

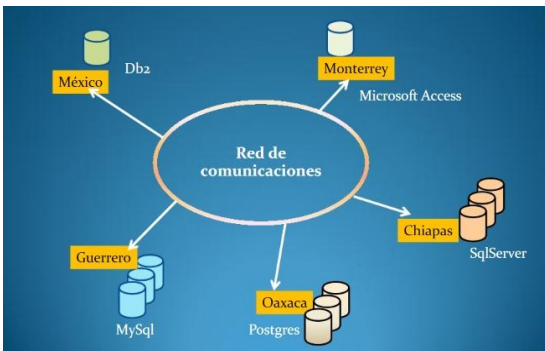
BASES DE DATOS DISTRIBUIDAS

Conceptos Teóricos

“No Colocar los Huevos en la misma Canasta”
Dicho Popular

- Docentes: Ing. Fernando Sato
A.S. Sebastian Trossero

versión: 20180908



RESUMEN

Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Transparencias
- Procesamiento Distribuido
- Transacciones Distribuidas

Bases de Datos Distribuidas

Introducción – Ubicación Arquitectura

Arquitecturas:

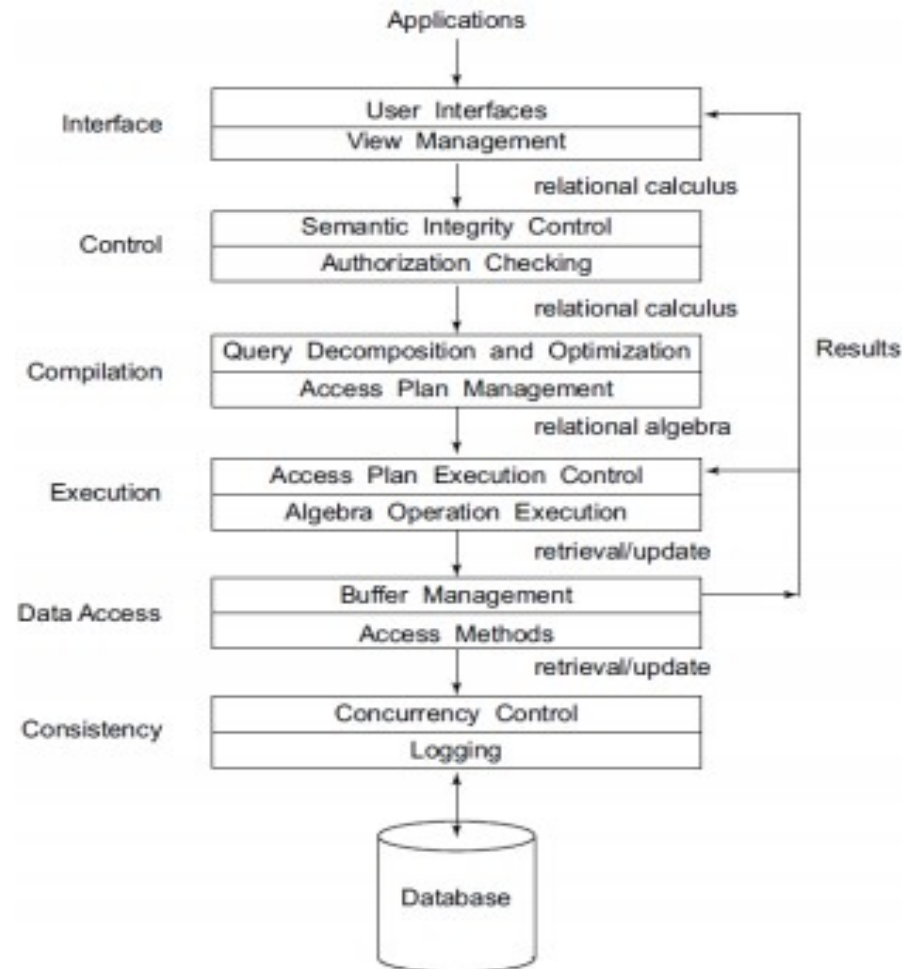
- Sistemas Centralizados.
- Sistemas Cliente/Servidor.
- Sistemas Paralelos.
- **Sistemas Distribuidos.**

Bases de Datos Distribuidas

Introducción – Ubicación Arquitectura

Arquitecturas:

- Sistemas Centralizados:
 - Sistemas MonoUsuario
*(Suelen ser Monolíticos,
Algunos No ACID,
Sin Posib.Recuperación)*
 - Sistemas MultiUsuario
*(Antiguas Implement.,
interface Caracter)*
- Sistemas Cliente/Servidor.
- Sistemas Paralelos.
- **Sistemas Distribuidos.**

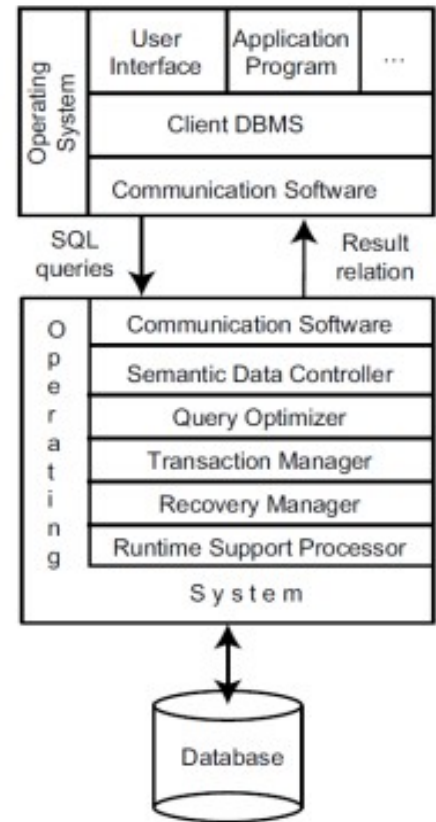


Bases de Datos Distribuidas

Introducción – Ubicación Arquitectura

Arquitecturas:

- Sistemas Centralizados:
- Sistemas Cliente/Servidor
*(2 Espacios de Direcciones,
Funciones divididas:
Back-End Server Propiamente dicho
Front-End cliente, algunos con propiedades
gráficas / representaciones no carácter, etc)*
- Sistemas Paralelos.
- **Sistemas Distribuidos.**

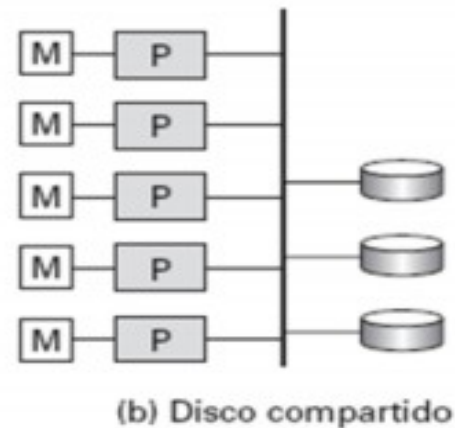
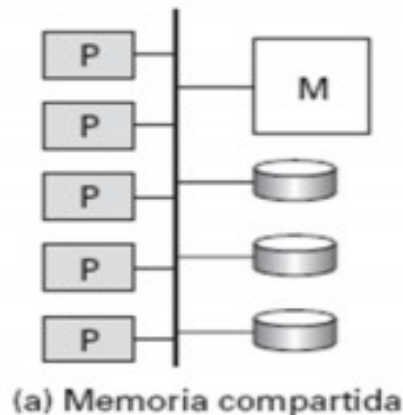


Bases de Datos Distribuidas

Introducción – Ubicación Arquitectura

Arquitecturas:

- Sistemas Centralizados.
- Sistemas Cliente/Servidor.
- Sistemas Distribuidos.
- **Sistemas Paralelos.**

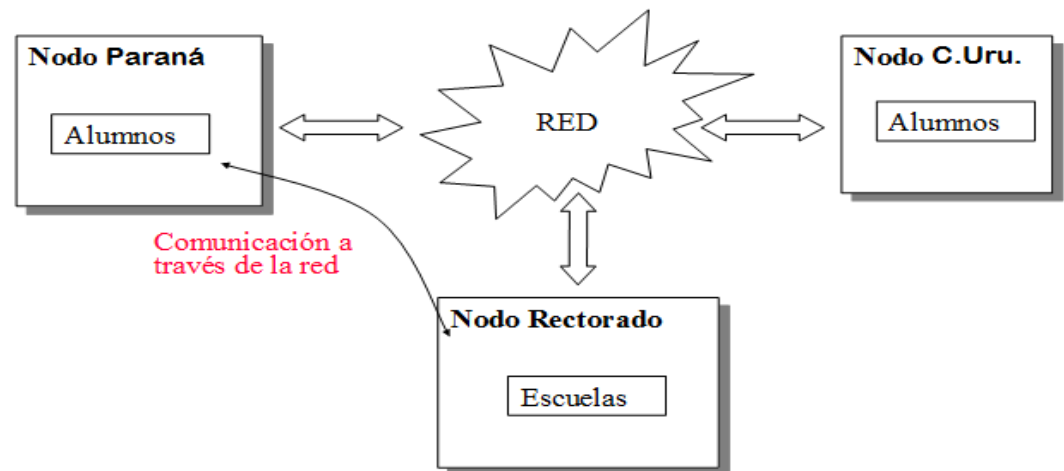


Bases de Datos Distribuidas

Introducción – Ubicación Arquitectura

Arquitecturas:

- Sistemas Centralizados.
- Sistemas Cliente/Servidor.
- Sistemas Paralelos.
- Sistemas Distribuidos *Basado en 2 Tecnologías*
 - i. *Bases de Datos*
 - ii. *Redes y Comunicaciones*



Bases de Datos Distribuidas

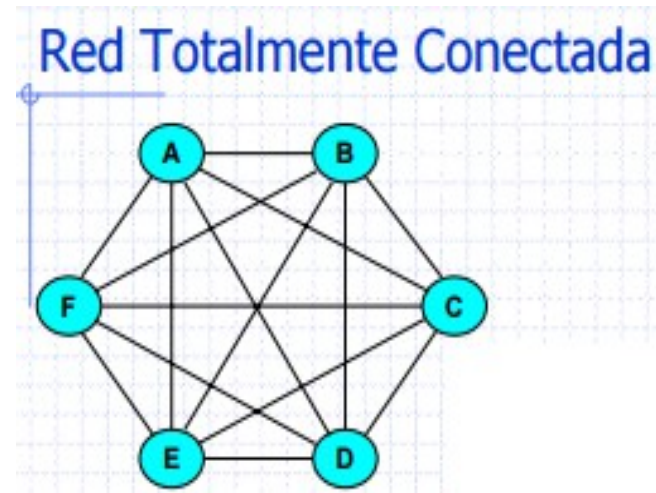
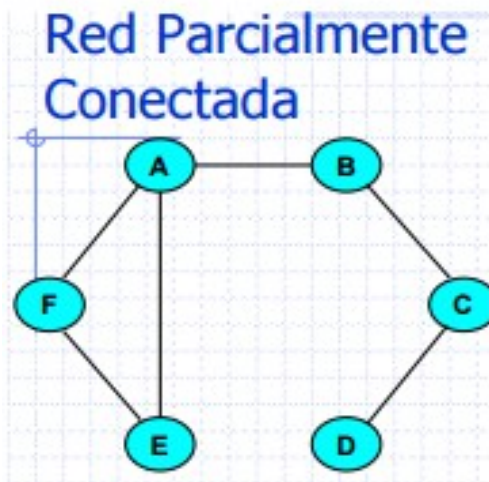
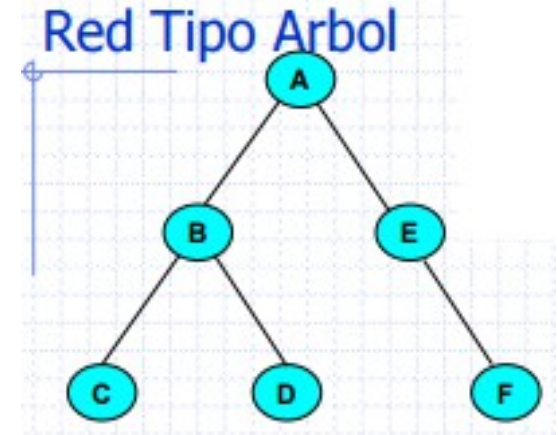
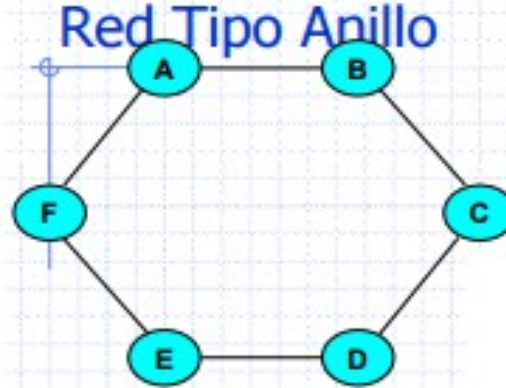
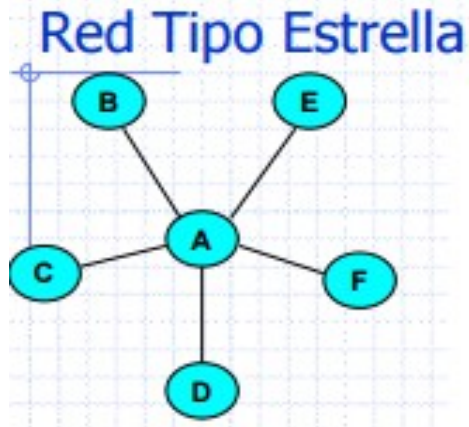
Introducción – Tecnología Com y Redes

Esta última tecnología ha avanzado mucho en los últimos años gracias a:

- Mejora medios físicos de comunicación por **inalámbricas, por satélite, Redes de Área Metropolitana (MAN)**.
- Estandarización de **protocolos como TCP/IP**
- y por sobre todo, gracias a la **explosión de Internet**

Bases de Datos Distribuidas

Introducción – Topologías de Redes



Bases de Datos Distribuidas

Introducción – Topologías de Organizaciones

Se promueve la **Descentralización** de las organizaciones:

Horizontal → Permite a las unidades autonomía.

Vertical → Unidades con función específica.

Fuerte ***impacto*** en los Sistemas de Información:

Primero: Sistemas Autónomos desconectados.

Se presentaron problemas de falta de integración y consistencia de la información.

Luego: Convergen a Sistemas Integrados.

Impulsando los Sistemas Distribuidos.

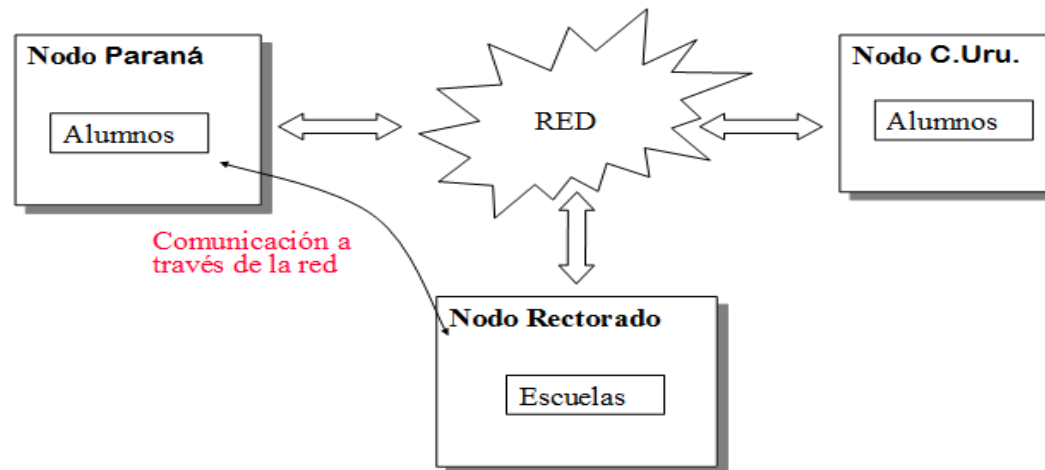
Bases de Datos Distribuidas

Introducción – Estructura Esquemática

Un Sistema de una Organización Distribuida Horizontalmente debe posibilitar dos tipos de consultas:

Consulta Alumnos Paraná (Facultad CYT): **local.**

Consulta Alumnos Nivel Universidad: **global.**



Bases de Datos Distribuidas

Introducción – Estructura Esquemática

Las **BDD** aportan al dominio de la gestión de bases de datos las **ventajas de la computación distribuida**.

En la cual **un conjunto de elementos de procesamiento** (no necesariamente homogéneos) **interconectados por una red** cooperan en forma coordinada en la ejecución de tareas, dividiendo problemas grandes e inmanejables en piezas más pequeñas que se resuelven en forma coordinada.

Bases de Datos Distribuidas

Introducción – Estructura Esquemática

La **viabilidad económica** de este enfoque tiene dos razones:

- (1) **Se aprovecha más la potencia del ordenador** en la resolución de tareas complejas.
- (2) **Cada elemento de procesamiento se puede gestionar en forma independiente** en el desarrollo de tareas locales (**autonomía**).

Bases de Datos Distribuidas

Introducción – Estructura Esquemática

Entonces, podemos definir:

Base de Datos Distribuida (BDD): colección de múltiples bases de datos interrelacionadas lógicamente, distribuidas sobre una red de computadores y manejadas por un **SGBDD**.

Sistema de Gestión de Bases de Datos Distribuido (SGBDD): Sistema de software que maneja BDDs y que hace de la distribución algo transparente al usuario.

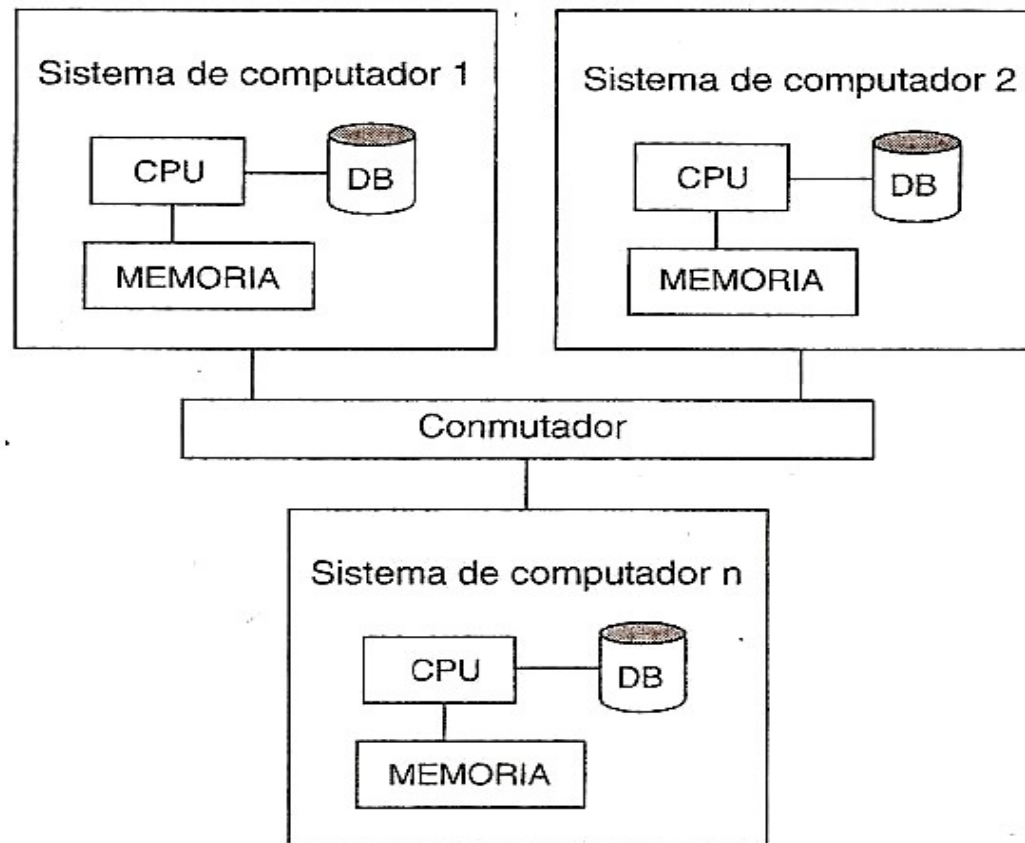
SISTEMA DE BDD = BDD + SGBDD

Bases de Datos Distribuidas

Introducción – Arquitecturas Especificas

Análisis de Arquitecturas
Compartidas →

Nada



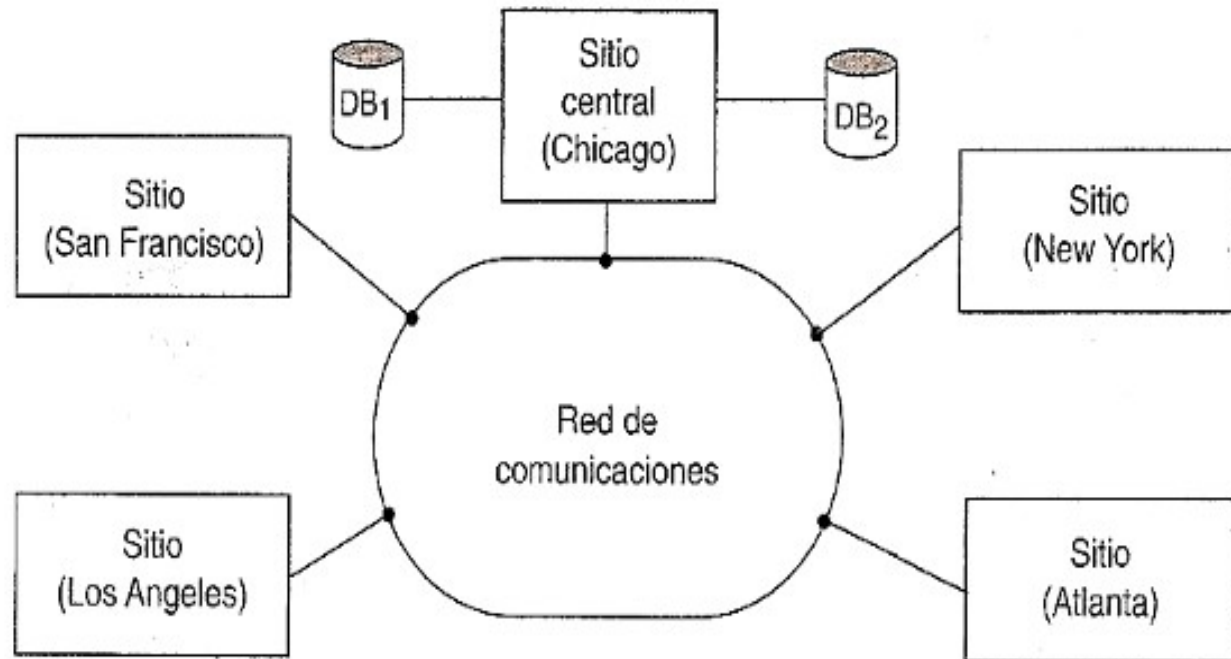
Bases de Datos Distribuidas

Introducción – Arquitecturas Especificas

Análisis de Arquitecturas →

En **Red** con **BD Centralizada** en uno de los sitios, o en la nube.

(b)

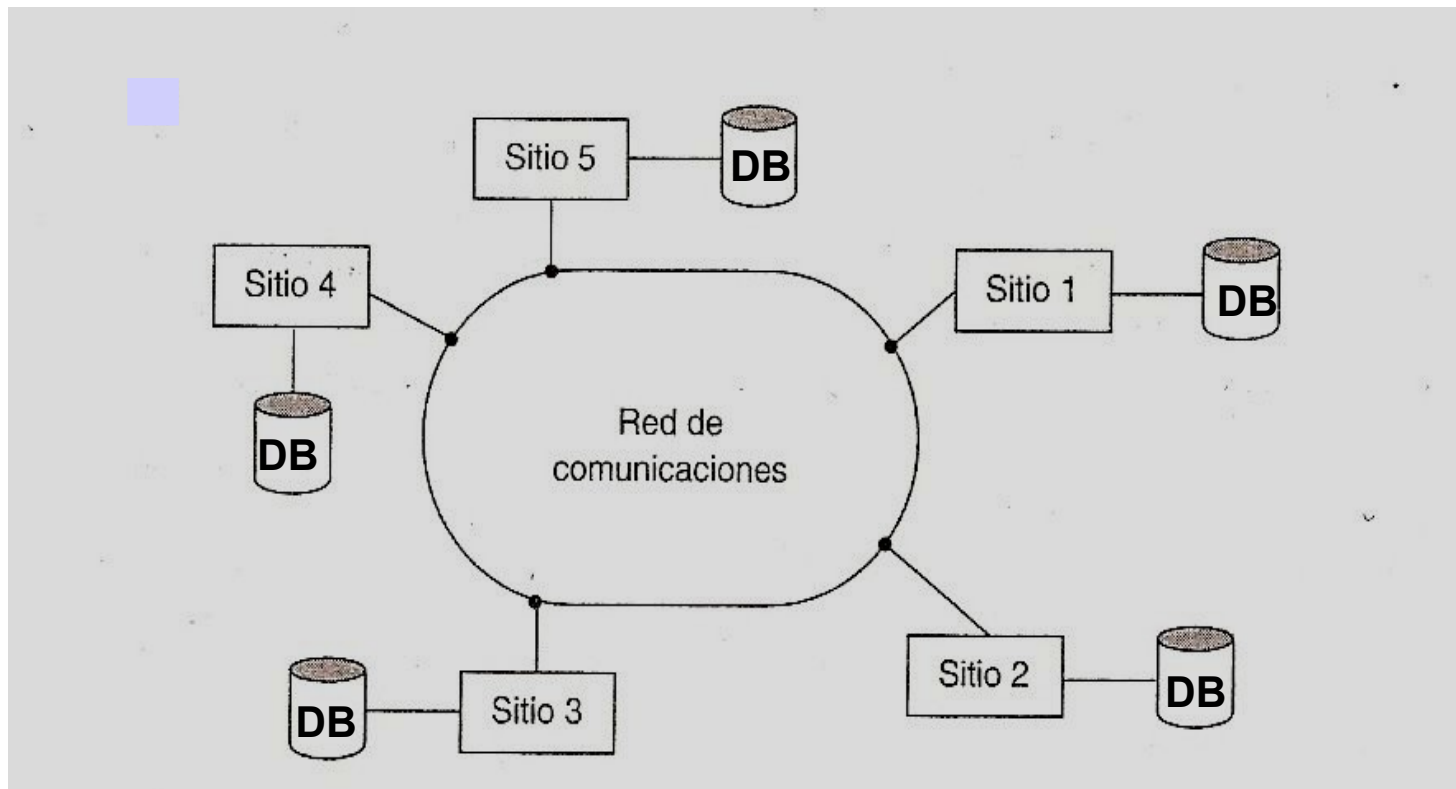


Bases de Datos Distribuidas

Introducción – Arquitecturas Especificas

Análisis de Arquitecturas →

Verdadera Arquitectura de BD distribuida.



Bases de Datos Distribuidas

Introducción – Arquitecturas Especificas

Los datos deben estar físicamente en más de una sede. Cada sede o computadora es un **nodo de la red**. Cada nodo proporciona un **entorno de ejecución de transacciones**, tanto **local**, como **global**.

Esquema lógico global único: Los **datos** han de estar **lógicamente integrados**, para realizar recuperación y actualización tanto en forma local como remota.

En una única operación se puede acceder (recuperar o actualizar) datos que se encuentran en más de una sede.

Bases de Datos Distribuidas

Introducción – Tipos Transacciones

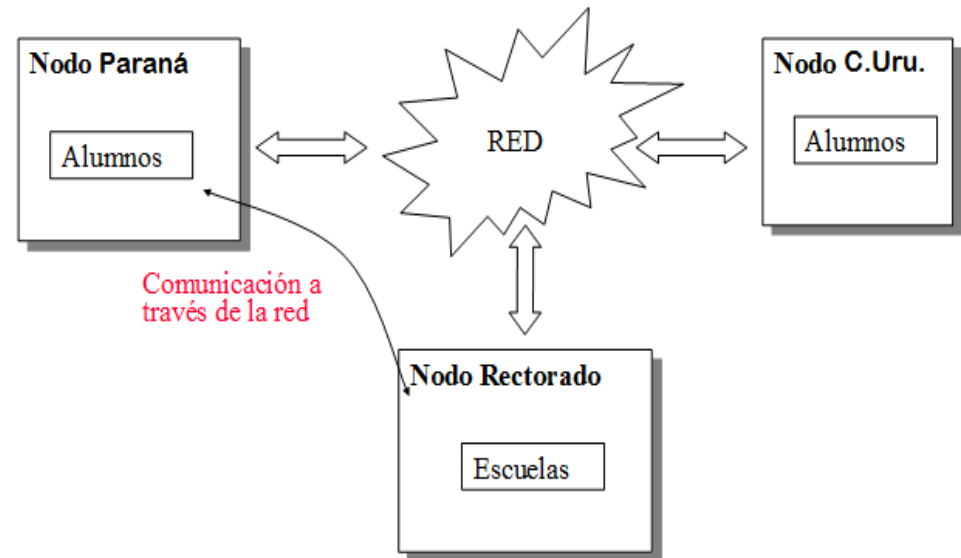
Tenemos 2 tipos posibles de transacciones:

Locales: cuando se accede a los datos de la sede que inició la transacción.

Globales: se accede a datos de sedes distintas al nodo que inició la transacción.

Ejemplo:
Consulta Alumnos Paraná:
local.

Consulta Alumnos Nivel
Universidad:
global.



Bases de Datos Distribuidas

Introducción – Conceptos de Fragmentación

¿Pero que es lo que divido en cada Nodo o Sitio?

Es preciso tomar decisiones respecto a los sitios en los que se almacenarán las “porciones” de la base de datos.

Debemos determinar las “unidades lógicas” de la base de datos que van a ser distribuidas, las “unidades lógicas” se llamarán: **fragmentos**.

Desafios:

¿Cual es la Fragmentación ideal?

¿Que grados de fragmentación se definiran?

Objetivos Alta Performance --> que la mayor parte de las operaciones sean sólo locales, lo cual reducirá el tráfico en la red.

Bases de Datos Distribuidas

Introducción – Conceptos de Fragmentación

Definición

La fragmentación consiste en dividir la relación en fragmentos menores, donde cada uno es una relación, cada fragmento se guarda bajo un DBMS diferente. Tiene como objetivo buscar alternativas para dividir las relaciones en relaciones más pequeñas.

Para ello tenemos las siguientes alternativas lógicas:

- La fragmentación se puede realizar por tuplas individuales (***fragmentación horizontal***)
- O por atributos (***fragmentación vertical***)
- O una combinación de ambas (***fragmentación híbrida***).

Bases de Datos Distribuidas

Introducción – Conceptos de Fragmentación

Ventajas

- Mejorar el rendimiento de las aplicaciones al trabajar con subconjuntos de relaciones.
- Dar una respuesta eficiente a aplicaciones que trabajan con los mismos datos en diferentes nodos
- Permitir el aumento del número de ejecuciones concurrentes.

Bases de Datos Distribuidas

Introducción – Conceptos de Fragmentación

Desventajas

- Aumento de la complejidad del SGBD.
- La comprobación de las restricciones de integridad es mas costosa.

Bases de Datos Distribuidas

Introducción – Conceptos de Fragmentación

Reglas de Fragmentación

Compleitud: La descomposición de una relación R en los fragmentos R_1, R_2, \dots, R_n es completa si y solamente si cada elemento de datos en R se encuentra en algún fragmento.

$$\forall d_i \in \text{en algún } R_j$$

Reconstrucción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , entonces debe existir algún operador (\Join) que permita reconstruir la Relación Original R .

$$R = \Join R_i, \forall R_i \in FR$$

Disyunción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , y el dato d_i está en R_j , entonces, no debe estar en ningún otro fragmento

Fragmentación Horizontal: Intersección Vacía entre los R_i

Fragmentación Vertical: Solo se repiten atributos claves.

Bases de Datos Distribuidas

Introducción – Conceptos de Replicación

¿Que pasa si queremos aumentar la disponibilidad aprovechando que contamos con mas de un SGBD?

Una relación no necesariamente debe estar en un solo sitio/nodo:

Si la relación se guarda, en forma idéntica en mas de un sitio/nodo, estamos haciendo un replicado.

La **replicación** copia y mantiene las tuplas de la relación afectada en uno o mas sitios/nodos.

Bases de Datos Distribuidas

Introducción – Conceptos de Replicación

Ventajas

Disponibilidad: El sistema tiene mas posibilidades de seguir funcionando ante la caída de un nodo.

Aumento de las posibilidades de paralelismo: Varios usuarios de nodos diferentes pueden efectuar consultas sobre la misma relación.

Ademas cuanto mas replicas existan

- mas paralelismo (en consultas)

- menos trafico de red (en consultas).

Bases de Datos Distribuidas

Introducción – Conceptos de Replicación

Desventajas

Aumento de la sobrecarga de las actualizaciones:

El sistema debe asegurar que todas las replicas de una relación sean consistentes.

Se deben propagar todos los cambios a todas las replicas.

Bases de Datos Distribuidas

Introducción - Reglas de Date

Date definió 12 reglas o requerimientos ideales que se pretende que las implementaciones de SGBDD cumplan:

1. Autonomía local
2. No depender de un sitio central
3. Operación continua
4. Independencia de la localización o localidad
5. Independencia de la fragmentación
6. Independencia de la replicación
7. Procesamiento de consultas distribuidas
8. Gestión de Transacciones distribuidas
9. Independencia del hardware
10. Independencia del SO
11. Independencia de la red
12. Independencia de DBMS

Bases de Datos Distribuidas

Introducción - Reglas de Date

1 Autonomía local

Los nodos o localidades de un SD deben ser independientes entre si en el mayor grado posible.

Características de cada nodo

- Tiene su propio DBMS
- El DBMS controla todos los aspectos del nodo
- Las operaciones de acceso a datos locales utilizan sólo recursos locales.
- La administración de los datos es local.
- Hay cooperación entre los nodos para el acceso distribuido de datos.

Bases de Datos Distribuidas

Introducción - Reglas de Date

2 No depender de un sitio central

Todos los sitios/nodos deben ser tratados como iguales

- De existir un sitio central, habría un cuello de botella
- De existir un sitio central, el sistema sería vulnerable, porque una falla haría fallar a todo el sistema
- Para el protocolo de commit de dos fases se necesita un servicio central pero sólo durante la ejecución de una transacción (Coordinador Lógico Centralizado).

Bases de Datos Distribuidas

Introducción - Reglas de Date

3 Operación Continua

Un sistema BDD no debería estar nunca fuera de servicio, debe operar 7x24

Para proporcionar mayor confiabilidad y mayor disponibilidad se requiere:

- Soporte para recuperaciones rápidas de BD
- DBMS tolerante a fallos (con soft y hardware acorde)
- Soporte para backups on line (hot backup), total o incremental. (Requisito indispensable en general actualmente mas alla de Bdd).

Bases de Datos Distribuidas

Introducción - Reglas de Date

4 Independencia de Localización

Los usuarios y las aplicaciones no necesitan conocer la ubicación física de los datos. Actúan como si fuesen locales a ellos.

Sin transparencia local deberían distinguirse los datos locales de los datos remotos.

Punto crítico: el **D**iccionario de **D**atos:

Usuarios y aplicaciones referencian objetos por un alias:

- El **DD** debe mantener una la relación de Objetos-Alias y ubicaciones.
- Un DDBMS debe **mantener y utilizar el DD aún cuando los datos se mueven entre localidades**
- El **DD** debe estar replicado en las localidades y las réplicas deben mantenerse actualizadas.

Bases de Datos Distribuidas

Introducción - Reglas de Date

5 Independencia de Fragmentación de Datos

Los usuarios pueden comportarse como si los datos no estuvieran fragmentados (no necesitan conocer los fragmentos físicos en que está dividida cada colección lógica de datos)

- La fragmentación es necesaria por razones de rendimiento.
- Los datos pueden estar almacenados en la ubicación donde son usados con mayor frecuencia para que la mayoría de las operaciones sean locales y se reduzca el tráfico de la Red.

Bases de Datos Distribuidas

Introducción - Reglas de Date

6 Independencia de Replicación de Datos

El usuario debe comportarse como si los datos no estuvieran replicados. Recordemos que esto significa:

Mejor rendimiento: las aplicaciones operan sobre copias locales en vez de comunicarse con sitios remotos.

Mejor disponibilidad: un objeto replicado está disponible mientras haya al menos una copia.

Desventaja: propagar las actualizaciones.

Bases de Datos Distribuidas

Introducción - Reglas de Date

7 Procesamiento de Consultas Distribuidas

La performance promedio de una consulta debe ser independiente del sitio donde se realiza la consulta

- El SD debe disponer de mecanismos para optimizar las consultas y en especial para reducir la carga de tráfico necesaria.

Bases de Datos Distribuidas

Introducción - Reglas de Date

8 Gestión de Transacciones Distribuidas

El SD debe disponer de mecanismos adecuados para el control de concurrencia y la recuperación de transacciones distribuidas

- Debe mantenerse la atomicidad de las transacciones.
- Control de recuperación de información.
- Control de concurrencia.
- Protocolos utilizado para preservar la atomicidad: **commit en dos o tres fases** los más conocidos.

Bases de Datos Distribuidas

Introducción - Reglas de Date

9 Independencia del Hardware

Es necesario tener la posibilidad de ejecutar el mismo DBMS en diferentes plataformas de Hardware, ya lo plantea o exige Ansi-Sparc (nivel externo, conceptual e interno).

10 Independencia del SO

Es necesario tener la posibilidad de ejecutar el mismo DBMS en diferentes Sistemas Operativos.

El sistema operativo puede ser: Unix, Win, VMS, etc.

Bases de Datos Distribuidas

Introducción - Reglas de Date

11 Independencia de la red

El SD debe poder operar con diferentes redes de comunicaciones.

12 Independencia del DBMS

Debe permitirse la **heterogeneidad**, es decir, que cada sitio pueda funcionar con un SGBD diferente, incluso basado en un modelo de datos diferente, siempre y cuando compartan una **interface común**.

Bases de Datos Distribuidas

Ventajas & Desventajas

Como ya anticipamos esto trae aspectos positivos y negativos, las **ventajas principales** se dividen en:

- Organizativas
 - Adaptación a la organización de la institución (unión de compañías/descentralización), respondiendo a cambios
Ej Sucursales de un Banco.
 - Almacenar los datos donde son generados y/o usados
 - Proporcionar autonomía local, controlándose desde cada nodo.
- Económicas
- Técnicas

Bases de Datos Distribuidas

Ventajas & Desventajas

Como ya anticipamos esto trae aspectos positivos y negativos, las **ventajas principales** se dividen en:

- Organizativas
- Económicas
 - Solución de compromiso entre Costos de Comunicación y de Creación de Pequeños Sitios.
- Técnicas

Bases de Datos Distribuidas

Ventajas & Desventajas

Las **ventajas principales** se dividen en:

- Organizativas
- Económicas
- Técnicas
 - Flexibilidad, acceso desde distintos lugares y por distintas personas a la vez
 - Fiabilidad/disponibilidad, en un determinado intervalo de tiempo. Varios sitios con replicas, menor riesgo x fallos
 - Modularidad
 - Mejora del rendimiento, BD más pequeñas, operaciones de menor volumen
 - Crecimiento incremental, añadiendo poder de procesamiento y almacenamiento

Bases de Datos Distribuidas

Ventajas & Desventajas

Y las **Desventajas**?

- Complejidad del Software, desarrollo de software más costoso, problemas de sincronización, dificultad para conocer la corrección de los algoritmos paralelos, detección de caídas.
- Dependencia de la red de comunicaciones (Relativo)
- Dificultad de diseño, fases adicionales, modelo de fragmentación y replicación.
- Poca madurez de los productos de la Tecnología de BD
- Administración mas compleja, sincronización y coordinación
- Dificultad de cambio, inexistencia de metodologías.
- Requiere Personal mas Especializado

Bases de Datos Distribuidas

Introducción – Diseño BDD

Pasos a seguir para diseñar una BDD:

1. Diseño del **"esquema conceptual"** el cual describe la BD integrada (esto es, todos los datos que son utilizados por las aplicaciones que tienen acceso a las bases de datos). **Artefactos:** Esquema de relaciones y/o DER.

2. Diseño **"físico" de la base de datos**, esto es, mapear el esquema conceptual a las áreas de almacenamiento y determinar los métodos de acceso.

Artefactos: Especificaciones Físicas, dominios tablas e índices.

3. Diseño de la **fragmentación**, este es, determinar la forma en que las relaciones globales se subdividen en fragmentos horizontales, verticales o mixtos.

Artefactos: Esquema de fragmentación.

4. Diseño de la **asignación de los fragmentos**, esto es, determinar la forma en que los fragmentos se mapean a las imágenes físicas de cada componente de BDD.

Artefactos: Esquema de asignación. **Veamos los artefactos.**

Bases de Datos Distribuidas

Diseño BDD - Esquema de Fragmentación

Esquema de Fragmentación:

Se plantea el tipo de fragmentación usada (horizontal o vertical) con el conjunto de relaciones y atributos del Esquema Original (Primer Diseño).

El esquema completo puede reconstruirse mediante **UNION** y **JOIN** según el tipo de fragmentación elegida.

Expresado preferentemente en álgebra relacional.

Nota: Se fragmenta intentando particionar horizontal y verticalmente el esquema de acuerdo con los requerimientos de información del sistema (query requeridos).

Bases de Datos Distribuidas

Diseño BDD - Esquema de Fragmentación

Esquema de Fragmentación: Ejemplo

Docentes (cuil, nombre, domicilio, celular,
 titulo, huelladigitalDer, huelladigitalIzq)
Alumnos (dni, nombre, domicilio, celular, facultad)

Esquema Fragmentación

DocentesRectorado= SELECT cuil, nombre, domicilio,
 titulo FROM Docentes

DocentesFacultad= SELECT cuil, huelladigitalDer,
 huelladigiZq FROM Docentes

AlumnosFcyt= SELECT dni, nombre, domicilio,
celular, facultad FROM Alumnos Where fac= 'fcyt'

AlumnosFHuman= SELECT dni, nombre, domicilio,
celular, facultad FROM Alumnos Where fac= 'fhum'

Bases de Datos Distribuidas

Diseño BDD - Esquema de Asignación

Esquema de Asignación:

- Búsqueda de distribución óptima de los fragmentos *F_i* en los nodos *S_i* (*S_i* = Server *i*ésimo).
- Características deseables:
 - Coste mínimo
 - Almacenamiento
 - Comunicación
 - Máximo rendimiento:
 - Tiempos de respuesta mínimos
 - Capacidad máxima de procesamiento
- Depende del tipo y frecuencia de transacciones.
- Existen algoritmos de asignación que buscan *distribución óptima*.

Nota: Si un fragmento se encuentra en más de un sitio, el fragmento está *replicado*.

Bases de Datos Distribuidas

Diseño BDD - Esquema de Asignación

Esquema de Asignación: Ejemplo

ServerRec

DocentesRectorado= SELECT cuil, nombre, domicilio,
titulo FROM Docentes

ServerFcyt

AlumnosFcyt= SELECT dni, nombre, domicilio,
celular, facultad FROM Alumnos Where fac = 'fcyt'

ServerFHuman

AlumnosFHuman= SELECT dni, nombre, domicilio,
celular, facultad FROM Alumnos Where fac = 'fhum'

Nota: Similar tratamiento hacer con docentes en
fragmentación / asignación a cada Facultad.

Bases de Datos Distribuidas

Introducción – Diseño BDD

Caso de Estudio

En las distintas ciudades de Argentina se desarrollan organizaciones que otorgan información del estado crediticio de las personas físicas. Estas organizaciones prestan un servicio que consiste en transacciones que origina un cliente con el dni y apellido y nombres de la persona, y el sistema genera:

Información filiatoria de la persona: fecha de nacimiento, nombres completos, sexo, domicilio, parientes.

Información laboral de las personas, Empleador, periodo, cargo, sueldo.

Información de créditos vigentes, con los siguientes datos: fecha, acreedor, cantidad de cuotas, cuotas pagas, cuotas impagas, garantes.

Bases de Datos Distribuidas

Introducción – Diseño BDD

Caso de Estudio

Información de créditos cancelados, con los siguientes datos: fecha, acreedor, cantidad de cuotas, monto de cuotas, monto de cuotas, garantes.

Información de incidentes negativos, moras, atrasos, deudas, con los siguientes datos: fecha, cuit y nombre empresa informante, deuda, concepto de la deuda.

Ademas de la información de personas físicas y su información, estas organizaciones manejan las siguientes relaciones:

- Clientes: Cuit, razonsocial, fecha_inscripcion, domicilio, categoria_iva.
- Empresas Informantes (No Clientes): Cuit, razonsocial, fecha_inscripción, domicilio, categoria_iva.

Bases de Datos Distribuidas

Introducción – Diseño BDD

Caso de Estudio

Se desea realizar un sistema distribuido y se contrata el estudio de factibilidad a una consultora, que como parte del trabajo debe entregar una fragmentación por organización de los datos, proponiendo replicación para aumentar las posibilidades de paralelismo en las relaciones mas estáticas de menos volatilidad de sus filas.

Como requisito funcional se plantea incluir el aspecto temporal de cada tipo de relación a efectos de no perder información, por la importancia que tiene la historia en el sistema.

Téngase en cuenta ademas que se puede conformar una relación con la información de las organizaciones de informes con :

- Cuit, razonsocial, fecha_inscripcion, domicilio, localidad y provincia, categoria_iva.

Bases de Datos Distribuidas

Introducción – Diseño BDD

Caso de Estudio

Las organizaciones/empresas de información se distribuyen en el país a razón de una por provincia. Cada Cliente solicita informes a la empresa de informaciones de su provincia, como así también notifican de créditos otorgados, pagos de créditos y cancelaciones.

Cada empresa de informes es dueña de sus datos, pero ofrecen informes provinciales o nacionales, para el caso de informes nacionales deben usar la información de la persona que reside en empresas de otras provincias. A fin de mes se realiza un clearing compensando los datos aportados entre todas las provincias y se facturan las diferencias entre organizaciones.

Finalizando las estructuras se tiene que un informe involucra, a una empresa que es la proveedora, un cliente (pers.jurid.) y refiere a una única persona:

Cuit_empresa_inf, razonsoc_empresa_inf, cuit_cliente, razonsoc_cliente, dni e inf de persona.

Bases de Datos Distribuidas

Introducción – Diseño BDD

Caso de Estudio – Caso de Uso Informe

Por otro lado se plantea el caso de uso principal denominado “*consultar informe comercial*” descrito a alto nivel como sigue:

- Cliente: ingresa dni y nombre de la persona a buscar.
- Sistema: responde otorgando la información básica de la persona, para validarlo visualmente.
- Cliente: informa si pretende un informe comercial local, o global integrado.
- Sistema responde con el informe solicitado de la persona, registrando la consulta, agregando el costo a la cta cte del cliente entre otras acciones.

RESUMEN

Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Transparencias
- **Fragmentación: Un analisis mas profundo.**
- Replicación: mas sobre este tema.
- Procesamiento Distribuido de Consultas
- Transacciones Distribuidas

Bases de Datos Distribuidas

Fragmentación - Historia

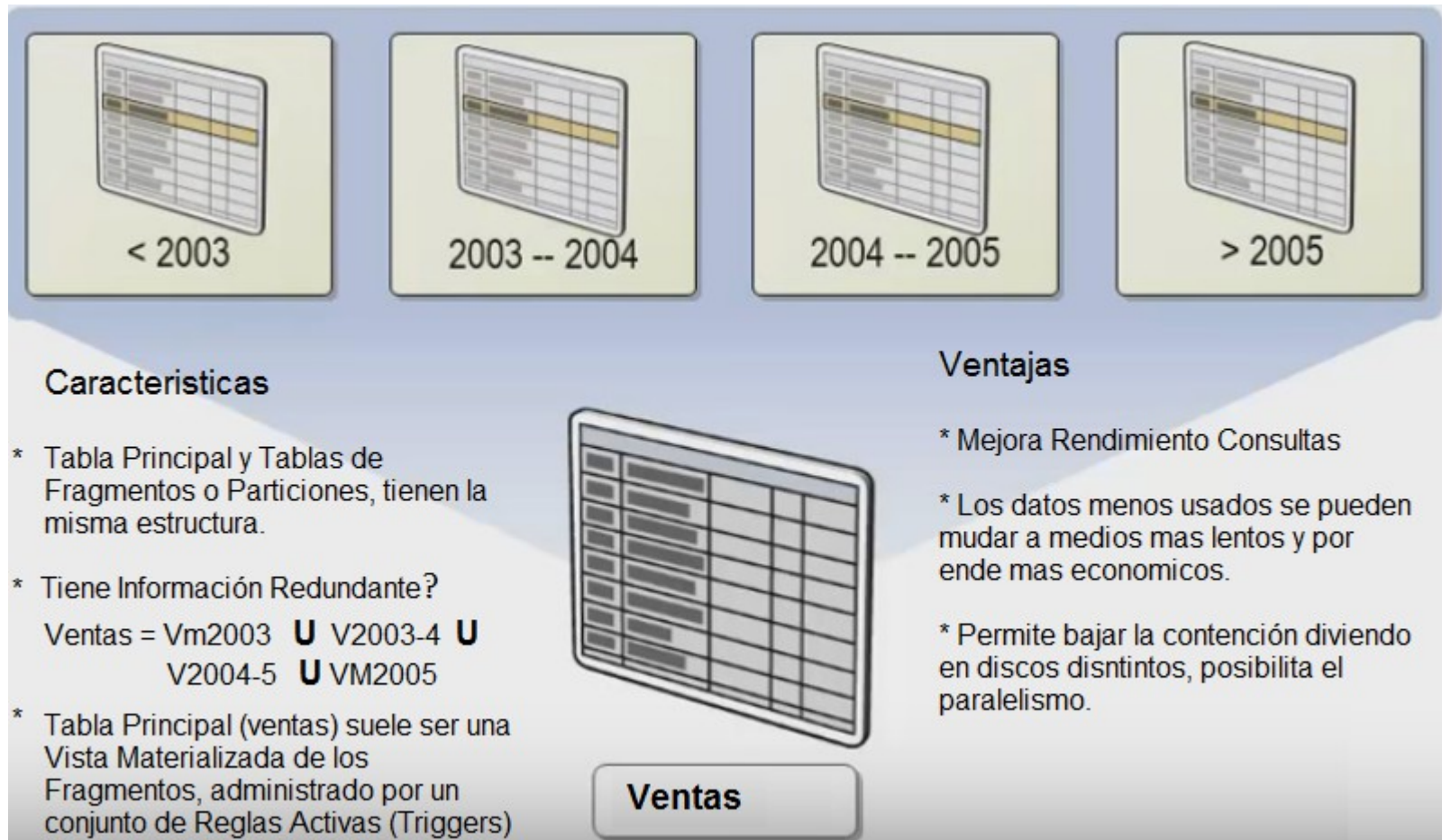
Esta idea nació con las Bases de datos Centralizadas y luego se utilizó en las BDD.

- **Horizontal:** Esta modalidad consiste en tener varias tablas con las mismas columnas en cada una de ellas y distribuir la cantidad de filas en estas tablas (generalmente se parte separando los datos por años, meses, u otro criterio).
- **Vertical:** Esta modalidad se aplica por ejemplo cuando tenemos una columna de tipo BLOB con una fotografía o un texto muy largo (CLOB), que tiene baja probabilidad de ser recuperado con el resto de los datos y decidimos ponerlo en otra tabla asociada con la misma clave primaria.

Bases de Datos Distribuidas

Frag. Horizontal - Idea → Tablas Particionadas

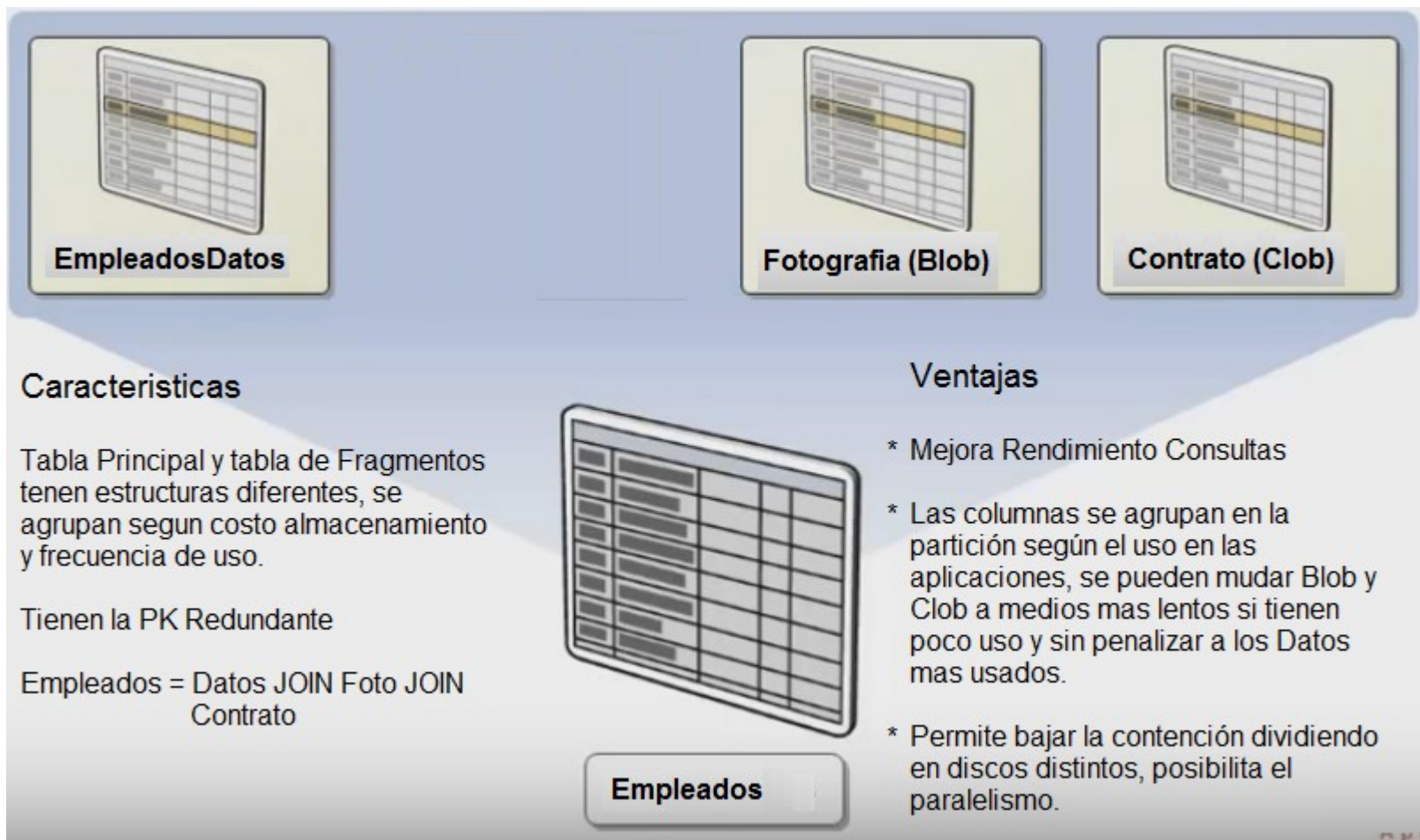
Idea **Antigua**: Datos particionados horizontalmente por rango.



Bases de Datos Distribuidas

Frag. Vertical - Idea → Tablas Particionadas

Idea **Antigua**: Datos particionados verticalmente por frecuencia uso.

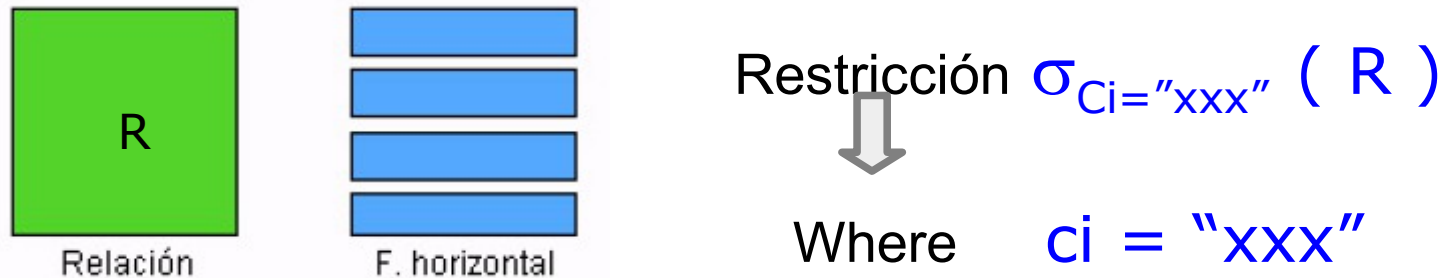


Bases de Datos Distribuidas

Fragmentación Horizontal

La Fragmentación tanto Horizontal como Vertical surgen como una técnica general de minimizar la contención y achicar los arboles de índices.

Fragmentar Horizontalmente una relación, implica realizar una selección sobre las tuplas, ocupando la operación de restricción/selección sobre uno o mas atributos:



Es decir cada fragmento dará origen a una nueva relación (una porción o subconjunto de la relación original).

La relación R se divide en los subconjuntos R1, R2, ..., Rn

Bases de Datos Distribuidas

Fragmentación Horizontal - Ejemplos

Ejemplo: Considere la Relación Alumnos

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Alumnos1 Escuela = 'CIME'

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME

Alumnos2 Escuela = 'EISIC'

Jno	NOMBRE	NOTA	ESCUELA
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Bases de Datos Distribuidas

Fragmentación Horizontal - Ejemplo

Ej:

Alumnos(Jno, nombre, nota, escuela)

Se aplica restricción $\sigma_{\text{escuela}=\text{"CIME"}}(\text{Alumnos})$

Select * from Alumnos where escuela = 'CIME'



Da origen a Alumnos1

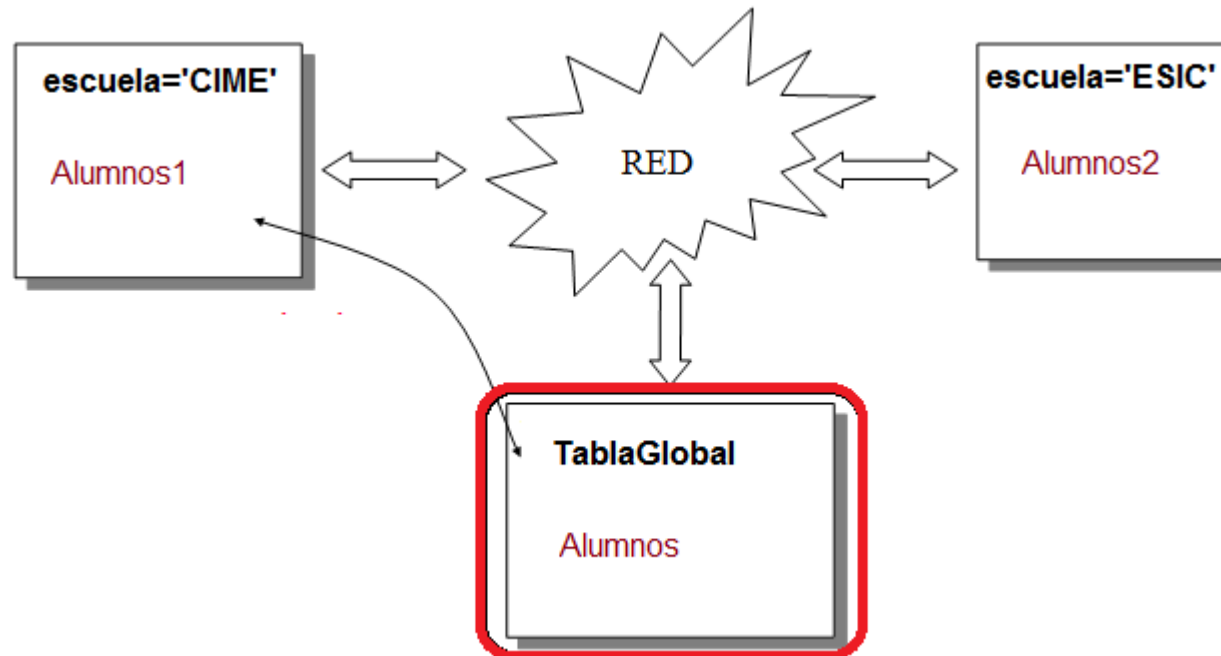
Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME

Bases de Datos Distribuidas

Fragmentación Horizontal - Ejemplos

Lo que el usuario ve:

Alumnos(Jno, nombre, nota, escuela)



Bases de Datos Distribuidas

Fragmentación Horizontal - Reconstrucción

Reconstrucción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , entonces debe existir algún operador que permita reconstruir la Relación Original R.

$$R = R_i, \forall R_i \in FR$$

El operador sería $U \rightarrow$ unión



Alumnos = Alumnos1 U Alumnos2

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Bases de Datos Distribuidas

Fragmentación Horizontal

Existen en realidad 2 tipos de Fragmentación Horizontal:

- **Fragmentación Horizontal Primaria:** Aplicado a relaciones que tienen baja dependencia con otras.
- **Fragmentación Horizontal Derivada:** Consiste en dividir una relación partiendo de los predicados definidos sobre alguna otra, debido a que la relación R depende de la relación Q , sobre cuyos atributos está definido el predicado de la fragmentación.

Bases de Datos Distribuidas

Fragmentación Horizontal Derivada

Las tres entradas necesarias para desarrollar la fragmentación horizontal derivada son las siguientes:

- El conjunto de fragmentos de la relación principal,
- La relación miembro o secundaria
- El conjunto de predicados resultados de aplicar el “join” entre la relación principal y secundaria.

Es decir la fragmentación de la tabla principal, se aplica a tablas secundarias, o bien se debe partir de una fragmentación horizontal primaria para derivarla.

Bases de Datos Distribuidas

Fragmentación Derivada - Ejemplo

Ejemplo: Considere las relaciones MATERIA y ALUMNO

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
1	Análisis Matemático	EISIC	4	1
2	Sistemas Operativos	EISIC	6	3
3	Programación II	CIME	6	2
4	Tecnología Eléctrica	EISIC	4	2
5	Técnicas de Aprendizaje	CIME	4	1
6	Dibujo Mecánico	CIME	6	3

Jno	NOMBRE	MATERIA	NOTA
J1	LUIS YANEZ	Sistemas Operativos	8
J2	ERIKA QUIROZ	Dibujo Mecánico	8
J3	DANIEL MURILLO	Técnicas de Aprendizaje	9
J4	MARIA JOSE MENDEZ	Análisis Matemático	10
J5	ADONIS PABON	Programación II	9

Bases de Datos Distribuidas

Fragmentación Derivada - Ejemplo

Se quiere fragmentar horizontalmente basado en la escuela en la cual el alumno está matriculado. La escuela no es un atributo de ALUMNO.

Escuela := EISIC (Fragmentación Horizontal primaria)

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
1	Análisis Matemático	EISIC	4	1
2	Sistemas Operativos	EISIC	6	3
4	Tecnología Eléctrica	EISIC	4	2

Fragmentación Horizontal derivada

Jno	NOMBRE	MATERIA	NOTA
J1	LUIS YANEZ	Sistemas Operativos	8
J4	MARIA JOSE MENDEZ	Análisis Matemático	10

Escuela = CIME (Fragmentación Horizontal primaria)

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
3	Programación II	CIME	6	2
5	Técnicas de Aprendizaje	CIME	4	1
6	Dibujo Mecánico	CIME	6	3

Fragmentación Horizontal derivada

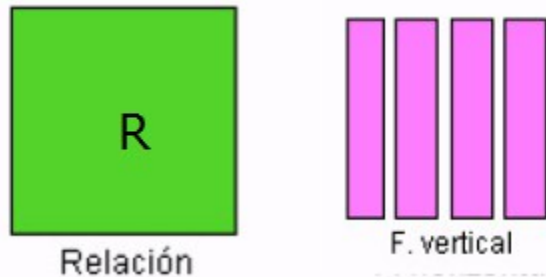
Jno	NOMBRE	MATERIA	NOTA
J2	ERIKA QUIROZ	Dibujo Mecánico	8
J3	DANIEL MURILLO	Técnicas de Aprendizaje	9
J5	ADONIS PABON	Programación II	9

Bases de Datos Distribuidas

Fragmentación Vertical

También la Fragmentación Vertical surge como técnica de las bases de datos centralizadas.

Fragmentar Verticalmente una relación, digamos, se obtiene realizando una descomposición o división por atributos, de acuerdo a la operación proyección sobre estos atributos:



Proyección $\pi_{R_i} (R)$

La Fragmentación significa dividir el conjunto de atributos en subconjuntos donde solo se repite la clave primaria o restricción unique.

Bases de Datos Distribuidas

Fragmentación Vertical - Ejemplo

Ej:

Alumnos(Jno, nombre, nota, escuela)

Se proyecta $\pi_{\text{nombre, escuela}}$ (Alumnos)

Select **Jno**, **nombre**, **escuela** from Alumnos

Da origen a Alumnos1

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME

Se proyecta π_{nota} (Alumnos)

Select **Jno**, **nota** from Alumnos

Da origen a Alumnos2

Jno	NOTA
J1	8
J2	8
J3	9
J4	10
J5	9

Bases de Datos Distribuidas

Fragmentación Vertical - Ejemplo

Ejemplo: Considere la Relación Alumnos

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Información de Notas

Jno	NOTA
J1	8
J2	8
J3	9
J4	10
J5	9

Información de Nombres y Escuelas

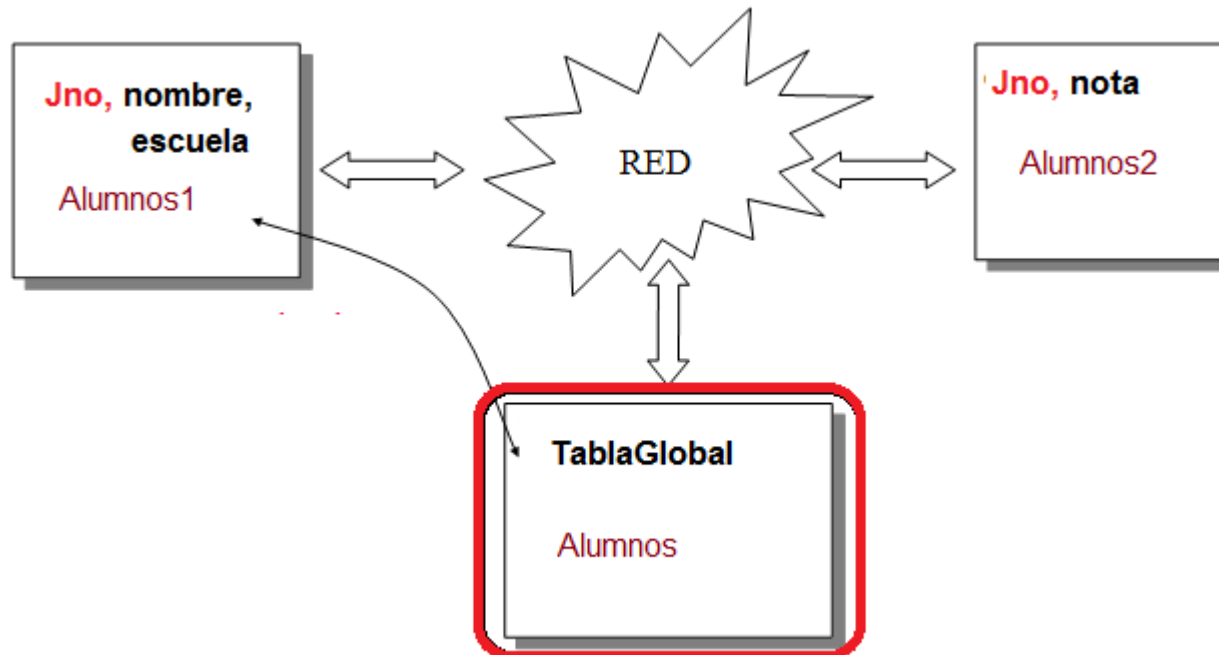
Jno	NOMBRE	ESCUELA
J1	LUIS YANEZ	CIME
J2	ERIKA QUIROZ	CIME
J3	DANIEL MURILLO	EISIC
J4	MARIA JOSE MENDEZ	EISIC
J5	ADONIS PABON	EISIC

Bases de Datos Distribuidas

Fragmentación Vertical - Ejemplo

Idem que Frag Horizontal, lo que el usuario ve:

Alumnos(Jno, nombre, nota, escuela)



Bases de Datos Distribuidas

Fragmentación Vertical - Reconstrucción

Reconstrucción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , entonces debe existir algún operador que permita reconstruir la Relación Original R.

$$R = R_i, \forall R_i \in FR$$

El operador es

 -> reunión ¿Pero cual de ellas?



Alumnos = Alumnos1 a1  a1.Jno = a2.Jno Alumnos2 a2

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Bases de Datos Distribuidas

Frag Vertical – Información Necesaria

El principal dato necesario relativo a las aplicaciones es la frecuencia de acceso.

Sea $Q = \{q_1, q_2, \dots, q_q\}$ el conjunto de consultas de usuario (aplicaciones) que funcionan sobre una relación $R(A_1, A_2, \dots, A_n)$.

Pero en gral estos no son suficientes para desarrollar la base de la escisión y la fragmentación de los atributos, ya que estos valores no representan el peso de las frecuencias de la aplicación. La dimensión de esta **frecuencia** puede incluirse en la definición de la medida de los **atributos afines** $afd(A_i, A_j)$, la cual mide el límite entre dos atributos de una relación de acuerdo a cómo las aplicaciones acceden a ellos.

Bases de Datos Distribuidas

Fragmentación Vertical

Existen dos enfoques heurísticos para la fragmentación vertical de relaciones:

- **Agrupación:** comienza asignando cada atributo a un fragmento, y en cada paso, junta algunos de los fragmentos hasta que satisface un determinado criterio.
- **Escisión:** a partir de la relación se deciden que fragmentos resultan mejores, basándose en las características de acceso de las aplicaciones a los atributos.

La escisión genera fragmentos no solapados mientras que la agrupación normalmente produce solapados.

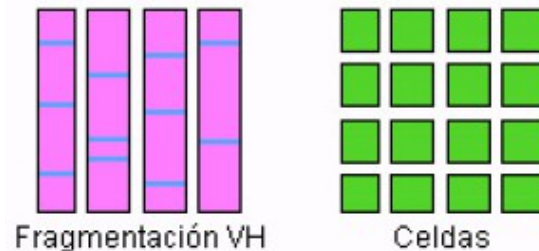
Por este motivo la **Escisión** es mas apropiada para BDD.

Bases de Datos Distribuidas

Fragmentación Mixta o Híbrida

Fragmentación Mixta o Híbrida

Este tipo de fragmentación surge cuando los dos tipos anteriores se combinan. En tal caso, la relación original puede reconstruirse aplicando las operaciones de **Unión** y **Reunión** en el orden apropiado.



Bases de Datos Distribuidas

Fragmentación Mixta o Híbrida

Se puede llevar a cabo de 3 maneras:

- Desarrollando primero la fragmentación vertical y, posteriormente, aplicando la fragmentación horizontal sobre los fragmentos verticales (denominada partición VH),
- Aplicando primero una división horizontal para luego, sobre los fragmentos generados, desarrollar una fragmentación vertical (llamada partición HV),
- De forma directa considerando la semántica de las transacciones

RESUMEN

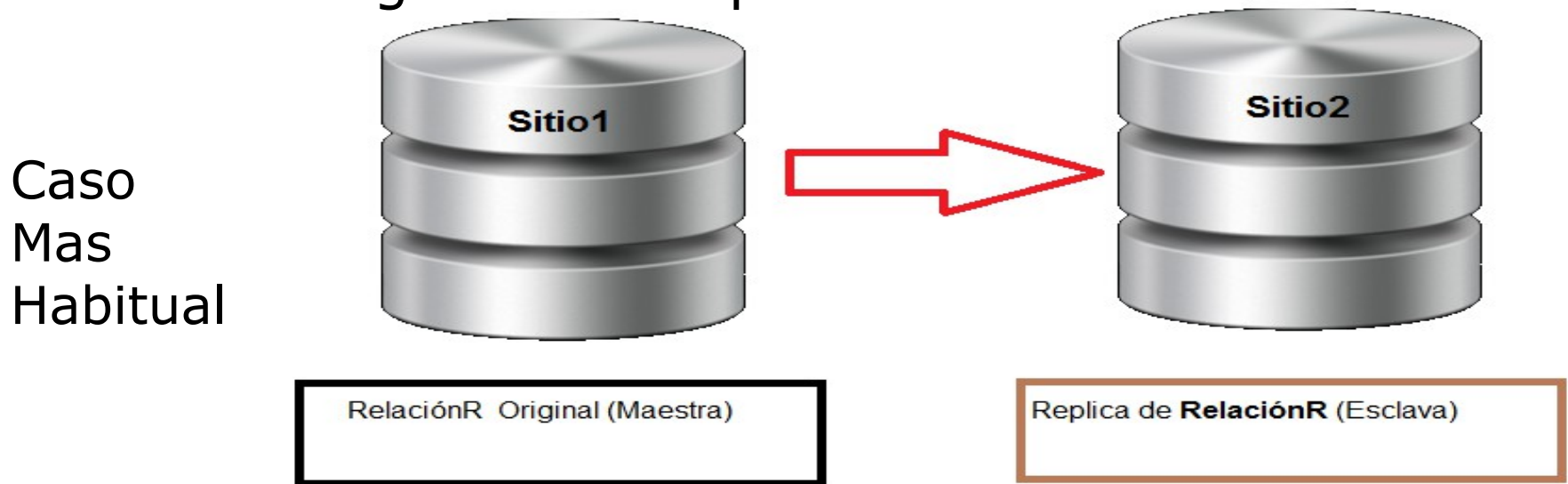
Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Transparencias
- Fragmentación: Un analisis mas profundo.
- **Replicación: mas sobre este tema.**
- Procesamiento Distribuido de Consultas
- Transacciones Distribuidas

Bases de Datos Distribuidas

Replicación – Alcance de Replicación

Antes de analizar mas profundamente este tema, definamos algunos conceptos:



Relación Maestra: Relación Original.

Relación Esclava: Replica de Relación Original.

Nota: Algunos casos pueden tener solo varios maestros, o un maestro y varios esclavos.

Bases de Datos Distribuidas

Replicación – Alcance de Replicación

Replicación Completa: Toda la BD es replicada en todos los nodos. Esto **mejora la disponibilidad** al extremo, la BD puede seguir funcionando con un solo sitio activo.

Mejora el rendimiento de consultas, ya que todas las consultas pasan a ser locales.

La desventaja de la replicación completa es que **penaliza las actualización**, una actualización deberá realizarse en todas y cada una de las copias de la BD a fin de mantener la consistencia.

Bases de Datos Distribuidas

Replicación – Alcance de Replicación

Sin Replicación: Caso extremo a la replicación total, solo hay una copia de cada fragmento en un único sitio.

Todos los fragmentos son disjuntos, con excepción de la porción de claves primarias en fragmentos verticales o mixtos.

Entre estos dos extremos (replicación nula o total), existe una amplia gama de **replicación parcial** de los datos.

Bases de Datos Distribuidas

Replicación – Alcances de Replicación

Replicación Parcial: Solo se seleccionan para replicar algunas partes de la BD en algunos sitios.

Esta replicación se define a través del **esquema de replicación**, este esquema tiene una descripción declarativa de la replicación de los fragmentos.

Cada copia o replica de un fragmento se debe asignar a un determinado sitio del sistema distribuido. Este proceso se denomina Distribución o **Asignación de los Datos**.

Bases de Datos Distribuidas

Replicación – Tipos de Replicación

Solo Lectura (read only): Replicas de sólo lectura cuyos datos se refrescan a intervalos especificados. Se usan basicamente por seguridad (backup) y alta disponibilidad. Concepto Maestro-Esclavo.

Actualizables (updateable): Se permite la modificación de datos sobre la instantánea. Estas modificaciones se propagan hasta tabla maestra (versión original de la relación).

Replicación avanzada: replicación de los datos en varios sitios maestros. Se puede actualizar una tabla en cualquiera de los sitios maestros y la actualización se propaga al resto de los sitios.

Bases de Datos Distribuidas

Replicación – Metodología

En Terminos Metodologicos Practicos

La elección de sitios y el grado de replicación dependen de los objetivos en cuanto a rendimiento y disponibilidad del sistema y de los tipos y frecuencias de transacciones previstas para cada sitio.

Encontrar una solución óptima, o incluso buena, para la asignación y replicación de datos distribuidos es un problema de optimización muy complejo.

Bases de Datos Distribuidas

Replicación y Fragmentación

Las técnicas de réplica y fragmentación se pueden aplicar a la misma relación de partida.

Un fragmento se puede replicar, es decir se combinar potenciando las posibilidades de efectuar consultas locales.

Consecuencias:

- Mayor Disponibilidad
- Mayor Seguridad
- Seguimos penalizando las actualizaciones.

RESUMEN

Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Fragmentación: Un analisis mas profundo.
- Replicación: mas sobre este tema.
- **Diccionario Global o Extendido**
- Procesamiento Distribuido de Consultas
- Transacciones Distribuidas

Bases de Datos Distribuidas

Servicios de una BDD

Comentamos que una Solución de BDD que constan de:

- Conjunto de BD de cada nodo (BD locales)
- Red de Comunicaciones
- SGBDD: principalmente, el cual incluye las funciones:
 - Diccionario o Directorio Global o Extendido
 - Optimizador Distribuido.
 - Gestión de las Transacciones distribuidas.

Bases de Datos Distribuidas

Diccionario Global o Extendido

Definición Conceptual

Un Diccionario de BDD exige implementar mas capacidades que un Diccionario de BD no Distribuida.

Administra el esquema de distribución, localización o ubicación, replicación y seguridad (usuarios / roles / permisos) de una BDD.

Es decir este elemento hace posible la transparencia de una BDD.

Bases de Datos Distribuidas

DDG y las Transparencias

Transparencia de la Fragmentación No se exige a los usuarios que conozcan el modo en que se ha fragmentado la relación.

Transparencia de la Replicación Los usuarios “ven” a cada objeto como único.

Transparencia de la Ubicación o Localización
No se exige a los usuarios que conozcan la ubicación física de los datos.

Bases de Datos Distribuidas

Diccionario Global o Extendido

¿Pero como se puede hacer de estos requerimientos una implementación real?

La respuesta a esto es un problema del Diccionario Global Extendido, trabaja a modo “Servicio de nombres”.

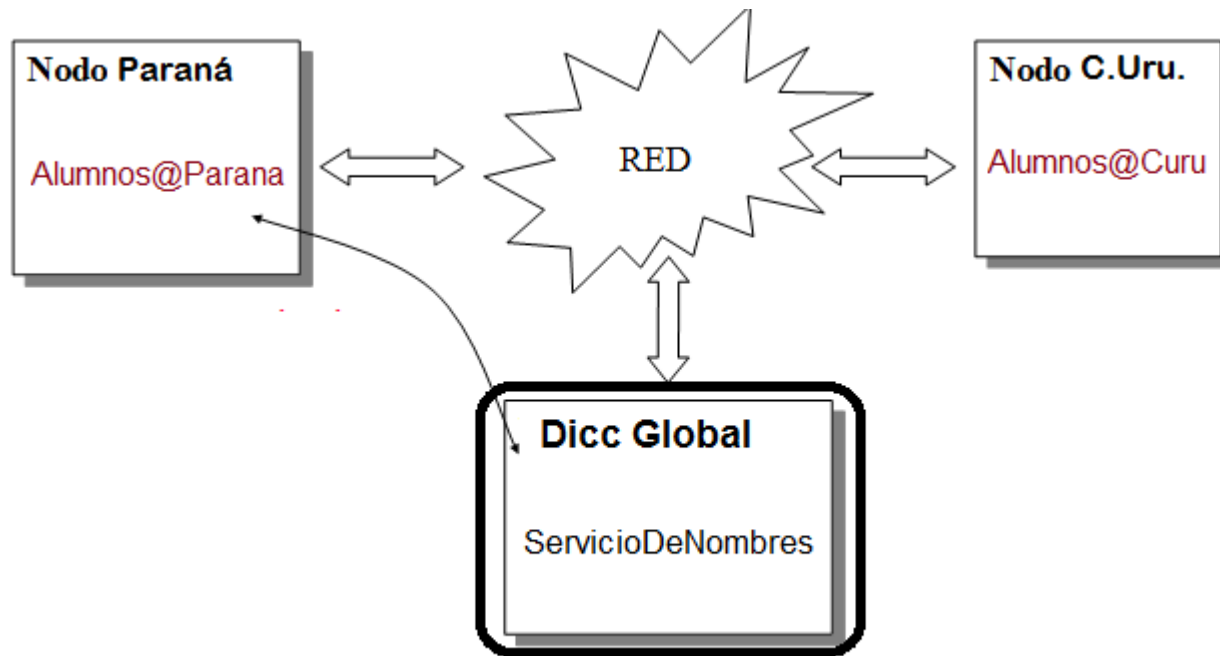
Cada Sitio/Nodo se registra en el Servicio de nombres del diccionario.

Bases de Datos Distribuidas

Diccionario Global o Extendido

Prácticamente

Una relación **Alumnos**, para sitios **Parana** y **Curu**.

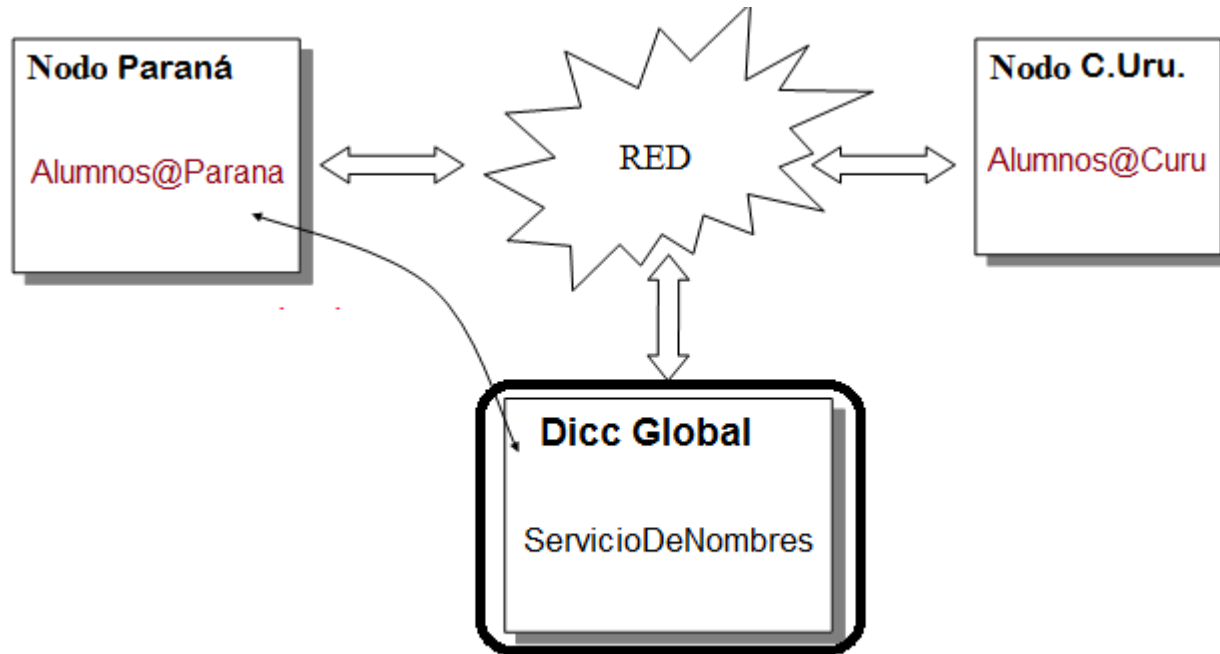


Alumnos = Alumnos@Parana U Alumnos@Curu

Bases de Datos Distribuidas

Diccionario Global o Extendido

Alias y Sinónimos Permite crear un Alias o sinónimo para Alumnos@Parana → AlumnosPna



Alumnos = Alumnos@Parana U Alumnos@Curu

Bases de Datos Distribuidas

Dioccionario Global o Extendido

Información que administra

Información habitual de un diccionario o catálogo:

- Relaciones (base: tablas y virtuales: vistas)
- Índices
- Usuarios
- etc

Información de control y objetivos:

- transparencia de fragmentación
- transparencia de réplica
- transparencia de localización

Bases de Datos Distribuidas

Dioccionario Global o Extendido

Almacenamiento / Alternativas

- **Centralizado.** Viola regla 2: “no dependencia de un sitio central”.
- **Réplicas en cada lugar:** Sumamente costoso.
- **Catálogo dividido entre todos lugares:** Operaciones remotas costosas.
- **Catálogo dividido y una copia global en un sitio:** Viola 2.

Conclusión: todos estos métodos tienen problemas.

Bases de Datos Distribuidas

Dioccionario Global o Extendido

Analicemos el problema a estudiar, que es R^* ?

R^* es la denominación que usaremos para una relación distribuida.

Esta relación R^* , es administrada íntegramente por el Diccionario Distribuido de Datos.

La relación distribuida estará formada por los fragmentos de relación asignados a cada nodo.

$R^* =$ Se integra por R_1, R_2, \dots, R_n .

Bases de Datos Distribuidas

Dioccionario Global o Extendido

Nominación de objetos en R*:

- Nombre de impresión (externo): Lo usan los usuarios.
 - Nombre de sistema: Identificador único interno.
 - ✓ Id. del creador (owner).
 - ✓ Id. del nodo actual (id de nodo actual).
 - ✓ Nombre local del objeto.
 - ✓ Id. del lugar de nacimiento (id nodo de tupla).
- PEPE@MADRID.**PIEZAS@ALICANTE**
- Sinónimos:
CREATE SYNONYM ALI_PIEZAS FOR PEPE@MADRID.**PIEZAS@ALICANTE**
 - Cada sitio mantiene:
 - Copia total de la tabla de sinónimos.
 - Localización actual de un objeto nacido aquí.
 - Relación de objetos actualmente aquí.

R: Versión distribuida*

Bases de Datos Distribuidas

Dioccionario Global o Extendido

Proceso de Consulta en R*:

- Consulta del objeto en la tabla de sinónimos
 - Consulta remota al lugar de nacimiento para conocer lugar actual del objeto.
 - Consulta remota al lugar actual para obtener el objeto.
- (*TOTAL = 3 mensajes, 2 de ellos remotos*)

Proceso de Migración en R*:

- Borrar objeto del lugar origen
- Insertar objeto en lugar destino
- Actualizar en lugar de nacimiento

RESUMEN

Introducción a las Bases de Datos Distribuidas:

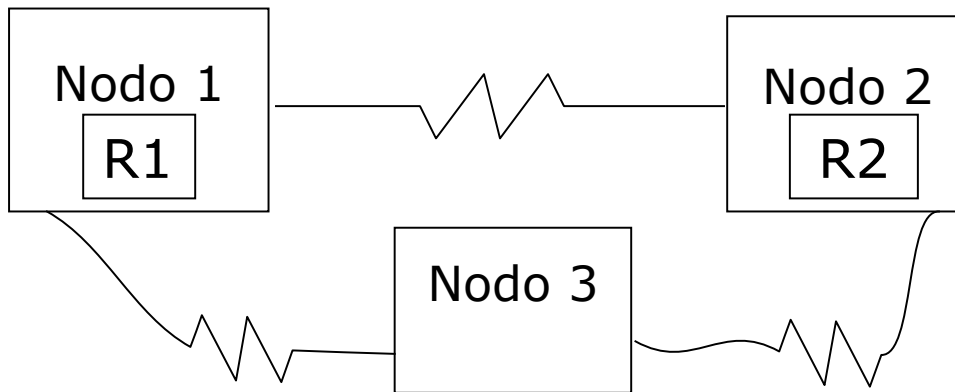
- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Transparencias
- Fragmentación: Un analisis mas profundo.
- **Procesamiento Distribuido** de Consultas
- Transacciones Distribuidas

Bases de Datos Distribuidas

Procesamiento Distribuido de Consultas

Proceso Básico de Optimización:

Consiste en evaluar 3 componentes de costo



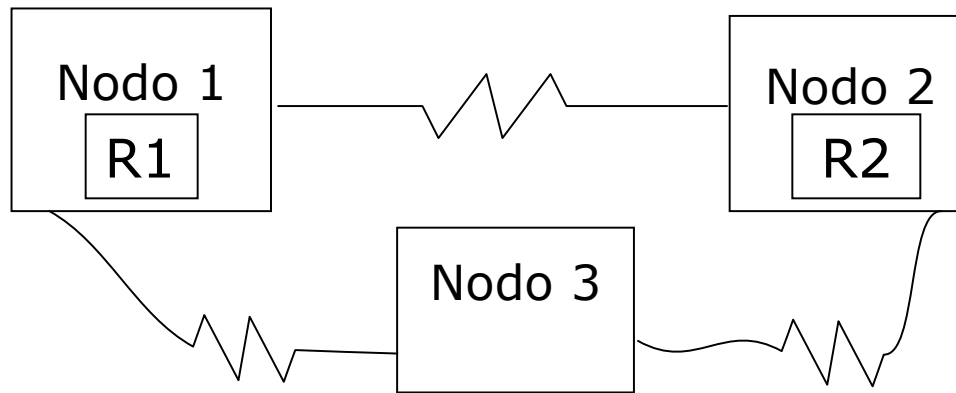
Costo total = costo I/O +
costo Procesamiento +
costo de Comunicaciones

Bases de Datos Distribuidas

Procesamiento Distribuido de Consultas

Si el costo de I/O y de Procesamiento es igual para todos los nodos:

Consiste en evaluar el peso de las comunicaciones de cada alternativa



Consulta en Nodo3 R1 U R2 \Rightarrow <decisión>

R1 viaja a N2
viaje Res a N3

R2 viaja a N1
viaje Res a N3

R1, R2 viajan a N3

Bases de Datos Distribuidas

Procesamiento Distribuido de Consultas

Se estudia el costo de las comunicaciones.

Objetivo:

la reducción de la cantidad de datos transferidos

Optimización mediante operación de *semijoin*

Obtenemos solo los atributos de R que participan en el semijoin (por izquierda), usando natural join:

$R \bowtie S = \Pi_{a1, \dots, an} (R * S)$; donde $a1, \dots, an$ son atributos de R

$\Join \rightarrow$ semijoin por derecha

Bases de Datos Distribuidas

Ej de Consulta Distribuida

Nodo1: **Empleado** -> 10.000 tuplas. (100 byte x tupla)

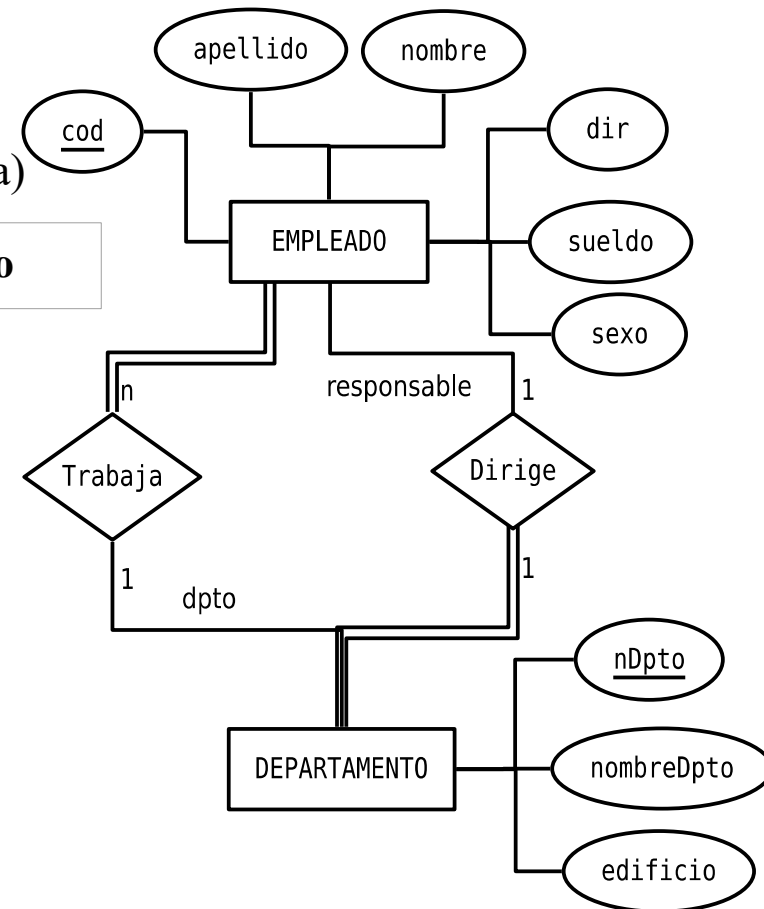
Nombre	Apellido	COD	Dir	Sexo	Sueldo	fechaNac.	Dpto.
--------	----------	-----	-----	------	--------	-----------	-------

Nodo2: **Departamento** -> 100 tuplas. (35 byte x tupla)

NombreDpto	NDpto	Responsable	Edificio
------------	-------	-------------	----------

“Por cada empleado, obtener el nombre del empleado y el nombre del departamento al que pertenece”

La consulta se realiza desde un tercer nodo (nodo 3).



Bases de Datos Distribuidas

Ej de Consulta Distribuida

Nodo1: Empleado

Nombre	Apellido	COD	Dir	Sexo	Sueldo	fechaNac.	Dpto.
--------	----------	-----	-----	------	--------	-----------	-------

10.000 tuplas.

Cada tupla tiene 100 bytes de longitud.

El campo COD tiene 9 bytes de longitud.

El campo Dpto tiene 4 bytes de longitud.

El campo Nombre tiene 15 bytes de longitud.

El campo Apellido tiene 15 bytes de longitud.

Etc.

Tamaño de la relación: $100 * 10.000 = 10^6 \text{bytes}$

Bases de Datos Distribuidas

Ej de Consulta Distribuida

Nodo2: Departamento

NombreDpto	NDpto	Responsable	Edificio
-------------------	--------------	--------------------	-----------------

100 tuplas.

Cada tupla tiene 35 bytes de longitud.

El campo NombreDpto tiene 10 bytes de longitud.

El campo NDpto tiene 4 bytes de longitud.

El campo Responsable tiene 9 bytes de longitud.

Tamaño de la relación: $35 * 100 = 3500$ bytes

Bases de Datos Distribuidas

Ej de Consulta Distribuida

*“Por cada empleado, obtener el **nombre del empleado** y el **nombre del departamento** al que pertenece”*

Q1₍₁₎: Π Nombre,Apellido,NombreDPto (EMPLEADO $\bowtie_{ndpto=dpto}$ DEPARTAMENTO)

```
SELECT nombre, apellido, nombredpto  
FROM empleado JOIN departamento  
ON ndpto = dpto
```

Se consulta desde nodo3 que no tiene los datos.

Cantidad de Filas:	10.000 tuplas.
Cantidad de Byte de cada tupla:	40 bytes.
Tamaño del Resultado:	400.000 bytes.

Existen tres alternativas para resolver la consulta.

Bases de Datos Distribuidas

Ej de Consulta Distribuida

*“Por cada empleado, obtener el **nombre del empleado** y el **nombre del departamento** al que pertenece”*

Primera alternativa:

Transferir, tanto la relación EMPLEADO, como la relación DEPARTAMENTO al nodo respuesta (nodo3) y realizar allí mismo la operación de join.

En éste caso se transfieren:

1.000.000 + 3.500 = 1.003.500 bytes.

¿Se podría realizar otra alternativa para este plan query?

Bases de Datos Distribuidas

Ej de Consulta Distribuida

*“Por cada empleado, obtener el **nombre del empleado** y el **nombre del departamento** al que pertenece”*

Segunda alternativa:

Transferir la relación EMPLEADO al nodo2, ejecutar el join en este nodo y enviar el resultado al nodo3.

En éste caso se transfieren:

1.000.000 + 400.000 (resultado) = 1.400.000 bytes

Bases de Datos Distribuidas

Ej de Consulta Distribuida

*“Por cada empleado, obtener el **nombre del empleado** y el **nombre del departamento** al que pertenece”*

Tercera alternativa:

Transferir la relación DEPARTAMENTO al nodo1, ejecutar el join en este nodo y enviar el resultado al nodo3.

En éste caso se transfieren:

3.500 + 400.000 (resultado) = 403.500 bytes.



Opción Optima

Bases de Datos Distribuidas

Ej de Consulta Distribuida

“Por cada Dpto, obtener el nombre departamento y el de su director”

Q2: Π NombreDpto, Nombre, Apellido (DEPARTAMENTO $\bowtie_{\text{responsable=cod}}$ EMPLEADO)

SELECT nombre AS nomdir, apellido AS apedir, nombredpto
FROM empleado JOIN departamento

ON responsable = cod

La consulta se lanza desde el nodo3. El resultado de ésta consulta constará de 100 tuplas (4.000 bytes).

Opción 1: transferimos las relaciones DEPARTAMENTO y EMPLEADO al *nodo3*.

$3.500 + 1.000.000 = 1.003.500$ bytes.

Opción 2: transferimos EMPLEADO al *nodo2* y enviamos resultado del join al *nodo3*.

$1.000.000 + 4.000 = 1.004.000$ bytes.

Opción 3: transferimos DEPARTAMENTO  al *nodo1* y enviamos el resultado join al *nodo3*.

$3.500 + 4.000 = 7.500$ bytes.

Opción Optima

Bases de Datos Distribuidas

Ej de Consulta Distribuida

NUEVO SUPUESTO: la consulta se lanza desde el **nodo2**

Opción 1: transferir relación EMPLEADO al **nodo2**. De esta manera para Q1 y Q2 se transfieren el mismo numero de Byte: **1.000.000 bytes**.

Opción 2: transferir la relación DEPARTAMENTO al **nodo1**, realizar el join y enviar el resultado al **nodo2**. Para este caso se transfieren:

Para Q1: 3.500 de DEPARTAMENTO y 400.000 de resultado = **403.500 bytes**

Para Q2: 3.500 de DEPARTAMENTO y 4.000bytes de resultado = **7.500 bytes**.

Optima Opción 2

Bases de Datos Distribuidas

Proceso Distribuido utilizando semijoin

Reducción del número de columnas antes de transferir.

- Se envía **la columna** con la que se va a realizar el join **de una relación R** al nodo donde se encuentra la otra relación, **allí se realiza el join con la otra relación S**
- Se envían **las columnas implicadas en el resultado** al nodo inicial y se vuelve a realizar el join con R.
- Sólo se transfieren las columnas de R que intervienen en la realización del join en una dirección y el subconjunto de columnas de S resultantes en la otra.

Empleado

Nombre	Apellido	COD	Dir	Sexo	Sueldo	fechaNac.	Dpto.
--------	----------	-----	-----	------	--------	-----------	-------

Departamento

NombreDpto	NDpto	Responsable	Edificio	Responsable Fk Emp(cod)
------------	-------	-------------	----------	----------------------------

Bases de Datos Distribuidas

Ej de Consulta Distribuida

Ej. de proceso distribuido de consultas utilizando semijoin en Q2...

Para cada Departamento obtener el nombre del depto y su director:

- **Nodo 1: empleado**
- **Nodo 2: departamento**

Paso 1.

Proyección de Departamento sobre atributos que intervienen en el join y transferencia a nodo 1.

Proyección de responsable

F2: Π Responsable(DEPARTAMENTO)

Tamaño: $9 * 100 = 900$ bytes transferidos del **nodo2** al **nodo1**

Bases de Datos Distribuidas

Ej de Consulta Distribuida

Ej. de proceso distribuido de consultas utilizando semijoin en Q2...

Paso 2.

Realización del join de los tuplas transferidas en el paso anterior.

Transferencia del resultado del join de nuevo al nodo2.

Se transfieren sólo los atributos necesarios para realizar el join final:

F2: Π Responsable(DEPARTAMENTO)

R2: Π Responsable,Nombre,Apellido(F2  EMPLEADO)
responsable=cod

Tamaño: $9 * 100 = 900$ bytes transferidos del **nodo2** al **nodo1**

Tamaño: $(9 + 15 + 15) * 100 = 3900$ bytes transferidos del **nodo1** al **nodo2**

Total transferido: $900 + 3900 = 4.800$ bytes para Q2 (vs **7.500 bytes**)

RESUMEN

Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Fragmentación: Un analisis mas profundo.
- Procesamiento Distribuido de Consultas
- **Transacciones Distribuidas**

Bases de Datos Distribuidas

Transacciones Distribuidas

Propagación de Actualizaciones

- Método: Propagar toda actualización a todas las copias.
- Problema: Lugares inaccesibles \Rightarrow fracaso
- Método de la “copia primaria”:
 - 1 copia primaria y n copias secundarias.
 - Las copias primarias se reparten (no centralización).
 - El proceso de actualización finaliza al actualizar la primaria.
 - La copia primaria se encarga de actualizar las demás (cuando puede).

Bases de Datos Distribuidas

Transacciones Distribuidas

Protocolo de commit en dos fases:

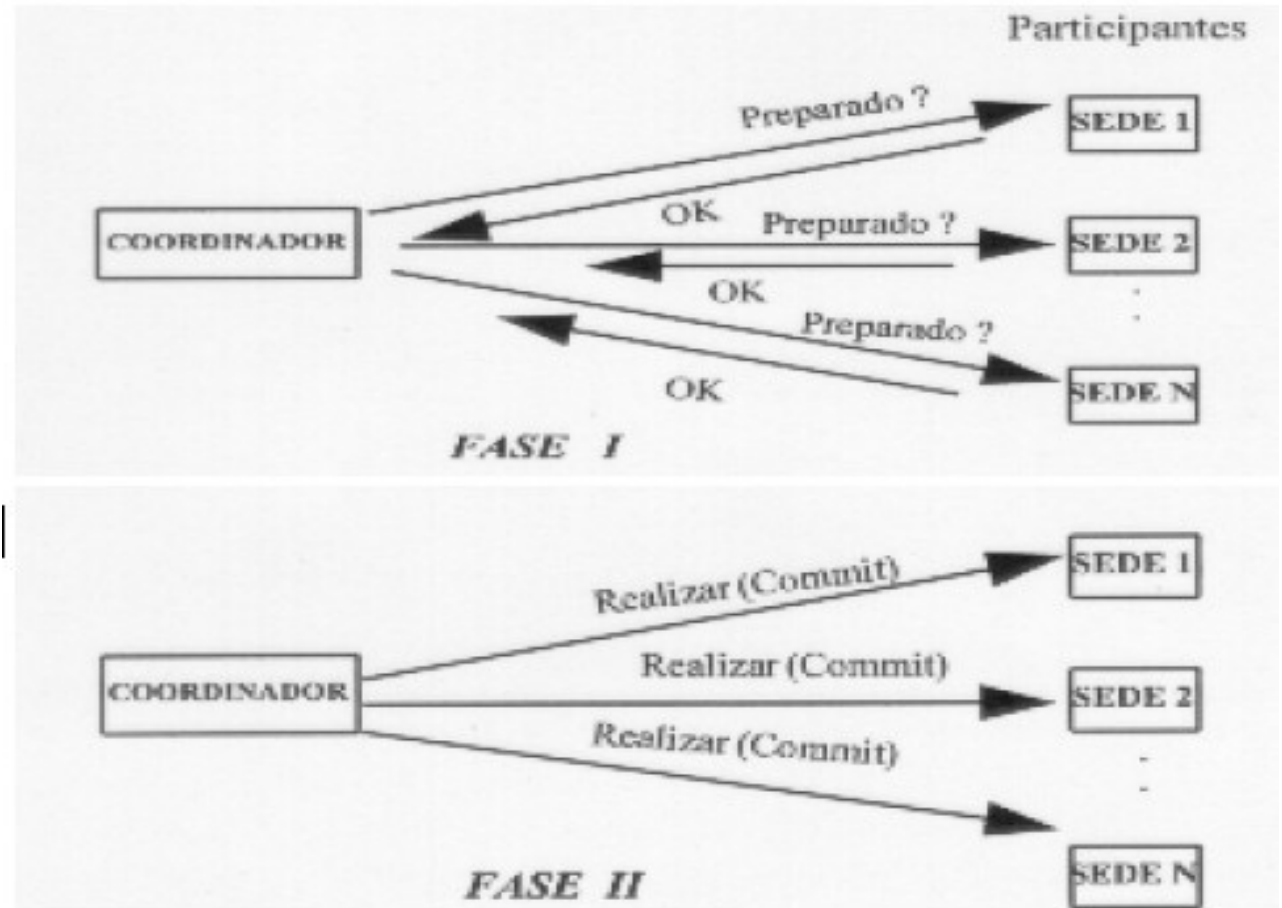
Two Phase Commit → commit-rollback

- Evitar medias operaciones: todo o nada.
- El coordinador ordena:
 - Fase 1:
 - Pedir OK o NOK a cada gestor local.
 - Cada gestor guarda una bitácora de las op's efectuadas.
 - Fase 2:
 - Si todos OK ⇒ COMMIT.
 - Si algún NOK ⇒ ROLLBACK.
 - El resultado se envía a cada gestor local para que confirmen o deshagan.
- Un fallo de interrupción se recupera de la bitácora.

Bases de Datos Distribuidas

Transacciones Distribuidas

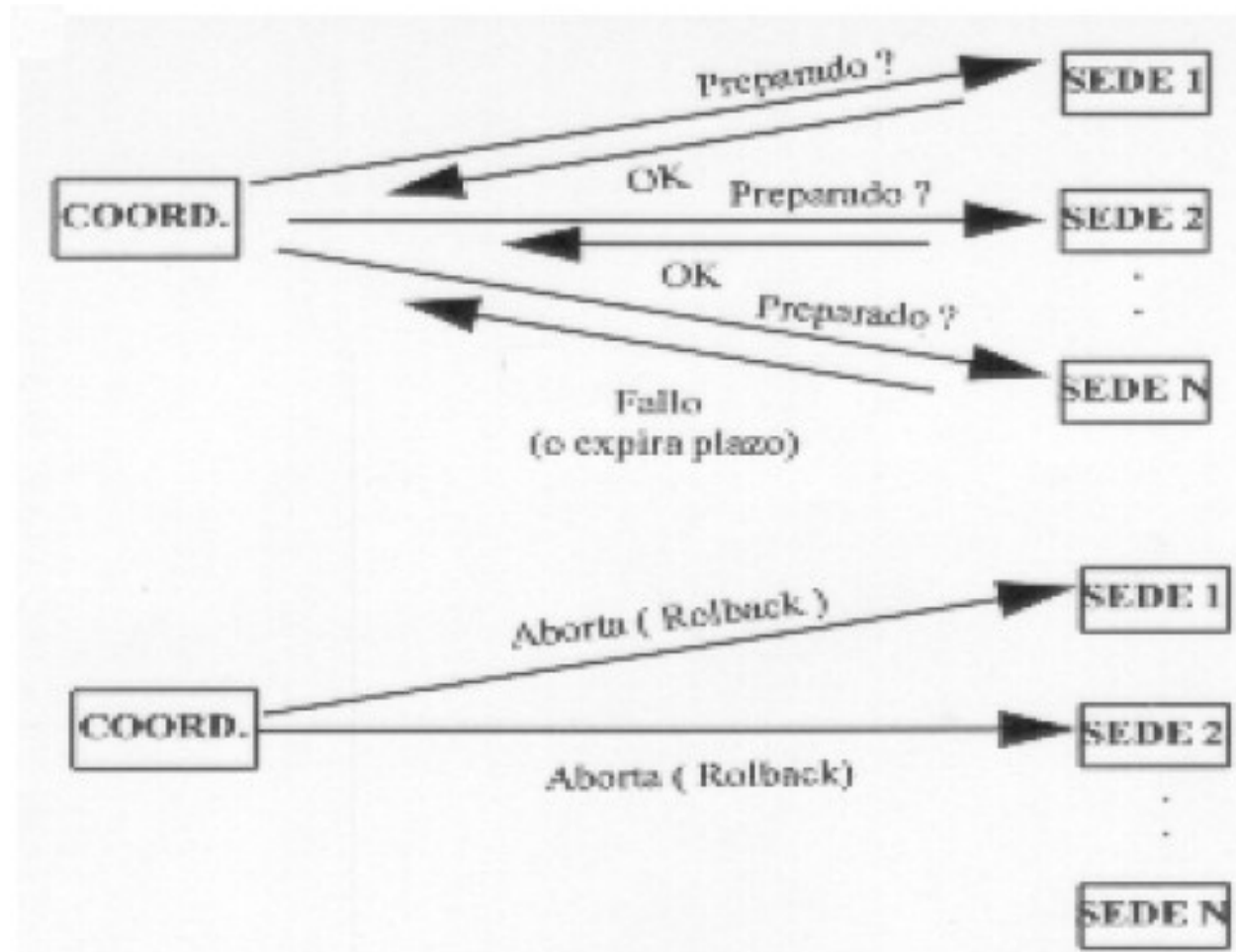
Transacción
Confirmada



Bases de Datos Distribuidas

Transacciones Distribuidas

Terminación
Con **Fallo**



Bases de Datos Distribuidas

Transacciones Distribuidas

Protocolo de commit en dos fases:

Two Phase Commit → commit-rollback

Consideraciones:

- No dependencia de un sitio central (El coordinador varia y si se cae se reelige por un sistema de votación).
- Comunicación coordinada
 - ⇒ participación ⇒ costo extra.
- Pérdida de autonomía local.

RESUMEN

Introducción a las Bases de Datos Distribuidas:

- Introducción y Conceptos
- Ventajas, Complejidad e Inconvenientes de BDD
- Almacenamiento distribuido de datos
- Fragmentación: Un analisis mas profundo.
- Procesamiento Distribuido de Consultas
- Transacciones Distribuidas
 - **Y la Standarización e Implementaciones (Informativo)**

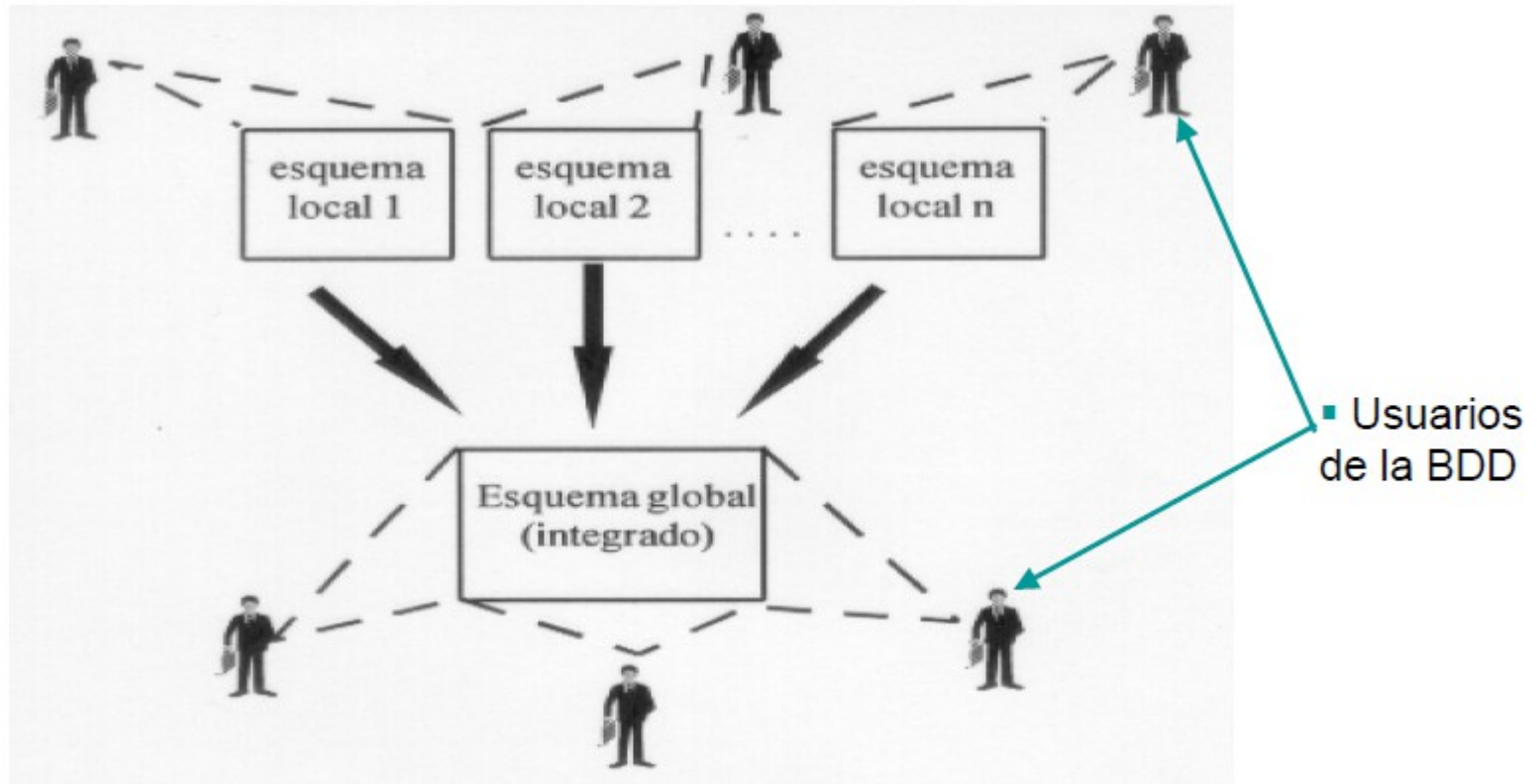
Bases de Datos Distribuidas

Diccionario Global según ANSI/SPARC

- **Esquema Global:** Administra la definición de todos los objetos que participan en la BDD y de las relaciones entre ellos. Como si se tratara de una BD no distribuida.
- **Esquema de Fragmentación:** Administra la definición de los fragmentos en los que se ha partido cada relación original.
- **Esquema de Localización:** Administra la ubicación física del sitio o conjunto de sitios donde esta cada fragmento.
- **Esquema de Mapeo Local:** Administra la relación entre los diccionarios globales y los objetos locales, entre los que se encuentran los fragmentos manejados por cada SGBD local.

Bases de Datos Distribuidas

Esquemas de una BDD



- Esquemas locales y esquema global
- Diccionario global integrado

Bases de Datos Distribuidas

Componentes de un SGBDD

Un SGBDD tiene, conceptualmente, tres **COMPONENTES** principales

Procesador de datos locales (PDL)

Gestión de datos local

Procesador de aplicaciones distribuidas (PAD)

Responsable de las funciones de distribución

Software de comunicaciones

Suministra las primitivas y los servicios de comunicaciones que necesita el PAD

Nota: Cada sede o nodo tiene un PDL y un PAD.

Bases de Datos Distribuidas

Procesadores Locales

Son todos aquellos componentes que procesan determinados servicios a nivel local, principalmente de manera autónoma.

- El procesador local de consultas
- El sistema de gestión de transacciones locales.
- El gestor de replica y recuperación de errores.
- El Persistidor (funciones de escritura) de la BD
- El administrador de memoria.
- El manejador de la seguridad local.
- Los procesos de motorización y optimización sobre BD local.

Bases de Datos Distribuidas

Procesadores Distribuidos

Son, por el contrario, los elementos que ejecutan servicios a nivel global, extendido o distribuido.

- El procesador de consultas distribuidas.
- El sistema de gestión de transacciones distribuidas.
- El subsistema de recuperación de errores.
- El administrador de fragmentos y replicas.
- El sistema de gestión de la integridad global y del diccionario.
- Los servicios de control de usuarios y seguridad global.

Bases de Datos Distribuidas

Consideraciones y Conclusiones Finales

- Las BDD se abordan desde el plano de investigación, y no están muy poco cubiertos por implementaciones de Bases de Datos concretas, al menos con productos heterogéneos de Bases de Datos.
- Algunos principios como la fragmentación se aplican en implementaciones de bases de datos NoSQL.
- La irrupción del Cloud Computing cambia las condiciones que fundamentan el principio de las bases de datos distribuidas.

Bases de Datos Distribuidas

Fuentes

Libro: Principles of Distributed Database Systems

Autor: Ozsú, T / Valduriez, P.

Editorial: Prentice Hall

Libro: Distributed Systems Principles and Paradigms

Autor: Tanenbaum A / M Van Steen

Editorial: Prentice Hall

Libro: Introducción a los Sistemas de Bases de Datos

Autor: C.J Date

Editorial: Prentice All

Libro: Fundamentos de Bases de Datos

Autor: Elmasri / Navathe

Editorial: Pearson