona en los modos de transacción siguientes:

Transacciones de confirmación automática

Cada instrucción individual es una transacción.

Transacciones explícitas

Cada transacción se inicia explícitamente con la instrucción BEGIN TRANSACTION y se termina explícitamente con una instrucción COMMIT o ROLLBACK.

Transacciones implícitas

Se inicia implícitamente una nueva transacción cuando se ha completado la anterior, pero cada transacción se completa explícitamente con una instrucción COMMIT o ROLLBACK.

Transacciones de ámbito de lote

Una transacción implícita o explícita de Transact-SQL que se inicia en una sesión de MARS (conjuntos de resultados activos múltiples), que solo es aplicable a MARS, se convierte en una transacción de ámbito de lote. Si no se confirma o revierte una transacción de ámbito de lote cuando se completa el lote, SQL Server la revierte automáticamente.

Niveles de aislamiento de transacción

SQL Server no garantiza que se respeten las sugerencias de bloqueo en consultas que tengan acceso a metadatos por medio de vistas de catálogo, vistas de compatibilidad, vistas del esquema de información y funciones integradas de emisión de metadatos.

Internamente, Motor de base de datos de SQL Server solo respeta el nivel de aislamiento READ COMMITTED para el acceso a metadatos. Si una transacción tiene un nivel de aislamiento que es, por ejemplo, SERIALIZABLE, y en la transacción se intenta obtener acceso a metadatos mediante vistas de catálogo o funciones integradas de emisión de metadatos, dichas consultas se ejecutan hasta que finalizan como READ COMMITTED. Sin embargo, en el aislamiento de instantánea, puede que el acceso a metadatos genere un error debido a operaciones DDL simultáneas. Esto se debe a que los metadatos no admiten versiones. Por tanto, puede que en el aislamiento de instantánea se genere un error al obtener acceso a:

Vistas de catálogo

Vistas de compatibilidad

Vistas de esquema de información

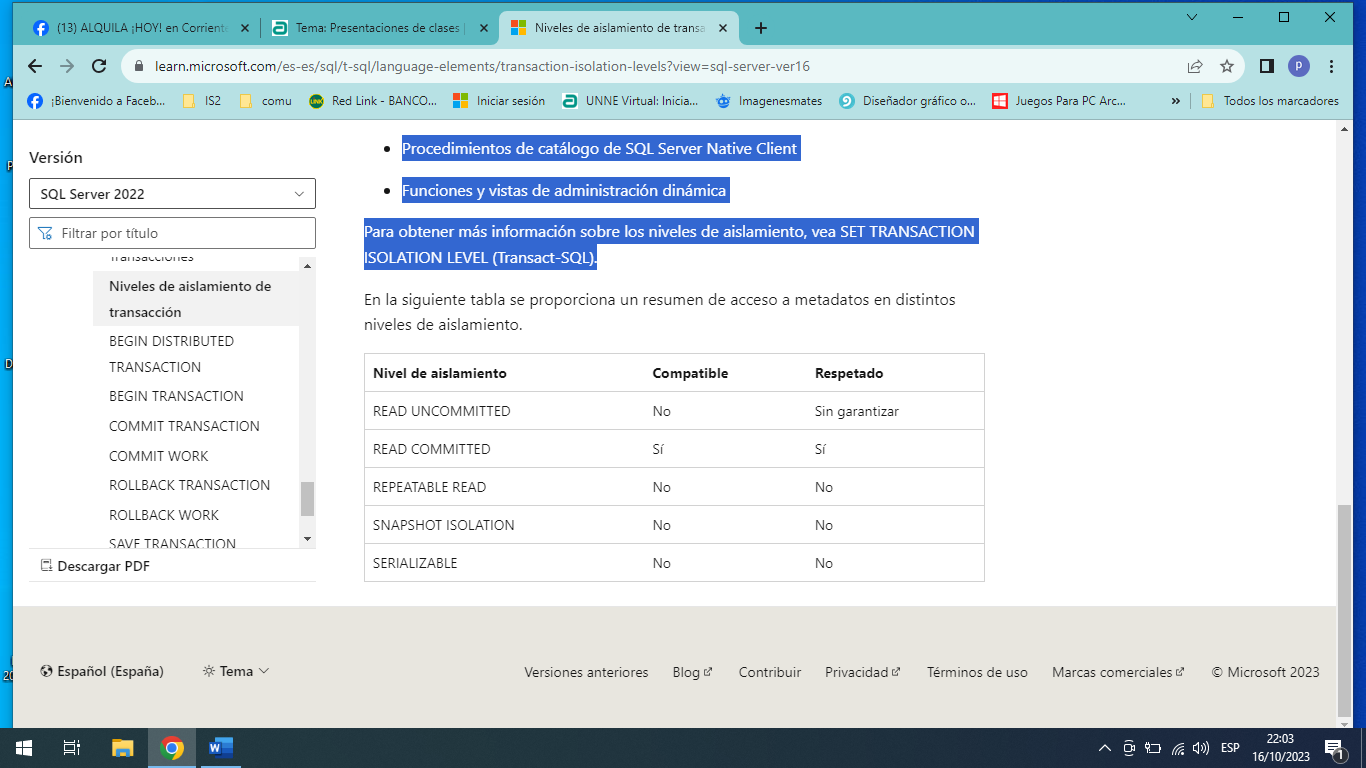
Funciones integradas de emisión de metadatos

Grupo de procedimientos almacenados sp\_help

Procedimientos de catálogo de SQL Server Native Client

Funciones y vistas de administración dinámica

Para obtener más información sobre los niveles de aislamiento, vea SET TRANSACTION ISOLATION LEVEL (Transact-SQL).



**Transacciones anidadas**

<https://sviudes.blogspot.com/2009/12/transacciones-anidadas-nested.html>

<https://programacion.net/articulo/transacciones_en_sql_server_299>

<https://www.fdi.ucm.es/profesor/fernan/DBD/apuntestema07.pdf>