

Módulo 5



Antonio Cano Galindo

Según la Wikipedia:

Una base 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 para su consulta.

- **Jerárquicas**
- **En red**
- **Relacionales**
- **Documentales**
- **Multidimensionales**
- **Orientadas a objetos**

- Una base de datos se compone de varias tablas, denominadas relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de campos (columnas) y registros (filas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las llaves primarias y llaves foráneas (o ajenas).
- Las llaves primarias son la clave principal de un registro dentro de una tabla y estas deben cumplir con la integridad de datos.
- Las llaves ajenas se colocan en la tabla hija, contienen el mismo valor que la llave primaria del registro padre; por medio de estas se hacen las formas relacionales.

Las bases de datos relacionales se normalizan para:

- Minimizar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las tablas.
- Proteger la integridad de datos.

En el modelo relacional es frecuente llamar «tabla» a una relación; para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada tabla debe tener su nombre único.
- No puede haber dos filas iguales. No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo.

Antonio Cano Galindo

- **Clave candidata:** es el conjunto mínimo de columnas que identifica unívocamente a cada fila
- **Clave primaria:** clave candidata designada como principal
- **Clave compuesta:** cuando una clave está formada por más de un campo
- **Clave ajena:** aquella columna que existiendo como dependiente en una tabla, es a su vez clave primaria en otra tabla

- **Todos los atributos son «atómicos».** Por ejemplo, en el campo teléfono no tenemos varios teléfonos.
- **La tabla contiene una clave primaria única.** Por ejemplo: el NIF para personas, la matrícula para vehículos o un simple id autoincremental. Si no tiene clave, no es 1FN.
- **La clave primaria no contiene atributos nulos.** No podemos tener filas para las que no haya clave (por ejemplo, personas sin NIF o vehículos sin matrícula).
- **No debe existir variación en el número de columnas.** Si algunas filas tienen 8 columnas y otras 3, pues no estamos en 1FN.
- **Los campos no clave deben identificarse por la clave.** Es decir, que los campos no clave dependen funcionalmente de la clave. Esto es prácticamente lo mismo que decir que existe clave primaria.
- **Debe Existir una independencia del orden tanto de las filas como de las columnas,** es decir, si los datos cambian de orden no deben cambiar sus significados.

Formas normales: 2FN y 3FN



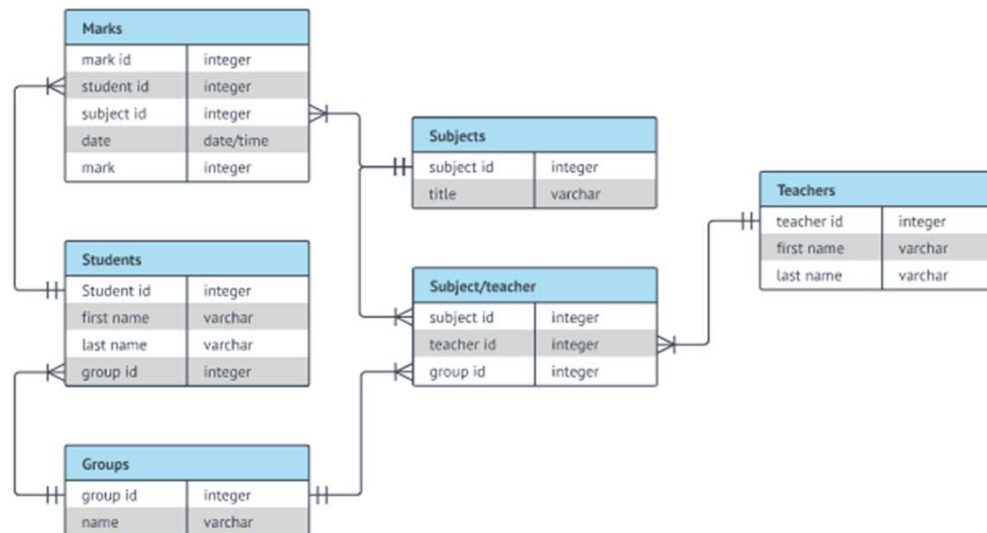
- **2FN:** una tabla está en 2FN si además de estar en 1FN cumple que los atributos no clave depende de TODA la clave principal.
- **3FN:** Una tabla está en 3FN si además de estar en 2FN no existe ninguna dependencia transitiva entre los atributos que no son clave.
- **Resumen**
 - **1FN.** No elementos repetidos o grupos de elementos
 - **2FN.** Sin dependencias parciales de llaves concatenadas
 - **3FN.** Sin dependencias de atributos que no son llaves

Antonio Cano Galindo

Diagramas Entidad-Relación



- Muestra relaciones entre entidades de información del sistema a modelar.
- Ayuda a conceptualizar elementos abstractos y entender la relación entre ellos.
- Está compuesto de entidades (cajas) y relaciones (conectores) con cardinalidad.



Antonio Cano Galindo

MariaDB es un fork de MySql.

Su creador es Michael (Monty) Widenius, creador también de MySQL.

El fork se produce a raíz de la venta de SUN Microsystem/MySQL a Oracle.

Se caracteriza por la alta compatibilidad con MySQL, asegurando que siempre será gratuito y open source.

Usaremos la instalación de MariaDB contenida en el paquete XAMPP.



PhpMyAdmin y MySQL Workbench



Son sistemas software de administración de bases de datos.

MySQL Workbench es un software de escritorio gratuito descargable desde la web de MySQL.

PhpMyAdmin es un portal web realizado en Php que permite la administración de sistemas basados en MySQL.

Tienen un grado de funcionalidad muy parecida.



Antonio Cano Galindo

Características de PhpMyAdmin



- Interfaz Web para la gestión gráfica.
- Administración y mantenimiento de base de datos MySQL y MariaDB.
- Explorar, eliminar bases de datos, tablas, vistas, campos e índices.
- Crear, copiar, eliminar, renombrar y modificar bases de datos, campos e índices.
- Mantenimiento de servidor, bases de datos y tablas, de cara a la configuración del servidor.
- Ejecutar, editar y marcar cualquier instrucción SQL, incluso peticiones por lotes.
- Administrar procesos almacenados.
- Importación de datos desde [CSV](#) y [SQL](#)
- Exporta datos a diferentes formatos: [CSV](#), [SQL](#), [XML](#), etc.
- Administración de múltiples servidores.
- Crea gráficos PDF del diseño de la base de datos.
- Crea consultas complejas usando *Query-by-Example* (QBE).
- Búsqueda global en una base de datos o un subconjunto de esta.
- *Live charts* para monitorear las actividades del servidor MySQL tales como conexiones, procesos, uso de CPU y/o memoria, etc.

Antonio Cano Galindo

DDL: Data Definition Language

- Base de datos (database)
- Tabla y campo (table and field)
- Tipo de dato (data type)
- Clave primaria y ajena (primary and foreign key)
- Índice (index)
- Disparador (trigger)
- Procedimiento almacenado (stored procedure)
- Vistas (view)

DML: Data Manipulation Language

- Select
- Insert
- Update
- Delete
- Truncate
- Replace

Comando CREATE TABLE

```
CREATE TABLE `cliente` (  
  `idCliente` int(10) UNSIGNED NOT NULL,  
  `nombre` varchar(30) NOT NULL,  
  `direccion` varchar(30) NOT NULL,  
  `poblacion` varchar(30) NOT NULL,  
  `provincia` varchar(30) NOT NULL,  
  `dni` char(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Una tabla está compuesta de campos, donde tenemos que especificar para cada uno de ellos el tipo de dato y algunas características más.

Antonio Cano Galindo

Valor especial que indica la ausencia de valor en un campo.
Los campos que forman parte de una clave primaria no puede permitir nulos.

```
CREATE TABLE `cliente` (  
  `idCliente` int(10) UNSIGNED NOT NULL,  
  `nombre` varchar(30) NULL,  
  `dni` char(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

El campo nombre permite guardar null como valor.

En una sentencia **INSERT** los campos no especificados asumen el valor null, siempre y cuando está definidos como null. En caso contrario, **INSERT** fallará.

Numéricos

Enteros: INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT

Reales: FLOAT, DOUBLE, DECIMAL

Fecha y hora

DATE, DATETIME, TIMESTAMP, TIME, YEAR

Cadenas

CHAR, VARCHAR, BLOB, TINYBLOB, MEDIUMBLOB, LONGBLOB, ENUM

Clave primaria



La clave primaria puede contener uno o varios campos
Si tiene más de un campo sería una clave primaria compuesta.
No permite que ningún campo pueda contener nulos

```
ALTER TABLE `facturadetalle`  
  ADD PRIMARY KEY (`idFactura`, `idProducto`);
```

La creación de una primary key implica la existencia de un índice único con esos campos.

Permite relacionar una tabla “padre” con una “hija” mediante la clave primaria de la tabla padre en relación a los mismos campos de la tabla hija.

Cuando borramos o actualizamos el registro de la tabla padre, se aplican las reglas de consistencia **ON DELETE** o **ON UPDATE: CASCADE, SET NULL, RESTRICT** o **NO ACTION**.

```
ALTER TABLE `facturadetalle`  
  ADD CONSTRAINT `factura` FOREIGN KEY (`idFactura`) REFERENCES  
  `factura` (`idFactura`),  
  ADD CONSTRAINT `producto` FOREIGN KEY (`idProducto`) REFERENCES  
  `producto` (`idProducto`) ON DELETE CASCADE ON UPDATE CASCADE;
```

La creación de una foreign key implica la existencia de un índice con esos campos.

Un índice es una estructura de datos que mejora la velocidad de acceso a los datos de una tabla.

Similar al índice de un libro. Guarda el valor a indexar y su posición en la tabla.

Pueden contener una o varias columnas.

Puede ser único o permitir duplicados.

Un índice FULLTEXT definido sobre campos de tipo cadenas de caracteres permite la búsqueda de palabras dentro de un campo.

```
ALTER TABLE `cliente`  
  ADD KEY `nombre` (`nombre`);
```

Devuelve un conjunto de datos de una o más tablas

Especifica los campos que contendrá la tabla así como la lista de tablas implicadas y las condiciones que deben cumplir los registros resultado.

```
SELECT `tabla 1`.`campo 1`, ..., `tabla n`.`campo n`, <expresión>
  FROM `tabla 1`, `tabla 2`, ..., `tabla n`
 WHERE <condiciones>
 ORDER BY `tabla 1`.`campo 1` [ASC|DESC]
 LIMIT <número de filas>
```

En <condiciones> especificamos las restricciones que deberá cumplir el resultado. Los campos en ORDER BY son los usados para ordenar la tabla resultado.

SELECT – Funciones internas



MAX : Máximo de un conjunto de valores.

MIN : Mínimo de un conjunto de valores.

AVG : Media de un conjunto de valores.

SUM : Suma total de un conjunto de valores.

COUNT : Cuenta el número de valores que hay en un conjunto.

```
SELECT MAX(population), COUNT(*)  
  FROM city  
 WHERE countrycode = "ESP"
```

Devuelve el máximo del campo población y el número de registros cuyo countrycode sea igual a "ESP".

Antonio Cano Galindo

SELECT – Otras funciones internas



CURRENT_TIMESTAMP () Fecha y hora actuales.

LOWER (cadena) Convierte una cadena de tipo varchar a minúsculas.

SUBSTRING (fuente, n, lon) Extrae de la cadena fuente una subcadena , comenzando en el carácter n-ésimo, con una longitud lon

UPPER (cadena) Convierte una cadena de tipo varchar a mayúsculas.

LEFT (cadena, lon) Devuelve de la cadena una subcadena, comenzando por la izquierda y con una longitud lon.

LEN (cadena) Devuelve la longitud de la cadena.

LTRIM (cadena) Quita los blancos de la izquierda en la cadena.

RTRIM (cadena) Quita los blancos de la derecha en la cadena.

STR (cadena) Devuelve la cadena alineada a la derecha.

RIGHT (cadena, num_elem) Devuelve el número de elementos de la cadena que están a la derecha.

RIGHT(CASA,2) = SA

ASCII (cadena) Devuelve el nº ASCII correspondiente a la cadena

Antonio Cano Galindo