

SGBD: Introducción a las unidades físicas y unidades lógicas

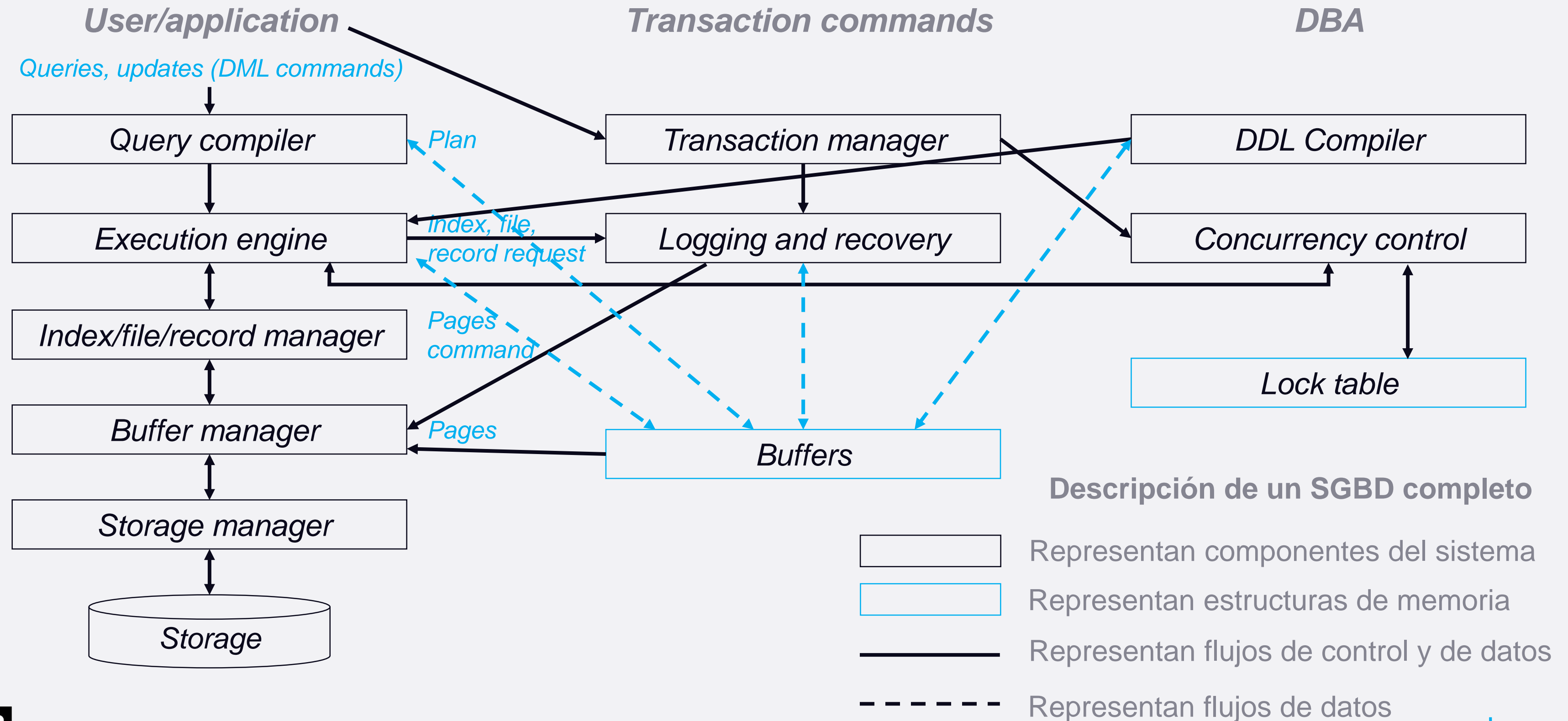
Unidades físicas:

- Almacenan los datos y la parte que describe la estructura de los datos (datos y metadata) (DB file)
- Almacenan la información necesaria para volver la base de datos a un estado consistente ante la ocurrencia de fallas que interrumpan el funcionamiento de la base de datos (LOG file).

Unidades lógicas:

- Porciones de memoria que procesan consultas y operaciones de definición y/o de manipulación para asegurar satisfacer la “**prueba de ACID**” para las transacciones, estas son:
 - Administrador de almacenamiento: Compuesto por el administrador de memoria, el administrador de lockeos (*dedlocks*), el administrador de I/O y el administrador de *fetch* o procesos.
 - Administrador de consultas: Básicamente compuesto por el administrador de parseo, por el administrador de optimización y el motor de ejecución, en algunos casos un compilador puede ser quien ejecute.
 - Administrador transacciones: Compuesto por un gestor de transacciones (*transaction manager*).

Unidades lógicas



SGBD: Análisis del gráfico

Existen dos fuentes de comandos:

- Los de los programas de aplicación y de los usuarios convencionales que consultan o modifican datos.
- Los de los DBAs, responsables de la estructura o esquema de la base de datos.

Análisis de los componentes del sistema

Procesamiento de los comandos del lenguaje de definición de datos (LDD):

- El procesador de comandos DDL (*DDL Compiler*) recibe y analiza los comandos de definición de datos y alteración de esquemas.
- Envía los comandos al motor de ejecución (EE - *Execution Engine*)
- El motor de ejecución pasa los datos a través del **administrador de registros, archivos e índices** para alterar la metadata (información del esquema de la BD)

Procesamiento de consultas:

Se inicia una acción por medio del lenguaje DML o de una consulta (*query*)

- Las sentencias DML se ejecutan mediante dos subsistemas:
 - Responder la consulta (*query*)
 - Procesamiento de transacciones (DML)

SGBD: Análisis del gráfico (continuación)

Procesamiento de consultas:

Responder la consulta:

- El compilador de consultas (*query compile*) analiza la consulta y la optimiza armando el plan de consulta para responderla. Pasa el plan al motor de ejecución (EE -- *Execution Engine*).
- El motor de ejecución emite una secuencia de solicitudes de pequeñas porciones de datos (registros o tuplas de relaciones) a un administrador de recursos que conoce los archivos de datos (*data files*) que contienen las relaciones, el tamaño y formato de los registros en esos archivos, y los archivos de índices, que ayudan a encontrar elementos de los archivos de datos rápidamente.
- La solicitud de datos se pasa al gestor de buffers (*buffer manager*).
- El *buffer manager* lleva las porciones apropiadas de los datos desde el almacenamiento secundario a los *buffers* de memoria principal (generalmente la página o bloque de disco es la unidad de transferencia entre *buffer* y disco).
- El *buffer manager* se comunica con el gestor de almacenamiento (*storage manager*) para obtener los datos del disco.
- El *storage manager* puede involucrar comandos del sistema operativo, pero lo más común es que el SGBD emita comandos directamente al controlador de disco (*disk controller*).

SGBD: Análisis del gráfico (continuación)

El procesador de consultas (*query processor*)

- Es la parte del SGBD que más afecta el rendimiento que ve el usuario.
- Está representado por dos componentes:
 - a. **Compilador de consultas (*query compiler*):**
 - Encargado de traducir la consulta en una forma interna llamada plan de consulta (*query plan*).
 - *Query plan*: secuencia de operaciones a ser ejecutada sobre los datos, generalmente implementada como operaciones de algebra relacional.
 - Utiliza los metadatos y estadísticas acerca de los datos para decidir qué secuencia de operaciones es la mejor candidata a ser la más rápida.
 - El compilador consta de tres partes:
 1. *Query parser*: Analizador que construye una estructura de árbol del texto de la consulta.
 2. *Query preprocessor*: Preprocesador de la consulta que lleva a cabo la comprobación semántica (por ej.: que existen las relaciones mencionadas en la consulta) y ejecuta algunas transformaciones en el árbol para convertirlo en un árbol de operadores algebraicos que representen el plan de consulta inicial.
 3. *Query optimizer*: Optimizador de consultas que transforma el plan de consulta inicial en la mejor secuencia disponible de operaciones sobre los datos actuales.

SGBD: Análisis del gráfico (continuación)

El procesador de consultas (*query processor*)

b. Motor de ejecución (*execution engine*):

- Encargado de ejecutar cada uno de los pasos seleccionados en el plan de consulta.
- Interactúa con la mayoría de los otros componentes del SGBD directamente o a través de los *buffers*.
- Lleva los datos desde la BD a los *buffers* con el objetivo de manipularlos.
- Interactúa con el planificador (*shceduler*) para evitar el acceso a los datos bloqueados y con el gestor de *logs* para asegurarse de que se registren todos los cambios en la BD.

SGBD: Análisis del gráfico (continuación)

Procesamiento de transacciones

- Los comandos DML se agrupan en transacciones (unidades que deben ser ejecutadas atómicamente y aisladas unas de otras).
- Una acción DML puede ser una transacción en sí misma.
- La ejecución de la transacción debe ser durable/perdurable, el efecto producido por una transacción que se ejecutó en forma completa debe ser preservado siempre, aún ante fallas del sistema.
- El procesador de transacciones se divide en dos partes:
 - Gestor del control de concurrencia (*concurrency control*): asegura la atomicidad y aislamiento de la transacción.
 - Gestor de recuperación y log (*logging and recovery*): responsable de la durabilidad de las transacciones.

SGBD: Análisis del gráfico (continuación)

Gestión de *buffers* y almacenamiento

- Su función es controlar la ubicación de los datos en disco y su movimiento entre el disco y la memoria principal.
- Los SGBD generalmente controlan el almacenamiento en disco directamente, bajo determinadas circunstancias y por razones de eficiencia.
- El gestor de almacenamiento lleva registro de la ubicación de los archivos en el disco y obtiene el bloque o bloques de un archivo según la solicitud del gestor de *buffers*.
- El gestor de *buffers* es el responsable del particionamiento de la memoria principal disponible en *buffers*.
- Los *buffers* disponibles son regiones de memoria donde se pueden transferir los bloques.
- Todos los componentes del SGBD que necesiten información del disco van a interactuar con los *buffers* y el gestor de *buffers*, que podrá ser directamente o a través del motor de ejecución.

SGBD: Análisis del gráfico (continuación)

Gestión de *buffers* y almacenamiento

- La clase de información que se puede necesitar incluye:
 - Datos: el contenido de la BD.
 - Metadatos: el esquema de la base de datos que describe la estructura y las restricciones de la BD.
 - Registros de log: información acerca de los cambios recientes a la BD.
 - Estadísticas: información reunida y almacenada por el SGBD acerca de las propiedades como tamaños y valores de las distintas relaciones y otros componentes.
 - Índices: estructuras de datos que respaldan el acceso eficiente a los datos.

SGBD: Análisis del gráfico (continuación)

Procesamiento de transacciones

- Un SGBD ofrece garantía de durabilidad sobre las transacciones.
- El gestor de transacciones (*transaction manager*) acepta comandos de transacción de una aplicación.
- La aplicación informa al gestor de transacciones cuando comienzan y cuando terminan las transacciones, y las expectativas sobre las mismas (alguna transacción podría solicitar la no atomicidad).
- Procesador de transacciones, tareas que realiza:
 - *Logging*
 - Control de concurrencia
 - Resolución de *deadlocks*

SGBD: Análisis del gráfico (continuación)

Procesamiento de transacciones

- Procesador de transacciones, tareas que realiza:
 - *Logging*:
 - Cada cambio en la BD se registra separadamente en disco, asegurando la durabilidad (ACID).
 - El gestor de *log* aplica alguna política para asegurar que ante una falla o caída de la BD, un gestor de recuperación va a examinar el registro de cambios y restaurar la BD a un estado consistente.
 - El gestor de *log* primero escribe el *log* en *buffers* y “negocia” con el gestor de *buffers* para asegurarse la escritura en disco a un tiempo apropiado en los archivos donde los datos puedan sobrevivir a una caída.

SGBD: Análisis del gráfico (continuación)

Procesamiento de transacciones

- Procesador de transacciones, tareas que realiza:
 - *Control de concurrencia:*
 - Las transacciones deben parecer que se ejecutan aisladamente a pesar de que hay muchas transacciones ejecutándose simultáneamente (ACID).
 - El planificador o *scheduler* (gestor de control de concurrencia) debe asegurar que las acciones individuales de múltiples transacciones se ejecuten en un orden tal que el efecto neto sea el mismo que si las transacciones se hubiesen ejecutado en forma completa una por vez.
 - El planificador utiliza bloqueos (*locks*) en ciertas piezas de datos para prever que dos transacciones accedan a la misma pieza de datos de forma que interactúen perjudicialmente.
 - Los bloqueos son mantenidos en la tabla de bloqueos (*lock table*) de memoria principal.
 - El planificador afecta la ejecución de consultas y otras operaciones de la BD prohibiéndole al motor de ejecución el acceso a las partes bloqueadas de la BD.

SGBD: Análisis del gráfico (continuación)

Procesamiento de transacciones

- Procesador de transacciones, tareas que realiza:
 - *Resolución de deadlocks:*
 - Las transacciones compiten por recursos a través de los bloqueos que concede el planificador.
 - Puede ocurrir que ninguna transacción pueda proseguir debido a que cada transacción necesita algo de lo que se bloqueó por otra transacción.
 - El procesador de transacciones interviene y cancela una o más transacciones para que otras puedan seguir su ejecución.

Índices

- El índice de una BD es una estructura de datos que mejora la velocidad de las operaciones para permitir un rápido acceso a los registros de una tabla en una BD.
- Los índices se suelen utilizar en aquellos campos sobre los cuales se hacen búsquedas en forma frecuente.
- Los índices pueden ser creados usando una o más columnas, proporcionando la base tanto para búsquedas rápidas al azar como de un ordenado acceso a registros eficiente.
- El espacio en disco requerido para almacenar el índice es típicamente menor que el espacio de almacenamiento de la tabla ya que generalmente contienen solo los campos que componen el índice y excluyen el resto de los campos de la tabla.
- En una BD relacional un índice es una copia de parte de la tabla.
- Algunas BD amplían la potencia del indexado al permitir que los índices sean creados de funciones o expresiones, nativas de la BD o definidas por el usuario, por ejemplo un índice sobre la función UPPER (APELLIDO) el cual almacenaría en el índice solamente las versiones mayúsculas del campo apellido.
- Un índice puede ser definido como único o no único. Un índice único actúa como una restricción en la tabla previniendo filas idénticas en el índice.

Índices

Tipos básicos de índices:

- Índices ordenados: basados en una disposición ordenada de los valores.
- Índices asociativos (*hash index*): basados en una distribución uniforme de los valores a través de una serie de cajones (*buckets*). El valor asignado a cada cajón (*bucket*) está determinado por una función de asociación (*hash function*)

Índices

Técnicas de indexación y asociación: No hay una técnica mejor que otra, cada técnica será la más apropiada para una aplicación específica de BD. Las técnicas serán valoradas según los siguientes criterios:

- Tipos de acceso: Tipos de acceso soportados eficazmente, pueden incluir la búsqueda de registros con un valor concreto en un atributo o para un rango específico de valores.
- Tiempo de acceso: Tiempo aplicado en la búsqueda de un determinado elemento de dato, o conjunto de elementos.
- Tiempo de inserción: Tiempo empleado en insertar el nuevo elemento de datos. Este valor incluye el tiempo empleado en la búsqueda del lugar apropiado para donde insertar el nuevo elemento de datos y el tiempo empleado en actualizar la estructura del índice.
- Tiempo de borrado: Tiempo empleado en borrar un elemento de dato. Este valor incluye el tiempo empleado en buscar el elemento a borrar y el tiempo empleado en actualizar la estructura del índice.
- Espacio adicional requerido: El espacio adicional ocupado por la estructura del índice. Muchas veces es razonable sacrificar el espacio para alcanzar un mejor rendimiento en la búsqueda de los datos.



© Universidad de Palermo

Prohibida la reproducción total o parcial de imágenes y textos.