BASES DE DATOS

Transacciones
Seguridad

Cap. 14, 15 y 16 C.J. Date

Agenda

- Transacciones
  - Recovery
  - Concurrencia
- Seguridad

Transacciones

“Representa una unidad lógica de procesamiento de base de datos que debe completarse en su totalidad.”

“ Conjunto de operaciones que deben ejecutarse todas o ninguna.”

Los límites de una transacción pueden explicitarse mediante …

Begin transaction

…

…

Commit, Rollback

Transacciones

> Las transacciones deben cumplir con ciertas propiedades denominadas ACID, que deben ser implementadas por el control de concurrencia y los métodos de recuperación del DBMS.

***A**tomicidad.*

***C**onsistencia.*

*Aislamiento (**I**solation)*

***D**urabilidad*

Transacciones

Atomicidad. Las operaciones correspondientes a una transacción se realizan en su totalidad o ninguna de ellas se realiza

Consistencia. La ejecución de una transacción conserva la consistencia de la base de datos.

Aislamiento. Cada transacción ignora al resto de las transacciones que se ejecuten concurrentemente en el sistema.

Durabilidad. Finalizada con éxito la transacción, los cambios realizados en la base de datos permanecen, incluso si hay fallos en el sistema.

Transacciones

“

Las transacciones están asociadas al **recovery** y a la **concurrencia**.

Las transacciones enviadas a la BD por varios usuarios pueden ejecutarse **concurrentemente**.

Pueden acceder y actualizar los **mismos** elementos de la base de datos.

Si esta ejecución concurrente no está controlada pueden surgir problemas, como una base de datos inconsistente.

En caso de fallos en alguna transacción la BD debe recuperarse automáticamente.

Para la recuperación se utilizan los archivos de **log**s**.**

Recovery

- Recuperarse de un fallo de una transacción normalmente significa que la base de datos se restaura al estado consistente más reciente, inmediatamente anterior al momento del fallo.
- Está basada en la ***redundancia*.**
- Para ello, el sistema debe guardar información sobre los cambios que las distintas transacciones aplicaron a los datos. Esta información se guarda en los ***logs*** del sistema.
- Las transacciones representan una unidad de recuperación (recovery).

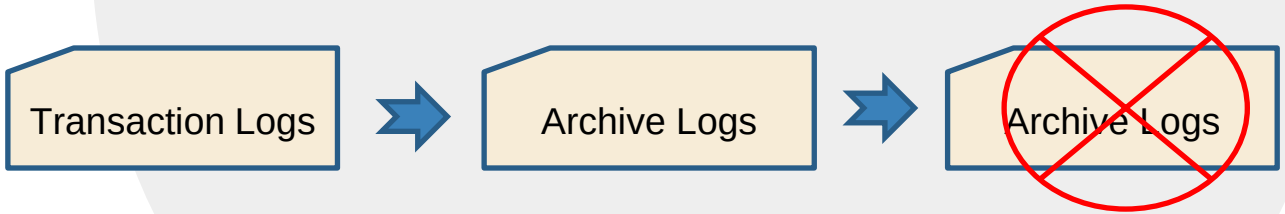
Recovery.

Transaction log (Redo Logs)

Archive logs

Administrador de transacciones

El transaction log es un archivo que almacena secuencialmente todas las modificaciones en la BD.



Recovery

Transacciones - Entradas en el log

1. [start_transaction, T]. Indicador de inicio de la transacción T.
2. [write_item, T, X, valor_viejo, valor_nuevo]. Indica que la transacción T ha cambiado el valor del elemento X de valor_viejo a valor_nuevo.
3. [read_item, T, X]. Indica que la transacción T ha leído el valor del elemento X de la base de datos.
4. [commit, T]. Indica que la transacción T se ha completado satisfactoriamente, y que puede persistirse en la base de datos.
5. [abort, T]. Indica que la transacción T se abortó
6. [Check Point]. Escribe físicamente el contenido de los búferes en la BD y un registro de punto de verificación en el log.

Recovery

El archivo de log

Permite deshacer o rehacer las operaciones ante fallos.

- Si se produce un fallo del sistema, se buscan hacia atrás en el **log** del sistema todas las transacciones T que han escrito un registro [start transaction, T] pero que no tengan su registro [commit, T]; de esta manera es posible anular estas transacciones para deshacer su efecto en la base de datos.
- Las transacciones que han escrito su registro commit en el **log** también deben haber registrado todas sus operaciones de escritura, para que su efecto en la base de datos pueda rehacerse a partir de las entradas del **log**.

Recovery

- Fallas del Medio
- Fallas del Sistema.

Fallas del Medio

Las **fallas del medio** son fallas del medio de almacenamiento en la que la BD se destruye físicamente.

Implica la restauración desde un Backup y los logs (almacenados).

Backup/Recovery

Recovery

Fallas del sistema

- Son aquellas en las que la BD se vuelve inconsistente.
- Se resuelven deshaciendo o rehaciendo operaciones de transacciones que provocaron la inconsistencia.
- Se pierde el contenido de la **memoria principal**
- Se realizan Check Points a ciertos intervalos
- Se recuperan mediante técnicas de ***actualización diferida*** o de ***actualización inmediata.***

Recovery

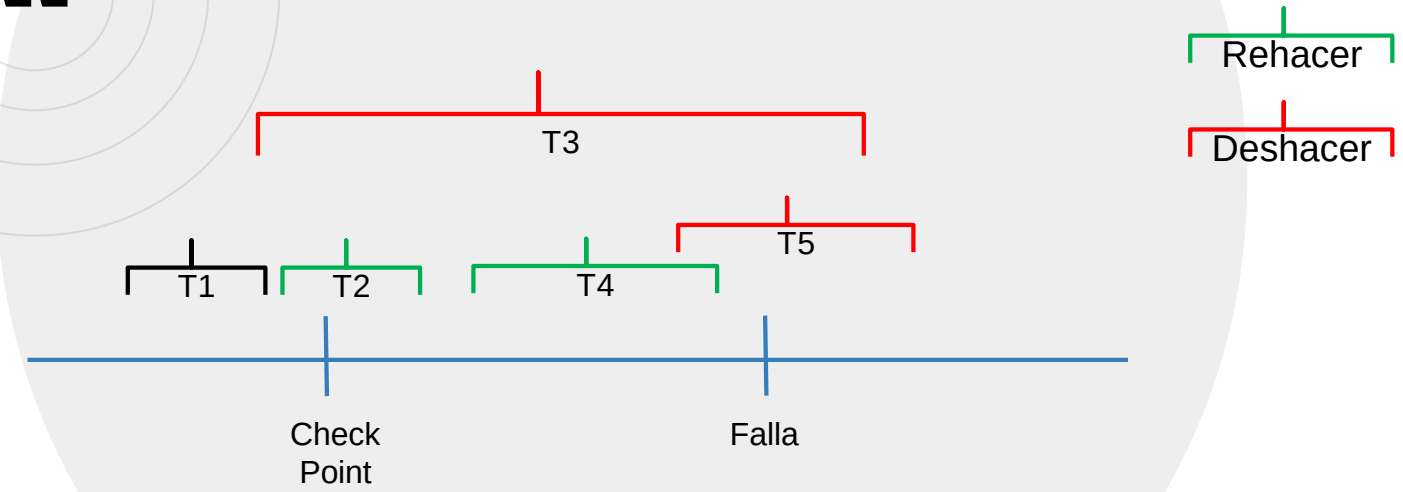
Fallas del sistema

Actualización diferida. No es necesario deshacer los cambios en la BD pero si puede que haya que aplicarlos pues se aplican en el log y no en la BD

Actualización inmediata. Puede que haya que deshacer y aplicar cambios en la BD pues se aplican los cambios tanto en el log como en la BD.

Recovery

Fallas del sistema



Concurrencia.

Problemas.

Lectura sucia. Ocurre cuando una transacción lee datos que aún no han sido confirmados *“comiteados”* por otra transacción.

Lectura no repetible. Supongamos que la transacción T1 recupera una fila, luego la transacción T2 **actualiza** esa fila (commit) y después la transacción T1 recupera nuevamente la "misma" fila. La transacción T1 ha recuperado la "misma" fila dos veces, pero ve dos valores diferentes.

Lectura fantasma. Supongamos que la transacción T1 recupera el conjunto de todas las filas que satisfacen alguna condición (por ej. todas las filas de proveedores que satisfacen la condición que residen en París). Supongamos que la transacción T2 **inserta** entonces una nueva fila que satisface la misma condición. Si la transacción T1 repite la lectura, verá una fila que antes no existía, un "fantasma".

Concurrencia

Niveles de aislamiento: Son los niveles de interferencia entre transacciones.

- Read uncommited
- Read commited
- Repeatable Reads
- Serializable

Aislamiento	Interferencia	Concurrencia
Menos	Menor	Mayor
Mas	Mayor	Menor

SET TRANSACTION ISOLATION LEVEL
[READ UNCOMMITTED **/ READ COMMITTED** / REPEATABLE READ / SERIALIZABLE]

Concurrencia.

T1	T	T2
drop table borrar create table borrar(col1 char(2), col2 int);	T1	SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED DBCC USEROPTIONS
insert into borrar values ('AA', 1), ('BB', 1), ('DD', 2);	T2	
begin transaction	T3	begin transaction
insert into borrar values ('EE', 3);	T4	
	T5	select * from borrar
rollback	T6	
	T7	select * from borrar

Qué valores devuelve ?

Qué valores devuelve ?

En el primer Select devuelve la fila ‘EE’ en el Segundo Select **no devuelve nada.**

Lectura sucia

Concurrencia.

Col1	Col2
AA	1
BB	1
DD	2

“

	T1	SET TRANSACTION ISOLATION LEVEL READ COMMITTED DBCC USEROPTIONS
	T2	begin transaction
begin Transaction	T3	select * from borrar where col1 = 'DD'
update borrar set col2 = 1 where col1 = 'DD';	T4	
commit	T5	
	T6	select * from borrar where col1 = 'DD'
	T7	commit

Qué valores devuelve ?

Qué valores devuelve ?

En el primer select devuelve (‘DD’, 2)
En el Segundo (‘DD’, 1).

Lectura no repetible

Concurrencia.

Col1	Col2
AA	1
BB	1
DD	1

“

	T1	SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
	T2	begin transaction
	T3	select * from borrar where col2 =1;
insert into borrarvalues ('EE', 1);	T4	
	T5	select * from borrar where col2 =1;
	T6	commit

Qué valores devuelve ?

Qué valores devuelve ?

En el primer select devuelve las filas ‘AA’, ‘BB’, ‘DD’.
En el Segundo devuelve además ‘EE’.

Lectura fantasma.

Concurrencia.

“

	T1	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
	T2	begin transaction
	T3	select * from borrar where col2 =1;
insert into borrar values ('EE', 1);	T4	
	T5	select * from borrar where col2 =1;
	T6	commit
(Recién ahora se inserta la fila)	T7	

Qué valores devuelve ?

Qué valores devuelve ?

Ambos devuelven los mismos datos.

Concurrencia.

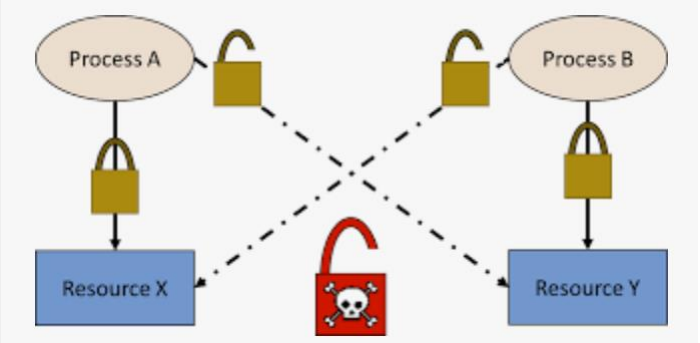
Niveles de Aislamiento	Lectura sucia	Lectura no repetible	Lectura fantasma
READ UNCOMMITED	SI	SI	SI
READ COMMITED	NO	SI	SI
REPEATABLE READ	NO	NO	SI
SERIALIZABLE	NO	NO	NO

SET TRANSACTION ISOLATION LEVEL
[READ UNCOMMITTED **/ READ COMMITTED** / REPEATABLE READ / SERIALIZABLE]

Concurrencia

Bloqueo Mortal (DeadLock)

Se da cuando transacciones esperan por recursos de otras bloqueándose a sí mismas.



Transacción 1	T1	Transacción 2
Bloquear A	T1	
	T2	Bloquear B
Bloquear B	T3	
Espera	T4	Bloquear A
Espera	T5	Espera

Seguridad

El **Administrador de la base de datos (DBA)** es el responsable principal de la gestión de la BD.

Entre las responsabilidades del DBA se encuentran:

- Conceder y retirar privilegios a los usuarios que necesitan utilizar el sistema.
- Clasificación de usuarios y datos en función de la política de la organización.

Para realizar sus funciones el DBA dispone de una cuenta de ***DBA*** en el ***RDBMS***.

Seguridad

Acciones relacionadas a seguridad:

- Creación de cuentas de usuarios. Mediante esta acción se crea una nueva cuenta y contraseña para posibilitar el acceso al DBMS a un usuario o grupo de usuarios.
- Concesión y retiro de privilegios. Esta acción permite al DBA conceder determinados permisos. A usuarios y Roles

Seguridad

El DBA crea los Usuarios y Contraseñas.

Con dicho usuario (o cuenta) se inicia sesión.

Los usuarios pueden ser Aplicaciones.

Los usuarios (y passwords) se almacenan en el **catálogo.**

Los ***logs*** del sistema registran las operaciones que realiza la cuenta.

Seguridad – Control de accesos

Control de accesos.

- Nivel de cuenta
- Nivel de tablas (Objetos)

Seguridad – Control de accesos

Control de accesos a Nivel de cuenta

Asigna permisos para ejecutar sentencias DDL.

- CREATE SCHEMA
- CREATE TABLE
- CREATE VIEW
- ALTER, para realizar cambios en el esquema como la agregación o eliminación de atributos
- DROP

Seguridad – Control de accesos

Control de accesos a Nivel de Tablas

Asigna permisos para ejecutar sentencias DML

- Privilegio de selección. Se concede a la cuenta el privilegio de utilizar la instrucción SELECT para obtener tuplas de R.
- Privilegio de modificación. Este privilegio se subdivide en privilegios sobre UPDATE, DELETE e INSERT. Los privilegios INSERT o UPDATE pueden especificar que sólo ciertos atributos de R pueden ser actualizados por la cuenta.
- Privilegio de referencia. Concede la posibilidad de referenciar la tabla R al especificar restricciones de integridad.

Seguridad

Control de accesos Mediante Vistas

Se puede crear una vista sobre una tabla T que sólo pueda ver ciertos atributos y a su vez condicionar la mediante un WHERE el acceso a ciertas filas.

Luego dar el privilegio de SELECT sobre la vista a cierto usuario.

Nota: Para crear una Vista se debe tener privilegio de SELECT sobre las tablas implicadas.

Seguridad

Propagación de privilegios

Si el propietario A de una tabla T concede un privilegio a otra cuenta B, dicho privilegio se puede conceder a B con o sin la opción ***GRANT OPTION.*** Si aparece esta opción, significa que B también puede conceder ese privilegio sobre T a otras cuentas.

Seguridad

Propagación y revocación de privilegios

Por ejemplo el usuario **A1** ejecuta

```
GRANT SELECT ON EMPLEADO TO A3 WITH GRANT OPTION
```

La cláusula WITH GRANT OPTION significa que el usuario **A3** puede propagar el privilegio a otras cuentas.

Ahora **A3** puede conceder el privilegio SELECT sobre la tabla EMPLEADO a **A4** ejecutando

```
GRANT SELECT ON EMPLEADO TO A4
```

A4 no puede propagar el privilegio SELECT

Supongamos ahora que **A1** decide revocar a **A3** el privilegio SELECT sobre la relación EMPLEADO. **A1** podría ejecutar el comando

```
REVOKE SELECT ON EMPLEADO FROM A3;
```

Por lo cual el usuario **A3** pierde los privilegios sobre EMPLEADO y para conceder permisos sobre EMPLEADO. **A4** sigue con privilegios.

Seguridad

Control de accesos basado en ROLES

Los roles se pueden crear mediante los comandos **CREATE ROLE** y borrar mediante **DROP ROLE**.

Los comandos **GRANT** y **REVOKE** que se vieron en los controles de acceso anteriores pueden ser usados para conceder y revocar privilegios a los roles.

Seguridad

```
GRANT [privilegios]
      ON [objeto]
       TO [usuarios/roles]
[WITH GRANT OPTION]
```

Privilegios DML

SELECT, INSERT, UPDATE, DELETE

```
REVOKE [GRANT OPTION FOR] [privilegios]
      ON [objeto]
    FROM [usuarios/roles]
         [CASCADE]
```

Preguntas ?