



Base de Datos 1

Unidad 1

Introducción – Conceptos Generales.

Profesor

- Jorge Insfran
 - jorge.insfran@docentes.unpaz.edu.ar

Objetivos

- Adquirir los conceptos básicos de la teoría de los sistemas de bases de datos. Diferenciar un sistema de Base de Datos de un Sistema de Archivos.
- Realizar el diseño conceptual de un Sistema de Información.
- Diseñar y modelar sistemas de bases de datos relacionales.
- Realizar consultas mediante la utilización de Algebra Relacional.
- Implementar consultas en SQL.

Organización de la materia

- 5 Unidades
- Dos exámenes parciales.
- 5 Trabajos por plataforma
- Trabajo Práctico

Material Bibliográfico

- Fundamentos de bases de datos. Quinta Edición. Abraham Silberschatz, Henry F. Korth y S. Sudarshan. McGraw Hill 2006
- Introducción a los sistemas de bases de datos. Séptima edición. C.J. Date. Prentice Hall (2001)
- Apuntes de la materia

Bases de Datos

- ¿Alguna vez nos encontramos con una?
- ¿Que contacto tuvimos?
- ¿Trabajamos con alguna cotidianamente?





Un poco de historia

- Antes de los 70:
 - Aplicaciones batch. Monotarea
 - Archivos de movimientos en cintas magnéticas (acceso secuencial)
 - Información redundante (para minimizar accesos)



Un poco de historia

- Con los discos y las redes, los archivos comenzaron a consultarse de manera on-line por distintos usuarios.
- Luego surgió la necesidad de poder actualizar también on-line
- Integración de aplicaciones => Integración de archivos => Eliminar redundancia
- Data Banks: primeros archivos interrelacionados, con estructuras complejas y compartidos (a inicios de los 70 se los comenzó a llamar Data Bases)

Un poco de historia

- El software de gestión de archivos era demasiado elemental (no contemplaba relaciones, actualizaciones simultáneas)
- Mediados de los 70: surgen los Data Base Management Systems (SGBD)



Definición

- Una base de datos de un Sistema de Información es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del Sistema de Información y de sus interrelaciones.
- Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.
- En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones



Base de Datos

- Colección de datos interrelacionados archivados juntos sin redundancias perjudiciales o innecesarias. Su finalidad es la de servir a una o más aplicaciones de la mejor manera posible. Los datos están almacenados de modo que resulten independiente de los programas que los usan. Se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.



Base de Datos

- Conjunto de datos operativos almacenados a los que acceden los sistemas de aplicación de una empresa.



Archivos vs. Bases de Datos

	Archivos	Bases de Datos
Entidades Tipo	tienen registros de una sola entidad tipo	tienen datos de varias entidades tipo
Interrelaciones	el sistema no interrelaciona archivos	el sistema tiene previstas herramientas para interrelacionar entidades
Redundancia	se crean archivos a la medida de cada aplicación, con todos los datos necesarios, aunque algunos sean redundantes respecto de otros archivos	todas las aplicaciones trabajan con la misma Base de Datos y la integración de los datos es básica, de modo que se evita la redundancia
Usuarios	sirven para un solo usuario o una sola aplicación. Dan una sola visión del mundo real	es compartida por muchos usuarios de distintos tipos. Ofrece varias visiones del mundo real

Evolución de los SGBD

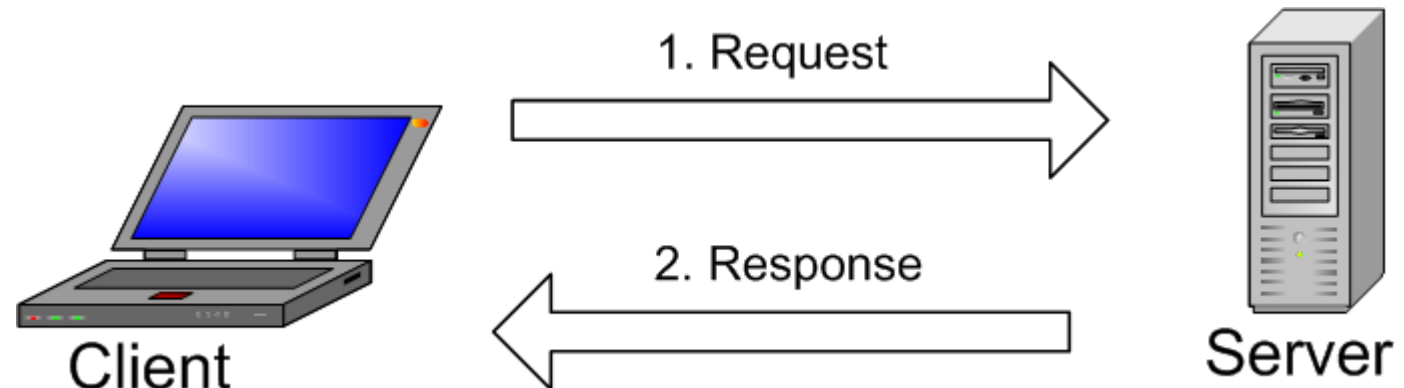
- 60-70: Sistemas Centralizados (Mainframe)
- Terminales con teclado, conectadas mediante línea telefónica
- SGBD ligados a software de comunicaciones y gestión de transacciones
- Programación en Cobol o PL/I, con subrutinas especializadas y complejas para tratar las BD (requerían conocimiento físico)
- Años 80: SGBD relacionales
- La aparición de los SGBD relacionales supone un avance importante para facilitar la programación de aplicaciones con BD y para conseguir que los programas sean independientes de los aspectos físicos de la BD
- 86: Se estandariza el SQL

Evolución de los SGBD

- Años 90: Distribución, C/S y 4GL
- Numerosas bases (departamentales) y varios SGBD en la empresa
- La necesidad de tener una visión global de la empresa y de interrelacionar diferentes aplicaciones que utilizan BD diferentes, junto con la facilidad que dan las redes para la intercomunicación entre computadoras, ha conducido a los SGBD actuales, que permiten que un programa pueda trabajar con diferentes BD como si se tratase de una sola. Es lo que se conoce como base de datos distribuida.

C/S

- La tecnología que se utiliza habitualmente para distribuir datos es la que se conoce como entorno (o arquitectura) cliente/servidor (C/S). Todos los SGBD relacionales del mercado han sido adaptados a este entorno.
- La idea del C/S es sencilla. Dos procesos diferentes, que se ejecutan en un mismo sistema o en sistemas separados, actúan de forma que uno tiene el papel de cliente o peticionario de un servicio, y el otro el de servidor o proveedor del servicio.





Tendencias actuales

- Hoy día, los SGBD relacionales están en plena transformación para adaptarse a tres tecnologías de éxito reciente, fuertemente relacionadas: la multimedia, la de orientación a objetos (OO) e Internet y la web.
- Manejos de grandes volúmenes de información sin estructura (Big Data)

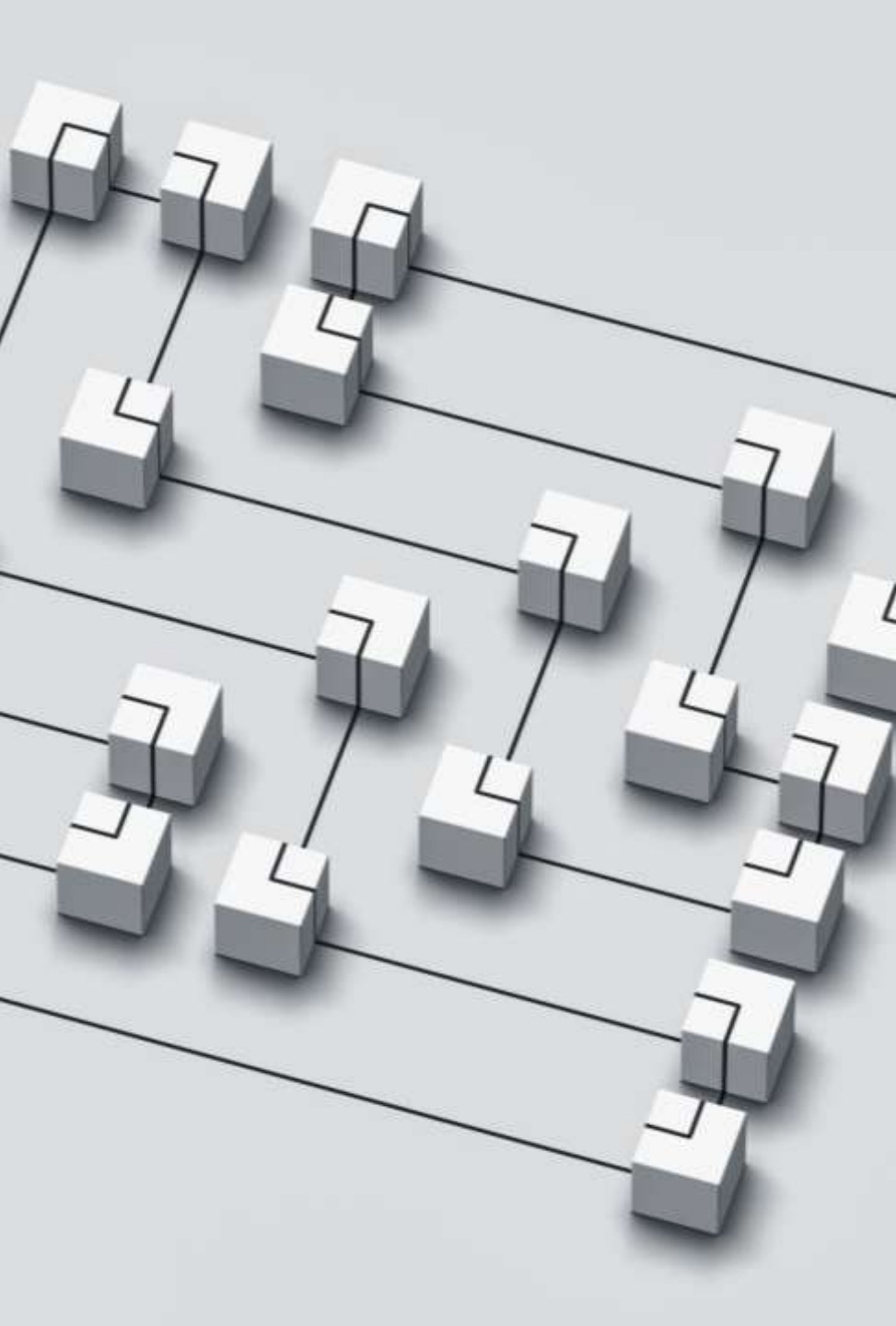
Objetivos y servicios de los SGBD

- Consultas no predefinidas y complejas (Ad hoc)
- Flexibilidad e Independencia (abstraerse de la implementación física)
- Problemas de redundancia
 - Inconsistencia
 - El SGBD debe permitir que el diseñador defina datos redundantes, pero entonces tendría que ser el mismo SGBD el que hiciese automáticamente la actualización de los datos en todos los lugares donde estuviesen repetidos.
- Integridad de los datos
 - Integridad de entidad / Integridad referencial / Reglas
- Concurrencia de usuarios
 - Transacciones *
- Seguridad



Concurrencia de usuarios

- Un objetivo fundamental de los SGBD es permitir que varios usuarios puedan acceder concurrentemente a la misma Base de Datos.



Transacción

- Denominamos transacción de Base de Datos o, simplemente, transacción un conjunto de operaciones simples que se ejecutan como una unidad.
- Los SGBD deben conseguir que el conjunto de operaciones de una transacción nunca se ejecute parcialmente.
- O se ejecutan todas, o no se ejecuta ninguna.

Propiedades ACID

- **Atomicidad.**
- **Consistencia.**
- **Aislamiento. (Isolation)**
- **Durabilidad.**

Arquitectura de los SGBD

- Esquemas y Niveles

- Los SGBD necesitan que les demos una descripción o definición de la BD. Esta descripción recibe el nombre de esquema de la BD, y los SGBD la tendrán continuamente a su alcance.
- Nivel Lógico: oculta los detalles de cómo se almacenan los datos, cómo se mantienen y cómo se accede físicamente a ellos. En este nivel sólo se habla de entidades, atributos y reglas de integridad
- Nivel Físico: índices, en que discos se almacenan los datos, tamaño de página, etc.

ANSI/SPARC

- De acuerdo con la arquitectura ANSI/SPARC, debía haber tres niveles de esquemas (tres niveles de abstracción). La idea básica de ANSI/SPARC consistía en descomponer el nivel lógico en dos: el nivel externo y el nivel conceptual. Denominábamos nivel interno lo que aquí hemos denominado nivel físico
 - Nivel externo se sitúan las diferentes visiones lógicas que los procesos usuarios (programas de aplicación y usuarios directos) tendrán de las partes de la Base de Datos que utilizarán. Estas visiones se denominan esquemas externos.
 - Nivel conceptual hay una sola descripción lógica básica, única y global, que denominamos esquema conceptual, y que sirve de referencia para el resto de los esquemas.
 - Nivel físico hay una sola descripción física, que denominamos esquema interno.

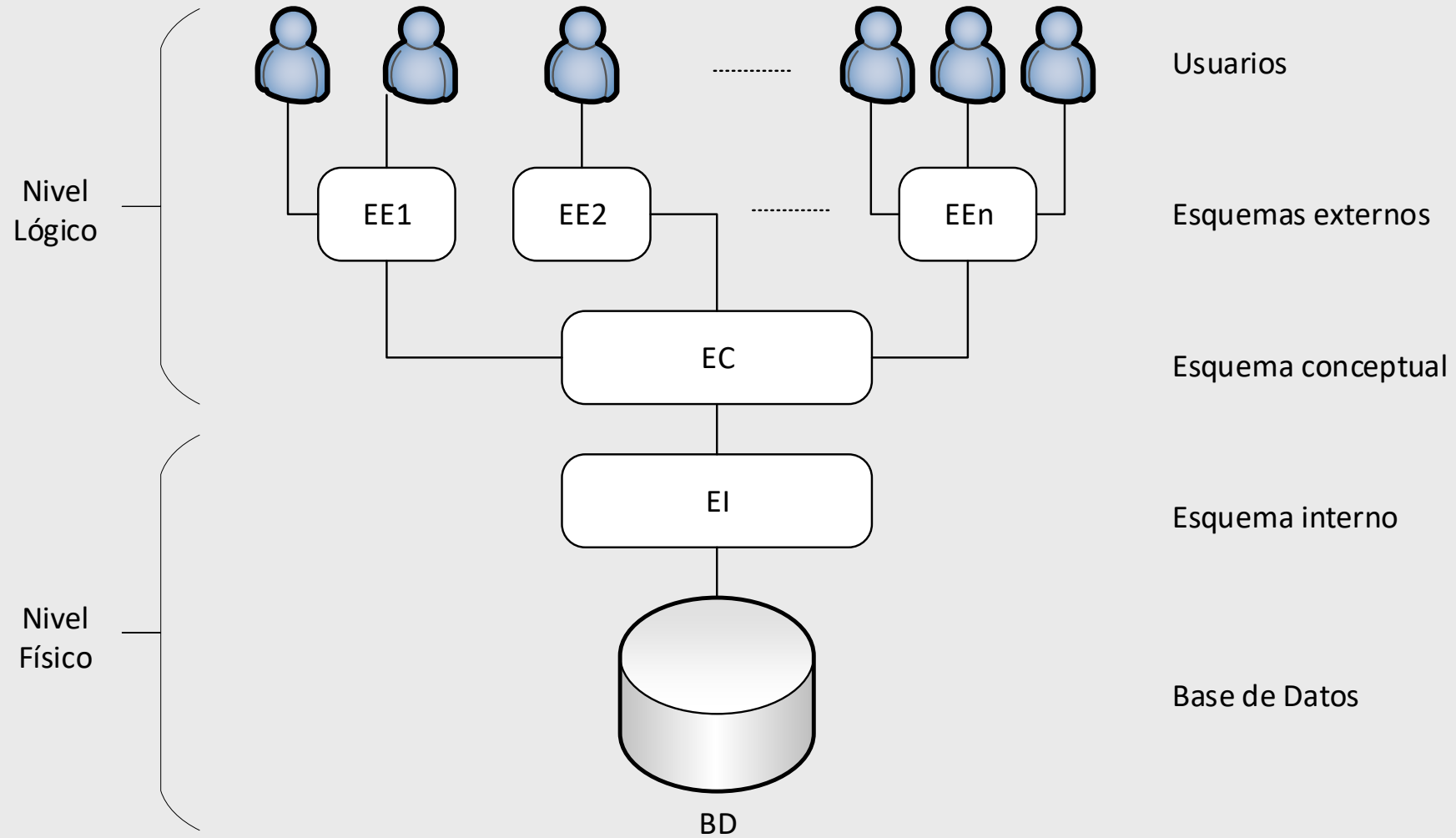
ANSI/SPARC

- Al definir un esquema externo, se citarán sólo aquellos atributos y aquellas entidades que interesen; los podremos red denominar, podremos definir datos derivados o redefinir una entidad para que las aplicaciones que utilizan este esquema externo creen que son dos, definir combinaciones de entidades para que parezcan una sola, etc.
- En el esquema conceptual se describirán las entidades tipo, sus atributos, las interrelaciones y las restricciones o reglas de integridad.
- El esquema interno o físico contendrá la descripción de la organización física de la Base de Datos: caminos de acceso (índices, hashing, punteros, etc.), codificación de los datos, gestión del espacio, tamaño de la página, etc.
- Para crear una Base de Datos hace falta definir previamente su esquema conceptual, definir como mínimo un esquema externo y, de forma eventual, definir su esquema interno. Si este último esquema no se define, el mismo SGBD tendrá que decidir los detalles de la organización física. El SGBD se encargará de hacer las correspondencias (mappings) entre los tres niveles de esquemas

ESQUEMAS Y NIVELES EN LOS SGBD RELACIONALES

- Se utiliza una terminología ligeramente diferente. No se separan de forma clara tres niveles de descripción. Se habla de un solo esquema –schema–, pero en su interior se incluyen descripciones de los tres niveles. En el schema se describen los elementos de aquello que en la arquitectura ANSI/SPARC se denomina esquema conceptual (entidades tipo, atributos y restricciones) y las vistas –view–, que corresponden aproximadamente a los esquemas externos.
- El estándar ANSI-ISO de SQL no habla en absoluto del mundo físico o interno; lo deja en manos de los SGBD relacionales del mercado. Sin embargo, estos SGBD proporcionan la posibilidad de incluir dentro del schema descripciones de estructuras y características físicas (índice, tablespace, cluster, espacios para excesos, etc.)

Esquemas y niveles



Independencia de los Datos

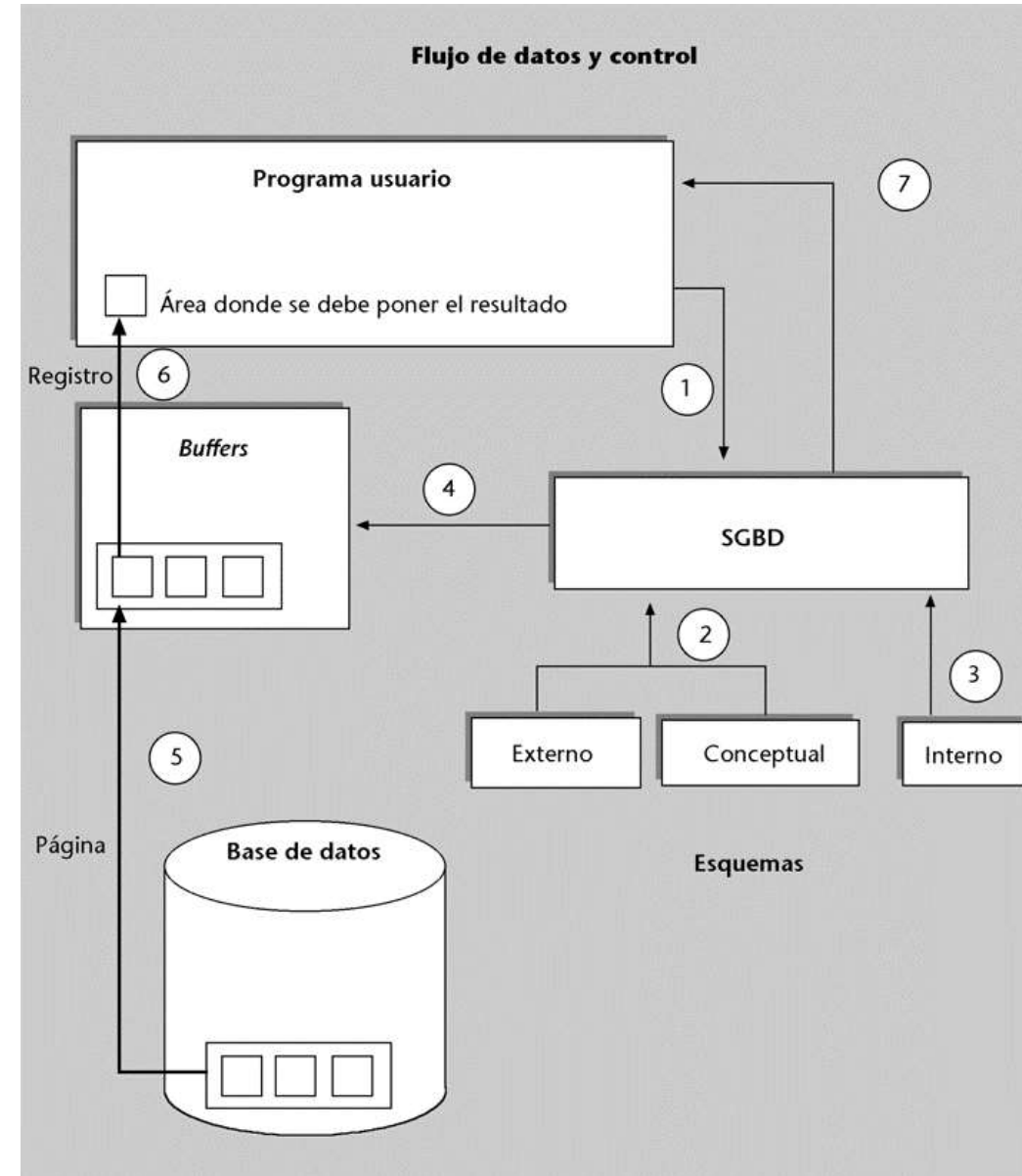
- Hay independencia física cuando los cambios en la organización física de la Base de Datos no afectan al mundo exterior (es decir, los programas usuarios o los usuarios directos).
- Hay independencia lógica cuando los usuarios no se ven afectados por los cambios en el nivel lógico.

Flujo de datos y control

- el SGBD, con la ayuda del SO, lee páginas (bloques) de los soportes donde está almacenada la Base de Datos física, y las lleva a un área de buffers o memorias caché en la memoria principal.
- El SGBD pasa registros desde los buffers hacia el área de trabajo del mismo programa

Flujo de datos y control

- Empieza con una llamada (1) del programa al SGBD. Se verifica que la sintaxis, autorización para hacerla, etc. El SGBD se basa (2) en el esquema externo con el que trabaja el programa y en el esquema conceptual.
- El SGBD determina, consultando el esquema interno (3), qué mecanismo debe seguir para responderla. Hay varias formas y diferentes caminos para responder a una consulta. Supongamos que ha elegido aplicar un hashing al valor del DNI, que es el parámetro de la consulta, y el resultado es la dirección de la página donde se encuentra (entre muchos otros) el registro del alumno buscado.
- El SGBD comprobará (4) si esta página ya se encuentra en el área de los buffers (tal vez como resultado de una consulta anterior de este usuario o de otro). Si no está, el SGBD, con la ayuda del SO, la busca en disco y la carga en los buffers (5). Si ya está, se ahorra el acceso a disco.
- La página deseada ya está en la memoria principal. El SGBD extrae, de entre los distintos registros que la página puede contener, el registro buscado, e interpreta la codificación y el resultado según lo que diga el esquema interno.
- El SGBD aplica a los datos las eventuales transformaciones lógicas que implica el esquema externo (tal vez cortando la dirección por la derecha) y las lleva al área de trabajo del programa (6).
- El SGBD retorna el control al programa (7) y da por terminada la ejecución de la consulta.

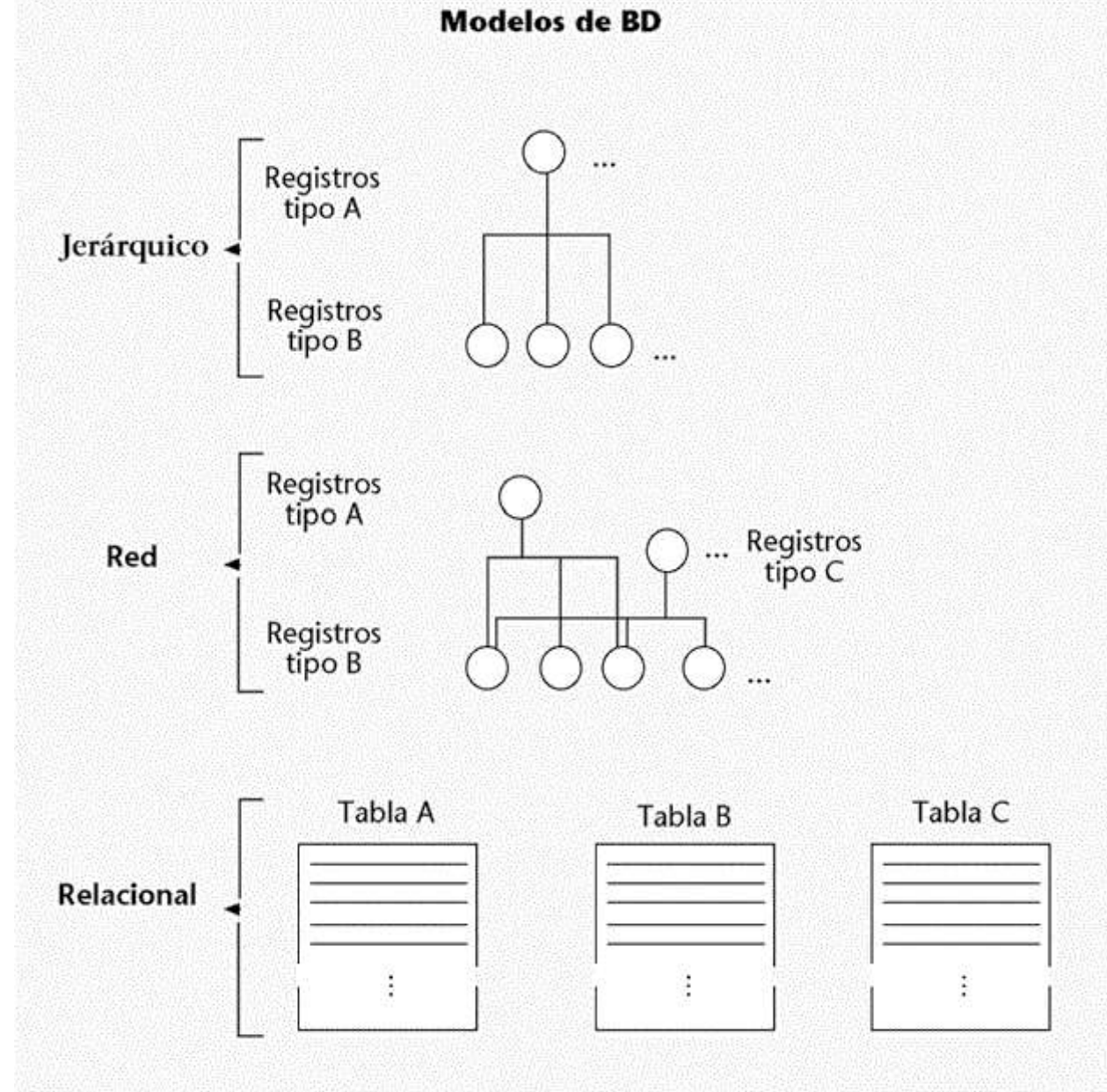


Modelos de BD

- Una BD es un modelo de la realidad
- En SGBD relacional el componente fundamental es la tabla
- Hay otros tipos de modelo de base de datos
 - Jerárquico
 - En Red
 - Relacional con objetos
- Todo modelo brinda
 - Estructuras (tablas, árboles, etc.)
 - Restricciones (reglas): dominios, claves, etc.
 - Operaciones

Evolución de los modelos

- Modelos prerelacionales
 - Años 60: Modelo jerárquico
 - Estructura: Registros interrelacionados en forma de árboles
 - IMD/DL1 de IBM
 - Principios 70: Modelo en red
 - Hay registros e interrelaciones, pero un registro no está limitado a ser “hijo” de un solo tipo de registro.
 - Comité CODASYL-DBTG propuso un estándar



Evolución de los modelos

- Años 80: aparecen SGBD basados en el modelo relacional propuesto por Codd en 1969, con SQL como lenguaje nativo
 - Estructura: tablas (relaciones)
 - Se limita al nivel lógico (independencia física)
- En la actualidad: Surgen modelos relacionales con objetos
 - Ampliación del modelo relacional.
 - Añade posibilidad de que los tipos de datos sean tipos de datos abstractos (TAD)
 - Acerca los sistemas relacionales al paradigma de la OO

Modelos semánticos y funcionales

- Se utilizan en etapa de análisis y diseño
- Permiten describir estructuras de información del mundo real, la semántica y las interrelaciones
- Se utilizan herramientas CASE (Computer Aided Software Engineering)
- El más utilizado es el ER (Entity-relationship)

Lenguajes y usuarios

- Usuarios informáticos expertos: procesos complejos – lenguajes complejos
- Usuarios no informáticos ocasionales: consultas – lenguaje sencillo
- Usuarios no informáticos dedicados: lenguajes eficientes y compactos
- SQL es el más utilizado en bases relacionales

Lenguajes

- Hay lenguajes especializados en la escritura de esquemas; es decir, en la descripción de la Base de Datos. Se conocen genéricamente como DDL o data definition language. Incluso hay lenguajes específicos para esquemas internos, lenguajes para esquemas conceptuales y lenguajes para esquemas externos.

Lenguajes

- Otros lenguajes están especializados en la utilización de la Base de Datos (consultas y mantenimiento). Se conocen como DML o data management language.
- Sin embargo, lo más frecuente es que el mismo lenguaje disponga de construcciones para las dos funciones, DDL y DML.

DML

- Lenguajes muy declarativos (o implícitos), con los que se especifica qué se quiere hacer sin explicar cómo se debe hacer.
- Lenguajes más explícitos o procedimentales, que nos exigen conocer más cuestiones del funcionamiento del SGBD para detallar paso a paso cómo se deben realizar las operaciones (lo que se denomina navegar por la Base de Datos).

DDL

- Declarativos por naturaleza

SQL

- Es básicamente declarativo, pero tiene posibilidades procedimentales

Herramientas Visuales

- Permiten usar las Bases de Datos siguiendo el estilo de diálogos con ventanas, iconos y ratón, puesto de moda por las aplicaciones Windows.
- No sólo son útiles a los usuarios no informáticos, sino que facilitan mucho el trabajo a los usuarios informáticos: permiten consultar y actualizar la Base de Datos, así como definirla y actualizar su definición con mucha facilidad y claridad.
- Por ejemplo: SSMS (SQL Server Management Studio)

Acceso a BD con lenguajes


- No suelen tener instrucciones para acceso a BD
 - Llamadas a funciones: librerías de funciones especializadas en BD (por ejemplo, las librerías ODBC). Incluir llamadas a las funciones deseadas dentro del programa escrito con el lenguaje habitual. Las funciones serán las que se encargarán de enviar las instrucciones (generalmente en SQL) en tiempo de ejecución al SGBD.
 - Lenguaje hospedado: Incluir directamente las instrucciones del lenguaje de Bases de Datos en nuestro programa. Exige utilizar un precompilador especializado que acepte en nuestro lenguaje de programación habitual las instrucciones del lenguaje de BD. Entonces SQL es el lenguaje hospedado o incorporado (embedded), y nuestro lenguaje de programación (Pascal, C, Cobol, etc.) es el lenguaje anfitrión (host).

Administración de BD

- Hay un tipo de usuario especial: el que realiza tareas de administración y control de la BD.
- Lleva a cabo una serie de funciones centralizadas de gestión y administración, para asegurar que la explotación de la BD es la correcta.
- Este conjunto de funciones se conoce con el nombre de administración de Bases de Datos (ABD), y los usuarios que hacen este tipo especial de trabajo se denominan administradores de Bases de Datos. (DBA)





















Tareas del DBA

- Mantenimiento, administración y control de los esquemas. Comunicación de los cambios a los usuarios.
- Asegurar la máxima disponibilidad de los datos; por ejemplo, haciendo copias (back-ups), administrando diarios (journals o logs), reconstruyendo la Base de Datos, etc.
- Resolución de emergencias.
- Vigilancia de la integridad y de la calidad de los datos.
- Diseño físico, estrategia de caminos de acceso y reestructuraciones.
- Control del rendimiento y decisiones relativas a las modificaciones en los esquemas y/o en los parámetros del SGBD y del SO, para mejorarlo.
- Normativa y asesoramiento a los programadores y a los usuarios finales sobre la utilización de la Base de Datos.
- Control y administración de la seguridad: autorizaciones, restricciones, etc.

The background of the slide is a dark blue gradient. On the left side, there is a complex, isometric digital structure composed of numerous glowing blue cubes and rectangular blocks. These blocks are interconnected by a network of thin, glowing lines that resemble data streams or circuitry. Several bright blue, green, and red points of light are scattered throughout the structure, some emitting small, colorful glows. The overall aesthetic is futuristic and technological.

Sistemas de Gestión de Base de Datos

Ranking de db-engines

Rank			DBMS	Database Model	Score		
Mar 2022	Feb 2022	Mar 2021			Mar 2022	Feb 2022	Mar 2021
1.	1.	1.	Oracle 	Relational, Multi-model 	1251.32	-5.51	-70.42
2.	2.	2.	MySQL 	Relational, Multi-model 	1198.23	-16.45	-56.59
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	933.78	-15.27	-81.52
4.	4.	4.	PostgreSQL  	Relational, Multi-model 	616.93	+7.54	+67.64
5.	5.	5.	MongoDB 	Document, Multi-model 	485.66	-2.98	+23.27
6.	6.	 7.	Redis 	Key-value, Multi-model 	176.76	+0.96	+22.61
7.	7.	 6.	IBM Db2	Relational, Multi-model 	162.15	-0.73	+6.14
8.	8.	8.	Elasticsearch	Search engine, Multi-model 	159.95	-2.35	+7.61
9.	9.	 10.	Microsoft Access	Relational	135.43	+4.17	+17.29
10.	10.	 9.	SQLite 	Relational	132.18	+3.81	+9.54

SGBD

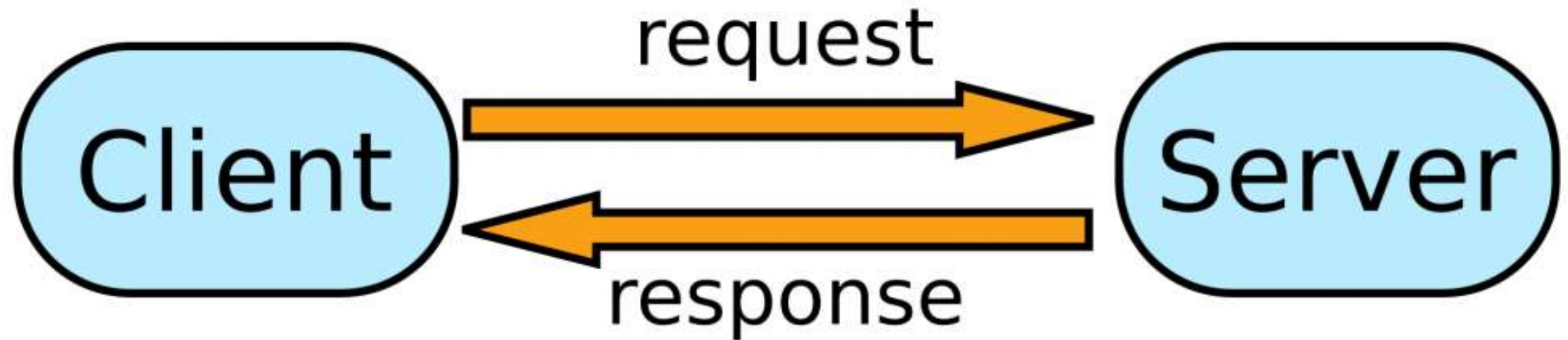
Microsoft SQL Server

- Es un producto de Microsoft
- Programado mayormente en C++, y algunas partes en C.
- Funciona en Windows y Linux

MySQL

- Es un producto de Oracle, pero es open source
- Programado en C y C++
- Funciona en múltiples plataformas

Cliente / Servidor





MS SQL Server

- Cuenta con dos opciones para poder trabajar sin problemas de licencia
 - Express
 - Developer

SQL Server 2019 Express

- Sistema Operativo
 - Windows 10, Windows Server 2016, Windows Server 2019
 - Procesador
 - Intel - compatible processor with a minimum speed of 1 GHz or a faster processor
 - RAM
 - Minimum 512MB
 - Disco
 - 4.2 GB of Disk Space
 - Linux Ubuntu, Red Hat o Suse
 - Docker

MySQL 8.0 Community Server

- Sistema Operativo:
 - Oracle Linux / Red Hat / CentOS
 - Oracle Solaris
 - Ubuntu
 - SUSE
 - Debian
 - Microsoft Windows Server (2012 R2 a 2022)
 - Microsoft Windows (10 u 11)
 - Apple (macOS 11 o 12)
 - Varios Linux



SQL Server / Clientes

- Oficiales de Microsoft
 - SQL Server Management Studio
 - Azure Data Studio
 - SQL Server Data Tools
 - Extensión de MSSQL para Visual Studio Code
 - sqlcmd
- Otros
 - dbForge Studio for SQL Server
 - Navicat for SQL Server
 - EMS SQL Management Studio for SQL Server
 - DataGrip

MySQL / Clientes

Oficial de Oracle

- MySQL Workbench

Otros

- BeeKeeper Studio
- DBeaver
- DataGrip by JetBrains
- HeidiSQL
- dbForge Studio for MySQL
- Navicat for MySQL
- SQLyog