

BASES DE DATOS Y SISTEMAS DE INFORMACIÓN (*BDA*)



TEMA 1

Bases de Datos Relacionales

DEFINICIONES

Sistema de gestión de bases de datos (SGBD): El **software** para hacer la base de datos.

Aplicaciones: Sirven para resolver problemas de cálculo, controlar procesos industriales, Manipular información de las organizaciones, distintas necesidades, persistencia de los datos...

Sistema de información

Un sistema de información (SI) es un **conjunto de elementos ordenadamente relacionados** según ciertas reglas y que **aporta la información necesaria** para cierto fin.

Puede ser perfectamente una libreta donde se lleven nombres de clientes, gastos... Si está soportado por un computador se le llama "informatizado".

Funciones: Recogida, Almacenamiento, Procesamiento, Recuperación... (De datos).

Partes: Contenido (datos), Equipo Físico, Equipo Lógico, Administrador, Usuarios...

Bases de Datos

Colección estructurada de datos, deben estar estructurado para poder ser manipulados. Estos tienen **Propiedades Estáticas y Dinámica**, Representas cosas de la vida real como condiciones que se deben cumplir o imposibilidades: Una persona no puede tener 2 DNIs...

Estáticas: Objetos, relaciones entre objetos, restricciones.

Dinámicas: Sucesos, restricciones. (No lo vamos a ver en el curso)

Sistema de gestión de bases de datos (SGBD)

Herramienta (software) para la **gestión** (creación y manipulación) de **bases de datos**.

Modelos y Estructuras de Datos: Los SGBS se basan en Modelos de Datos, estos proporcionan estructuras de datos predefinidas con sus operadores asociados.

Un modelo de datos es un conjunto de conceptos y reglas que permiten estructurar los datos resultantes de la observación de la realidad, de forma que queden representadas todas sus propiedades, tanto estáticas como dinámicas

Modelo Relacional: Hay varios modelos y familias pero el **modelo más importante** es el **Modelo relacional** que usa *estructuras de registro con relaciones*.



CARACTERÍSTICAS QUE HAN DE CUMPLIR LAS BASES DE DATOS

- **Integración de toda la información de la organización:** Todo tiene que ir por el mismo sitio y se tiene que usar para una parte importante de la organización, no vas a estar haciendo que cada uno entre a un sitio diferente. *Evita inconsistencias y + problemas.*
- **Persistencia de los datos:** Los datos deben estar disponibles en todo momento
- **Accesibilidad simultánea distintos usuarios:** Permitir concurrencia y lo que supone.
- **Unificación e independiente de los programas:** Descripción unificada de los datos e independiente de los programas. *Que el programa no me dice como monto la base, ya lo hago con el SGBD.*
Si modifico algo que no se está usando no debe afectar al programa y si yo modifico el lugar donde se guarde la base de dato y de cosas así, NO debe afectar al programa.
- **Definición de vistas parciales de los datos para distintos usuarios:** Si tengo toda la información en un sitio, cada usuario SOLO podrá ver lo que necesite → *Permisos.*
- **Mecanismos para controlar la integridad y la seguridad de los datos:** Que la base de datos cumpla sus requisitos, cosas de claves primarias y que funcione todo...

MODELO RELACIONAL DE DATOS

Modelo relacional: Presentación informal

Definición informal de una base de datos relacional: La información se **organiza en filas y columnas**.
Columna única propiedad con mismo tipo de datos, **filas** valores y datos.

Objetivos de una BD

Manipulación informal de una base de datos relacional: *Guardar la información*

- **Añadir nueva información:** **INSERTAR** filas.
- **Eliminar información:** **ELIMINAR** filas.
- **Modificar información:** **ACTUALIZAR** filas.

Consulta informal de una base de datos relacional: *Respuesta siempre en **forma de tabla**.* Se realizan las consultas sobre tablas para que nos devuelva una tabla SIN MODIFICAR LA BASE DE DATOS. Se usan operadores que pueden ser: **Conjuntistas, Auxiliar o relacionales.**

- **Conjuntistas:** **Unión, Intersección, Diferencia y Producto cartesiano**, comparaciones.
- **Auxilia "renombrar":** Este operador permite **cambiar el nombre de una columna**. La operación SOLO es para el resultado, **EN VERDAD NO SE CAMBIA**.
- **Relacional:** **Selección, Proyección y Concatenación**.
 - **Selección:** Sobre una tabla, devuelve las filas donde en x columna valor = y.
 - **Proyección []:** Devuelve una tabla solo con la columna SIN valores repetidos.
 - **Concatenación ⊗:** Sobre 2 tablas y, sobre las columnas **EN COMÚN**, busca valores iguales en las dos tablas, y los concatena: *Añade atributos de uno y del otro en la misma fila.*

Tienen orden: Paréntesis, Luego Operadores unarios y binarios (conjuntistas + concatenación). Si = de izq a der.

El modelo relacional: Presentación formal

Ahora usar los nombres como toca, que no queremos que nos peguen.

Tipos de datos: Conjunto de valores y operaciones que se puede hacer sobre ellos.

Término Informal	Término Formal
Tabla	Relación
Fila	Tupla
Columna	Atributo
Valores posibles	Tipo de datos

Tupla y Relación

Ahora en vez de con **filas/registros** trabajamos con **Tuplas** y en vez de con **Columnas** con **Atributos**. Cada **Atributo** tiene un **valor** y un **tipo asociado**, dicho valor será de ese tipo. Juntando varios de estos podemos hacer esquemas:

Esquema de relación y Relación: Un esquema de relación son y los **tipos de tupla que tenemos** (Cuantos atributos tiene y su tipo). El conjunto de estas tuplas crea **una relación**, lo que *entendemos como una tabla*.

- **Consultar el valor de una tupla:** “t.Ai o t(Ai)” es el valor de la tupla t en el atributo Ai (de la fila t columna Ai).

Propiedades de una relación

- **Grado de una relación:** Número de **atributos de su esquema**. *No debe de cambiar mucho.*
- **Cardinalidad de una relación:** Al número de **tuplas que la forman**. *Cambia constantemente.*

Notación: $R(A1: T1, A2: T2, \dots, An: Tn)$. Donde R es la relación, Ai es el atributo y Ti es el tipo de datos de Ai. Por comodidad se usan tablas con filas y columnas para visualizar los datos, pero el orden NO ES IMPORTANTE.

Orden: En las relaciones **el orden NO existe**, es irrelevante. No se puede decir “el tercer atributo” o “la última tupla”, no existe una ordenación entre los componentes por lo que **NO se puede trabajar con eso**.

Esquema Lógico/ relacional: Un esquema Lógico es El **conjunto de esquemas de relación** que representa un sistema de información (o sea las tablas, sus tuplas sus atributos y sus tipos). Si añadimos **los valores** (o extensiones) de las relaciones del esquema **en un instante determinado** constituyen la base de datos.

Es otra manera de definir bases de datos, estas son sus tablas (relaciones) y los valores de estas en un momento concreto.

Información faltante: valor nulo

Se supone que, dada una relación, todas sus tuplas, en cada atributo deben de tener un valor (de su tipo asociado). Pero si no sé conoce el valor de ese atributo se añade el “valor nulo”. **Representa la información desconocida**.

Nulo en todos los tipos de datos: Para poder implementar esta propiedad todos los tipos de valores han de incluir un valor nulo, se puede representar como (?). *Es solo una representación, el interrogante NO es un valor.*

Lógica revaluada: Ahora en vez de tener solo true o false, tenemos **Verdadero**, **Falso** e **Indefinido**. Si se usa alguno de los operandos de comparación y **alguno de los valores a comparar es nulo** el **resultado será “Indefinido”**.

Ahora algunas de las operaciones como la conjunción y disyunción funcionan un poco diferente con el indefinido. *Funcionan casi igual solo que si, el resultado depende de uno de los valores indefinidos para concretarse, el resultado será indefinido. Pero si con uno de ellos se puede determinar se queda igual. Ej: $false \wedge indefinido = false$.*

Comprobar si es Nulo: Como no tenemos valor con el que hacer “= null”, necesitamos el predicado “NULO()”.

- **Ejemplo:** **NULO(t.dni)** es = falseo si tiene un valor y = verdadero si se desconoce ese valor.

Restricciones de integridad

Las **restricciones representan una propiedad del mundo real**. Ya que la **base de datos representa el mundo real debe de cumplir dichas restricciones en todo momento**, esto quiere decir que **sus valores sean validos** en base esas restricciones. *De esto se encarga en sistema de gestión de la base de datos.*

Si una base de datos **cumple cierta Restricción de integridad** se dice que **la satisface**. Si no, se dice que **la Viola**. *Estas restricciones han de ser comprobadas después de cada actualización de la base de datos.*

Tipos de Restricciones de Integridad

- **Sobre atributos: De tipo de dato:** El tipo ha de ser el indicado para ese atributo.
- **Sobre atributos: De valor no nulo "VNN":** El valor se debe conocer.
- **De comprobación "COM":** Se cumple cierta propiedad en algún/os atributos (*un campo mayor que otro*).
- **De unicidad "UNI":** Ese valor **NO se puede repetir** (*Diferente de clave primaria*). No es = poner UNI{a,b} que UNI{a} y UNI{b}. Si es de forma conjunta, es que la combinación de los 2 no se repita, el otro es comprobar por separado que cada uno no se repita. *Si es UNI puede ser nulo, es un comodín para los 2 casos.*
- **De clave primaria "CP":** Identifica las tuplas, El atributo que sea CP cumple **UNICIDAD** y **VALOR NO NULO**.
- **De integridad referencial / Clave ajena "CAJ":** Dadas 2 relaciones R y S, en R el atributo Aj es o **Clave Primaria** o **cumple Unicidad** y S, aparece ese mismo atributo (mismo tipo) de tal forma que, **o NO tiene valor**, o si lo tiene, **esa clave ajena existe en la tabla original**.

Clave ajena general

Para que haya varias claves ajenas en una tabla, hace falta que todas ellas existan en otra tabla, sean claves primarias de la cual o cumplan unicidad y sean del mismo tipo. Si se cumple, puede haber 3 tipos de integridad referencial:

- **Débil:** Se habla por cada tupla de manera individual, de manera que se cumple si, para dicha tupla, al menos **un valor es nulo** o, si en caso de **NO haber nulos**, se puede **encontrar una tupla** en la otra tabla que tenga los **mismos valores para cada atributo**.
- **Parcial:** Igual que antes, por tuplas, si **no hay ningún nulo**, que haya una tupla en la otra table que tenga **mismos valores para cada atributo** y, en caso de ser **alguno nulo**, que el **resto de los valores coincidan** con los de alguna tupla. **Si es toda nula también** cumple.
- **Completa:** Como en las otras, tupla por tupla, se cumple si o **TODOS son nulos**, o **ninguno es nulo** y todos **coinciden con los valores** de alguna de las tuplas de la otra tabla. *No acepta nulos a medias de uno si uno no.*

Cuando se define una clave ajena en una relación, se debe especificar el tipo de integridad referencial que se exige, excepto en caso de que la clave ajena conste sólo de un atributo o cuando todos ellos sufran restricción de valor no nulo, ya que en estos dos casos los tres tipos de integridad referencial coinciden

Comprobación de restricciones: El SGBD se encarga de ver que, **tras cada modificación**, se cumplan las restricciones de integridad, en caso de no cumplirlas, no las acepta. Sin embargo, con la integridad referencial se puede ser algo más permisivo, esto son las **directrices**: *Aparecen cuando, habiendo claves ajenas, se incumple la RI.*

Si desde una **tabla "S"** hay registros que apuntan a otra **"R"**, **NUNCA se aceptarán cambios en "S" que violen la RI** (*insertar tuplas o modifica valores a otros erróneos*), pero si se **modifica "R"** se puede hacer un **borrado/modificado** en **cascada o a nulos**:

- **A nulos:** Donde antes estaba el valor que se ha modificado/borrado, **ahora se pone un nulo**.
- **Cascada:** Se borran/modifican (**se extiende la operación**) a aquellas tuplas que tuvieran ese valor.

Otras restricciones: Que solo afecten a un tributo o varios (no puede ser más grande que x, su valor solo puede ser entre uno de estos...) O combinaciones de estas (todo profe debe impartir docencia en al menos una asignatura).

Concepto de transacción

Si yo a mi base de datos, quiero añadir a una relación una nueva tupla, cabe la posibilidad de que, la operación de sea rechaza, ej: *Quiero añadir un profesor, pero como todos los profesores tienen que impartir docencia (y aún no he puesto esa tupla donde toca) me lo rechaza*. De igual forma, si quiero añadir la de docencia antes que la de profesor, la BD me lo va a rechazar, no podré añadir un profesor que aún no existe.

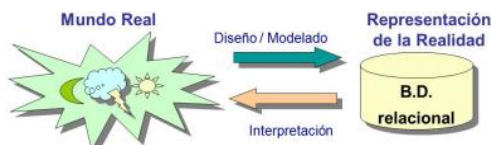
Transacciones: **Encapsulan varias operaciones simples**, de manera que se **hacen sin comprobarse por medio las restricciones de integridad**: Se hacen las 2 operaciones y al terminar se comprueba que todo cuadre, sino la rechaza.

INTERPRETACIÓN DE UNA BASE DE DATOS

INTRODUCCIÓN

Diseño de una base de datos: Proceso de obtención de un esquema relacional que represente las propiedades de interés del mundo real.

Interpretación de una base de datos: Una vez tenemos ya la base de datos (*esquema relacional*), llamamos Interpretación de una base de datos al **proceso de comprensión del mundo real que esta modela**. (inverso del diseño)



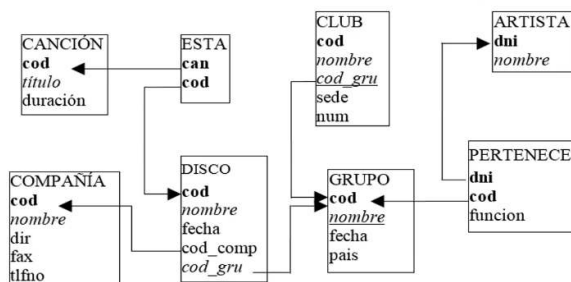
INTERPRETACIÓN DE UNA BASE DE DATOS RELACIONAL

Para comprender el mundo representado en una base de datos relacional, hay que identificar Objetos y relaciones:

- **Objetos:** De los que se quiere almacenar información incluyendo las propiedades de interés para cada uno.
- **Relaciones:** Relaciones entre los objetos y propiedades de las relaciones si es que existen.

Esquema relacional ejemplo

- **Negrita:** Clave primaria
- *Cursiva:* Valor NO nulo
- Subrayado: Unicidad
- Flecha →: Clave ajena de donde viene ese atributo



Identificación de objetos

Lo primero que habría que hacer es **buscar las relaciones** (tablas) que **su clave primaria no depende de ninguna clave ajena**: Estas son, las relaciones que no son como "ESTA", o "PERTENECE", tablas de muchos a muchos cuyas claves primarias son claves ajenas. Cada una de estas relaciones **representa un objeto** y cada **atributo** (no C.Ajena) es una propiedad de este objeto.

Identificación de relaciones

Ahora hay que fijarse en las claves ajenas de todas las relaciones. Podemos tener 2 casos según lo de antes:

- **En un objeto (1:M):** Se pueden representar **relaciones 1 a muchos** (1:M), ya que ese objeto solo puede tener una clave ajena de ese tipo, pero esa clave ajena puede estar en muchas tuplas de ese objeto: *En disco, solo puede tener un código de compañía, pero el código de esa compañía lo pueden tener muchos discos.*
- **En un objeto (1:1):** Se pueden representar **relaciones 1 a 1** (1:1), esto sucede cuando **una clave ajena cumple unicidad**, por lo que ese objeto solo tiene una de esas claves, pero es que solo ese objeto puede tener justo esa misma clave: *Un club solo puede tener un grupo, pero como ese código no se puede repetir, ese grupo no podrá estar en ningún otra club.*
- **En un NO objeto (M:M):** Se pueden representar **relaciones muchos a muchos** (M:M), esto ya que en esa relación, las claves ajenas se pueden repetir y combinar con otras claves ajenas: *En ESTA, tenemos que una canción pueda estar en muchos discos y que en mismo disco pueden estar muchas canciones.*

Interpretación de las restricciones: Podemos identificar relación con cosas como: Si la **CAj cumple unicidad** es una **relación 1:1**. Y si cumple **valor NO nulo** es que **se tiene que relacionar obligatoriamente con algo**.

