

**Bases de datos distribuidas.**

**Administración y programación de Base  
de Datos**

**Martita Muñoz**

# Introducción.

- Las Bases de Datos Distribuidas (DDB) emergió gracias a la unión de dos tecnologías: **Base de Datos** y **Comunicación de datos y de redes**.
- Las organizaciones han estado muy interesadas en la **descentralización** del procesamiento a la vez de conseguir una **integración** de las diferentes fuentes de información.
- Las DDB son implementadas bajo la arquitectura **Cliente Servidor**.

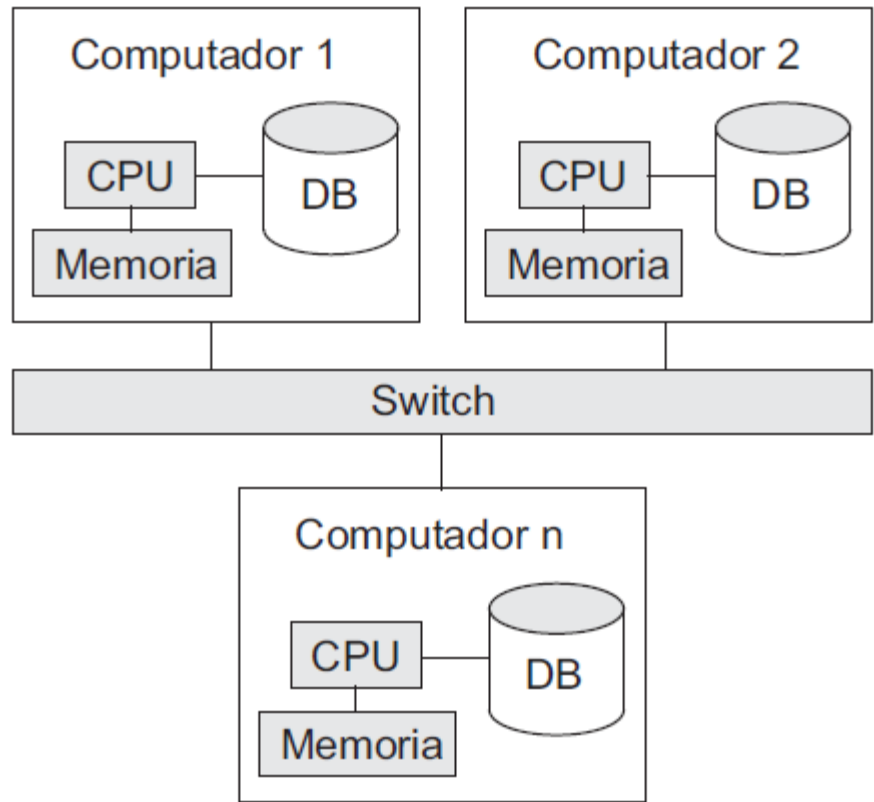
# Conceptos.

- **DDB (Base de datos distribuida, *Distributed DataBase*):** colección de múltiples bases de datos distribuidas interrelacionadas de forma lógica sobre una red de computadores.
- **DDBMS (Sistema de administración de bases de datos distribuidas, *Distributed DataBase Management System*):** el software encargado de administrar la base de datos distribuida mientras hace la distribución transparente para el usuario.

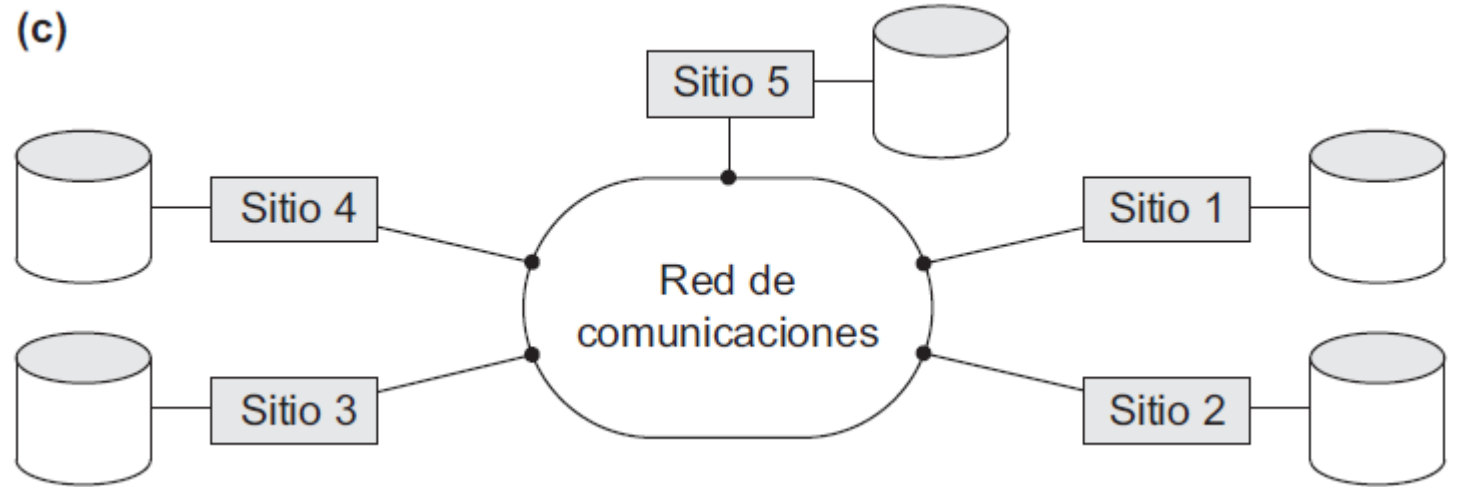
# Tecnología paralela frente a distribuida.

- **Sistemas de administración paralelos de bases de datos:** emplean tecnología de procesador paralelo, compartiendo alguna clase de memoria.
- **Arquitectura “nada compartido”:** no existe memoria común, y esos procesadores se comunican mediante una red de interconexión de alta velocidad (bus o switch).
  - Existe simetría y homogeneidad de nodos; esto no se cumple en un entorno de base de datos distribuida.

# Tecnología paralela frente a distribuida.



Arquitectura “nada compartido”



Arquitectura de base de datos distribuida auténtica.

# Ventajas.

1. **Administración de datos distribuidos con distintos niveles de transparencia:** ocultar los detalles de dónde está físicamente ubicado cada fichero (tabla, relación) dentro del sistema.
2. **Incremento de la fiabilidad y la disponibilidad:**
  - **Fiabilidad:** la probabilidad de que un sistema esté funcionando en un momento de tiempo.
  - **Disponibilidad:** la probabilidad de que el sistema esté continuamente disponible durante un intervalo de tiempo.
3. **Rendimiento mejorado:** mantiene la información lo más cerca posible del punto donde es más necesaria.
4. **Expansión más sencilla:** la expansión del sistema en términos de incorporación de más datos, incremento del tamaño de las bases de datos o la adición de más procesadores es mucho más sencilla.

# Funciones adicionales.

- Seguimiento de los datos.
- Procesamiento de consultas distribuidas.
- Administración de transacciones distribuidas.
- Administración de datos replicados.
- Recuperación de una base de datos distribuida.
- Seguridad.
- Administración del directorio (catálogo) distribuido.

# Técnica de fragmentación.

- **Fragmentación:** División de una relación en unidades lógicas más pequeñas llamadas **fragmentos**.
- **Replicación de datos:** permite que cierta información se almacene en más de un sitio.
- **Asignación de fragmentos:** Definir qué sitios almacenará los fragmentos y sus réplicas.

*Estas técnicas se emplean durante el diseño de la base de datos distribuida.*



# Fragmentación horizontal.

- La **fragmentación horizontal** divide una relación *horizontalmente* agrupando filas para crear subconjuntos de tuplas, cada uno de ellos con un cierto significado lógico.
- Las tuplas pertenecientes al fragmento horizontal se especifican mediante una condición sobre uno o más atributos de la relación.

# Fragmentación horizontal.

EMPLEADO (Dno=5)(Dno=4)(Dno=1)

## EMPLEADO

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	01-09-1965	Eloy I, 98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	12-05-1968	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	29-03-1969	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	10-11-1937	Las Peñas, 1	H	55000	NULL	1

# Fragmentación horizontal.

- Cada fragmento horizontal de una relación  $R$  puede especificarse con una operación  $\sigma_{C_i}(R)$  en el álgebra relacional, donde  $C_i$  es la condición para generar el fragmento  $i$ .
- **Fragmentación horizontal completa:** Cada tupla de  $R$  satisface  $(C_1 OR C_2 OR ... OR C_n)$ .
- **Fragmentación horizontal disjunta:** ninguna tupla de  $R$  satisface  $(C_i AND C_j)$  para cualquier  $i \neq j$ .
- Para reconstruir la relación  $R$ : **UNION**.

# Fragmentación vertical.

- La fragmentación vertical divide una relación “verticalmente” por columnas.
- Un **fragmento vertical** de una relación sólo mantiene ciertos atributos de la misma.
- Es necesario incluir la clave primaria, o alguna otra clave candidata, en **cada fragmento vertical** de modo que nos permita reconstruir la relación completa a partir de ellos.

# Fragmentación vertical.

EMPLEADO (Dni, Nombre, FechaNac, Dirección, Sexo)(Dni, Sueldo, SuperDni, Dno)

EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	01-09-1965	Eloy I, 98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	12-05-1968	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	29-03-1969	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	10-11-1937	Las Peñas, 1	H	55000	NULL	1

# Fragmentación vertical.

- Un fragmento vertical de una relación  $R$  puede especificarse mediante una operación  $\pi_{L_i}(R)$  en el álgebra relacional.
- **Fragmentación vertical completa:** Un conjunto de fragmentos verticales cuyas listas de proyección  $L_1, L_2, \dots, L_n$  incluyen todos los atributos de  $R$  pero que sólo comparten el atributo de clave primaria. Cumple con las siguientes condiciones:
  - $L_1 \cup L_2 \cup \dots \cup L_n = ATTRS(R)$
  - $L_i \cap L_j = PK(R)$  para cualquier  $i \neq j$ .
- Para reconstruir la relación  $R$ : **FULL OUTER JOIN**.

# Fragmentación mixta (híbrida).

- Combinación de la fragmentación horizontal y vertical.
- **Esquema de fragmentación:** un conjunto de fragmentos que incluyen **todos** los atributos y tuplas de esa base de datos y que permiten reconstruirla mediante operaciones de OUTER JOIN y UNION.
- **Esquema de ubicación:** es un mapa que indica, por cada fragmento, el (los) sitio(s) en el que está almacenado.

# Replicación y ubicación de datos.

## Base de datos distribuida totalmente replicada.

- Mejora la disponibilidad.
- Mejora el rendimiento en la recuperación de consultas globales.
- Puede ralentizar drásticamente las operaciones de actualización.
- Las técnicas de control de concurrencia y recuperación son más caras.

## No replicación o ubicación no redundante.

- Cada fragmento está almacenado en un sitio.
- Todos los fragmentos **deben ser disjuntos**.



# Ejemplo de aplicación.

Suponga que una empresa con tres sitios operativos (uno por cada departamento) presenta la siguiente base de datos a distribuir:

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
--------	-----------	-----------	------------	----------	-----------	------	--------	----------	-----

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
------------	-------------------	-------------	----------------------

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
-------------------	----------------------

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
----------------	--------------------	-------------------	-----------------

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
--------------------	----------------	-------

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relación
--------------------	------------------------	------	----------	----------

PROYECTO

NombreProyecto	NumProyecto	UbicacionProyecto	NumDptoProyecto
ProductoX	1	Valencia	5
ProductoY	2	Sevilla	5
ProductoZ	3	Madrid	5
Computación	10	Gijón	4
Reorganización	20	Madrid	1
Comunicaciones	30	Gijón	4

EMPLEADO

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	01-09-1965	Eloy I, 98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	12-05-1968	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	29-03-1969	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	10-11-1937	Las Peñas, 1	H	55000	NULL	1

DEPARTAMENTO

NombreDpto	NumeroDpto	DniDirector	FechaIngresoDirector
Investigación	5	333445555	22-05-1988
Administración	4	987654321	01-01-1995
Sede Central	1	888665555	19-06-1981

LOCALIZACIONES\_DPTO

NumeroDpto	UbicacionDpto
1	Madrid
4	Gijón
5	Valencia
5	Sevilla
5	Madrid

TRABAJA\_EN

DniEmpleado	NumProy	Horas
123456789	1	32,5
123456789	2	7,5
666884444	3	40,0
453453453	1	20,0
453453453	2	20,0
333445555	2	10,0
333445555	3	10,0
333445555	10	10,0
333445555	20	10,0
999887777	30	30,0
999887777	10	10,0
987987987	10	35,0
987987987	30	5,0
987654321	30	20,0
987654321	20	15,0
888665555	20	NULL

# Ejemplo de aplicación.

El departamento 1, ubicado en el sitio 1, corresponde a la oficina central y accede regularmente a toda la información de empleados y proyectos, además de controlar los datos de SUBORDINADO con miras a los seguros.

Los departamentos 5 y 4, ubicados en los sitios 2 y 3 respectivamente, se espera un acceso frecuente a la información de EMPLEADO y PROYECTO para aquellos asalariados que trabajan en ese departamento y los proyectos controlados por ese departamento. Específicamente, de EMPLEADO accede fundamentalmente a los atributos Nombre, Dni, Sueldo y SuperDni.

# Ejemplo de aplicación.

- Según todos estos requerimientos, toda la base de datos puede estar contenida en el sitio 1.
- Para determinar los fragmentos a replicar en los sitios 2 y 3, primero podemos fragmentar horizontalmente DEPARTAMENTO por su clave NúmeroDpto.
- Luego, se puede proyectar dicha fragmentación a las relaciones
- EMPLEADO, PROYECTO y LOCALIZACIONES\_DPTO usando la clave foránea sobre DEPARTAMENTO (Dno, NumDptoProyecto y NúmeroDpto, respectivamente).
- Posteriormente, podemos partir verticalmente los fragmentos EMPLEADO resultantes para incluir sólo los atributos {Nombre, Dni, Sueldo, SuperDni, Dno}.
- ¿Cómo dividir TRABAJA\_EN?

# Ejemplo de aplicación.

EMPD\_5

Nombre	Apellido1	Apellido2	Dni	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	30000	333445555	5
Alberto	Campos	Sastre	333445555	40000	888665555	5
Fernando	Ojeda	Ordóñez	666884444	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	25000	333445555	5

DEP\_5

NombreDpto	NúmeroDpto	DniDirector	FechaIngresoDirector
Investigación	5	333445555	1988-05-22

TRABAJA\_EN\_5

DniEmpleado	NumProy	Horas
123456789	1	32,5
123456789	2	7,5
666884444	3	40,0
453453453	1	20,0
453453453	2	20,0
333445555	2	10,0
333445555	3	10,0
333445555	10	10,0
333445555	20	10,0

PROYS\_5

NombreProyecto	NumProyecto	UbicaciónProyecto	NumDptoProyecto
Producto X	1	Valencia	5
Producto Y	2	Sevilla	5
Producto Z	3	Madrid	5

Datos del sitio 2

EMPD\_4

Nombre	Apellido1	Apellido2	Dni	Sueldo	SuperDni	Dno
Alicia	Jiménez	Celaya	999887777	25000	987654321	4
Juana	Sainz	Oreja	987654321	43000	888665555	4
Luis	Pajares	Morera	987987987	25000	987654321	4

DEP\_4

NombreDpto	NúmeroDpto	DniDirector	FechaIngresoDirector
Administración	4	987654321	1995-01-01

DEP\_4\_LOCS

NúmeroDpto	Ubicación
4	Gijón

TRABAJA\_EN\_4

DniEmpleado	NumProy	Horas
333445555	10	10,0
999887777	30	30,0
999887777	10	10,0
987987987	10	35,0
987987987	30	5,0
987654321	30	20,0
987654321	20	15,0

PROYS\_4

NombreProyecto	NumProyecto	UbicaciónProyecto	NumDptoProyecto
Computación	10	Gijón	4
Comunicaciones	30	Gijón	4

Datos del sitio 3

# Tipos de sistemas de bases de datos distribuidas.

De acuerdo al **grado de homogeneidad**.

- **Homogéneo:** Todos los servidores (o DBMSs locales individuales) y todos los usuarios (clientes) usan idéntico software.
- **Heterogéneo:** No es homogéneo.

De acuerdo al **grado de autonomía local**.

- **Baja autonomía:** un DDBMS que *tiene el aspecto* de un DBMS centralizado para el usuario.
- **Alta autonomía:** *DDBMS federado o sistema de "múltiples bases de datos"*, en el que cada servidor es un DBMS centralizado independiente y autónomo que tiene sus propios usuarios, transacciones y DBA locales.

# Federated DataBase System.

- En un FDBS heterogéneo, un servidor puede ser un DBMS relacional, otro un DBMS de red y un tercero un DBMS de objetos o jerárquico.

## Problemas que esto implica:

- Diferencias en los modelos de datos.
- Diferencias en las restricciones.
- Diferencias en los lenguajes de consulta.

# Procesamiento de consultas.

El principal objetivo de los DDBMS es:

*Reducir la cantidad de la transferencia de datos a la hora de elegir una estrategia de ejecución de una consulta distribuida.*



# Ejemplo de procesamiento de consulta.

Sitio 1:

Relación completa: 1.000.000 bytes

EMPLEADO 10.000 registros

100 bytes

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
15 bytes	15 bytes		9 bytes						4 bytes

Sitio 2:

Relación completa: 3.500 bytes

DEPARTAMENTO 100 registros

35 bytes

NombreDpto	NúmeroDpto	DniDirector	FechaIngresoDirector
10 bytes	4 bytes	9 bytes	

# Ejemplo de procesamiento de consulta.

El Sitio 3 requiere la siguiente consulta C1: *por cada empleado, recuperar su nombre y el del departamento para el cual trabaja.*

- Tamaño de la respuesta: 40 bytes \* 10.000 registros: 400.000 bytes.

## Estrategias a seguir:

1. Transferir EMPLEADO y DEPARTAMENTO al Sitio 3 y procesar allí la consulta. Se transfieren 1.003.500 bytes.
2. Transferir EMPLEADO al Sitio 2, procesar la consulta y enviar el resultado al Sitio 3. Se transfieren 1.400.000 bytes.
3. Transferir DEPARTAMENTO al Sitio 1, procesar la consulta y enviar el resultado al Sitio 3. Se transfieren 403.500 bytes.

# Ejemplo de procesamiento de consulta.

El Sitio 3 requiere la siguiente consulta C2: *por cada departamento, recuperar su nombre y el de su director.*

- Tamaño de la respuesta: 40 bytes \* 100 registros: 4.000 bytes.

## Estrategias a seguir:

1. Transferir EMPLEADO y DEPARTAMENTO al Sitio 3 y procesar allí la consulta. Se transfieren 1.003.500 bytes.
2. Transferir EMPLEADO al Sitio 2, procesar la consulta y enviar el resultado al Sitio 3. Se transfieren 1.004.000 bytes.
3. Transferir DEPARTAMENTO al Sitio 1, procesar la consulta y enviar el resultado al Sitio 3. Se transfieren 7.500 bytes.

# Semijoin.

- Pretende reducir el número de tuplas antes de ser transferida.
- Lo que pretende hacer es enviar la **columna de concatenación** de una relación  $R$  al sitio en el que se encuentra la otra relación  $S$  para, a continuación, concatenar esta columna con  $S$ .
- Operación de semijoin:  $R \bowtie_{A=B} S$
- La relación no es conmutativa.

# Semijoin.

Ejemplo en C1: *por cada empleado, recuperar su nombre y el del departamento para el cual trabaja.*

1. Enviamos  $F = \pi_{\text{NumeroDepto}}(\text{DEPARTAMENTO})$  del Sitio 2 al Sitio 1. Su tamaño es 4 bytes \* 100 registros = 400 bytes.
2. Enviamos  $R = \pi_{\text{Dno,Nombre,Apellido1}}(F \bowtie_{\text{NroDepto=Dno}} \text{EMPLEADO})$  del Sitio 1 al Sitio 2. Su tamaño es de 34 \* 10.000 = 340.000 bytes.
3. Se presenta  $\pi_{\text{Nombre,Apellido1,NombreDepto}}(R \bowtie_{\text{Dno=NroDepto}} \text{DEPARTAMENTO})$  como resultado.

**¿Qué ocurre con C2?**

# Descomposición de una actualización y una consulta.

- En un DDBMS **sin transparencia de distribución**, el usuario articula una consulta directamente en términos de fragmentos específicos.
- Un DDBMS que soporta *distribución total, fragmentación y transparencia de replicación* permite que el usuario especifique una consulta **exactamente igual que si el DBMS estuviera centralizado**.
- En el caso de actualizaciones, el DDBMS es responsable de mantener la **consistencia entre los elementos replicados** usando uno de los algoritmos de control de concurrencia distribuida.
- Para las consultas, un módulo de **descomposición de consulta** debe partir, o **descomponer**, esas consultas en **subconsultas** que puedan ejecutarse individualmente en los distintos sitios.

# Control de concurrencia.

Problemas que se dan en DBMS distribuido que no se dan en un entorno centralizado:

- **Tratar con *múltiples copias* de los datos:** Mantener la consistencia.
- **Fallo de los sitios individuales:** DDBMS debe seguir operando.
- **Fallo de los enlaces de comunicación:** Particionamiento de la red.
- **Confirmación distribuida:** problemas con el commit.
- **Estancamiento distribuido:** Problemas con el deadlock.

# Técnicas de control de concurrencia.

## Técnica de sitio primario:

- Se designa un único **sitio primario** para que sirva como **sitio coordinador** *de todos los elementos de la base de datos*.
- Ventaja: no es demasiado complejo.
- Desventaja: pueden sobrecargarlo y generar un cuello de botella en el sistema.

## Sitio primario con sitio de respaldo:

- Toda la información sobre bloqueos se mantiene tanto en el sitio primario como en el de respaldo para que, si falla el primero, el segundo tome el control y se elija un nuevo sitio de respaldo.
- Ventaja: simplifica el proceso de recuperación cuando falla el sitio primario.
- Desventaja: se ralentiza el proceso de adquisición de bloqueos.



# Elección de un nuevo sitio coordinador en el caso de producirse un fallo.

- Siempre que falla un sitio coordinador en cualquiera de las técnicas anteriores, el que aún permanece activo debe elegir un nuevo coordinador.
- Si todos los sitios coordinadores están caídos, se puede llamar a un proceso de **elección** para elegir el nuevo sitio coordinador.
- En este proceso, cualquier sitio *Y* que intente comunicar repetidamente con el sitio coordinador sin obtener respuesta puede asumir que éste está caído e iniciar el proceso de elección.
- Para ello, expide un mensaje a todos los sitios en ejecución proponiéndose él mismo como nuevo sitio coordinador.
- Tan pronto como *Y* reciba una mayoría de votos afirmativos, puede declararse como el nuevo sitio coordinador.

# Recuperación distribuida.

- El proceso de recuperación en una base de datos distribuida es algo complejo.
- En ciertos casos, es incluso más **complicado determinar si un sitio está caído** sin intercambiar numerosos mensajes con otros sitios.
- Otro problema que aparece en la recuperación distribuida es la **confirmación distribuida**. Cuando una transacción está actualizando datos en varios sitios, **no puede realizar una confirmación** hasta que no esté seguro de que el efecto de la misma no pueda perderse en **cada sitio**.

**FIN**