

Introducción a los Sistemas de Gestión de Bases de Datos

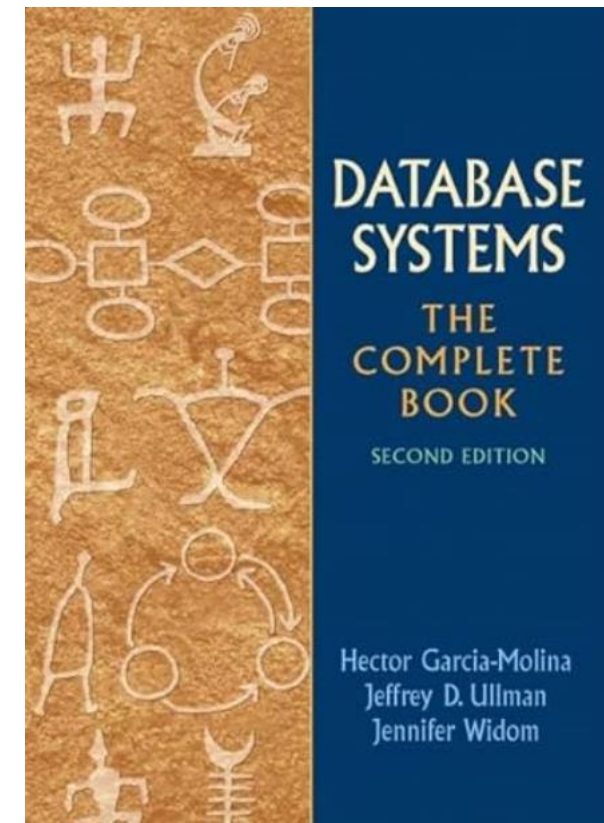
Autor: Sergio D'Arrigo

¿Qué es una base de datos?

- Analicemos los resultados de la consulta...



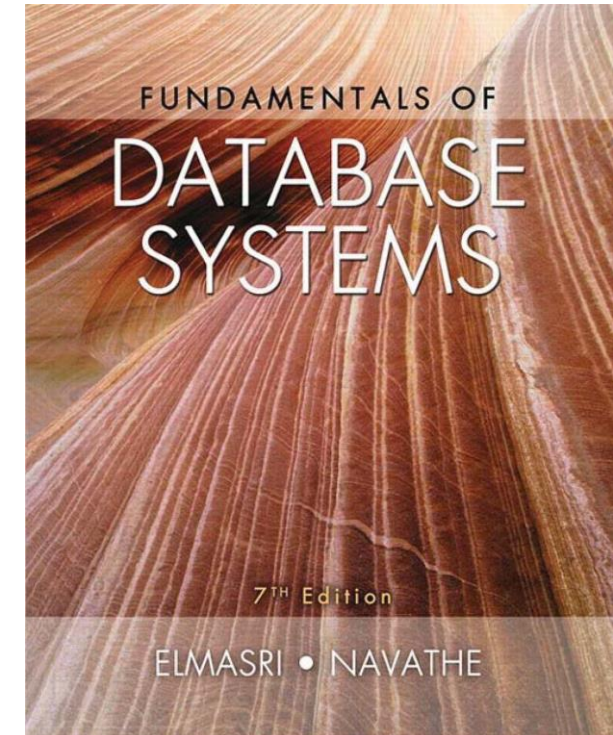
¿Qué es una base de datos?



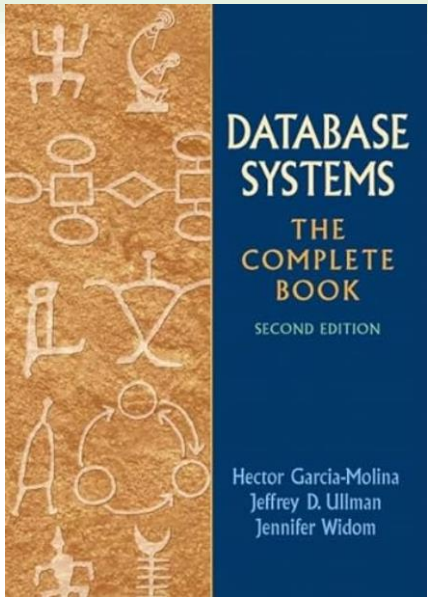
- Es una **colección de información** que existe durante **largos períodos de tiempo**, a menudo muchos años.
- En lenguaje común, base de datos se refiere a una colección de datos que son **gestionados por un Sistema de Gestión de Base de Datos** (DBMS)

¿Qué es una base de datos?

- Una base de datos es una **colección de datos relacionados**.
 - **Datos**: *Hechos conocidos que pueden ser registrados y que tienen un significado implícito.*
- En un sentido no tan general, una base de datos tiene las siguientes propiedades:
 - Representa algún aspecto del mundo real, llamado minimundo o **universo de discurso**
 - Es una colección **lógicamente coherente** de datos con algún significado inherente
 - Está diseñada, construida y poblada para algún **propósito específico**

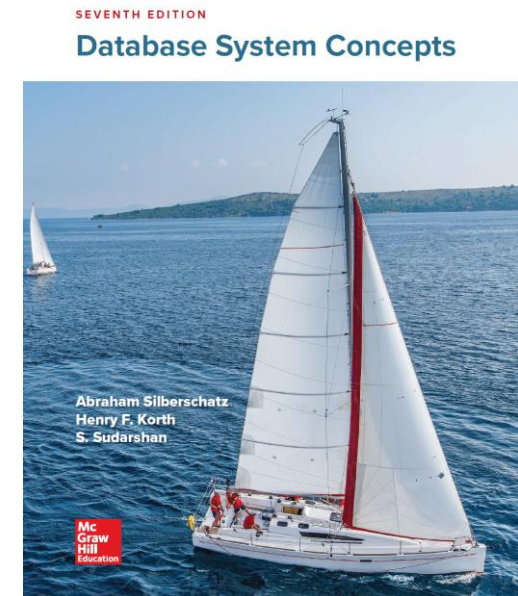


¿Qué es un Sistema de Gestión de Base de Datos?



- Un DBMS es una herramienta poderosa para **crear y gestionar grandes cantidades de datos** de manera **eficiente** y permitir que persistan durante **largos períodos de tiempo** de forma **segura**.

- Un sistema de gestión de bases de datos (DBMS) es una **colección de datos interrelacionados** y un **conjunto de programas para acceder a esos datos**.
- El objetivo principal de un DBMS es proporcionar una forma de **almacenar y recuperar información** de la base de datos que sea tanto **conveniente** como **eficiente**.

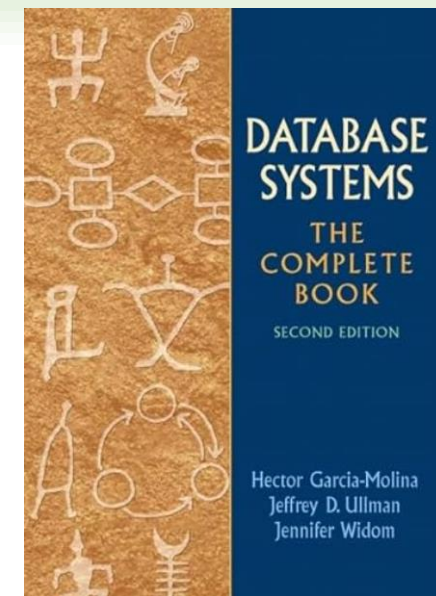


¿Esto es una base de datos?

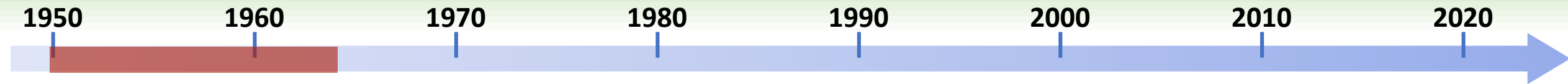
- Un conjunto de archivos Excel y/o csv con cierta cohesión lógica, en carpeta del disco rígido?
- Veamos algunos ejemplos de uso, y algunas limitaciones...
- Tamaño grande (millones de registros)
- Que pasa si dos usuarios tomamos a la vez el saldo de un mismo csv para comprar distintas cosas?
- Que pasa si estamos transfiriendo plata de una cuenta a otra (su saldo está en diferentes excel) y luego de guardar el primer Excel se nos corta la luz?
- Y si alguien visualiza el archivo y tiene algunos datos confidenciales?

¿Qué se espera de un SGBD?

- ☐ Permite a los usuarios **crear** nuevas bases de datos y especificar sus esquemas.
- ☐ Brinda a los usuarios la capacidad de **consultar** y **modificar** los datos.
- ☐ Soporta el **almacenamiento de cantidades muy grandes** de datos durante un largo período de tiempo, permitiendo un **acceso eficiente** a los mismos.
- ☐ Permite la **durabilidad**, es decir, la recuperación de la base de datos ante fallos, errores de diversos tipos o uso indebido intencional.
- ☐ Controla el acceso a los datos de muchos usuarios a la vez:
 - ☐ Sin permitir interacciones inesperadas entre los usuarios (**aislamiento**)
 - ☐ Sin que las acciones sobre los datos se realicen parcialmente (**atomicidad**).



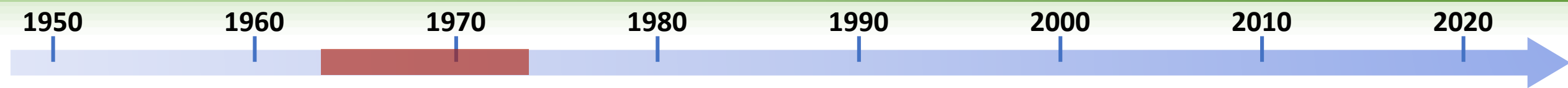
Algo de historia...



- Almacenamiento de datos en cintas magnéticas
 - Para procesar datos, se leía el input de cintas y se persistía el output en una nueva cinta
 - El acceso era **secuencial**, y los datos estaban **en un único orden**
- ✂ *Era muy importante el orden de los datos para agilizar el procesamiento*

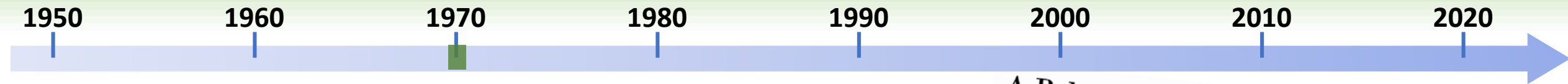


Algo de historia...



- Almacenamiento de discos rígidos, con acceso directo a los datos.
 - Surgen distintos modelos de datos para describir la estructura. Se destacan:
 - **Jerárquicos**: basados en árboles. IBM IMS se sigue usando en la actualidad en mainframes.
 - **De red**: basados en grafos, estandarización CODASYL a fin de los 60s). CA IDMS se sigue usando en la actualidad en mainframes.
 - Los sistemas **no soportaban lenguajes de consulta de alto nivel**.
- ✂ *Los programadores tenían que visualizar los datos **tal como estaban almacenados**, y programar la manera de saltar de un dato a otro, **con mucho detalle**.*

Algo de historia...



1970: **Edgar F. Codd** publica el paper ***"A Relational Model of Data for Large Shared Data Banks."***

Este es un **artículo clave**:

Se define el modelo relacional y formas no procedurales de consultar datos en dicho modelo.

*A partir de este paper, surgieron las **bases de datos relacionales**.*

Modelo simple, que permite al programador trabajar con los datos sin necesidad de conocer detalles de implementación

A Relational Model of Data for Large Shared Data Banks

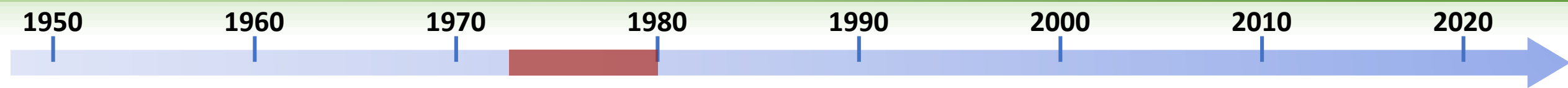
*E. F. Codd
IBM Research Laboratory, San Jose, California*

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information. Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity

CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

Algo de historia...



Comenzó el tiempo de las **Bases de Datos Relacionales!!**

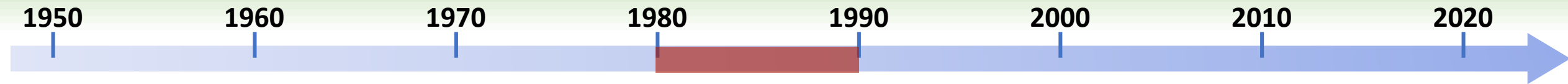
- Basados en paper de Codd, con una base teórica muy sólida.
- Los sistemas presentan a los usuarios vistas de los datos organizados como **tablas**, que implementan **relaciones** (en el sentido de la Teoría de Conjuntos).
- Las consultas de los datos se realizan con un lenguaje de alto nivel **declarativo**: el más importante es SQL (Structured Query Language)

✂ *Los desarrolladores no tienen que preocuparse de la estructura de almacenamiento de los datos.*

Los desarrollos iniciales no eran competitivos en términos de performance con las BBDD jerárquicas y de red.

- System R – IBM!!! RDBMS eficiente!!!
- También surgieron primeras versiones de otros sistemas comerciales Ingres, Oracle y DEC Rdb

Algo de historia...



Se consolidan y maduran las Bases de Datos Relacionales con un liderazgo superlativo.

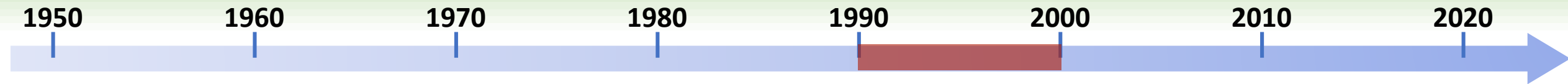
- Gran performance y facilidad de uso.
- **Propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)**
- El uso principal era el procesamiento **transaccional** (OLTP), intensivo en actualizaciones, por lo general a nivel de transacción individual.

1986: primer estándar SQL (Structured Query Language)

También en este período

- Investigación sobre bases de datos paralelas (múltiples hilos concurrentes de procesamiento) y distribuidas (datos distribuidos físicamente en diferentes nodos de almacenamiento y procesamiento)
- Inicios de investigación en bases de datos orientadas a objetos

Algo de historia...



Los RDBMS eran el estándar de hecho para las organizaciones.

1992: primera revisión importante del estándar SQL, adquiere mayor madurez.

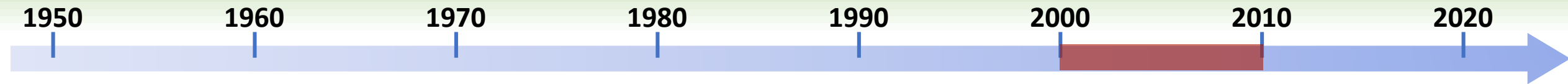
Se desarrollaron Bases de Datos orientados a objetos (OODBMS)

Existían muchas aplicaciones, datos globales contenidos en varias BBDD, necesidad de contar con información para tomar decisiones, se requiere integración de información:

- Surgen los Data Warehouses
- El uso principal es el procesamiento analítico (**OLAP**), intensivo en consultas con grandes volúmenes

En esta década crece la World Wide Web de manera explosiva, impactando en los requerimientos sobre las bases de datos: muy altas tasas de procesamiento transaccional, disponibilidad 7 x 24,

Algo de historia...



En este período aparece la necesidad de manejar tipos de **datos no estructurados**.

- Los RDBMS comienzan a soportar datos semiestructurados (XML y JSON) y datos espaciales.
- Aumenta el uso de RDBMS **Open Source**, principalmente PostgreSQL y My SQL.

Con el crecimiento de las plataformas de redes sociales, surge la necesidad de representar conexiones entre personas y sus conexiones. Esto motiva el desarrollo de **bases de datos de grafos**.

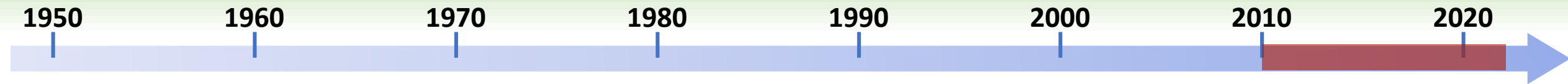
Las aplicaciones comienzan a requerir un uso más intensivo de datos de tipos diversos, así como facilidades para el desarrollo rápido.

Esto motiva la aparición de las bases de datos NoSQL (Not Only SQL), generalmente muy livianas.

- Puntos débiles en relación a las RDBMS: falta de un lenguaje de consultas declarativo estándar, y falta de consistencia estricta (proveen consistencia eventual)
- A favor: manejan nativamente tipos de datos semiestructurados o no estructurados, escalabilidad y disponibilidad.

En la segunda parte de la década, las organizaciones comenzaron a dedicar esfuerzo al análisis de datos y data mining. Surge también el framework map-reduce, orientado a facilitar el paralelismo en el procesamiento analítico de los datos.

Algo de historia...



Se profundiza el uso de aplicaciones Cloud. Profundización de almacenamiento y gestión de BBDD en la nube

Las BBDD NoSQL evolucionan para soportar nociones de consistencia más estricta, y para proveer lenguajes de consulta de alto nivel.

2016: última revisión (a la fecha) del estándar SQL. Actualmente cuenta con soporte para JSON, expresiones regulares, operaciones analíticas y estadísticas.

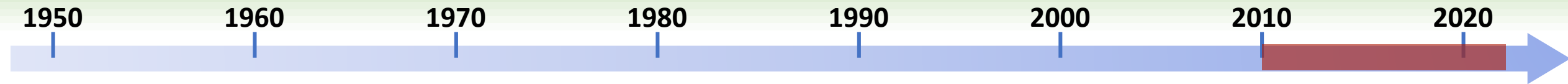
Surgen las Bases New SQL: aparecen para proveer escalabilidad, performance y cumplimiento de propiedades ACID de las transacciones.

Surgen los Data Lakes, inicialmente muchos on premise, la tendencia actual es la nube.

En relación al procesamiento analítico, aparecen

- Data Lakehouses en nube
- Bases de Datos HTAP : Para intentar resolver sobre una plataforma única el procesamiento Transaccional y Analítico

Algo de historia...



La migración a la nube no es trivial, tiene puntos positivos y desafíos por resolver.

Cuestiones: Bajo costo, alta disponibilidad, desafíos en cuanto a seguridad (sobre todo para uso gubernamental) y protección de datos personales.

La masificación de aplicaciones que hacen uso intensivo de los datos y la analítica, ponen en primer plano un conflicto no resuelto : derecho de la sociedad a saber y conocer vs derecho individual a la privacidad.

Evolución de las tecnologías de BBDD

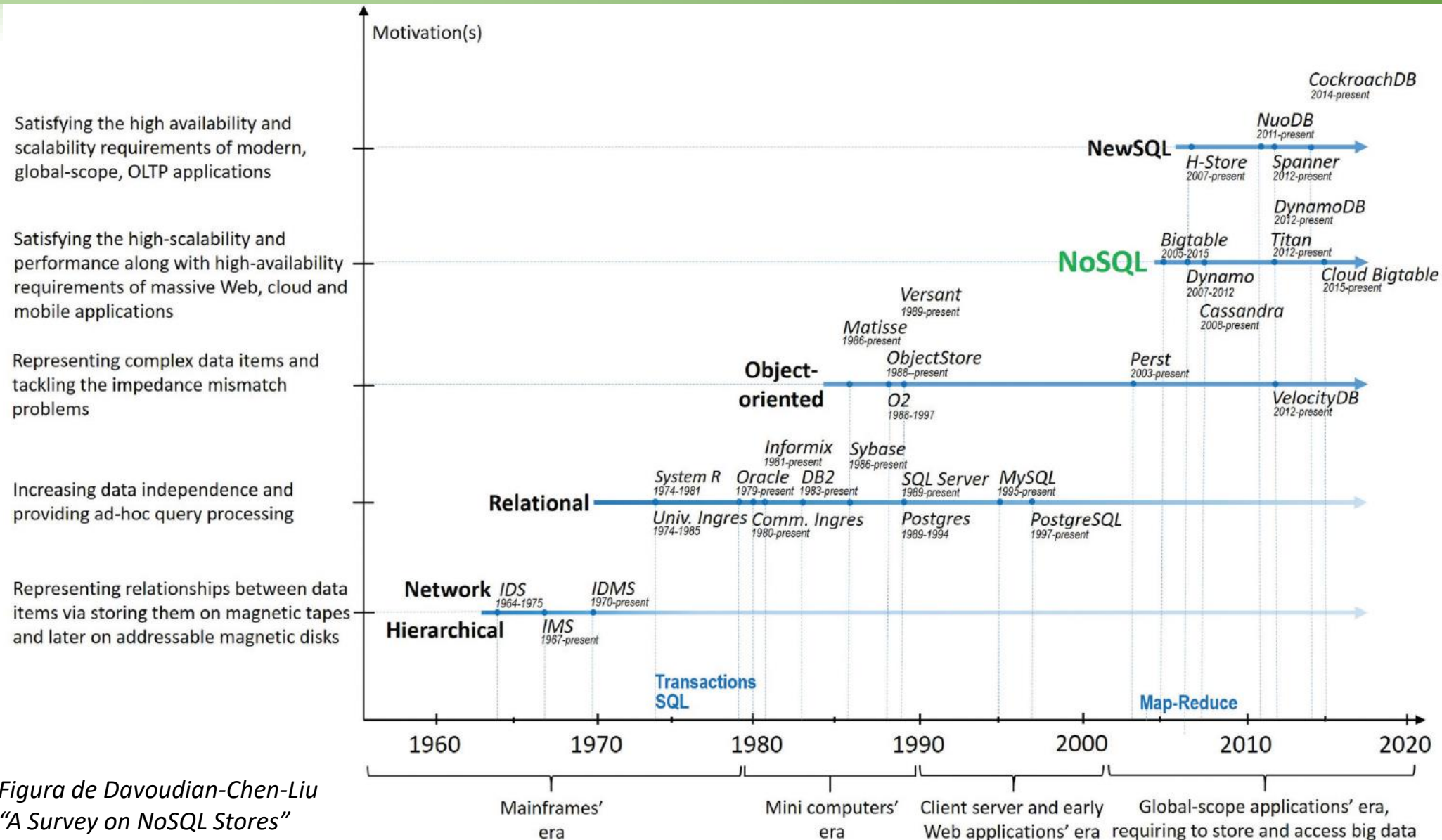


Figura de Davoudian-Chen-Liu
"A Survey on NoSQL Stores"

Ejemplos de Motores de BBDD Relacionales

Comerciales



Open Source



Ejemplos de Motores de BBDD NoSQL

Clave-Valor



Wide Column



Grafos



Documentos

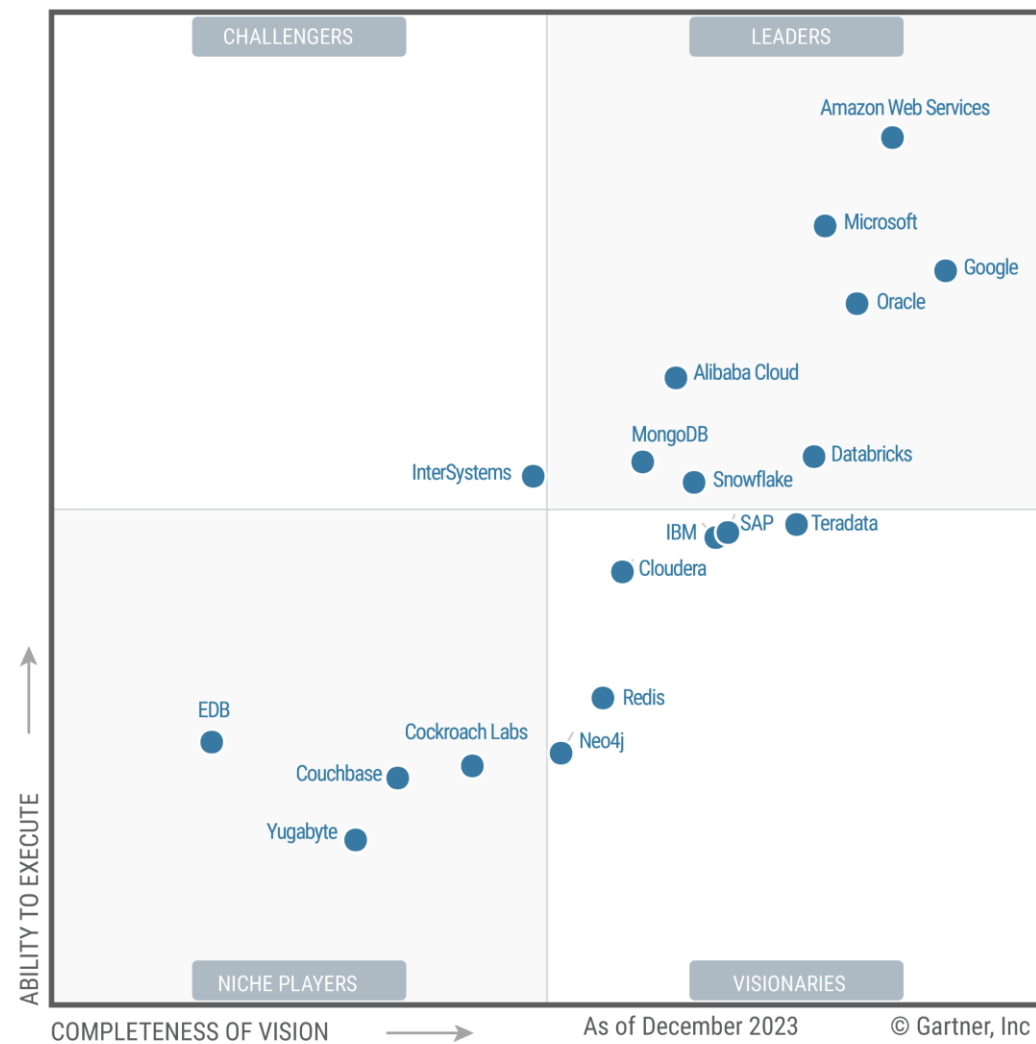


Multimodelo



Bases de Datos en la nube

Figure 1: Magic Quadrant for Cloud Database Management Systems

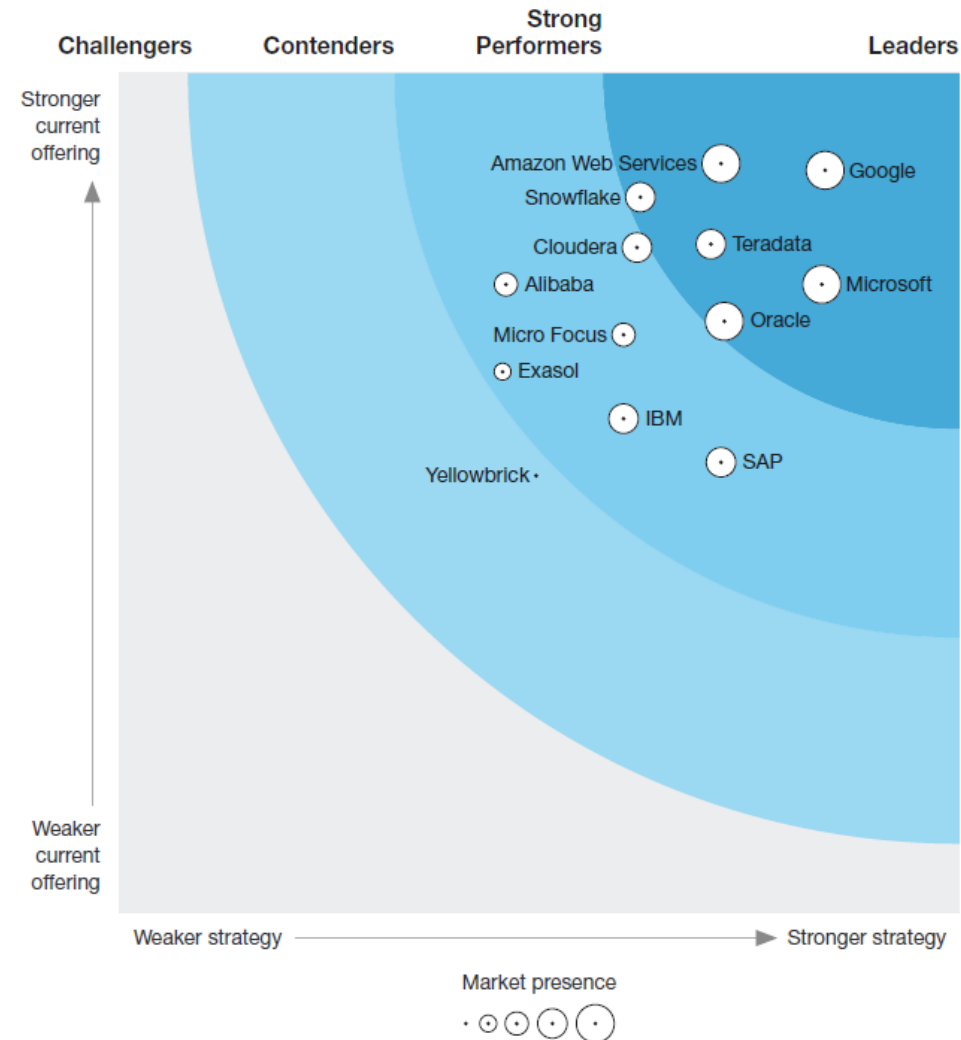


Source: Gartner

Data Warehouses en la nube

Cloud Data Warehouse

Q1 2021



Componentes Básicos de un DBMS

Aquí se presenta un esquema simplificado de un DBMS relacional que se ejecuta en un servidor centralizado.

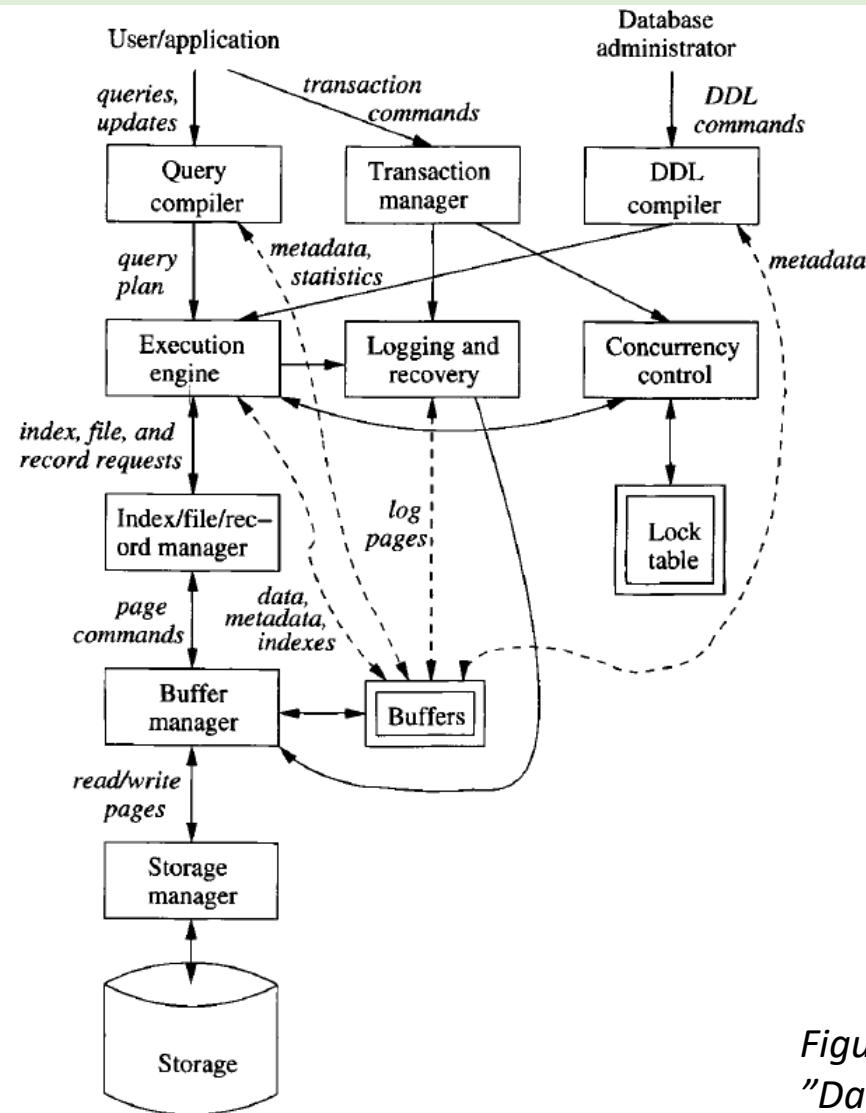


Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Figure 1.1: Database management system components

Componentes Básicos de un DBMS

Aquí se presenta un esquema simplificado de un DBMS relacional que se ejecuta en un servidor centralizado.

Veamos la interacción entre los componentes con un ejemplo del procesamiento de una consulta de recuperación de datos...
(la operación más común de uso de un motor de BBDD)

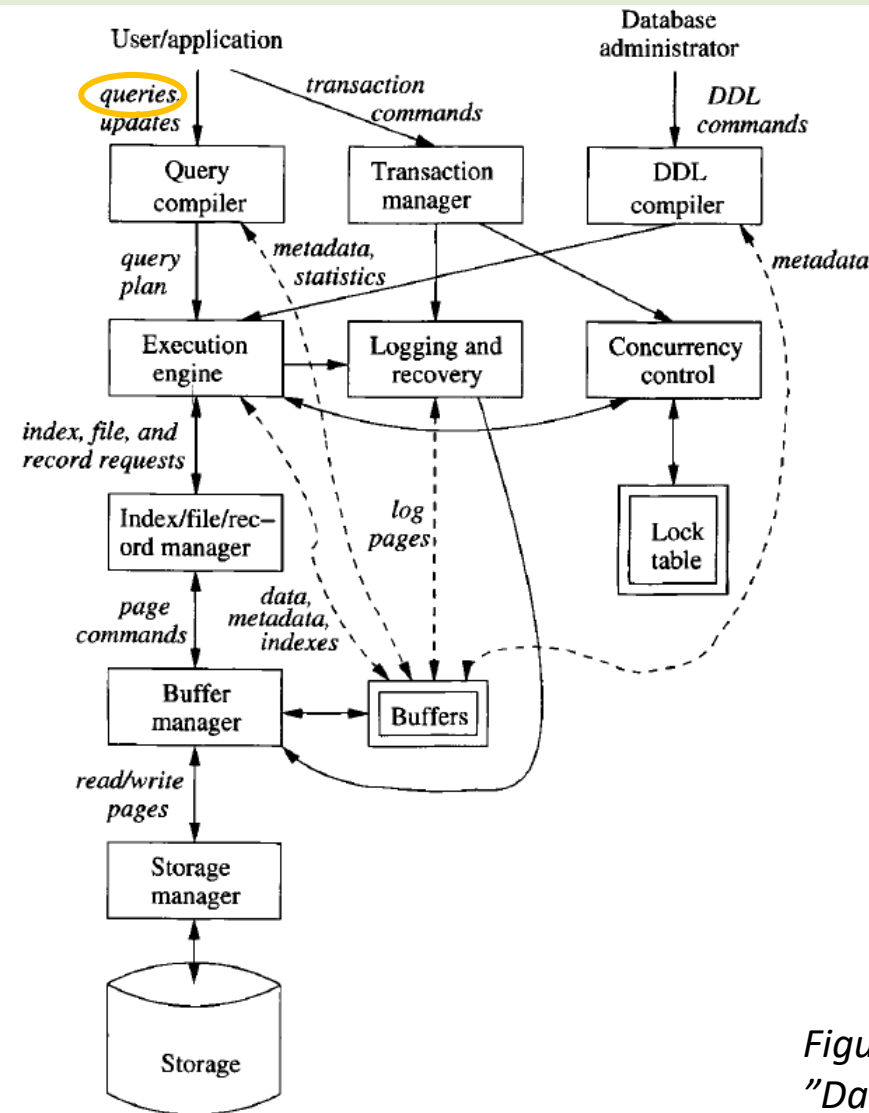


Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Figure 1.1: Database management system components

Componentes Básicos de un DBMS

- El motor recibe una **consulta** escrita en **lenguaje SQL**.

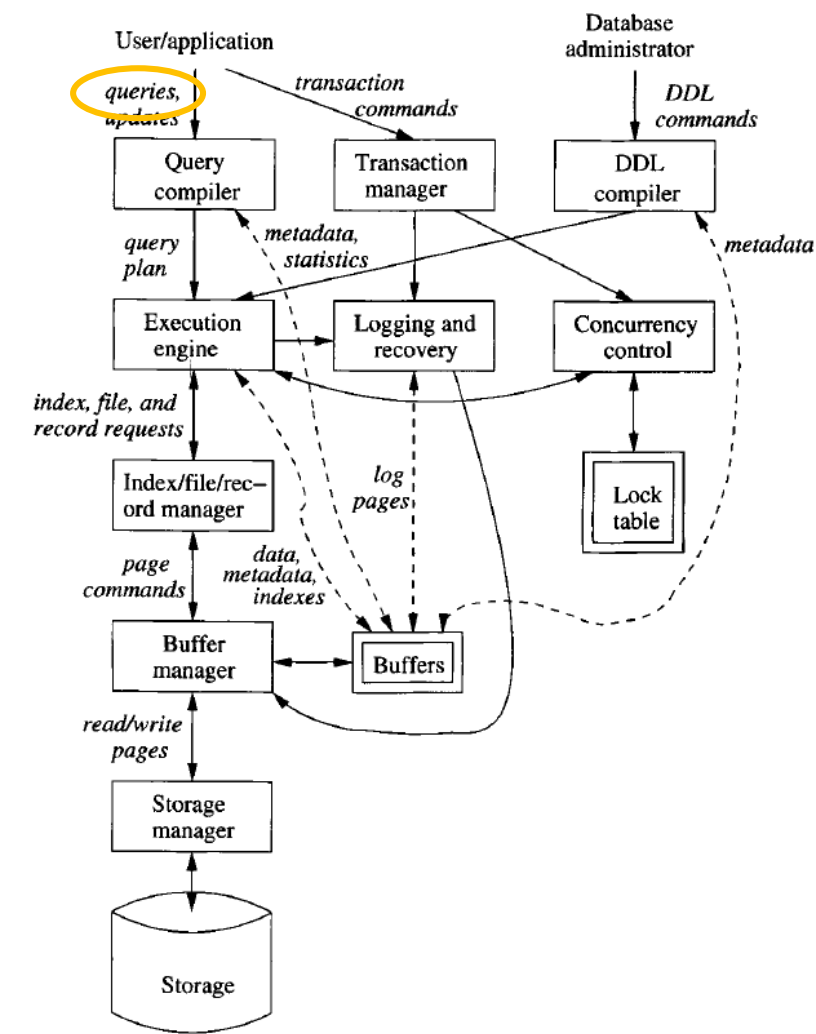
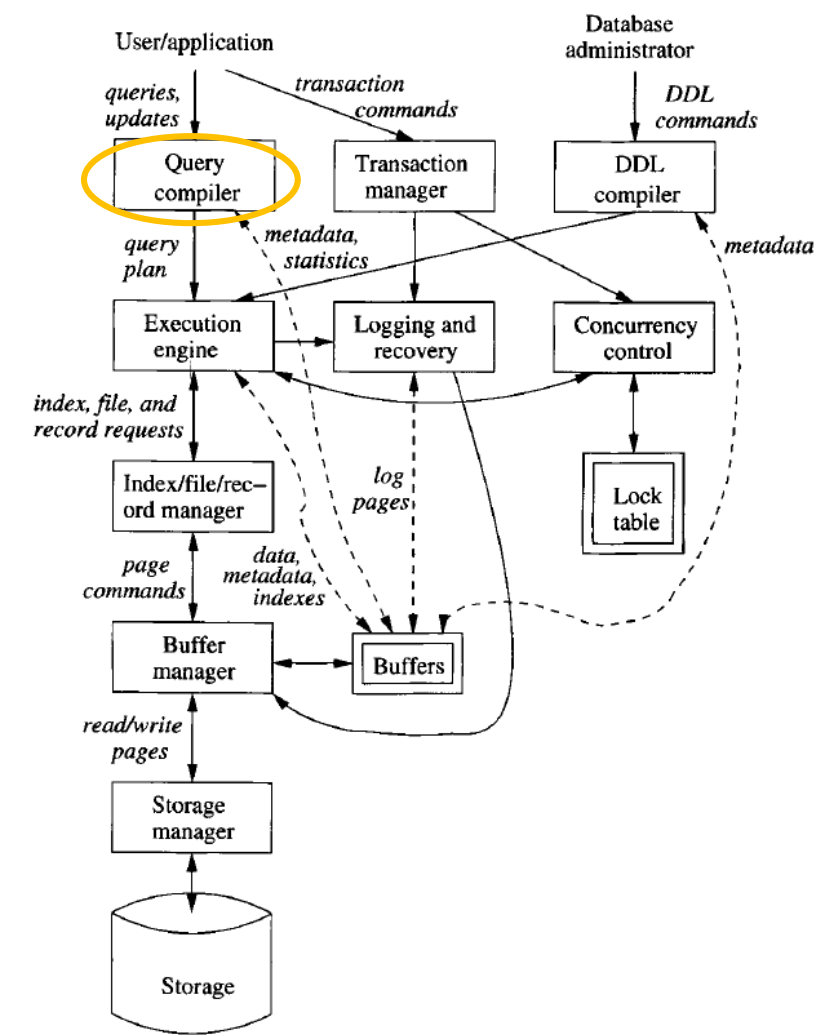


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

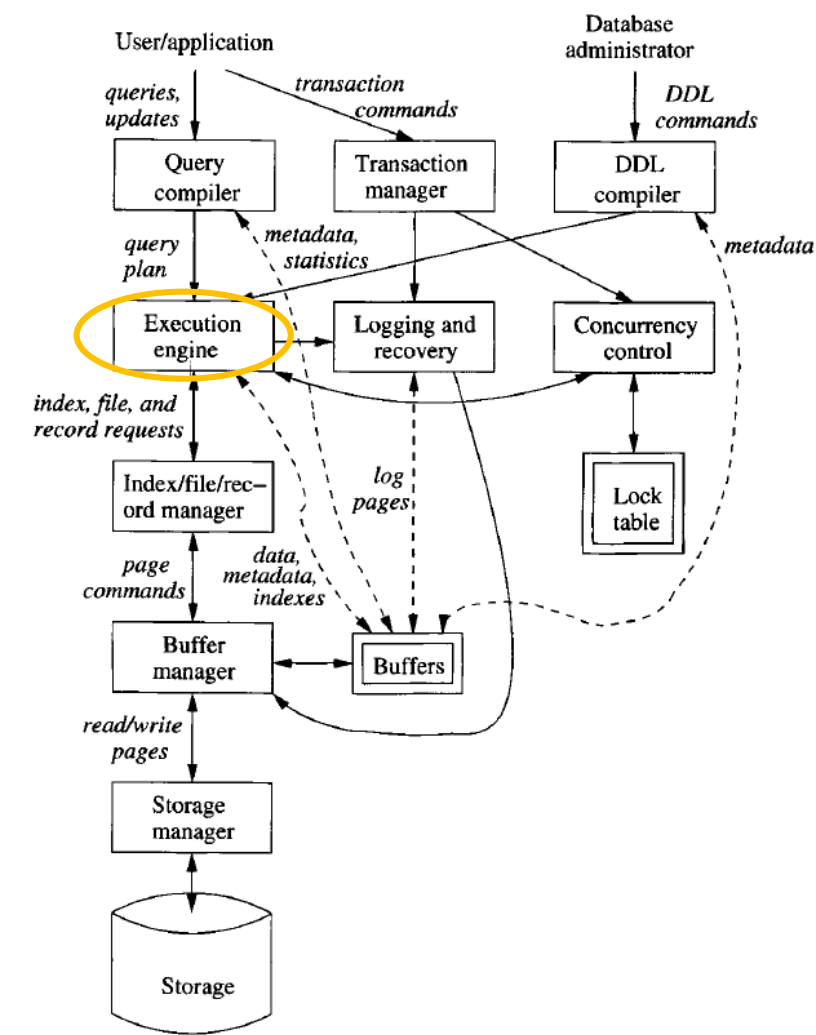


- El motor recibe una consulta escrita en lenguaje SQL.
- La consulta es parseada y optimizada por el **Compilador de Consultas**, generando como resultado un **Plan de Ejecución**.
 - ✍ *Plan de Ejecución: secuencia de acciones que debe realizar el DBMS para resolver la consulta*

Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

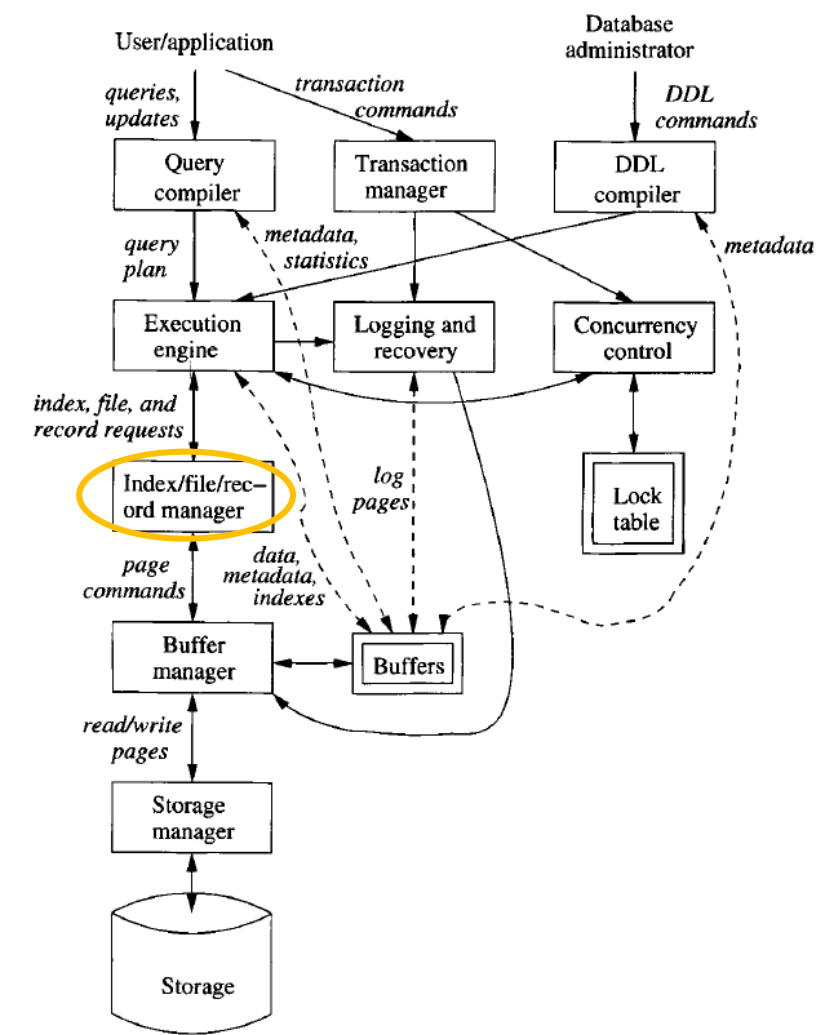


- El motor recibe una consulta escrita en lenguaje SQL.
- La consulta es parseada y optimizada por el Compilador de Consultas, generando como resultado un Plan de Ejecución.
 - Plan de Ejecución: secuencia de acciones que debe realizar el DBMS para resolver la consulta*
- El Plan de Ejecución se le pasa al **Motor de Ejecución**, el cual genera una secuencia de pedidos de datos (acorde al plan de ejecución) que será enviada al **Gestor de Recursos**.

Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

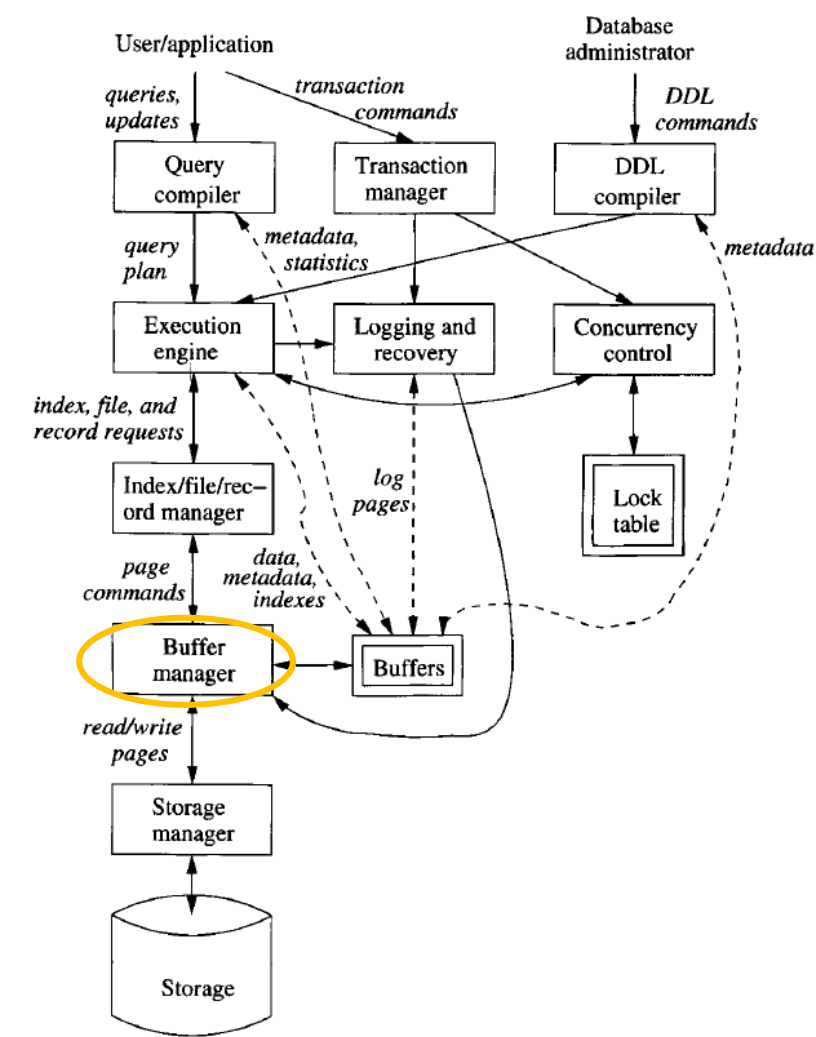


- El motor recibe una consulta escrita en lenguaje SQL.
- La consulta es parseada y optimizada por el Compilador de Consultas, generando como resultado un Plan de Ejecución.
 - ✍ *Plan de Ejecución: secuencia de acciones que debe realizar el DBMS para resolver la consulta*
- El Plan de Ejecución se le pasa al Motor de Ejecución, el cual genera una secuencia de pedidos de datos (acorde al plan de ejecución) al Gestor de Recursos.
- El **Gestor de Recursos** conoce los archivos de datos, el formato y tamaño de los registros, y los archivos de índices existentes. Con esta información, resuelve cuáles son los bloques o páginas que necesita recuperar, y se las solicita al **Gestor de Buffer**.

Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS



- El motor recibe una consulta escrita en lenguaje SQL.
- La consulta es parseada y optimizada por el Compilador de Consultas, generando como resultado un Plan de Ejecución.
 - ✍ *Plan de Ejecución: secuencia de acciones que debe realizar el DBMS para resolver la consulta*
- El Plan de Ejecución se le pasa al Motor de Ejecución, el cual genera una secuencia de pedidos de datos (acorde al plan de ejecución) al Gestor de Recursos.
- El Gestor de Recursos conoce los archivos de datos, el formato y tamaño de los registros, y los archivos de índices existentes. Con esta información, resuelve cuáles son los bloques o páginas que necesita recuperar, y se las solicita al Gestor de Buffer.
- El **Gestor de Buffer** se encarga de llevar los bloques o páginas requeridas que están en el almacenamiento secundario (disco) a buffers en memoria. Para esto se necesita comunicar con el **Gestor de Almacenamiento**.

Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

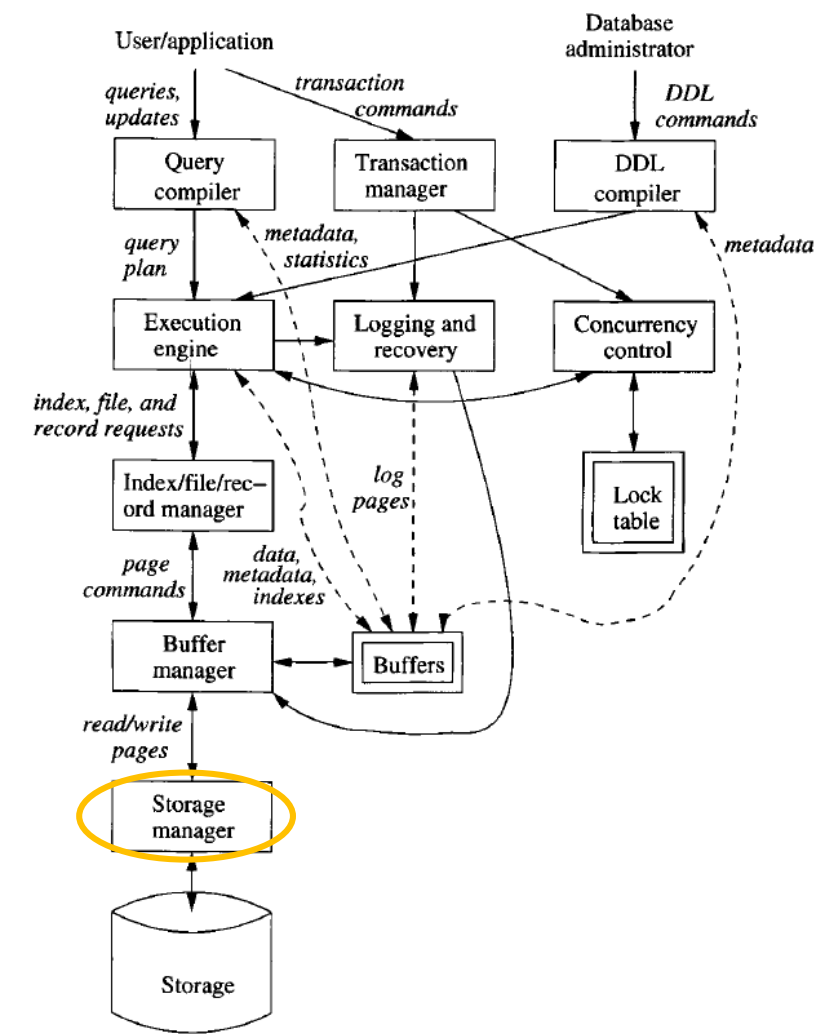


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

- El motor recibe una consulta escrita en lenguaje SQL.
- La consulta es parseada y optimizada por el Compilador de Consultas, generando como resultado un Plan de Ejecución.
 - ✍ *Plan de Ejecución: secuencia de acciones que debe realizar el DBMS para resolver la consulta*
- El Plan de Ejecución se le pasa al Motor de Ejecución, el cual genera una secuencia de pedidos de datos (acorde al plan de ejecución) al Gestor de Recursos.
- El Gestor de Recursos conoce los archivos de datos, el formato y tamaño de los registros, y los archivos de índices existentes. Con esta información, resuelve cuáles son los bloques o páginas que necesita recuperar, y se las solicita al Gestor de Buffer.
- El Gestor de Buffer se encarga de llevar los bloques o páginas requeridas que están en el almacenamiento secundario (disco) a buffers en memoria. Para esto se necesita comunicar con el Gestor de Almacenamiento
- El **Gestor de Almacenamiento** es quien recupera del disco, ya sea invocando comandos del Sistema Operativo, o ejecutando comandos directamente sobre el controlador de disco

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Compilador de Consultas

- Traduce la consulta externa en un formato interno llamado Plan de Ejecución (secuencia de acciones que debe realizar el DBMS para resolver la consulta).

Consiste de 3 principales unidades:

- **Parser de Consultas:** construye una estructura de árbol que representa la consulta recibida de forma textual
- **Preprocesador de Consultas:** realiza chequeos semánticos y realiza transformaciones al árbol para convertirlo en un árbol de operaciones algebraicas que representan la plan de ejecución inicial
- **Optimizador de Consultas:** transforma el plan de ejecución inicial en una secuencia optimizada de operaciones sobre los datos actuales.

✍ *El Compilador de Consultas utiliza metadatos y estadísticas de la Base de Datos para resolver cuál secuencia de operaciones sería más eficiente.*

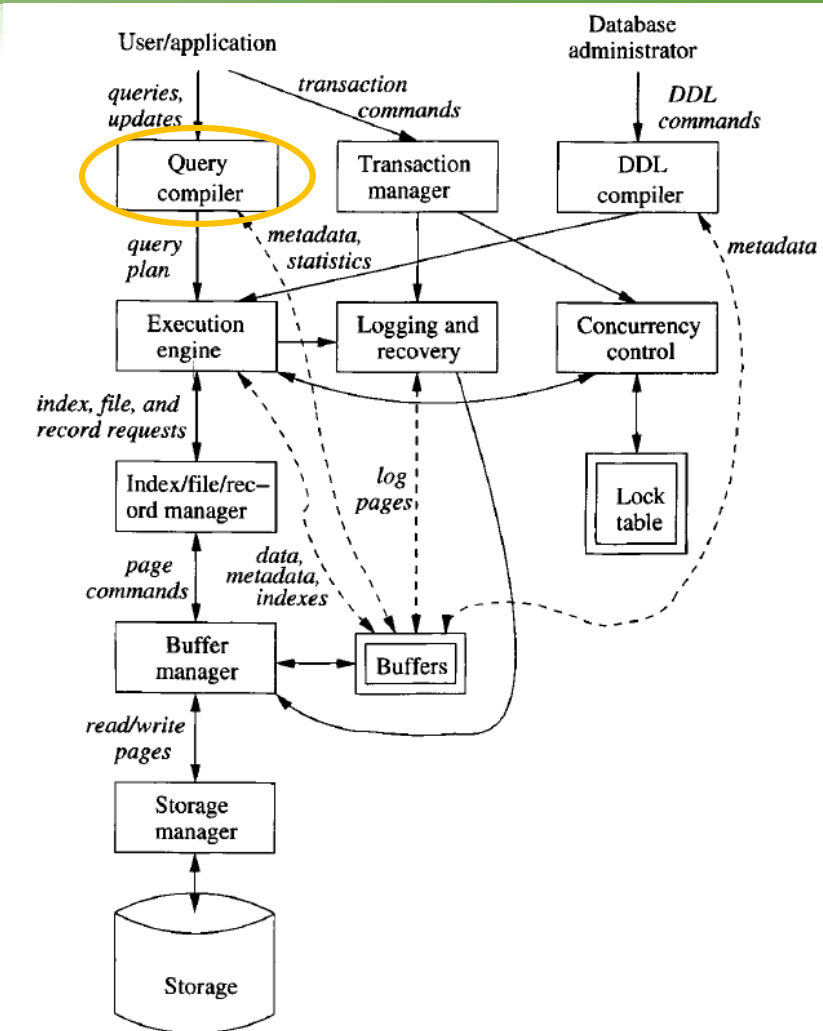


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Motor de Ejecución

- Tiene la responsabilidad de ejecutar cada uno de los pasos del plan de ejecución elegido.
- El motor de ejecución interactúa con la mayoría de los otros componentes del DBMS, ya sea directamente o a través de los buffers.
- Debe obtener los datos de la base de datos en los buffers para manipular esos datos.
- Necesita interactuar con el planificador para evitar acceder a datos que están bloqueados, y con el gestor de logs para asegurarse de que todos los cambios en la base de datos estén correctamente registrados.

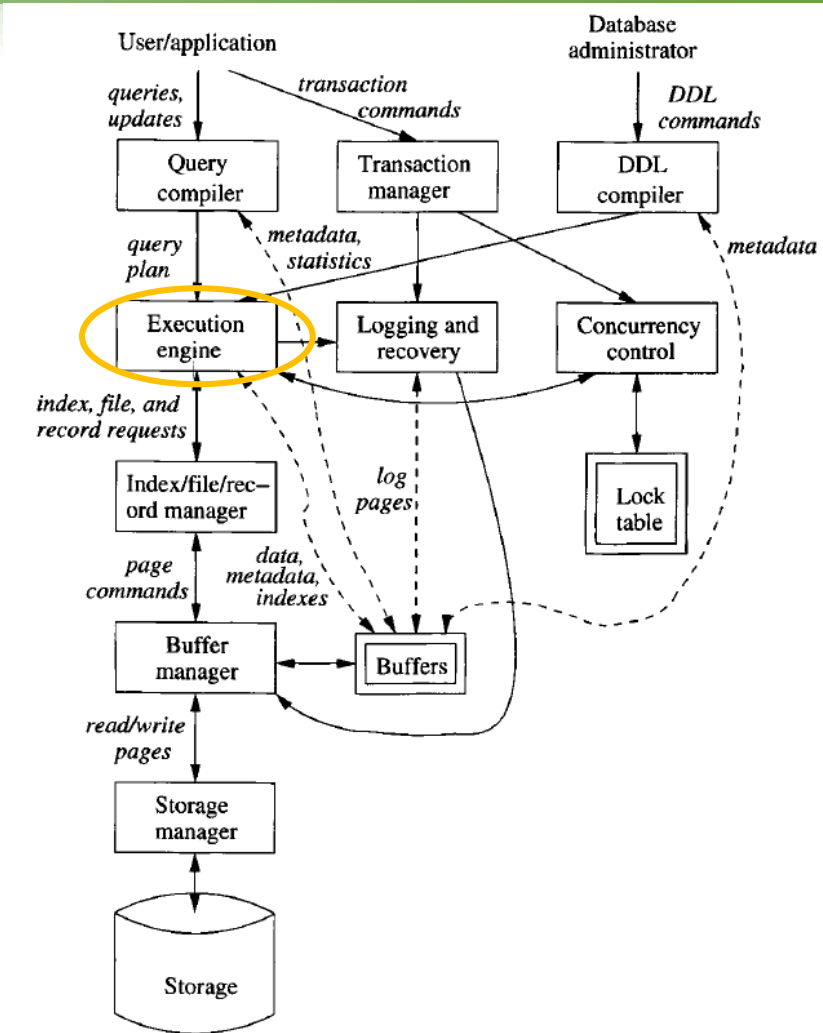


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Gestor de Almacenamiento

- Los datos de una base de datos normalmente residen en almacenamiento secundario (generalmente disco). Sin embargo, para realizar cualquier operación útil en los datos, esos datos deben estar en la memoria principal.
- Es tarea del gestor de almacenamiento controlar la ubicación de los datos en el disco y su movimiento entre el disco y la memoria principal.
- El gestor de almacenamiento puede acceder a través del file system del Sistema Operativo, aunque muchas veces controla el almacenamiento en el disco directamente. Realiza un seguimiento de la ubicación de los archivos en el disco y obtiene el bloque o bloques que contienen un archivo a solicitud del gestor de búferes.

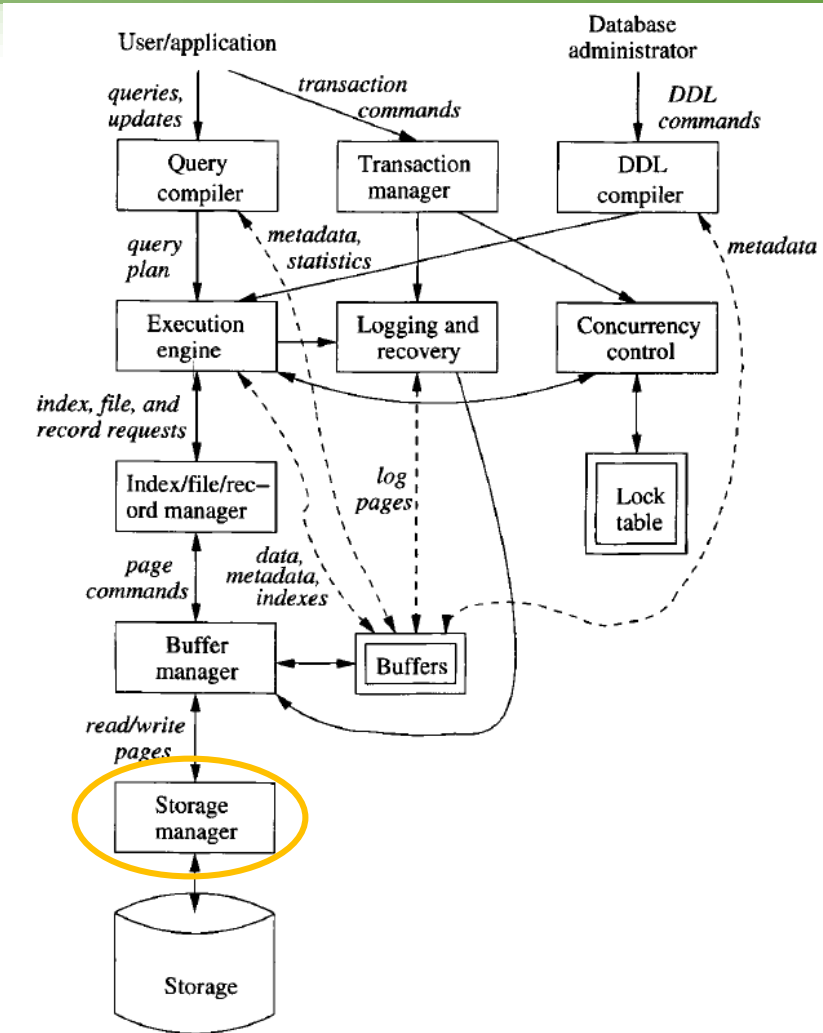


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Gestor de Buffer

Es responsable de dividir la memoria principal disponible en búferes, que son regiones del tamaño de una página en las cuales se pueden transferir bloques de disco.

Todos los componentes del DBMS que necesiten información del disco interactuarán con los búferes y el gestor de búferes, ya sea directamente o a través del motor de ejecución.

Los tipos de información que varios componentes pueden necesitar incluyen:

- Datos: el contenido de la base de datos en sí.
- Metadatos: el esquema de la base de datos que describe la estructura y restricciones de la base de datos.
- Logs: información sobre cambios recientes en la base de datos; estos respaldan la durabilidad de la base de datos.
- Estadísticas: información recopilada y almacenada por el DBMS sobre propiedades de los datos, como los tamaños y valores de varias relaciones u otros componentes de la base de datos.
- Índices: estructuras de datos para facilitar acceso eficiente a los datos.

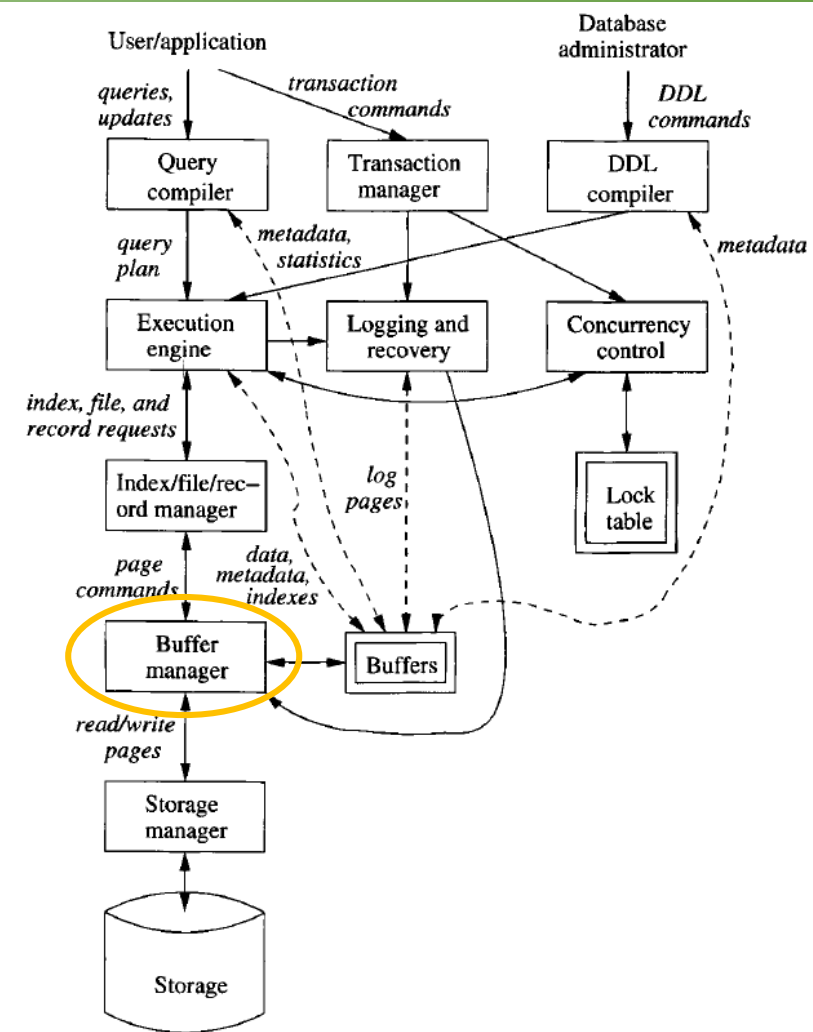


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Gestor de Transacciones

✍ *Transacción: Unidad de trabajo que debe ejecutarse atómicamente y aislada del resto de las transacciones.*

Es común que varias operaciones de BBDD se agrupen para ejecutarse transaccionalmente.

El Gestor de Transacciones es responsable de la gestión de la ejecución apropiada de transacciones, garantizando adicionalmente la durabilidad de los resultados de las transacciones completadas.

Realiza las siguientes tareas:

- Logging (interactuando con el Gestor de Logs y Recuperación, lo veremos a continuación)
- Control de concurrencia (interactuando con el planificador, o Gestor de Control de Concurrencia, lo veremos a continuación)
- Resolución de deadlocks: en caso de ocurrir, el Gestor de Transacciones tiene la responsabilidad de intervenir y cancelar (haciendo “rollback” o cancelando) una o más transacciones para permitir que las otras continúen.

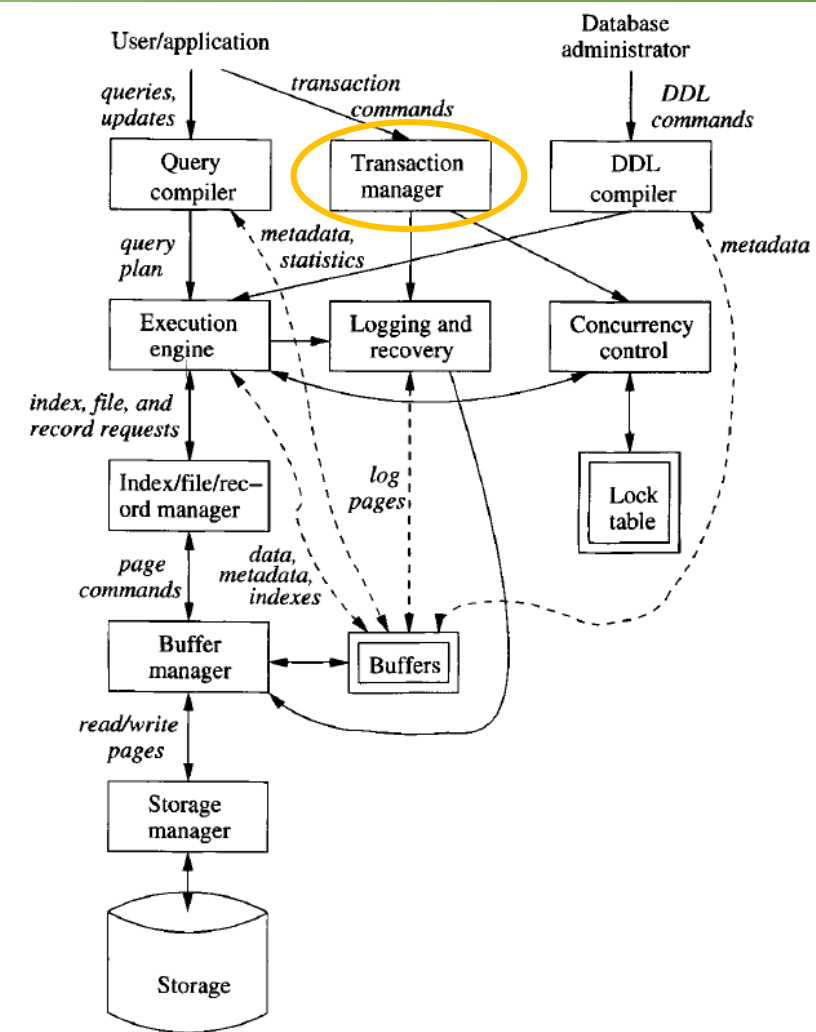


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Gestor de Logs y Recuperación

- Para garantizar la durabilidad, cada cambio en la base de datos se loguea por separado en disco.
- El gestor de logs sigue una de varias políticas diseñadas para asegurar que, sin importar cuándo ocurra una falla del sistema o "crash", un gestor de recuperación podrá examinar el registro de cambios y restaurar la base de datos a un estado consistente.
- El gestor de logs inicialmente escribe el registro en búferes y negocia con el gestor de búferes para asegurarse de que los búferes se escriban en disco (donde los datos pueden sobrevivir a un fallo) en momentos apropiados.

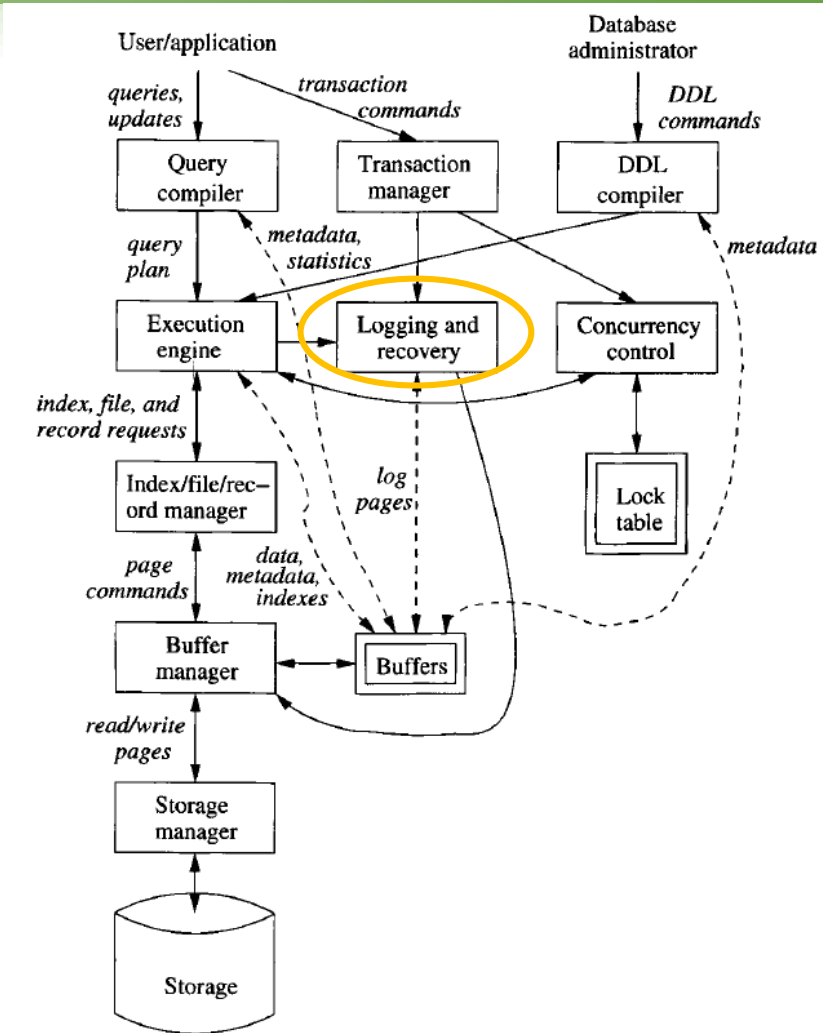


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes Básicos de un DBMS

Veamos algunos de los componentes con más detalle..

Gestor de Control de Concurrency

- En la mayoría de los sistemas, en realidad habrá muchas transacciones ejecutándose concurrentemente.
- Para garantizar el aislamiento de ejecuciones, el planificador (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 hubieran ejecutado en su totalidad, una a la vez.
- El planificador puede necesitar bloquear ciertas partes de la base de datos, de manera de evitar que dos transacciones accedan a la misma pieza de datos de maneras que interactúen mal.
- Los bloqueos generalmente se almacenan en una tabla de bloqueo en la memoria principal.

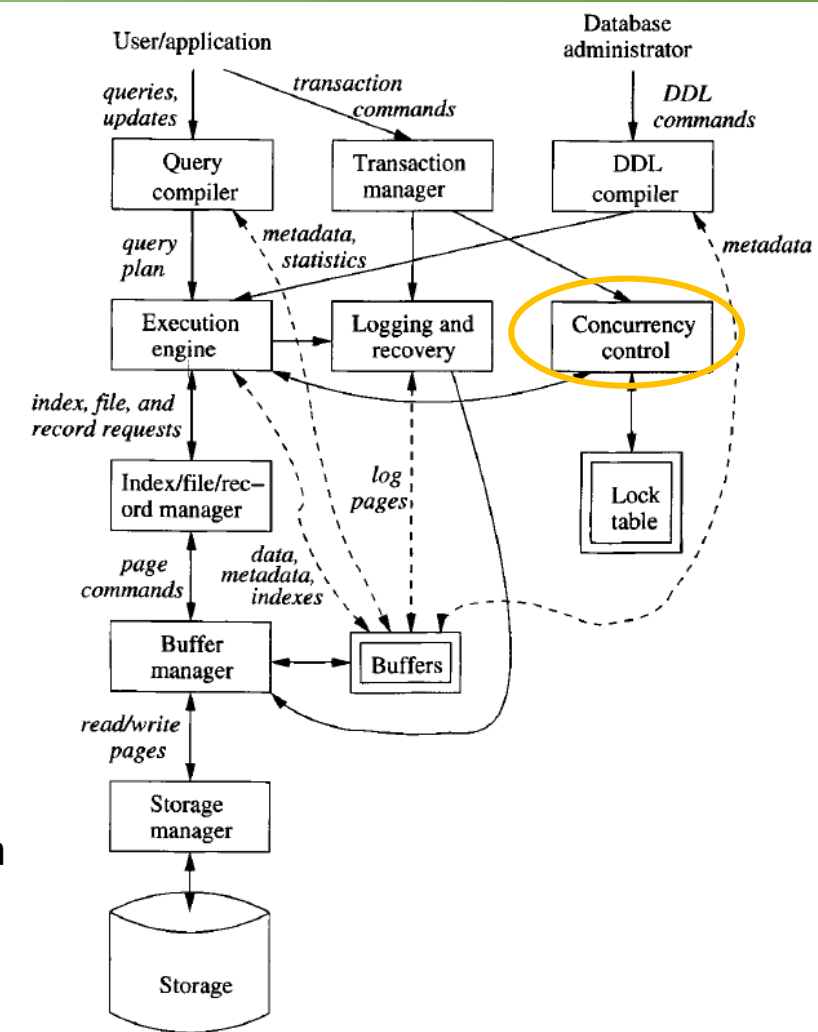


Figure 1.1: Database management system components

Figura de García Molina- Ullman-Widom
"Database Systems: The Complete Book"

Componentes de un DBMS

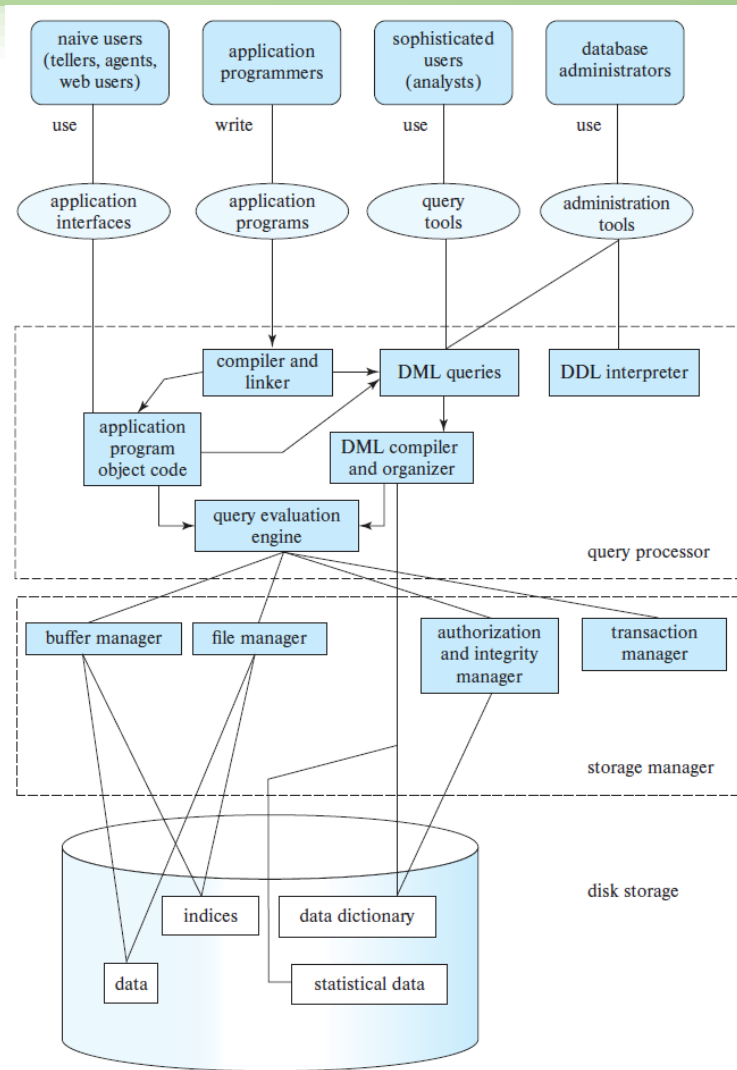


Figure 1.3 System structure.

Figura de Silberschatz-Sudarshan
"Database System Concepts"

En este otro esquema simplificado aparece otro componente importante, que no había quedado explícito:

Gestor de Autorizaciones e Integridad

Es responsable de comprobar las restricciones de integridad de la BBDD, así como de verificar la autoridad de los usuarios para acceder a los datos solicitados o realizar las operaciones requeridas.

¿Y de todo esto, qué vamos a ver en la materia?

RDBMS y Modelo Relacional

Diseño de aplicaciones de BBDD



- ☐ Modelado Conceptual/Modelo Entidad Relación T P
- ☐ Modelo Relacional/Diseño Lógico Relacional T P
- ☐ Dependencias Funcionales y Normalización T P

Lenguajes de Consulta



- ☐ Algebra Relacional T P
- ☐ CRT T P
- ☐ SQL L

Funciones Clave de los RDBMS



- ☐ Índices T P L
- ☐ Optimización de Consultas (*) T L
- ☐ Transacciones y concurrencia T P
- ☐ Recuperación ante fallas T P

Paradigma NoSQL y Otros tipos de BBDD

BBDD NoSQL



- ☐ Documentos T P
- ☐ Clave Valor T P
- ☐ Wide Column T P
- ☐ Grafos T

Otros tipos de Bases de Datos



- ☐ XML
 - ☐ Modelo Objeto Relacional
 - ☐ NewSQL
 - ☐ Otras
- T

Otros Conceptos Generales

Gestión de Datos



- ☐ Gobierno de Datos
- ☐ Seguridad
- ☐ Data Driven
- ☐ Datos Abiertos

Otros Conceptos



- ☐ Arquitecturas
 - ☐ OLTP/OLAP/DWH
 - ☐ Big Data
 - ☐ Data Mining
- T
- T

Bibliografía

- *García Molina H., Ullman J. & Widom J. (2009) Database Systems: The Complete Book, (2da. Ed.), Pearson, Cap. 1*
- *Silberschatz A., Korth H. & Sudarshan S.(2020) Database System Concepts (7ma. Ed.), Mc.Graw Hill, Cap. 1*
- *Elmasri R. & Navathe S. (2016) Fundamentals of Database Systems (7ma. Ed.) Pearson, Cap. 2*

Dudas?



Muchas gracias!