

# Mierda apuntes sacados de las mierda transparencias de *FBD*

Eduardo Espuch



Universitat d'Alacant  
Universidad de Alicante

## Resumen

Documento en el cual se tratara de explicar los conceptos relacionados con las bases de datos, conociendo el porque fue necesario su surgimiento y las herramientas y componentes que se deben de usar para mantener el SGBD (sistema de gestion de base de datos) operativo.

## Índice

<b>1. Historia de las bases de datos</b>	<b>2</b>
1.1. Inicios de la computacion, antes de las bases de datos . . . . .	2
1.2. Motivos para la aparicion de las bases de datos . . . . .	3
1.3. Introduccion a la base de datos . . . . .	3
1.4. Ampliacion sobre los ficheros . . . . .	4
<b>2. Modelo de datos</b>	<b>5</b>
2.1. Esquemas, proceso de analisis-diseño-implementacion . . . . .	6
2.2. Modelado, ampliacion . . . . .	6
2.3. Ampliando informacion . . . . .	7
2.3.1. Concepcion de las distintas claves . . . . .	7

# 1. Historia de las bases de datos

Veamos un poco el panorama que existia para que fuese necesario la aparicion de las bases de datos para asi, a la vez, explicar que son y de que requieren.

## 1.1. Inicios de la computacion, antes de las bases de datos

Lo primero que hubo para tener registro de valores fue el uso de entidades fisicas, la mano, el ábaco, ... el uso de la base 10 fue en parte consecuencia natural de no tener mas dedos con los que ir contando, pero bueno. ... hecho el chiste veamos lo que esta mas relacionado con la informatica.

En el siglo XIX surgieron la cinta perforada en telaras, la maquina analitica, el algebra de Boole(logica) y la maquina tabuladora, este ultimo invento siendo de Hollerith dando el salto a la alimentacion digital de datos. Se usaron tarjetas perforadas como almacenamiento para la recopilacion de datos(del censo estadounidense).

Aproximadamente a mitad del siglo XX surgen los primeros ordenadores electronicos, el ABC el primero en sí y el ENIAC como el primero en usar valvulas de vacio las cuales sustituyeron los relees, elemento mecanico cuya velocidad de trabajo venia condicionada por la velocidad del interruptor que realmente eran. De esto creo que basta con saber que su utilidad se restringia a calculos complejos ya que eran sistemas muy costosos de mantener ya que muchas piezas debian de sustituirse al ponerse de nuevo en funcionamiento.

A partir de esta fecha, los dispositivos de almacenamiento fueron apareciendo en el siguiente orden:

1. Las ya mencionadas tarjetas perforadas de Hollerith
2. Cinta magnetica, usada por primera vez en el Edvac.
3. Disco duro (HDD)
4. Floppy (FDD), discos magneticos flexibles, siendo el de 8 pulgadas el primero y posteriormente, de igual capacidad de almacenamiento (500kB), el  $5\frac{1}{4}$  pulgadas. Mas tarde surgio el de  $3\frac{1}{2}$  pulgadas, de 1.44MB y con otra estructura y composicion (se le denota con HD, con confundir con HDD).
5. Disco optico CD, DVD y BlueRay
6. Memoria de estado solido (SDD)

Este almacenamiento era necesario para guardar, entendiendo guardar como leer y escribir, datos. En este contexto, se trabajaba directamente sobre los ficheros, siendo importante la organizacion y el tipo de acceso a estos, diseñando ademas una estructura de registros adecuada sobre la que programaremos la navegacion por ella. Surgieron varios metodos para organizar los ficheros, cada uno mejor que otro para determinadas situaciones:

1. Organizacion secuencial para listados ordenados (los datos estan numerados y podemos acceder a un dato si sabemos su posicion)
2. Indexado para mayor flexibilidad con la organizacion (puedes ampliar la informacion si en el indice dejas indicado su posicion)
3. De forma directa cuando el campo clave lo permite (para informacion que ya es conocida? esta explicado raro)
- ...

Pero con el tiempo los programas se van haciendo mas complejos al igual que lo hacen los requisitos y necesidades de este. Si mezclamos el estado tecnologico de la epoca con las practicas de programacion a las que podriamos denotar como ineficientes entre otras cosas (mas cercanas al arte que a la ingenieria. . . , no se quien habra hecho estos apuntes pero me c. . . ) provocaron una serie de problemas o cuellos de botella que acabarian dando lugar a lo que conocemos como la crisis del software.

## 1.2. Motivos para la aparicion de las bases de datos

La base de datos, junto con el nacimiento de la ingenieria del software (que fue un jarro de agua fria para aquellos que programaban sin documentar y significo el comenzar de cero muchos programas, con sus consecuencias de tener que hacer mas trabajo) y otras cosas, surgio como parte de la solucion a la crisis del software. Existian, generalizando, tres grandes problemas en dicha crisis: redundancia, dependencia y seguridad.

1. Con la redundancia debemos de tener en cuenta el propio programa, si estaba documentado o entendible o funcional, y los datos, ya que al ir variando los programadores se puede dar el caso en el que, si no esta bien organizado, se dupliquen datos innecesarios. Con el mantenimiento del programa se nota lo importante que es el que sea facil de entender y no suponga muchas dificultades.
2. Similar a la redundancia de datos, la dependencia de los programas respecto a los datos era un problema, en especial de aquellos que no se usan. Al trabajar sobre los ficheros, los programas dependian de ellos y en el caso de modificarse los datos, debian de considerarse tal cambio en el programa, otra vez, manten tú el programa y luego me lloras.
3. Con la seguridad se tiene que tener en cuenta los accesos simultaneos, la recuperacion de ficheros y la autorizacion. Las SGBD fueron de especial ayuda para este apartado, veamos cada campo con mas detalle:
  - a) Accesos simultaneos: considera que acceden a un fichero varios usuarios y estos hacen modificaciones sobre el a la vez, el resultado seria que no se considera un orden de ejecucion y la ultima modificacion es la que se queda, cuando en realidades deberian encadenarse las operaciones.
  - b) Recuperacion de datos: tener a mano un plan b en el caso de fallo, un ejemplo claro seria el tener o no un seguro de casa.
  - c) Autorizacion: a quien dejo acceder a los ficheros, demostrar que es quien dice ser pero a la vez proteger su informacion privada del resto de usuarios.

## 1.3. Introduccion a la base de datos

Las bases de datos fueron una propuesta temporal que gusto y se fue desarrollando que aglutinaba todas las soluciones a los problemas vistos. Consistia en centralizar las definiciones de datos ofreciendo vistas parciales a los clientes e implementar mecanismos necesarios para garantizar la independencia, integridad y seguridad de datos. System R, creado por Codd y C.J. Date fue el primer SGBD.

Aporta unas aplicaciones que solo recuperan aquellos datos que le son estrictamente necesarios ya que es el SGBD el que se comunica con el sistema operativo y no al reves. La SGBD debiera de facilitarnos las tareas mas comunes, administracion, definicion de datos, consulta e interfaces, almacenamiento, acceso, . . .

La aparicion de las bases de datos independizo a los programas de la estructura de los ficheros de datos, ademas de centralizar la informacion impidiendo la redundancia e impidiendo el acceso a los ficheros totalmente o su modificacion sin la correcta identificacion.

Las BD trabajan con estructuras, un modelo de datos, datamodel, un modelo abstracto que organiza elementos de datos y estandariza el como se relacionan entre si y a propiedades de elementos

del mundo real. A este modelo, en especial la estandarización de relaciones, lo llamaremos modelo relacional de datos (MR), con SQL por ejemplo tratamos con tablas para relacionar datos.

Podemos considerar este tema como un aburrimiento, que no estariamos muy alejado de la realidad, o como una toma de contacto para todo lo que viene, ampliando la información de los que son las bases de datos y de como funcionan.

#### 1.4. Ampliación sobre los ficheros

Los ficheros es donde se almacena la información, hemos nombrado algunos dispositivos de almacenamiento anteriormente, ahora veamos con mas detalles ya que, para los ficheros, es tan importante tanto la parte física como la digital y como estos se organizan.

Respecto a la parte física tenemos lo que conocemos como almacenamiento secundario, es decir, un tipo de almacenamiento masivo y permanente (no volátil) con mayor capacidad de para almacenar datos e información que la memoria primaria, que si es volátil, pero de menor velocidad en comparación con la primaria. La memoria primaria seria la cache, las RAM, . . . como se ha dicho, memorias volátiles o con gran tendencia a cambiar de forma poco previsible y la memoria secundaria serian los discos magnéticos (rígidos o flexibles), memoria solida, . . . dispositivos externos a la cpu pero que cooperan con ella. Existe tambien la memoria terciaria que no es usual en sistemas de particulares (es un sistema automatizado en el que un brazo mecanico conecta o desconecta un medio de almacenamiento masivo fuera de linea segun la solicitud del computador) pero no lo tendremos en cuenta, solo saber que existe.

En la parte física, o Hardware, repasemos con detalle algunos dispositivos:

1. La cinta: almacenamiento masivo y secuencial, se debe recorrer toda la cinta (en ambos sentidos). Contenido principalmente de audio y video, modulador-demodulador de señal.
2. El disco: dividido en cilindro/cabezal/sector para facilitar el direccionamiento ya que se agrupaban sectores en bloques, siendo un bloque el tamaño mínimo que se transfiere a la RAM (tamaño definido por el SO). Era mas eficiente que la búsqueda secuencial ya que si queriamos acceder a un dato bastaba con localizar el bloque en el que se encuentra en lugar de recorrer la información anterior o posterior, segun la posición en la que se quedara.

Respecto a la parte digital o lo que sucede en el software, veremos como los ficheros se almacenan y que tipos hay:

1. Texto: registro unico o sin registro, no dispone de estructura y tiene la organización mas básica. Conocemos el principio y el final de este tipo de fichero y por lo tanto podemos desplazarnos por el desplazandonos una cantidad de bytes en un sentido.
2. El registro: al fichero le damos una estructura, en la cual se definen unos campos que toman un tipo de dato. El registro define a un objeto y los campos son sus características.

Estos campos deben de tener una manera de reservar un espacio en la memoria, podemos definir una longitud fija en la que cada campo reservara una cantidad exacta o una longitud variable, en la que el orden de los datos tendra importancia ya que, ya no se accedera por 'bloques' a los datos, si no de forma secuencial de forma que seran la cantidad de separadores entre los distintos datos de los campos. Puede hacerse mixta sin ningun problema, solo que vuelve a ser lo mismo, habra que leer el registro hasta obtener el dato que nos interesa en lugar de saltar a la posición de interes.

Como se ha comentado, los ficheros deben de localizarse, ya que tendremos que leerlos o escribir sobre ellos. Este proceso sigue estos pasos:

1. Un programa solicita al SO una operación sobre un fichero.

2. El SO busca en el almacenamiento el fichero y devuelve un bloque de datos a la RAM. Este bloque consiste en un conjunto de datos cercanos a la posición en la que se encuentran los datos que nos interesan ya que se suele acceder a posiciones cercanas, manteniendo el funcionamiento del almacenamiento secundario al mínimo.
3. En la RAM, el bloque de datos es manipulado para obtener el registro de interés para el programa. Es importante el efecto del espacio malgastado, bytes no ocupados, de ahí que hagamos hincapié ahora en cómo organizar correctamente un fichero.

Las organizaciones de un fichero vienen determinadas tanto por la tecnología disponible (el almacenamiento secundario es lento en comparación) y el rendimiento que queremos que tenga el fichero, es decir, nuestra propia decisión. Para ello, tenemos tres tipos de organización, la secuencial (leemos todos los datos hasta obtener el que queremos), aleatoria (reserva espacio para registros) e indexada (más flexible y en la cual existe un fichero de índice aparte para localizar más rápidamente el registro de interés). Veámoslos:

1. Secuencial: Desordenado (nuevos datos se añaden al final) u ordenado (se desplazarían los datos para introducir el nuevo donde corresponda).
2. Aleatoria: Directa (Actúa como la secuencial ordenada pero en lugar de desplazar, ha reservado previamente espacio para insertar nuevos registros) o hashing (más flexible que la directa pero exige mayor mantenimiento, permite almacenar más posiciones clave pero provoca colisiones en ocasiones, de ahí la necesidad del mantenimiento, también la posición se obtiene mediante una función dada).
3. Indexada: de eficiencia balanceada, con campos clave y no clave, distinguimos tres variedades según los índices. No densos si obliga al fichero que los campos clave estén ordenados, Agrupados donde existen varias campos clave que facilitan la búsqueda de un registro y Multinivel donde se busca ayudar al algoritmo de búsqueda binaria restringiéndola sucesivamente a menos datos.
4. Parecida a la indexada, los árboles B, B+ trabajan con índices multinivel dinámicos, con un algoritmo más complicado de mantener, las operaciones son más rápidas si consideramos una cantidad masiva de registros entre los que buscar. Muy usados en los SGBD actuales.

Toda esta organización no es de gran importancia para el usuario, ya que accede a dicha información y luego desaparece de su sistema, sino para el administrador, el cual debe de gestionar de forma eficiente dicha información y ser capaz de suministrarla rápidamente además de aprovechar la memoria disponible.

## 2. Modelo de datos

Los datamodel, o modelo de datos, deben de modelar el sistema, es decir, representar una realidad pero eliminando aquellos detalles que no son relevantes para el objetivo final. Hace un conjunto de cosas relacionadas ordenadamente entre sí para cumplir dicho objetivo.

Las propiedades del sistema de información podemos categorizarlas en:

- Estáticas: entidades, propiedades de las entidades y relaciones entre entidades.
- Dinámicas: operaciones sobre entidades, propiedades o relaciones entre operaciones.
- De integridad: sobre entidades y operaciones.

Todos los datamodel deberían de tener unas características comunes ya que todas aspiran a modelar un sistema de información con estos tipos de propiedades. Es más, la intención de modelar los sistemas de información es el poder mecanizarlos y así automatizar ciertas operaciones repetitivas y de gestión,

entre otras cosas. En realidad, los datamodel no son mas que lenguajes muy precisos y limitados a un problema muy concreto, de ahi que deba de tener características comunes los ditintos datamodel para poder entender el lenguaje usado.

Podriamos decir, para resumir el concepto de datamodel, que son la escritura que usamos para describir sistemas de informacion con unas 'palabras' y 'gramatica' definidas ya que el lenguaje usual puede ser muy ambiguo para representar un determinado concepto. Es un herramienta intelectual no tangible que nos permite estructurar los datos de forma que se capte la semantica de los mismos, siendo el resultado de aplicar un datamodel a un sistema de informacion lo que llamaremos esquema, siendo este la representacion del sistema de informacion.

## 2.1. Esquemas, proceso de analisis-diseño-implementacion

Con los esquemas nos introduciremos en el mundo del modelo relacional y la entidad-relacional y mas adelante los relacionaremos para aplicarlo en los SGBD. Para ello, veamos el proceso de analisis-diseño-implementacion.

1. Analisis: en la gestion de datos, el analisis nos dara un esquema conceptual, relacionaremos entidades entre si, una descripcion general de nuestro sistema de informacion independiente de si vamos a utilizar una maquina o no.
2. Diseño: trasladamos las ideas recogidas del esquema conceptual a un nuevo esquema logico, el cual si depende de la maquina y el software(la base de datos) que usaremos. El esquema logico lo encontraremos en modelo relacional, es decir, mediante el uso de tablas.
3. Implementacion: traducimos el esquema logico a un esquema fisico, es decir, lo ponemos en practica, dependera mas de como el propio motor de la base de datos escogido funciona que de el formato de los ficheros que almacenan los datos.

Mas adelante nos introduciremos con estos esquemas, en especial con el conceptual y el logico, ya que es de gran importancia el entenderlos y traducirlos entre ellos. Hay que entender que con el lenguaje, los esquemas representan un nivel de independencia de datos en el cual las tablas no tienen porque ser ficheros, sino una simple interpretacion del propio motor de la base de datos.

Es mas, el lenguaje de un datamodel actua como un lenguaje natural, con conceptos y reglas de composicion, como si fuese el lexico y gramatica, y se suelen distinguir el Lenguaje de Definicion de Datos (el LDD define el esquema) y el Lenguaje de Manipulacion de Datos(el LMD maneja los datos almacenados segun el esquema, te permite navegar por ellos o especificar uno en particularm entre otras cosas). Con esto, nos evitamos el querer conocer como el motor encuentra los datos que nos interesa, unicamente nos interesa realizar la operacion adecuada para un objetivo en particular.

Anteriormente hemos dicho que veriamos los esquemas conceptual y logico de forma ampliada pero, el fisico es el que nos permite poner en practica el resultado de estos dos, el esquema fisico seria los SGBD ya que un SGBD relacional es un sistema informatico capaz de manejar una BD relacional y esta, a su vez, son los datos estructurados en una BD conforme a un Modelo Relacion, es decir, el resultado de implementar un datamodel es obtener una SGBD funcional. Algunas SGBD estan orientadas a objetos, objetos-relaciones o son jerarquicas, pero nos fijaremos por ahora en datamodel estandares hasta que se diga lo contrario.

## 2.2. Modelado, ampliacion

Como se ha dicho al inicio del tema, los datamodel deben modelar el sistema pero, ¿como podemos modelar correctamente? Para modelar habia que saber escoger los datos de relevancia para nuestro sistema de informacion, estructurandose, almacenandose y recuperandose utilizando un data-model determinado, la informacion innecesaria se desecha. Para escoger los datos relevantes, o abstraer

dichos datos, se tienen en cuenta varios mecanismos, la clasificacion (definimos clases de objetos basicas), agregacion (se construyen clases de objetos complejas) y la generalizacion (se establecen tipos entre las posibles instancias de la clase), ademas de definir restricciones de dominio, identificacion y correspondencia entre clases.

La generalizacion o especializacion consiste en determinar que unos atributos son comunes a toda una clase de objetos que dentro de ella se pueden distinguir tipos o subconjuntos. De hecho, toda generalizacion deberia ir acompañada de sus propiedades de cobertura para saber exactamente como queremos que se comporte nuestra base de datos. Diremos que es total si todos los objetos pertenecen a una clase especializada y parcial si no todos pertenecen y, ademas, diremos que es disjunta o solapa segun si los conjuntos especializados son disjuntos o no. Es decir, todos los objetos de un campo pertenecen a una clase o si los objetos de una clase puede estar en varios campos a la vez.

La clasificacion la podemos distinguir entre primitivos (son simples sistemas de fichero convencional, como los ya estudiado anteriormente para ficheros) que no se consideran modelos de datos, los clasicos (que encontramos el jerarquico, el de red y el relacional), los semanticos (entidad-relacional, orientado a objetos... estos tipos de datos se les supone con mayor capacidad expresiva y mayor interpretacion conceptual) y otros que son con intereses muy particulares, en los cuales se trabaja para un area muy concreta y con una problematica especifica (como el Sistema de Informacion Geografica GIS entre otros).

Para nosotros, trabajaremos principalmente con el sistematico y el clasico, en particular el de entidad-relacion y el modelo relacional.

## 2.3. Ampliando informacion

En las transparencias vienen desordenadas pero vamos a tratar de agruparlo para que sea mas comodo de entender. En este apartado veremos con mas lujo de detalle los siguientes conceptos, introduciendo algunos de los que usaremos en los siguientes temas, los cuales son:

### 2.3.1. Concepcion de las distintas claves

Con la abstraccion hemos hablado de los conceptos de clasificacion, agregacion y generalizacion ademas de definir unas restricciones y, aunque es en estos conceptos en los que nos centraremos en este apartado, debemos entender que son las claves, tanto las candidatas como las ajenas, y el valor nulo siendo estas conceptos relacionados con las restricciones sobre todo. Veamoslo poco a poco:

1. Denotado por NULL/NULO, el valor nulo significa la ausencia de valor a lo groso modo, desconocemos si existe valor para tal campo y si lo hay no sabemos cual es. Si sabemos que un campo no puede ser nulo, es decir, debe haber al menos un valor (dato importante para cardinalidad) se le denota con VNN (Valor No Nulo).
2. Claves candidatas: atributos que diferencian unas tuplas de otras, no admite nulos ni duplicados, es decir, si tenemos una clave candidata debe de existir y ser unica, forzando a la tabla a comportarse como una relacion. Este atributo o conjunto de atributos unicos para una entidad es decision del administrador, no tiene porque coincidir con la realidad.

INCISO: la forma en la que definen matematicamente una clave candidata carece de logica alguna... Vamos a intentar explicarlo mejor, dada una tupla (sucesion de elementos) tal que  $X = \{A_a, B_b, C_c, D_d\} = \{A_a : D_d\}$ , que entenderemos como posibles combinaciones que se puedan formar con la informacion, diremos que, dada un relacion R que contiene todas las combinaciones, X incluida en R sera CC si esta X dada define tuplas con valores distintos para cada tupla y es irreducible, por ejemplo, si decimos que  $\{A, B\}$  es CC, ni A ni B por si solas podrian ser CC (ya que no es irreducible) pero el resto de combinaciones posibles si podrian serlo salvo que contenga a  $\{A, B\}$ . Puede ser algo confuso pero la idea reside en que busques distintas formas de relacionar dos entidades por algun atributo comun.

Podemos distinguir dos claves candidatas:

- a)* Clave primaria: se define siempre, actua como identificador principal, es el atributo que diferencia cada entidad. Siempre existe.
- b)* Clave alternativa: se define aveces, cuando se dan varias CC y la alternativa no coincide con la primaria. Se puede utilizar para mayor restriccion en la correspondencia entre clases (mas atributos identificadores con los que comparar) o por su mera existencia.

Ningun tipo de CC puede contener Nulo, por lo tanto todas las CC seran VNN

- 3. Clave ajena: Conjunto de atributos que hace referencia a la clave primaria de otra relacion (o a ella misma), esta si admite nulos y duplicados. Es decir, o hace referencia a un valor nulo o hace referencia a un valor almacenado en alguna clave primaria y lo puede hacer varias veces.