

# UNIDAD DIDÁCTICA 1

## LOS SISTEMAS DE ALMACENAMIENTO DE LA INFORMACIÓN

### Contenido

1. Ficheros.....	2
1.1. Tipos de Ficheros y Formatos .....	2
1.2. Ficheros de texto .....	3
1.3. Ficheros binarios .....	3
2. Bases de Datos .....	4
2.1. Terminología .....	5
2.2. Estructura de una base de datos.....	7
2.3. Usos de las Bases de Datos .....	9
2.4. Evolución y Tipos de Bases de Datos.....	10
3. SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD) .....	13
3.1 Concepto de Sistema de Gestión de Bases de Datos .....	13
3.2 Funciones del SGBD .....	14
3.3. El lenguaje SQL .....	15
3.4. Tipos de SGBD .....	16

### OBJETIVOS

- Analizar los sistemas lógicos de almacenamiento y sus características.
- Identificar los distintos tipos de bases de datos.
- Evaluar la utilidad de un sistema gestor de bases de datos
- Reconocer la función de cada uno de los elementos de un sistema gestor de bases de datos
- Clasificar los sistemas gestores de bases de datos

# 1. Ficheros

Un ordenador almacena muchos tipos de información, desde datos administrativos, contables o bancarios hasta música, películas, partidas de vídeo juegos, páginas web, etc. Toda esta información está almacenada en los dispositivos de almacenamiento del ordenador, es decir, discos duros, DVD, pen drives, etc. Para poder organizar la información en estos dispositivos se utilizan los ficheros o archivos.

Los ficheros son estructuras de información que crean los sistemas operativos de los ordenadores para poder almacenar datos. Suelen tener un nombre y una extensión, que indica el formato de la información que contiene.

## 1.1. Tipos de Ficheros y Formatos

El formato y el tipo de fichero determina la forma de interpretar la información que contiene, ya que lo único que se almacena en un fichero es una colección de bits (ceros y unos), de forma que es necesario interpretarlos para dar sentido a la información almacenada.

Por ejemplo, para almacenar una imagen en un ordenador, podemos usar un fichero *imagen.bmp*, que almacena un vector de datos con los colores que tiene cada *pixel* que forma la imagen. Además, la imagen tiene una paleta de colores y unas dimensiones, información necesaria que también hay que almacenar en el fichero. Todos estos datos se ordenan según un formato, y el sistema operativo, o la utilidad que trate los gráficos, debe conocer este formato para poder extraer los píxeles y mostrarlos por pantalla en la forma y dimensiones correctas. Si abrimos el gráfico con el Bloc de Notas, que solo sabe interpretar texto, el resultado será ilegible.

Los ficheros se clasifican por diversos criterios, según su contenido (texto o binario), según su organización (secuencial, directa, indexada) o según su utilidad (maestros, históricos, movimientos).

El **contenido** de un fichero puede ser tratado como texto, o como datos binarios, es decir, los bits almacenados en un fichero pueden ser traducidos por el sistema operativo a caracteres alfabéticos y números que entendemos los seres humanos, o pueden ser tratados como componentes de estructuras de datos más complejas, como ficheros que almacenan sonido, vídeo, imágenes, etc.

La **organización** de un fichero implica la forma en que podemos acceder a los datos. Por ejemplo, los datos de un fichero con organización **secuencial**, están organizados en una secuencia ordenada unos detrás de otros. Se caracterizan por tener que recorrer todos los datos anteriores para llegar a uno en concreto. Los ficheros de organización **directa**, permiten acceder a un dato en concreto sin necesidad de acceder a todos los anteriores. Los de organización **indexada** acceden a la información consultando un índice, es decir, una estructura de datos que permite acceder a la información rápidamente, de forma parecida al índice de un libro. Existen variantes de las organizaciones anteriores que permiten mezclar las características de unas y otras.

La **utilidad** de un fichero indica el uso que se va a hacer con él, por ejemplo, puede contener datos fundamentales para una organización como los datos de los clientes, que se almacenan en un fichero principal llamada **maestro**. Si hay variaciones (altas, modificaciones y bajas de clientes) en los ficheros maestros, se almacenan en los ficheros de **movimientos** que más tarde se comparan con los maestros para añadir las modificaciones. Cuando existen datos que ya no son necesarios para su proceso diario pasan a formar parte de los ficheros **históricos**.

Actualmente, los ficheros de organización y utilidad han quedado en desuso. Desde la aparición de las bases de datos modernas, ya no se clasifican según su organización y utilidad.

Actualmente los sistemas operativos tratan los ficheros desde dos puntos de vista:

- Según su contenido (texto o datos binarios)
- Según su tipo (imágenes, ejecutables, vídeos, etc.)

## 1.2. Ficheros de texto

Los ficheros de texto se llaman también ficheros planos o ficheros **ascii** (*American Standard Code for Information Interchange* = Código Estándar Americano para Intercambio de la Información).

ASCII es un código estándar que asigna un valor numérico a cada carácter, por lo que se puede representar los documentos de Texto Plano, es decir, los que son directamente legibles por seres humanos.

La asignación de valores numéricos a caracteres viene en una tabla de códigos Ascii, se caracteriza por utilizar 1 byte para la representación de cada carácter. Con  $x$  bits se pueden representar  $2^x$  combinaciones distintas de caracteres, y como 1 byte = 8 bits, existe  $2^8 = 256$  caracteres en la tabla de códigos Ascii, numerados del 0 al 255.

Los ficheros de texto suelen tener extensiones para conocer el tipo de texto que se guarda en el fichero, por ejemplo:

- **Ficheros de configuración:** son ficheros cuyo contenido es texto sobre configuraciones del sistema operativo o de alguna aplicación. Pueden tener las extensiones .ini, .inf, .conf.
- **Ficheros de código fuente:** su contenido es texto con programas informáticos. Ejemplos: .c, .cpp, .java, .sql
- **Ficheros de páginas web:** las páginas son ficheros de texto con hipertexto<sup>1</sup> que interpreta el navegador. Ejemplo: .html, .php, .css, .xml
- **Formatos enriquecidos:** son textos que contienen código de control para ofrecer una visión del texto más elegante. Ejemplo: .rtf, .ps, .tex

## 1.3. Ficheros binarios

Los ficheros binarios son todos los que no son de texto, y necesitan un formato para ser interpretados. Ejemplo de ficheros binarios:

- **De imagen:** .jpg, .gif, .tiff, .bmp, .wmf, .png, .pcx, etc.
- **De vídeo:** .mpg, .mov, .avi, .qt
- **Comprimidos o empaquetados:** .zip, .Z, .tar, .rar, .lhz
- **Ejecutables o compilados:** .ex, .com, .cgi, .o, .a
- **Procesadores de texto:** .doc, .docx, .odt

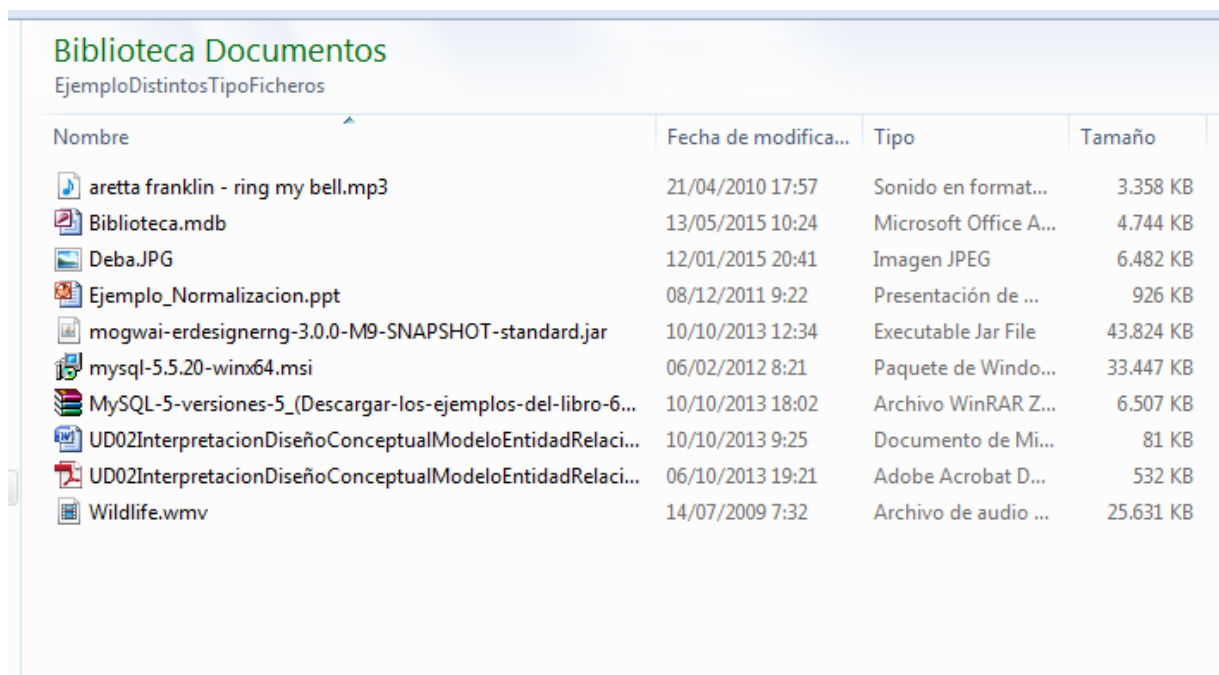
Generalmente los ficheros que componen una base de datos son de tipo binario, ya que la información que hay almacenada en ellos debe tener una estructura lógica y organizada para que las aplicaciones puedan acceder a ellas de manera universal, es decir, siguiendo un estándar. Esta

---

<sup>1</sup> El hipertexto es una forma de escritura, no secuencial, con bifurcaciones, que permite que el lector elija qué secuencia seguir y se muestra en una pantalla interactiva para facilitar la navegación.

estructura lógica es muy difícil de expresar en ficheros de texto, por ello, la información de una base de datos se suele guardar en uno o en varios ficheros:

- La base de datos Oracle guarda la información en múltiples ficheros llamados “datafiles”, “tempfiles”, “logfiles” etc.
- Un gestor de MySQL guarda su información en 3 ficheros de datos binarios con extensión .frm, .myd, .myi
- Access guarda la información en un único fichero con extensión .mdb



Nombre	Fecha de modifica...	Tipo	Tamaño
aretta franklin - ring my bell.mp3	21/04/2010 17:57	Sonido en format...	3.358 KB
Biblioteca.mdb	13/05/2015 10:24	Microsoft Office A...	4.744 KB
Deba.JPG	12/01/2015 20:41	Imagen JPEG	6.482 KB
Ejemplo_Normalizacion.ppt	08/12/2011 9:22	Presentación de ...	926 KB
mogwai-erdesignerng-3.0.0-M9-SNAPSHOT-standard.jar	10/10/2013 12:34	Executable Jar File	43.824 KB
mysql-5.5.20-winx64.msi	06/02/2012 8:21	Paquete de Windo...	33.447 KB
MySQL-5-versiones-5_(Descargar-los-ejemplos-del-libro-6...	10/10/2013 18:02	Archivo WinRAR Z...	6.507 KB
UD02InterpretacionDiseñoConceptualModeloEntidadRelaci...	10/10/2013 9:25	Documento de Mi...	81 KB
UD02InterpretacionDiseñoConceptualModeloEntidadRelaci...	06/10/2013 19:21	Adobe Acrobat D...	532 KB
Wildlife.wmv	14/07/2009 7:32	Archivo de audio ...	25.631 KB

Ilustración 1 Distintos tipos de Fichero

## 2. Bases de Datos

Una base de datos es una colección de información perteneciente a un mismo contexto (o problema), que está almacenada de forma organizada en ficheros.

Una base de datos está organizada en **tablas**<sup>2</sup>, que almacenan información concerniente a algún objeto o suceso. Estas tablas se relacionan formando vínculos o relaciones entre ellas, que ayudan a mantener la información de los diversos objetos de forma ordenada y coherente.

Cada una de estas tablas es una estructura que se parece a las hojas de cálculo, ya que se presenta en forma de **filas** y **columnas**. Cada fila almacena un **registro** con tantos **campos** o columnas como tenga la tabla.

Por ejemplo, tenemos una tabla Empleados donde cada fila o registro contiene la información de un empleado de la empresa y cada columna o campo representa un trozo de información sobre cada empleado, por ejemplo el nombre o el salario, etc.

---

<sup>2</sup> Las tablas en las bases de datos relacionales se llaman relaciones base.

Em_codigo	Em_codigod	Em_extelf	Em_fecnac	Em_fecing	Em_salario	Em_comisio	Em_nhijos	Em_nombre
110	121	350	10/11/1980	01/03/2005	2945		3	Pons Gutierrez, Cesar
120	112	840	15/02/1956	08/11/1983	1900	1	1	Lasa Olalde, Mario
130	112	810	09/09/1950	03/10/1980	2755	1100	0	Perez Arrieta, Julio
150	121	340	01/05/1970	18/07/1988	4180		0	Terol Gomez, Luciano
160	111	740	11/04/1946	25/06/1976	2945	1100	2	Perez Sanchez, Marcos
180	110	500	18/10/1976	18/03/1996	4560	500	2	Aguirre Berraondo, Aureo
190	121	350	05/12/1945	02/11/1977	2850		4	Veiga Lopez, Juliana
210	100	200	20/09/1965	22/01/1980	3610		2	Galvez Rordiguez, Pilar
240	111	760	26/02/1960	24/02/2001	2660	1000	3	Sanz Guisasaol, Lavinia
290	120	910	30/11/1955	14/02/1988	2565		3	Gil Bueno, Gloria
310	123	480	21/11/1961	15/01/1983	3990		0	Garcia Barreras, Augusto
320	130	620	25/12/1957	05/02/1978	3847,5		2	Sanz Andonegui, Cornelio
330	112	850	19/08/1952	01/03/1972	2660	900	0	Diaz Ochoa, Amelia
350	130	610	13/04/1977	10/05/2000	4275		1	Camps Cabrera, Aurelio
360	111	750	28/10/1985	10/10/2006	2375	1000	2	Lara Perez, Dorinda
370	121	360	22/06/1983	05/04/2007	1900		1	Ruiz Ormaechea, Fabiola
380	112	800	30/03/1968	01/01/1988	1710		0	Martin Azpiazu, Micaela
390	110	500	19/02/1966	08/10/1986	2150		1	Moran Pombo, Carmen
400	111	780	18/08/1969	01/11/1987	1757,5		0	Lara Peraz, Lucrecia
410	130	660	14/07/1968	13/10/1988	1750		0	Muguerza Muguruza, Azucena

Ilustración 2 Ejemplo de tabla Microsoft Access

## 2.1. Terminología

### Dato

El dato es un trozo de información concreta sobre algún concepto o suceso.

Un dato es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Los datos describen hechos empíricos, sucesos y entidades. Es un valor o referente que recibe el computador por diferentes medios, los datos representan la información que el programador manipula en la construcción de una solución o en el desarrollo de un algoritmo.

Por ejemplo: Beatriz es texto que representa el nombre de una persona. 25/02/1990 es una fecha que representa la fecha de nacimiento de una persona. 2.500 € es un número que representa el sueldo de una persona.

Los datos se caracterizan por pertenecer a un tipo.

### Tipo de Dato

El tipo de dato indica la naturaleza del campo. Pueden ser **numéricos**, son los datos con los que se pueden realizar cálculos aritméticos (sumas, restas, etc.) y **alfanuméricos**, son los que contienen caracteres alfabéticos y dígitos numéricos (letras, dígitos, signos, etc.). Los tipos de datos, alfanuméricos y numéricos, se pueden combinar para obtener tipos de datos más complejos. Por ejemplo, el tipo de datos Fecha contiene tres datos numéricos, el día, el mes y el año.

### Campo

Un campo es un identificador para toda una familia de datos. Cada campo pertenece a un tipo de dato. Por ejemplo el campo *em\_nombre* representa todos los nombres de las personas que hay en la tabla, el campo *em\_fecnac* representa todas las fechas de nacimiento de los empleados que hay en la tabla. *em\_nombre* es del tipo de dato alfanumérico y *em\_fecnac* es del tipo de dato Fecha. A los campos también se les llama **columnas**.

## **Registro**

Es una colección de datos referentes a un mismo concepto o suceso. Por ejemplo, los datos de un empleado pueden ser su DNI, nombre, fecha de nacimiento, dirección, sueldo, etc. A los registros también se les llama **tuplas** o **filas**.

## **Campo clave**

Es un campo especial que identifica de forma unívoca a cada registro. Por ejemplo, el DNI es único para cada persona, por lo tanto puede ser el campo clave.

## **Tabla**

Es un conjunto de registros agrupados con un mismo nombre, que representa el conjunto de todos ellos. Por ejemplo, todos los empleados de una empresa se almacenan en una tabla cuyo nombre es Empleados.

## **Consulta**

Es una instrucción para hacer peticiones a una base de datos. Puede ser una búsqueda simple de un registro específico o una solicitud para seleccionar todos los registros que satisfagan un conjunto de criterios. También puede ser eliminaciones, inserciones o modificaciones de registros. Este tipo de consultas alteran los valores de las tablas.

## **Índice**

Es una estructura que almacena los campos clave de una tabla, organizándolos para hacer más fácil encontrar y ordenar los registros de esa tabla. El índice funciona de forma similar al índice de un libro, guardando parejas de elementos, el elemento que se desea indexar y su posición en la base de datos. Para buscar un elemento que esté indexado solo hay que buscar en el índice de dicho elemento para, una vez encontrado, devolver el registro que se encuentre en la posición marcada por el índice.

## **Vista**

Es una transformación que se hace a una o más tablas para obtener una nueva tabla. Esta nueva tabla es una tabla **virtual**, es decir, no está almacenada en los dispositivos de almacenamiento del ordenador, aunque sí se almacena su definición.

## **Informe**

Es un listado ordenado de los campos y registros seleccionados en un formato fácil de leer. Generalmente se usan como peticiones expresas de un tipo de información por parte de un usuario. Por ejemplo, un informe a las facturas impagadas del mes de septiembre ordenado por nombre del cliente.

## **Guiones o scripts**

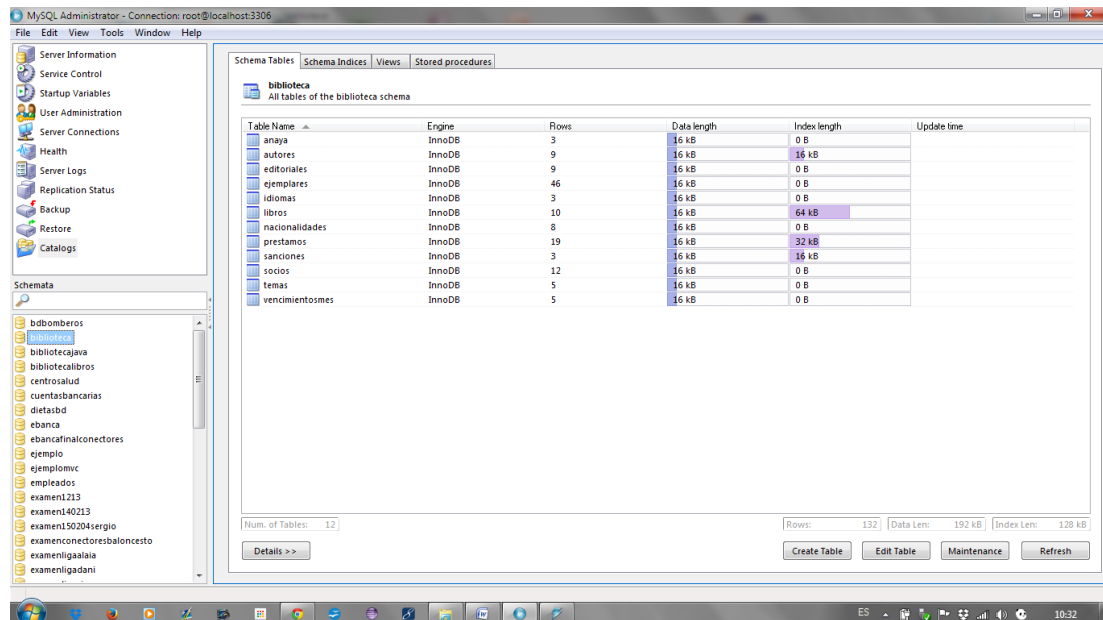
Son un conjunto de instrucciones, que ejecutadas de forma ordenada, realizan operaciones avanzadas de mantenimiento de los datos almacenados en la base de datos.

## **Procedimientos**

Son un tipo especial de *scripts* que está almacenado en la base de datos y que forma parte de su esquema.

## 2.2. Estructura de una base de datos

Una base de datos almacena los datos a través de su **esquema** (schema). El esquema es la definición de la estructura donde se almacenan los datos, contiene todo lo necesario para organizar la información mediante tablas, registros (filas) y campos (columnas). También puede contener otros objetos necesarios para el tratamiento de los datos (procedimientos, vistas, índices, etc.). Al esquema también se le suele llamar *metainformación*, es decir, información sobre la información o *metadatos*.



MySQL Administrator - Connection: root@localhost:3306

Schema Tables Schema Indices Views Stored procedures

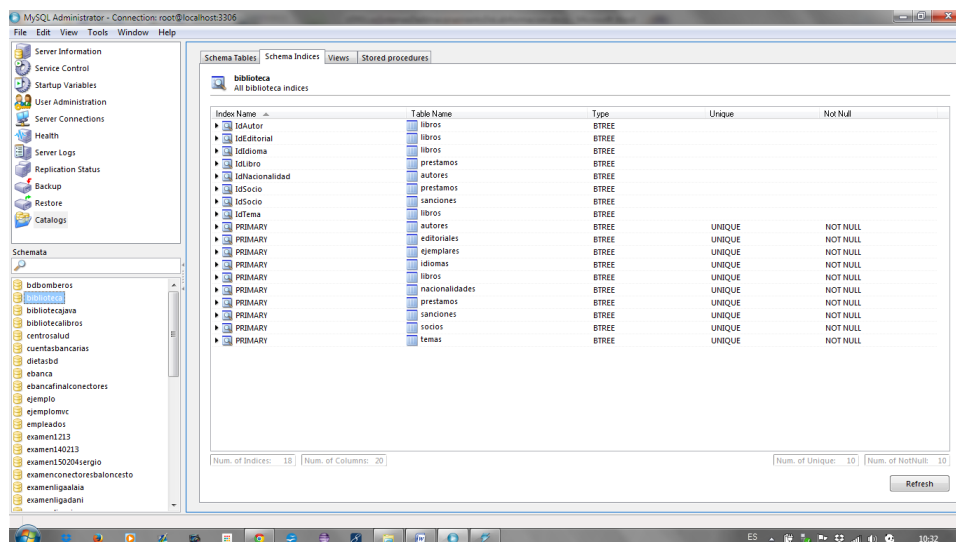
**biblioteca**  
All tables of the biblioteca schema

Table Name	Engine	Rows	Data length	Index length	Update time
anaya	InnoDB	3	16 kB	0 B	
autores	InnoDB	9	16 kB	16 kB	
editoriales	InnoDB	9	16 kB	0 B	
ejemplares	InnoDB	46	16 kB	0 B	
idiomas	InnoDB	3	16 kB	0 B	
libros	InnoDB	10	16 kB	64 kB	
nacionalidades	InnoDB	8	16 kB	0 B	
prestamos	InnoDB	19	16 kB	32 kB	
sanciones	InnoDB	3	16 kB	16 kB	
socios	InnoDB	12	16 kB	0 B	
temas	InnoDB	5	16 kB	0 B	
vencimientos	InnoDB	5	16 kB	0 B	

Num. of Tables: 12 Rows: 132 Data Len: 192 kB Index Len: 128 kB

Details >> Create Table Edit Table Maintenance Refresh

Ilustración 3 Esquema de las Tablas de la BBDD Biblioteca



MySQL Administrator - Connection: root@localhost:3306

Schema Tables Schema Indices Views Stored procedures

**biblioteca**  
All biblioteca indices

Index Name	Table Name	Type	Unique	Not Null
IdAutor	libros	BTREE		
IdEditorial	libros	BTREE		
IdIdioma	libros	BTREE		
IdLibro	prestamos	BTREE		
IdNacionalidad	autores	BTREE		
IdSocio	prestamos	BTREE		
IdSocio	sanciones	BTREE		
IdTemas	libros	BTREE		
PRIMARY	autores	BTREE	UNIQUE	NOT NULL
PRIMARY	editoriales	BTREE	UNIQUE	NOT NULL
PRIMARY	ejemplares	BTREE	UNIQUE	NOT NULL
PRIMARY	idiomas	BTREE	UNIQUE	NOT NULL
PRIMARY	libros	BTREE	UNIQUE	NOT NULL
PRIMARY	nacionalidades	BTREE	UNIQUE	NOT NULL
PRIMARY	prestamos	BTREE	UNIQUE	NOT NULL
PRIMARY	sanciones	BTREE	UNIQUE	NOT NULL
PRIMARY	socios	BTREE	UNIQUE	NOT NULL
PRIMARY	temas	BTREE	UNIQUE	NOT NULL

Num. of Indices: 18 Num. of Columns: 20 Num. of Unique: 10 Num. of NotNull: 10

Refresh

Ilustración 4 Esquema de los índices de la BBDD Biblioteca

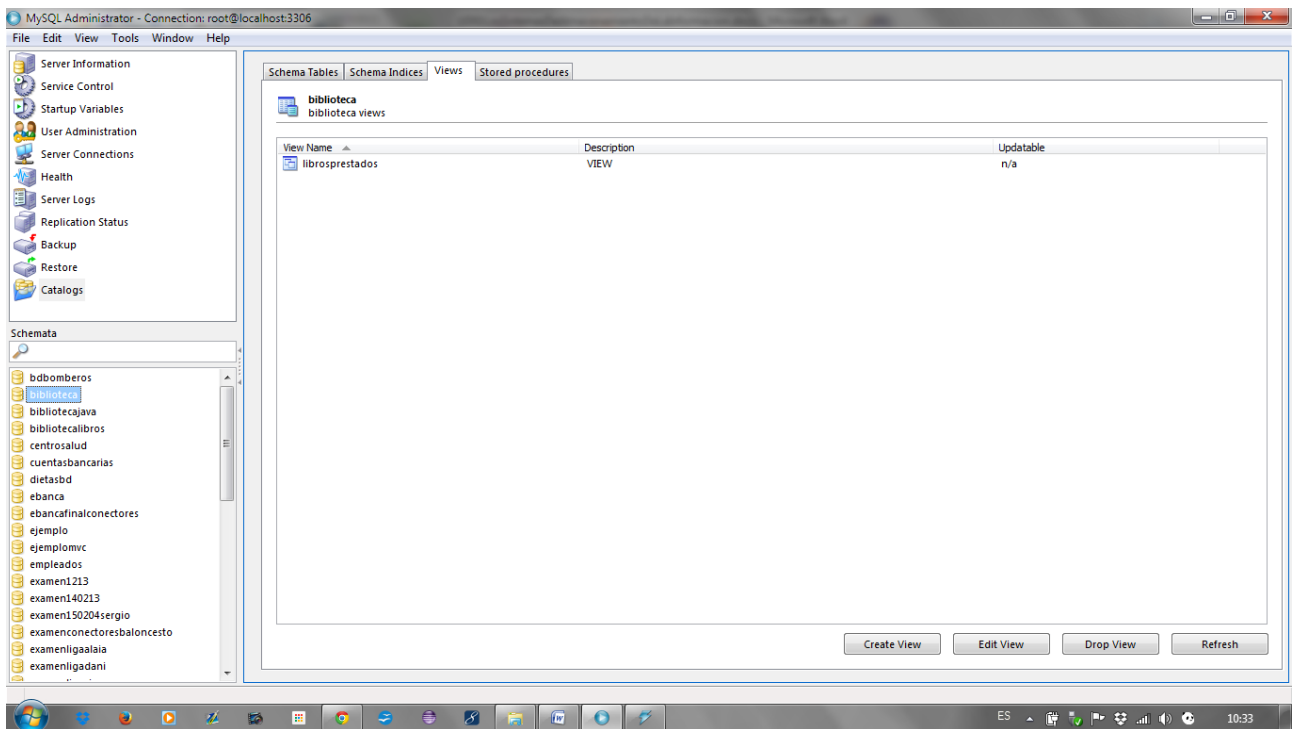


Ilustración 5 Esquema de las Vistas de la BBDD Biblioteca

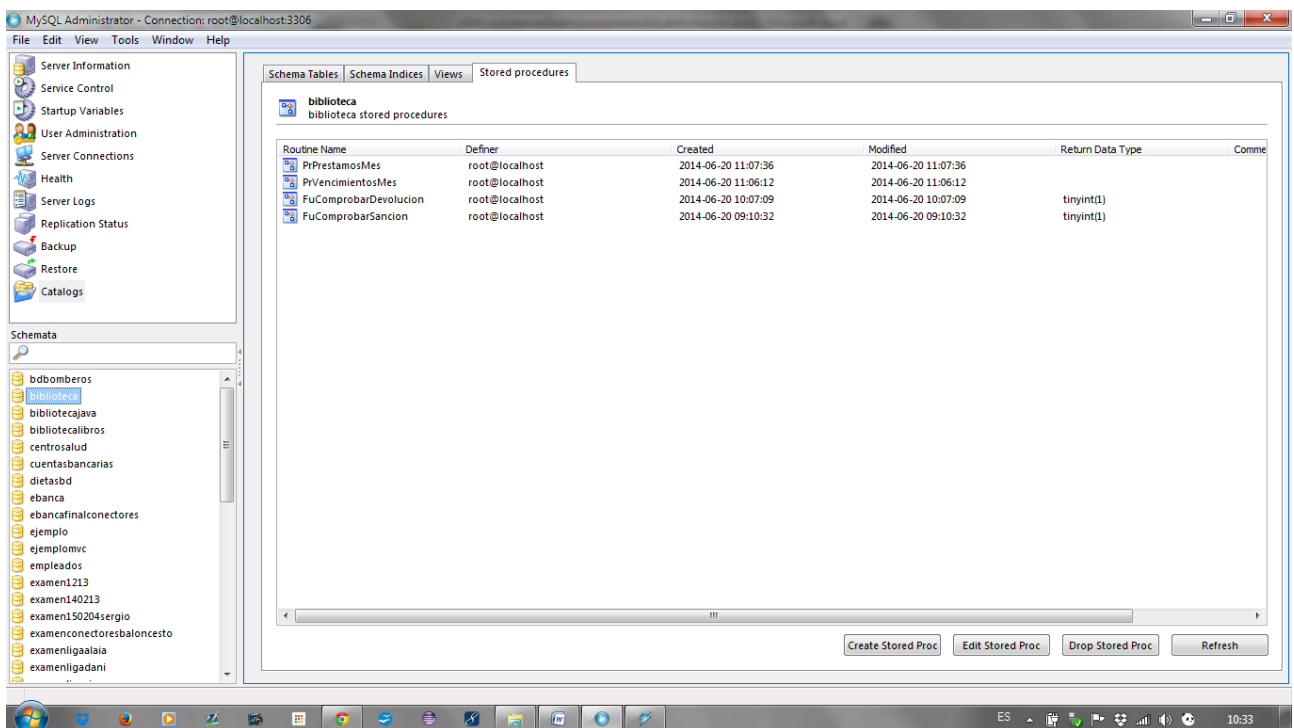


Ilustración 6 Esquema de los Procedimientos de la BBDD Biblioteca

Los gestores de las BBDD actuales, Oracle, MySQL, etc., almacenan los esquemas de las BBDD en tablas, de tal manera que el propio esquema de la BBDD se puede tratar como si fueran datos comunes de la BBDD.



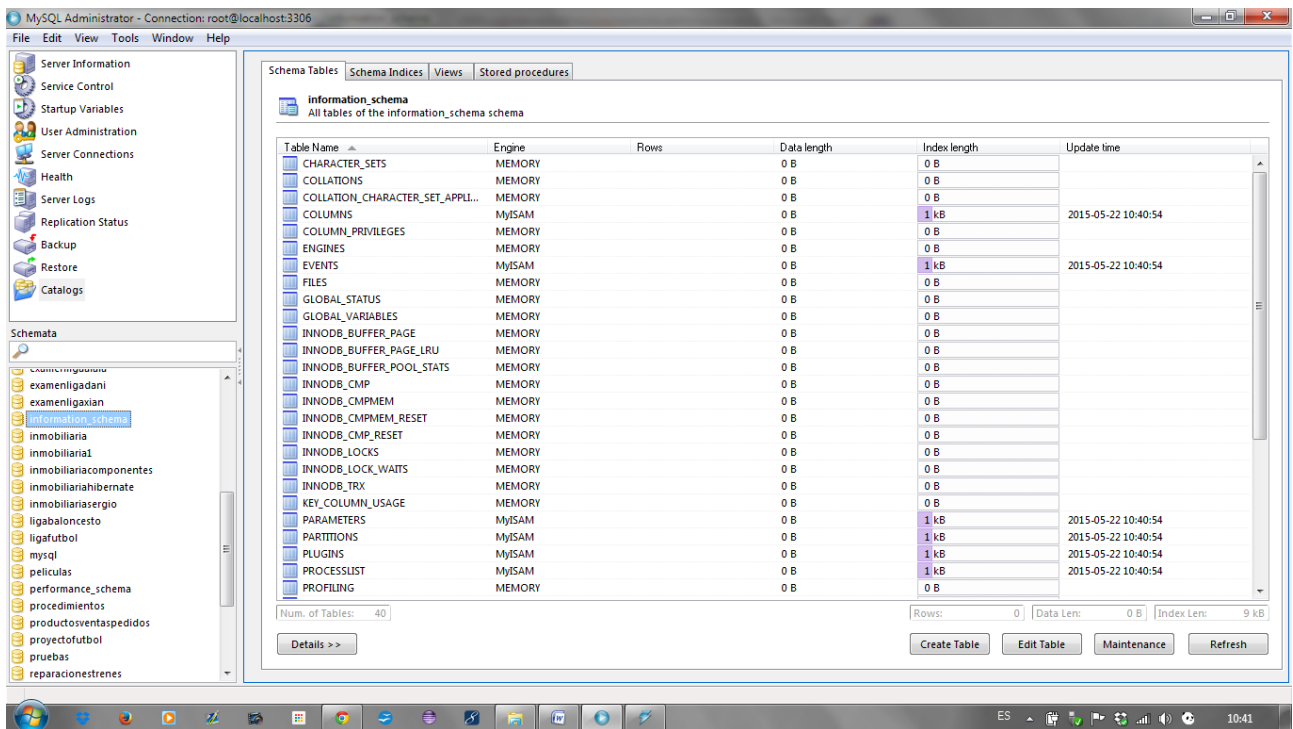


Ilustración 7 Schema de la BBDD Information\_Schema

```

1 SELECT table_schema, table_name, table_rows FROM information_schema.tables
2 WHERE table_schema = 'Biblioteca';

```

table_schema	table_name	table_rows
biblioteca	anaya	3
biblioteca	autores	9
biblioteca	editoriales	9
biblioteca	ejemplares	46
biblioteca	idiomas	3
biblioteca	libros	10
biblioteca	librosprestados	NULL
biblioteca	nacionalidades	8
biblioteca	prestamos	19
biblioteca	sanciones	3
biblioteca	socios	12
biblioteca	temas	5
biblioteca	vencimientosmes	5

Ilustración 8 Consulta de una Schema de una BBDD en MySQL

## 2.3. Usos de las Bases de Datos

Las bases de datos se almacenan en cualquier sistema informático. Algunos usos de los más frecuentes son:

- Bases de Datos Administrativas: guardan información de los clientes, facturas, pedidos, proveedores, etc., la mayoría de las empresas las utilizan en la actualidad.
- Base de Datos Contables: gestionan los pagos, balances, declaraciones a hacienda, etc.
- Bases de Datos para motores de búsqueda: por ejemplo Google o Altavista, tienen una base de datos gigante donde almacenan información sobre todos los documentos de internet. Posteriormente los usuarios buscamos en las bases de datos de esos motores.

- Científicas: Recolección de datos climáticos y medioambientales, químicos, geológicos, etc.
- Configuraciones: almacenan datos de configuración de un sistema informático, por ejemplo, el registro de Windows.
- Bibliotecas: almacenan información bibliográfica, por ejemplo, la tienda Amazon o la biblioteca de una ciudad.
- Censos: guardan información demográfica de pueblos, ciudades y países.
- Virus: los antivirus guardan información sobre todos los potenciales software maliciosos.
- Otros: militares, vídeo juegos, deportes, etc.

Las bases de datos pueden ser de cualquier tamaño y tener diversos grados de complejidad.

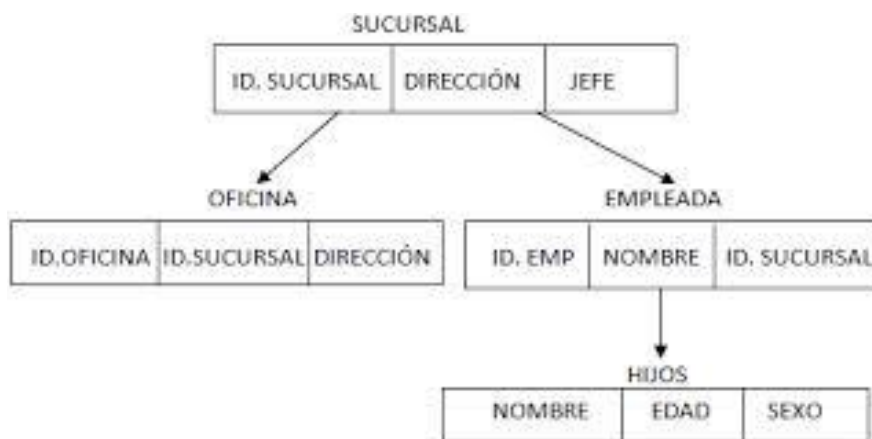
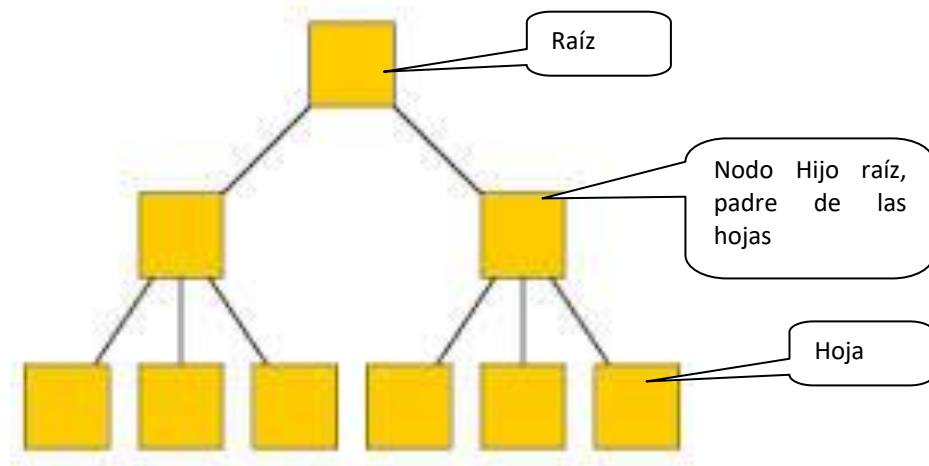
Por ejemplo, **la lista de nombres y direcciones** antes mencionada puede contener apenas unas cuantas centenas de registros, cada uno de ellos con una estructura muy simple. Por otro lado, **el catálogo de una biblioteca** grande puede contener medio millón de tarjetas clasificadas por categorías distintas —apellido del primer autor, tema, título, etc.— y ordenadas alfabéticamente en cada categoría. Las **autoridades fiscales** mantienen una base de datos todavía más grande y compleja para llevar el control de las declaraciones fiscales que presentan los contribuyentes. Si suponemos que en España hay 41 millones de contribuyentes y que cada uno aproximadamente unos 200 caracteres de información por formulario de declaración y son 5 formularios de media, las autoridades fiscales de ese país manejarían una base de datos con  $41 \cdot (10^6) \cdot 200 \cdot 5$  de caracteres (bytes) de información. Suponiendo que dichas autoridades conservan las últimas tres declaraciones de cada contribuyente, además de la actual, tendríamos una base de datos de  $4 \cdot 41 \cdot (10^9)$  bytes. Esta enorme cantidad de información debe organizarse y controlarse para que los usuarios puedan buscar, obtener y actualizar los datos cuando sea necesario.

## 2.4. Evolución y Tipos de Bases de Datos

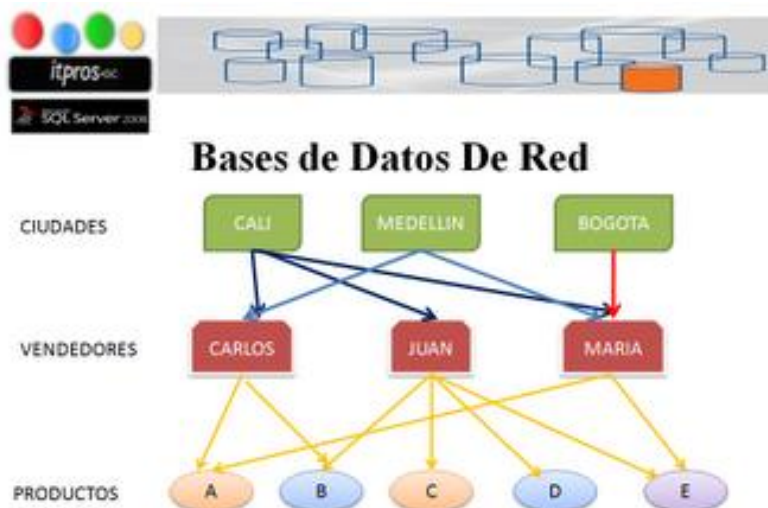
La clasificación de las bases de datos se realiza por su evolución histórica. A medida que avanza la tecnología, las bases de datos han ido mejorando en la forma de representar y extraer la información.

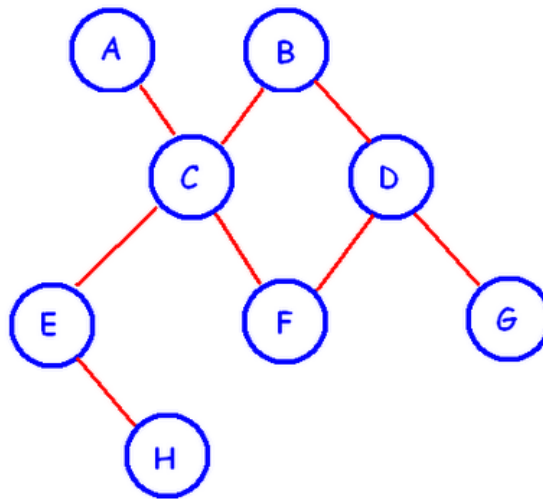
En los años 50 comienzan a utilizar las cintas magnéticas que permitían tratar la información de forma secuencial. Las cintas almacenaban ficheros con registros que se procesaban secuencialmente con ficheros de movimientos para generar nuevos ficheros actualizados. A este tipo de aplicaciones de les llama *“aplicaciones basadas en sistemas de ficheros”* y forman la generación cero de las bases de datos, ya que en esos momentos no existía el concepto de base de datos.

En la década de los 60 se generaliza el uso de discos magnéticos, que se caracterizan por qué se puede acceder a la información de forma directa sin tener que pasar por todas las anteriores. Con esta tecnología aparecen las bases de datos **jerárquicas** y en **red**, que aprovechan la capacidad de acceso directo a la información de los discos magnéticos para estructurar la información en forma listas enlazadas y árboles de información. La filosofía de las bases de datos en red es que un concepto principal o padre puede tener numerosas relaciones con conceptos secundarios o hijos.



Las bases de datos jerárquicas, han evolucionado para admitir varios padres para un concepto hijo.





En los años 70, Edgar Frank Codd, científico informático de IBM, publicó un artículo “Un modelo relacional de datos para grandes bancos de datos compartidos” donde sentó las bases del modelo relacional, basado en la lógica de los predicados y la teoría de conjuntos. Así surgieron las **bases de datos relacionales** o segunda generación de bases de datos.

Larry Ellison, fundador de Oracle, se inspiró en el artículo de Codd para desarrollar el motor de bases de datos de Oracle, que comenzó como un proyecto para la CIA.

Las bases de datos relacionales se basan en las leyes de Codd, que son un conjunto de 13 reglas cuya finalidad es establecer las características que debe tener una base de datos relacional. Actualmente todos los sistemas gestores de bases de datos implementan estas reglas.

Hoy en día, el modelo relacional de Codd, sigue siendo el más utilizado.

En la década de los 80 IBM lanza el motor de bases de datos DB2, por la plataforma MVS (Múltiple Virtual Storage = Múltiple Almacén Virtual). Unos años después IBM creó SQL (*Structured Query Language* = *Lenguaje Estructurado de Consultas*). SQL es un potente lenguaje de consultas para manipular información de bases de datos relacionales.

A mediados de 1990, IBM y Oracle incorporan a sus bases de datos la capacidad de manipular objetos, creando así, las *bases de datos orientadas a objetos*. Estas bases de datos orientadas a objetos se basan en la existencia de objetos persistentes que se almacenan para su procesamiento mediante programas orientados a objetos. En lugar de la filosofía de almacenar relaciones y tablas, se almacenan colecciones de objetos, que además de información, tienen comportamientos (instrucciones sobre cómo procesar los datos).

La aparición de Internet y el comienzo de la era de la información, crean nuevos requerimientos para las bases de datos. La cantidad de información comienza a crecer. De esta forma, se crean las *bases de datos distribuidas*, que consisten en multiplicar el número de ordenadores que controlan una base de datos (llamados nodos), intercambiándose información y actualizaciones a través de la red. Este aumento de datos a almacenar, organizados a veces en datos estadísticos recopilados con el transcurso de los años, hizo necesaria la aparición de un software llamado **Software de ayuda a la decisión**. Este software avanzado trata de dar respuestas concretas examinando múltiples datos estadísticos que se han recopilado a lo largo del tiempo en base de datos multidimensionales, formando lo que se llaman cubos de información.

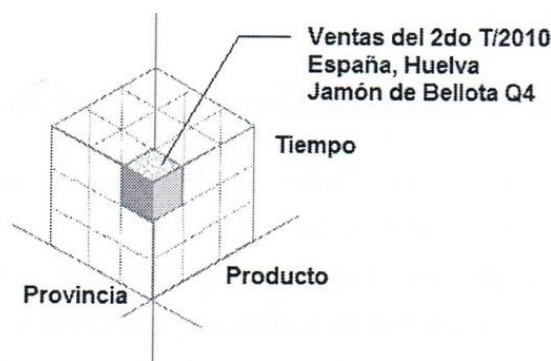


Ilustración 9 Cubo en una BBDD multidimensional

Otros tipos de bases de datos:

- Bases de datos *espaciales o geográficas*: Son bases de datos que almacenen mapas y símbolos que representan superficies geográficas. *Google Earth* es una aplicación que lanza consultas a bases de datos de este tipo.
- Bases de datos documentales: Permiten la indexación de texto para poder realizar búsquedas complejas en textos de gran longitud.
- Bases de datos *deductivas*: Es un sistema de bases de datos que almacena hechos y que permite, a través de procedimientos de inferencia, extraer nuevos hechos. Se basan en la lógica, por ello también se suelen llamar bases de datos lógicas.

Base de Datos	Datos almacenados	Ubicación
Sistema de ficheros Jerárquicas En red	Datos en ficheros Estructuras de datos (listas y árboles) Estructuras de datos (árboles y grafos)	Varios ficheros
Relacionales Orientadas a objeto Geográficas Deductivas Documentales	Teoría de conjuntos y relaciones Objetos complejos con comportamiento Puntos, Líneas y Polígonos Hechos y Reglas Documentos	Una o varias BBDD
Distribuidas Multidimensionales	Múltiples cubos	Varias BBDD en varios ordenadores

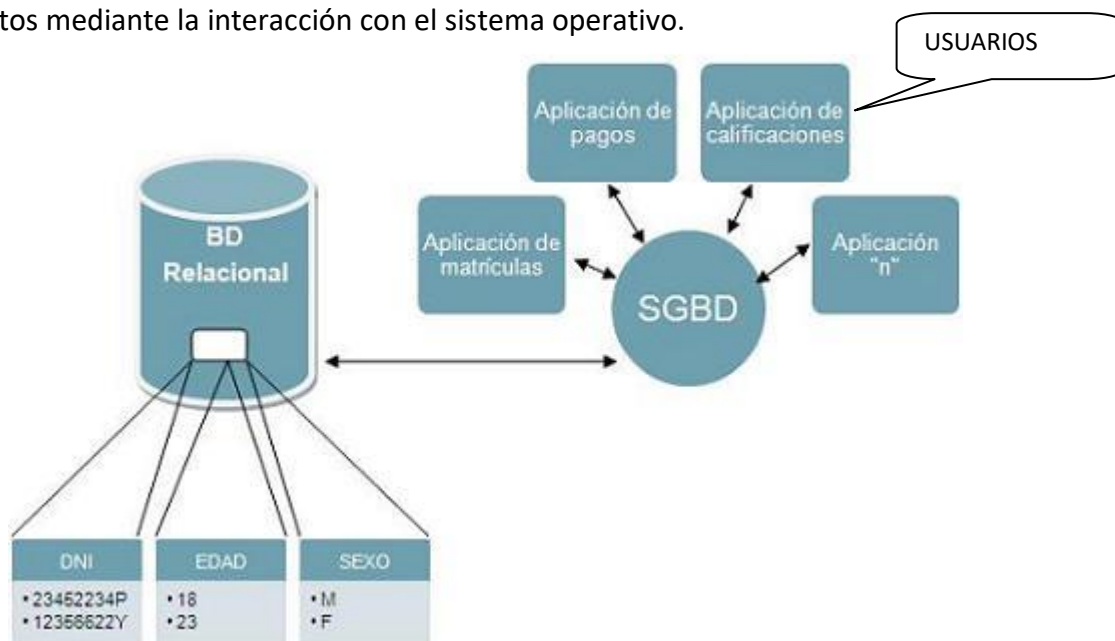
### 3. SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD)

#### 3.1 Concepto de Sistema de Gestión de Bases de Datos

Se define un SGBD como el conjunto de herramientas que facilitan la consulta el uso y actualización de una base de datos. Por ejemplo Oracle 11g es un SGBD que incorpora un conjunto de herramientas software que son capaces de estructurar en múltiples discos los ficheros de una base de datos, permitiendo el acceso a sus datos tanto a partir de herramientas gráficas como a partir de potentes lenguajes de programación (PL-SQL, php, C++, Java, etc.)

En la siguiente imagen se representa el funcionamiento de un sistema de información en el que los usuarios acceden a la información usando aplicaciones que, a su vez, se comunican con sistemas gestores, que son los que en última instancia acceden a los datos almacenados en las

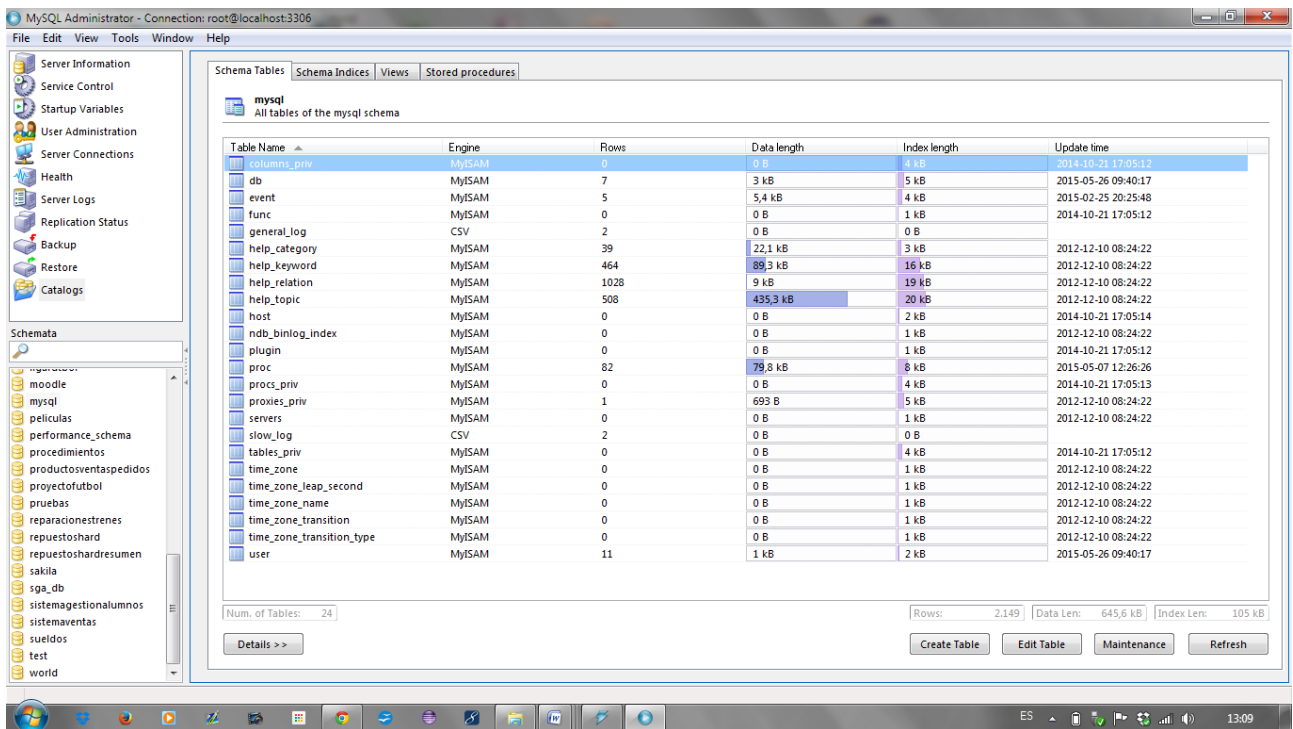
bases de datos mediante la interacción con el sistema operativo.



### 3.2 Funciones del SGBD

Para conseguir los objetivos anteriores la mayoría de los SGBD comerciales y libres incorporan las siguientes características y funciones:

1. Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla y con un gran rendimiento, ocultando la complejidad y las características físicas de los dispositivos de almacenamiento. Ejemplo insertar un nuevo cliente, consultar los clientes de Vigo.
2. Garantizan la integridad de los datos, respetando las reglas y restricciones que dicte el programador de la base de datos. Es decir, no permiten operaciones que dejen cierto conjunto de datos incompletos o incorrectos. Ejemplo validar datos, permitir nulos.
3. Integran, junto con el sistema operativo, un sistema de seguridad que garantiza el acceso a la información exclusivamente a aquellos usuarios que dispongan de autorización.
4. Proporcionan un diccionario de **metadatos**, que contiene el esquema de la base de datos, es decir, cómo están estructurados los datos en tablas, registros y campos, las relaciones entre los datos, usuarios, permisos, etc. Este diccionario de datos debe ser también accesible de la misma forma sencilla que es posible acceder a los datos.



5. Permite el uso de transacciones que garantizan que todas las operaciones de la transacción se realicen correctamente, y en caso de alguna incidencia, deshacen los cambios sin ningún tipo de complicación adicional. Ejemplo realizar una transferencia bancaria.
6. Ofrecen herramientas con las que podemos obtener estadísticas sobre el uso del gestor, registrando operaciones efectuadas, consultas solicitadas, operaciones fallidas y cualquier tipo de incidencia. Es posible de este modo, monitorizar el uso de la base de datos, y permiten analizar hipotéticos malos funcionamientos.
7. Permiten la concurrencia, es decir, varios usuarios trabajando sobre un mismo conjunto de datos. Además, proporcionan mecanismos que permiten arbitrar operaciones conflictivas en el acceso o modificación de un dato al mismo tiempo por parte de varios usuarios.
8. Independizan los datos de la aplicación o usuario que esté utilizándolos, haciendo más fácil su migración a otras plataformas.
9. Ofrecen conectividad con el exterior. De esta manera se puede replicar y distribuir bases de datos. Además, todos los SGBD incorporan herramientas estándar de conectividad. El protocolo ODBC<sup>3</sup>
10. Incorporan herramientas para salvar y restaurar la información en el caso de que se produzca algún problema. Algunos gestores, tiene mecanismos para poder restablecer el estado de una base de datos en cualquier punto anterior en el tiempo. Además de ofrecer sencillas herramienta para la importación y exportación automática de la información.

### 3.3. El lenguaje SQL

La principal herramienta de un gestor de base de datos es la interfaz de programación con el usuario. Este interfaz consiste en un lenguaje muy sencillo mediante el cuál el usuario realiza preguntas al servidor, contestando este a las demandas del usuario. Este lenguaje es SQL,

<sup>3</sup> ODBC Open Database Connectivity, es un estándar de acceso a datos desarrollado por Microsoft.

Structured Query Language o Lenguaje Estructurado de Consultas, está estandarizado por la ISO<sup>4</sup>, es decir, todas las bases de datos que soportan SQL deben tener la misma sintaxis a la hora de aplicar el lenguaje. Se divide en 4 sublenguajes, el total de ellos permite al SGBD cumplir con las funcionalidades requeridas por CODD:

- **Lenguaje DML** o Lenguaje de Manipulación de Datos (*Data Manipulation Lenguaje*). Este lenguaje permite 4 sentencias: seleccionar determinados datos (*SELECT*), insertar datos (*INSERT*), modificar los datos (*UPDATE*), y borrarlos (*DELETE*).
- **Lenguaje DDL** o Lenguaje de Definición de Datos (*Data Definition Lenguaje*). Este lenguaje permite crear toda la estructura de una base de datos; desde tablas hasta usuarios. Sus cláusulas son del tipo: *DROP* (Eliminar objetos), *CREATE* (Crear objeto), *ALTER* (Modificar objetos).
- **Lenguaje DCL** o Lenguaje de Control de Datos (*Data Control Lenguaje*). Incluye comandos (*GRANT*, *REVOKE*) que permiten al administrador gestionar el acceso a los datos contenidos en la base de datos.
- **Lenguaje TCL** o Lenguaje de Control de Transacciones. El propósito de este lenguaje es permitir ejecutar varios comandos de forma simultánea como si fuera un comando atómico e indivisible. Si es posible ejecutar todos los comandos se aplica la transacción (*COMMIT*), y si, en algún paso de la ejecución, sucede algo inesperado, se pueden deshacer todos los pasos dados (*ROLLBACK*).

### 3.4. Tipos de SGBD

Los SGBD se pueden clasificar de varias formas, según las bases de datos que gestionan: Bases de datos relacionales, Bases de datos orientadas a objetos, etc. En la actualidad la mayoría de los SGBD integran múltiples filosofías y tipos de funcionamiento, por ello vamos a clasificar los SGBD según su capacidad y potencia:

- **Los Gestores de Bases de Datos Ofimáticas:** son aquellos que manipulan bases de datos pequeñas (ofimáticas) orientadas a almacenar datos domésticos o de pequeñas empresas. Incluso estos gestores permiten construir pequeñas aplicaciones para ayudar a un usuario inexperto a manipular los datos de una base de datos de forma sencilla e intuitiva. Un ejemplo de un SGBD ofimático es Microsoft Access, que posee tanto una interfaz de usuario sencilla como un potente lenguaje de programación (VBA = Visual Basic for Applications) para ofrecer a usuarios avanzados otras posibilidades de gestión mucho más específicos.
- **Los Gestores de Bases de Datos Corporativas:** son aquellas que tienen la capacidad de gestionar bases de datos enormes, de grandes o medianas empresas con una carga de datos y transacciones que requieren un servidor de grandes dimensiones (generalmente un Servidor Unix, o un Windows 200X Server con altas prestaciones). Estos gestores son capaces de manipular grandes cantidades de datos de forma muy rápida y eficiente para poder resolver la demanda de muchos usuarios.  
Un ejemplo típico de servidor de bases de datos corporativas es Oracle junto con DB2, el servidor de base de datos más potente y más caro del mercado. Debido a este elevado coste se ha recurrido a una solución intermedia entre gestores de base de datos ofimáticas y corporativas. Una solución intermedia es MySQL, un gestor de base de datos que, además de ser gratuito y sencillo, es capaz de manipular gran cantidad de datos cumpliendo

---

<sup>4</sup> ISO, International Organization for Standardization.



prácticamente todos los estándares de la arquitectura ANSI SPARC. Aunque implementa SQL, no tiene un lenguaje de programación propio como SQL Server u Oracle, pero a cambio se integra fácilmente en soluciones XAMPP, que son paquetes que incluyen, además de MySQL, una versión del servidor Web Apache y varios lenguajes de script (php, perl, etc.) que dotan a MySQL de potentes herramientas para acceso y publicación de los datos.

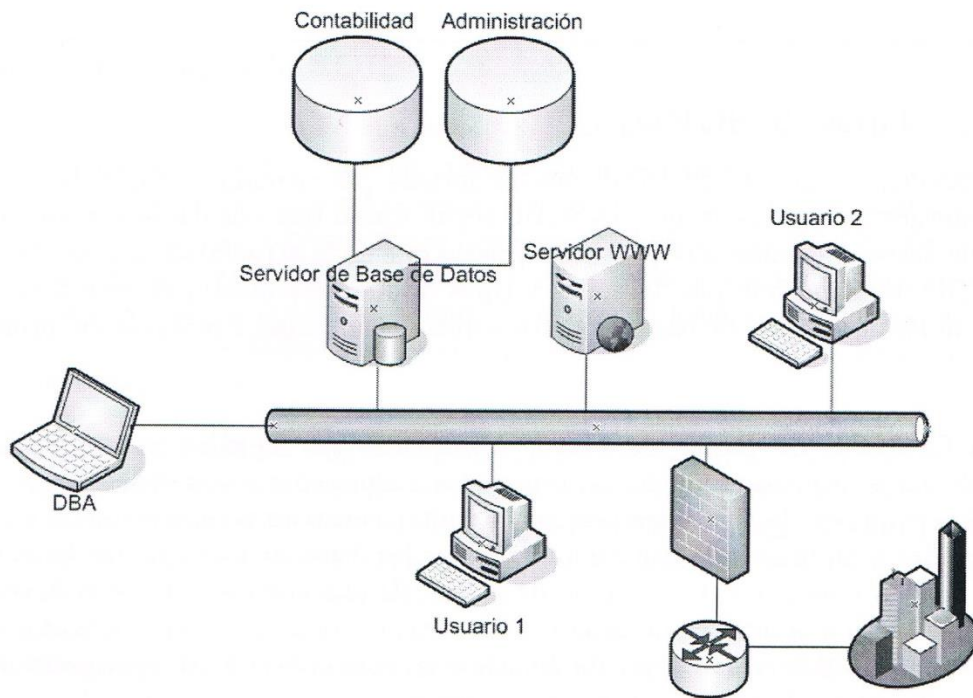


Ilustración 10. Esquema de un SGBD Corporativo