

IMF Smart Education

IMF Smart Education

## **Almacenamiento de la información © IMF Smart Education**

IMF Smart Education

IMF Smart Education

IMF Smart Education

IMF Smart Education

-tion

-tion

# Índice

<b>Almacenamiento de información</b>	<b>3</b>
I. Introducción	3
II. Objetivos	3
III. Contenidos	3
3.1. Ficheros	3
3.1.1. Ficheros planos	4
3.1.2. Ficheros indexados	5
3.1.3. Ficheros de acceso directo o aleatorio	6
3.2. BBDD. Conceptos, usos y tipos según el modelo de datos, la ubicación de la información	6
3.2.1. Diseño	7
3.2.2. Gestión de datos	8
3.2.3. Tipos de BBDD	9
3.3. Sistemas Gestores de Bases de Datos	14
3.3.1. Funciones básicas de un SGBD	15
3.3.2. Administrador de base de datos	16
3.3.3. Tareas de un SGBD	18
3.3.4. Componentes de un SGBD	19
3.3.5. Tipos de SGBD	20
3.4. SGBD comerciales	24
3.5. SGBD libres	25
3.6. BBDD centralizadas	26
3.7. BBDD distribuidas	29
3.8. Servidor MySQL	31
3.8.1. Elementos	31
3.8.2. Motores de almacenamiento	32
3.8.3. InnoDB motor de almacenamiento transaccional	34
3.8.4. Administrador: MySQLadmin	36
3.9. Grupo de trabajo	37
IV. Actividades interactivas	38
Actividad 1: Localiza los distintos tipos de BBDD	38
Actividad 2: Establece las relaciones correctas	38
V. Resumen	39
VI. Lecturas obligatorias	39
<b>Ejercicios</b>	<b>40</b>
Ejercicio 1	40
Enunciado	40
Se pide	40
Solución	40
<b>Recursos</b>	<b>42</b>
Enlaces de Interés	42
Bibliografía	42
Glosario.	42

# Almacenamiento de información

## I. Introducción

En esta unidad se presentan los sistemas lógicos de almacenamiento y sus características. Entre ellos, los más importantes son las bases de datos (BBDD), de las cuales se conocerán las más significativas según su modelo de datos y la ubicación de la información.

Veremos la utilidad de un sistema gestor de bases de datos (SGBD) y las funciones del administrador de la base de datos (DBA). Identificaremos los elementos de un SGBD, los diferentes tipos y sus funcionalidades más características.

Además, estudiaremos las ventajas y desventajas que tienen las BBDD centralizadas con respecto a las distribuidas y mostraremos un SGDB que se utilizará a lo largo del curso: MySQL.

## II. Objetivos

Conocer las características generales de un SGDB y las funciones de un DBA.

Saber la utilidad de los SGBD, las características de los distribuidos y centralizados.

Presentar los distintos modelos de datos que implementan los sistemas.

## III. Contenidos

### 3.1. Ficheros

## Almacenamiento de la información

El almacenamiento de información en sistemas de almacenamiento secundario (es decir, discos o cintas magnéticas) ha sido objeto de estudio dentro del campo de la computación por la necesidad de guardar y recuperar grandes cantidades de datos en el menor tiempo posible.

Según el tipo de dato a almacenar y el volumen de los mismos existen varias alternativas. A continuación, se presentan algunos tipos de ficheros:

### 3.1.1. Ficheros planos

Los ficheros planos se identifican por un nombre y la descripción de la carpeta o directorio que lo contiene. Son **ficheros sin formato**.

La información que contienen son letras, números y signos de puntuación. También soportan caracteres especiales (tabuladores o saltos de línea), que no son imprimibles pero tienen efecto en la impresión. Su representación en el computador se realiza por medio de sistemas de codificación como:

#### ASCII (American Standard Code for Information Interchange)

El ASCII estándar consta de 128 caracteres y no incluye los diacríticos propios de algunos idiomas como: á, ä, ÿ, ó, ñ... Posteriormente, se estableció el ASCII-extendido, con la finalidad de incluirlos, pasando así a tener 256 caracteres. Dentro de esta extensión, existen dos normas:

##### ISO-8859-1

Se trata de la norma ISO que incluye todos los caracteres latinos y los diacríticos.

##### UTF-8

Se trata de la codificación de caracteres Unicode e ISO 10646, que utilizan símbolos de longitud variable. Es muy común en correos electrónicos y páginas web.

La información de estos ficheros se puede abrir con cualquier editor de texto, como Notepad o Vim.

## Almacenamiento de la información

Caracteres ASCII de control			Caracteres ASCII imprimibles				ASCII extendido									
00	NULL	(carácter nulo)	32	espacio	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(inicio encabezado)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(inicio texto)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
03	ETX	(fin de texto)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	õ
04	EOT	(fin transmisión)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö
05	ENQ	(consulta)	37	%	69	E	101	e	133	å	165	Ñ	197	†	229	Õ
06	ACK	(reconocimiento)	38	&	70	F	102	f	134	à	166	ª	198	‡	230	µ
07	BEL	(timbre)	39	'	71	G	103	g	135	ç	167	º	199	Ä	231	þ
08	BS	(retroceso)	40	(	72	H	104	h	136	ê	168	¿	200	Å	232	ƒ
09	HT	(tab horizontal)	41	)	73	I	105	i	137	ë	169	©	201	Æ	233	ù
10	LF	(nueva línea)	42	*	74	J	106	j	138	è	170	¬	202	Œ	234	û
11	VT	(tab vertical)	43	+	75	K	107	k	139	ï	171	½	203	Ŧ	235	ü
12	FF	(nueva página)	44	,	76	L	108	l	140	î	172	¾	204	‡	236	ý
13	CR	(retorno de carro)	45	-	77	M	109	m	141	í	173	¿	205	=	237	ÿ
14	SO	(desplaza afuera)	46	.	78	N	110	n	142	Ä	174	«	206	†	238	—
15	SI	(desplaza adentro)	47	/	79	O	111	o	143	Å	175	»	207	‡	239	'
16	DLE	(esc.vínculo datos)	48	0	80	P	112	p	144	É	176	×	208	ø	240	=
17	DC1	(control disp. 1)	49	1	81	Q	113	q	145	æ	177	÷	209	Ð	241	±
18	DC2	(control disp. 2)	50	2	82	R	114	r	146	Æ	178	×	210	È	242	—
19	DC3	(control disp. 3)	51	3	83	S	115	s	147	ó	179	—	211	É	243	¾
20	DC4	(control disp. 4)	52	4	84	T	116	t	148	ö	180	—	212	Ê	244	¼
21	NAK	(conf. negativa)	53	5	85	U	117	u	149	ò	181	À	213	Ï	245	§
22	SYN	(inactividad sinc)	54	6	86	V	118	v	150	û	182	Á	214	Í	246	÷
23	ETB	(fin bloque trans)	55	7	87	W	119	w	151	ù	183	Â	215	Î	247	°
24	CAN	(cancelar)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(fin del medio)	57	9	89	Y	121	y	153	Û	185	ª	217	Œ	249	°
26	SUB	(sustitución)	58	:	90	Z	122	z	154	Ü	186	»	218	Œ	250	°
27	ESC	(escape)	59	;	91	[	123	{	155	ø	187	ª	219	Œ	251	°
28	FS	(sep. archivos)	60	<	92	\	124		156	£	188	ª	220	Œ	252	°
29	GS	(sep. grupos)	61	=	93	]	125	}	157	Ø	189	ª	221	Œ	253	°
30	RS	(sep. registros)	62	>	94	^	126	~	158	x	190	¥	222	Œ	254	■
31	US	(sep. unidades)	63	?	95	_			159	f	191	γ	223	■	255	nbsp
127	DEL	(suprimir)														

**Tabla 1.** American Standard Code for Information Interchange.

*Fuente:* American Standard Code for Information Interchange.

### 3.1.2. Ficheros indexados

Los indexados son ficheros que incluyen índices en la estructura del archivo. Los índices refieren a las porciones de información que se guarda y que se denominan “registros”, como la información de contacto de una persona.

Los registros están organizados en campos, por ejemplo, el nombre, calle, número de teléfono... del contacto.

Uno de los campos (o una combinación de ellos) puede ser un campo clave, que significa que distingue un registro de otro; por ejemplo, el nombre del contacto.

Los índices permiten que sea más fácil y rápido buscar algún registro concreto sin necesidad de recorrer todo el archivo. El índice consiste en un listado de los valores del campo clave que se encuentran en el archivo, junto con la posición de registro correspondiente en el almacenamiento masivo.

### 3.1.3. Ficheros de acceso directo o aleatorio

Los ficheros de acceso directo están formados por “trozos” de igual tamaño, y, por lo que el ordenador será capaz de saltar al registro que nosotros queramos para leerlo o escribirlo, desplazándose dentro del dispositivo que almacena la información (por ejemplo, un disco) en esas unidades de tamaño.

Los registros se corresponden con la estructura de un tipo de datos definido por el usuario y, para acceder a ellos, solo es necesario calcular su posición inicial a partir del número de registro al que se quiere acceder y del tamaño de cada registro.

## 3.2. BBDD. Conceptos, usos y tipos según el modelo de datos, la ubicación de la información

Las BBDD son una colección de datos relacionados que representan aspectos del mundo real. Es una colección coherente con significados inherentes. Esta se desarrolla, construye y puebla de datos con un propósito específico y desempeña un papel fundamental en casi todas las áreas de la computación, además de estar presentes en prácticamente todos los campos de aplicación.

Los **elementos** de una base de datos son los siguientes:

### **Tabla**

Entidad lógica de almacenamiento formado por columnas y tuplas.

### **Columnas o campos**

Dato del mismo tipo; parte de la información que se guarda para un elemento.

## Almacenamiento de la información

### Tupla o registro

Todos los datos (campos): asociados a un solo elemento.

### 3.2.1. Diseño

El diseño de una BD está enfocado al mantenimiento, ya que es importante que este sea fácil y seguro. Para realizar un buen diseño, es necesario tener en cuenta los siguientes **factores**:

#### Especificación de requisitos

Descripción de aquello que se desea que cumpla la BD. A partir de la especificación de requisitos, se realiza un diseño conceptual de la BD.

#### Normalización

Permitirá el establecimiento de un diseño lógico que establecerá el diseño físico de los datos.

#### Número de instancias

Una instancia es la colección de información almacenada en la BD en un determinado momento de tiempo.

#### Esquema

Diseño global de la BD. Los SGBD disponen de varios esquemas, divididos de acuerdo con los niveles de abstracción: físico, conceptual y varios subesquemas.

### Independencia de datos

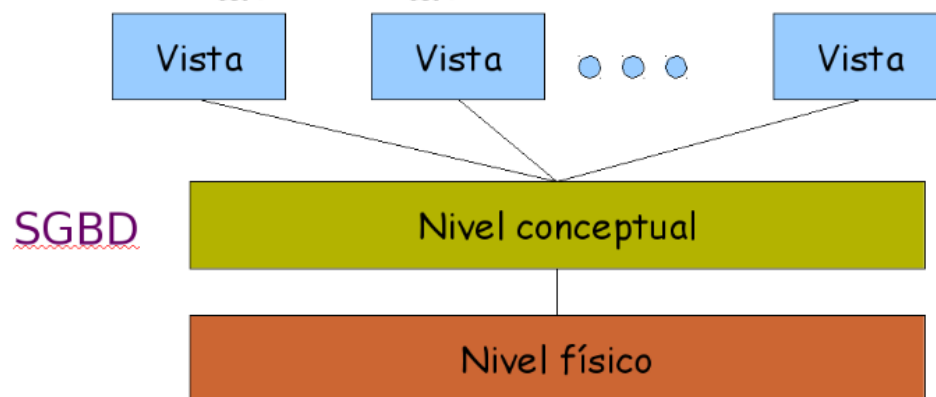
Capacidad de modificar una definición de un esquema en un nivel sin afectar al de nivel superior. Niveles de independencia:

- **Independencia física de datos:** permite modificar el esquema físico sin provocar que se vuelvan a reescribir los programas.
- **Independencia lógica de datos:** permite modificar el esquema conceptual sin provocar que se vuelvan a reescribir los programas (difícil).

### Niveles de abstracción

Descripciones jerárquicas de modelos de datos. Se definen tres niveles:

- **Nivel físico:** describe cómo se almacenan los datos y las estructuras de datos en detalle.
- **Nivel conceptual:** describe qué datos son almacenados y la relación entre ellos.
- **Nivel de visión:** describe las vistas de la BD y, normalmente, las que serán partes de la BD.



**Imagen 1.** Niveles de abstracción.

*Fuente:* elaboración propia.

### 3.2.2. Gestión de datos

Para rellenar una BD, es necesaria una definición de las estructuras que deben utilizarse para el almacenamiento de la información.

Los datos necesitan unos **mecanismos para su gestión**, centrados en la seguridad y el control:



## Almacenamiento de la información

Se debe mantener la seguridad de la información almacenada tanto frente a caídas del sistema como a intentos de acceso no autorizados al mismo.

El SGBD debe suministrar un control eficiente de la información, asegurando la integridad de los datos cuando estos son compartidos por varios usuarios.

También debe proveer las herramientas necesarias para poder realizar análisis forenses tras el fallo o accesos no controlados (tales como logs o registros de accesos, etcétera).

### 3.2.3. Tipos de BBDD

Las bases de datos se pueden clasificar utilizando diferentes criterios. En esta unidad se usarán dos: según la **variabilidad** de los datos y según el **modelo** de base de datos utilizado.

Según la variabilidad de los datos están las bases de datos estáticas y las dinámicas.

#### **BBDD estáticas**

Son bases de datos orientadas a la lectura, utilizadas principalmente para operar sobre históricos de datos (históricos de sucesos, esto es, cosas que pasan en momentos concretos de tiempo).

Estos sistemas están diseñados para potenciar las búsquedas y las operaciones de análisis, aunque eso signifique penalizar las operaciones de escritura y borrado (muchas veces ni siquiera se permiten los borrados, aunque, normalmente, es una limitación más de diseño lógico que un problema propio del sistema gestor usado). Con frecuencia eso implica, además, permitir datos duplicados o poco normalizados.

Habitualmente, mezclan datos de carácter heterogéneo con el objetivo de poder cruzar datos en los análisis para obtener conocimiento.

Un ejemplo de este tipo de sistemas sería uno que guardase los accesos a un sistema de cada usuario, así como el tiempo de sesión y las páginas abiertas. Cada registro se inserta una vez y no se modifica (pues, al ser un suceso pasado, no es susceptible de cambio); además, se utiliza para obtener estadísticas de accesos.

### BBDD dinámicas

Son bases de datos donde la información almacenada se modifica con el tiempo, lo que permite operaciones como la actualización, el borrado y la edición de datos, además de las operaciones fundamentales de consulta. También son llamadas bases de datos operacionales y son las más frecuentes, pues casi cualquier aplicación que utilice una base de datos suele tener una base de datos de este estilo (aunque eso no descarta que pueda, además, disponer o generar bases de datos estáticas para otros propósitos).

Por ejemplo, una tienda *online* almacenará los productos a la venta, los usuarios, precios y demás en una base de datos dinámica, y podrá almacenar el histórico de pedidos en una base de datos estática para poder realizar estadísticas o análisis de datos de una forma más rápida y cómoda.

Atendiendo a su modelo de datos, es posible encontrar una gran variedad de bases. En esta unidad se describirán solo las más comunes, a saber: relacionales, transaccionales, orientadas a objetos, no estructuradas (NoSQL) y distribuidas.

### BBDD relacionales

Las BBDD relacionales son las más comunes (y en las que se centrará este módulo). Almacenan la información en tablas bidimensionales de forma que cada columna se corresponde con una propiedad y cada fila (registro o tupla) corresponde a una unidad de información almacenada. Además, las diferentes tablas están relacionadas entre sí y se utilizan llaves (primarias o ajenas) para implementar esa relación. En la tabla 2 se puede ver un ejemplo de tabla de una base de datos relacional.

Nombre	Edad	Categoría	Salario
Raúl	45	Director	45.000€
Ángel	38	Jefe	35.000€
Andrés	35	Informático	18.000€

**Tabla 2.** BD en columnas.

*Fuente:* elaboración propia.

### BBDD Transaccionales

Las BBDD transaccionales sirven para almacenar transacciones (como su propio nombre indica) y tienen como objetivo almacenar y procesar gran cantidad de datos de forma rápida y, normalmente, con tolerancia a fallos. Son muy poco comunes y suelen utilizarse como bases de datos de apoyo de otras bases relacionales.

Un ejemplo de uso sería una cola de procesos que requiera una rápida respuesta al encolado y se deba o pueda desencolar poco a poco. Si se necesita almacenar las transacciones en una plataforma de venta de entradas de un concierto, por ejemplo, que se haga de forma masiva y muy rápida y su tramitación se pueda realizar *a posteriori*.

### BBDD orientadas a objetos

Las bases de datos orientadas a objetos están diseñadas para albergar información en un programa orientado a objetos y deben ser capaces de incorporar los elementos característicos de este paradigma (encapsulación, herencia y polimorfismo).

Este modelo es más reciente que los anteriores y aún poco utilizado (soportado, además, por un conjunto bastante pequeño de lenguajes). La principal característica es que, como parte de la definición del modelo de datos, se pueden definir operaciones con estos a través de una interfaz, de forma que, aparentemente, se simplifique el código que desarrollar en la aplicación.

### BBDD no estructuradas (NoSql)

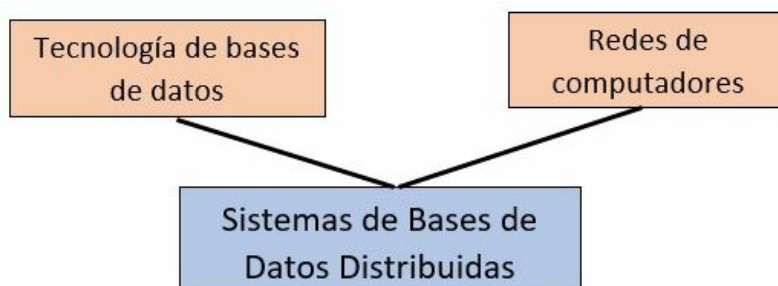
Las bases de datos no estructuradas no tienen una definición clara y definida de la organización de los datos y, por lo tanto, no se puede interactuar con ellas utilizando el lenguaje SQL.

Existen multitud de bases de datos NoSQL que se utilizan para los propósitos más diversos, por lo que es complicado encasillarlas o localizar puntos en común (más allá de la desestructuración). Uno de los casos de uso más habituales es su utilización para motores de búsqueda (Solr, Elastic Search), sistemas montados sobre bases de datos NoSQL especializados en realizar búsquedas complejas, ya sea en documentos o en datos operacionales, lo que permite características complejas como facetas o sugerencias.

Tras la aparición y popularización de MongoDB se comenzó a utilizar en ámbitos web y de desarrollo de aplicaciones de todo tipo creando nuevos paradigmas.

### BBDD distribuidas

Las BBDD distribuidas se organizan combinando la tecnología de BD y las redes de computadores.

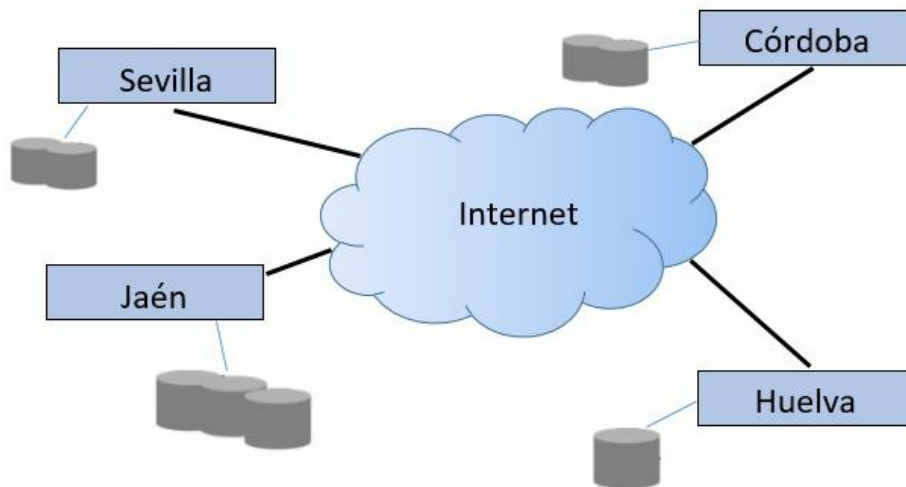


**Imagen 2.** Tecnologías de BBDD distribuidas.

*Fuente:* elaboración propia.

## Almacenamiento de la información

Una BD distribuida es un conjunto de múltiples BBDD relacionadas entre sí y que se encuentran distribuidas en diferentes sitios y servidores interconectados por una red de comunicaciones:



**Imagen 3.** Bases de datos distribuida en red.

*Fuente:* elaboración propia.

Actualmente, las bases de datos distribuidas están al alza debido a la aparición de las cadenas de bloques (*blockchains*). Aunque es una tecnología compleja, las cadenas de bloques proporcionan ciertas características que no existían hasta el momento, lo que permite almacenar la información de forma distribuida y consistente. Es una base de datos no estructurada donde la información de un bloque contiene metainformación de los bloques anteriores (por lo que no se pueden reescribir) y, además, se almacena en multitud de equipos, lo que dota a los datos de una inalterabilidad impropia de otros sistemas. Debido a esto se ha popularizado para multitud de casos de uso como los *bitcoins* (la moneda virtual) y aparecido como solución para certificar la propiedad de algo (certificando la transacción).

### BBDD documentales

Una BD documental es una base que almacena documentos. En estas, cada registro o tupla se corresponde con una referencia a un documento original, que puede ser:

- Una publicación independiente.
- Una parte con autonomía de contenido dentro de otra publicación.

Hay muchos tipos de BBDD documentales en función de cómo se almacena la información o del tipo de documento de que se trate, entre otras cosas. Según el tipo de **contenido**, se clasifican en:

#### Referenciales

Información del documento pero no su contenido.

## Almacenamiento de la información

### De texto completo

Contiene el texto del documento y permite búsquedas por contenido.

### Con reproducción del documento

Contiene las páginas escaneadas.

### BBDD geográficas con datos espaciales

Estas BBDD almacenan las características geográficas de los objetos descritos (ubicación, dimensión o forma).

Por ejemplo, los puntos que conforman el perímetro de una población, los cuales están almacenados en cierto tipo de archivos que pueden ser interpretados luego por las aplicaciones geográficas que se encuentran en el mercado.

### BBDD geográficas con datos no espaciales

Las BBDD geográficas no espaciales almacenan características cuantitativas de los objetos mientras que los datos descriptivos de estos se almacenan generalmente en tablas y se administran mediante algún manejador de BBDD.

### BBDD temporales

En estas BBDD, se maneja el tiempo junto con el dato. Los objetos y sus relaciones existen en periodos de tiempo definidos, y los eventos ocurren en puntos específicos del tiempo.

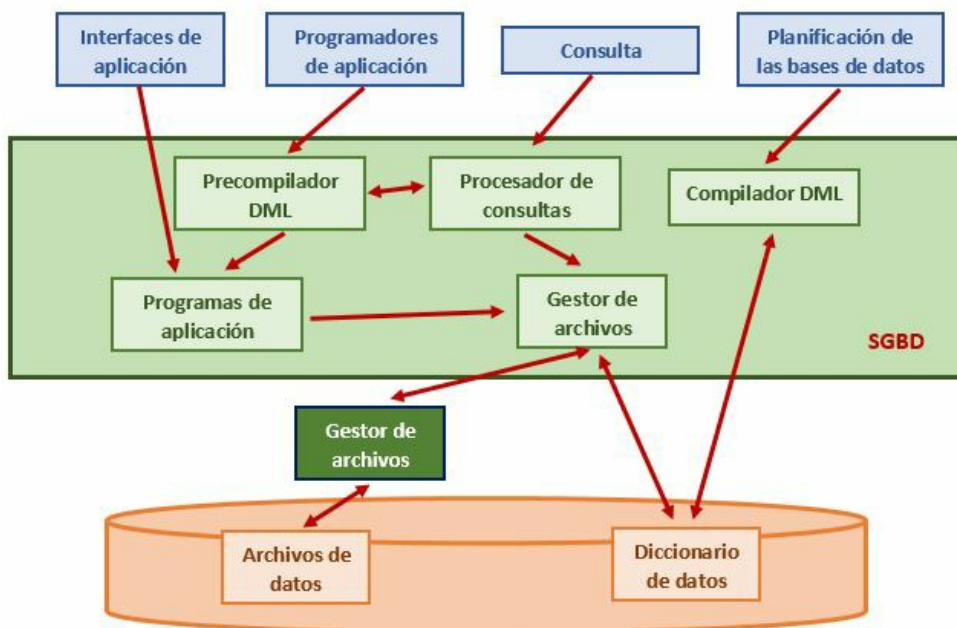
En ellas, se maneja el concepto de “tiempo válido”, el tiempo en el cual el hecho se dio o se dará en la realidad, independientemente del tiempo en que se registró en la BD. Dicho tiempo puede estar limitado o no. Suele estarlo y darse en el pasado para la mayoría de aplicaciones.

### 3.3. Sistemas Gestores de Bases de Datos

Un SGDB consiste en una colección de datos interrelacionados junto con unos programas para acceder a esos datos. Su estructura es la de un módulo de programa que proporciona la interfaz entre los datos de bajo nivel almacenados en la BD y los programas de aplicación y consultas del usuario.

Tiene como objetivo establecer un entorno amigable y eficiente para extraer y almacenar información. Los SGDB están diseñados para gestionar grandes bloques de información.

En la imagen siguiente, se presenta un esquema en que destaca (parte superior) los perfiles de usuarios y las acciones que estos realizan con el SGDB. En una capa intermedia están los diferentes módulos del SGDB que atienden y operan con el recurso físico del almacenamiento (simplicadamente un disco) donde se almacenan los datos dentro de archivos.



**Imagen 4.** Ubicación de un SGDB.

*Fuente:* elaboración propia.

Por lo tanto, interpretando la visión estructural de la imagen, un SGDB es una colección de archivos con un conjunto de programas que permite a los usuarios acceder y modificar estos archivos.

Sus **características** son:

## Almacenamiento de la información

Permitir la organización de la información dentro de un entorno operativo.

Abstraer detalles de cómo se almacenan y mantienen los datos.

Ejecutar las funciones necesarias para controlar y gestionar un conjunto de datos.

Proporcionar un método sencillo de extracción de los datos para que el sistema sea manejable.

### 3.3.1. Funciones básicas de un SGBD

**1**

Evitar la redundancia inesperada e inconsistencia de los datos.

**2**

Facilitar el acceso a los datos, permitiendo su recuperación de forma conveniente y eficiente.

## Almacenamiento de la información

3

Proporcionar uniformidad en el tratamiento de los datos, evitando el almacenamiento de datos en ficheros y formatos diferentes.

4

Mejorar el funcionamiento del sistema y obtener un tiempo de respuesta más rápido cuando se accede concurrentemente a los datos.

5

Establecer un sistema eficiente de seguridad para controlar el acceso de los usuarios a los datos, estableciendo y otorgando privilegios de acceso.

6

Solventar problemas de integridad, controlando que los datos almacenados satisfagan ciertos tipos de restricciones de integridad.

### 3.3.2. Administrador de base de datos

El DBA es el responsable del mantenimiento y la gestión de los datos. Se trata de un rol que suele asumir un usuario con un identificador y clave dentro del sistema. Con esta premisa, se debe encargar de:

La definición de esquema de la base de datos.



## Almacenamiento de la información

La definición de la estructura de almacenamiento y del método de acceso.

La integridad de los datos y su disponibilidad.

La modificación del esquema y de la organización física.

La distribución de los datos.

Las soluciones de almacenamiento.

La concesión de autorización para el acceso a los datos.

La especificación de las restricciones de integridad.

## Almacenamiento de la información

La gestión de las BBDD corporativas.

La gestión y el control de los servidores de BBDD.

La seguridad de las BBDD, que consiste en realizar copias periódicas de seguridad y llevar a cabo la recuperación de los datos ante situaciones de pérdida.

### 3.3.3. Tareas de un SGBD

**1**

Interacción con el gestor de archivos.

**2**

Implantación de la integridad.

**3**

Implantación de la seguridad.

## Almacenamiento de la información

4

Copia de seguridad y recuperación.

5

Control de concurrencia.

### 3.3.4. Componentes de un SGBD

Todo SGDB debe proporcionar una serie de servicios y medios para la definición y manipulación de datos:

#### **dll (data description language)**

Lenguaje de descripción de datos, es decir, el conjunto de definiciones que especifican un esquema de BD. Consisten en operaciones que permiten crear, borrar y actualizar los esquemas y subesquemas que se establecen al diseñar una BD.

#### **dml (data manipulation language)**

Lenguaje que permite a los usuarios acceder o manipular los datos en función de su modelo, permitiendo:

La inserción de nueva información.

La supresión de información.

## Almacenamiento de la información

La modificación de los datos almacenados.

Una vez definidas y creadas las tablas (o definición, en caso de ser una DB que no utilice tablas) mediante el DDL, el DML permite la inserción, borrado y actualización de los datos, así como su gestión mediante mecanismos eficientes.

### **servicios generales integrados**

- Seguridad.
- Recuperación.
- Estadísticas.
- Diccionario de datos.
- Monitor de teleproceso.
- Lenguaje de consultas.
- Servicios de conexión.

### 3.3.5. Tipos de SGBD

Para proporcionar lenguajes de definición, manipulación de datos y consultas, así como algunos servicios generales, los SGBD se basan en modelos de datos. A continuación se describen los más representativos.

#### **Modelos lógicos basados en objetos**

Se utilizan para describir datos en los niveles conceptual y de visión. El de mayor popularidad es el modelo entidad-relación.

#### **Modelo entidad-relación (E-R)**

Se basa en una percepción del mundo real, es decir, modela de forma esquemática la información asociada a los elementos que se desea manejar. Estos elementos son:

- **Entidad:** objeto básico, es el objeto físico o conceptual (persona/proyecto).

## Almacenamiento de la información

- **Atributos:** propiedades particulares que describen una entidad.
- **Relación:** asociación entre varias entidades.

Es un modelo de datos conceptual de alto nivel muy utilizado en el desarrollo de sistemas informáticos y aplicaciones para sistematizar la definición de requisitos, la realización del análisis y el diseño lógico de un sistema para garantizar que gestione la información de forma adecuada. Cuando se utiliza el modelo E-R en un desarrollo informático, es necesario establecer sin entrar en detalles de implementación:

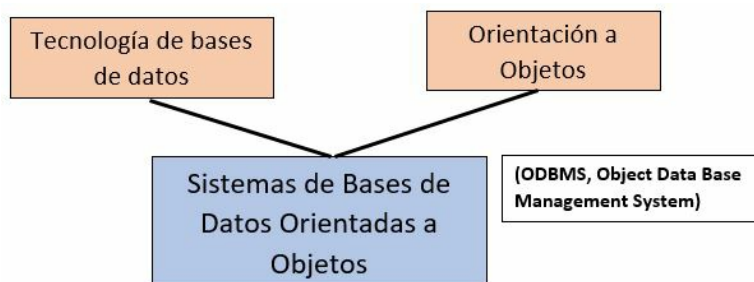
- El esquema conceptual de la BD.
- Contar con una descripción concisa de requisitos de datos.
- Establecer las entidades, relaciones y restricciones de estos datos.

Es el modelo más utilizado para diseño de BBDD y está soportado por multitud de herramientas que proporcionan un sistema gráfico cuya notación esquemática soporta los diagramas E-R.

### Modelo orientado a objetos (ODBMS, Object Database Management System)

Hace que los objetos de la BD aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte. Un ODBMS extiende los lenguajes con datos persistentes de forma transparente, control de concurrencia, recuperación de datos, consultas asociativas y otras capacidades.

Están diseñadas para almacenar objetos complejos, que incorpora todos los conceptos claves del paradigma de la orientación a objetos. Los datos se representan mediante los objetos, que son instancias de las clases definidas en un sistema orientado a objetos.



**Imagen 5.** Tecnologías de un ODBMS.

*Fuente:* elaboración propia.

La integración de ambas da como resultado es un SGBD orientado a objetos.

El sistema hace que los lenguajes de programación incorporen características propias de las BBDD como si fuesen propiedades del lenguaje.

### Modelos lógicos basados en registros

Se utilizan para describir datos en los niveles conceptual y de visión. Especifican la estructura lógica global de la BD y proporcionan una descripción a alto nivel de la implementación. Los datos se estructuran mediante registros de formato fijo, definiendo varios tipos. Cada registro se define por un número fijo de campos o atributos, normalmente de longitud fija.

### Modelo relacional

Representa los datos y las relaciones entre los datos mediante tablas. Estas tienen un número de columnas con nombres únicos, de este modo se guardan los datos. Para almacenar relaciones, la información de una tabla está enlazada con la información de otra u otras tablas mediante claves.

Por ejemplo, en la siguiente imagen se muestra una tabla con datos de conductores de coches: nombre, dirección, ciudad y un identificador único –ID–, que son números consecutivos en este caso. Este identificador se presenta en otra tabla asociados a las diferentes matrículas de coche, estableciendo la relación entre conductores y coches.

Nombre	Dirección	Ciudad	ID
Jesús López	Buenavista, 5	Sevilla	1
Pedro Hernando	Vista Alegre, 15	Barcelona	2
Ángel Pérez	Malavista, 7	Málaga	3
Pedro Hernando	Vista Alegre, 15	Barcelona	4
Jesús López	Buenavista, 5	Sevilla	2

ID	Matrícula
1	4848-xyr
2	5634-zfr
3	4578-ver
4	6432-yga

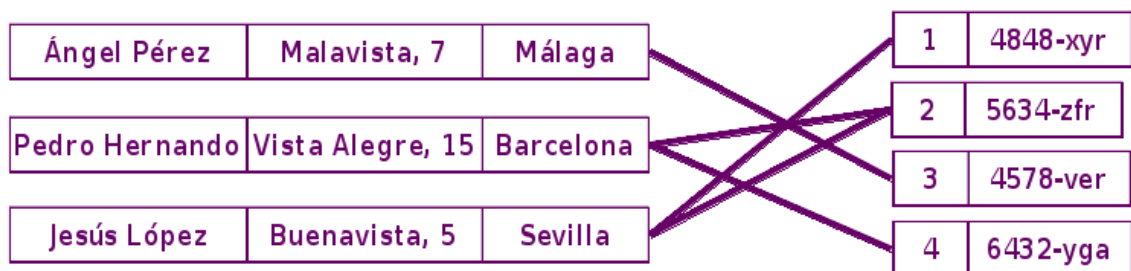
**Tabla 3.** Típicas tablas de un modelo racional.

*Fuente:* elaboración propia.

### Modelo en red

Este representa los datos mediante colecciones de registros, y las relaciones se representan mediante enlaces (punteros) que establecen las claves definidas en el modelo relacional.

Los registros de la base de datos se organizan como colecciones de grafos arbitrarios. El grafo establece las relaciones existentes entre las entidades del modelo. Gráficamente, un grafo se representa como un conjunto de puntos (vértices) unidos por líneas que permiten representar relaciones binarias (uno a uno) entre elementos, que, en este caso, son registros de la base de datos. En la imagen, cada registro de la izquierda (conductores) tiene una o varias relaciones con los de la derecha (coches).



**Imagen 6.** Organización de los datos en un modelo en red.

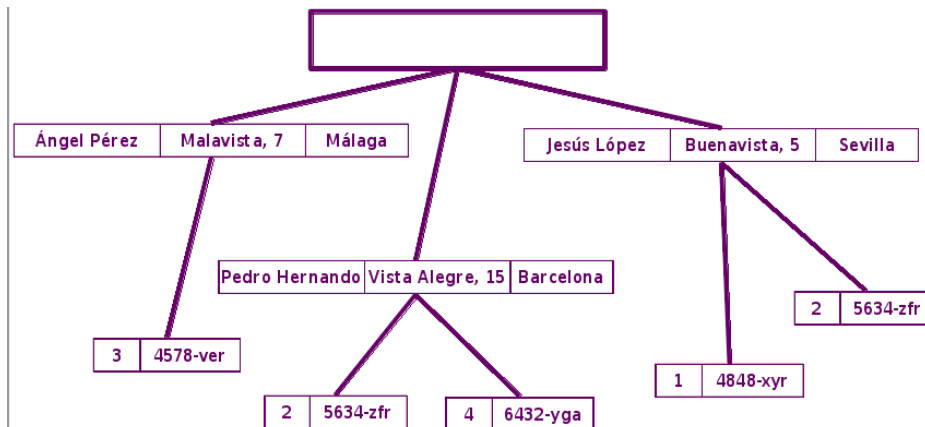
*Fuente:* elaboración propia.

### Modelo jerárquico

Representa los datos mediante colecciones de registros y las relaciones se representan mediante enlaces (punteros).

Los registros de la base de datos se organizan como árboles. En computación, un árbol (usualmente, se nombra como “árbol invertido”) es un concepto que define una estructura jerárquica con un nodo raíz que se encuentra en la parte superior y unos nodos hijos, con los que está relacionado mediante unos enlaces, que se encuentran en el siguiente nivel inferior que, a su vez, vuelven a tener estructura de árbol. Es una estructura de tipo recursivo: cada nodo es un nuevo árbol.

En la figura adjunta se aprecia el mismo ejemplo de conductores y coches en el modelo jerárquico:



**Imagen 7.** Organización de los datos en un modelo jerárquico.

*Fuente:* elaboración propia.

### 3.4. SGDB comerciales

Entre los SGDB's comerciales más conocidos, están Paradox, Informix, Sybase, Microsoft SQL Server y Oracle Database.

El principal motivo que alegan las empresas y organizaciones para utilizar SGDB comerciales es el soporte al usuario. Al ser organizaciones las que están detrás, tienen un soporte técnico del que muchas veces los sistemas libres carecen. El más utilizado de los SGDB comerciales es Oracle, cuyas principales características se verán a continuación.

#### Ventajas y características

Es el SGDB más usado a nivel mundial.



## Almacenamiento de la información

Dispone del lenguaje de programación PL/SQL.

Es muy potente y dispone de características como tipos de datos complejos, tablas anidadas, orientación a objetos, etcétera.

Tiene una integración sencilla con una gran cantidad de herramientas y *software* de Oracle.

Dentro del mundo de las grandes corporaciones es muy utilizado, por lo que incorpora mecanismos para adaptar el sistema a los complejos sistemas que se suelen utilizar.

### ¿Qué desventajas tiene?

Las licencias son excesivamente caras.

Se necesita un soporte continuo por parte de la casa matriz.

## 3.5. SGBD libres

Entre los SGBD libres, el más conocido es MySQL. Tras la absorción de MySQL por parte de Oracle, el responsable del proyecto, Michael Widenius, y parte de los principales desarrolladores de MySQL se desvincularon del proyecto para crear MariaDB (en honor a su hija María). No obstante (y a pesar de que era el principal temor de Michael y lo que le hizo desvincularse del proyecto) tanto MySQL como MariaDB son *software* libre.

## Almacenamiento de la información

### Ventajas

Cualquiera puede usar y modificar el *software*, ya que es libre.

Su SGDB es muy rápido, fiable y fácil de usar.

Tiene soporte para subconsultas, procedimientos almacenados y control de transacciones.

Incorpora un lenguaje de programación de estilo a PL/SQL.

### Desventajas

No tiene un panel de control gráfico y detallado que facilite las labores de mantenimiento. No obstante, existen multitud de programas y herramientas que permiten interactuar con este SGDB, por lo que no es un problema real.

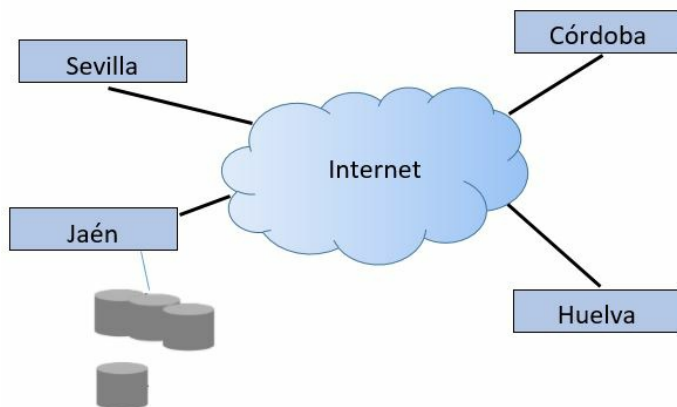
El soporte para disparadores es básico, hay limitaciones en lo que puede hacerse con ellos.

## 3.6. BBDD centralizadas

## Almacenamiento de la información

Las BBDD centralizadas almacenan la totalidad de la información en un solo lugar físico. Se ejecutan en un único computador.

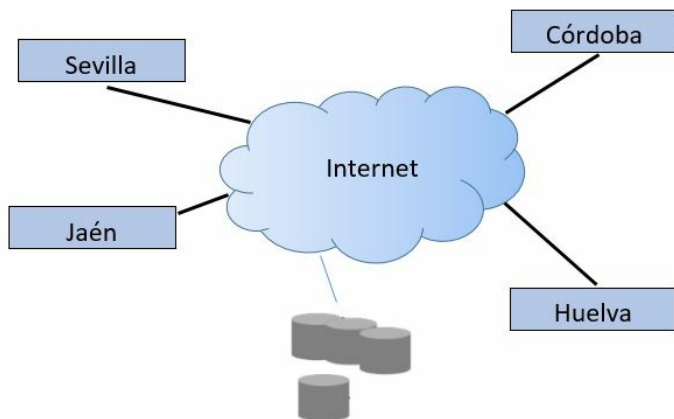
Aunque la empresa disponga de una amplia organización distribuida por diferentes ubicaciones, la BD se encuentra físicamente en una de ellas.



**Imagen 8.** BD centralizada en un nodo de la red.

*Fuente:* elaboración propia.

Aunque la red se encuentre en una ubicación diferente y no en una de las ubicaciones de la empresa, está igualmente concentrada en un lugar y, por lo tanto, centralizada.



**Imagen 9.** BD concentrada en la red.

*Fuente:* elaboración propia.

### Ventajas

## Almacenamiento de la información

Evita la redundancia de información de forma automática, sin configuración extra.

Evita la inconsistencia de forma automática, sin configuración extra.

Pueden aplicarse restricciones de seguridad de una manera más simple.

Puede conservarse la integridad de forma automática, sin configuración extra.

El procesamiento de los datos ofrece un mejor rendimiento y resulta más confiable que los sistemas distribuidos.

En definitiva, es un sistema más simple y, por lo tanto, más sencillo de crear y mantener.

### Desventajas

Cuando el sistema falla, se pierde toda disponibilidad de procesamiento y de información.

Las cargas de trabajo siempre se ejecutan en la misma máquina.

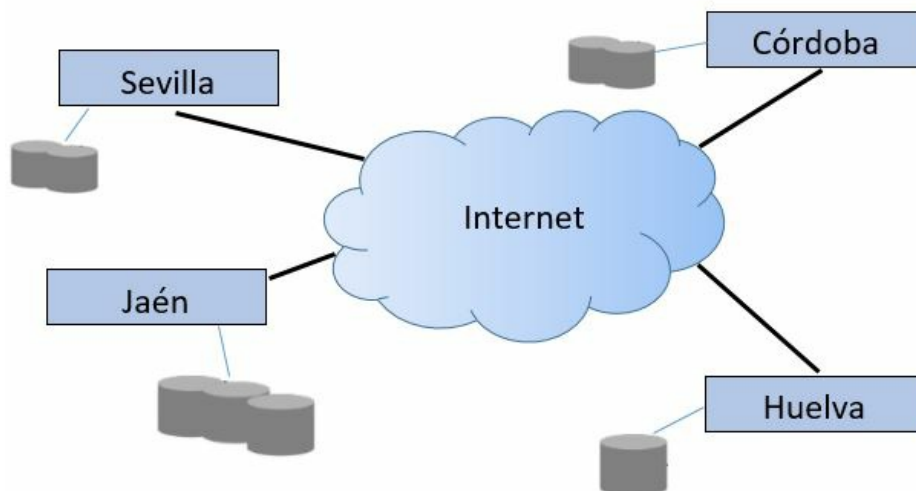
Escala mal. Ante un volumen demasiado grande de transacciones, no queda más remedio que mover el sistema a una base de datos distribuida.

### 3.7. BBDD distribuidas

Un SGDB distribuido es el sistema en el cual múltiples sitios de BBDD están ligados por un sistema de comunicaciones. Un usuario puede acceder en cualquier sitio a los datos alojados en cualquier parte de la red, exactamente como si estuvieran almacenados localmente en su sitio.

La aplicación trabaja, desde un punto de vista lógico, como un solo SGBD, ejecutado en una sola máquina, siendo transparente para el usuario su deslocalización.

**Imagen 10. BD distribuida**



**Imagen 10.** BD distribuida.  
*Fuente:* elaboración propia.

#### Ventajas

## Almacenamiento de la información

Si los datos están localizados en un lugar más cercano se tiene un acceso y procesamiento más rápido y una comunicación entre nodos menos cargada.

La interconexión resulta fiable y flexible en los SGDB existentes.

Mejor escalabilidad: siempre se pueden añadir nuevos nodos ante una carga de trabajo superior.

Se pueden colocar físicamente los datos en el lugar donde se accede más frecuentemente.

Mejor tolerancia a fallos. La caída de un nodo no implica la indisponibilidad total del sistema.

### Desventajas

Suele ser un sistema más caro.

Es un sistema complejo de configurar y mantener, pues es necesario asegurar la consistencia y validez de la información.

Con frecuencia presenta limitaciones ante grandes requisitos de escritura (aunque solventables).

Es necesario llevar control de concurrencia y mecanismos de recuperación ya que los datos pueden estar replicados.

En definitiva, es un sistema más complejo de mantener y, frecuentemente, más caro, pero la escalabilidad y la alta disponibilidad que aporta hacen que sea el sistema utilizado en cualquier proyecto de envergadura media o alta.

### 3.8. Servidor MySQL

Es un SGDB libre y fácil de utilizar, por lo que es con el que se realizan todas las prácticas a lo largo del curso.

[Manual de instalación de MySQL en Linux.](#)

[Manual de instalación de MySQL en MacOS.](#)

[Manual de instalación de MySQL en Windows.](#)

#### 3.8.1. Elementos

Los **componentes** del paquete MySQL son los siguientes:

##### **MySQLd (demonio MySQL)**

Es un proceso que está ejecutándose siempre en espera de que le lleguen peticiones, un proceso servidor. Los clientes se conectan a él para acceder a la BD.

## Almacenamiento de la información

### MySQLd-max

Versión aumentada del servidor con características adicionales.

### MySQLd\_safe

Alternativa para arrancar el servidor MySQL. Se intenta inicializar MySQLd-max si existe y MySQLd en caso contrario.

### MySQL.server

Se usa en sistemas que utilizan directorios de ejecución que contienen *scripts* que inician servicios para niveles de ejecución particular. **MySQLd\_multi**: comando que puede arrancar o parar varios servidores instalados en el sistema.

### MySQL\_install\_db

Comando de instalación. Crea las tablas de permisos de MySQL con privilegios por defecto. Normalmente se ejecuta solo una vez, cuando se instala por primera vez MySQL en el sistema.

### MySQL\_fix\_privilege\_tables

Comando que se usa tras una actualización, para actualizar las tablas de permisos con cualquier cambio que se hayan hecho en nuevas versiones de MySQL.

Cabe destacar que MySQL viene incluido en la mayoría de *suites* de desarrollo web Xamp (Apache-MySQL-PHP; la X hace referencia al sistema operativo, siendo W para Windows —WAMP— y L para Linux —LAMP—). Por lo que otra forma sencilla de instalar MySQL es instalar algunas de estas *suites*.

### 3.8.2. Motores de almacenamiento

Los motores de almacenamiento son los programas que implementan distintas técnicas para almacenar la información en la base de datos. La característica principal está en cómo se realiza la indexación de los datos, de forma que el acceso a la información sea lo más rápida y eficiente posible.



## Almacenamiento de la información

Es **muy importante** cómo establece los niveles de bloqueo para los accesos a modificación de datos, si bloquea a nivel de registro, de tabla... Los bloqueos son propios de los accesos a datos para su modificación (escritura). Se trata de evitar la concurrencia de dos o más procesos sobre el mismo dato con el propósito de modificarlo (no hay conflictos en la lectura simultánea de datos). Por ejemplo, en un sistema de compra de billetes de tren, si se realizan dos accesos simultáneos de compra contra un mismo billete, no puede otorgarse a ambos procesos ya que solo hay un billete. Es necesario realizar bloqueos para que esto no suceda y garantizar que solo un proceso tenga acceso a la modificación.

Originalmente, el bloqueo se realizaba a toda la BD, esto hacía que cuando un proceso iba a modificar un dato de una tabla, ningún otro proceso podía acceder a esa ni a ninguna otra tabla; por lo tanto, todos los usuarios se quedaban parados hasta que el proceso bloqueante terminaba. Los motores de almacenamiento fueron evolucionando y el bloqueo pasó a ser a la tabla donde estaba la información a modificar, luego a un conjunto de registros cercanos y finalmente bloqueando sólo el registro que se iba a modificar.

En esta última modalidad, retomando el ejemplo de compra del tren, es posible que haya un usuario comprando un asiento y otro proceso comprando el de al lado sin que se interfieran entre sí, ya que se trata de billetes para asientos distintos.

Al elegir un motor de inferencia u otro al desarrollar una aplicación, se puede ganar en velocidad, funcionalidad, operativa y eficiencia, factores que mejorarán la funcionalidad de la aplicación globalmente.

Los tipos de **motores de almacenamiento más importantes** son los siguientes:

### **InnoDB**

Permite tablas transaccionales con bloqueo a nivel registro y manejo de claves foráneas.

## Almacenamiento de la información

### MyISAM

Motor binario portable.

**Otros motores** de almacenamiento:

### MEMORY

Los datos se almacenan solo en memoria.

### MERGE

Colección de tablas MyISAM usadas como una sola tabla. También conocido como MRG\_MyISAM

### CSV

Tablas que almacenan registros en valores separados por comas.

### HEAP

Motor de almacenamiento MEMORY (HEAP).

### 3.8.3. InnoDB motor de almacenamiento transaccional

Este motor dota a MySQL de características que incrementan el rendimiento y la capacidad de gestionar múltiples usuarios simultáneos. Es el motor por defecto en MySQL a partir de su versión 4.0 (antes era MyISAM).

#### características

## Almacenamiento de la información

Tiene capacidades de aplicación de operaciones previamente definidas, marcha atrás de operaciones y recuperación de fallos.

Realiza bloqueos a nivel de registro.

Cuenta con funciones de lectura consistente sin bloqueo.

Soporta restricciones *Foreign Key*.

En consultas SQL, aún dentro de la misma consulta, pueden incluirse libremente tablas del tipo InnoDB y tablas de otros tipos.

Sin embargo, MyISAM presenta una mayor velocidad que InnoDB en la mayoría de los casos, dado que el primero no tiene que garantizar la integridad referencial (pues no la tiene soportada, al no soportar claves foráneas). Por otro lado, por cómo almacena MyISAM los datos en el sistema, presenta limitaciones en los tamaños máximos de tablas relacionadas con los tamaños máximos de ficheros en el SO en el que se alberga.

Además, MyISAM tiene peor respuesta ante el error, pues su sistema de recuperación de errores (y de recuperación tras una salida inesperada) es peor.

En definitiva, la elección de motor de almacenamiento no es una decisión menor y en algunos sistemas puede ser incluso algo crítico; aunque en la mayoría de los casos InnoDB sea la elección más acertada, no se habla de una tautología.

### 3.8.4. Administrador: MySQLadmin

El programa que realiza labores de administración en MySQL se llama “MySQLadmin”.

MySQLadmin [opciones] comando comando .....

Mediante las opciones, se establecen valores a un conjunto de variables que regulan la operativa del SGDB MySQL.

Los comandos realizan diferentes acciones en el servidor MySQL.

Algunos **comandos representativos y su función** son:

**create databasename**

Crea una nueva BD.

**drop databasename**

Borra una BD y sus tablas.

**kill id,id...**

Detiene threads MySQL.

**old-password new-password**

Cambia la contraseña.

## Almacenamiento de la información

### ping

Comprueba si MySQLd está vivo.

### shutdown

Para el servidor.

### status

Información de estado del servidor.

### variables

Muestra los valores de las variables.

### version

Da la versión.

## 3.9. Grupo de trabajo

Dentro de los grupos de trabajo que gestionan y operan BBDD, se pueden distinguir varios **perfiles**:

**1**

Responsable de realizar todas las actividades de configuración y mantenimiento.

**2**

Gestor de perfiles de usuarios y administración de permisos.

## Almacenamiento de la información

3

Responsable del control de acceso a la BD.

4

Responsable de la gestión de copias de seguridad.

5

Usuario que tiene permisos de ejecución de cualquier comando del lenguaje de definición de datos (DDL).

6

Usuario que tiene permisos de ejecución para poder agregar, eliminar o cambiar datos en todas las tablas (DML).

7

Usuario que tiene permisos de lectura de todos los datos de todas las tablas.

## IV. Actividades interactivas

### Actividad 1: Localiza los distintos tipos de BBDD



Localiza los distintos tipos de BBDD

### Actividad 2: Establece las relaciones correctas



Establece las relaciones correctas

## V. Resumen

En esta unidad, hemos aprendido:

- Los sistemas de almacenamiento y las características que los definen, centrándose especialmente en las BBDD.
- A definir los diferentes modelos de BBDD, sus características y su enfoque más apropiado, así como las ventajas y desventajas que tienen las centralizadas con respecto a las distribuidas.
- Los elementos más significativos de una BD: sistema gestor de bases de datos (SGBD) y administrador de la base de datos (DBA).

## VI. Lecturas obligatorias

- Hueso Ibáñez, L. *Administración de sistemas gestores de base de datos*. Madrid: RA-MA; 2016.
- Capacho, J. R. *Diseño de bases de datos*. Colombia: Universidad del Norte; 2017.

## Ejercicios

### Ejercicio 1

30

Este ejercicio es **opcional**, si tienes alguna duda puedes contactar con tu tutor.

### Enunciado

Se pretende almacenar en una base de datos relacional información acerca de equipos de fútbol y los grandes jugadores que han tenido a lo largo de la historia.

De cada equipo se pretende almacenar lo siguiente: su nombre, año de fundación, número de ligas ganadas y el nombre de su estadio.

De cada jugador se pretende almacenar lo siguiente: su nombre, el año de nacimiento, los goles totales, el año del fichaje y el equipo en el que jugó.

### Se pide

Diseña una o varias tablas para almacenar esta información en una base de datos relacional. Añade ejemplos de equipos y jugadores (puedes inventarlos) para ejemplificar cómo quedarían las distintas tuplas en las tablas.

### Solución

Deberían quedar un par de tablas similares a estas:

id	Nombre	Año de fundación	Ligas	Estadio
1	FC Barcelona	1899	27	Camp Nou
2	Real Madrid	1901	34	Santiago Bernabéu



## Almacenamiento de la información

Id	Nombre	Año de nacimiento	Goles	Equipo
1	Raúl González	1977	404	2
2	Luis Enrique García	1970	14	3
3	Luis Enrique García	1970	15	2
4	Luis Enrique García	1970	73	1

### Mejores soluciones

Utilizar una tercera tabla para almacenar los jugadores y año de nacimiento, dejando la tabla segunda como tabla para relacionar jugador-equipo.

### Peores soluciones

Usar el nombre del equipo en lugar de su identificador.

Este ejercicio pretende ayudar a comprender los problemas de adaptación de datos a bases, lo que se resolverá mejor en unidades futuras.

## Recursos

### Enlaces de Interés

- <https://dev.mysql.com/doc/refman/8.0/en/linux-installation.html>:
- <https://dev.mysql.com/doc/refman/8.0/en/macos-installation.html>:
- <https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html>:

### Bibliografía

- **Fundamentos de sistemas de bases de datos. (LECTURA RECOMENDADA)** : Elmasri, R.; Shamkant B.N. Fundamentos de sistemas de bases de datos. Madrid: Pearson Educación S.A.; 2007.
- **Fundamentos de sistemas de bases de datos. (LECTURA RECOMENDADA)** : Elmasri, R.; Shamkant B.N. Fundamentos de sistemas de bases de datos. Madrid: Pearson Educación S.A.; 2007.
- **Introducción a los sistemas de bases de datos. (LECTURA RECOMENDADA)** : Date, C.J. Introducción a los sistemas de bases de datos. Ed: Alhambra Mexicana; 2001.
- **Introducción a los sistemas de bases de datos. (LECTURA RECOMENDADA)** : Date, C.J. Introducción a los sistemas de bases de datos. Ed: Alhambra Mexicana; 2001.
- **MySQL para Windows y Linux** : Pérez López, C. *MySQL para Windows y Linux*. Madrid: RA-MA; 2007.
- **MySQL para Windows y Linux** : Pérez López, C. *MySQL para Windows y Linux*. Madrid: RA-MA; 2007.
- **Sistemas de bases de datos: diseño, implementación y administración** : Rob, P.; Coronel, C. *Sistemas de bases de datos: diseño, implementación y administración*. Madrid: Paraninfo; 2004.
- **Sistemas de bases de datos: diseño, implementación y administración** : Rob, P.; Coronel, C. *Sistemas de bases de datos: diseño, implementación y administración*. Madrid: Paraninfo; 2004.

### Glosario.

- **ASCII**: American Standard Code for Information Intechange.
- **Consulta**: Sentencia que solicita la recuperación de información.

## Almacenamiento de la información

- **DBA:** Administrador de la base de datos (Data Base Administrator).
- **DDL:** Lenguaje de descripción de datos (Data Description Language).
- **Diccionario de datos:** Archivo especial del gestor de BD donde se almacenan un conjunto de tablas que son el resultado de las compilaciones de sentencias DDL.
- **DML:** Lenguaje de manipulación de datos (Data Manipulation Language).
- **Lenguaje de consultas:** Paquete de sentencia de lenguaje de manipulación de datos que implica recuperación de información.
- **Mainframe:** (computadora central): computador grande y potente que se utiliza para el proceso de gran cantidad de datos en procesadores paralelos. Actualmente están siendo sustituidas por muchos computadores conectados en red.
- **Metadatos:** Directorio de datos, se consulta antes de leer o modificar los datos reales en el sistema de base de datos.
- **Notepad:** Editor de texto de Windows.
- **SGBD:** Sistema Gestor de Bases de Datos.
- **Vim:** Gama de editores de texto de UNIX/Linux.