**BASE DE DATOS – TRANSACCIONES**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Venimos ejecutando sentencias que impactaban a la bd. Todo lo que se ejecuta en la base de datos, forma parte de una transacción.**

**Atómico: o todo o nada. (lo que se ejecuta).**

**El que desarrolla la bd, va a decir donde empieza, y donde termina la transacción.**

**Si no se ejecuta todo, se va a deshacer todo, se vuelve a un estado previo.**

**Diagrama

Descripción generada automáticamente con confianza media**

**Si se cae la aplicación del banco, se corta la luz, o pasa otra cosa. No voy a poder retirar esos 10k. ¿En qué momento está tratando de consultar ese saldo? Si la transacción uno no se realizó correctamente, la transacción 2 no se puede ejecutar.**

**Nosotros decidimos cuando desarrollamos, qué queremos que pase y que no, el motor nos provee todas las herramientas para poder hacer estas transacciones**

**Diagrama

Descripción generada automáticamente**

**Commit: tipo de señal que confirma que todas las operaciones terminaron.**

**Parcialmente confirmada: pasa a un estado de rollback porque se cortó la luz u otra cosa, entonces vuelve al inicio. Si algunos de los estados fallan, se produce rollback -> deshace todas las operaciones, pero en realidad deshace todas las anteriores**

**¿Cómo y qué se puede rollbackear? :p**

**. Se guarda una imagen previa**

**Diagrama

Descripción generada automáticamente**

**Comando begin transaction. Commit transaction: comando que dice si terminó la operación.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Hago un select pero, ¿cómo viene la transacción a todo?:**

**implícitamente generó 2 transacciones. Si ya se ejecutó todo, el commit se hizo y la transacción se realizó.**

**Texto

Descripción generada automáticamente con confianza mediaasí se hace una sola transacción.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente fuimos a otra ventana, pero hasta que no insertemos el segundo valor, no se va a mostrar para evitar algo que no se debería mostrar.**

**- Nomás ejecutó el begin transaction:**

**Esto permita que cosas que no se terminaron de transaccionar, no se muestra, bloquea el recurso.**

**- Con el commit transaction, sí se va a mostrar todo.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente La transacción 63 pudo mostrar los datos, una vez que terminó la 54.**

**Tengo que decidir si hacer el auto commit, o no.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente no terminó la transacción -> deshice todo lo que se codeo.**

**MANEJO DE EXCEPCIONES O ERRORES (TRY/CATCH):**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente con confianza bajaTRATA DE EJECUTAR ESTO, SI DA ERROR, VA A IR AL CATCH**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteTIRA EL ERROR PORQUE SE REPITEN CLAVES PRIMARIAS.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteTABLA DE ERRORES QUE MUESTRA CUANDO NO SE PUEDE EJECUTAR.**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteSi el primero no se pudo insertar, Lola no se va a ejecutar y va a ir al catch para printear el error.**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteen este caso, ejecutó Juan, pero como Ana ya estaba metida, el resto no se insertó. Entonces se necesita un rollback.**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Acá Juan no está porque se hizo el rollback.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Esto pasa en las empresas cuando se llama a un web service, y no responde. Entonces para evitar errores, se hace un rollback.**

**Texto, Carta

Descripción generada automáticamente**

**Conservación de consistencia:**

**- ejemplo: yo no puedo insertar un valor de una fk que no existe en una tabla primaria.**

**Aislamiento: una transacción no debería interferir en otra transacción.**

**Durabilidad: Va a ser durable en el tiempo, por más que se haya cortado la luz. Si algo existió previo antes del error, tiene que volver a resurgir.**

**NO TODAS LAS TRANSACCIONES PUEDEN SER IDEALES SE PUEDE MANIPULAR EL AISLAMIENTO Y LA CONSISTENCIA. LAS OTRAS DOS NO SE MODIFICAN.**

**Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente**

**Con esto, muestra los dos registros igual.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente ahora se ejecuta el rollback.**

****

**SE ALTERA EL AISLAMIENTO**

**Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamenteahora no hay nada, te mostraba datos falsos -> DIRTY READ.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**¿QUÉ PASA SI 200 TRANSACCIONES QUIEREN HACER DEPOSITOS?**

**Control de concurrencia: Se pone un Schedule para esos procedimientos.**

**Control de recuperación: cómo va a recuperar los datos cuando el motor vuelva a la vidaDiagrama

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

El t2 hace como si el T1 nunca hubiese existido, para esto hay que hacer que termine un procedimiento, y después otro

X en t2 de principio vale 10. El anterior cálculo se hizo en memoria. NO SE GUARDA EN REGISTRO

**Cada línea, es una línea de tiempo.**

**Imagen que contiene Tabla

Descripción generada automáticamente**

**Con la transacción t1, no se commitió el valor. El valor t2, va a hacer una lectura falsa. Leyó un valor que nunca fue comitteado en la bd.**

**EJ: viene alguien, me aumenta el sueldo, pero ya había otro haciéndolo. Entonces el que vino después no puede leer el valor hasta que el otro primero termine de actualizarme el sueldo.**

**Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente**

**X tiene un valor en el inicio de la transacción, pero en la segunda lectura, va a tener otro porque la t2 ya lo escribió y lo comittió. LA LECTURA NO ES REPETIBLE.**

**A diferencia del dirty read: ACÁ EL VALOR SÍ SE COMMITEA.**

**Por ahí en principio tu sueldo era de 10k y en el medio se hace una modificación gracias a la otra transacción, y después pasa 20k.**

**IGUAL ESTO PUEDE PASAR QUE SE QUIERA HACER A PROPÓSITO EN ALGÚN CASO**

**Texto

Descripción generada automáticamente**

**Si hay una transacción en proceso, no voy a poder ver mi plata en la caja de ahorro.**

**Serializable: si ejecuto en forma intercalada, o en serie, da el mismo resultado.**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Equivalentes por resultados: No interesa cómo intercambia las operaciones, interesa el resultado nomás. Pero puede haber un problema, como en la matem. Se llega al mismo resultado, con dif procedimientos, pero por ahí de casualidad. No es muy confiable. Por eso se utiliza más:**

**Equivalentes por conflictos: plan serializable por conflicto**

**Texto

Descripción generada automáticamenteno hay problemas cuando se HACEN DOS LECTURAS**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Gráfico, Diagrama

Descripción generada automáticamente**

**El grafo dice que hay dos transacciones, y tiene un conflicto en x**

**HAY CONFLICTOS PORQUE HAY DOS WRITES(x).**

**El plan equivalente va a correr de modo aislado, y no va a haber problemas.**

**Diagrama

Descripción generada automáticamente**

**O lee el dirty bit, o se pierde la otra transacción**

**Ordené entre las transacciones de dif manera que la anterior.**

**Conflicto: ya se había escrito X.**

**La flecha de t2 a t1, lee de x, y escribe x en t1. Cada arco es un conflicto. Acordarse de que las únicas que no tiran error son las de leer y leer**

**Diagrama

Descripción generada automáticamente**

**Read(X)**

**Ciclico(Excepción): serializable, porque empiezo por t3. No puedo ir a t2 porque la flecha viene de t1. Entonces tengo que resolver de T1 a T2. TERMINO EN T2.**

**No hay nadie que le entre al t3.**

**¡Podría existir más de un plan!**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Es como un candado que no me permite utilizar el recurso.**

**Texto

Descripción generada automáticamente**

**Diagrama

Descripción generada automáticamente**

**Texto, Carta

Descripción generada automáticamente**

**Bloqueo para lectura cuando quiero leer, bloqueo escritura cuando quiero escribir…**

**Diagrama

Descripción generada automáticamente**

**Lock-s: lectura**

**Lock-x: escritura**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media**

**Diagrama

Descripción generada automáticamente con confianza baja**

**SELECT: SHARED Exclusive: modifico/actualizo**

**CUANDO LEO UNA TRANSACCIÓN, NO PUEDO LEER SI ESTÁ BLOQUEADO NORMALMENTE. LOCK. PORQUE NO SE IDENTIFICÓ QUE LOCK ES**

**CUANDO YO BLOQUEO DE LECTURA, ME ESTOY ASEGURANDO QUE NADIE ESTÁ ESCRIBIENDO AL MISMO TIEMPO. SI ESTOY LEYENDO MI RECIBO DE SUELDO, NADIE ME VA A PODER MODIFICAR EL VALOR.**

**VICEVERSA PARA EL OTRO**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Diagrama

Descripción generada automáticamente**

**CASO:**

**PUEDE SER QUE EN LA FASE DE LIBERACIÓN: SI SE ME CAE EL MOTOR LA BASE DE DATOS -> HACE ROLLBACK (SI YA LIBERÉ EL RECURSO, PUEDE HABER ALGUIEN QUE HAYA TOMADO ESE RECURSO QUE TOMÉ)**

**POR ESO ESTE PROTOCOLO SE TRANSFORMA EN: Gráfico de líneas

Descripción generada automáticamente con confianza media**

**TO FACE COMMITED (LOS QUE USAN LOS MOTORES COMERCIALES)**

**VENTAJA:**

**Básicamente cuando se libera un recurso, si una persona lo toma es porque ya se commitió o yo se rollbackeó.**

**No puede cambiar la transacción antes de tiempo**

**LOS RECURSOS VAN A ESTAR BLOQUEADOS HASTA EL FINAL**

**¡No es eficiente, pero seguro! -> tiene que hacer más commits para reducir el tamaño de las transacciones**

**A VECES LAS BD TARDAN PORQUE:**

**- ESTÁN ESPERANDO UN RECURSO**

**- EL PLAN DE EJECUCIÓN ES INCORRECTO**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Imagen que contiene Diagrama

Descripción generada automáticamente**

**PUEDE SER CUALQUIERA DE LOS 4 TIPOS. EN ESTOS ACEPTOS LOS TIPOS DE ERROR. SINO EL RECURSO LO VA A BLOQUEAR HASTA QUE TERMINE LA TRANSACCIÓN**

**TIENE QUE SER SERIALIZABLE PARA QUE SEA UNA TRANSACCIÓN IDEAL.**

**Texto, Correo electrónico

Descripción generada automáticamente**

**PHANTOM: los registros no existían previamente, y en medio de la transacción existen.**

**Cuanto más restricciones tengo, más se va a quedar esperando. Por eso se baja el nivel de aislamiento, para permitir más allá de que estos problemas sucedan.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

Mínima granularidad

**Yo quiero que me responda rápido. Respecto a la granularidad: voy a tratar de bloquear lo menos que pueda al principio, ya que sino no se va a mostrar los datos de principio. Después le voy a meter más bloqueos**

**Para esto se puede empezar por hacerse un índice.**

**Imagen que contiene Tabla

Descripción generada automáticamente**

**Espera infinita**

**El rollback o el commit libera los recursos, entonces mata a la t2 se podría sacar para liberar Y, y el t1 termina de ejecutar. SOLO SE PUEDE ROLLBACKEAR CON UNO SOLO**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Diagrama

Descripción generada automáticamente**

**Significa hacerlo más grande al lockeo -> desde un lock-s (share) a un lock-x (exclusive)**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteno puedo esperar tanto tiempo por el recurso, entonces aborto. Se establece un período por default.**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteImagen que contiene Diagrama

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente con confianza baja**

**Este log tiene un tamaño, no es infinito. Cuando llega al tope de espacio, si es circular el log se empieza a pisar.**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Gráfico, Diagrama

Descripción generada automáticamente**

**T4 no tiene que hacer nada, ya que no se pasó los datos a discos, ya que no se produjo un checkpoint. No se vuelve a ejecutar porque se hace un rollback, y t4 se va.**

**Todos los bloques de t3, habían terminado bien. Entonces el motor va a tener que leer del archivo de log y ver dónde está el dato. Así con los anteriores. TODO ESTO POR EL PROCESO DE DURABILIDAD.**

**EL CHECKPOINT COMITEA TODO LO QUE HAY EN VUELO.**

**EL PEDAZO DE T2 Y T3, NO IMPACTARON EN EL DATAFILE, SINO EN EL LOG.**

**Diagrama

Descripción generada automáticamente**

**LOG ES CIRCULAR: SE REPITE LA HISTORIA.**

**SECUENCIALES: TENGO TODA LA HISTORIA QUE VOY A HACER EN EL DÍA. DE PUNTA A PUNTA.**