

Sistemas de gestión de bases de datos y lenguajes de consulta

Miguel Expósito Martín

Universidad de Cantabria

miguel.exposito@unican.es

26/11/2018

Visión general

1 Definición

2 Arquitectura

3 Categorización

Definición

Un sistema de gestión de bases de datos (SGBD) es una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, a la que se conoce normalmente como base de datos, contiene información relevante para una organización. El objetivo primordial de un SGBD es ofrecer un medio conveniente y eficiente para almacenar y recuperar información de la base de datos.[[Silberschatz, 2010](#)]

Un SGBD es un paquete de software utilizado para definir, crear, utilizar y mantener una base de datos. Típicamente consiste de varios módulos software, cada uno con su propia funcionalidad.[[Lemahieu, 2018](#)]

Valor de mercado

Según Gartner, el valor de mercado total de los SGBD en 2015 era de 35.9 billones de dólares, un 8,7 % más comparado con el año anterior.

Architecture of a DBMS

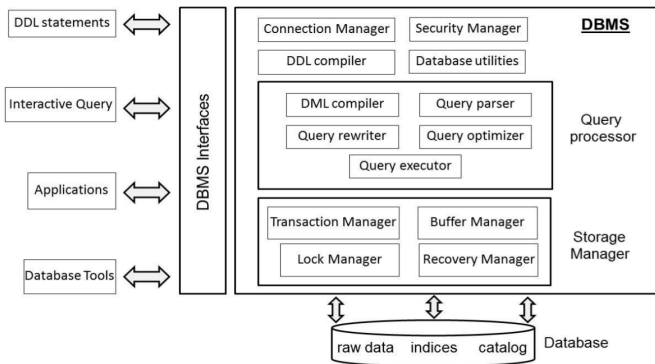


Figura: Arquitectura de SGBDs.¹

¹Fuente: [Lemahieu, 2018]

Arquitectura: gestor de conexiones y seguridad

- Proporciona facilidades para iniciar una conexión (remota) a una base de datos.
- Verifica credenciales de inicio de sesión (usuario/contraseña) y devuelve un manejador de conexión.
- Verifica si un usuario tiene los privilegios suficientes para ejecutar las acciones requeridas.

Arquitectura: compilador de DDL

- Compila las definiciones de datos especificadas en DDL(*Data Definition Language*)
- Parsea las definiciones de DDL, comprueba su corrección sintáctica y lo traduce a un formato interno.
- Una vez compilado con éxito, se registran las definiciones de datos en el catálogo.

Arquitectura: procesador de consultas

Una de las partes más importantes de un SGBD, asiste en la ejecución de consultas a la base de datos tales como recuperación, inserción, actualización o borrado de información. Generalmente consta de los siguientes componentes:

- **Compilador DML** (*Data Manipulation Language*): compila las sentencias de manipulación de datos. SQL es un DML declarativo.
- **Procesador de consultas**: parsea la consulta a un formato de representación interno. Puede reescribir la consulta para optimizarla independientemente del estado de la BD, utilizando un conjunto de reglas predefinidas específicas del SGBD.
- **Optimizador de consultas**: optimiza la consulta en función del estado de la BD, evaluando los distintos planes de ejecución. Estas estimaciones se llevan a cabo utilizando el catálogo y procedimientos de inferencia estadística.
- **Ejecutor de consultas**: se encarga de la ejecución de la consulta devuelta por el optimizador recuperando los datos solicitados desde el gestor de almacenamiento.

Arquitectura: procesador de consultas

- Gestor de almacenamiento: gobierna el acceso físico a los archivos y supervisa el correcto almacenamiento de datos. Consta de:
 - Gestor de transacciones: supervisa la ejecución de transacciones, creando una planificación con operaciones de lectura y escritura mejorando la eficiencia y rendimiento globales y garantizando atomicidad, consistencia, aislamiento y durabilidad (ACID).
 - Gestor de buffer: gestiona la memoria de buffer del SGBD. Es responsable de su cacheo y monitorización.
 - Gestor de bloqueos: juega un importante papel en el control de concurrencia, asegurando la integridad de los datos en todo momento. Evita conflictos entre transacciones previniendo actualizaciones simultáneas sobre un mismo elemento de datos.
 - Gestor de recuperación: recoge todas las operaciones de base de datos en un fichero de histórico, pudiendo ser utilizado para deshacer acciones o transacciones abortadas, así como para recuperación de caídas.

Arquitectura: utilidades e interfaces

- Un SGBD cuenta con diversas utilidades:
 - Utilidad de carga: soporta la importación de datos en la BD desde diferentes fuentes.
 - Utilidad de reorganización: reorganiza automáticamente la BD para mejorar su rendimiento.
 - Utilidades de monitorización: tiempos de respuesta, espacio de almacenamiento, etc.
 - Utilidades de gestión de usuarios: usuarios, roles y privilegios.
 - Utilidades de copia de seguridad y recuperación.
- Un SGBD necesita interactuar con diversos actores: diseñadores, administradores, aplicaciones, usuarios finales... Para facilitar dicha comunicación, se proporcionan varios interfaces de usuario (web, línea de comandos, GUI de escritorio, etc.). Ejemplo: **MySQL Workbench**.

Categorización

- Basada en el modelo de datos
 - Jerárquica: modelo basado en árbol. Procedural y orientada a registro. No tiene procesador de consultas. Ejemplo: el registro de *Windows*.
 - De red: modelo basado en red, más flexible que el jerárquico. Procedural y orientada a registro, tampoco tiene procesador de consultas. CODASYL es uno de los tipos de BD jerárquica más populares.
 - Relacional: utiliza el modelo relacional, el más usado en la industria. SQL se utiliza tanto para DML como para DDL. Es declarativo y orientado a conjuntos. Sí tiene procesador de consultas y presenta separación entre modelo lógico y de datos internos. Ejemplos: *Oracle*, *MySQL*, *SQL Server*...

Categorización

- Basada en el modelo de datos
 - Orientada a objetos: basada en el modelo del mismo nombre. Ejemplo: *db4o*. No son muy populares en la industria debido a su complejidad.
 - Relacional extendida: utiliza el modelo relacional extendido con conceptos de orientación a objetos. La mayor parte de las BBDD relacionales incorporan este tipo de extensiones.
 - XML: utiliza el modelo XML para almacenar datos. Las BBDD relacionales también incorporan extensiones para XML. Ejemplo: *eXist*.
 - NoSQL (documentos, grafos): almacenes clave-valor, de documentos (JSON), de grafos (RDF)... Ejemplos: *MongoDB*, *Neo4j*.
- Basada en el grado de acceso simultáneo
 - Monousuario: solo un usuario a la vez. Ej: *MS Access*.
 - Multiusuario: múltiples usuarios interactuando simultáneamente.

Categorización

- Basada en la arquitectura
 - Centralizada: mainframes que procesan todas las consultas en un único host.
 - Cliente-servidor: clientes activos solicitan servicios a servidores pasivos.
 - N-capas: GUI + servidor de aplicaciones + BD.
 - En la nube: alojada por terceros, pueden distribuir los datos entre múltiples máquinas. Ej: *Cassandra*.
 - Federada: proporciona una interfaz uniforme a múltiples fuentes de datos subyacentes.
 - En memoria: almacena los datos en memoria. Se utilizan para propósitos de tiempo real.

Categorización

- Basada en el tipo de uso
 - Transaccional (OLTP): enfocada en gestión operacional o transaccional (ej: TPV). Deben tener buen soporte para grandes volúmenes de consultas pequeñas.
 - Analítica (OLAP): enfocada en utilizar datos operativos con enfoques tácticos o estratégicos. En este paradigma, un número limitado de usuarios lanza consultas complejas para analizar altos volúmenes de datos.
 - Multimedia: permite almacenar datos en forma de texto, imágenes, audio, vídeo...Proporcionan motores de consulta basados en el contenido.
 - Espacial: soporta el almacenamiento y consulta de datos espaciales. Es la base de los GIS. Ej: PostgreSQL.
 - Móvil: se ejecuta sobre *smartphones*. Presenta una baja huella de memoria, ideal para potencia de cálculo limitada. Ejemplos: *SQLite*, *Oracle Lite*, *SQL Server Compact*, etc.

Resumen

Referencias



Abraham Silberschatz et al. (2010)

Database System Concepts, 6th edition



Wilfried Lemahieu et al. (2018)

Principles of Database Management