

BASE DE DATOS

Introducción

Una base de datos, o banco de datos, es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados (ordenados) para su consulta.

Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Motores de bases de datos

Un motor de Bases de Datos es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

Los usuarios pueden acceder a la información usando herramientas específicas de consulta y / o de generación de informes, o bien mediante aplicaciones al efecto.

En base a esto, podemos definir que los motores de base de datos sirven para **definir, construir y manipular** una base de datos.

1. **Definir una base de datos:** consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.
2. **Construir una base de datos:** es el proceso de almacenar los datos sobre algún medio de almacenamiento.
3. **Manipular una base de datos:** incluye funciones como consulta, actualización, etc. de bases de datos.

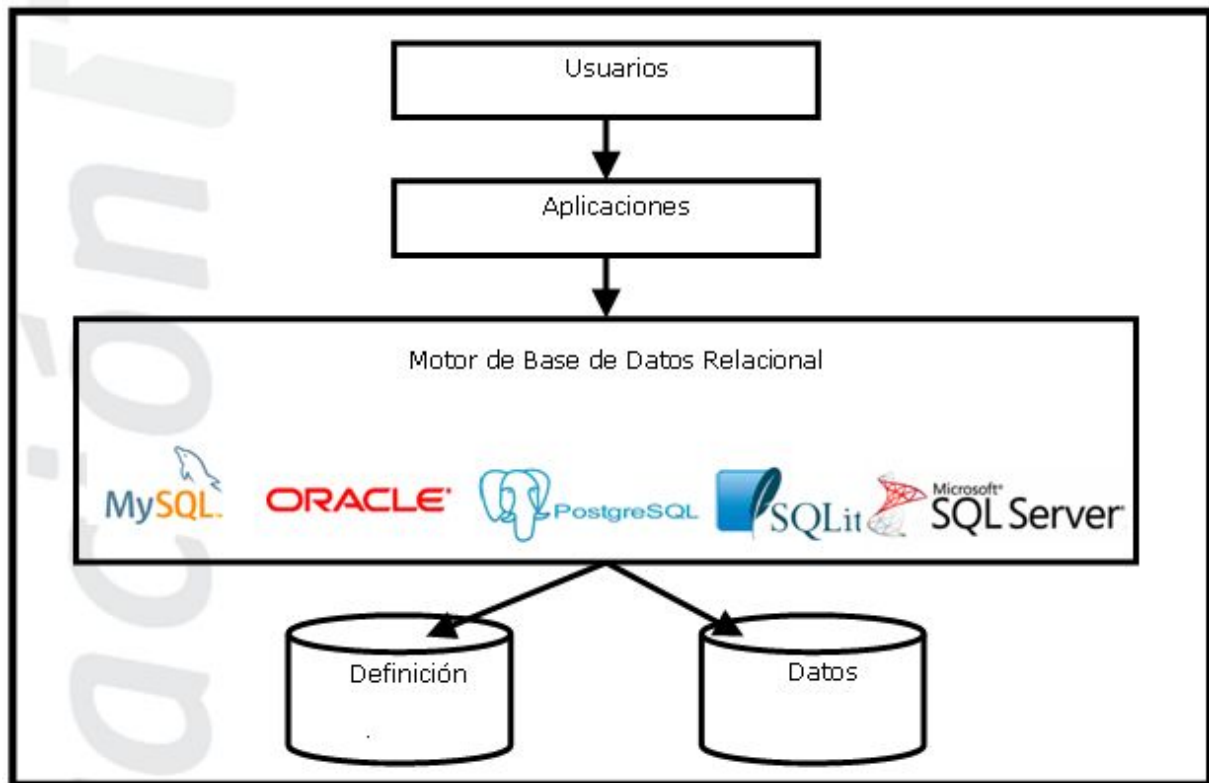
Las principales marcas del mercado son ORACLE, MySQL, SQL Server, PostgreSQL, SQLite.



Repositorio de datos

1. Un repositorio, depósito, o archivo de datos es un sitio centralizado donde se almacena y mantiene información digital, habitualmente de bases de datos o archivos informáticos.
2. Los depósitos de datos pueden distribuirse habitualmente sirviéndose de una red informática como Internet o en un medio físico. Pueden ser de acceso público o pueden estar protegidos y necesitar de una autenticación previa. Los depósitos más conocidos son los de carácter académico e institucional.
3. A diferencia de los ordenadores personales o de escritorio, los depósitos suelen contar con sistemas de respaldo (Backup) para mantenimiento preventivo y correctivo, lo que hace que la información se pueda recuperar en el caso que la máquina quede inutilizable.
4. Los depósitos se utilizan de forma intensiva en sistemas GNU/Linux, almacenando en su mayoría paquetes de software disponibles para su instalación mediante un gestor de paquetes.

Enfoque de una base de datos



- **Dato:** Unidad mínima de información, sin sentido en sí misma, pero que adquiere significado en conjunto con otras precedentes de la aplicación que las creó. Es un conjunto de símbolos que unidos de cierta forma dan un significado lógico.
- **Definición de Datos:** realiza una descripción de la estructura de los datos (su tipología, la forma en que se relacionan, etc.), de las operaciones que pueden realizarse con ellos (añadir, eliminar, modificar, recuperar) y de las restricciones referentes a su integridad (aquellas condiciones que todos los datos deben respetar para que se transmita la información de manera correcta).
- **Motores de bases de datos relacional:** son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los motores de bases de datos es el de gestionar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.
- **Aplicaciones:** usan el motor de datos para acceder a la información y luego poder presentarla al usuario. Las aplicaciones son desarrolladas por desarrolladores (programadores) y pueden ser escritas en diversos lenguajes. Los lenguajes más comunes son **JAVA, PHP, Visual Basic, C#, C++**, etc.
- **Usuarios:** Son las personas que pueden acceder a los datos. Ellos pueden ser catalogados como Usuarios Comunes, Usuarios Avanzados, Desarrolladores, Administradores.

Control de concurrencias

Debido a que los sistemas de bases de datos brindan servicios en forma simultánea a una comunidad de usuarios y desarrolladores, que normalmente están por fuera del ordenador que posee la base de datos (conectados vía red local, Internet, Intranet, etc.), se hace imprescindible contar con un control de concurrencia que revise la consistencia de cada operación y asegura que no se creen conflictos por usar en forma de escritura un mismo registro en forma simultánea.

Conceptos Preliminares

Tablas

- Las tablas son objetos compuestos por una estructura (conjunto de columnas) que almacenan información interrelacionada (filas) acerca de algún objeto en general.
- Las tablas tienen un solo nombre y es único en toda la base de datos.
- Están compuestas por registros (filas) y campos (columnas).
- Los registros y campos pueden estar en diferentes órdenes.
- Una base de datos puede contener varias tablas, y cada tabla almacena información con respecto a un objeto en particular.

Código	Nombre	Precio	Stock
1	iPod	299	200
2	iPhone	399	300
3	iPad	499	250
4	MacBook Pro	1199	150

Ejemplo de tabla de 4 campos (columnas) y 4 registros

Tipos de datos

Al diseñar nuestras tablas tenemos que especificar el tipo de datos y tamaño que podrá almacenar cada campo. Una correcta elección debe procurar que la tabla no se quede corta en su capacidad, que destine un tamaño apropiado a la longitud de los datos, y la máxima velocidad de ejecución.

Básicamente MySQL admite dos tipos de datos: números y cadenas de caracteres. Junto a estos dos grandes grupos, se admiten otros tipos de datos especiales: formatos de fecha, etc.

Datos numéricos

En este tipo de campos sólo pueden almacenarse números, positivos o negativos, enteros o decimales, en notación hexadecimal, científica o decimal.

Los tipos numéricos tipo **integer** admiten los atributos **SIGNED** y **UNSIGNED** indicando en el primer caso que pueden tener valor negativo, y solo positivo en el segundo.

Los tipos numéricos pueden además usar el atributo **ZEROFILL** en cuyo caso los números se completarán hasta la máxima anchura disponible con ceros (columna INT(5) zerofill => valor 23 se almacenará como 00023).

BIT o BOOL	Para un número entero que puede ser 0 ó 1
TINYINT	Es un número entero con rango de valores válidos desde -128 a 127. Si se configura como unsigned (sin signo), el rango de valores es de 0 a 255
SMALLINT	Para números enteros, con rango desde -32768 a 32767. Si se configura como unsigned, 0 a 65535
MEDIUMINT	Para números enteros. El rango de valores va desde -8.388608 a 8388607. Si se configura como unsigned, 0 a 16777215
INT	Para almacenar números enteros, en un rango de -2147463848 a 2147483647. Si configuramos este dato como unsigned, el rango es 0 a 4294967295
BIGINT	Número entero con rango de valores desde -9223372036854775808 a 9223372036854775807. Unsigned, desde 0 a 18446744073709551615
FLOAT (m,d)	Representa números decimales. Podemos especificar cuántos dígitos (m) pueden utilizarse (término también conocido como ancho de pantalla), y cuantos en la parte decimal (d). MySQL redondeará el decimal para ajustarse a la capacidad
DOUBLE	Número de coma flotante de precisión doble. Es un tipo de datos igual al anterior cuya única diferencia es el rango numérico que abarca
DECIMAL	Almacena los números como cadenas

Caracteres o cadenas de texto

Las Cadenas de texto se utilizan para almacenar una serie de caracteres, palabras y/o frases de texto en donde cada carácter es lo mismo que un byte.

CHAR	Este tipo se utiliza para almacenar cadenas de longitud fija. Su longitud abarca desde 1 a 255 caracteres. Un campo CHAR ocupará siempre el máximo de longitud que le hayamos asignado, aunque el tamaño del dato sea menor (añadiendo espacios adicionales que sean precisos).
VARCHAR	Al igual que el anterior se utiliza para almacenar cadenas, en el mismo rango de 1-255 caracteres, pero en este caso, de longitud variable. VARCHAR solo almacena la longitud del dato, permitiendo que el tamaño de la base de datos sea menor ^(*)
TINYTEXT	Texto de longitud variable que puede tener hasta 255 caracteres
TEXT	Texto de longitud variable que puede tener hasta 65535 caracteres
MEDIUMTEXT	Texto de longitud variable que puede tener hasta 16777215 caracteres
LONGTEXT	Texto de longitud variable que puede tener hasta 4294967295 caracteres
BLOB	Dato binario que puede almacenar archivos o texto. En este caso, los tipos TINYBLOB, BLOB, MEDIUMBLOB y LONGBLOB son idénticos a sus homólogos TEXT, con la diferencia de que las búsquedas en un tipo BLOB tienen en cuenta las mayúsculas y minúsculas

() El acceso a los datos CHAR es más rápido que VARCHAR. No pueden alternarse columnas CHAR y VARCHAR en la misma tabla. MySQL cambiará las columnas CHAR a VARCHAR. También cambia automáticamente a CHAR si usamos VARCHAR con valor de 4 o menos.*

Otros Tipos de Datos

DATE	Para almacenar fechas. El formato por defecto es YYYY MM DD desde 0000 00 00 a 9999 12 31
DATETIME	Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas:minutos:segundos
TIME	Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'
YEAR	Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.

Nota: Existen otros tipos de datos que no estamos utilizando en este curso

Estructura del Lenguaje SQL

SQL es un lenguaje de consulta estructurado (por sus siglas en inglés Structured Query Language) declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar (de una forma sencilla) información de interés de una base de datos, así como también hacer cambios sobre ella.

Existen dos clasificaciones del lenguajes para el manejo de bases de datos según el propósito: **DDL** y **DML**.

Creación de una Base de Datos

Sintaxis

```
CREATE DATABASE ComercioIT;
```

Creación de una Tabla

La sentencia **CREATE TABLE** creará una tabla con las columnas que indiquemos.

Sintaxis

```
CREATE TABLE Productos
(
    idProducto INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    Nombre     VARCHAR(50)      NOT NULL,
    Precio     DOUBLE           NULL,
    Marca      VARCHAR(30)      NOT NULL,
    Categoria  VARCHAR(30)      NOT NULL,
    Stock      INT(6)           NOT NULL,
    Disponible BOOLEAN          NULL DEFAULT false
);
```

Eliminar una Tabla

Sintaxis

```
DROP TABLE Productos;
DROP TABLE IF EXISTS Productos;
```

*Nota: la cláusula **IF EXISTS** devuelve una Advertencia en caso de que no exista la tabla a eliminar*

Restricciones de las Tablas

Puntos claves

- 1- Los nombres de las tablas deben ser únicos en la base de datos.
- 2- Los nombres de las columnas debe ser únicos en la tabla.
- 3- No pueden haber dos registros con el mismo valor de la clave primaria.

Columnas No Descomponibles

- 1- Son aquellas columnas que contienen cierta información que no puede ser en dos o más columnas.
- 2- Son fáciles de actualizar.
- 3- Son fáciles de consultar.
- 4- Mejores para mantener la integridad de los datos.

Restricciones en las columnas

- 1- NOT NULL (no permite valores nulos, que es lo mismo que estar vacía).
- 2- NO DUPLICATES (no permite nombres de columnas duplicadas).

Clave Primaria (PRIMARY KEY)

Una tabla suele tener una columna o una combinación de columnas cuyos valores **identifican de forma única** a cada registro de la tabla. Estas columnas se denominan **claves principales** de la tabla y exigen la integridad de

entidad de la tabla (un solo registro con ese valor de indicador único). Puede crear una clave principal mediante la definición de una restricción **PRIMARY KEY** cuando cree o modifique una tabla.

Una tabla sólo puede tener una restricción **PRIMARY KEY** y ninguna columna a la que se aplique una restricción **PRIMARY KEY** puede aceptar valores **NULL**. Debido a que las restricciones **PRIMARY KEY** garantizan datos únicos, con frecuencia se definen en una columna de identidad.

Cuando se especifica una restricción del tipo **PRIMARY KEY** en una tabla, el Motor de Base de Datos exige la unicidad de los datos mediante la creación de un índice único para las columnas de la clave principal. Este índice también permite un acceso rápido a los datos cuando se utiliza la clave principal en las consultas. De esta forma, las claves principales que se eligen deben seguir las reglas para crear índices únicos.

Si se define una restricción **PRIMARY KEY** para más de una columna, puede haber valores duplicados dentro de la misma columna, pero cada combinación de valores de todas las columnas de la definición de la restricción **PRIMARY KEY** debe ser única.

Anexo: Otros Comandos MySQL

En las siguientes líneas se detallan comando necesarios para la navegación dentro del motor MySQL que no se encuentran contemplados en el ANSI SQL.

Comando SHOW DATABASES

Muestra el catálogo de base de datos del servidor.

Sintaxis

```
SHOW DATABASES;
```

Comando USE

Activa una base de datos, para su uso. Es obligatorio tener una base de datos activa para poder operar.

Sintaxis

```
USE NombreDeBaseDeDatos;
```

Comando SHOW TABLES

Muestra el catálogo de tablas de la base de datos.

Sintaxis

```
SHOW TABLES;
```

Comando DESCRIBE

Devuelve la descripción de campos y detalles de una tabla.

Sintaxis

```
DESCRIBE NombreDeTabla;
```

