

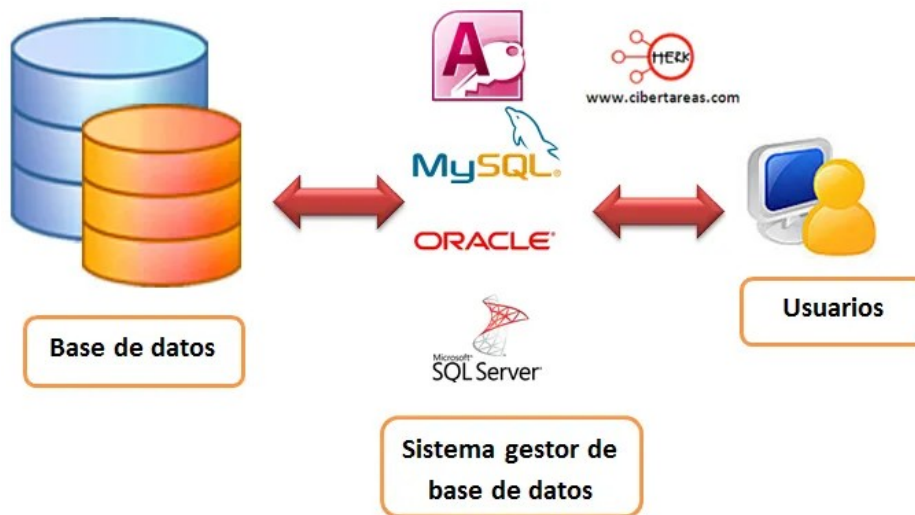
BASES DE DATOS

TEMA 5

ORACLE

1. SGBD

El SGBD (**Sistema Gestor de Bases de Datos**) actúa como una herramienta intermedia entre los usuarios, las aplicaciones y los datos almacenados en la base de datos para describir, almacenar y manipularlos.



Un SGBD es una herramienta de propósito general que permite crear diferentes BBDD de cualquier tamaño y complejidad

2. Objetivos

- **Independencia de datos:** los programas de aplicación deben verse afectados lo menos posible por cambios efectuados en datos que no usan.
- **Integridad de los datos:** la información almacenada en la BD debe cumplir ciertos requisitos de calidad; para ello hace falta, en el momento de introducirse los valores sean válidos y que posteriormente no se deterioren.
- **Seguridad de los datos:** a la información almacenada en la BD sólo pueden acceder las personas autorizadas y de la forma autorizada.



3. Funciones

- **Definición y control centralizado de los datos.** Definición de todos los elementos de datos en la base de datos.
- **Descripción de los datos** (campos, grupos, registros, tablas), **interrelaciones entre las diferentes estructuras de datos.** A esta información se le conoce como metadatos y se puede consultar en el llamado **Catálogo o Diccionario de datos.**
- **Manipulación de los datos.** El SGBD debe ser capaz de atender las solicitudes del usuario para extraer, modificar o añadir datos a la base de datos.
- **Seguridad:** Proporciona los medios para definir y gestionar las Autorizaciones de acceso, ya sea mediante claves de acceso al sistema, o mediante la definición de vistas externas de usuario.



4. Funciones

- **Integridad.** Proporciona así mismo los medios para garantizar la integridad y la consistencia de los datos definiendo restricciones sobre los valores que pueden tomar.
- Proporcionando capacidades de **recuperación ante fallos y de copia de seguridad.**
- Garantiza la **disponibilidad de la información** asegurando el acceso concurrente a varios usuarios simultáneos.
- **Gestiona la base de Datos**, monitoriza su funcionamiento, genera estadísticas, control de espacios utilizados, fragmentación



5. Arquitectura de un SGBD

El grupo **ANSI/SPARC** realizó una propuesta de arquitectura de un SGBD a tres niveles:

- Externo
- Conceptual
- Interno

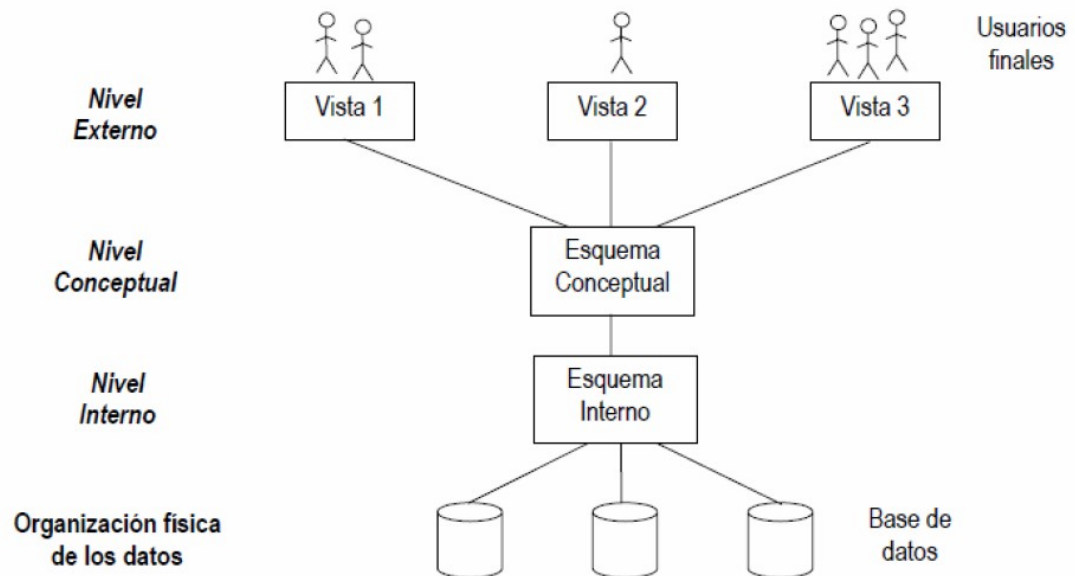


Figura 1.21. Arquitectura de un SGBD

6. Arquitectura de un SGBD

Nivel físico

Es una descripción de la representación en memoria externa (disco) del ordenador de los datos definidos en el esquema lógico de la base de datos, es decir, cómo se almacenan los datos contenidos en el conjunto de tablas que forman la BD.

A este nivel se define:

- La **descripción de los archivos** (nombre, método de acceso, organización)
- Definición del **espacio asignado** para almacenar el conjunto de datos.
- **Optimización del tiempo de respuesta** (definición de claves, índices, punteros) y espacio de memoria utilizado como buffer de datos.
- **Criptografía y compresión de datos.**



7. Arquitectura de un SGBD

Nivel conceptual/ lógico

Visión del Sistema de Información desde la BD expresada en términos del modelo de datos en que se base el SGBD que se vaya a utilizar.

A este nivel se define:

- Las tablas que formarán la base de datos.
- Las reglas de validación de los datos.
- Relaciones entre las tablas utilizando los campos correspondientes que definen estas relaciones.
- Control de la integridad de los datos.
- Restricciones de los campos.



8. Arquitectura de un SGBD

Nivel externo

Visión del sistema desde el punto de vista del usuario.

Se define la visión que cada usuario tiene del sistema, de forma que a cada usuario se le muestre únicamente la parte de la base de datos a la que tiene acceso.

Esta parte de la BD se corresponde con un subconjunto del esquema lógico.

Gracias al esquema externo a los usuarios se les oculta la complejidad interna de la base de datos..



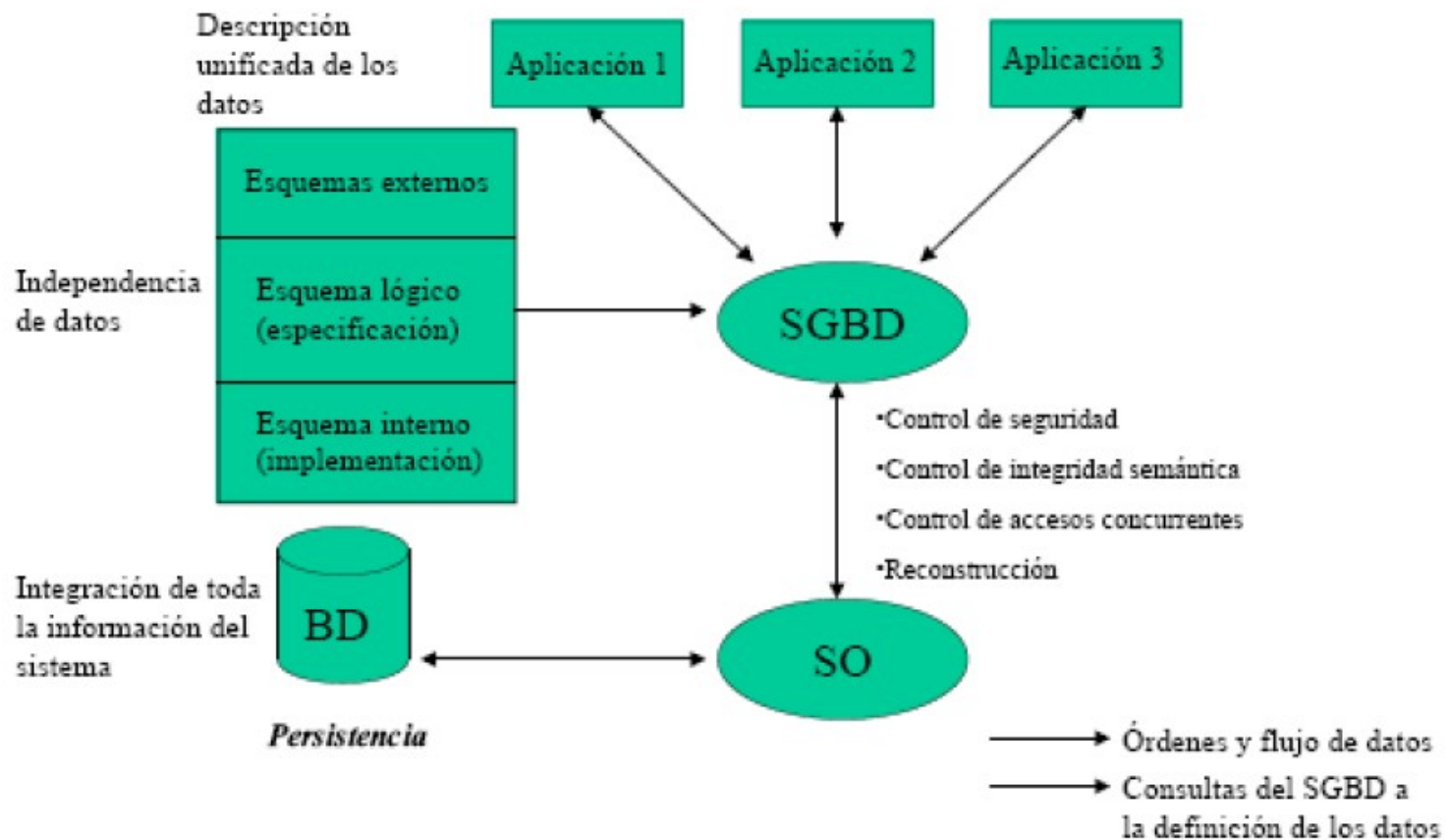
9. Transformación entre niveles

La integración de los tres modelos requiere una serie de transformaciones:

- Transformación **del modelo externo al modelo conceptual** cuando un usuario desde una aplicación accede a los datos.
- Una vez el **esquema conceptual** en el diseño lógico accede a los datos correspondientes, **se pasa al nivel interno** que se encarga de seleccionar la estructura de almacenamiento de datos sobre la que se hace la petición y realiza las operaciones necesarias para dar respuesta al usuario.
- Por último se realiza la transformación contraria.

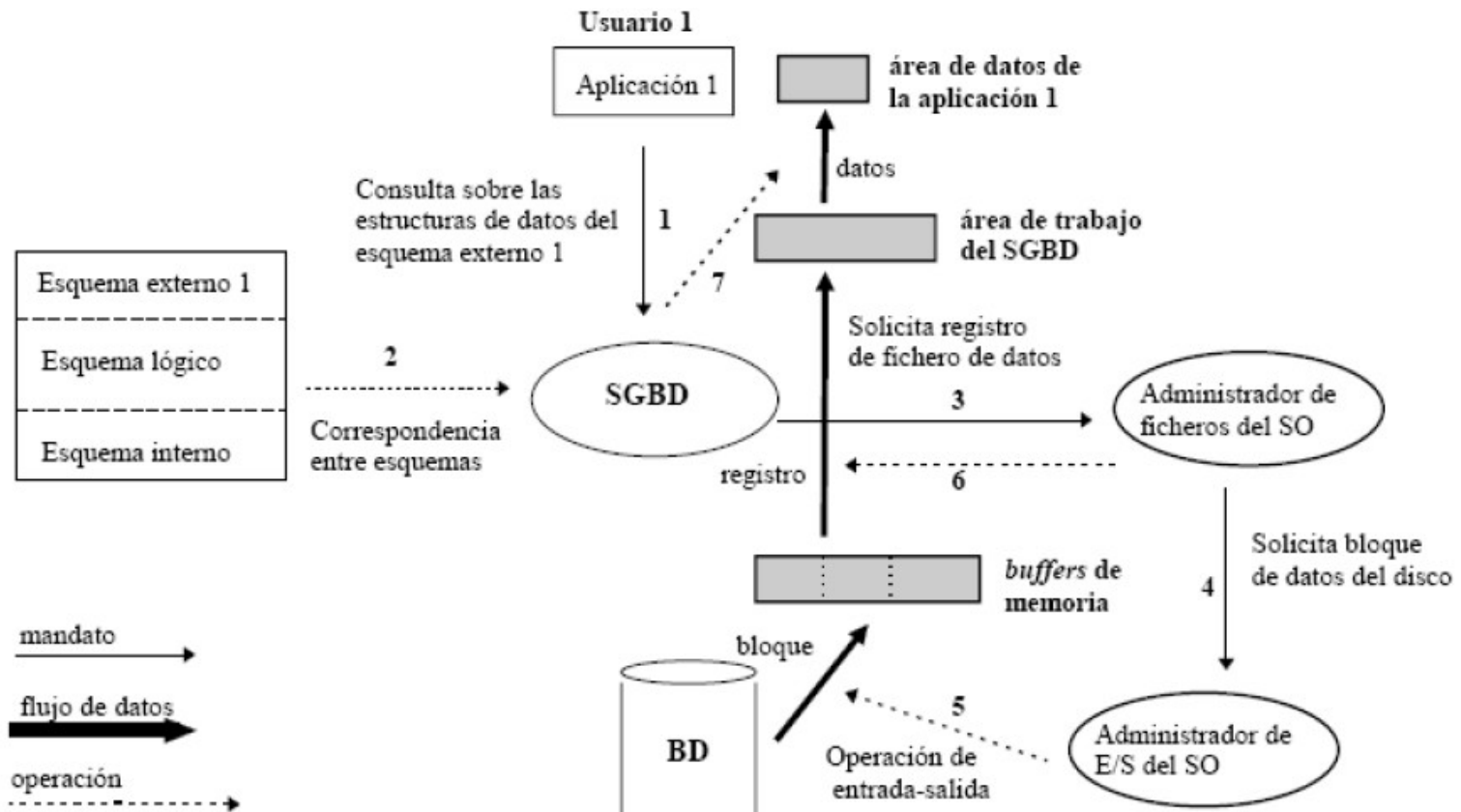


10. Transformación entre niveles



11. Transformación entre niveles

Una petición de la aplicación se resolverá en los diez pasos siguientes:



12. Transformación entre niveles

1. El programa de aplicación hace una llamada al SGBD.
2. El SGBD obtiene el esquema externo de la aplicación y verifica que la petición es correcta.
3. El SGBD obtiene el esquema lógico y realiza la transformación correspondiente entre el esquema externo y el esquema lógico.
4. El SGBD examina mediante el esquema interno la descripción física de los datos requeridos.
5. El SGBD emite órdenes al S.O. indicando qué debe leer y dónde está almacenado.
6. El S.O. interacciona con el almacén físico en el que se encuentran los datos.
7. Los datos se transfieren de la memoria externa (disco) a los buffers intermedios del S.O.
8. El SGBD, comparando el esquema externo correspondiente a la aplicación y el esquema lógico, deduce los datos pedidos por la aplicación realizando las transformaciones necesarias.
9. El SGBD transfiere los datos transformados al área de trabajo del programa que realizó la petición donde éste ya puede operar con ellos.
10. El SGBD suministra información sobre el estado de la petición indicando algún posible error.

13. Componentes de un SGBD

Los componentes mínimos de un SGBD pueden deducirse de las funciones antes especificadas:

- **Núcleo** : Interfaz de la BD con el Sistema Operativo, se encarga de transformar peticiones de la BD en llamadas al Sistema Operativo y de ofrecer facilidades que permiten el acceso al almacenamiento físico.
- **Lenguajes de definición**: Permite la definición y relación entre los diversos elementos de la base de datos. Para ello se utiliza el Lenguaje de Definición de Datos. **(DDL)**. Se encuentran almacenados en un catálogo o diccionario de datos
- **Lenguajes de manipulación** Trabaja sobre la organización de los datos en las tablas, permite ordenar los datos, añadir, borrar y consultar datos. Para ello se utiliza el Lenguaje de consulta y manipulación. **(DML)**

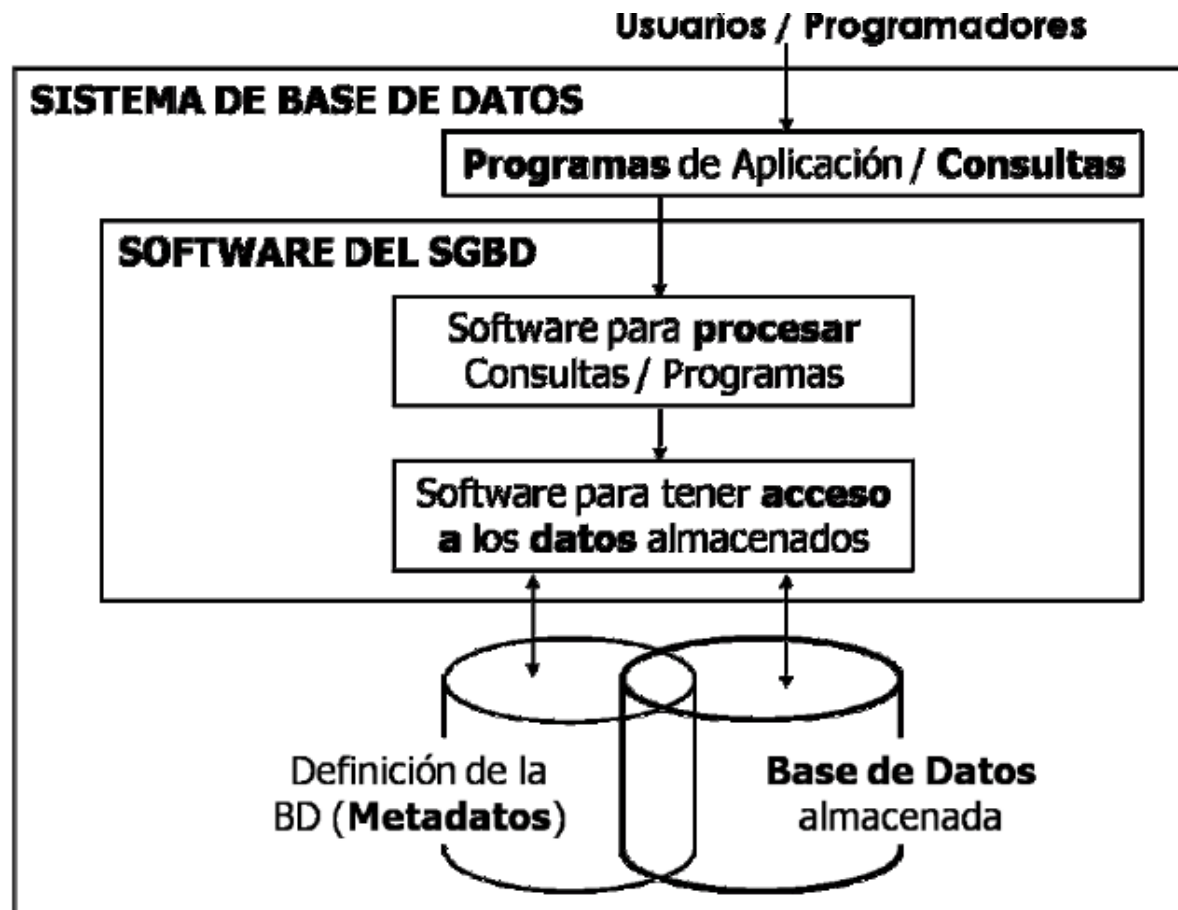


14. Componentes de un SGBD

- **Lenguaje de control:** Permite gestionar usuarios, definiendo perfiles de usuario y asignándole permisos. Estas órdenes también son incluidas en la definición de los datos pero las modificaciones entran en la función de control. (DCL)
- **Diccionario de datos:** Contiene información sobre los elementos que conforman cada una de las bases de datos, información para la manipulación: tablas, registros, campos, relaciones y sus estructuras de modo que siendo esta información legible por los usuarios, puedan hacerse una buena idea de la estructuración de los contenidos de información almacenados en cada base de datos.



15. Componentes de un SGBD



16. Componentes de un SGBD

Componentes auxiliares:

- Módulos de reestructuración: Programas que obtiene copias de la BD.
- Módulos de reconstrucción de BD dañadas.
- Módulos de impresión.
- Módulos de obtención de estadísticas.

17. Estructuras operacionales

CLIENTE SERVIDOR

En una estructura cliente servidor tenemos:

- **La aplicación cliente** es la responsable de verificar y aceptar las entradas de los usuarios. Si se acepta la petición del usuario envía una consulta al servidor de bases de datos basada en la petición de datos anterior. Esta petición es procesado por el servidor, que envía de vuelta a la aplicación cliente los resultados de la consulta. La aplicación cliente formatea los datos de acuerdo con la petición y los muestra al usuario.



18. Estructuras operacionales

CLIENTE SERVIDOR

En una estructura cliente servidor tenemos:

- **La capa de acceso a datos** posee un interfaz de programación de aplicaciones (API) que es el encargado de enviar las consultas al servidor. Esta capa es transparente al usuario. Los programadores pueden crear aplicaciones que interactúan con la capa de acceso a datos. Es independiente del tipo de redes utilizado para comunicar las aplicaciones cliente con el servidor, con lo que hay cierta independencia del software de la base de datos respecto del software de red. JDBC.ODBC



19. Estructuras operacionales

CLIENTE SERVIDOR

En una estructura cliente servidor tenemos:

- **El servidor de base de datos** acepta las consultas de los clientes, las procesa y devuelve los resultados El lenguaje de consulta habitualmente empleado en los Sistemas Gestores de Bases de Datos cliente/servidor es el **SQL**



20. Estructuras operacionales

BASES DE DATOS DISTRIBUIDAS

La distribución de datos a través de las distintas sedes o departamentos de una organización permite que estos datos residan donde han sido generados o donde son más necesarios, pero continuar siendo accesibles desde otros lugares o departamentos diferentes.

El hecho de guardar varias copias de la base de datos en diferentes sitios permite que puedan continuar las operaciones sobre la base de datos aunque algún sitio se vea afectado por algún desastre natural como una inundación, un incendio o un terremoto.



21. Estructuras operacionales

BASES DE DATOS DISTRIBUIDAS

En un sistema distribuido de bases de datos se almacena la base de datos en varias computadoras. Varios medios de comunicación, como las redes de alta velocidad o las líneas telefónicas, son los que pueden poner en contacto las distintas computadoras de un sistema distribuido. No comparten ni memoria ni discos. Las computadoras de un sistema distribuido pueden variar en tamaño y función pudiendo abarcar desde las estaciones de trabajo a los grandes sistemas.

Hay varias razones para construir sistemas distribuidos de bases de datos, incluyendo el compartimiento de los datos, la autonomía y la disponibilidad.



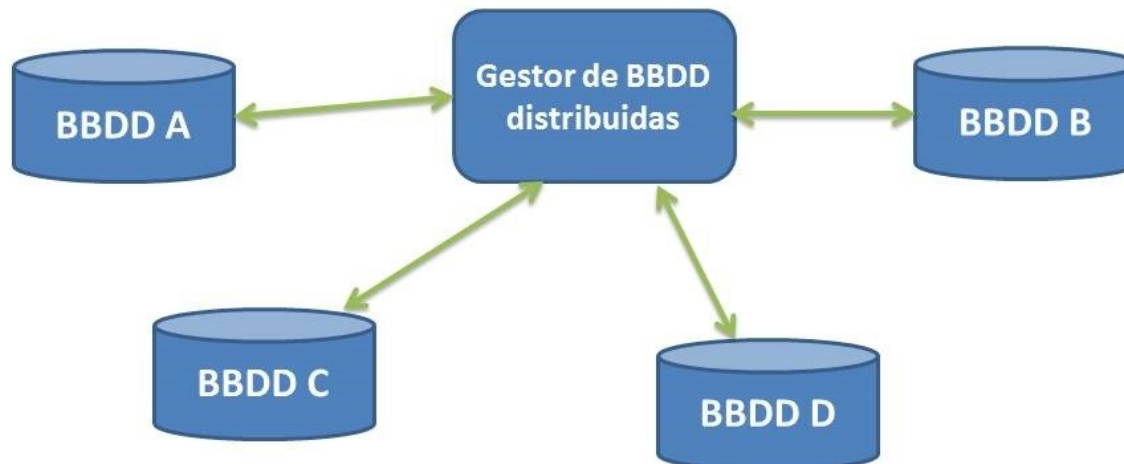
22. Estructuras operacionales

BASES DE DATOS DISTRIBUIDAS

- **Datos compartidos.** La principal ventaja de construir un sistema distribuido de bases de datos es poder disponer de un entorno donde los usuarios puedan acceder desde una única ubicación a los datos que residen en otras ubicaciones.
- **Autonomía.** La principal ventaja de compartir datos por medio de distribución de datos es que cada ubicación (sitio distribuido) es capaz de mantener un grado de control sobre los datos que se almacenan localmente. En un sistema centralizado, el administrador de bases de datos de la ubicación central controla la base de datos. En un sistema distribuido, existe un administrador de bases de datos global responsable de todo el sistema. Una parte de estas responsabilidades se delegan al administrador de bases de datos local de cada sitio

23. Estructuras operacionales

- **Disponibilidad.** Si un sitio de un sistema distribuido falla, los sitios restantes pueden seguir trabajando. En particular, si los elementos de datos están replicados en varios sitios, una transacción que necesite un elemento de datos en particular puede encontrarlo en varios sitios. De este modo, el fallo de un sitio no implica necesariamente la caída del sistema.



24. Arquitectura BD distribuidas

- **Global Schema:** Define todos los datos que están incluidos en la bd distribuida tal como si la bd no fuera distribuida. Consiste de una definición de relaciones globales.
- **Fragmentation Schema:** Traducción entre relaciones globales y fragmentos (parte de la base de datos que está en un sitio distribuido). (Una relación global puede consistir de varios fragmentos, pero un fragmento está asociado con sólo una relación global).
- **Allocation Schema:** Define el sitio (o sitios) en el cual un fragmento está localizado.



25. ¿Cómo almacenar los datos?

Replica

El sistema conserva varias copias o réplicas idénticas de una tabla. Cada réplica se almacena en un nodo diferente.

Ventajas:

- **Disponibilidad:** El sistema sigue funcionando aún en caso de caída de uno de los sitios o nodos.
- **Aumento del paralelismo:** Varios sitios o nodos pueden realizar consultas en paralelo sobre la misma tabla. .

Inconveniente:

- **Aumento de la sobrecarga en las actualizaciones:** El sistema debe asegurar que todas las réplicas de la tabla sean consistentes. Cuando se realiza una actualización sobre una de las réplicas, los cambios deben propagarse a todas las réplicas de dicha tabla a lo largo del sistema distribuido.

26. ¿Cómo almacenar los datos?

Fragmentación

Existen tres tipos de fragmentación la horizontal, la vertical y la mixta

- **Fragmentación Horizontal.** Una tabla T se divide en subconjuntos, T1, T2, ...Tn.

Los fragmentos se definen a través de una operación de selección y su reconstrucción realizará con una operación de unión de los fragmentos componentes. Cada fragmento se sitúa en un nodo.

Ejemplo:

DNI	Escuela	Nombre	Nota ingreso	Beca
87633483	EUI	Concha Queta	5.6	No
99855743	EUI	Josechu Letón	7.2	Si
33887293	EUIT	Oscar Romato	6.1	Si
05399075	EUI	Bill Gates	5.0	No
44343234	EUIT	Pepe Pótamo	8.0	No
44543324	EUI	Maite Clado	7.5	Si
66553234	EUIT	Ernesto Mate	6.6	No

Fragmento de la EUI: $\sigma_{\text{Escuela}=\text{"EUI"}}(T)$

DNI	Escuela	Nombre	Nota ingreso	Beca
87633483	EUI	Concha Queta	5.6	No
99855743	EUI	Josechu Letón	7.2	Si
05399075	EUI	Bill Gates	5.0	No
44543324	EUI	Maite Clado	7.5	Si

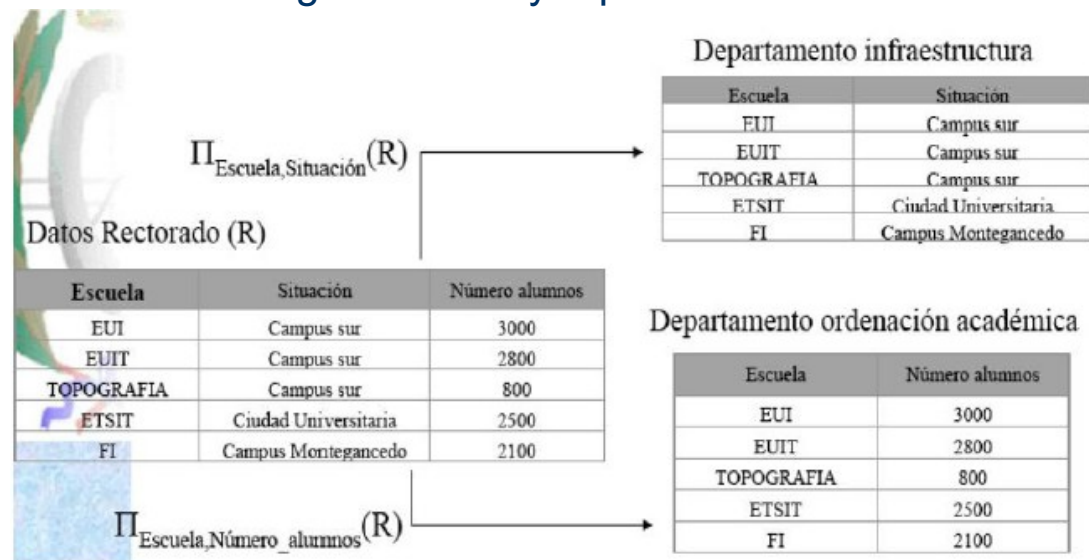
Fragmento de la EUIT: $\sigma_{\text{Escuela}=\text{"EUIT"}}(T)$

DNI	Escuela	Nombre	Nota ingreso	Beca
33887293	EUIT	Oscar Romato	6.1	Si
44343234	EUIT	Pepe Pótamo	8.0	No
66553234	EUIT	Ernesto Mate	6.6	No

27. ¿Cómo almacenar los datos?

- Fragmentación Vertical.** Una tabla T se divide en subconjuntos, T_1, T_2, \dots, T_n . Los fragmentos se definen a través de una operación de proyección. Cada fragmento debe incluir la clave primaria de la tabla. Su reconstrucción se realizará con una operación de join de los fragmentos componentes, pueden existir fragmentos no disjuntos: combinación de fragmentación y replicación.

Ejemplo:



29. ¿Cómo almacenar los datos?

- **Fragmentación Mixta.** Como su nombre indica es una combinación de las dos anteriores.

Réplica y Fragmentación:

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

