

# Introducción a las bases de datos

Bases de Datos

Curso 2018-2019

**Jesús Correas – [jcorreas@ucm.es](mailto:jcorreas@ucm.es)**

**Departamento de Sistemas Informáticos y Computación  
Universidad Complutense de Madrid**

(Basado en material creado por Mercedes García Merayo y Luis Garmendia)

# Bibliografía

- Bibliografía básica:

- ▶ R. Elmasri, S.B. Navathe. **Fundamentals of Database Systems** (6a Ed). Addison-Wesley, 2010. (en español: **Fundamentos de Sistemas de Bases de Datos** (5a Ed). Addison-Wesley, 2007).  
Capítulos 1 y 2.

- Bibliografía complementaria:

- ▶ A. Silberschatz , H. F. Korth, S. Sudarshan. **Fundamentos de bases de datos** (5a Ed), McGraw-Hill, 2006.  
Capítulo 1.

# Contenido

- Introducción. Almacenamiento y representación de la información.
- Definición de base de datos.
  - ▶ Ejemplo de base de datos.
- Sistema de Gestión de Bases de Datos.
  - ▶ Características.
  - ▶ Propiedades.
  - ▶ Lenguajes de operación.
- Actores que intervienen en un SGBD.
- Arquitectura de funcionamiento de un SGBD.
- Diseño de bases de datos.

# Almacenamiento y representación de la información

- Prácticamente todos los sistemas informáticos necesitan **representar información** y **almacenar y manipular datos**.
- En las asignaturas básicas de programación se utilizan **ficheros**:
  - ▶ Proporcionan **persistencia**,
  - ▶ Permiten distintos modos de acceso a los datos: **secuencial** (ficheros de texto), **directo** (binarios).
- También se han visto **estructuras de datos**.
- Todo esto está definido en función de la **estructura física de los dispositivos de almacenamiento (discos, memoria)**.
- La estructura de la información de una aplicación **puede ser muy compleja**.

Es necesario utilizar técnicas que permitan la **representación y almacenamiento de la información** de forma **independiente del dispositivo** y del **modo de almacenamiento**.

# Ejemplo de base de datos

- Sistema para la gestión de matriculación en la universidad.
- **Elementos** (*entidades*): información relativa a **estudiantes**, **asignaturas**, **departamentos**, **profesores**, **calificaciones**, **requisitos**, ...
- **Atributos**: propiedades que tienen los elementos del sistema: **edad**, **número de créditos**, **tipo de asignatura (troncal, optativa)**, **itinerario**, **aula**, ...
- **Relaciones** entre los elementos del sistema de matriculación:
  - ▶ Cada estudiante se puede matricular de varias asignaturas.
  - ▶ Cada asignatura es ofertada por un departamento.
  - ▶ Una asignatura debe ser impartida por al menos un profesor.
  - ▶ Cada profesor pertenece a un departamento y solo a uno.
  - ▶ Las asignaturas pueden tener requisitos de matriculación.
  - ▶ Una asignatura tiene un cupo máximo de alumnos que se pueden matricular.
  - ▶ ...

# Ejemplo de base de datos

- **Operaciones** que se deben poder realizar:

- ▶ Consultar la lista de alumnos matriculados en una asignatura.
- ▶ Consultar el expediente de un alumno.
- ▶ Calcular la nota media obtenida en una asignatura en un curso determinado.
- ▶ Anular la matrícula de un alumno.
- ▶ Dar de alta una nueva asignatura.
- ▶ Consultar los alumnos que han cursado todas las asignaturas obligatorias del grado y han obtenido una media global mayor a 8 (p. ej. para solicitar becas).
- ▶ Consultar la lista de alumnos que han tenido al mismo profesor en dos o más asignaturas.
- ▶ ...

# Almacenamiento y representación de la información

El **sistema de base de datos** debe proporcionar una **solución a los siguientes problemas**:

- ❶ Evitar inconsistencias por información duplicada: **redundancia**.
- ❷ Permitir representar **relaciones complejas** entre los datos.
- ❸ Permitir distintas formas de **acceso eficiente** a los datos:
  - ▶ **Consultas complejas sobre los datos, múltiples vistas.**
- ❹ **Formato homogéneo** y uniforme de los datos.
- ❺ Mecanismos para proteger la **integridad de los datos**.
- ❻ Permitir realizar operaciones complejas **como si fueran atómicas**.
- ❼ Almacenamiento centralizado de los datos: **acceso concurrente**.
- ❽ **Seguridad** en el acceso a los datos (acceso inapropiado).
- ❾ **Seguridad y recuperación** de fallos del sistema.

# Definición de base de datos

- Los sistemas de bases de datos ofrecen una **solución integral** a todos estos problemas.

## Base de Datos (BD)

Es una **colección de datos relacionados**.

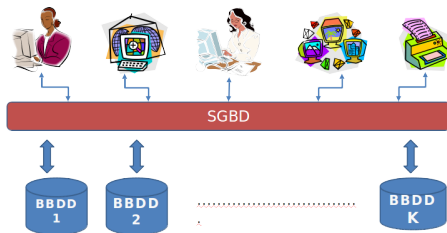
- Una BD representa algún aspecto específico del mundo real: el **universo de discurso** (o “minimundo”).
- Es una colección de datos lógicamente coherente con algún significado inherente.
- Con un propósito específico y un grupo de usuarios y aplicaciones que harán uso de los datos.
- Nos centraremos en un tipo específico de BD: las **BD relacionales**.



# Sistema de Base de Datos

- **Sistema de Gestión de Bases de Datos (SGBD):** conjunto de programas (de propósito general) que permiten crear y mantener una base de datos.
- **De propósito general:** el SGBD es el mismo para cualquier BD.

**Sistema de Bases de Datos = BD + SGBD**



- Un Sistema de BD ofrece una serie de **características:**

# Sistema de Base de Datos. Características

## 1. Un sistema de BD es **autodescriptivo**:

- ▶ Para la definición de la BD se utiliza una BD especial denominada **catálogo** que contiene la estructura de la BD específica.
- ▶ Son los denominados **metadatos**.

## 2. **Aislamiento entre programas y datos**:

- ▶ La estructura de la BD está separada de los programas: es posible modificar la BD sin necesidad de modificar (todos) los programas que acceden a esa BD.

## 3. Un Sistema de BD proporciona una **visión abstracta de los datos**:

- ▶ Una **representación conceptual** de los datos.
- ▶ Oculta los detalles de almacenamiento o implementación de operaciones.

# Sistema de Base de Datos. Características

## 4. Permite utilizar **varias vistas de los datos**:

- ▶ Un **subconjunto** de los datos (ej.: para un profesor, mostrar solo los datos de alumnos matriculados en la asignatura que imparte).
- ▶ Una **vista parcial** de los datos (dependiendo del tipo de usuario, ocultando datos confidenciales).
- ▶ Datos *virtuales* derivados de otros datos: cálculo de totales, resúmenes, etc. que no están almacenados en la BD.

## 5. **Compartición de datos entre múltiples usuarios concurrentes**:

- ▶ Los datos se mantienen en un único almacén al que pueden acceder varios usuarios y aplicaciones.
- ▶ **Transacciones** para controlar accesos **concurrentes**.

## 6. **Seguridad** frente a **accesos no autorizados y fallos del sistema**.

## 7. **Eficiencia**.

# Propiedades **ACID**

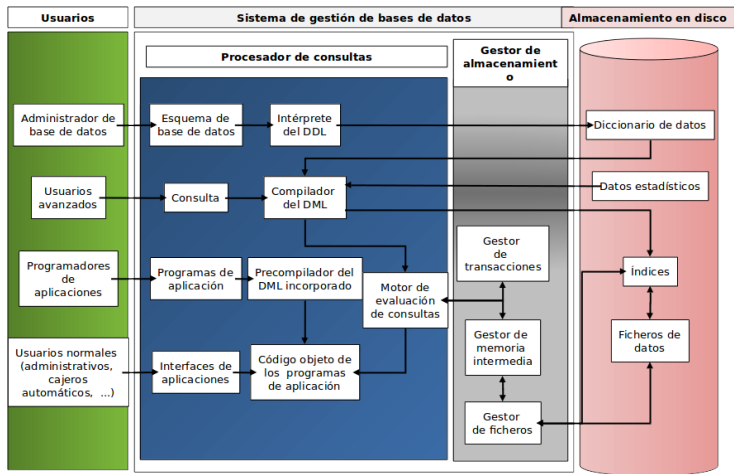
- Los SGBD relacionales agrupan varias operaciones de modificación de datos en una **transacción**.
  - ▶ **Ejemplo:** en el sistema de matrícula de la universidad, al realizar la matrícula de un alumno en una asignatura se debe:
    - ★ Decrementar el número de plazas disponibles en el grupo elegido.
    - ★ Registrar la asignatura en los datos de matrícula del alumno.
    - ★ Actualizar la contabilidad para realizar el cobro de la matrícula.
- **Un SGBD debe proporcionar propiedades **ACID** sobre transacciones:**
  - ▶ **A de atomicidad:** En una transacción, todas las operaciones se terminan, o bien no se realiza ninguna.
  - ▶ **C de consistencia:** Una consulta debe ser consistente con el estado de la base de datos **en el instante de inicio** de la ejecución de la consulta.
  - ▶ **I de aislamiento:** Una transacción no completada (con commit) es invisible al resto del mundo (las demás transacciones).
  - ▶ **D de durabilidad:** Cuando se completa una transacción con commit, entonces es imposible que la base de datos la pierda.

# Lenguajes de operación sobre un SGBD

- Para definir y operar sobre una BD en un SGBD se utilizan diversos **lenguajes**.
  - ▶ **DDL**: Lenguaje de **definición** de datos.
    - ★ Para definir la estructura de la BD (*esquema* de la BD).
    - ★ Restricciones: de dominio, de integridad referencial, aserciones.
    - ★ Mecanismos de acceso lógico y físico (almacenamiento).
    - ★ Crear y eliminar objetos de la BD (tablas, índices, etc.)
  - ▶ **DML**: Lenguaje de **manipulación** de datos.
    - ★ Consultas complejas:  

```
SELECT dpto.nombre, asig.nombre FROM dpto, asig  
WHERE dpto.id = asig.idDpto AND asig.creditos > 9
```
    - ★ Modificación de datos: INSERT, DELETE, UPDATE.
  - ▶ **DCL**: Lenguaje de **control** de datos.
    - ★ Control de acceso a los datos contenidos en la BD.
    - ★ Definiciones de usuarios, roles, grupos de usuarios.
- El lenguaje estándar para las **BD relacionales** es **SQL** (*Structured Query Language*), que incluye los tres lenguajes anteriores.

# Estructura general de funcionamiento de un SGBD

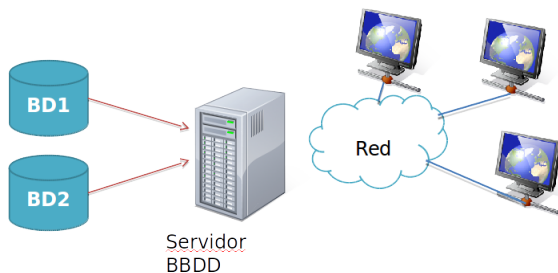


# Actores que intervienen en un Sistema de BD

- **Usuarios finales:** Utilizan la BD a través de aplicaciones (con consultas SQL preconfiguradas).
- **Usuarios avanzados:** Utilizan un lenguaje de consulta más avanzado o directamente SQL.
- **Desarrolladores de aplicaciones:** Programan aplicaciones que utilizan DML.
- **Diseñadores de BD:** Diseñan la estructura conceptual y lógica de la base de datos.
- **Administradores de BD:**
  - ▶ Definen y modifican el esquema de la BD.
  - ▶ Definen la estructura de almacenamiento, acceso y organización física.
  - ▶ Asignan derechos de acceso a roles y usuarios.
  - ▶ Mantenimiento: seguridad, *backup & recovery*, eficiencia (índices, consumo de recursos, interbloqueos, optimización de SQL, etc.).

# Arquitectura de funcionamiento de un SGBD

- Normalmente utilizan una **arquitectura cliente-servidor**:
  - ▶ Los clientes son programas de aplicaciones, navegadores (que invocan programas en el servidor), etc. que realizan consultas sobre el SGBD.
  - ▶ El servidor atiende peticiones de los clientes.
  - ▶ Las herramientas de administración también son clientes.





# Arquitectura de funcionamiento de un SGBD

- **Procesador de consultas.**

- ▶ Traduce las sentencias en operaciones sobre la BD.
- ▶ Las sentencias DML pueden ser **muy complejas**: se generan planes de ejecución.
- ▶ Se elige el plan más eficiente (sobre datos estadísticos).

- **Gestor de transacciones y control de concurrencia.**

- ▶ Para permitir múltiples accesos simultáneos.
- ▶ Garantiza la consistencia de la BD (propiedades **ACID**).

- **Gestor de almacenamiento.**

- ▶ Interfaz con las estructuras de bajo nivel de la BD: gestor de archivos, memoria intermedia (En el caso de Oracle, datafiles, SGA, *redo log*, etc.)

# Diseño de Bases de Datos

- Una BD es un componente fundamental del **sistema de información** de una organización.
- El diseño de la BD (y del S.I.) se realiza en varias fases:
  1. **Recopilación y análisis de requisitos.**
  2. **Diseño conceptual.**
    - ▶ Con los requisitos se construye un **modelo conceptual de datos**.
    - ▶ Descripción de alto nivel: no se especifican detalles de almacenamiento físico.
    - ▶ Utilizaremos el **modelo entidad-relación**.
  3. **Diseño lógico.**
    - ▶ Traduce el esquema conceptual a un **modelo lógico de datos**.
    - ▶ Se utilizan técnicas de **normalización** para detectar y resolver problemas potenciales.
    - ▶ Resultado: un modelo de datos que se puede implementar en un SGBD.
    - ▶ Utilizaremos el **modelo relacional**.
  4. **Diseño físico.**
    - ▶ Refinamiento para optimización del rendimiento.
    - ▶ organización de archivos, índices, desnormalización, etc.

# Lo que veremos durante el curso

En las siguientes clases estudiaremos diferentes aspectos de las bases de datos en el siguiente orden:

2. Diseño conceptual: modelo entidad-relación.
3. Diseño lógico: modelo relacional. Álgebra relacional.
4. SQL: Structured Query Language.
5. Introducción a PL/SQL. Disparadores.
6. Introducción a transacciones y Control de Concurrency
7. Conceptos avanzados.