

Ingeniería en Sistemas Computacionales

Taller de Base de Datos

Unidad V: Transacciones

Este material está desarrollado para la asignatura Taller de Base de Datos, de la carrera de Ingeniería en Sistemas Computacionales, plan de estudios ISIC-2010-224

TALLER DE BASE DE DATOS

Unidad	Temas	Subtemas
5	Transacciones	5.1 Conceptos básicos 5.2 Propiedades de las transacciones 5.3 Grados de consistencia 5.4 Niveles de aislamiento 5.5 Commit y rollback

Competencia: Aplicar transacciones para garantizar la consistencia de la base de datos.

TALLER DE BASE DE DATOS

Transacciones

- Una transacción es un conjunto de acciones llevadas a cabo por un usuario o un programa de aplicación, que acceden o cambian el contenido de la base de datos.

- Las transacciones representan eventos del mundo real, como:

Ejemplo

Una compra puede ser una transacción que conste de dos operaciones:

- Insertar un registro del pago del producto
- Insertar el producto en el inventario.

No se debe insertar un producto que no se haya pagado, ni pagar un producto que no esté en el inventario, por lo tanto, las dos operaciones forman una transacción.

TALLER DE BASE DE DATOS

- Las características que se debe recoger de cada transacción son las siguientes:
 1. Datos que utiliza la transacción.
 2. Características funcionales de la transacción.
 3. Salida de la transacción.
 4. Importancia para los usuarios.
 5. Frecuencia de utilización.

TALLER DE BASE DE DATOS

- **En las transacciones de recuperación** se accede a los datos para visualizarlos en la pantalla a modo de informe.
- **En las transacciones de actualización** se insertan, borran o actualizan datos de la base de datos.
- **En las transacciones mixtas** se mezclan operaciones de recuperación de datos y de actualización.

TALLER DE BASE DE DATOS

Propiedades de las Transacciones

- Las transacciones deben cumplir cuatro propiedades ACID:

1. Atomicidad (**A**tomicity)
2. Consistencia (**C**onsistency)
3. Aislamiento (**I**solation)
4. Permanencia (**D**urability):

TALLER DE BASE DE DATOS

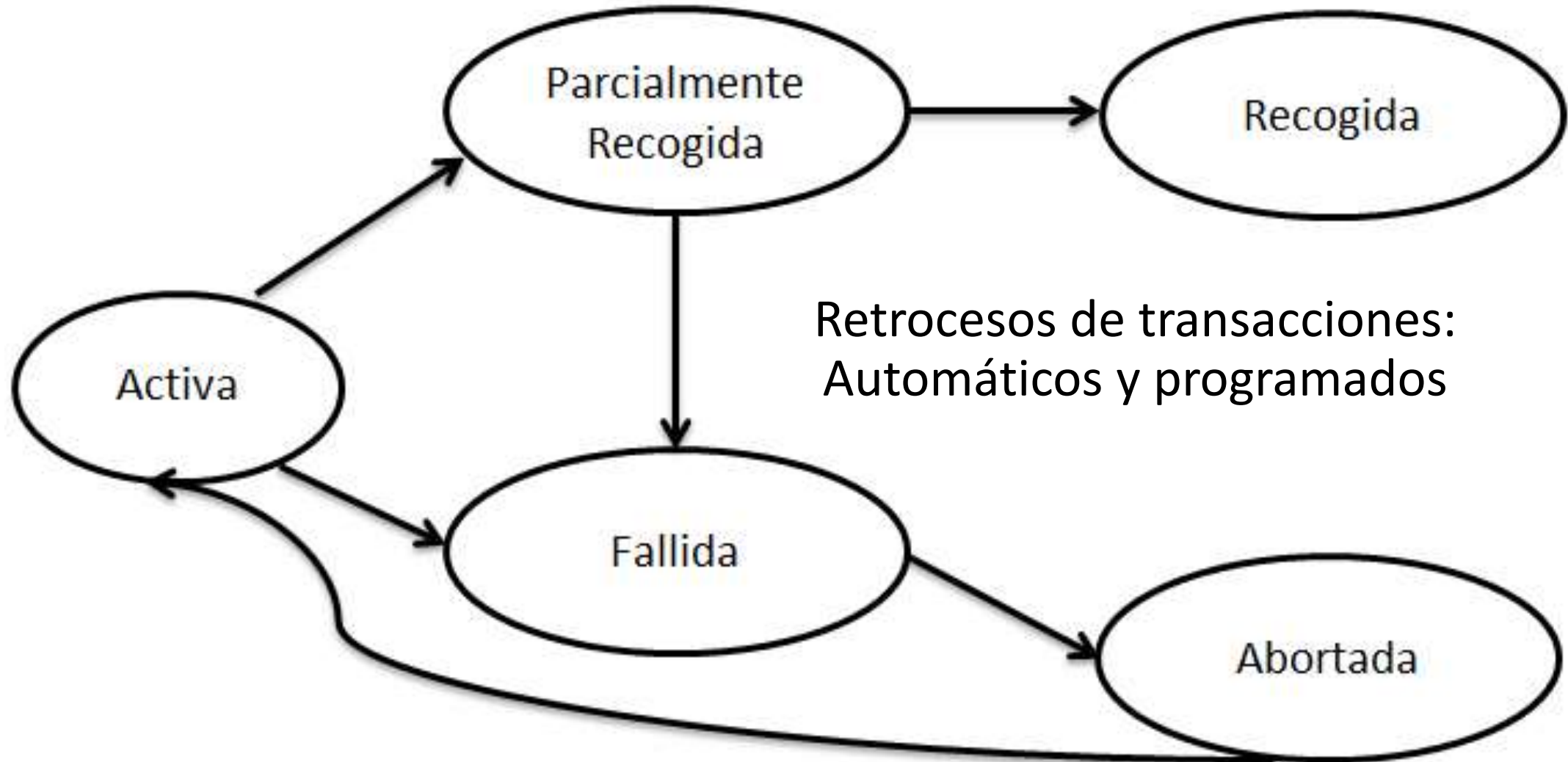
- **Atomicidad (Atomicity):** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia (Consistency):** es la propiedad que asegura que sólo se empiece aquello que se puede acabar. Por lo tanto, se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.

TALLER DE 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.
- **Permanencia (Durability):** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
 - **Nota:** Si la transacción no se puede finalizar por cualquier motivo, el SGBD garantiza que los cambios realizados por esta transacción son deshechos.

TALLER DE BASE DE DATOS

Diagrama de Estados de una Transacción



TALLER DE BASE DE DATOS

Grados de Consistencia de las Transacciones

- Consistencia es un término más amplio que el de integridad. Podría definirse como la coherencia entre todos los datos de la base de datos.
- Cuando se pierde la integridad también se pierde la consistencia. Pero la consistencia también puede perderse por razones de funcionamiento.

TALLER DE BASE DE DATOS

- Una transacción mantendrá la consistencia de la base de datos. Esto es, si la base de datos se encuentra en un estado consistente antes de ejecutar la transacción, una vez que ésta termine la consistencia de la base de datos deberá conservarse. Por consistente se debe entender, internamente consistente.
- **Consistencia de datos:** Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

TALLER DE BASE DE DATOS

- En términos de base de datos esto significa que se satisfacen todas las restricciones en cuanto a su integridad que incluyen:
 - Todos los valores de la llave primaria son únicos.
 - La base de datos mantiene integridad referencial lo que significa que los registros solo referencian información que existe.
 - Ciertos predicados se mantienen. Por ejemplo, la suma de los gastos es menor o igual al presupuesto.
 - A diferencia de la atomicidad, el aislamiento y la durabilidad, la consistencia es una práctica de programación. La atomicidad, el aislamiento y la durabilidad están aseguradas estén o no programadas para preservar la consistencia. Es responsabilidad del desarrollador de la aplicación asegurar que su programa preserva la consistencia.

TALLER DE BASE DE DATOS

- Una transacción finalizada (***confirmada parcialmente***) puede no confirmarse definitivamente (***consistencia***).
 - Si se confirma definitivamente el sistema asegura la persistencia de los cambios que ha efectuado en la base de datos.
 - Si se anula los cambios que ha efectuado son deshechos.
- La ejecución de una transacción debe conducir a un estado de la base de datos consistente (**que cumple todas las restricciones de integridad definidas**).
 - Si se confirma definitivamente el sistema asegura la persistencia de los cambios que ha efectuado en la base de datos.
 - Si se anula los cambios que ha efectuado son deshechos.

TALLER DE BASE DE DATOS

- Una transacción que termina con éxito se dice que está recogida o comprometida (**committed**), una transacción que haya sido recogida llevará a la base de datos a un nuevo estado consistente que debe permanecer incluso si hay un fallo en el sistema

TALLER DE BASE DE DATOS

Niveles de Aislamiento

¿Qué es el Aislamiento?

- En bases de datos, el **aislamiento** es una propiedad que define cómo y cuándo los cambios producidos por una operación se hacen visibles para las demás operaciones concurrentes. Aislamiento es una de las 4 propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) aplicables a una base de datos transaccional.

TALLER DE BASE DE DATOS

- De las cuatro propiedades ACID de un Sistema de gestión de bases de datos relacionales (SGBDR) la de aislamiento es la que más frecuentemente se relaja. Para obtener el mayor nivel de aislamiento, un SGBDR generalmente hace un bloqueo de los datos o implementa un Control de concurrencia mediante versiones múltiples (MVCC), lo que puede resultar en una pérdida de concurrencia. Por ello se necesita añadir lógica adicional al programa que accede a los datos para su funcionamiento correcto.
- Este es el nivel de aislamiento *más alto*. Especifica que todas las transacciones ocurran de modo aislado, o dicho de otro modo, como si todas las transacciones se ejecutaran de modo serie (una tras otra). La sensación de ejecución simultánea de dos o más transacciones que perciben los usuarios sería una ilusión producida por el SGBD.

TALLER DE BASE DE DATOS

Lecturas Recogidas (Read Committed)

- En este nivel de aislamiento, un SGBDR que implemente el control de concurrencia basado en bloqueos mantiene los bloqueos de escritura -de los datos seleccionados - hasta el final de la transacción, mientras que los bloqueos de lectura se cancelan tan pronto como acaba la operación de SELECT (por lo que el efecto de las ***lecturas no repetibles*** puede ocurrir, como se explica más abajo). Al igual ocurría en el nivel anterior, no se gestionan los *bloqueos de rango*.

TALLER DE BASE DE DATOS

Lecturas no Recogidas (Read Uncommitted)

- Este es el menor nivel de aislamiento. En él se permiten las ***lecturas sucias*** (ver más abajo), por lo que una transacción puede ver cambios no cometidos aún por otra transacción.

TALLER DE BASE DE DATOS

Nivel de Aislamiento por Defecto

- El nivel de aislamiento por defecto de distintos SGBDR varía ampliamente. La mayoría de bases de datos que gestionan transacciones permiten al usuario establecer cualquier nivel de aislamiento.
- Algunos SGBDR requieren sintaxis especial cuando se realiza una operación SELECT que efectúa bloqueos (e.g. *SELECT ... FOR UPDATE* para bloquear para escritura aquéllas filas accedidas)

TALLER DE BASE DE DATOS

Commit y Rollback

¿Qué es commit?

- En el contexto de la Ciencia de la computación y la gestión de datos, **commit**(acción de recoger o comprometer) se refiere a la idea de consignar un conjunto de cambios "tentativos, o no permanentes".
- Un uso popular es al final de una transacción de base de datos.

TALLER DE BASE DE DATOS

Una sentencia **Commit**

- En SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional (RDBMS) y pone visibles todos los cambios a otros usuarios. El formato general es emitir una sentencia BEGIN WORK, una o más sentencias SQL, y entonces la sentencia COMMIT. Alternativamente, una sentencia ROLLBACK se puede emitir, la cual deshace todo el trabajo realizado desde que se emitió BEGIN WORK. Una sentencia COMMIT publicará cualquiera de los savepoints (puntos de recuperación) existentes que puedan estar en uso.
- En términos de transacciones, lo opuesto de **commit** para descartar los cambios "en tentativa" de una transacción, es un **rollback**.

TALLER DE BASE DE DATOS

¿Qué es rollback?

- En tecnologías de base de datos, un **rollback** es una operación que devuelve a la base de datos a algún estado previo. Los Rollbacks son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas.
- Son cruciales para la recuperación de crashes de un servidor de base de datos; realizando rollback(devuelto o deshecho) cualquier transacción que estuviera activa en el tiempo del crash, la base de datos es restaurada a un estado consistente.

TALLER DE BASE DE DATOS

- En SQL, ROLLBACK es un comando que causa que todos los cambios de datos desde la última sentencia BEGIN WORK, o START TRANSACTION sean descartados por el sistema de gestión de base de datos relacional (RDBMS), para que el estado de los datos sea "rolled back"(devuelto o deshecho) a la forma en que estaba antes de que aquellos cambios tuvieran lugar.

TALLER DE BASE DE DATOS

- BEGIN TRANSACTION (Transact-SQL): Marca el punto de inicio de una transacción local explícita

Sintaxis

```
BEGIN { TRAN | TRANSACTION }  
    [ { transaction_name | @tran_name_variable }  
      [ WITH MARK [ 'description' ] ]  
    ]  
[ ; ]
```


TALLER DE BASE DE DATOS

COMMIT

- Esta instrucción de SQL se utiliza para confirmar como permanentes las modificaciones realizadas en una transacción.

BEGIN TRANSACTION

Insert into Mvtos Values ('0150', 'Dep', 1500, '08-10-2008')

Insert into Mvtos Values ('7120', 'Ret', 1500, '08-10-2008')

Update Cuentas Set Saldo = Saldo + 1 where Numero = '0150'

Update Cuentas Set Saldo = Saldo - 1 Where Numero = '7120'

COMMIT TRANSACTION

- Con la instrucción **commit tran** se pone una marca para saber hasta que punto se hizo la transacción.

TALLER DE BASE DE DATOS

Ejemplos

- Asignar un nombre a una transacción En el siguiente ejemplo se muestra cómo asignar un nombre a una transacción.

```
DECLARE @TranName VARCHAR(20);  
SELECT @TranName = 'MyTransaction';  
BEGIN TRANSACTION @TranName;  
    USE AdventureWorks2012;  
    DELETE FROM AdventureWorks2012.HumanResources.JobCandidate  
        WHERE JobCandidateID = 13;  
COMMIT TRANSACTION @TranName;  
GO
```

TALLER DE BASE DE DATOS

- Marcar una transacción En el siguiente ejemplo se muestra cómo marcar una transacción. Se marca la transacción CandidateDelete.

```
BEGIN TRANSACTION CandidateDelete
```

```
    WITH MARK N'Deleting a Job Candidate';
```

```
GO
```

```
USE AdventureWorks2012;
```

```
GO
```

```
DELETE FROM AdventureWorks2012.HumanResources.JobCandidate
```

```
    WHERE JobCandidateID = 13;
```

```
GO
```

```
COMMIT TRANSACTION CandidateDelete;
```

```
GO
```