

Dpto. Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur

## ELEMENTOS DE BASES DE DATOS

Segundo Cuatrimestre 2014

### Clase 20: Arquitecturas de Sistemas de Bases de Datos

Mg. María Mercedes Vitturini  
[mvitturi@uns.edu.ar]



## DBMS - Arquitecturas

La “arquitectura” del DBMS está influenciada por el *framework* que le da soporte, incluyendo:

- **Conexión en red:** permite distribuir las tareas entre las que se resuelven en el servidor y las que se ejecutan en el cliente (cliente–servidor).
- **Capacidad de procesamiento en paralelo:** capacidad del servidor de contar con elementos para trabajar en paralelo. Se aceleran las actividades: tiempos de respuesta y número de transacciones por unidad de tiempo.
- **Capacidades de distribución de datos:** que permite que
  - los datos residan donde se generan/ usan, permaneciendo accesibles a todos los sitios.
  - mantener múltiples copias de datos, que refuerza la disponibilidad


EBD2014\_20 - Mg. Mercedes Vitturini

## Arquitecturas de DBMS

### Estilos arquitectónicos

- **Las arquitecturas centralizadas**
  - Sistemas centralizados: monousuario y multiusuario
  - Sistemas cliente-servidor
    - Servidor de transacciones
    - Servidor de datos
- **Arquitecturas paralelas**
- **Arquitecturas distribuidas**

- Arquitecturas para Sistemas de Información



EBD2014\_20 - Mg. Mercedes Vitturini

## Arquitecturas Centralizadas

### Sistemas Centralizados Sistemas Cliente-Servidor



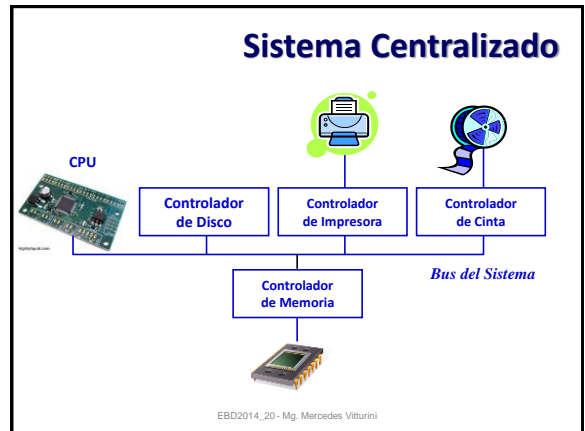
EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Centralizado

Considera **una o unas pocas CPU, memoria principal** y una serie de **controladores** conectados a través de un bus común que provee acceso a memoria compartida.

- Corre sobre **una única computadora** y no necesita interacción con otras.
- La CPU y los controladores ejecutan acciones de manera concurrente, compitiendo por el acceso a memoria.
- El uso de memorias caché reduce el número de accesos de la CPU a la memoria principal.
- **Se distinguen** los sistemas:
  - *mono-usuarios*: computadoras personales y workstation,
  - *multi-usuario*.

EBD2014\_20 - Mg. Mercedes Vitturini



## Primeras Organizaciones Centralizadas

### Mono Usuario

- Usados y administrados por una persona al mismo tiempo.
- Sistema operativo mono-usuario.
- No requieren control de concurrencia.
  - Ejemplos: **Computadoras Personales, Estaciones de Trabajo (Workstations).**

### Multi Usuarios (mainframes)

- Atienden a varios usuarios que operan el sistema al mismo tiempo.
- Disponen de mayor capacidad de disco, memoria y CPU.
- El sistema operativo es multiusuario.
- Disponen de facilidades para multitareas.
- Los usuarios se conectan desde terminales.

EBD2014\_20 - Mg. Mercedes Vitturini

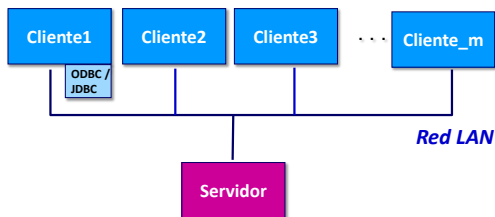
## Entorno Cliente-Servidor

- Es un tipo de organización **ampliamente usada**.
- Las **computadoras personales (PC)** reemplazan a la terminal y **asumen la responsabilidad de la funcionalidad asociada a la interface**.
- El sistema actúa como un **sistema servidor** que **satisface los requerimientos de los clientes**.



EBD2014\_20 - Mg. Mercedes Vitturini

## Entorno Clientes-Servidor



El **sistema servidor** satisface los requerimientos para **m sistemas clientes**

EBD2014\_20 - Mg. Mercedes Vitturini

## DBMS y Cliente-Servidor

- Los **servicios del DBMS** se dividen en:
  - **Back-End (en el servidor)**: manejar las estructuras de acceso, evaluación y optimización de consultas, control de concurrencia y recuperación ante fallos.
  - **Front-End (en el cliente)** abarcan herramientas como generadores de reportes y facilidades para el diseño de interfaces gráficas, herramientas de análisis y data mining.
- La **interface entre back-end y front-end** es a través de SQL o de los programas de aplicación.



EBD2014\_20 - Mg. Mercedes Vitturini

## Funcionalidades Front-End y Back-End

### Front-End



Interface (SQL + API)

Back-End



EBD2014\_20 - Mg. Mercedes Vitturini

## Arquitectura del Sistema Servidor

- **SISTEMA SERVIDOR DE TRANSACCIONES (o servidores de consulta)**: proveen una interface para que el cliente les envíe requerimientos y retornar los resultados.
- **SISTEMA SERVIDOR DE DATOS**: los clientes interactúan con el servidor a través de requerimiento de lectura o escritura de datos, en unidad de archivo o páginas.

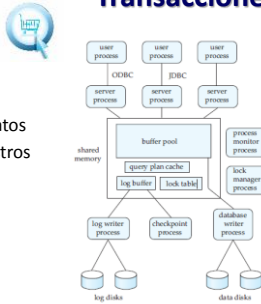


EBD2014\_20 - Mg. Mercedes Vitturini

## Una Arquitectura de Servidor de Transacciones

### Procesos del servidor:

- Procesos de los clientes.
- Proceso lock manager.
- Proceso actualizador de datos
- Proceso actualizador registros log.
- Proceso de checkpointing.
- Proceso de monitoreo y recuperación



EBD2014\_20 - Mg. Mercedes Vitturini

## Servidor de Datos

### Arquitectura del servidor de datos:

Se requiere:

- Redes locales con alta conectividad entre cliente y servidor.
- Los clientes tienen alta capacidad de procesamiento.
- Instalan servicios full back-end en el cliente.
- Las arquitecturas data-server son usadas en sistemas de bases de datos orientados a objetos.



EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas de Información

### Estilos Arquitectónicos para SI asistidos por SMBD



## Sistemas de Información

**Sistemas Información (SI):** dado un negocio, la base de datos es apenas una parte de la estructura total, que considera:

- **Datos:** información persistente.
- **Procesos:** que manipulan los datos.
- **Personas:** que llevan adelante los procesos
- **Redes:** que transportan los datos dentro de una organización y fuera de ella.
- **Hardware:** CPU, discos, impresoras, etc.
- **Software de base:** sistema operativos, DBMS, programas de aplicación, etc.

EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas de Información

La **organización lógica** típica en “capas” o niveles de un SI identifica:

- **Capa de presentación (PL):** se encarga de aspectos de presentación e interacción con el usuario.
- **Capa de Lógica de Negocio (BL):** se ocupa de representar a los conceptos y funcionalidades del negocio.
- **Capa de Acceso a Datos (DAL):** administra las conexiones con la capa de datos.
- **Capa de Datos (DL):** concentra la persistencia de datos en la base de datos

EBD2014\_20 - Mg. Mercedes Vitturini

## Organización Lógica en Capas

Componente Lógico	Responsabilidades
Capa de Presentación (PL) – Procesos de interfaz de usuario	Interactuar con distintos tipos de usuario. Interactuar con otras aplicaciones. Validación básica de las entradas al sistema. Presentar los resultados.
Capa de Negocio (BL) – Procesos de Negocio	Realizar cálculos y procedimientos de negocio. Hacer ejecutar las reglas de negocio. Representar entidades del negocio.
Capa de Acceso a Datos (DAL) – Procesos de acceso a datos	Proveer acceso a datos. Administrar las conexiones. Proveer seguridad en el acceso.
Capa de Datos (DL) – Procesos de datos	Almacenamiento y manipulación de datos (persistencia). Hacer cumplir las restricciones sobre los datos.

EBD2014\_20 - Mg. Mercedes Vitturini

## Tiers y Layers

**Layer (capa):** separación lógica del sistema.

**Tier:** separación física del sistema (servidores).

### Two Tiers

#### Presentación

- Interfaz de usuario
- Llamadas al dominio
- Mostrar resultados

#### Dominio

### Three Tiers

#### Presentación

- Interfaz de usuario
- Mostrar resultados

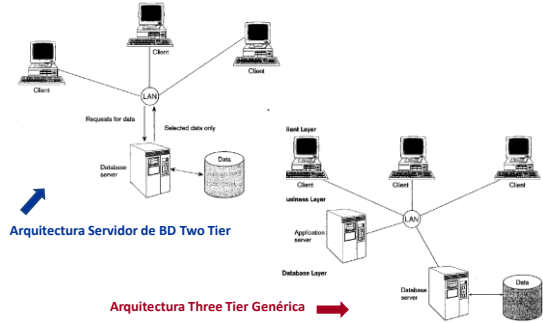
#### Lógica de Aplicación

- Llamadas al dominio
- Reglas del negocio

#### Dominio

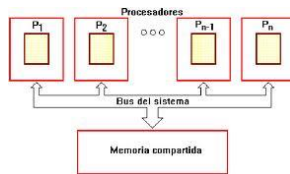
EBD2014\_20 - Mg.

## Arquitecturas Cliente-Servidor



EBD2014\_20 - Mg. Mercedes Vitturini

## Arquitecturas Paralelas



## Arquitecturas Paralelas

Por un tiempo se dejaron de lado, pero actualmente han vuelto a ser comercializables. Algunas razones:

- Han crecido los sistemas transaccionales y desde la Web surgen **sitios con millones de usuarios que hacen que las compañías tengan bases de datos cada vez más grandes.**
- Creció el interés y la demanda de **sistemas de soporte de decisión** que consultan y generan información de **bases de datos muy grandes (terabytes)**, que no tendrían tiempos de respuesta aceptables si se considera un único procesador.
- Micro procesadores son más accesibles y surgieron nuevas **tecnologías multicore.**
- La **naturaleza orientada a conjuntos de las consultas** hace natural pensar en procesamiento paralelo.

EBD2014\_20 - Mg. Mercedes Vitturini

## DBMS y Arquitecturas Paralelas

- Un servidor con arquitectura paralela considera:
  - **múltiples procesadores** y/o
  - **múltiples discos** y/o
  - **múltiples memorias.**
- El objetivo es **mejorar tiempos de procesamiento y/o la velocidad de E/S.**
- Para ellos, las **operaciones se desarrollan en paralelo** (simultáneamente).
- Surgen junto con la demanda de aplicaciones que usan bases de datos muy grandes (del orden de terabytes) y de miles de transacciones por minuto.

EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Paralelos

- Dos organizaciones:
  - **Coarse-Grain (granularidad gruesa):** máquinas con un pequeño número de procesadores muy potentes.
  - **Massively parallel o Fine-Grain (granularidad fina):** máquinas con cientos de procesadores pequeños.
- Se consideran dos parámetros de performance para un SMDB:
  - **Productividad (Throughput):** el número de tareas que pueden terminarse en un período de tiempo.
  - **Tiempo de Retorno:** el tiempo que toma para completar una única tarea.

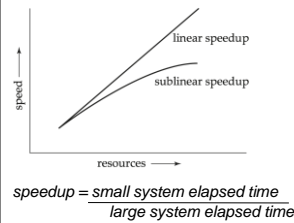
EBD2014\_20 - Mg. Mercedes Vitturini

## Performance de Sistemas Paralelos

- Los sistemas paralelos aumentan el rendimiento *procesando varias transacciones en paralelo.*
- El tiempo de retorno también se puede mejorar ya que pueden desarrollar *subtareas de una transacción en paralelo.*
- Cuestiones de estudio en sistemas paralelos:
  - **Speedup:** analiza el tiempo de ejecutar la misma tarea en un equipo que incrementa el paralelismo.
  - **Scaleup:** se refiere al respuesta de a tareas más complejas incrementando el grado de paralelismo.

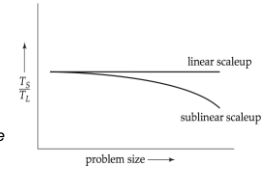
EBD2014\_20 - Mg. Mercedes Vitturini

## Speedup



$$\text{speedup} = \frac{\text{small system elapsed time}}{\text{large system elapsed time}}$$

## Scaleup



$$\text{scaleup} = \frac{\text{small system small problem elapsed time}}{\text{big system big problem elapsed time}}$$

EBD2014\_20 - Mg. Mercedes Vitturini

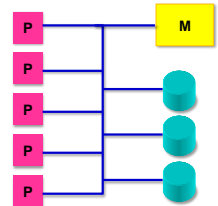
## Arquitecturas Paralelas

- Existen distintos modelos de arquitectura para máquinas paralelas:
  - **Memoria compartida:** todos los procesadores comparten una memoria común.
  - **Disco compartido:** todos los procesadores comparten un conjunto de discos en común (clusters).
  - **Nada compartido.**
  - **Jerárquico:** una combinación de los anteriores.

EBD2014\_20 - Mg. Mercedes Vitturini

## Arq. Paralelas – Memoria Compartida

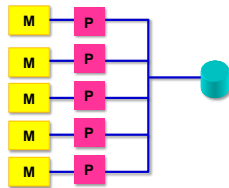
- Procesadores y discos comparten la memoria en común vía bus de comunicación o interconexión de red.
  - '+' Es eficiente la comunicación entre procesadores, los datos compartidos pueden ser accedidos por cualquier procesador.
  - '-' No admite más de 32 a 64 procesadores ya que el bus de interconexión resulta un cuello de botella.



EBD2014\_20 - Mg. Mercedes Vitturini

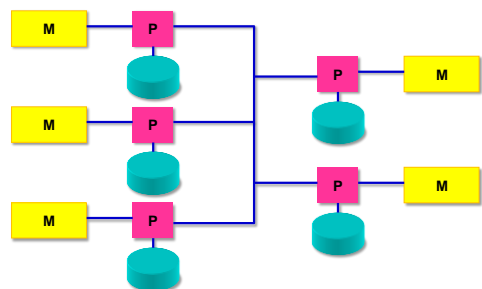
## Arq. Paralelas – Disco Compartido

- Todos los procesadores comparten el medio de almacenamiento.
- Cada procesador cuenta con su memoria principal.
  - '+' el bus no es cuello de botella
  - '+' proporcionan tolerancia a fallas de procesador.
  - '-' el cuello de botella ocurre a nivel de conexión con el disco. Esto limita el crecimiento en cuanto a escalabilidad (scaleup).



EBD2014\_20 - Mg. Mercedes Vitturini

## Arq. Paralelas – Nada compartido



EBD2014\_20 - Mg. Mercedes Vitturini

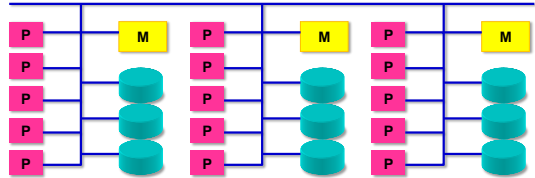
## Arq. Paralelas – Nada compartido

- Cada nodo consiste de procesador, memoria y uno o más discos.
- La comunicación entre procesadores es a través de la red.
  - ‘+’ Son ampliables y pueden soportar gran número de procesadores.
  - ‘+’ minimiza las interfaces por recursos compartidos.
  - ‘-’ La desventaja esta en el costo de comunicación y acceso a disco no locales.

EBD2014\_20 - Mg. Mercedes Vitturini

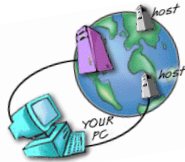
## Arq. Paralelas – Jerárquica

- Es un sistema híbrido entre memoria compartida, disco compartido y nada compartido.
- Las arquitecturas de memoria virtual distribuida han surgido de la idea de arquitecturas jerárquicas.



EBD2014\_20 - Mg. Mercedes Vitturini

## DBMS Distribuido



**Organización para datos almacenados en varias computadoras (sitios o nodos) comunicadas entre si**

## DBMS Distribuidos

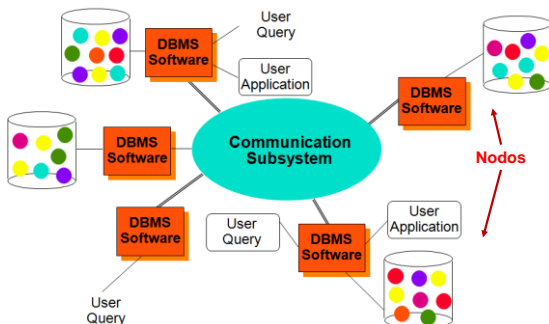
En un *sistema de manejo de base de datos distribuido (SMBDD o DDBMS)* la base de datos se almacena en diversas computadoras, denominados sitios o nodos, comunicados entre si.

### Características

- Los nodos están en equipos que **no comparten** memoria ni disco y se conectan por la red.
- Los nodos generalmente se ubican geográficamente separados.
- Las computadoras pueden variar en tamaño y función.
- Las transacciones pueden acceder a datos que está físicamente en sitios distintos.

EBD2014\_20 - Mg. Mercedes Vitturini

## DBMS Distribuidos



EBD2014\_20 - Mg. Mercedes Vitturini

## Bases de Datos Distribuidas

- **Bases de datos distribuidas homogéneas:**
  - El mismo software/esquema en todos los sitios, los datos son particionados/replicados entre los sitios.
  - **Objetivo:** proveer la vista de una sola base de datos ocultando detalles de distribución.
- **Bases de datos distribuidas heterogéneas:**
  - Diferentes software/esquema en los sitios.
  - **Objetivo:** integrar distintas bases de datos para proveer una función útil.

EBD2014\_20 - Mg. Mercedes Vitturini

## Bases de Datos Distribuidas Homogéneas

### BD Distribuidas Homogéneas – características:

- Todos los sitios tienen software de base de datos idéntico.
- Cada sitio sabe de la existencia de los otros y están de acuerdo en cooperar con la atención de los requerimientos de usuarios.
- Cada sitio sacrifica autonomía en beneficio de conservar un esquema idéntico.
- Para los usuarios es como si existiera un único sistema.

EBD2014\_20 - Mg. Mercedes Vitturini

## Bases de Datos Distribuidas Heterogéneas

### BD Distribuidas Heterogéneas – características:

- Sitios distintos pueden usar esquemas y software de bases de datos distintos:
  - **Diferencias en los esquemas** afectan al procesamiento de consultas.
  - **Diferencias en el software** afectan principalmente al problema de procesamiento.
- Los sitios pueden no saber de la existencia de los otros y solo proveer facilidades acotadas de cooperación en procesamiento de transacciones.

EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Distribuidos Homogéneos

### Características

- Cada sitio conoce la existencia de los otros.
- Cada sitio provee un entorno para ejecutar tanto transacciones locales como globales.

### TRANSACCIONES LOCALES Y TRANSACCIONES GLOBALES

- Una **transacción local** accede a datos de un único sitio donde se inició la transacción.
- Una **transacción global** accede tanto a datos del sitio donde se originó como a datos de varios sitios diferentes.

EBD2014\_20 - Mg. Mercedes Vitturini

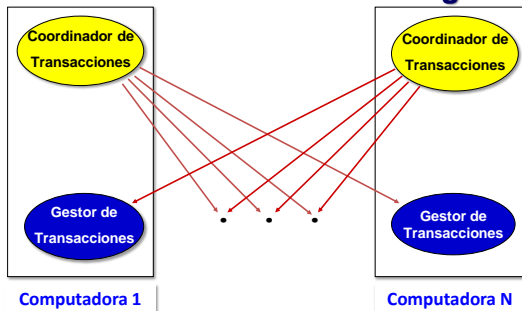
## Modelo de Transacciones Distribuido

### Cada sitio cuenta con dos subsistemas:

- **El gestor de transacciones:** gestiona la ejecución de aquellas transacciones (o subtransacciones) que acceden a los datos almacenados en el sitio local.
  - Mantener la bitácora.
  - Participar en el esquema de control de concurrencia del sitio local.
- **El coordinador de transacciones:** encargado de la ejecución de las transacciones (locales y globales) iniciadas en el sitio local.
  - Iniciar las transacciones.
  - Dividir las transacciones en subtransacciones.
  - Coordinar la ejecución de subtransacciones en los distintos sitios.

EBD2014\_20 - Mg. Mercedes Vitturini

## Modelo Transaccional Distribuido Homogéneo



EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Distribuidos - Ventajas

- **Datos compartidos** – los usuarios de un sitio pueden acceder a datos de otros sitios.
- **Autonomía** – cada sitio es capaz de mantener el control de los datos que están almacenados localmente. Existe un administrador global del sistema y cada sitio cuenta con un administrador local.
- **Mayor Disponibilidad** – si un sitio falla, el resto de los sitios puede continuar operando. Si la información se replica, el fallo de un sitio puede no afectar a todo el sistema.

EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Distribuidos - Desventajas

- **Costo del Desarrollo de Software** – es más difícil y costoso implementar un sistema de base de datos distribuido.
- **Mayor Probabilidad de Errores** – puesto que los sitios que constituyen el sistema distribuido operan en paralelo, es más difícil asegurar la correctitud de los algoritmos.
- **Incremento en la Sobrecarga de Procesamiento** – el intercambio de mensajes y los cómputos adicionales para coordinar los sitios constituyen una sobrecarga importante.

EBD2014\_20 - Mg. Mercedes Vitturini

## Sistemas Distribuidos vs. Paralelos

- Los **sistemas distribuidos** se asemejan a los **sistemas paralelos con estructura nada compartido**.
- Sin embargo, generalmente están en **lugares diferentes** (geográficamente separados), son **administradas independientemente** y tienen una **interconexión más lenta**.
- Otra diferencia es el tipo de transacciones que admiten. Las **transacciones locales** acceden a datos de un único sitio mientras que las **transacciones globales** pueden acceder a datos que están en sitios diferentes a donde se inicia la transacción.

EBD2014\_20 - Mg. Mercedes Vitturini

## ¿Qué vamos a estudiar?

- Vamos a analizar los siguientes tópicos en relación a **Sistemas de Manejo de Bases Datos Distribuidos Homogéneos**:
  - Cómo almacenan la información y acceden a la información.
  - Cuestiones de transparencia de red y denominación de los ítems de datos.
  - Procesamiento y concurrencia de transacciones distribuidas.
  - Tratamiento de fallos y deadlocks.

EBD2014\_20 - Mg. Mercedes Vitturini

## Almacenamiento de los Datos Distribuido

- Replicación** – el sistema mantiene varias copias (réplicas) idénticas de una relación  $r$ . Cada réplica se almacena en un sitio diferente. Mejoras
  - Tiempo de respuesta y tolerancia a fallos.
- Fragmentación** – una relación  $r$  se particiona en varios fragmentos almacenados en sitios diferentes.
- Replicación y Fragmentación** – una relación  $r$  es particionada en varios fragmentos y el sistema mantiene varias copias idénticas de los fragmentos.

EBD2014\_20 - Mg. Mercedes Vitturini

## Replicación de Datos

- Se dice que **una relación o fragmento de una relación está replicado** si se almacena replicado en dos o más nodos un sistema de bases de datos distribuida:
  - **Replicación total de una relación** en el caso en que la relación se copia en todos los sitios.
  - **Replicación total de la base de datos** cuando todos los sitios tienen una copia completa de toda la base.
  - En otros casos se habla de replicación parcial.

EBD2014\_20 - Mg. Mercedes Vitturini

## Replicación de Datos – Ventajas

- + **Disponibilidad**: si uno de los sitios que contiene a la relación  $r$  falla, entonces se la puede buscar en otro sitio.
- + **Incremento del paralelismo**: en casos de mayoría de accesos de lectura, los sitios procesando consultas sobre  $r$  se ejecutan en simultáneo.
- + **Reduce el tiempo de transferencia**: la relación  $r$  está localmente disponible en cada sitio que contiene una copia de  $r$ .

EBD2014\_20 - Mg. Mercedes Vitturini



## Replicación de Datos – Desventajas

- **Incremento del overhead en actualizaciones:** el sistema debe asegurar que **TODAS** las réplicas de la relación  $r$  sean consistentes. Cada actualización sobre la relación  $r$  debe ser propagada a los sitios conteniendo las respectivas réplicas.
- **Incremento en la complejidad del control de concurrencia:** la actualización concurrente de copias distintas puede llevar a datos inconsistentes. Se deben implementar un mecanismo de control de concurrencia distribuidos.

EBD2014\_20 - Mg. Mercedes Vitturini

## Fragmentación de los datos

**Fragmentación** – Divide una relación  $r$  en fragmentos  $fr_1, fr_2, \dots, fr_n$ . Los fragmentos contienen información suficiente como para reconstruir la relación original  $r$ .

Tipos de fragmentación:

- **Fragmentación Horizontal** (por filas): separa a una relación  $r$  dividiendo las tuplas en subconjuntos.
- **Fragmentación Vertical** (por columnas): separa a una relación  $r$  a través de descomposiciones en subesquemas.
- **Combinada:** separa a una relación  $r$  horizontal y verticalmente.

EBD2014\_20 - Mg. Mercedes Vitturini

## Fragmentación Horizontal

Apellido	Nombres	Registro	Domicilio	Código-Carrera
Andrade	Luis	44455	Casanova 654	128
García	Federico	35689	Alem 1233	50
García	Federico	25689	Alem 1233	128
López	Ernesto	56661	San Juan 20	50
Luceletti	Marcos	52256	Córdoba 456	128
Ramírez	José	52231	Salta 102	50
Reyes	Francisco	42467	12 de Octubre 980	128
Sánchez	Andrés	48995	Paraguay 20	50

$r = r1 \cup r2$

Apellido	Nombres	Registro	Domicilio	Código-Carrera
Andrade	Luis	44455	Casanova 654	128
García	Federico	35689	Alem 1233	50
García	Federico	25689	Alem 1233	128
López	Ernesto	56661	San Juan 20	50

$r1$

Apellido	Nombres	Registro	Domicilio	Código-Carrera
Luceletti	Marcos	52256	Córdoba 456	128
Ramírez	José	52231	Salta 102	50
Reyes	Francisco	42467	12 de Octubre 980	128
Sánchez	Andrés	48995	Paraguay 20	50

$r2$

EBD2014\_20 - Mg. Mercedes Vitturini

## Fragmentación Vertical

Id-Tupla	Apellido	Nombres	Registro	Domicilio	Código-Carrera
1	Andrade	Luis	44455	Casanova 654	128
2	García	Federico	35689	Alem 1233	50
3	García	Federico	25689	Alem 1233	128
4	López	Ernesto	56661	San Juan 20	50
5	Luceletti	Marcos	52256	Córdoba 456	128
6	Ramírez	José	52231	Salta 102	50
7	Reyes	Francisco	42467	12 de Octubre 980	128
8	Sánchez	Andrés	48995	Paraguay 20	50

$r = r1 \bowtie r2$

Id-Tupla	Apellido	Nombres	Registro
1	Andrade	Luis	44455
2	García	Federico	35689
3	García	Federico	25689
4	López	Ernesto	56661
5	Luceletti	Marcos	52256
6	Ramírez	José	52231
7	Reyes	Francisco	42467
8	Sánchez	Andrés	48995

$r1$

Id-Tupla	Domicilio	Código-Carrera
1	Casanova 654	128
2	Alem 1233	50
3	Alem 1233	128
4	San Juan 20	50
5	Córdoba 456	128
6	Salta 102	50
7	12 de Octubre 980	128
8	Paraguay 20	50

$r2$

EBD2014\_20 - Mg. Mercedes Vitturini

## Fragmentación – Ventajas

### Fragmentación horizontal:

- Permite **procesamiento paralelo** sobre fragmentos de una relación.
- Permite que una relación se distribuya de forma que **cada tupla se ubique donde más frecuentemente se utiliza**.

### Fragmentación vertical:

- Permite que una relación se distribuya de forma que **cada parte de la tupla se ubique donde más frecuentemente se utiliza**.
- Permite **procesamiento paralelo** sobre una relación.

EBD2014\_20 - Mg. Mercedes Vitturini

## Transparencia

**Transparencia** – se refiere a cuánto un usuario del sistema puede abstraerse de detalles sobre **cómo** y **dónde** se almacena un ítem de datos en un sistema distribuido.

- Transparencia en relación a:
  - Transparencia de fragmentación.
  - Transparencia de replicación.
  - Transparencia de ubicación.

EBD2014\_20 - Mg. Mercedes Vitturini

## Nombrado de Ítems de datos

### Principios

1. **Todo ítem de dato debe tener un nombre** único en el sistema distribuido.
2. Debería ser posible **encontrar la ubicación de un ítem de dato en forma eficiente**.
3. Debería ser posible **cambiar la ubicación de un ítem de dato en forma transparente**.
4. **Cada sitio debería ser capaz de crear nuevos ítems de dato en forma autónoma**.

EBD2014\_20 - Mg. Mercedes Vitturini

## Servidor de Nombres

### Servidor de nombres – Esquema centralizado

- Genera todos los nombres.
- Los sitios consultan al servidor de nombres por la ubicación de los datos no locales.
- **Ventaja:** Cumple con los principios 1-3.
- **Desventajas:**
  - El servidor de nombres es el cuello de botella (y afecta a la *performance*).
  - Si se cae el servidor de nombres, ningún otro sitio puede continuar.
  - No cumple con el principio 4.

EBD2014\_20 - Mg. Mercedes Vitturini

## Esquema de nombrado de datos

Alternativa al esquema centralizado:

- Cada sitio agrega como **prefijo** a los nombres que crea su nombre de su sitio. **Ejemplo:** [sitio11.cuentas](#)
- **Desventaja:**
  - Falla al principio de transparencia.
- **Solución:** crear nombres alternativos o *alias* y almacenar el mapeo entre alias y nombre real en todos los sitios.
  - Los usuarios denominan a los datos por su *alias* y el sistema los traduce a los nombres completos.
  - El usuario no necesita conocer la ubicación física de los datos.

EBD2014\_20 - Mg. Mercedes Vitturini

## Nombrado de fragmentos y réplicas

- Similarmente se pueden usar técnicas similares para nombrar fragmentos y réplicas y sus alias.
  - **Ejemplo:** [sitio1.cuentas.f3.r2](#) para referirse a la segunda réplica del tercer fragmento de la relación cuentas en el sitio 1.
- (-) No es deseable que el usuario pueda referirse a un/una fragmento/réplica particular.
- El **sistema** es el que debe determinar qué copia referenciar en caso de un *read* y actualizar todas las réplicas en caso de un *write*.

EBD2014\_20 - Mg. Mercedes Vitturini

## Transparencia y actualizaciones

- Proveer transparencia con actualización es complejo. Hay que asegurar que **todas las réplicas y fragmentos** de un ítem de dato se actualicen.
  - Si se actualiza un ítem de datos fragmentado, debe determinarse que fragmento/s debe/n modificarse, dependiendo si se utiliza fragmentación horizontal, vertical o combinada.
  - Si se actualiza un ítem de datos replicado, debe actualizarse cada una de las réplicas.

EBD2014\_20 - Mg. Mercedes Vitturini

## Datos Lógicos y Datos Físicos

- En SMBDD existen dos niveles de datos:
  - DATOS LÓGICOS Y DATOS FÍSICOS**
- Si para un dato **A** existe una única copia, *el dato lógico y el dato físico son iguales*.
- Si un dato **A** (dato lógico) tiene **n copias**  $A_1, \dots, A_n$  (datos físicos), entonces las transacciones solicitando un ítem **A** deben enviar mensajes de requerimientos de bloqueo a los sitios donde reside la copia.
- El **gestor de bloqueos** de cada sitio puede **conceder o negar** el bloqueo, enviando el mensaje respectivo.
- El criterio de compatibilidad entre bloqueos es igual que en sistemas centralizados.

EBD2014\_20 - Mg. Mercedes Vitturini

## Temas de la clase de hoy

- **DBMS: Arquitecturas**
  - Sistemas centralizados. Cliente Servidor.
  - Sistemas Paralelos
  - Sistemas distribuidos.
  - Entorno de red
- **Sistemas Distribuidos**
  - Replicación y Fragmentación de Datos.
  - Datos Físicos y Datos Lógicos.
- **Bibliografía**
  - “Principles of Database and Knowledge-Base Systems” – J. Ullman. Capítulo 10.
  - “Database Systems Concepts” – A. Silberschatz. Capítulos 20 y 22 (ed. 2005) / Capítulos 17 y 19 (ed. 2010).
  - Modern Database Management – J. Hoffer Cap. 9 (9th ed)

EBD2014\_20 - Mg. Mercedes Vitturini

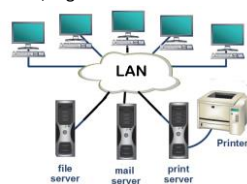
## Tipos de Redes

- Las bases de datos distribuidas y los sistemas cliente/servidor se construyen sobre la base de **redes de comunicación**
- Existen dos tipos de redes:
  - **Redes de área local (LAN)**: compuesta de procesadores distribuidos en áreas geográficas pequeñas.
  - **Redes de área amplia (WAN)**: compuestas por procesadores distribuidos en áreas geográficas mayores.
- Entre ellas varían en velocidad y confiabilidad.

EBD2014\_20 - Mg. Mercedes Vitturini

## Redes LAN

- **LANs - Local Area Networks (1970s).**
  - Cubren áreas geográficas pequeñas (por ejemplo, un edificio o varios edificios cercanos).
  - Conexiones por cable coaxial o fibra óptica.
  - Ancho de banda: 1MB/seg a 10GB/seg.
  - Conectan grupos de trabajo.
  - Son económicas.
  - Permiten compartir recursos



EBD2014\_20 - Mg. Mercedes Vitturini

## Redes WAN

- **WANs - Wide Area Networks (1960s).**
  - Cubren áreas geográficas extensas (ciudades, países, continentes).
  - Conexiones típicamente por teléfono, soportadas por líneas de fibra óptica o por satélite.
  - Las conexiones pueden ser discontinuas (UUCP de Unix) o continuas (Internet).



EBD2014\_20 - Mg. Mercedes Vitturini

## Diferentes Topologías de Redes

- **Red Totalmente Conectada**: existe un vínculo entre cada par de nodos.
- **Red Parcialmente Conectada**: algunos pares de nodos no están conectados.
- **Red Tipo árbol**: existe una estructura arbórea con un nodo raíz, sus hijos. Cada nodo puede tener 0 o más nodos hijos pero sólo un nodo padre.
- **Red Estrella**: existe un nodo al cual todos están conectados.
- **Red Anillo**: los nodos están conectados circularmente.

EBD2014\_20 - Mg. Mercedes Vitturini