

Bases de Datos

UD 3: Sistemas de Gestión de Bases de Datos

Grado en Ingeniería Informática

Semestre: A

Curso: 2021-2022

Profesor: Juan Carlos Casamayor y Laura Mota

Objetivos

- Presentar la propiedad de *independencia de los datos* y la arquitectura ANSI/SPARC con esquema y niveles de abstracción.
- Presentar el procesamiento transaccional y el control de la *integridad* tanto a nivel semántico (restricciones de integridad) como el asociado con la concurrencia.
- Presentar los aspectos de *recuperación* y *seguridad* en base de datos

Índice

1 Arquitectura ANSI/SPARC



1.1 Esquemas y niveles de abstracción

1.2 Funcionamiento básico de un sistema de gestión de bases de datos

1.3 Independencia de datos

2 Transacciones, integridad y concurrencia

2.1 Concepto de transacción

2.2 Integridad semántica

2.3 Control de accesos concurrentes

3 Recuperación y seguridad

3.1 Reconstrucción de la base de datos

3.2 Seguridad

1 Arquitectura ANSI/SPARC

SGBD



herramienta (software) para la gestión (creación y manipulación) de bases de datos

SGBD



modelo de datos



- ✓ estructuras de datos
- ✓ operadores asociados

SGBD



debe mantener



- ✓ independencia
- ✓ integridad
- ✓ seguridad

1.1 Esquemas y niveles de abstracción

esquema conceptual: descripción del sistema de información desde un punto de vista organizativo independiente del SGBD que se utilice, e incluso de que se utilicen o no técnicas de bases de datos. En este esquema se describe la información y las funciones de la organización desde un punto de vista no informático.

esquema lógico: definición de la base de datos expresada en términos del modelo de datos en que se base el SGBD que se vaya a utilizar, sin entrar en detalles de su representación física.

esquema físico: definición de la representación de la base de datos en la memoria secundaria del computador.

esquema externo: definición de los datos de interés para un grupo de usuarios. Cada esquema externo consiste en un conjunto de estructuras derivadas definidas a partir de las estructuras del esquema lógico.

Descripción del sistema de información

Una pequeña inmobiliaria desea mantener información sobre los edificios cuya venta gestiona. Se quiere saber:

- De cada edificio, el código, la ubicación, el distrito, el propietario, el precio solicitado por éste y el agente encargado de la venta si ya está asignado.
- De cada propietario, el DNI, el nombre y el teléfono.
- De cada agente el DNI, el nombre, la comisión por cada venta, los años de antigüedad y el teléfono.

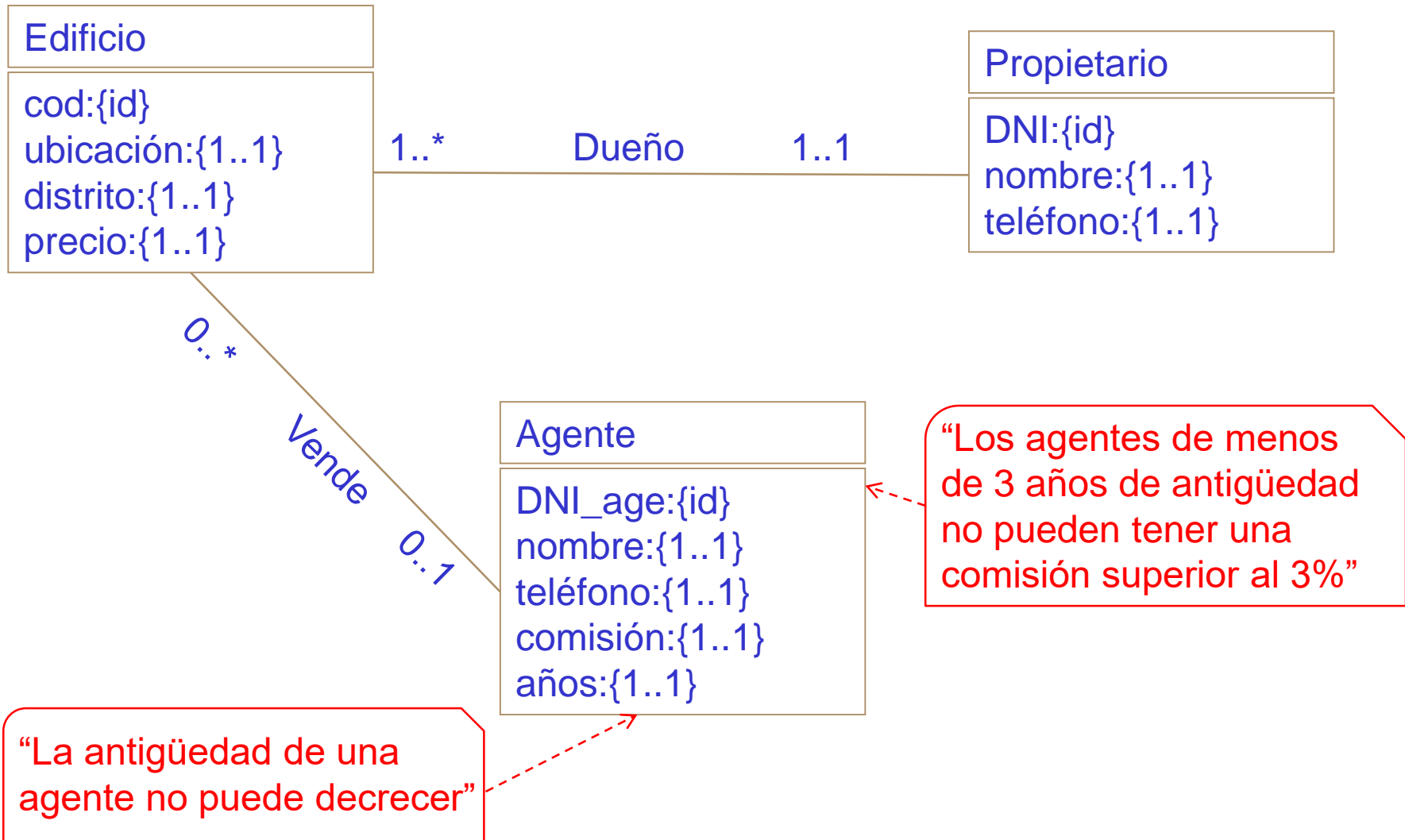
Las restricciones que deben cumplirse son las siguientes:

- La comisión de un agente no puede exceder el 3% si su antigüedad es menor de 3 años.
- No se quiere tener información de propietarios si no se tiene al menos un edificio del propietario para la venta.
- La antigüedad de un agente no puede decrecer.

Hay dos grupos de usuarios:

- El jefe que sólo está interesado en consultar el código, la ubicación y el distrito de los inmuebles de más de 500.000€.
- Los administrativos que pueden consultar toda la información.

Esquema conceptual



Esquema lógico

Propietario (Dni, Nombre, Teléfono)

CP: {Dni}

VNN: {Nombre} VNN: {Teléfono }

Agente (Dni_age, Nombre, Comisión, Años, Teléfono)

CP: {Dni_age}

VNN: {Nombre, Años, Teléfono, Comisión}

Edificio (Código, Precio, Ubicación, Distrito, Dueño, Dni_age)

CP: {Código}

VNN: {Precio, Ubicación, Distrito}

VNN: {Dueño}

CAj: {Dni_age} → Agente Modificación en cascada

CAj: {Dueño} → Propietarios, F(Dueño) → Dni

Modificación en cascada, Borrado en cascada

Restricciones de integridad

“Los agentes de menos de 3 años de antigüedad no pueden tener una comisión superior al 3%”

“Todo propietario tiene al menos un edificio”

“La antigüedad de un agente no puede decrecer”

Esquema lógico

```
CREATE TABLE Propietario
  (Dni INTEGER PRIMARY KEY, Nombre CHAR(40) NOT NULL,
  Teléfono CHAR(10) NOT NULL);

CREATE TABLE Agente
  (Dni_age INTEGER PRIMARY KEY,
  Nombre CHAR(40) NOT NULL,
  Comisión FLOAT NOT NULL CHECK (Comisión >0 AND Comisión <100),
  Años INTEGER NOT NULL, Teléfono CHAR(10) NOT NULL,
  CHECK NOT (Años<3 AND Comisión >3));

CREATE TABLE Edificio
  (Código CHAR(4) PRIMARY KEY,
  Precio FLOAT NOT NULL CHECK (precio >0),
  Ubicación CHAR(50) NOT NULL, Distrito CHAR(4) NOT NULL,
  Dueño INTEGER NOT NULL,
  Dni_age INTEGER REFERENCES Agente
    ON UPDATE CASCADE,
    ON DELETE NO ACTION,
  FOREIGN KEY (Dueño) REFERENCES Propietario (dni)
    ON UPDATE CASCADE,
    ON DELETE CASCADE);
```

Esquema lógico

```
CREATE ASSERTION no_prop_sin_edif CHECK
    NOT EXISTS(SELECT *
                FROM Propietario
                WHERE Dni NOT IN
                    (SELECT Dueño FROM Edificio))
    DEFERRABLE INITIALLY DEFERRED;
```

/*La restricción de que la antigüedad de un propietario no puede decrecer se controlará por programa o disparadores*/

Esquemas externos

ESQUEMA EXTERNO DEL JEFE

```
CREATE VIEW Más_de_5 AS  
    (SELECT E.Código, E.Ubicación, E.districto  
     FROM Edificio E  
     WHERE E.Precio > 500.000) WITH CHECK OPTION;
```

```
GRANT ALL ON Más_de_5 TO Jefe;  
GRANT ALL ON Agentes TO Jefe;  
GRANT ALL ON Propietarios TO Jefe;
```

ESQUEMA EXTERNO PAS

Todo el esquema lógico.

```
GRANT ALL ON Edificios TO los_demás;  
GRANT ALL ON Agentes TO los_demás;  
GRANT ALL ON Propietarios TO los_demás
```

Esquema interno o físico

- | | |
|-------------|---|
| Edificios | → Fichero disperso por <i>dni_age</i>
→ Índice B ⁺ sobre (<i>distrito+precio</i>) |
| Propietario | → Fichero disperso por <i>dni</i>
→ Índice B ⁺ sobre <i>nombre</i> |
| Agentes | → Fichero desordenado |

1.1 Esquemas y niveles de abstracción

Un SGBD que soporte la arquitectura de niveles debe:

- Permitir definir los distintos esquemas de la base de datos y, en cada uno de ellos, los aspectos que interesan de los datos y sus interrelaciones.
- Establecer las correspondencias entre los esquemas
- Aislar los esquemas, de manera que los cambios en un esquema no afecten a los esquemas de nivel superior y, en última instancia, a los programas de aplicación.



Independencia de datos

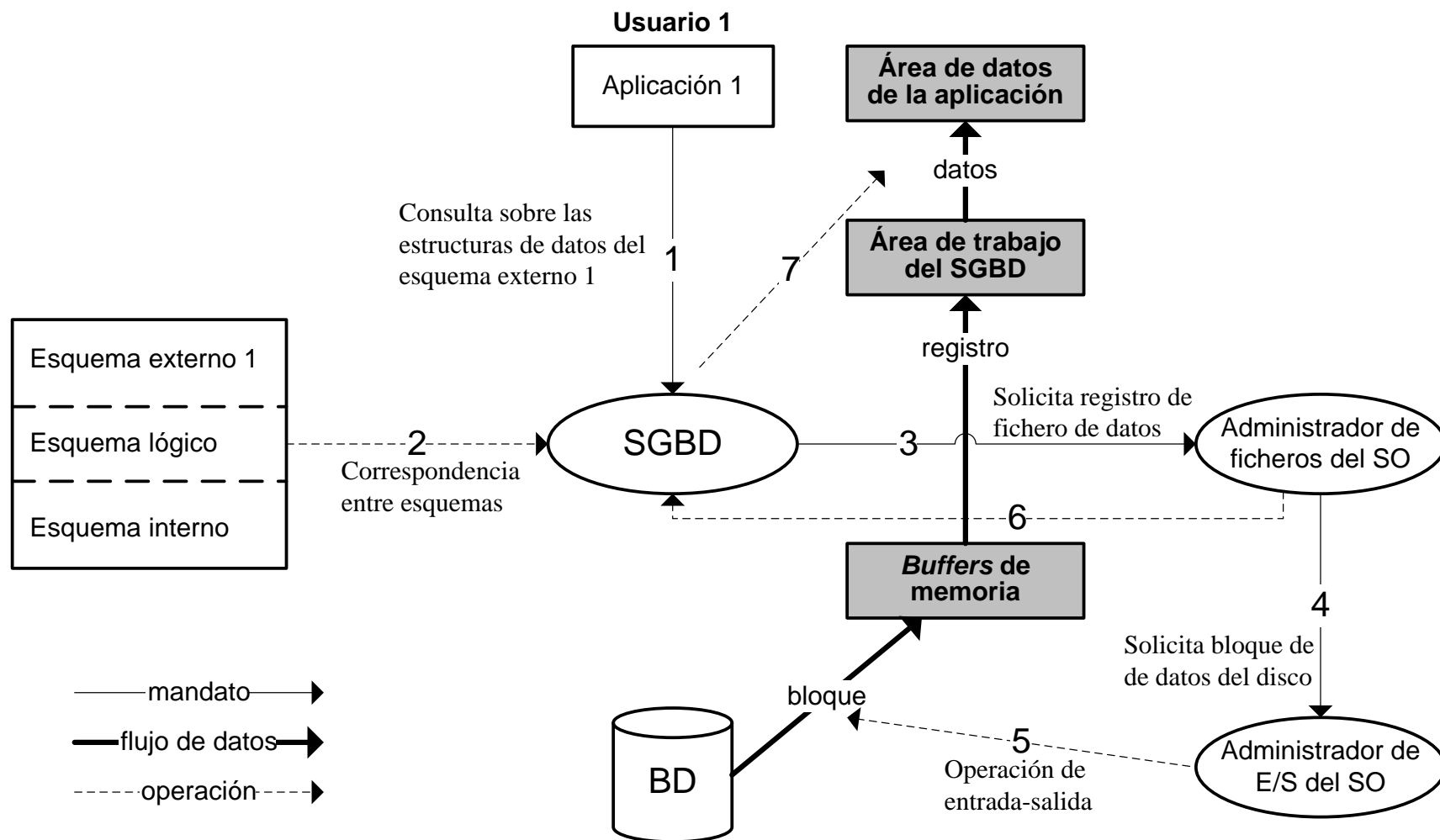
1.2 Funcionamiento Básico de un SGBD

Objetivos de las técnicas de BD	Funciones del SGBD	Componentes del SGBD
Descripción unificada de los datos e independiente de las aplicaciones	<p>Definición de la base de datos a varios niveles. Esquemas:</p> <ul style="list-style-type: none">• Esquema lógico (definición de las estructuras de datos)• Esquema interno (implementación de las estructuras del esquema lógico)• Esquemas externos (definición de estructuras derivadas) <p>Establecimiento de las correspondencias entre esquemas</p>	Lenguajes para la definición de esquemas y traductores asociados
Independencia de las aplicaciones respecto de la representación física de los datos		
Definición de vistas parciales de los datos para los distintos tipo de usuarios		

1.2 Funcionamiento Básico de un SGBD

Objetivos de las técnicas de BD	Funciones del SGBD	Componentes del SGBD
Gestión de los datos	<p>Manipulación: consulta y actualización.</p> <p>Gestión y administración de la base de datos.</p>	<p>Lenguajes de manipulación y traductores asociados.</p> <p>Herramientas para la:</p> <ul style="list-style-type: none">• Reestructuración• Simulación• Obtención de estadísticas• Impresión de datos
Integridad y Seguridad de los datos	<p>Control de:</p> <ul style="list-style-type: none">• Integridad semántica• Accesos concurrentes• Reconstrucción en caso de fallo• Seguridad	<p>Herramientas para:</p> <ul style="list-style-type: none">• Control de la integridad• Reconstrucción frente a fallos• Control de la seguridad

1.2 Funcionamiento Básico de un SGBD



1.2 Funcionamiento Básico de un SGBD

El jefe se pregunta:

¿código y ubicación de los edificios del distrito 05?

1. La aplicación interpreta la selección del jefe como:

SELECT código, ubicación

FROM más_de_5 **WHERE** distrito = '05';

2. El SGBD convierte la consulta del esquema externo al esquema lógico:

SELECT código, ubicación

FROM Edificios E

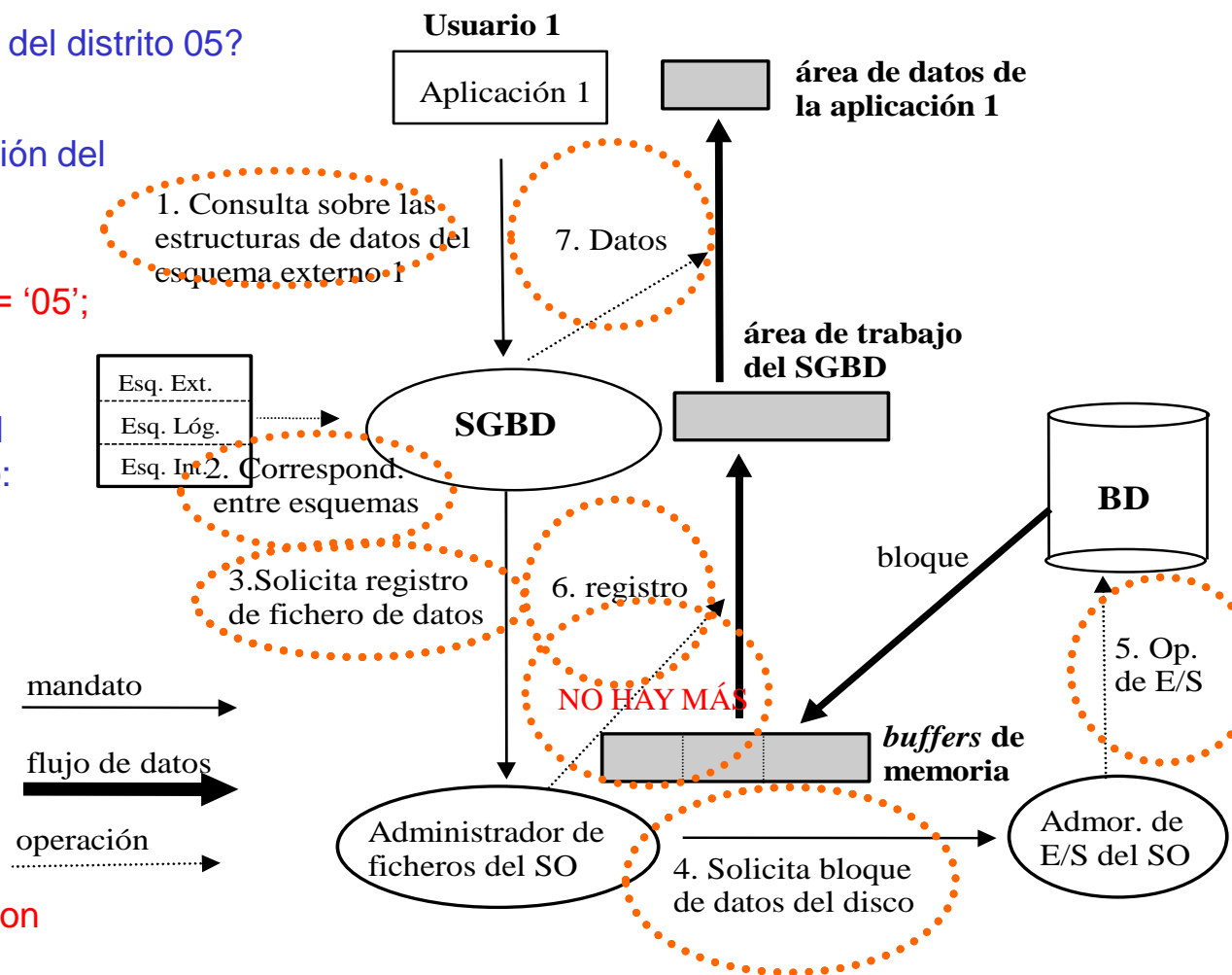
WHERE E.precio >= 500000

AND E.distrito = '05';

3,4,5,6. REPETIR:

“Leer usando el índice B+ sobre (distrito + precio) el primer registro con distrito = '05' y precio >= 500000:

HASTA QUE NO HAYA MÁS REGISTROS

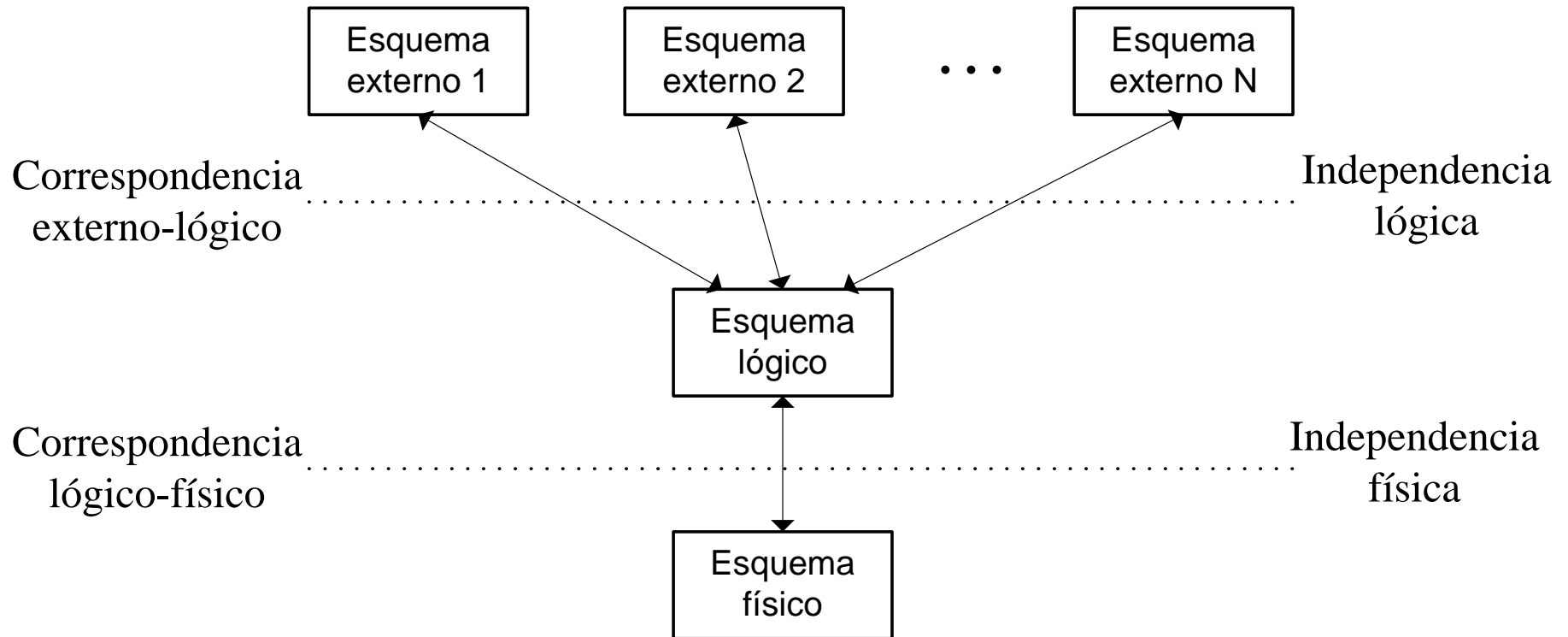


7. Eliminar los atributos que no se han solicitado

1.3 Independencia de datos

Propiedad que asegura que los programas de aplicación sean independientes de los cambios realizados en **datos que no usan** o en **detalles de representación física de los datos a los que acceden**.

1.3 Independencia de datos



1.3 Independencia de datos

- *Independencia lógica* entre el esquema lógico y los externos:
 - Los esquemas externos y los programas de aplicación no deben verse afectados por modificaciones del esquema lógico sobre datos que no usan.

EJEMPLO: Si al edificio se le añade un campo “Estado_de_conservación”, el esquema externo del jefe no cambia y la aplicación del jefe no se tiene que modificar.

1.3 Independencia de datos

- *Independencia física* entre el esquema interno y el lógico:
 - el esquema lógico no debe verse afectado por cambios en el esquema interno referentes a la implementación de las estructuras de datos, modos de acceso, tamaños de páginas, caminos de acceso, etc.

EJEMPLO: Si la relación Edificio se cambia de localización física (p.e. a un disco duro más rápido), el esquema lógico no se ve afectado.

Índice

1 Arquitectura ANSI/SPARC

1.1 Esquemas y niveles de abstracción

1.2 Funcionamiento básico de un sistema de gestión de bases de datos

1.3 Independencia de datos

2 Transacciones, integridad y concurrencia

2.1 Concepto de transacción

2.2 Integridad semántica

2.3 Control de accesos concurrentes

3 Recuperación y seguridad

3.1 Reconstrucción de la base de datos

3.2 Seguridad

2 Transacción, integridad y concurrencia

Objetivo de la tecnología de bases de datos



Calidad de la información:

“los datos deben estar estructurados reflejando adecuadamente los objetos, relaciones y las restricciones existentes en la parcela del mundo real que modela la base de datos”

- Representación de los objetos, relaciones y restricciones en el esquema de la base de datos.
- Cambios en la realidad → Actualizaciones de los usuarios.
- La información contenida en la base de datos debe preservar la definición del esquema.

2 Transacción, integridad y concurrencia

- Calidad de la información (perspectiva de la integridad):
 - SGBD debe asegurar que los datos se almacenan correctamente.
 - SGBD debe asegurar que las actualizaciones de los usuarios sobre la base de datos se ejecutan correctamente y que se hacen permanentes.

2 Transacción, integridad y concurrencia

- Herramientas del SGBD orientadas a la integridad:
 - Comprobar (frente a actualizaciones) las restricciones de integridad del esquema.
 - Controlar la ejecución correcta de las actualizaciones (entorno concurrente).
 - Recuperar (reconstruir) la base de datos en caso de pérdidas o accidentes.

2.1 Concepto de transacción

- La integridad de la base de datos se ve en peligro generalmente por las operaciones de acceso de las aplicaciones.
- Las operaciones de acceso a una base de datos se organizan en transacciones.

TRANSACCIÓN { Secuencia de operaciones de acceso a la base de datos que constituyen una unidad lógica de ejecución

2.1 Concepto de transacción

- Operaciones que pueden aparecer en una transacción:
 - Operaciones de consulta y actualización de los datos.
 - Principio.
 - Anulación (*rollback de usuario*).
 - Confirmación (*commit de usuario*): la transacción puede acabar con éxito (*commit de SGBD*) o anulada (*rollback del SGBD*)

2.1 Concepto de transacción

Propiedades que deben cumplir el correcto procesamiento de transacciones:

- **Atomicidad**: una transacción es una unidad atómica de ejecución (o se ejecutan todas sus operaciones o ninguna).
- **Consistencia**: la transacción debe dar lugar a un estado de la base de datos consistente (se cumplen todas las restricciones de integridad).
- **Aislamiento**: una transacción debe ejecutarse como si estuviera ejecutándose ella sola y no concurrentemente con otras transacciones.
- **Persistencia**: la confirmación implica la grabación de los cambios introducidos en la base de datos, de forma que no se puedan perder por fallo del sistema o de otras transacciones.

ACID= Atomicity+Consistency+Isolation+Durability

2.2 Integridad semántica

Restricción de integridad  propiedad del mundo real del cual la BD es una representación

- Las restricciones se definen en el esquema lógico y el SGBD debe velar por su cumplimiento.
- La comprobación se realiza cuando la base de datos cambia (se ejecuta una transacción).
- Las restricciones que no se incluyen en el esquema de la base de datos deben ser comprobadas por los programas de las aplicaciones.

2.2 Integridad semántica

- Tipos de restricciones de integridad:
 - **Estáticas:** se deben cumplir en cada estado de la base de datos

EJEMPLO:

- “Los agentes de menos de 3 años de antigüedad no pueden tener una comisión superior al 3%”
- “Todo propietario tiene al menos un edificio”

- **De transición:** se deben cumplir en dos estados consecutivos.

EJEMPLO:

- “La antigüedad de un agente no puede decrecer”

2.2 Integridad semántica

- Restricciones en el SQL:

- restricciones estáticas:

- ◇ sobre valores posibles de los atributos

- EJEMPLO: Precio NUMBER(10,2) CHECK (Precio >0)

- ◇ sobre atributos

- EJEMPLO: Precio NOT NULL

- ◇ sobre relaciones

- EJEMPLO: PRIMARY KEY

- se puede especificar cuándo se comprueba la restricción:

- ◇ después de cada operación individual (IMMEDIATE)

- ◇ al final de la transacción (DEFERRED).

- se pueden especificar acciones restauradoras (integridad referencial).

2.3 Control de accesos concurrentes

Para mantener la integridad de la base de datos el SGBD debe controlar los accesos concurrentes:

- Evitando que los resultados de la ejecución de varios procesos (usuarios o programas) simultáneamente puedan llevar a resultados **incorrectos**, **incoherentes** o que se **pierdan**.
- Problemas por interferencia de accesos concurrentes:
 - a) pérdida de actualizaciones,
 - b) obtención de información incoherente correspondiente a varios estados válidos de la base de datos, y
 - c) lectura de datos actualizados (no confirmados) que han sido sometidos a cambios que todavía pueden ser anulados.

2.3 Control de accesos concurrentes

a) Pérdida de las actualizaciones

T1: subir 500€ el precio del edificio E123

T2: subir 800€ el precio del edificio E123

Edificio		
código	precio	...
E123	150.000	...
E505	800.000	...
...

Edificio		
código	precio	...
E123	150.800	...
E505	800.000	...
...

Tiempo	T1	T2
t1	leer(E123,precio)	
t2		leer(E123,precio)
t3	precio ← precio+500	
t4		precio ← precio+800
t5	escribir(E123,precio)	
t6		escribir(E123,precio)

La actualización de T1 se ha perdido

2.3 Control de accesos concurrentes

b) Obtención de información incoherente

T1: Obtención del código de cada agentes con la cantidad de edificios que gestionan.

T2: Cambiar el edificio E123 del agente a1 al agente a2.

Edificio		
código	dni_age	...
E123	a1	...
E505	a1	...
E777	a2	...

Edificio		
código	dni_age	...
E123	a2	...
E505	a1	...
E777	a2	...

Tiempo	T1	T2
t1	calcular cuántos edificios tiene a1 2	
t2		cambiar el agente del edificio E123 a a2
t3	calcular cuántos edificios tiene a2 2	

La información obtenida no se corresponde con ninguno de los dos estados de la base de datos

2.3 Control de accesos concurrentes

c) Lectura de datos actualizados sin confirmar

T1: Sube el precio de edificio E123 en 500€.

T2: Lee el precio del edificio E123 y lo usa.

Edificio		
código	precio	...
E123	150.000	...
E505	800.000	...
...

Edificio		
código	precio	...
E123	150.500	...
E505	800.000	...
...

Tiempo	T1	T2
t1	leer(E123,precio)	
t2	precio ← precio+500	
t3	escribe(E123,precio)	
t4		leer(E123,precio)
t5		usa el precio leído
t6		confirma
t7	anula	

La información obtenida no se corresponde con ninguno de los estados de la base de datos

2.3 Control de accesos concurrentes

Técnicas:

- Reserva de Ocurrencias de Datos (Locks)
 - Ejemplos a) y c) se reserva un registro (una fila de Edificio).
 - Ejemplo b) se reservan todos (todas las filas de Edificio).
- Necesidad de controlar bloqueos (*deadlocks*).
- Otras soluciones (para el ejemplo c): anulación en cascada o aislamiento de transacciones.

Índice

1 Arquitectura ANSI/SPARC

1.1 Esquemas y niveles de abstracción

1.2 Funcionamiento básico de un sistema de gestión de bases de datos

1.3 Independencia de datos

2 Transacciones, integridad y concurrencia

2.1 Concepto de transacción

2.2 Integridad semántica

2.3 Control de accesos concurrentes

3 Recuperación y seguridad

3.1 Reconstrucción de la base de datos

3.2 Seguridad

3 Recuperación y seguridad

Hay dos aspectos irrenunciables en la tecnología actual de bases de datos:

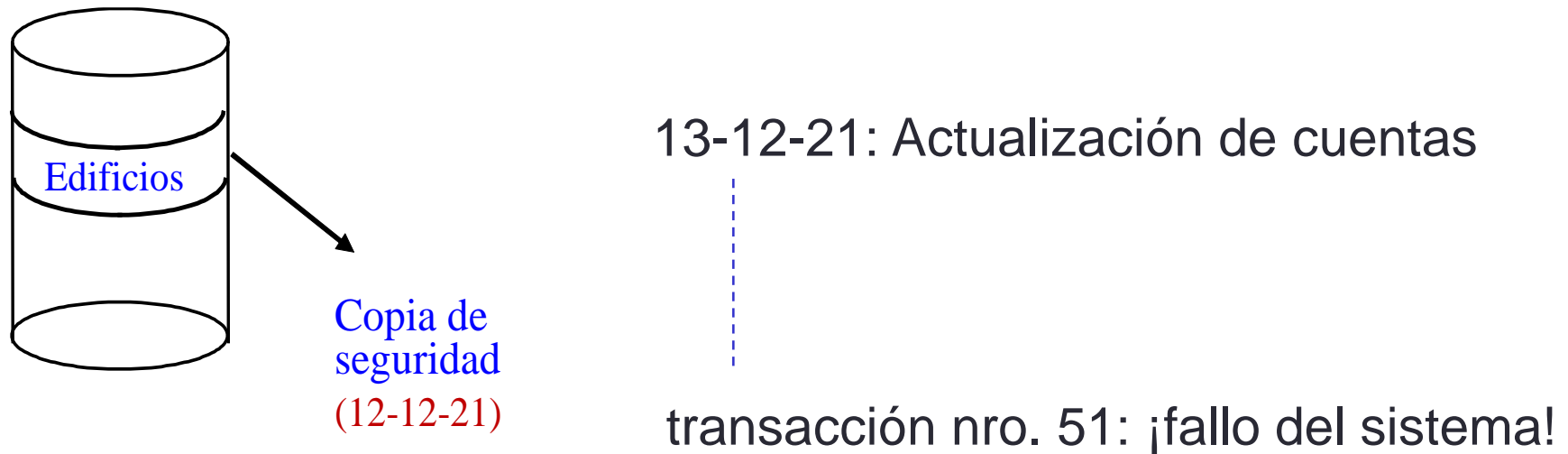
- Recuperación (parte de la integridad, pero no desde el punto de vista de la consistencia): una base de datos ha de poderse recuperar ante prácticamente cualquier tipo de fallo.
- Seguridad: una base de datos no puede tener accesos no autorizados.

3.1 Reconstrucción de la base de datos

Las propiedades de atomicidad y persistencia de una transacción obligan al SGBD a asegurar que:

- Si se confirma, los cambios efectuados se graban en la base de datos y no se pierdan.
- Si se anula, los cambios efectuados sobre la base de datos se deshacen.

3.1 Reconstrucción de la base de datos



Procedimiento de recuperación:

- sustituir el fichero de *Edificio* por su copia de seguridad

Efecto negativo:

- se han perdido las actualizaciones de 50 transacciones

3.1 Reconstrucción de la base de datos

Las copias de seguridad, por sí mismas, no son la solución al problema de recuperación.

- El aumento de la frecuencia de copias de seguridad no es una solución viable.

La tecnología de bases de datos proporciona técnicas mucho más eficientes y robustas de cara a la reconstrucción de la base de datos.

La pérdida de datos “confirmados” es inadmisibile con la tecnología actual.

3.1 Reconstrucción de la base de datos

Causas del fallo de una transacción:

- Locales a la transacción (**funcionamiento del sistema normal**):
 - Errores en la transacción (acceso a la base de datos incorrecto, cálculos fallidos, etc.).
 - Excepciones (violación de la integridad, de la seguridad, etc.).
 - Control de la concurrencia (estado de bloqueo entre dos transacciones).
 - Decisiones humanas (por programa o explícitas).

3.1 Reconstrucción de la base de datos

Causas del fallo de una transacción:

- Externas a la transacción (**errores del sistema**)
 - Fallos del sistema con pérdida de la memoria principal.
 - Fallos del sistema de almacenamiento con pérdida de la memoria secundaria.

3.1 Reconstrucción de la base de datos

Pérdidas de memoria principal:

- En el espacio de tiempo entre la confirmación de una transacción y la grabación de sus campos en memoria secundaria.
- La transacción está confirmada y sus cambios están memoria principal.
- En dicho intervalo se produce un fallo con pérdida de memoria principal y los cambios se pierden.

3.1 Reconstrucción de la base de datos

Pérdidas de memoria secundaria:

- Transacción confirmada cuyos cambios están grabados en la base de datos.
- Fallo en la memoria secundaria y estos cambios se pierden.

3.1 Reconstrucción... : fallo con pérdida MP

Objetivos:

- Recuperar transacciones confirmadas que no han sido grabadas.
- Anular transacciones que han fallado.

Herramientas:

- Fichero diario (*log o journal*).
- Puntos de control (*checkpoints*).

3.1 Reconstrucción... : fallo con pérdida MP

Fichero diario:

- Registrar las operaciones de actualización de las transacciones.
- Se almacena en disco para evitar la desaparición por un fallo del sistema.
- Se graba periódicamente a una unidad de almacenamiento masiva.

3.1 Reconstrucción... : fallo con pérdida MP

Puntos de verificación

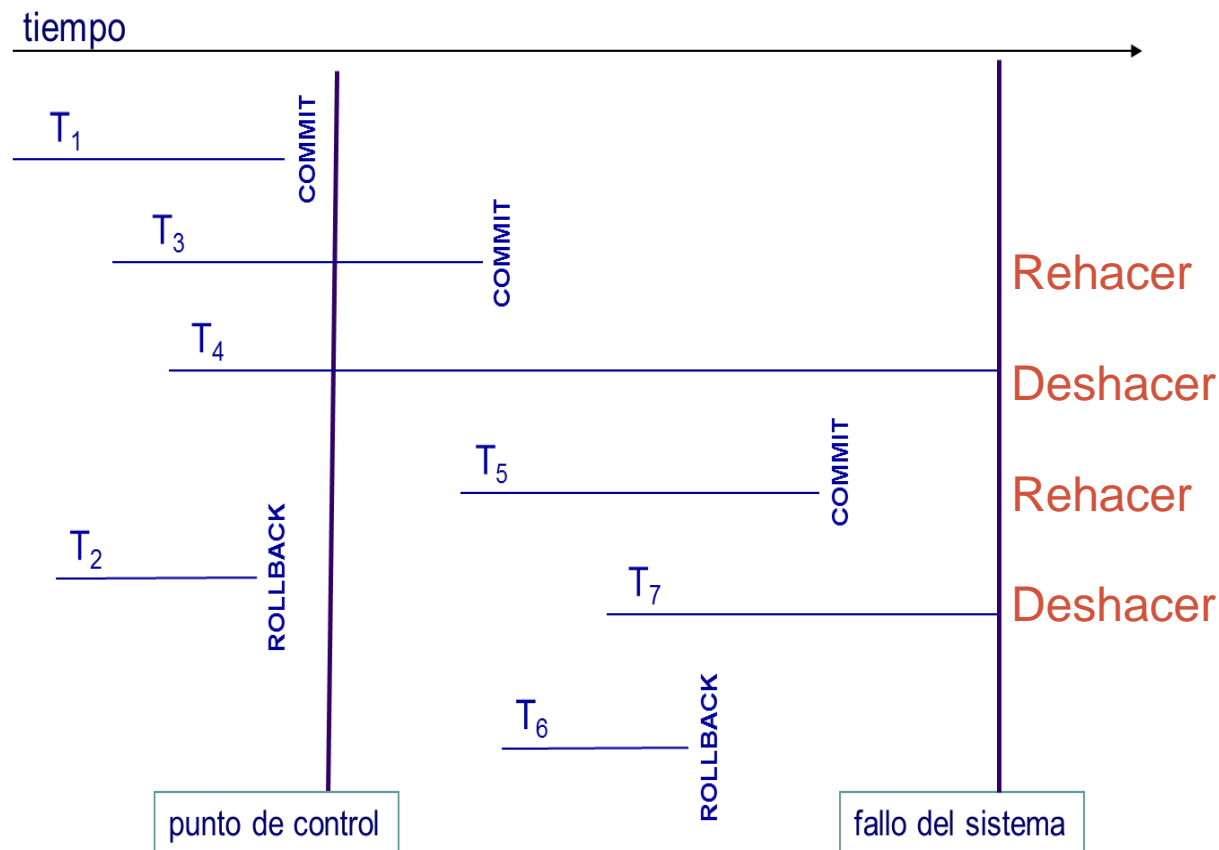


Se graban en el diario
periódicamente

- Suspenden temporalmente la ejecución de transacciones.
- Anotar en el diario el punto de verificación.
- Forzar la grabación de todas las actualizaciones de las transacciones confirmadas (copiar los *buffers* de MP a disco).
- Reanudar la ejecución de las transacciones suspendidas.

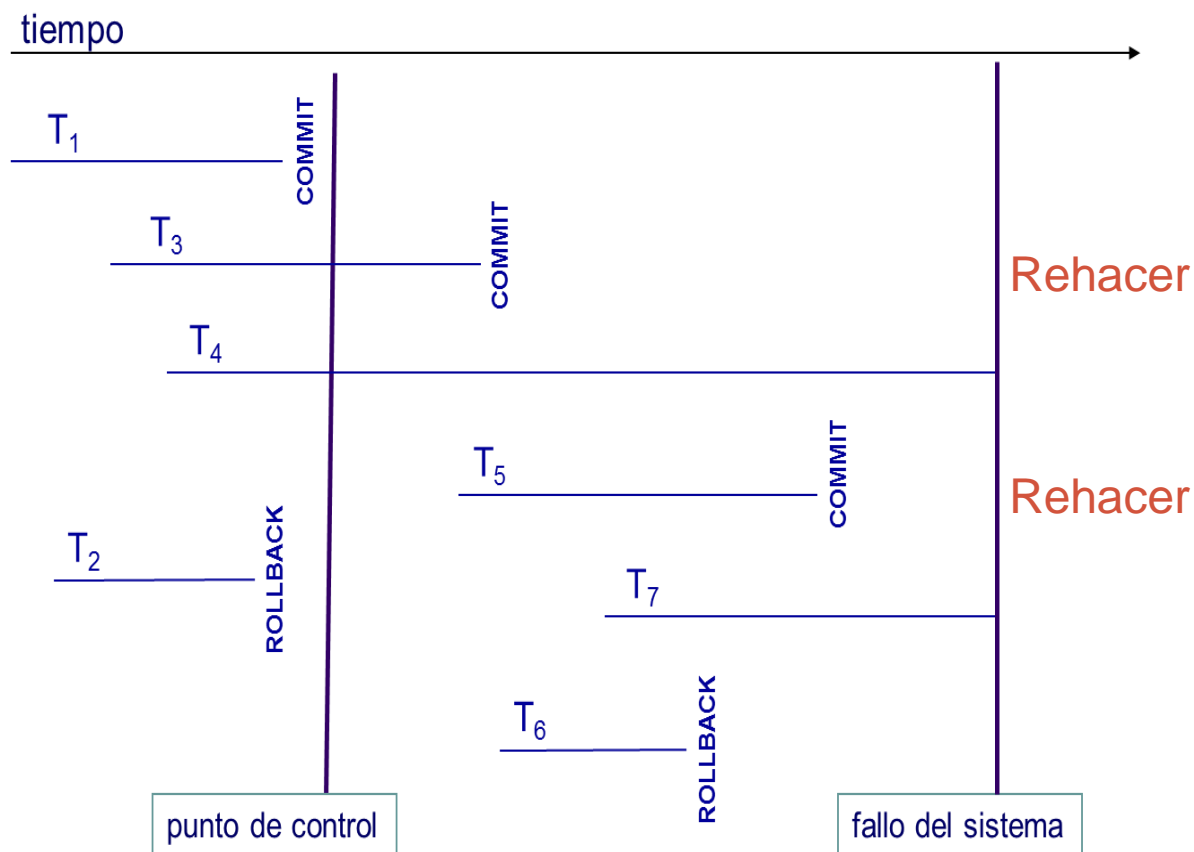
3.1 Reconstrucción... : fallo con pérdida MP

ACTUALIZACIÓN INMEDIATA



3.1 Reconstrucción... : fallo con pérdida MP

ACTUALIZACIÓN DIFERIDA



3.1 Reconstrucción... : fallo con pérdida MS

Reconstrucción frente a fallos del sistema de almacenamiento

- Pérdida de memoria secundaria.
- Base de datos puede estar dañada total o parcialmente.
- Técnica: reconstruir la base de datos a partir de
 - La copia de seguridad más reciente.
 - A partir del instante de la copia utilizar el diario para rehacer las operaciones realizadas por las transacciones confirmadas.

3.2 Seguridad

Objetivo:

Sólo pueden acceder a la información las personas y procesos autorizados y en la forma autorizada.

3.2 Seguridad

Técnicas:

- Identificación del usuario.
- Determinación de los accesos permitidos:
 - Modos {

 - Lista de autorizaciones (objeto y operaciones permitidas) por usuario.
 - Niveles de autorización (menos flexible).
- Gestión de autorizaciones transferibles: traspaso de autorizaciones de un usuario a otro.

3.2 Seguridad

Gestión de autorizaciones transferibles:

- Conocimiento de las autorizaciones de acceso de cada usuario (cuáles son transferibles a terceros y cuáles no).
- Transferencia de una autorización de un usuario a otro (en modo transferible o no).
- Revocación posterior de una autorización de acceso:
 - Si se otorgó en modo transferible, revocación de las autorizaciones que partieron de ella.
- Revocación independiente de una autorización de acceso otorgada de forma múltiple.

3.2 Seguridad

```
GRANT { ALL
      SELECT
      INSERT [(columna1,..., columnan)]
      DELETE
      UPDATE [(columna1,..., columnan)] }
ON OBJETO TO {usuario1,..., usuarion | PUBLIC}
[WITH GRANT OPTION]
```

3.2 Seguridad

REVOKE [GRANT OPTION FOR]

{ ALL

SELECT

INSERT [(*columna*₁,..., *columna*_{*n*})]

DELETE

UPDATE [(*columna*₁,..., *columna*_{*n*})] }

ON OBJETO TO {*usuario*₁,..., *usuario*_{*n*} | PUBLIC}