



6. Bases de dades distribuïdes

NF1. Conceptes bàsics

UF1: Introducció a les bases de dades

Desenvolupament d'Aplicacions Multiplataforma

M02 – Bases de dades. Versió 1.0

© M^a Carmen Brito Ruiz

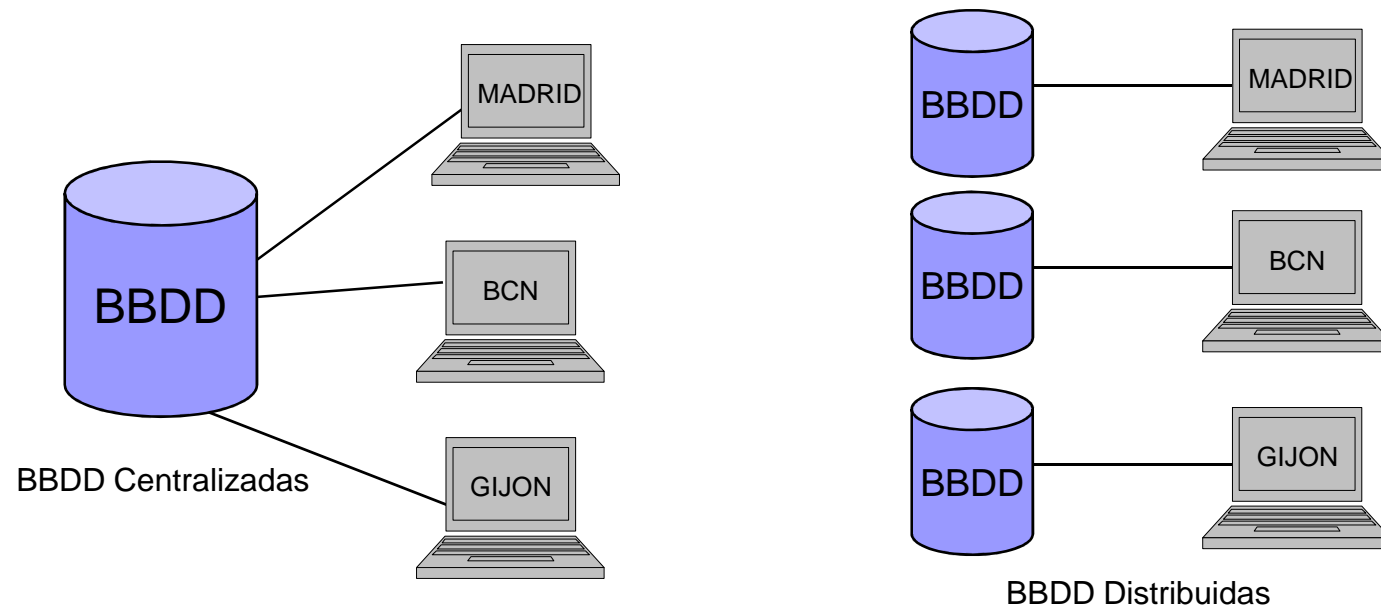
- 6.1. Introducció a las bases de datos distribuidas
- 6.2. Componentes a las base de datos distribuidas
- 6.3. Tipología de las BDD y distribución de los datos
- 6.4. Procesamiento distribuidos de consulta y optimización de consultas
- 6.5. Bloqueos y concurrencia. Transacciones distribuidas
- 6.6. Seguridad y recuperación de la información en las bases de datos distribuidas
- 6.7. Arquitectura de los sistemas distribuidos
- 6.8. Diseño y gestión de bases de datos distribuidas

6.1. Introducció a las base de datos distribuidas

6.1.1. BBDD Centralizadas vs. BBDD Distribuidas

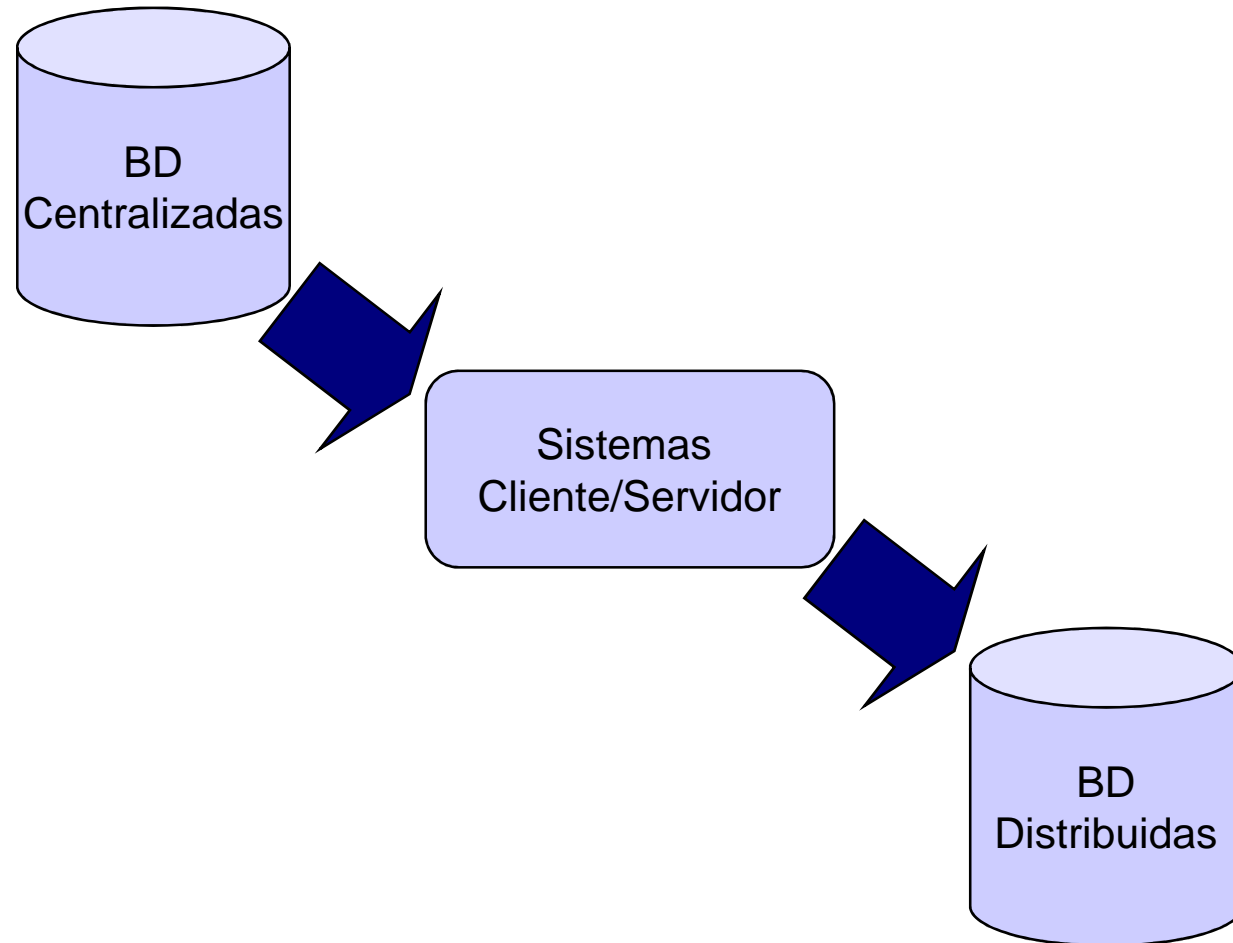
Inicialmente la información se almacenaba de forma centralizada, pero cada vez había más información y la necesidad de tratar dicha información iba en aumento.

Estos problemas y otros, impulsaron la creación de almacenar la información de forma distribuida: combinación de las redes de comunicación y las bases de datos.



Bases de dades distribuïdes

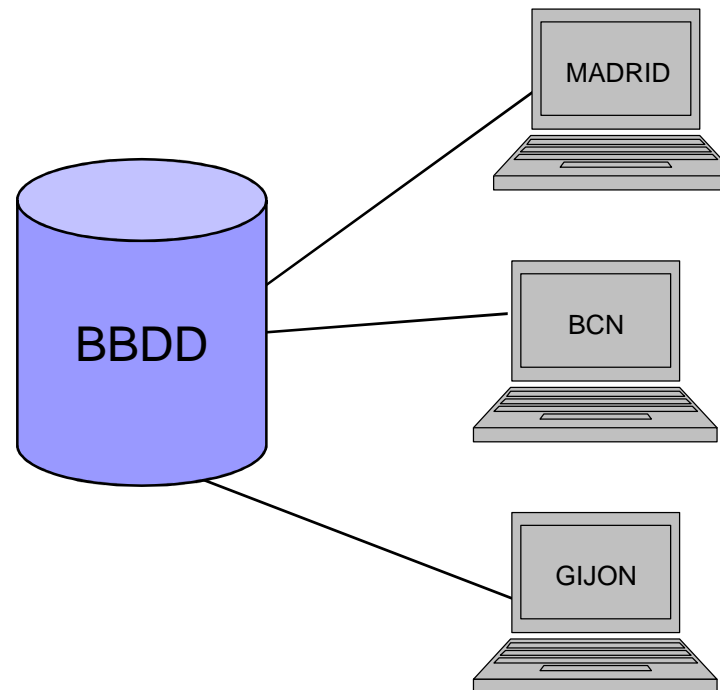
Evolución



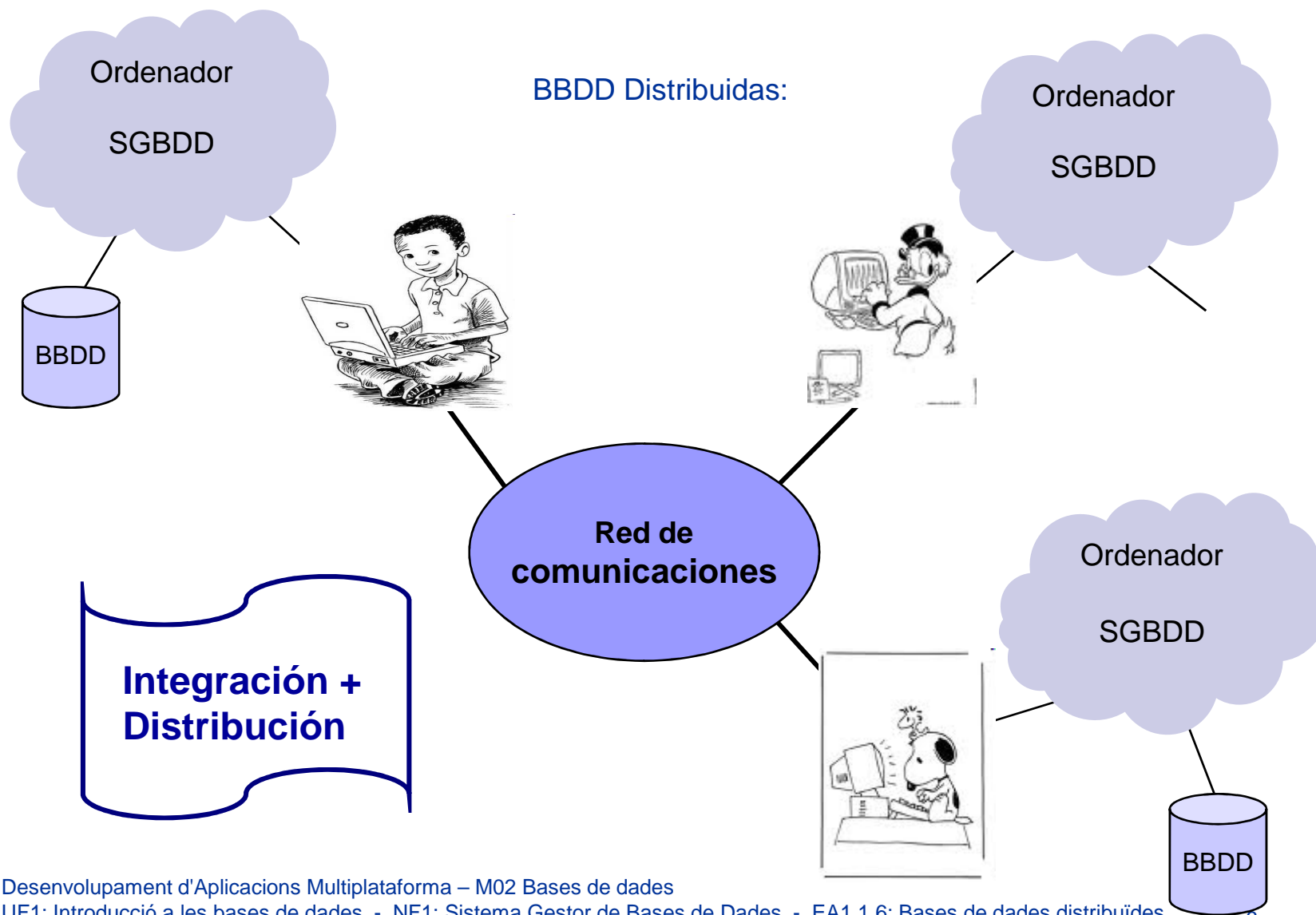
BBDD Centralizadas:

Sólo hay una base de datos central y los servidores regionales y/o locales son clientes que 'atacan' al servidor central: Arquitectura cliente/servidor.

Problemas: sobrecarga del servidor y sensibles a la conectividad (líneas dedicadas).



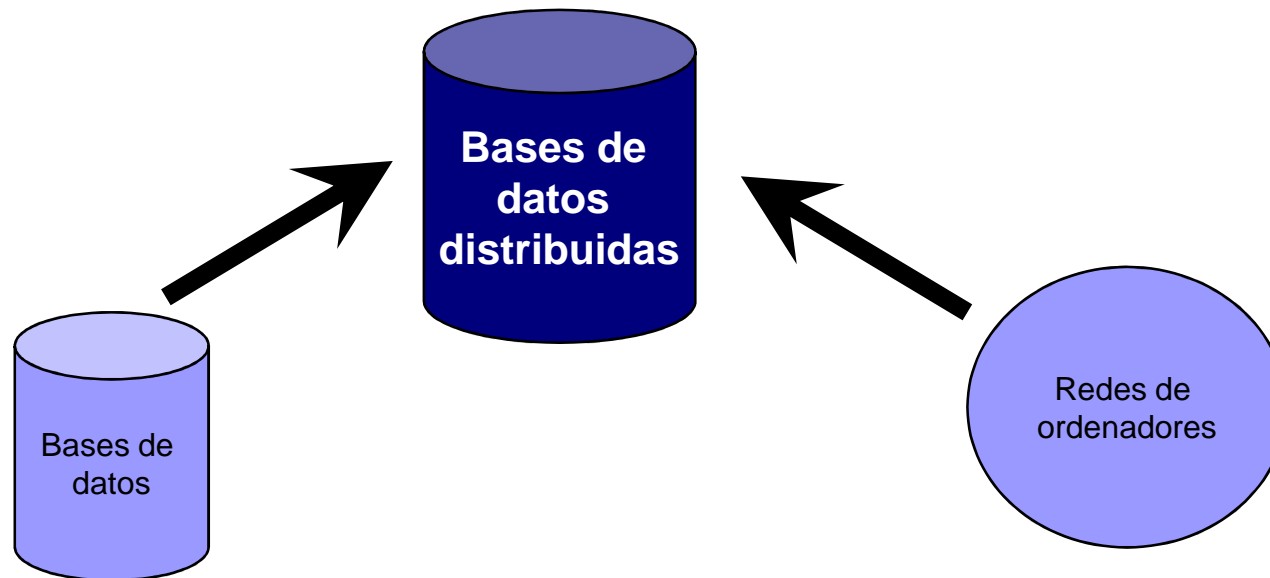
Bases de dades distribuïdes



6.1.2. Conceptos básicos

Bases de Datos distribuida (BDD)

Una base de datos distribuida, es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas entre diferentes sitios interconectados por una red de comunicaciones.



Sistemas de Gestión de Bases de Datos distribuida (SGBDD)

Es el software que se encarga del manejo de la BDD y proporciona un mecanismo de acceso que hace que la distribución sea transparente a los usuarios.

Sistema de Bases de Datos distribuida (SBDD):

Sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones de tal forma que, un usuario en cualquier sitio puede acceder los datos en cualquier parte de la red exactamente como si los datos estuvieran siendo accedidos de forma local.

$$\text{BDD} + \text{SGBDD} = \text{SBDD}$$

6.1.3. Características de las BDD

- En un sistema distribuido de bases de datos se almacenan en varios ordenadores, por tanto, los datos están físicamente distribuidos en más de un ordenador (diferentes sedes o nodos).
- Las sedes o nodos han de estar interconectados mediante una red y por tanto, cada sede es un nodo de la red.
- En una operación se puede acceder (recuperar o actualizar) datos que están en más de un nodo, por tanto, se accede a datos locales o remotos.
- Los datos han de estar lógicamente integrados tanto en local como en remoto: esquema lógico global y único.
- El acceso a más de un nodo es transparente al usuario: transparencia de distribución para el usuario.

6.1.4. Ventajas e inconvenientes de BDD

Ventajas:

- *Descentralización y Autonomía local*
- *Económicas*
- *Mejora de rendimiento*
- *Mejora de fiabilidad y disponibilidad*
- *Crecimiento*
- *Flexibilidad*
- *Disponibilidad*
- *Control de Concurrencia*

Desventajas:

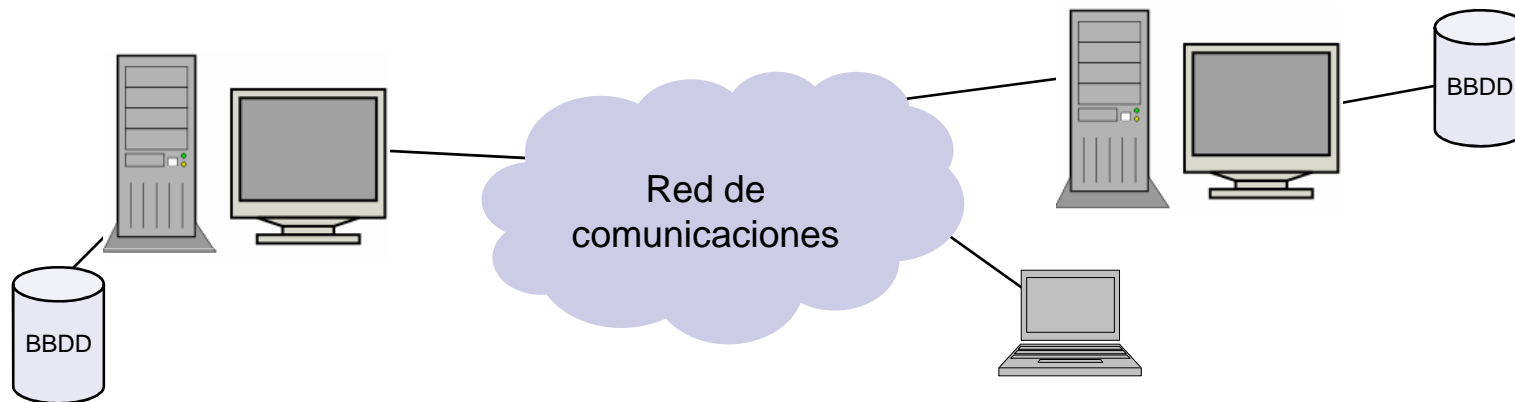
- *Complejidad del sistema:*
 - software es más caro, por su complejidad.
 - problemas de sincronización.
 - detección de caídas de nodos.
- *Mayor posibilidad de errores*
- *Mayor tiempo extra de procesamiento*
- *Dificultad del diseño:* fases adicionales.
- *Productos comerciales poco maduros*
- *Las funciones de la administración son compleja*
- *Dificultad al cambio:*
- *Se necesita un personal especializado.*

6.2. Componentes a las base de datos distribuidas

- Hardware
- Red de comunicaciones
- Nodos
- Bases de datos locales
- Diccionario o directorio global
- Software:
 - Administrador de transacciones distribuidas (DTM)
 - Sistema Manejador de base de datos (SBDD)
 - Sistemas Gestores de bases de datos distribuidos (SGBDD)

➤ Hardware – Red de comunicaciones – Nodos - Bases de datos locales:

El hardware utilizado no difiere mucho del hardware utilizado en un servidor normal: servidores y la red de comunicaciones.



En un primer momento se pensó que debía ser un hardware especial, con el fin de que fuera más rápido y más eficiente, pero se fue comprobando que el descentralizar y no compartir nada (*shared-nothing*) era más barato y eficaz.

➤ Software:

- Sistema Manejador de base de datos (SBDD o DBM “*Distributed Base Management*”)

Es un programa que procesa cierta porción de la base de datos distribuida. Se encarga de recuperar y actualizar datos del usuario y generales de acuerdo con los comandos recibidos de los Administradores de transacciones distribuidas (DTM).

- Administrador de transacciones distribuidas (ATD o DTM “*Distributed Transaction Management*”)

Este es un programa que recibe las solicitudes de procesamiento de los programas de consulta o transacciones y las traduce en acciones para los administradores de la base de datos. Se encargan de coordinar y controlar estas acciones. El DTM puede ser propietario o desarrollado en casa.

- Sistema Gestor de Base de Datos Distribuida (SGBDD o DDBMS “*Distributed Data Base Management System*”)

Es un conjunto de programas que operan en diversos ordenadores, estos programas pueden ser subsistemas de un único SGBDD de un fabricante o podría consistir de una colección de programas de diferentes fuentes.

Este sistema está formado por las transacciones y los administradores de la base de datos distribuidos.

Es decir, el SGBDD es el encargado de crear y mantener una BDD y cuenta con funciones de un SGBD centralizado, además de otras adicionales.

6.3. Tipología de las BDD y distribución de los datos

Tipología de las BDD

- Según distribución:
- Según grado de homogeneidad:
- Según el grado de autonomía:

Distribución de los datos

- Centralización
- Réplica: copiar en diferentes nodos.
- Fragmentación: dividir en fragmentos y se almacena en diferentes nodos.
- Réplica y fragmentación: dividir en fragmentos que se replican en diferentes nodos.

6.3. Tipología de las BDD y distribución de los datos

Tipología de las BDD

- Según distribución:
 - datos distribuidos
 - datos centralizados
- Según grado de homogeneidad:
 - SGBDD homogéneos: todos los SGBD locales son iguales.
 - SGBDD heterogéneos: los SGBD locales son diferentes.
- Según el grado de autonomía:
 - Sistemas compuestos (estrechamente integrados): no tienen ninguna autonomía funcional con el SGBD local, el acceso a través del SGBDD.
 - Sistemas Multi-Base de Datos (autonomía total): SGBD locales autónomos e independientes.
 - Sistemas federados (semiautónomos): SGBD locales autónomos y permite el acceso del SGBDD a partes de la base de datos.

Distribución de los datos

- Centralización
- Réplica: copiar en diferentes nodos.
- Fragmentación: dividir en fragmentos y se almacena en diferentes nodos.
- Réplica y fragmentación: dividir en fragmentos que se replican en diferentes nodos.

➤ Centralizada:

Es similar al modelo de Cliente/Servidor, donde BDD está centralizada en un lugar y los usuarios están distribuidos. Su ventaja es tener procesamiento distribuido, pues en la disponibilidad y fiabilidad de los datos no se gana nada.

➤ Replicada:

El esquema de BDD de replicación consiste en que cada nodo debe tener su copia completa de la base de datos. Es decir, el sistema conserva varias copias idénticas de una tabla.

Cabe destacar, el alto coste de:

- almacenamiento de la información.
- alto coste de escritura, pues la actualización de los datos debe ser realizada en todas las copias

Se usa en sistemas en el que se va a escribir pocas veces y leer muchas, y dónde la disponibilidad y fiabilidad de los datos sea de máxima importancia.

➤ **Particionada o fragmentada:**

En este caso solo hay una copia de cada elemento, pero la información está distribuida a través de los nodos. En cada nodo se aloja uno o más fragmentos disjuntos de la base de datos. Como los fragmentos no se replican esto disminuye el costo de almacenamiento, pero también sacrifica la disponibilidad y fiabilidad de los datos. Algo que se debe tomar en cuenta cuando se desea implementar este modelo es la granularidad de la fragmentación.

Hay tres tipos de fragmentación: Horizontal, Vertical y Mixto.

Fragmentación Horizontal:

Consiste en dividir una tabla en subconjuntos. Los fragmentos son subconjuntos de la tabla y se definen con una operación de selección; su reconstrucción se efectúa con una operación de unión.

Cada fragmento se sitúa en un nodo.

Se puede dar el caso de combinar fragmentación y replicación.

Ejemplo: Tabla de alumnos con nota final de estudios (T)

| DNI | Nombre y apellidos | Centro Estudio | Nota Final | Promociona |
|-----------|---------------------|----------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 232445445 | Rosa Garrido Ruiz | ET1 | 7.5 | Si |
| 349877589 | Manel Bel Pol | ET1 | 5.3 | Si |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |
| ... | ... | ... | ... | ... |

Fragmento de ET1: $\Omega_{\text{centroestudio}=\text{"ET1"}} (T)$

| DNI | Nombre y apellidos | CentroEstudio | Nota Final | Promociona |
|-----------|--------------------|---------------|------------|------------|
| 232445445 | Rosa Garrido Ruiz | ET1 | 7.5 | Si |
| 349877589 | Manel Bel Pol | ET1 | 5.3 | Si |

Fragmento de ET2: $\Omega_{\text{centroestudio}=\text{"ET2"}} (T)$

| DNI | Nombre y apellidos | CentroEstudio | Nota Final | Promociona |
|-----------|---------------------|---------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |

La fragmentación horizontal es similar a una consultar con una restricción.

Fragmentación Vertical:

Consiste en dividir una tabla en subconjuntos. Los fragmentos son subconjuntos de la tabla y se definen con una operación de proyección. Su reconstrucción se efectúa con operaciones de join de los fragmentos.

En este caso cada fragmento ha de incluir la clave primaria (campo que identifica de manera única un registro) de la tabla.

Se puede dar el caso de combinar fragmentación y replicación.

Ejemplo: Tabla de alumnos con nota final de estudios (T)

| DNI | Nombre y apellidos | Centro Estudio | Nota Final | Promociona |
|-----------|---------------------|----------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 232445445 | Rosa Garrido Ruiz | ET1 | 7.5 | Si |
| 349877589 | Manel Bel Pol | ET1 | 5.3 | Si |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |
| ... | ... | ... | ... | ... |

Datos para el equipo docente:

$\Omega_{\text{Nombreyapellidos, notafinal, promociona}}$ (T)

| Nombre y apellidos | Nota Final | Promociona |
|---------------------|------------|------------|
| Josep García García | 7.6 | Si |
| Anna Roig Pérez | 4.5 | No |
| Rosa Garrido Ruiz | 7.5 | Si |
| Manel Bel Pol | 5.3 | Si |
| Maria Roca Sola | 9.4 | Si |

Datos académicos para generar títulos: $\Omega_{\text{DNI, Nombreyapellidos, centroestudio, notafinal}}$ (T)

| DNI | Nombre y apellidos | Centro Estudio | Nota Final | Promociona |
|-----------|---------------------|----------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 232445445 | Rosa Garrido Ruiz | ET1 | 7.5 | Si |
| 349877589 | Manel Bel Pol | ET1 | 5.3 | Si |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |

La fragmentación vertical son subconjunto de los atributos de la tabla con sus valores, por tanto, es una proyección de la tabla.

Fragmentación Mixta:

Los fragmentos es una combinación de restringir y proyectar una tabla.

Ejemplo: Tabla de alumnos con nota final de estudios (T)

| DNI | Nombre y apellidos | Centro Estudio | Nota Final | Promociona |
|-----------|---------------------|----------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 232445445 | Rosa Garrido Ruiz | ET1 | 7.5 | Si |
| 349877589 | Manel Bel Pol | ET1 | 5.3 | Si |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |
| ... | ... | ... | ... | ... |

Alumnos del centro ET2, visualizando DNI, Nombre y apellidos y Nota Final:

1) $\Omega_{\text{CentroEstudio}=\text{"ET2"}} (T)$

| DNI | Nombre y apellidos | Centro Estudio | Nota Final | Promociona |
|-----------|---------------------|----------------|------------|------------|
| 123456789 | Josep García García | ET2 | 7.6 | Si |
| 324543245 | Anna Roig Pérez | ET2 | 4.5 | No |
| 342344535 | Maria Roca Sola | ET2 | 9.4 | Si |

2) $\Omega_{\text{DNI}, \text{Nombreyapellidos}, \text{notafinal}} (T)$

| DNI | Nombre y apellidos | Nota Final |
|-----------|---------------------|------------|
| 123456789 | Josep García García | 7.6 |
| 324543245 | Anna Roig Pérez | 4.5 |
| 342344535 | Maria Roca Sola | 9.4 |

Ventaja de la fragmentación:

las consultas SQL también se fragmentan por lo que su procesamiento es en paralelo y más eficiente. Aunque hay operaciones especiales como el producto (juntar datos de tablas) que involucran varios fragmentos de la BDD.

➤ Réplica y fragmentación:

Es la división en fragmentos que se replican en diferentes nodos.

Las técnicas de réplica y fragmentación se pueden aplicar sucesivamente misma relación de partida. Un fragmento se puede replicar y a su vez ser fragmentada, para luego replicar alguno de esos fragmentos.

6.4. Procesamiento distribuidos de consulta y optimización de consultas

El procesamiento de consulta, consiste en convertir transacciones de usuario en instrucciones para manipular los datos. Según el orden de las transacciones, la velocidad de respuesta del sistema varía.

El objetivo es reducir la cantidad de datos transferidos y es importante estudiar el coste de las comunicaciones.

Para optimizar las consultas se opera con **semijoin**.

En un sistema distribuido hay varios factores adicionales que complican el proceso de consulta en comparación con los sistemas centralizados; por el *coste de transferencia de datos a través de la red*:

la información se envía a ficheros intermedios, que a su vez se envían a otros nodos para nuevos procesos. Después los ficheros resultantes han de volver a enviarse al nodo que lanzó la consulta.

Objetivo de los algoritmos de consulta:

reducción de la cantidad de datos transferidos.

6.5. Bloqueos y concurrencia. Transacciones distribuidas

Las bases de datos distribuidas comparten datos y acceden a ella más de un usuario/recurso, por tanto, hablamos de bloqueos, concurrencia y transacción.

Bloqueo:

se produce cuando una acción que se ha de realizar está esperando a un evento.

Concurrencia:

es la propiedad de los sistemas que permiten que múltiples procesos sean ejecutados al mismo tiempo, y que puedan interactuar entre sí

Transacción:

es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

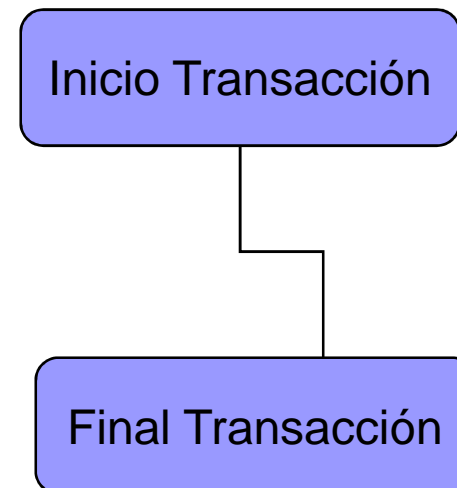
Transacciones distribuidas.

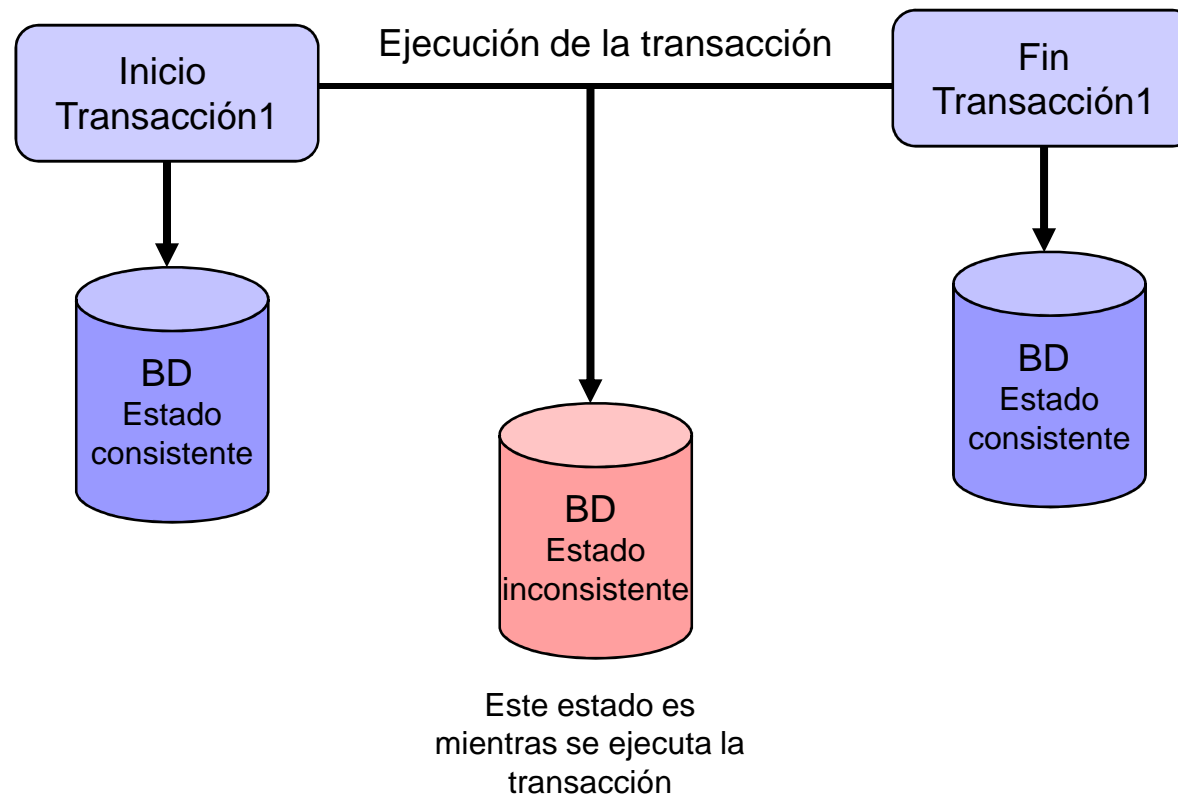
Es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

Las órdenes pueden ser actualizar, consultar, insertar, eliminar, etc; pero desde el punto de vista del usuario és una única operación.

Ejemplos: matriculación vía on-line a un curso, sacar dinero de un cajero, transferencia de fondos, etc.

Si una acción de la transacción no se puede ejecutar, entonces ninguna acción dentro de la secuencia que conforma la transacción tendrá efecto.





Propiedades Transacciones distribuidas: **ACID**

- Atomicidad:

Una transacción es una unidad atómica de procesamiento, se realiza o no se realiza.

- Consistencia:

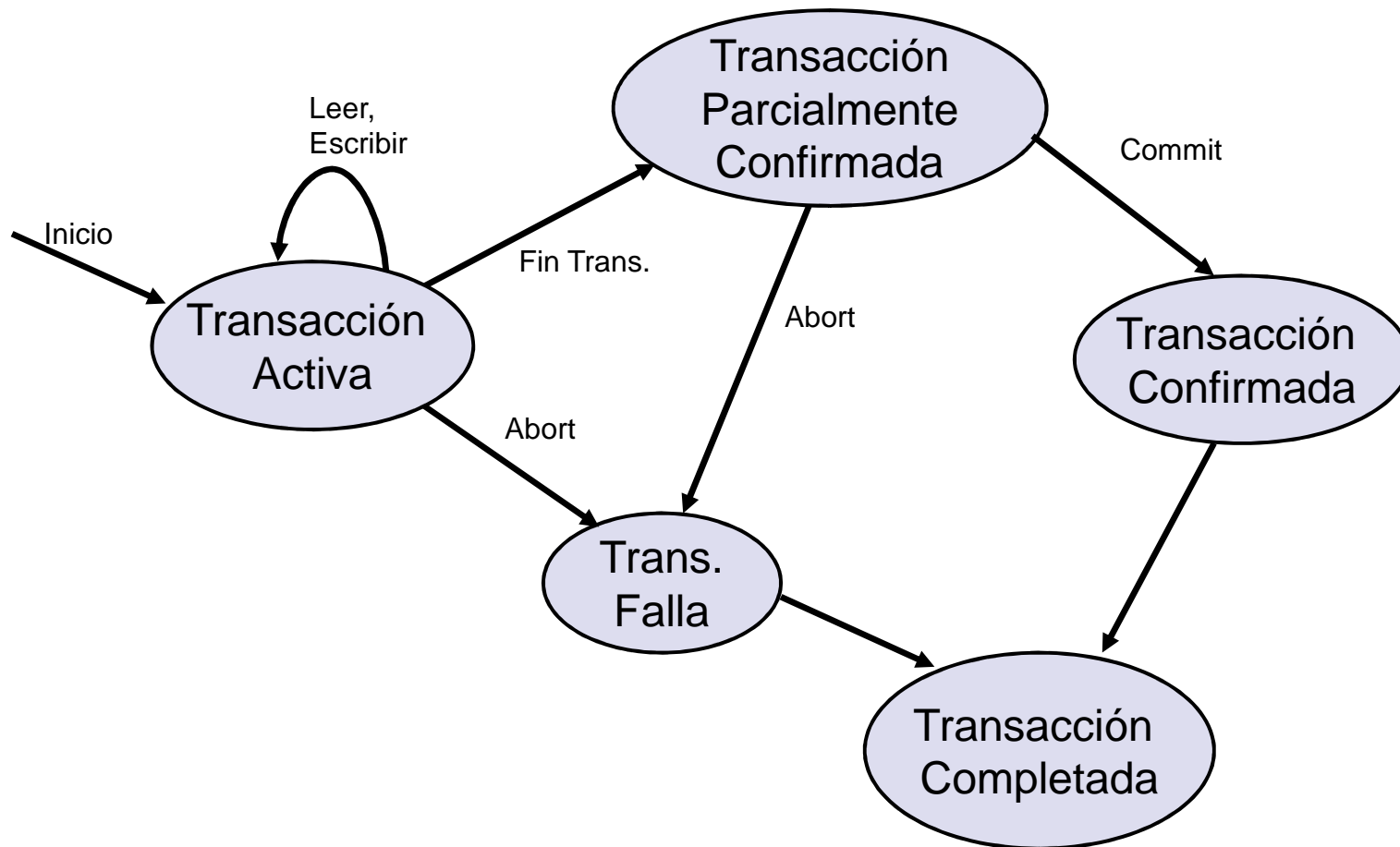
Preservar las restricciones de integridad.

- Aislamiento:

Una transacción no hará visibles sus modificaciones a otras transacciones hasta que termine de ejecutarse completamente. Una transacción desconoce si otras transacciones se estén ejecutando en el sistema.

- Durabilidad:

Una vez una transacción se ejecuta con éxito y realiza cambios sobre el sistema, estos cambios nunca se deben perder a causa de fallas en el sistema.



Ejemplos de transacción

a) Tenemos la siguiente consulta SQL, que incrementa el 16% el salario a todos los empleados de la base de datos que trabajan en el departamento de INFORMATICA:

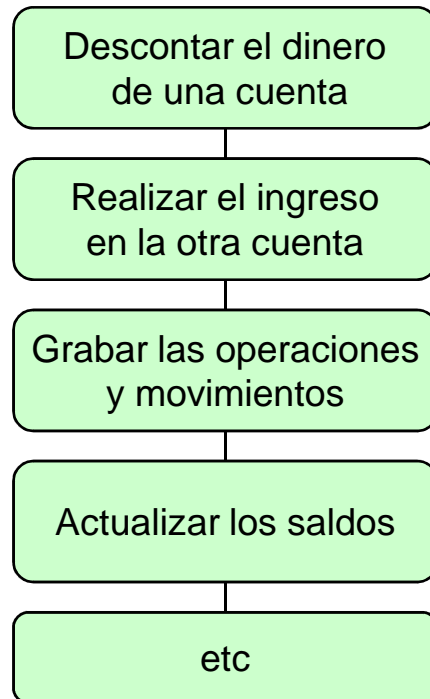
```
UPDATE empleado  
SET salario = salario * 1.16  
WHERE departamento = "INFORMATICA";
```

La misma consulta pero indicándola como una transacción seria:

```
BEGIN_TRANSACTION Actualiza_salario_dept_informatica  
BEGIN  
    UPDATE empleado  
    SET salario = salario * 1.16  
    WHERE departamento = "INFORMATICA"  
END;
```

b) Transferencias bancarias.

Para realizar una transferencia de dinero entre dos cuentas bancarias debemos: descontar el dinero de una cuenta, realizar el ingreso en la otra cuenta y grabar las operaciones y movimientos necesarios, actualizar los saldos



Si en alguno de estos puntos se produce un fallo en el sistema podríamos hacer descontado el dinero de una de las cuentas y no haberlo ingresado en la otra. Por lo tanto, todas estas operaciones deben ser correctas o fallar todas.

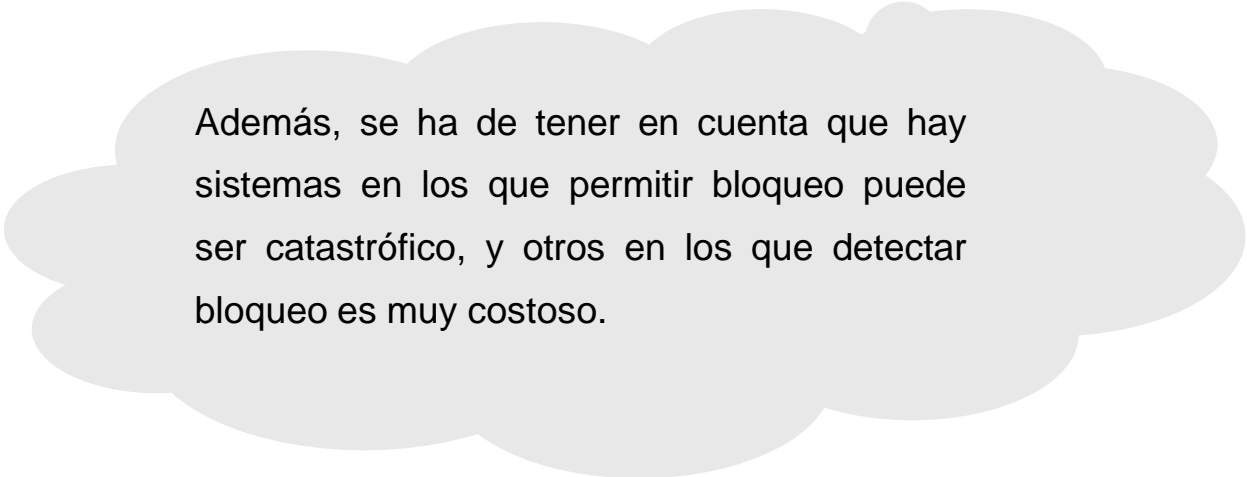
En estos casos, al confirmar la transacción (COMMIT) o al deshacerla (ROLLBACK) garantizamos que todos los datos quedan en un estado consistente.

- COMMIT (guarda con éxito las operaciones de la transacción).
Esta cláusula le informa al DTM que las operaciones han sido correcta y por tanto, la base de datos esta o debería estar en un estado consistente.
- ROLLBACK (deshace las operaciones de la transacción).
Esta cláusula le informa al DTM que ha habido algún error, se aborta las operaciones y por tanto, la base de datos podría estar en un estado inconsistente.

Bloqueo

Es cuando una acción que debe ser realizada está esperando a un evento.

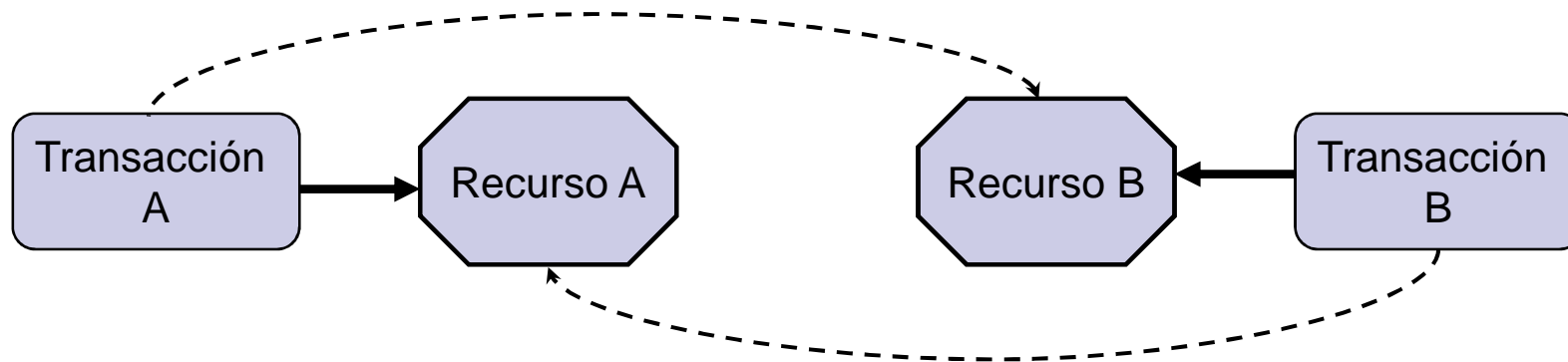
Para manejar los bloqueos, hay diferentes posturas: la prevención, la detección y la recuperación.



Además, se ha de tener en cuenta que hay sistemas en los que permitir bloqueo puede ser catastrófico, y otros en los que detectar bloqueo es muy costoso.

Concurrencia

Ejemplo de bloqueo mutuo



Ejemplos específicos en las bases de datos distribuidas podemos destacar:

- Actualización perdida: cuando dos transacciones concurrentes borran el efecto una de la otra
- La extracción inconsistente: acceder a información modificada parcialmente por una transacción

Soluciones de Concurrency

Es complicado buscar soluciones para controlar la concurrencia y detectar y manejar los bloqueos, pues no existe ningún algoritmo. Algunos factores que se han de tener en cuenta para buscar las soluciones son:

- Información duplicada en una BDD.
- Si falla un nodo o la red de comunicación mientras se actualiza, el SGBDD ha de gestionar la recuperación ante el fallo.
- Sincronización de las transacciones en múltiples nodos, pues es difícil ya que hay nodos que no pueden tener información inmediata de acciones realizadas por otros nodos concurrentemente.
- Control de bloqueos mutuos: no hay solución viable y la forma más simple es el tiempo máximo de espera en las peticiones de bloqueos.

6.6. Seguridad y recuperación de la información en las bases de datos distribuidas

El SGBD facilita normalmente mecanismos para prevenir los fallos (subsistema de control), para detectarlos una vez que se han producido (subsistema de detección) y para corregirlos después de haber sido detectados (subsistema de recuperación).

Aspectos que se han de tener en cuenta:

- La protección de los datos deberá llevarse a cabo contra fallos físicos, fallos lógicos y fallos humanos (intencionados o no).
- Encriptación.
- Accesibilidad. La información debe estar disponible.
- Integridad. Permite asegurar que los datos no han sido falseados.
- Control de concurrencia.

- Confidencialidad. No desvelar datos a usuarios no autorizados. Comprende también la privacidad de datos (protección de datos personales).
 - Cifrar los datos mediante algoritmos de codificación, de tal manera que usuarios no autorizados no puedan descifrar los datos.
 - Confiar en usuarios, en la red y en los sitios remotos.
 - Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
 - Necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles.
 - Acceso a usuarios a porciones selectas de una base de datos sin tener acceso al resto
⇒ identificación y autenticación remotas, a nivel de usuarios y/o nodos. Este aspecto lo ha de controlar el DBA y se encarga de las siguientes acciones:



Privilegios a los usuarios, clasificar usuarios y acceso a datos según política de la organización

DBA

1. Creación de cuentas
2. Concesión de privilegios
3. Revocación de privilegios.
4. Asignación de niveles de seguridad.



Recuperación

Fallo de los nodos.

Copias múltiples de fragmentos de datos.

Transacción distribuida correcta

Fallo de las conexiones en las comunicaciones

6.7. Arquitectura de los sistemas distribuidos

Arquitecturas Disponibles:

- Centralizada
- Cliente-Servidor
- Paralelas
- Distribuidas

6.8. Diseño y gestión de bases de datos distribuidas

Para diseñar BDD, se ha de tener en cuenta mayormente: la ubicación de los *datos* y los *programas* que se usarán a través de los diferentes sitios de una red de computadoras.

La decisión de donde colocar a las aplicaciones es tanto con el software del SGBDD como con las aplicaciones que se van a ejecutar sobre la base de datos.

El diseño de las bases de datos centralizadas contempla la definición de definición del *esquema conceptual* (descripción de la base de datos) y el *esquema físico* de la base de datos (almacenamiento de los datos y accesos a ellos).

Bases de dades distribuïdes

| Sistema Centralizado | Sistema Distribuido |
|--|--|
| Control centralizado: un solo DBA | Control jerárquico: DBA global y DBA local |
| Independencia de datos: la organización de los datos es transparente para el programador | Transparencia en la distribución: la localización de los datos es un aspecto adicional a la independencia de los datos |
| Reducción de redundancia: hay una única copia y se comparte | Replicación de datos: hay varias copias de datos incrementando la localización y disponibilidad de los datos. |
| Estructuras físicas complejas para tener accesos eficientes | No hay estructuras intersticios. Optimización global para reducir transferencia de datos. |
| Seguridad favorable | Problemas de seguridad intrínsecos |

Preguntes!!!!

