

# 1

# Introducción a las BD y a los SGBD

Grado en Ingeniería Informática

Javier Arroyo

Yolanda García

Virginia Francisco Gilmartín

Iván Martínez

Fernando Sáenz

Facultad de Informática  
Universidad Complutense



# Índice

---

1. Formas de almacenar datos
2. Definición BD
3. SGBD
4. Modelos de datos
5. SQL
6. Diseño de una BD
7. Usuarios de BD
8. Historia de las BD



# ¿Cómo podemos almacenar datos?

---

1. Datos en el propio programa
2. Datos en archivos binarios
3. Bases de datos



# Datos almacenados en el propio programa

---

- ✓ Gran ocupación memoria
- ✓ Alto procesamiento datos
- ✓ Lentitud ejecución
- ✓ Modificaciones en el diseño o en el contenido de los datos implican recompilar los programas



# Datos almacenados en archivos binarios (I)

---

- ✓ Redundancia e inconsistencia de datos: Duplicación de información en diferentes archivos, lo que dificulta que todos mantengan la misma información
- ✓ Necesidad de escribir un nuevo programa para realizar una nueva tarea
- ✓ Formatos diferentes según los ficheros
- ✓ Cuando se necesitan datos distribuidos por distintos ficheros la consulta y modificación de datos no puede hacerse directamente, hay que desarrollar un sistema de acceso y modificación
- ✓ Las modificaciones en realidad son “simuladas” se borra un dato y se inserta el nuevo



# Datos almacenados en archivos binarios (II)

---

- ✓ Problemas de integridad de los datos: Difícil mantener restricciones de consistencia
- ✓ Problemas en la atomicidad (todo o nada) de las operaciones. Ej: transferencia de dinero de una cuenta a otra debería ocurrir de manera completa o no producirse en absoluto
- ✓ Dificultad en acceso concurrente. Ej: retirada de dinero de dos clientes de manera “simultánea” debe dejar la BD en un estado consistente
- ✓ Seguridad: Definición de distintos niveles de acceso a los datos: ¿pueden acceder todos los programas o usuarios a todos los datos?



# Bases de datos

---

- ✓ Colección **persistente** de datos relacionados usada para representar información
  - Se almacena una única copia de los mismos
    - Consistencia de datos
    - Menor coste de almacenamiento
  - Consulta y modificación de datos directa
    - Las modificaciones son reales
  - Acceso concurrente controlado
  - Seguridad (permite decidir qué usuarios pueden ver qué información)
  - ...



# Ficheros vs. Bases de datos

---

- ✓ Los ficheros son preferibles a las BD cuando:
  - Datos y aplicaciones simples, bien definidas y sin visos de cambio
  - Pocos datos y requisitos de acceso en tiempo real que no pueden cubrir los SGBD
  - No se requiere acceso concurrente



# ¿Dónde se usan las BD?



<http://cdn.memegenerator.net/instances/400x/30454693.jpg>



# ¿Qué es una base de datos?

---

- **Base de datos:** Colección de datos pertenecientes a un mismo contexto que se almacenan para su posterior uso
- **Datos:** Hechos conocidos que se pueden grabar y que tienen un significado implícito
  - Productos que vende una empresa
  - Clientes de un taller de coches
  - Usuarios de un sitio web
  - ...



# Ejemplo. Base de datos de una biblioteca

---

- ✓ Permite la gestión de la información relativa a libros, películas, juegos y usuarios
- ✓ **Elementos:** Libros, autores, usuarios, préstamos, etc.
- ✓ **Relaciones entre los elementos:**
  - Un libro tiene varios autores
  - Existen varios tomos con el mismo título
  - Los usuarios pueden pedir prestados 4 libros como mucho
- ✓ **Consulta y actualización de la base de datos:**
  - Listado de películas
  - Listado de usuarios con préstamos vencidos
  - Listado de libros reservados a usuarios
  - Alta de nuevos libros
  - Alta de usuarios



# Sistema de Gestión de Bases de Datos (SGBD) (I)

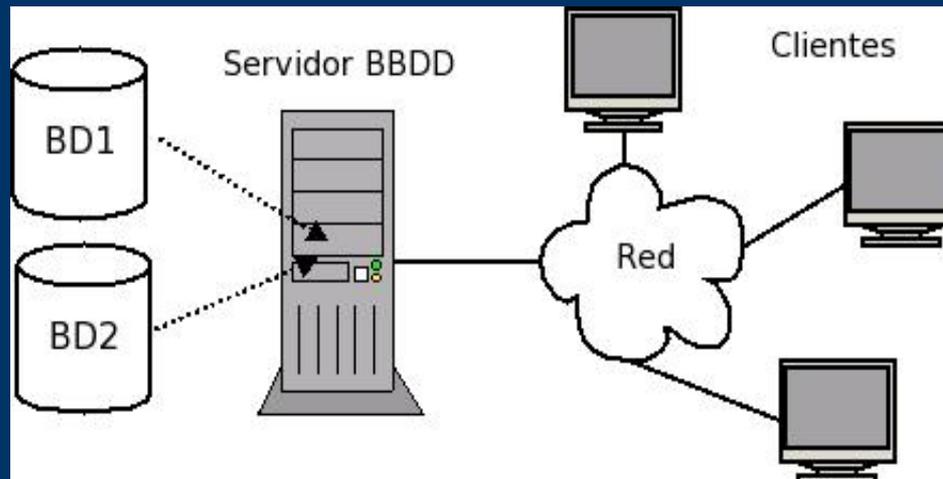
---

- Tiene como objetivo proporcionar un entorno práctico y eficiente para almacenamiento y recuperación de datos. Debe garantizar:
  - ✓ fiabilidad ante caídas del sistema
  - ✓ integridad de los datos (también ante accesos simultáneos)
- Los SGBD proporcionan: concurrencia, integridad y seguridad



# Sistema de Gestión de Bases de Datos (SGBD) (II)

- Habitualmente están implementados mediante una arquitectura cliente servidor



- Sistema de base de datos: Base de datos + SGBD



# SGBD. Características (I)

---

- Seguridad
  - ✓ Un SGBD debe contar con un subsistema de seguridad y autorización que permita al administrador del sistema crear cuentas y especificar privilegios
  - ✓ Debe ser posible indicar qué usuarios pueden hacer según qué cosas
- Suministro de múltiples interfaces con los usuarios. Desde interfaces con menús hasta intérpretes de SQL o APIs para programadores
- Cumplimiento de restricciones de integridad
  - ✓ Se debe permitir dotar de tipo a los datos, y el sistema debe asegurar que se respetan esos tipos
  - ✓ Se deben asegurar que las relaciones entre registros se mantengan de manera consistente



# SGBD. Características (II)

---

- Representación de vínculos entre datos. Un SGBD debe permitir establecer relaciones entre datos, de modo que se puedan utilizar para obtener y actualizar con rapidez y eficiencia datos mutuamente relacionados
- Multiusuario y control de concurrencia. Se debe permitir y controlar el acceso simultáneo de varios usuarios a los datos
- Recuperación de fallos. El SGBD debe estar preparado para situaciones de “caídas” en los sistemas, de modo que si una operación ha sido interrumpida, se vuelva al estado anterior en los datos



# SGBD. Ventajas

---

- ✓ Normalización en el desarrollo de aplicaciones
- ✓ Tiempo de desarrollo de aplicaciones reducido
- ✓ Flexibilidad para añadir o eliminar datos necesarios
- ✓ Disponibilidad inmediata de las actualizaciones de datos para todos los usuarios



# SGBD. Visión de los datos

---

- Los SGBD proporcionan al usuario una visión abstracta de los datos. Es decir, oculta al usuario la implementación real del almacenamiento de datos (¡que es la parte compleja!)
- Podemos hablar de tres niveles de abstracción
  - Nivel físico: Describe la implementación del almacenamiento de los datos => esquema físico
  - Nivel lógico: Describe qué datos son almacenados realmente en la BD y las relaciones existentes entre ellos => esquema lógico. Lo define el administrador
  - Nivel conceptual: Representación al más alto nivel de abstracción



# Modelos de datos (I)

---

- ✓ Ofrecen un modo de describir el diseño de bases de datos en los niveles físico, lógico y conceptual
- ✓ Los modelos de datos son una descripción de los contenedores de datos y de los métodos para almacenar y recuperar información de esos contenedores
- ✓ Ofrecen un modo de describir el diseño de bases de datos en los niveles físico, lógico y conceptual
- ✓ Colección de herramientas para describir:
  - Datos
  - Relaciones entre los datos
  - Semántica de los datos
  - Restricciones de consistencia e integridad



# Modelos de datos (II)

---

- ✓ Existen varios tipos de modelos de datos:
  - Modelo conceptual:
    - ✓ Representación a un alto nivel de abstracción
    - ✓ Descripción de la BD fácil de entender
    - ✓ Se crea a partir de los requisitos del usuario
  - Modelo lógico o de implementación:
    - ✓ Descripción de la estructura de la BD en términos de las estructuras que procesa un SGBD
    - Depende del tipo de SGBD
  - Modelo físico:
    - ✓ Descripción de la implementación de la BD (estructuras de almacenamiento y métodos usados para tener acceso a los datos)
    - ✓ Proporciona el detalle de cómo se almacenan los datos en el ordenador
    - Depende del SGBD concreto



# Modelo conceptual

---

- ✓ Representación a un alto nivel de abstracción
- ✓ Tipos:
  - Modelo Entidad-Relación:
    - ✓ Consiste en una colección de objetos básicos (entidades) y de las relaciones entre ellos
    - ✓ Se usa mucho en el diseño de bases de datos
  - Modelo de datos orientado a objetos
    - ✓ Extensión del modelo entidad-relación con los conceptos de encapsulación, métodos e identidad de los objetos



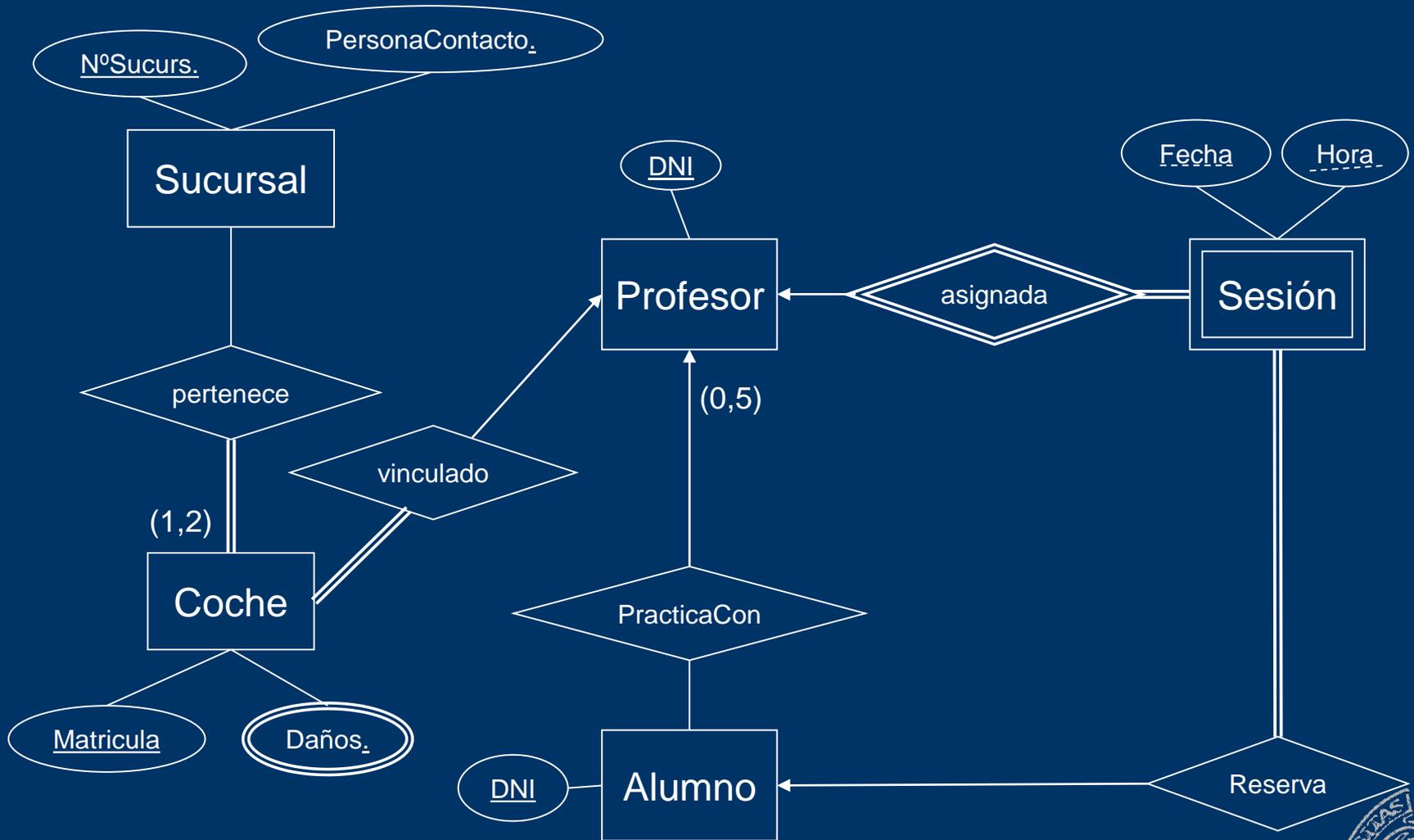
# Modelo Entidad-Relación

---

- ✓ Es un tipo de modelo conceptual
- ✓ Se representa a través de diagramas (ayudan a entender los datos y como se relacionan entre ellos)
- ✓ Consta de objetos básicos (**entidades**) y de relaciones entre ellos (**relaciones**)
- ✓ Entidad: Objeto sobre el que queremos almacenar información. Por ejemplo, una persona
- ✓ Las entidades se describen por un conjunto de **atributos** que definen o identifican las características de la entidad. Por ejemplo, el DNI, el nombre y los apellidos de una persona
- ✓ Diagrama E-R
  - Rectángulos => entidades
  - Elipses => atributos
  - Rombos => relaciones



# Modelo entidad-relación. Ejemplo



# Modelo lógico

---

- ✓ A veces llamados Modelos de Implementación
- ✓ Descripción de la estructura de la BD en términos de las estructuras que procesa un SGBD
- ✓ Tipos:
  - Modelo relacional
    - ✓ Usa una colección de tablas para representar tanto los datos como sus relaciones
    - Modelo más ampliamente usado
  - Modelo de datos semi-estructurado: Permite especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos (XML)
  - Modelo de Objetos: Utiliza conceptos similares a los lenguajes de programación orientado a Objetos
  - Modelos NoSQL: Grafos, Orientadas a Documentos, Clave-Valor
  - ...



# Modelo relacional (I)

---

- ✓ Es un tipo de modelo de implementación de datos
- ✓ Se llega a él a partir del modelo entidad-relación
- ✓ El modelo relacional emplea colección de tablas para representar los datos y sus relaciones
- ✓ El concepto principal es la tabla o relación
  - Cada tabla debe tener un nombre exclusivo
  - Cada tabla o relación es un conjunto de tuplas que se corresponde con una fila de la tabla



# Modelo relacional. Ejemplo

## ✓ Ejemplo de tabla:

usuarios (dni, apellidos, nombre, dirección, cc)

dni	apellidos	nombre	dirección	cc
08932583P	López Amor	María	Calle Loma 2	004024090248091245
74657235U	García Mancha	Luis	Plaza Retamar 4	857345784683639848
37651947T	Pozo Mares	Fernando	Calle Fernando el Santo 56	765736572365473483
78538465G	Salinas Pérez	Antonio	Calle Ferrocarril 14	376581248912481736
73256356M	Arroyo Mata	Marta	Plaza Remonta 13	5762499241928419



# Modelo relacional. Terminología (I)

- ✓ **Esquema de una tabla:** Viene dado por el nombre de la tabla y su lista de atributos

Alumnos (DNI, Apellidos, Nombre, Telefono, Acceso)

- ✓ **Instancia de una tabla:**

- Conjunto de tuplas de la tabla
- Ejemplo: Instancia de la tabla Alumnos:

- ✓ **Notación matemática:**

{ (01234567Z, Vázquez Montalbán , Manuel, 9112345678, normal), .... }

- ✓ **Notación en forma de tabla**

DNI	Apellidos	Nombre	Telefono	Acceso
01234567Z	Vázquez Montalbán	Manuel	9112345678	normal
...				



# Modelo relacional. Terminología (II)

- ✓ **Campo** de una tabla: Elemento de información dentro de un registro. Tiene tipo
- ✓ **Registro** de una tabla: Conjunto de campos correspondientes al mismo elemento de la BD

CONTACTOS (NOMBRE APE1 APE2, TELEFONO )							Esquema
Luz	Pérez	Pérez	623	12	12		Instancia
Juan	López	López	687	65	43		Instancia
Registro	Ana	Saz	García	634	56	78	Instancia

Campe

- ✓ **Esquema de Base de Datos:**
  - Colección de todos los esquemas de las tablas individuales
  - Representa el diseño lógico de la Base de Datos
- ✓ **Instancia o ejemplar de la Base de Datos:**
  - Instantánea de los datos almacenados en la BD en un momento dado



# Modelos físicos

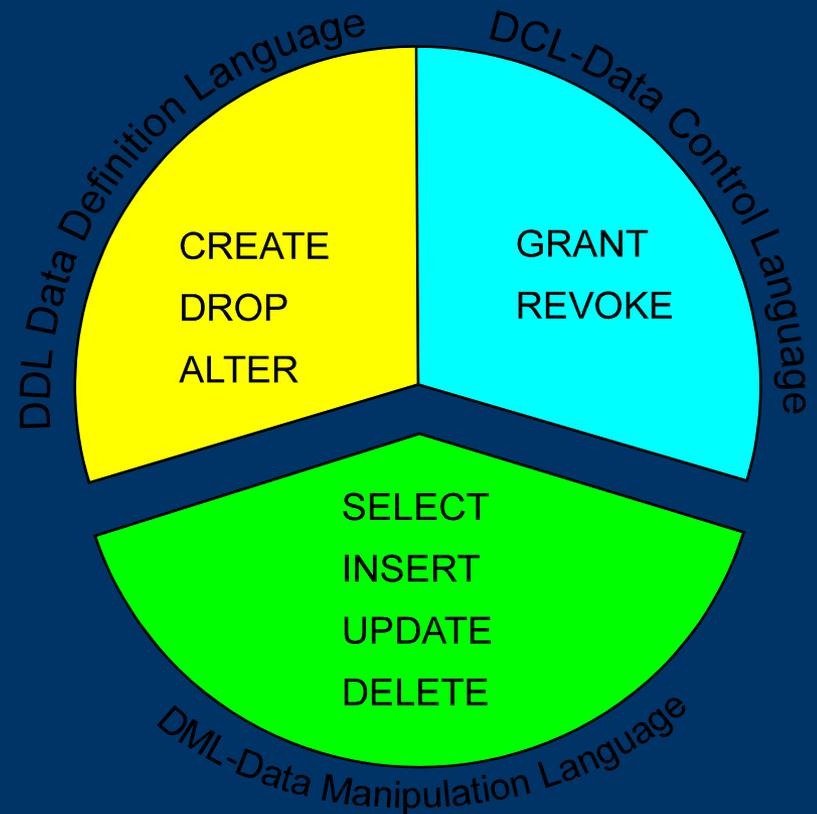
---

- ✓ Muy determinados por el SGBD: p.ej. Oracle o SQLServer
- ✓ Proporcionan detalles de cómo se almacenan los datos en el ordenador
- ✓ Representan información como el formato y el orden de registros, caminos de acceso a datos, etc.
- ✓ Dirigidos a especialistas en computación y gestión de datos, no a usuarios finales



# Lenguaje relacional SQL

- ✓ SQL (Structured Query Language)
- ✓ Es el lenguaje que permite manejar bases de datos relacionales
- ✓ Se compone de:
  - Lenguaje de definición de datos (LDD): Creación y modificación de las estructuras que albergarán los datos
  - Lenguaje de manipulación de datos (LMD): Consultas y modificaciones de los datos
  - Lenguaje de Control de Datos (DCL): Gestión de permisos



# Lenguaje de definición de datos (LDD)

---

- ✓ El LDD sirve para crear y definir los esquemas de la BD
  - La descripción de un esquema de base de datos se almacenará en una BD del SGBD llamada diccionario de datos
- ✓ Las tablas del diccionario de datos contienen metadatos
  - Esquema de la base de datos:
    - ✓ Nombre de las tablas y de sus atributos
    - ✓ Dominio y longitud de los atributos
  - Restricciones de integridad (aseguran la coherencia de los datos en la BD)



# LDD. Ejemplo

---

```
CREATE TABLE usuarios(  
  dni          CHAR(9)          NOT NULL      ,  
  apellidos   CHAR(30)         NOT NULL      ,  
  nombre      CHAR(20)         NOT NULL      ,  
  direccion   VARCHAR(150)     ,  
  cc          CHAR(5)          ,  
  edad        INT              ,  
  fecha_nacimiento DATE        ,  
  last_updated  TIMESTAMP     NOT NULL      ,  
                                     WITH DEFAULT ,  
  PRIMARY KEY (dni)              ,  
  FOREIGN KEY (cc) REFERENCES localidades  
);  
  
ALTER TABLE usuarios  
ADD CONSTRAINT POPUL CHECK (  
  edad BETWEEN 18 AND 65  
);  
  
DROP TABLE usuarios;
```



# Lenguaje de Manipulación de Datos (LMD)

---

- ✓ Sirven para la inserción, eliminación, modificación y consulta de información en la base de datos
- ✓ Es lo que usan los usuarios (y aplicaciones) finales
- ✓ Sentencias SELECT, INSERT, UPDATE, DELETE
- ✓ Ejemplo de LMD:

```
SELECT nombre, apellidos  
FROM usuarios  
WHERE edad >= 25;
```



# Lenguaje de Control de Datos (LCD)

---

- ✓ Lo utilizan los administradores de bases de datos para:
  - controlar el acceso a los datos por parte de usuarios y grupos (roles)
  - definir las operaciones admitidas a cada usuario/grupo
- ✓ Ejemplos:

```
GRANT
    SELECT,
    UPDATE (edad)
ON usuarios
TO usu_maria;
```

```
REVOKE
    SELECT,
    UPDATE(nombre, apellido)
ON TABLE usuarios
FROM rol_antonio;
```



# Diseño de una base de datos

---

- ✓ Especificación de requisitos: Identificar completamente los requisitos de datos de los usuarios de la base de datos
- ✓ Diseño conceptual: Construcción de un esquema conceptual (datos y su relaciones) a partir de los requisitos del usuario
  - **Modelo entidad-relación**
- ✓ Diseño lógico:
  - Crear un esquema lógico que represente correcta y eficientemente al esquema conceptual
    - **Modelo relacional**
  - Normalización. Técnicas formales para verificar la calidad del esquema lógico → redundancia de datos
- ✓ Diseño físico:
  - Decidir los detalles de la implementación física.
  - Organización de archivos e índices → puede afectar en gran medida al rendimiento de las operaciones sobre la BD



# Usuarios de la base de datos

---

- ✓ Usuarios normales: Usan la BD a través de aplicaciones escritas previamente
- ✓ Programadores de aplicaciones: Interaccionan con la BD a través de llamadas que usan LMD
- ✓ Usuarios avanzados: Consultas en SQL
- ✓ Usuarios especializados: Escriben aplicaciones de bases de datos especializadas que no encajan en el marco tradicional del procesamiento de datos
  - Ej: sistemas expertos, data mining



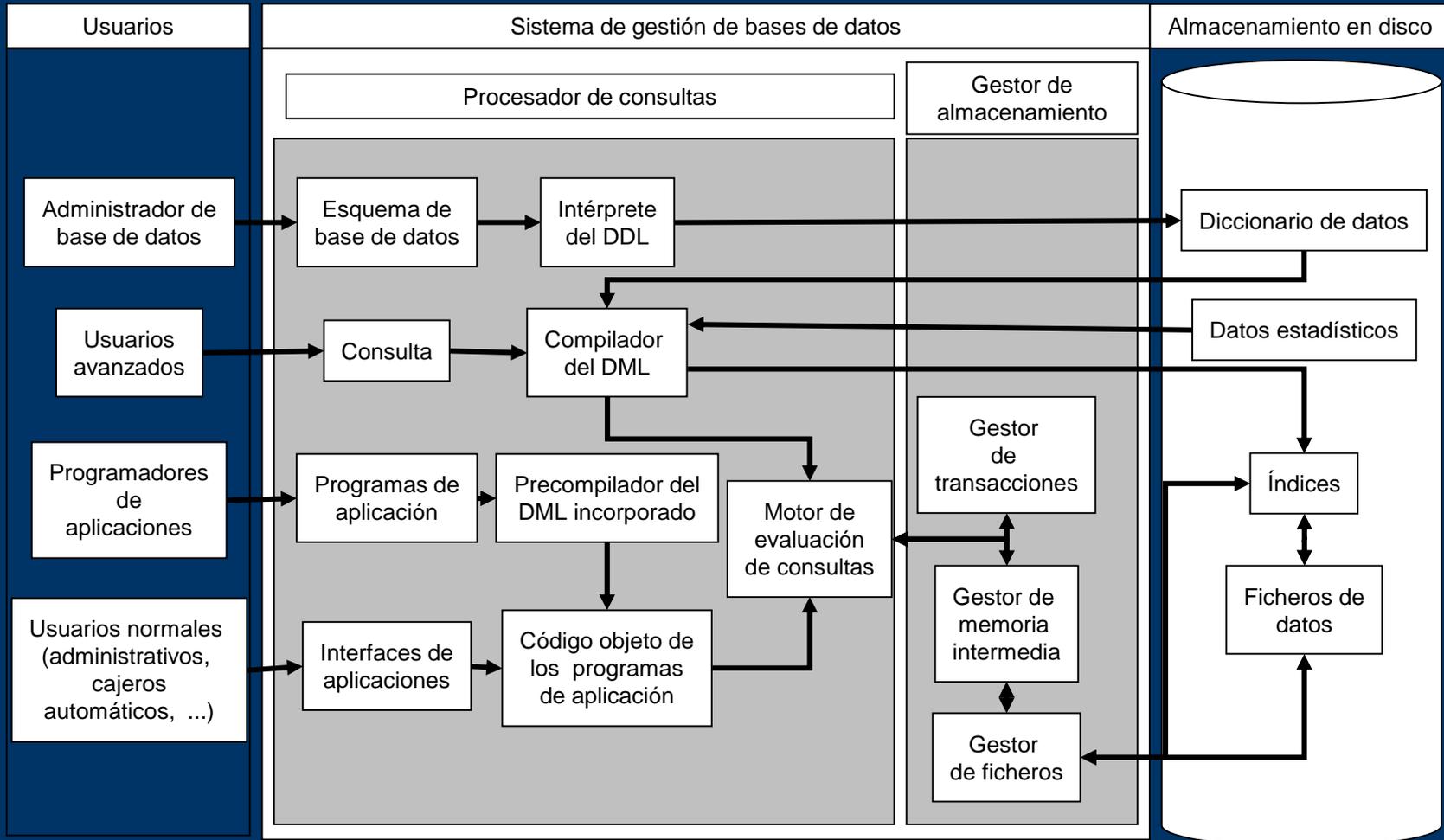
# Administrador de la base de datos

---

- ✓ Definición del esquema
- ✓ Definición de la estructura de almacenamiento y del método de acceso
- ✓ Modificación del esquema y la organización física
- ✓ Concesión de derechos de acceso
- ✓ Mantenimiento



# Estructura global de un SGBD



# SGBD. Componentes

---

- ✓ Gestor de concurrencia y transacciones: Permite controlar la ejecución de transacciones que operan en paralelo, accediendo a información compartida y, por lo tanto, interfiriendo potencialmente unas con otras
  - Ejemplo: Reservar un asiento en una avión desde una web cuando decenas de personas en el mundo pueden estar reservándolo también
- ✓ Gestor de almacenamiento: El gestor de almacenamiento ejerce de interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y las consultas remitidas al sistema
- ✓ Procesamiento de consultas: Transforma una consulta escrita en lenguaje de alto nivel (SQL, por ejemplo) en una secuencia eficiente de operaciones en el nivel físico
- ✓ Estructuras de datos



# Historia de los sistemas de bases de datos (I)

---

- ✓ Antigüedad
  - ✓ Bibliotecas y toda clase de registros
  - ✓ Se utilizaban bases de datos para recoger información sobre las cosechas y los censos
  - ✓ Búsqueda manual lenta y poco eficaz al no contar con máquinas
  - ✓ Al aparecer las primeras computadoras el concepto de base de datos quedó ligado a la informática
- ✓ Década de los 50: Procesamiento de datos usando cintas magnéticas para el almacenamiento
  - ✓ Las cintas sólo permitían acceso secuencial
  - ✓ Tarjetas perforadas para entrada de datos



# Historia de los sistemas de bases de datos (I)

---

- ✓ Década de los 60:
  - ✓ Los discos duros permiten acceso directo a los datos
  - ✓ Modelos de datos en red y jerárquico
- ✓ Década de los 70:
  - ✓ Ted Codd define el modelo de datos relacional
    - Gran éxito debido a la simplicidad del modelo y a la posibilidad de ocultar los detalles de la implementación
  - ✓ Larry Ellison desarrolló Oracle



# Historia de los sistemas de bases de datos (II)

---

- ✓ Década de los 80:
  - ✓ Prototipos de bases de datos relacionales evolucionan hacia sistemas comerciales (Ej: System R => SQL / DS (IBM))
    - ✓ SQL se convierte en el estándar industrial
  - ✓ Sistemas de bases de datos paralelos y distribuidos
- ✓ Década de los 90:
  - ✓ Sistemas de bases de datos orientados a objetos
  - ✓ Excel, Access...
  - ✓ Aplicaciones de minería de datos y de toma de decisiones
  - ✓ Aparición del comercio electrónico en la Web



# Historia de los sistemas de bases de datos (y III)

---

- ✓ Años 2000
  - ✓ XML tecnología de almacenamiento de bases de datos y XQuery como lenguaje de consultas a XML
  - ✓ Administración de bases de datos automática
- ✓ Años 2010 : Auge del movimiento NoSQL (No SQL o Not only SQL) que da mejor respuesta a las necesidades de almacenamiento masivo, escalabilidad y elevado volumen de consultas
  - ✓ Uso de bases de datos distribuidas y de escasa estructuración
    - Ej: Necesidades de bases de datos de redes sociales como Facebook



# Acerca de *Creative Commons*

---

## Licencia CC (*Creative Commons*)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

- ① Reconocimiento (*Attribution*):  
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
- Ⓜ No comercial (*Non commercial*):  
La explotación de la obra queda limitada a usos no comerciales.
- Ⓒ Compartir igual (*Share alike*):  
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Material original elaborado por Iván Martínez y Fernando Saez, con modificaciones de Javier Arroyo, Yolanda García y Virginia Francisco.

