

BASES DE DATOS Y SISTEMAS DE INFORMACIÓN

Tema 3



Tema 3

Sistemas de Gestión de Bases de Datos SGBD

ARQUITECTURA ANSI/SPARC

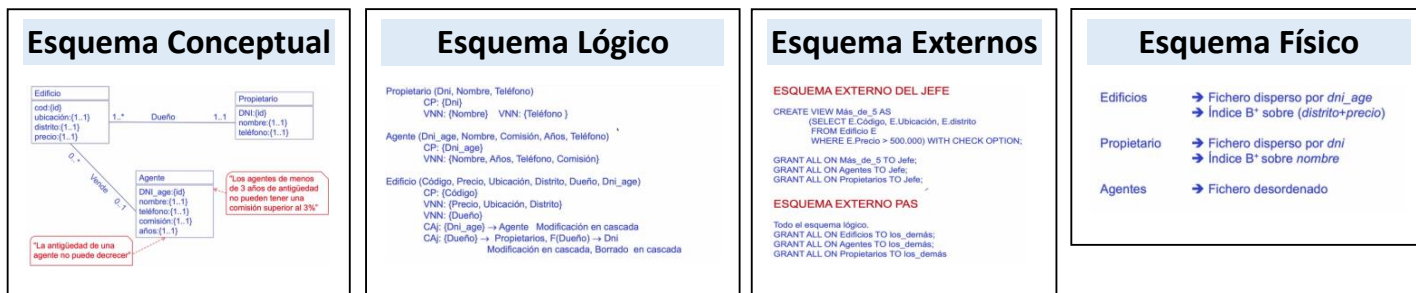
Un SGBD es una Herramienta software que permite la creación, manipulación y gestión de bases de datos. Todas usan un **modelo de datos** para **almacenar datos**. Este conlleva una **estructura de datos** y sus **operaciones asociadas**.

El SGBD debe mantener una independencia en los datos, una integridad y seguridad sobre los datos.

Esquemas y niveles de abstracción de datos

Hay 3 niveles, **conceptual**, **interno** y **externo**, además se pueden definir esquemas en base a ellos + **esquema lógico**. Las definiciones de las bases de datos de un SGBD que soporte niveles y abstracción **deben de cumplir, en cada uno de los niveles y esquemas**: Establecer **correspondencias entre cada uno** (*este dato aquí es este allí*) y **aislar los esquemas**: si toco algo del nivel inferior no haya efectos en los niveles superiores → IDEPENDENCIA DE DATOS.

- **Esquema Conceptual**: Es la **representación gráfica** independientemente del SGBD usado. Es de **alto nivel**.
- **Esquema Lógico**: Es una **definición de los datos** usando términos/lenguaje que se use después en el SGBD. *Como que ya lo has elegido y ahora para definirlo ya te acercas más a lo que será la representación final.*
- **Esquema Interno (físico)**: Es como se **representa físicamente en memoria**. Depende del SGBD.
- **Esquema Externos**: Son las **distintas vistas parciales del sistema** por los **tipos de usuarios**. Cada sistema externo es concreto para cada tipo de usuario, el cual tendrá interés en ciertos datos (no en todos). Se les representa dichos datos usando estructuras del esquema lógico (Tablas, las descripciones de lo ejercicios...).



Funcionamiento básico de un SGBD

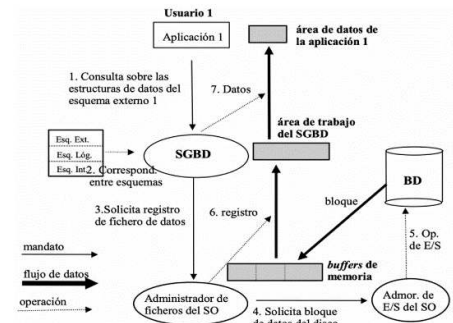
OBJETIVOS DE LAS TÉCNICAS BD	FUNCIONES DEL SGBD	COMPONENTES DEL SGBD
Descripción unificada de los datos e independiente de las aplicaciones	-Definición de la base de datos a varios niveles: esquemas	Lenguajes para la definición de esquemas y los traductores asociados
Independencia de las aplicaciones respecto a la representación física de los datos (Independencia de datos)	-Serían los esquemas de antes: Lógico, interno y externo.	
Definición de vistas parciales de los datos para distintos usuarios	-Además de establecer las correspondencias con los esquemas	
Gestión de la información	-Manipular, Consultar y Actualizar. -Gestión y administración de la base de datos.	Lenguajes de manipulación y traductores. Herramientas para Restructurar, simular e imprimir.
Integridad y seguridad de los datos	Control de integridad, accesos concurrentes, reconstrucción y seguridad de la base de datos.	Herramientas para control de integridad, reconstrucción y control de la seguridad.

Para esto, el SGBD se coloca entre las aplicaciones y los datos, siendo una **capa sobre el sistema operativo**. Estos datos para ser persistente se **guardan en memoria principal**. SGBD es el **único que accede a ellos y a su organización**.

Como funciona por dentro

Se hace una **consulta**, esta le **llega al SGBD** y comprueba lo necesario. Como la consulta **está hecha en base a una vista del usuario**, **traduce la vista del esquema externo al esquema lógico**.

Cuando tiene eso le **pide al SO que lea cierto registro y fichero** (el SO sabe cómo está hecho por dentro y sabe dónde mirar). El SO solicita el Bloque y lo que sea al **administrador de E/S de memoria**. Este **devuelve la información deseada** y a través del buffer se lo hace **llegar** al área de trabajo **donde el SGBD lo pueda usar**.



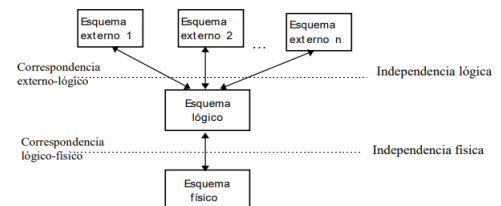
Esto **lo hace en cada iteración**: por cada fila o cosa que te devuelva la consulta SQL hasta que no haya más. Cuando **tenga** seleccionados **todos** los datos, **hace una proyección** sobre esto de lo que le interesa al usuario y lo devuelve.

Independencia de datos

Asegura que los **programas de aplicación** escritos por los usuarios sean **independientes de los cambios realizados en datos que no usan** o en los **detalles de representación física** (implementación) de los datos a los que acceden.

Si uso la tabla "x" y modifico la "y" no tengo problema. Tampoco si uso la tabla "x" y cambias la representación física o algo de esta misma tabla "x". Obvio si uso la "x" y modifico filas/columnas pues le afecta.

Esto se aplica entre cada capa y nivel de los esquemas. El cambio en uno no ha de afectar a los de arriba.



TRANSACCIONES, INTEGRIDAD Y CONCURRENCIA

Las bases de datos tienen que mantener la **integridad de los datos** -> **Calidad de los datos**: Que estén estructurados **reflejando correctamente** los **objetos**, **relaciones** y **restricciones** del mundo real que modelo la base de datos.

Cuando haya cambios en el mundo real se deben reflejar en la BD. El SGBD debe almacenar todo correctamente y cualquier cambio realizado por el usuario debe ser correcto y se hace **integro/permanente** y no se pierde.

Para ello debe comprobar las restricciones de integridad, se ha controlar la concurrencia (que no hay problemas por ella) y que se pueda recuperar la BD en caso de que de haga falta.

Concepto de transacción

Una transacción es una **secuencia de operaciones de acceso a la base de datos** que se hacen como **"una"**. No se comprueban las cosas hasta que no se hayan hecho todas. Cuando hay alguna de estas es **posible que la integridad de la base de datos no se cumpla**. Hay 4 propiedades que ha de cumplir un procesamiento de transacciones.

Operaciones de la transacción: Inicio, anulación de la transacción, confirmación (si ta okey todo éxito, sino falla).

Propiedades para el correcto funcionamiento

- **Atomicidad**: Una transacción es atómica, o se **ejecutan todas** sus operaciones o **no se ejecuta ninguna**.
- **Consistencia**: La transacción **debe dar un estado** de la base de datos **consistente**. Un **estado consistente** es aquel en el que **se cumplen todas las restricciones** de integridad especificadas en el esquema de la BD.
- **Aislamiento**: Una transacción **debe ejecutarse como si estuviera ejecutándose ella sola** y no concurrentemente con otras transacciones. Pueden haber otras pero la **concurrencia no debe dar problemas**.
- **Persistencia**: Cuando una transacción es confirmada, sus **cambios** deben ser **grabados** sobre la base de datos y **no deben perderse** debido a fallos de otras transacciones o del sistema.

Según las **siglas en inglés** a esto de le llama **ACID** (Hola, soy una línea que puedo llenar de palabras... tetas).

Integridad semántica

Cuando pongo una restricción son cosas que se han de cumplir en la BD. Se definen en el esquema lógico y el SGBD se encarga de que se cumplan. Esto lo comprueba tras una transacción. Hay 2 tipos de restricciones de integridad:

- **Restricciones estáticas:** Expresan propiedades que deben satisfacerse **en cada estado** de la BD
 - **Ejemplos:** *Todo propietario tiene al menos 1 edificio. Todo propietario debe tener > 18 años...*
- **Restricciones de transición:** Expresan propiedades que deben **cumplir en cada par de estados consecutivos**. Se comprueba entre el estado que estoy y el que quiero poner/crear tras hacer cambios.
 - **Ejemplos:** *La antigüedad/edad de una persona no puede decrecer... Decrecer implica comprar.*

Restricciones en SQLo

Restricciones estáticas: Se pueden hacer sobre **valores posibles de atributos** (Ej check (value >0)), sobre **atributos** (Ej Not null) y sobre **relaciones**, conjunto de filas de una tabla (Ej primary key).

Cuando hacer comprobaciones: Se puede especificar. Si es **"IMMEDIATE"** cuando quieres que se haga para cada operación de una transacción o **"DEFERRED"** cuando quieres que solo se haga una vez termine la transacción.

Restricciones de integridad en CAjenas: También con los "update TO null cascade" y "on delete..." se gestiona.

Control de accesos concurrentes

Lo de siempre que, si condiciones de carrera y datos resultantes sin sentido, incoherentes... Posibles problemas:

1. Pérdida de actualizaciones.
2. Obtención de información incoherente correspondiente a varios estados válidos diferentes de la BD.
3. Lectura de datos actualizados (*no confirmados*) que han sido sometidos a cambios que pueden ser anulados.

Técnicas para prevenir problemas: **Reserva de ocurrencias de datos (locks)** que cuando se empieza a hacer algo sobre la base de datos bloquea otras acciones sobre ella para que se hagan luego (requiere control de deadlocks). También está la **anulación de transacciones** si se detecta que causan problemas o **hacer transacciones aisladas** evitando que estas lean datos que aún no han sido confirmados por otras.

RECUPERACIÓN Y SEGURIDAD

Dos cosas principales: **Recuperación** (*forma parte de la integridad pero no como consistencia*), que una base de datos se **pueda recuperar ante cualquier fallo** (no perder información). Y **seguridad**, **evitar accesos NO autorizados**.

Recuperación

Las copias de seguridad son **útiles pero NO suficientes**: *Si hago copia hoy, después hago cambios y luego peta, todos los cambios que he hecho los pierdo, malamente*. Se quiere que **NO se pierda ningún dato que hay sido confirmado**.

Fallos del sistema: pérdida de memoria principal, La volátil

Causadas por Interrupción del suministro eléctrico, error del software del SGBD, error del sistema operativo, virus informáticos, etc. Ya que las transacciones no se pasan a memoria secundaria (persistente) cada vez que se hace commit, o sea que pueden estar un rato memoria principal, podemos confirmar algo y que pierda.

Herramientas: Para evitar perder esta información tenemos **Ficheros diario** (*log o journal*) y **Puntos de control**.

- **Fichero diario:** Registra las operaciones de actualización de las transiciones y las almacena en disco.
- **Punto de control:** PARA todo y guarda en disco TODO lo confirmado hasta ahora. *El fich diario es notificado.*

Fallos del sistema de almacenamiento: pérdida de memoria secundaria, La persistente

Fallo en la lectura de los discos, fallos del controlador de discos, accidentes o catástrofes, virus informáticos, etc. Lo que se hace es **volcar la última copia de seguridad** y **ejecutar de nuevo todas las transacciones** que se han hecho desde que se hizo la copia de seguridad. Estas transacciones han de estar apuntadas en el diario.

Seguridad

Es la privacidad de la información. Que solo puedan acceder a la información quienes estén autorizados y de la forma autorizada. Hace falta identificar a cada usuario y determinar que cosas puede y no puede hacer con cada elemento.

- **Listas de autorizados:** Por cada objeto se dice si el usuario puede cosas una u otra cosa: leer, cambiar...
- **Niveles de permisos:** Cada nivel tienes ciertos permisos. *Menso flexible no se usa.*

Estos permisos han de ser traspasables: Que un usuario le pueda dar o quitar permisos a otro etc.

SQLo: Se usa "GRANT" y "REVOKE". Especificando qué permiso y si ese permiso luego se puede volver a dar.