**CONCEPTOS CLAVE ADBD**

# Tema 1: Introducción

* Datos: valores que se guardan y analizan. Los que nos interesan son los **persistentes** en memoria secundaria. Los guardamos en forma de tabla, por lo que están **estructurados.**
  + Analizar: Recuperar los datos, incluyendo condiciones lógicas, ordenaciones, agrupaciones, operaciones aritméticas.
* Información: datos analizados que sirven para tomar decisiones (interpretación).
* Conocimiento: comprensión de la información en base a la experiencia sobre una materia.
* Estructura sin bases de datos: Fichero donde se almacenan los datos <-> Programa <-> Interfaz. Muchos problemas (tipo de fichero, programa con acceso al fichero, soporte de accesos concurrentes, …).
* Estructura con bases de datos: BD <-> Programa huésped <-> Interfaz.
* Base de Datos: colección organizada de datos, que modela aspectos relevantes de la realidad y da soporte a procesos de información.
* SGBD (Sistema Gestor de Bases de Datos): Sistema software que gestiona los datos.
  + Gestión: Almacenamiento, extracción, modificación, borrado, …

A continuación, se hará mención de los distintos niveles de representación dentro de una BD:

**NIVEL LÓGICO(TABLAS)/CONCEPTUAL(ESQUEMAS)**

* Los datos se almacenan en tablas:
  + Fila = Instancia = Entidad = Registro.
  + Columna = Campo = Atributo.
  + Esquema de datos: Segunda línea de la tabla (cabecera). Indica el nombre de las columnas e identifica la tabla junto con el nombre de esta.

**NIVEL FÍSICO**

* Estructuras de datos en memoria secundaria.

**NIVEL ALMACENAMIENTO**

* Almacenamiento físico para la persistencia.
* Grupo repetitivo**:** Celda con un número indeterminado de campos.
* Inconsistencia: Información repetida e inconsistente. Ejemplo: Una misma fecha de nacimiento para una persona que está en 2 tablas, pero que tiene valores distintos.
* Integridad de la información: Datos mal introducidos o repetidos. En un momento dado, no se sabe que dato es correcto o incorrecto. Tipos:
  + De dominio (datos bien metidos en función del dominio de representación. Ej.: La temperatura de un sensor, que sea positiva siempre que se mida en grados centígrados).
  + De identificación. No puede haber 2 identificadores iguales para 2 instancias distintas.
  + Referencial (no la tenemos en cuenta en la asignatura). Que 2 tablas estén relacionadas por un identificador igual y correcto.
* Redundancia: Repetición de datos. Puede provocar inconsistencia.
* Dato erróneo: Un dato mal metido por el usuario que no afecta a la BD como sistema.
* Dato incorrecto: Lo contrario al anterior. Por ejemplo, una fecha de nacimiento superior a la actual.
* Niveles de abstracción:
  + Esquema externo -> vista o tabla que combina datos para una representación específica del usuario. Se define sobre el conceptual.
  + Esquema conceptual -> incluye todas las tablas/relaciones con información sobre entidades y relaciones.
  + Esquema interno -> esquema físico con los detalles de almacenamiento.

# Tema 2: Modelado de datos

**ETAPAS DEL DISEÑO**

* Análisis de requisitos: Necesidades del cliente, etapa muy costosa en tiempo y esfuerzo. Es una etapa crucial.
* Diseño conceptual: Descripción de alto nivel de los datos y sus restricciones. Modelo Entidad-Relación.
* Diseño lógico: Esquema de la BD acorde al SGBD elegido. Modelo relacional.
* Refinamiento de los esquemas: Reestructuración para garantizar propiedades importantes (normalización).
* Diseño físico: Mejora de rendimiento en base a cargas típicas (estructuras de datos).
* Diseño de aplicaciones y seguridad: Procesos relacionados con las aplicaciones. Tareas, flujos de trabajo, accesibilidad y seguridad.

**DISEÑO CONCEPTUAL (tablas)**

* Clave (identificador): Nombra de manera unívoca una entidad de la tabla. En UML -> nombre\_identificador : tipo {id}. Puede estar formado por varios atributos.
  + Superclave: exceso de atributos.
  + Existen claves alternativas dentro de la BD.
  + Debe ser un conjunto mínimo de atributos, para que no se de la superclave.
* ER: Descripción de como se estructuran los datos.
* Entidades: Objetos que engloban los datos de interés. Dentro de una tabla (tipo de entidad) son filas. Se consideran elementos de un conjunto. Se nombran en singular.
* Atributos(:dominios) y claves: Definición de datos de cada instancia.
* Relaciones y conjuntos de relaciones. Cardinalidades o multiplicidades: Asociación entre 2 o más entidades. Pueden tener atributos y/o clases asociativas/intermedias(débiles de forma implícita) -> Una clase intermedia puede usarse en lugar de una asociativa.
  + La cardinalidad [n]..m marca la mínima y máxima respectivamente, siendo la mínima opcional.
  + Existen relaciones reflexivas/recursivas, que relacionan una entidad consigo misma (Ej. Personas relacionadas con Personas).
* Entidades débiles: Se identifican considerando el id de otra entidad. Relación obligatoria con dependencia en existencia, y debilidad **SOLO en identificación.**
* Trampas de conexión: Fan traps y chasm traps.
* **Generalización/especialización:** Relaciones IS\_A. Entidades que tienen atributos más específicos que una Entidad que está por encima, también porque una de las entidades especializadas tiene relaciones distintas.
* Agregación y composición: Partes de algo. En la composición (rombo relleno de negro) coinciden la creación y la destrucción, en la agregación (rombo vacío), no.
* Restricciones: Condiciones de validez. Las BD no las implementan por el problema de satisfacibilidad, que es de complejidad exponencial (np completo).

# Tema 3: Modelo relacional

* Edgar F.Codd: Definición y consultas. Este modelo facilita las consultas y hace que la representación de datos sea más sencilla.
* Este modelo es una normalización del Entidad-Relación.
* DDL -> Data Definition Language (CREATE, INSERT, …)/ DML -> Data Manipulation Language. Estándar del ’92, aunque se usan cosas del ’99.
* BD Relacional: Conjunto de relaciones entre entidades(tablas). 2 partes en una relación.
  + Esquema: nombre de la relación, y nombre y tipo de cada columna (atributo, campo).
  + Instancia: valores de la tabla (registro, tupla); todas las tuplas son distintas.
* Creación de relaciones: Por cada entidad del modelo ER habrá una tabla en la BD (en relaciones \*-\* suelen surgir tablas adicionales).
  + CREATE TABLE nombreTabla(

Nombre\_atributo tipo,

Nombre\_atributo2 tipo

);

* Modificación y destrucción
  + ALTER TABLE nombreTabla ADD COLUMN nombreColumna TIPO.
  + DROP TABLE nombreTabla.
* Adición de tuplas
  + INSERT INTO nombreTabla(listaDeAtributos) VALUES (valoresDeAtributos).
* Borrado de tuplas
  + DELETE FROM nombreTabla nombreCorrelacion WHERE nombreCorrelacion.atributo=valor.
* Modificación de tuplas
  + UPDATE nombreTabla nombreCorrelacion SET nombreCorrelacion.atributo = nuevoValor WHERE nombreCorrelacion.atributo = valor.
* Restricciones de integridad(RI): De dominio, de clave primaria y de clave foránea. Una instancia es legal si satisface todas.
  + Son condiciones que tienen que cumplirse para cualquier instancia de la BD.
  + Se especifican al definir el esquema.
  + Se comprueban cuando se definen relaciones.
* Claves primarias y claves candidatas: identificación unívoca de tuplas con el menor número de campos posible (en caso de exceso, tenemos **superclave**). Sin repetición de valores ni permisión de nulos.
  + Claves candidatas se especifican con UNIQUE, y las primarias con PRIMARY KEY: CREATE TABLE nombreTabla(

nombreAtributo Tipo,

nombreAtributo2 Tipo,

UNIQUE (nombreAtributo2),

CONSTRAINT pk\_nombreTabla PRIMARY KEY (nombreAtributo)

)

* Claves foráneas, integridad referencial: Referencia a otra tupla, siendo el campo referenciado la clave primaria de esa otra tupla.
  + Integridad referencial: Imposición de restricciones referenciales. Ej: Sólo los estudiantes listados en Estudiantes son admitidos en la Matrícula de cursos.
    - CREATE TABLE nombreTabla(

nombreAtributo Tipo,

CONSTRAINT pk\_nombreTabla PRIMARY KEY (nombreAtributo),

CONSTRAINT fk\_nombreTabla PRIMARY KEY (nombreAtributoDeOtraTabla)

)

* + Se pueden llegar a permitir nulos. De haber un valor en una clave foránea, este debe apuntar a un valor existente de otra tabla.
  + **Cumplimiento de la integridad referencial**
    - Inserción con referencia no existente: NO ACTION. Rechaza la inserción de una tupla que referencia a otra no existente.
    - Borrado de una tupla referenciada: NO ACTION(rechaza borrado), ON CASCADE (borrado en cascada de todas las tuplas que referencian a la borrada), SET DEFAULT (mala opción), SET NULL.
    - Actualización de clave primaria: Mismo funcionamiento.
* Cardinalidades/Multiplicidades
  + 1-1: Cardinalidad máxima. Comprobamos si puede ponerse en términos de una única entidad (normalmente se arregla introduciendo un atributo).
  + 1-\* : Referenciamos en el lado \*, el valor del lado 1.
  + \*-\*: Se arregla con una entidad asociativa o una entidad intermedia.
* Entidades débiles: Necesitan identificarse con la FK de la entidad a la que referencian (la FK forma parte de la PK).
* Jerarquías ISA(herencia)
  + Participación: mandatory, optional(obligatorio u opcional. Un elemento debe ser obligatoriamente de una subclase o no).
  + Separación: disjoint, nondisjoint.(disjunto o no disjunto. Los elementos pueden pertenecer a sólo una subclase o a varias).
  + Criterios
    - Tablas para las subclases: diferencias entre ellas. Todo en una misma tabla puede dar lugar a muchos nulos.
    - Tabla para la superclase.

# Tema 6: Normalización

* Se basa es que hay que evitar futuros problemas que puedan surgir en la BD por la existencia de redundancia en los datos. Surge por el mantenimiento de la integridad de los datos.
  + Redundancia: Almacenamiento de la misma información en más de un sitio. Genera: Almacenamiento redundante, posibilidad de inconsistencia, anomalías (actualización, inserción, borrado).
  + Cuando se actualiza un dato con estas características, debe ser actualizado en todos los sitios donde tenga ocurrencia.
* Descomposición: Sustituir una relación por otras 2, o más, cada una con un subconjunto de campos y en conjunto, incluyendo todos los originales. Requisitos:
  + Reunión sin pérdida.
  + Conservación de las dependencias. **Las dependencias existentes son funcionales** (siempre que aparezca x debe aparecer y, porque es su valor asociado). Se entiende como implicaciones x -> y.
  + Consultas pueden obligar a la reunión de relaciones descompuestas
* Dependencia funcional: Generaliza el concepto de clave, con el cual, teóricamente, cualquier atributo depende de la clave, aunque esta no pueda repetirse. La clave debe ser mínima, si no, se considera **superclave.** Las condiciones de clave primaria son: sin nulos, sin repeticiones, mínima.
* **Formas normales:**
  + Basadas en dependencia funcional: 1NF, 2NF, 3NF, BCNF (las 2 últimas son importantes en el diseño de BD).
  + Basadas en otro tipo de dependencias:
    - Multivaluadas: 4NF
    - Reunión: 5NF
  + **BCNF**
    - De forma intuitiva, las únicas dependencias son aquellas en las que una clave determina algún atributo; no se pueden inferir valores.
    - Requiere que no existan dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata. Se dice que una tabla está en FNBC si y solo si está en 3FN y cada dependencia funcional no trivial tiene una clave candidata como determinante.
  + 3NF: Problema técnico con BCNF
    - Múltiples claves candidatas solapadas.
* Descomposición de una relación: Algoritmo de descomposición en BCNF
  + 1. R no en BCNF, X⊂R ,A atributo, X→A que provoca el no cumplimiento.  
    2. Descomponer en R-A y XA.  
    3. Si R-A o XA no en BCNF, aplicar recursivamente.

# Tema 7: Administración de Bases de Datos

* Tipos de usuarios distintos dentro de la BD
  + Usuario: Interactúa con la BD a través de aplicaciones.
  + Desarrollador (de aplicaciones):
    - Diseño y desarrollo de la aplicación, incluyendo la estructura de la BD.
    - Estimación de requisitos de almacenamiento, eficiencia, seguridad, etc.
  + Administrador de la BD (DBA):
    - Instala y actualiza el DBMS y las herramientas asociadas.
    - Establece y reserva el sistema de almacenamiento.
    - Crea los objetos primarios (tablas, vistas, índices), una vez que los desarrolladores han llevado a cabo el diseño.
    - Modifica la estructura de la BD, integrando la información de distintas aplicaciones.
    - Gestiona el acceso de usuarios y sus privilegios.
    - Establece políticas de seguridad y las monitoriza.
    - Monitoriza y optimiza el rendimiento de la BD.
    - Establece procedimientos de copias / recuperaciones.
    - Gestiona el soporte técnico con la distribuidora del DBMS.
* Tareas típicas: Instalar DBMS; Creación, arranque y parada de la BD; Gestión de usuarios; Implementación del diseño lógico; Optimización del rendimiento (ajuste, tuning).

# Tema 8: Técnicas avanzadas de gestión de la información

* Datos estructurados: A lo que estamos acostumbrados hasta ahora, bases de datos relacionales. Según el contexto, estas BD pueden no ser adecuadas, aunque este tipo de BD son las más estandarizadas, junto con SQL (existen lenguajes parecidos -> SQLike)
  + Esquema predefinido.
  + Consistencia estricta.
  + Escalado central.
  + No es flexible/versátil.
* **Big Data**: Se relaciona con el análisis de datos y la toma de decisiones en base a ellos (Business Intelligence).
  + Captura, integración, limpieza, anotación.
  + Almacenamiento, disponibilidad, acceso.
  + Análisis, visualización, modelos de conocimiento.
  + **Las 4 V’s:**
    - Volumen -> Escalibilidad de los datos.(Muchos datos).
    - Velocidad -> Se generan datos rápidamente.
    - Variedad -> Diferentes formas en los datos (Heterogéneos).
    - Veracidad -> Desconocimiento de los datos (Deben ser válidos).
    - ‘Valor’ -> Valor real de los datos.
* **NoSQL**:
  + Clave-valor: Datos como colección indexada de registros (diccionario). Modelos sencillos, acceso por clave.
  + Documentos: Datos semi-estructurados en formato documento. Atributos/cabeceras con grandes volúmenes de datos documentales.
  + Column: Tratamiento orientado a columna (tipo de datos). Gran rendimiento en agregaciones, datos derivados, business intelligence.
  + Graph: Entidades conectadas, aplicaciones con representación de conectividad, interacción, etc.
* **Web semántica**:
  + Contenido semántico, procesable por máquinas.
  + Web de los datos.
  + Componentes habituales:
    - XML, sintaxis básica de estructuración de contenidos
    - XML Schema, proporciona y restringe la estructura de contenidos
    - RDF, modelo de datos en forma de <sujeto, propiedad, objeto>; normalmente en sintaxis XML
    - RDF Schema, vocabularios de propiedades
    - OWL, ontologías / conocimiento
    - SPARCQL, lenguaje de consulta
  + Linked data. Reformulación de la web semántica.