

MP_0486. Acceso a datos

**UF2. Manejo de conectores a
bases de datos relacionales**

2.1. Introducción a las bases de datos



Índice

Introducción a la unidad formativa	4
Objetivos.....	4
Introducción a las bases de datos	5
Definición de base de datos.....	5
Modelos de base de datos	6
Estructura de una base de datos.....	6
Dominio de los campos o atributos.....	8
El lenguaje SQL	8
La sentencia SQL <i>INSERT</i>	8
La sentencia SQL <i>SELECT</i>	9
La sentencia SQL <i>UPDATE</i>	9
La sentencia SQL <i>DELETE</i>	9
Los sublenguajes de SQL	10
Despedida	11
Resumen.....	11

El contenido de esta lección ya lo estudiaste en la asignatura de Programación, pero te recomendamos que vuelvas a estudiarlo como repaso, ya que te ayudará a continuar con éxito el resto de contenidos y a realizar las actividades propuestas sin dificultad.

Adelante, te costará poco esfuerzo recordar estos conceptos y volverlos a poner en la práctica.

Introducción a la unidad formativa

Objetivos

Con esta unidad perseguimos los siguientes objetivos:

- Definir el término "base de datos".
- Enumerar y describir todos los conceptos asociados a las bases de datos: tablas, filas o registros, atributos o campos, dominio de los atributos, etc.

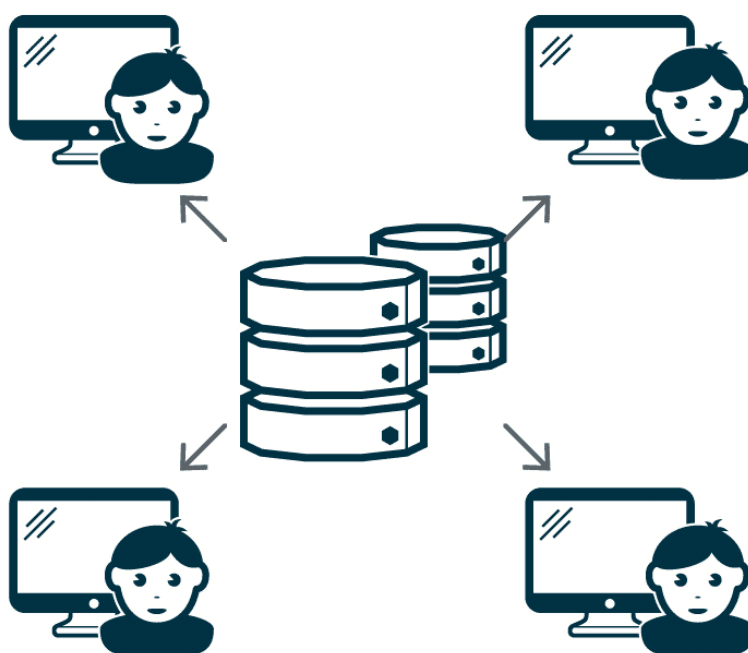
¡Ánimo y adelante!

Introducción a las bases de datos

Definición de base de datos

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que, posteriormente, podamos encontrarla y utilizarla fácilmente.

Una base de datos es gestionada por un programa gestor de bases de datos (**DBMS** - Data Base Management System o **SGBD** - Sistema Gestor de Base de Datos).



El DBMS garantiza que la base de datos esté disponible para los usuarios y programas que la requieran.

Algunos ejemplos de DBMS son:

- Microsoft SQL Server.
- MySQL.
- Oracle Database.
- Postgre SQL.
- Microsoft Access.
- IBM DB2.

Modelos de base de datos

Un modelo de datos determina la estructura lógica en la que una base de datos va a ser construida. Por lo tanto, determina de qué manera los datos son almacenados, organizados y manipulados. Cada DBMS trabaja con un modelo de base de datos determinado. Existen numerosos modelos que determinan la estructura lógica de una base de datos. Entre ellos:

- Modelo jerárquico.
- Modelo en red.
- Modelo de estrella.
- Modelo relacional.
- Modelo de base de datos orientada a objetos.

En esta lección estudiaremos el modelo relacional, donde una base de datos está formada por un conjunto de tablas o relaciones de datos.

Estructura de una base de datos

Una base de datos está formada por una o más tablas, donde cada tabla guarda una relación de entidades o elementos del mismo tipo, por ejemplo: clientes, facturas, productos, etc.

CLIENTES				
NIF	NOMBRE	DOMICILIO	TLF	CIUDAD
43434343-A	DELGADO PEREZ MARISA	C/MIRAMAR, 84 3ºA	925-200-967	TOLEDO
51592939-K	LOPEZ VAL SOLEDAD	C/PEZ, 54 4ºC	915-829-394	MADRID
51639989-K	DELGADO ROBLES MIGUEL	C/OCA, 54 5ºC	913-859-293	MADRID
51664372-R	GUTIERREZ PEREZ ROSA	C/CASTILLA, 4 4ºA	919-592-932	MADRID

Tabla que guarda una relación de clientes. Cada fila contiene la información de un cliente.

Características de una tabla

- Cada tabla se compone de **filas y columnas**.
- **Cada fila de una tabla conforma un registro** que guarda la información completa sobre una determinada entidad o elemento. En nuestro ejemplo de la imagen anterior cada registro contiene la información de un cliente.
- Las **columnas** guardan una información concreta sobre cada elemento: un nombre, un DNI, etc. A las columnas las denominamos **campos**.

Fila o registro →

CLIENTES				
NIF	NOMBRE	DOMICILIO	TLF	CIUDAD
43434343-A	DELGADO PEREZ MARISA	C/MIRAMAR, 84 3ªA	925-200-967	TOLEDO
51592939-K	LOPEZ VAL SOLEDAD	C/PEZ, 54 4ªC	915-829-394	MADRID
51639989-K	DELGADO ROBLES MIGUEL	C/OCA, 54 5ªC	913-859-293	MADRID
51664372-R	GUTIERREZ PEREZ ROSA	C/CASTILLA, 4 4ªA	919-592-932	MADRID

↑
Columna o campo

Cada tabla suele tener una clave principal, que identifica a cada uno de los registros. La clave principal no podrá tener valores duplicados.

CLIENTES				
NIF	NOMBRE	DOMICILIO	TLF	CIUDAD
43434343-A	DELGADO PEREZ MARISA	C/MIRAMAR, 84 3ªA	925-200-967	TOLEDO
51592939-K	LOPEZ VAL SOLEDAD	C/PEZ, 54 4ªC	915-829-394	MADRID
51639989-K	DELGADO ROBLES MIGUEL	C/OCA, 54 5ªC	913-859-293	MADRID
51664372-R	GUTIERREZ PEREZ ROSA	C/CASTILLA, 4 4ªA	919-592-932	MADRID

↑
El NIF sirve para identificar inequívocamente a cada cliente.

La clave principal de una tabla sirve además para establecer interrelaciones con otras tablas de la base de datos. Una tabla de facturas podría tener entre sus campos el NIF del cliente al que corresponde dicha factura, así se establece una asociación entre los clientes y sus facturas.

CLIENTES				
NIF	NOMBRE	DOMICILIO	TLF	CIUDAD
43434343-A	DELGADO PEREZ MARISA	C/MIRAMAR, 84 3ªA	925-200-967	TOLEDO
51592939-K	LOPEZ VAL SOLEDAD	C/PEZ, 54 4ªC	915-829-394	MADRID
51639989-K	DELGADO ROBLES MIGUEL	C/OCA, 54 5ªC	913-859-293	MADRID
51664372-R	GUTIERREZ PEREZ ROSA	C/CASTILLA, 4 4ªA	919-592-932	MADRID

FACTURAS				
NUMERO	FECHA	PAGADO	NIF	IMPORTE
5440	05-sep-11	Sí	43434343A	27,00 €
5441	05-sep-11	Sí	51639989K	53,00 €
5442	06-sep-11	No	43434343A	85,00 €
5443	10-oct-11	Sí	51639989K	43,00 €
5444	13-oct-11	Sí	51664372R	12,00 €
5445	14-oct-11	No	43434343A	18,00 €

Dominio de los campos o atributos

Como vimos en el apartado anterior, cada columna de una tabla se corresponde con un campo, también denominado atributo.

El dominio de un campo es el conjunto de valores posibles que puede tomar dicho campo.

Cuando se define el diseño de una tabla **hay que especificar el tipo de dato que corresponde a cada uno de los campos**. Además del tipo de dato, también pueden definirse restricciones. Por ejemplo, el tipo de dato del campo *IMPORTE* de la tabla *FACTURA* es numérico, y una restricción podría ser que no admita números negativos. El dominio del campo viene finalmente determinado por el tipo de datos y otras restricciones aplicadas.

En los siguientes apartados verás los tipos de datos que se pueden establecer en una base de datos relacional creada en MySQL. Si bien en otros DBMS pueden variar ligeramente.

El lenguaje SQL

SQL es un lenguaje estándar para almacenar, manipular y recuperar información de una base de datos. Al conjunto de operaciones que podemos realizar con el lenguaje SQL se le denomina **CRUD** abreviatura de:

1. **Create:** crear nuevos registros o filas. Se lleva a cabo con la sentencia **INSERT**.
2. **Read:** leer filas. Se lleva a cabo con la sentencia **SELECT**.
3. **Update:** modificar filas. Se lleva a cabo con la sentencia **UPDATE**.
4. **Delete:** borrar filas. Se lleva a cabo con la sentencia **DELETE**.

La sentencia SQL *INSERT*

La sentencia *INSERT* se utiliza para añadir nuevas filas o registros a una tabla. Tiene el siguiente formato:

INSERT INTO nombre_tabla[(campo_1,...,campo_m)] VALUES [(campo_1,...,campo_m)]

```
INSERT INTO CLIENTES (NIF, NOMBRE, DOMICILIO, TLF, CIUDAD)
VALUES ('51639936K', 'MALDONADO GÓMEZ CARLOS', 'C/ LUNA, 48', '913683060',
'MADRID');
```

Añade una nueva fila en la tabla *CLIENTES*.

La sentencia SQL *SELECT*

La sentencia *SELECT* se utiliza para recuperar un conjunto de filas a partir de una o varias tablas. Tiene el siguiente formato:

SELECT campos_a_seleccionar FROM tabla [WHERE condición]

```
SELECT NOMBRE, TLF FROM CLIENTES  
WHERE CIUDAD='MADRID';
```

Devuelve una relación con el nombre y teléfono de los clientes de Madrid.

NOMBRE	TLF
LOPEZ VAL SOLEDAD	915-829-394
DELGADO ROBLES MIGUEL	913-859-293
GUTIERREZ PEREZ ROSA	919-592-932

La sentencia SQL *UPDATE*

La sentencia *UPDATE* se utiliza para modificar uno o varios campos de uno o varios registros. Tiene el siguiente formato:

*UPDATE nombre_tabla SET [campo_1 = valor_1, ..., campo_m = valor_m]
[WHERE condición]*

```
UPDATE CLIENTES SET DOMICILIO='C/ SANTA ENGRACIA, 17 3ºA', TLF='913802060'  
WHERE NOMBRE='DELGADO PEREZ MARISA';
```

Modifica el domicilio y teléfono del cliente 'DELGADO PEREZ MARISA'.

La sentencia SQL *DELETE*

La sentencia *DELETE* se utiliza para eliminar una o varias filas o registros. Tiene el siguiente formato:

DELETE FROM nombre_tabla [WHERE condición]

```
DELETE FROM CLIENTE WHERE NOMBRE='DELGADO PEREZ MARISA';
```

Elimina la fila correspondiente al cliente 'DELGADO PEREZ MARISA'.

Las sentencias SQL deben finalizar con un punto y coma. No es obligatorio si solo vamos a ejecutar una sentencia, pero cuando vamos a ejecutar varias sentencias SQL en bloque, el punto y coma sirve como delimitador para marcar que termina una sentencia y comienza la siguiente.

Los sublenguajes de SQL

SQL además está dividido en dos sublenguajes:

- **DDL** (Data Definition Language): formado por el conjunto de sentencias SQL que tienen que ver con la creación y modificación de la estructura de la base de datos (crear tablas, borrar tablas, etc). Las sentencias que forman parte de este sublenguaje son: *CREATE*, *ALTER* y *DROP*.
- **DML** (Data Management Language): formado por el conjunto de sentencias SQL que tienen que ver con la manipulación de los datos (añadir registros, borrar registros, modificar registros, recuperar registros). Las sentencias que forman parte de este subconjunto son: *INSERT*, *SELECT*, *UPDATE* y *DELETE*.

Despedida

Resumen

Has terminado la lección, vamos a ver los puntos más importantes que hemos tratado.

- **Una base de datos es un "almacén"** que nos permite guardar grandes cantidades de información de forma organizada para que luego la podamos encontrar y utilizar fácilmente.
- Una base de datos está formada por una o más **tablas**.
- Cada tabla está compuesta por **filas (registros)** y **columnas (campos)**.
- Cada tabla suele tener una **clave principal**, que identifica a cada uno de los registros. La clave principal no podrá tener valores duplicados. La clave principal de una tabla sirve además para establecer interrelaciones con otras tablas de la base de datos.
- **SQL** es un lenguaje estándar para almacenar, manipular y recuperar información de una base de datos.
- Al conjunto de operaciones que podemos realizar con el lenguaje SQL se le denomina **CRUD** (*Create, Read, Update, Delete*).