TEMA 5. SQL (I)

LENGUAJE DDL.
CREACIÓN Y GESTIÓN DE TABLAS

IES Valle del Jerte PLASENCIA Bilingual Section

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

INTRODUCCIÓN

- SQL es el lenguaje fundamental de los SGBD Relacionales.
- Es uno de los lenguajes más utilizados de la historia de la Informática.
- Es un lenguaje declarativo, define lo que se va a hacer por encima del cómo se va a hacer.
- Agrupa todas las funciones que se le pueden pedir a una Base de Datos, por lo que es utilizado tanto por administradores como por programadores y usuarios.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

HISTORIA DEL LENGUAJE SQL

- Nació en 1970 y fue creado por Codd.
- Dos años después, IBM adopta las directrices formuladas por Codd y crea el Standard English Query Lenguage (Lenguaje Estándar Inglés para Consultas) SQUEL.
- Más tarde cambió su nombre por SQL.
- En 1979 Oracle presenta la primera implementación comercial del lenguaje.
- Poco después se convierte en el estándar en Bases de datos avalado por lo organismos internacionales de estandarización ISO y ANSI.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

VERSIONES DE SQL

- 1989: Aparece el estándar ISO y ANSI SQL89 O SQL1.
- 1992: Aparece la versión más conocida de SQL, llamada SQL92 o SQL2
- 1999: SQL99 incorpora mejoras que incluyen triggers, procedimientos, funciones, …
- 2006: SQL2006, SQL puede utilizarse conjuntamente con XML.
- 2011: SQL2011, incluye mejoras en las funciones de ventana y de la cláusula FETCH.
- 2016: SQL2016, permite búsqueda de patrones, funciones de tabla polimórficas y compatibilidad con los ficheros JSON

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

PROCESO DE LAS INSTRUCCIONES SQL

Hay distintas maneras de trabajar con SQL:

- SQL INTERACTIVO: las instrucciones se introducen a través de un cliente conectado directamente al servidor SQL.
- SQL EMBEBIDO O INCRUSTADO: las instrucciones de SQL forman parte del código de otro lenguaje que se considera anfitrión (C, Java, Pascal, Cobol,...)
- SQL a través de clientes gráficos: un software que permite conectar a la BD y manejarla de forma gráfica. Más cómodo para usuarios.
- SQL dinámico: instrucciones de SQL incrustadas en módulos especiales que serán llamados desde aplicaciones.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

¿CÓMO SE PROCESA UNA INSTRUCCIÓN SQL?

1. Se analiza la instrucción para comprobar la sintaxis de la misma
2. Si es correcta se valora si los metadatos de la misma son correctos. Se comprueba esto en el diccionario de datos.
3. Si es correcta, se optimiza, a fin de consumir los mínimos recursos posibles.
4. Se ejecuta la sentencia y se muestra el resultado.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

TIPOS DE INSTRUCCIONES SQL

DDL **Instrucciones de Definición de Datos**	Crear objetos de la BD:CREATE Modificar objetos: ALTER Borrar objetos: DROP
DML **Instrucciones de Manipulación de datos**	Recuperar información: SELECT Actualizar información**:** Añadir INSERT Modificar UPDATE Borrar DELETE
DCL **Instrucciones de Control de acceso**	Dar privilegios de acceso a datos GRANT Quitar privilegios de acceso REVOKE Control de transacciones : ROLLBACK y COMMIT
Instrucciones SQL Programático	Uso de cursores (DECLARE,OPEN,FETCH,CLOSE) Procedimientos, funciones y disparadores Instrucciones repetitivas y de control de flujo

ELEMENTOS DEL CÓDIGO SQL

- Comandos. Las distintas instrucciones que se pueden realizar desde SQL . (SELECT, CREATE…)
- Cláusulas. Son palabras especiales que permiten modificar el funcionamiento de un comando (WHERE, ORDER BY,...)
- Operadores. Permiten crear expresiones complejas. Pueden ser aritméticos (+,-,*,/,...) lógicos (>, <, !=,<>, AND, OR,...)
- Funciones. Para conseguir valores complejos (SUM(), DATE(),...)
- Literales. Valores concretos para las consultas: números, textos, caracteres,... Ejemplos: 2, 12.34, calle el Rey…

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJEMPLOS DE SENTENCIA SQL

```
CREATE DATABASE PRUEBAS;
```

```
SELECT CURSO, NOMBRE, NOTA
FROM ALUMNOS
WHERE ASIGNATURA = ‘matematicas’;
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

“**Una manera de hacer Europa**”. Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

NORMAS DE ESCRITURA DE LAS SENTENCIAS SQL

- En SQL no se distingue entre mayúsculas y minúsculas.
- Las instrucciones finalizan con el signo de punto y coma.
- Se pueden tabular líneas para facilitar la lectura si fuera necesario.
- Comentarios:
  - una o varias líneas comentadas: /* */

```
/* esto es un comentario */
SELECT * FROM PERSONAS;
```

```
/* esto es
un comentario */
SELECT * FROM PERSONAS;
```

  - una única línea comentada: --

```
-- esto es un comentario
SELECT * FROM PERSONAS;
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

TIPOS DE DATOS

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

DATOS DE TIPO TEXTO

Los utilizaremos para los campos que van a almacenar cadenas de caracteres, ya sean de longitud fija o variable.

Tipos de datos	Descripción
CHARACTER (longitud) CHAR (longitud)	Cadenas de caracteres de longitud fija.
CHARACTER VARYING (longitud) VARCHAR(longitud)	Cadenas de caracteres de longitud variable.
NCHAR (longitud)	Cadenas de caracteres de longitud fija para caracteres nacionales
NVARCHAR(longitud)	Cadenas de caracteres de longitud variable para caracteres nacionales.
BLOB Binary Large Objects	Almacenan datos de gran tamaño que cambian de forma dinámica (imágenes, archivos de sonido y otros objetos multimedia) No todos los SGBD son compatibles con los BLOB.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

DATOS DE TIPO NUMÉRICO

Los utilizaremos para guardar datos que almacenen cantidades numéricas.

	Tipos de datos	Descripción
ENTEROS	SMALLINT	Enteros pequeños (2 bytes).
	INTEGER INT	Enteros normales (4 bytes)
	BIGINT	Enteros largos (8 bytes)
DECIMALES	FLOAT DOUBLE REAL	Decimal de coma variable
	NUMERIC (m,d) DECIMAL (m,d)	Decimal de coma fija m indica el nº de dígitos totales d indica el nº de decimales

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

DATOS DE TIPO FECHA

Los utilizaremos para almacenar información de fechas

Tipos de datos	Descripción
DATE	Fecha (año, mes y día)
DATETIME	Almacena una fecha (año-mes-día) y una hora (horas-minutos-segundos). Su rango oscila entre ‘1000-01-01 00:00:00’ y ‘9999-12-31 23:59:59’.
TIMESTAMP	Almacena una fecha y hora UTC. El rango de valores oscila entre ‘1970-01-01 00:00:01’ y ‘2038-01-19 03:14:07’.

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

“**Una manera de hacer Europa**”. Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

DATOS DE TIPO LÓGICO

Tipos de datos	Descripción
BOOLEAN	Toma valores verdadero / falso
BIT	Binario

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

Sentencias de Definición de Datos (DDL)

IES Valle del Jerte PLASENCIA Bilingual Section

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

DEFINICIÓN

- El DDL es la parte del lenguaje SQL que realiza la función de definición de datos del SGBD.
- Se encarga de la creación, modificación y eliminación de los objetos de la Base de Datos.
- Los objetos de la Base de Datos pueden ser: tablas, vistas, índices…
- Aunque las instrucciones de SQL suelen ser las mismas para todos los SGBD, existen algunas diferencias de sintaxis.
- Veremos su utilización en el SGBD MySQL.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CREACIÓN DE UNA BASE DE DATOS

CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_base_datos;

- DATABASE y SCHEMA son sinónimos.
- IF NOT EXISTS crea la base de datos sólo si no existe una base de datos con el mismo nombre.
- Si no especificamos el set de caracteres en la creación de la base de datos, se usará **latin1** por defecto.
- Las bases de datos que vamos a crear durante el curso usarán el set de caracteres utf8 o utf8mb4.
- Si dejamos todos los valores por defecto, la sintaxis a utilizar será:

CREATE DATABASE nombre_base_datos;

Ejemplo:

CREATE DATABASE EMPRESA

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

BORRADO DE UNA BASE DE DATOS

DROP {DATABASE | SCHEMA} [IF EXISTS] nombre_base_datos;

- DATABASE y SCHEMA son sinónimos.
- IF EXISTS elimina la la base de datos sólo si ya existe.

Ejemplo:

DROP DATABASE nombre_base_datos;

DROP DATABASE EMPRESA;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR UNA BASE DE DATOS

ALTER {DATABASE | SCHEMA} [nombre_base_datos]
  alter_specification [, alter_especification] ...

Ejemplo:

ALTER DATABASE EMPRESA CHARACTER SET utf8;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CONSULTAR EL LISTADO DE BASES DE DATOS DISPONIBLES Y CONECTAR CON UNA BD

SHOW DATABASES;

Muestra un listado con todas las bases de datos a las que tiene acceso el usuario con el que hemos conectado a MySQL.

USE nombre_base_datos;

- Antes de comenzar a trabajar con una BD debemos conectarnos a ella.
- El comando USE se utiliza para indicar la base de datos con la que queremos trabajar.

Ejemplo:

```
USE EMPRESA;
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CREACIÓN DE TABLAS

- Veremos una sintaxis resumida de creación de tablas, sólo con las opciones que usaremos a lo largo del curso.
- Consultar documentación en página oficial Mysql

```
CREATE TABLE nombre_tabla
(
    nombre_columna tipo [ NULL | NOT NULL ] [PRIMARY KEY]     [,Restricciones ... ]
 )
```

Ejemplo:

```
CREATE TABLE EMPLEADOS
 (
Cod_Emp INT PRIMARY KEY,
Nombre CHAR(20) NOT NULL
);
```


UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MOSTRAR LA ESTRUCTURA DE UNA DE UNA TABLA

Para ver la descripción o estructura de una tabla se utiliza la instrucción

DESCRIBE nombre_tabla

Ejemplo:

DESCRIBE EMPLEADOS;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS

1. RENOMBRAR UNA TABLA:

ALTER TABLE nombre RENAME TO NuevoNombre;

Ejemplo:

ALTER TABLE EMPLEADOS RENAME TO EMP;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS:
2. AÑADIR CAMPOS NUEVOS A UNA TABLA:

ALTER TABLE nombreTabla ADD(nombreColumna TipoDatos [Propiedades] [,columnaSiguiente tipoDatos [propiedades]...)

Por defecto las nuevas columnas se añaden al final de la tabla.

Ejemplo:

ALTER TABLE EMPLEADOS ADD SALARIO DECIMAL(10,2);

Algunos SGBD requieren escribir la palabra COLUMN tras la palabra ADD.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS:

2. AÑADIR CAMPOS NUEVOS A UNA TABLA EN UN LUGAR DETERMINADO DE LA TABLA:

ALTER TABLE nombreTabla
ADD(nombreColumna TipoDatos [Propiedades]
AFTER AFTER nombre_columna_que_va_antes

Las nuevas columnas se añaden DESPUÉS DE LA COLUMNA

Ejemplo:

ALTER TABLE EMPLEADOS
ADD SALARIO DECIMAL(10,2) AFTER APELLIDO;

.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS:
2. CAMBIAR LA POSICIÓN DE UNA COLUMNA EN UNA TABLA

ALTER TABLE tabla
MODIFY columna_a_mover TIPO_COLUMNA
AFTER nombre_columna_que_va_antes

Ejemplo:

```
ALTER TABLE EMPLEADOS
MODIFY SALARIO DECIMAL(10,2)
AFTER DIRECCION;
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS
3. BORRAR CAMPOS DE UNA TABLA

ALTER TABLE nombreTabla
DROP (columna [,columnaSiguiente,...]);

- Elimina la columna o columnas indicadas de manera irreversible.
- No se puede eliminar una columna si es la única columna que queda en la tabla.

Ejemplo:

ALTER TABLE EMPLEADOS DROP FECHA_NAC;

Al igual que en el caso anterior, en SQL estándar se puede escribir el texto COLUMN tras la palabra DROP.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS
4. CAMBIOS EN TIPOS DE DATOS EN LOS CAMPOS

Sólo se permiten algunos cambios:

- Aumentar la anchura de los tipos de datos (char/varchar)
- Reducir la anchura del campo: sólo es posible cuando la columna tiene nulos en todos los registros, o todos los valores existentes tienen un tamaño menor o igual a la nueva anchura (char/varchar)
- Cambiar de tipo CHAR a VARCHAR y viceversa.
- Cambiar de tipo DATE a TIMESTAMP y viceversa
- Cualquier otro cambio de tipo sólo será posible si la tabla está vacía.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS

4. MODIFICACIÓN DE TIPO DE UN CAMPO EN UNA TABLA:

En Mysql/Oracle:

ALTER TABLE NombreTABLA
MODIFY COLUMNA TIPO_DE DATO;

Ejemplo:

ALTER TABLE EMPLEADOS

MODIFY NOMBRE VARCHAR(50);

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

MODIFICAR TABLAS

5. RENOMBRAR CAMPOS DE UNA TABLA:

Permite cambiar el nombre de un campo o columna.

ALTER TABLE NombreTabla
RENAME COLUMN nombre TO NombreNuevo

Ejemplo:

ALTER TABLE EMPLEADOS

RENAME COLUMN Fecha TO FechaAlta;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

BORRAR TABLAS

DROP TABLE nombre_tabla [RESTRICT|CASCADE];

- Si utilizamos la opción RESTRICT, la tabla no se borrará si está referenciada por otra tabla.
- Si usamos la opción CASCADE, todo lo que referencie a la tabla se borrará junto con sus referencias.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CAMPOS AUTO_INCREMENT

- Los campos AUTO_INCREMENT son campos cuyo valor se incrementa automáticamente para cada registro que se añade a la tabla.
- Son campos numéricos, generalmente INT y suelen actuar como campos clave primaria.
- Sólo puede haber un campo AUTO_INCREMENT por tabla.

Ejemplo:**CREATE TABLE** EMPLEADOS

(Cod_Emp INT AUTO_INCREMENT PRIMARY KEY,

Nombre CHAR(20));

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CAMPOS CON VALORES ESTABLECIDOS POR DEFECTO (DEFAULT)

- A cada campo se le puede asignar un valor por defecto durante su creación mediante la propiedad default o en una modificación (comando ALTER TABLE)
- El valor indicado con DEFAULT se aplica cuando añadimos filas a una tabla y no damos un valor al campo. En lugar de ponerlo a NULL, se le asignará el valor por defecto indicado.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CAMPOS CON VALORES ESTABLECIDOS POR DEFECTO (DEFAULT)

Ejemplo de valor por defecto al crear la tabla

```
CREATE TABLE EMPLEADOS (
        COD_EMPLEADO INT,
        NOMBRE VARCHAR(25),
        CODPOSTAL CHAR(5) DEFAULT '10600');
```

Ejemplo para poner valor por defecto al modificar una tabla:

```
ALTER TABLE EMPLEADOS
MODIFY CODPOSTAL CHAR(5) DEFAULT '10600';
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CAMPOS CON VALORES NULOS (NULL)

- Cuando rellenamos los registros de las tablas, los campos pueden quedar vacíos, en ese caso decimos que el campo tiene el valor NULL (NULO).
- Permitir valores nulos es la opción por defecto para cualquier campo, excepto para la PK y aquellos campos donde hayamos puesto explicitamente la restricción NOT NULL.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CAMPOS CON VALORES NULOS (NULL)

Ejemplo de tabla sin restricción de valores nulos:

```
CREATE TABLE PRODUCTOS (
        COD_PRODUCTO INT,
        NOMBRE VARCHAR(25)
        );
```

TODOS LOS CAMPOS PERMITEN VALORES NULOS.

Ejemplo de tabla con restricción de valores NO NULOS

```
CREATE TABLE PRODUCTOS (
        COD_PRODUCTO INT NOT NULL,
        NOMBRE VARCHAR(25) NOT NULL
        );
```

TODOS LOS CAMPOS SON OBLIGATORIOS

Ejemplo para poner restricción NOT NULL modificando la tabla:

```
ALTER TABLE PRODUCTOS
MODIFY NOMBRE TIPO NOT NULL;
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIONES (CONSTRAINT)

IES Valle del Jerte PLASENCIA Bilingual Section

AENOR ER Empresa Registrada UNE-EN ISO 9001 ER-0271/2016

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

Informática y Comunicaciones

DEFINICIÓN

- Una restricción es una condición de obligado cumplimiento para una o más columnas de una tabla.
- Se denominan CONSTRAINT.
- Las restricciones se pueden realizar cuando estamos creando o modificando una tabla.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

TIPOS DE RESTRICCIONES

Los tipos de restricciones son los siguientes:

- **NOT NULL**
- **UNIQUE.**
- **PRIMARY KEY.**
- **FOREIGN KEY.**
- **CHECK (regla de validación)**

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

NOMENCLATURA DE RESTRICCIONES:

Tres letras para el nombre de la tabla +Guión bajo+
Dos letras con la abreviatura del tipo de restricción.
La abreviatura de restricción será:

NN →NOT NULL.
PK → PRIMARY KEY
UK→ UNIQUE
FK → FOREIGN KEY
CK → CHECK

Ejemplo: restricción de clave primaria en el campo cod_articulo en la tabla ARTICULOS:

CONSTRAINT ART_PK PRIMARY KEY (COD_ARTICULO);

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

NOMBRE DE LAS RESTRICCIONES

- Es una buena práctica que le asignemos un nombre a cada restricción que implementemos. De no hacerlo, será la propia base de datos la que asigne nombre a la restricción.
- Los nombres de restricción no se pueden repetir para el mismo esquema, debemos de buscar nombres únicos. Utilizaremos la misma nomenclatura para todas las restricción que creemos.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIONES AL DEFINIR LA TABLA

Cuando estamos creando la tabla hay dos maneras de poner restricciones:

1. Poner la restricción seguido a la definición de la columna. Este tipo es obligatorio cuando estamos poniendo la restricción de NOT NULL.

   Sintaxis:

   ...

   columna tipo [CONSTRAINT nombre] tipo,

   ...

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIONES AL DEFINIR LA TABLA

2.- Poner la restricción al final de la lista de columnas. La única restricción que no se puede definir de esta forma es la de tipo NOT NULL. El resto se harían siguiendo esta sintaxis:

columna1 definición1,

columna2 definición2,

...,

últimaColumna últimaDefinición,

[CONSTRAINT nombre] tipo(listaColumnas)

[,...otras restricciones...]

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN NOT NULL

- Lo hemos visto anteriormente.
- Esta restricción obliga a que un campo de la tabla tenga un valor para que el registro pueda ser almacenado (NO PUEDE QUEDAR VACÍO).
- Se crea durante la creación o modificación del campo añadiendo NOT NULL detrás del tipo.

Ejemplo:

```
CREATE TABLE ARTICULO
(
COD_ARTICULO INT  NOT NULL,
NOMBRE VARCHAR(25) NOT NULL,
PRECIO DECIMAL (10,2) DEFAULT '3.5' NOT NULL
);
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN UNIQUE (VALORES ÚNICOS)

- Esta restricción OBLIGA a que para esa columna no se puedan repetir valores en distintas filas.

Ejemplo:

```
CREATE TABLE EMPLEADOS
(
COD_EMP VARCHAR(9) NOT NULL,
DNI  VARCHAR2(9) UNIQUE NOT NULL
);
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN UNIQUE (VALORES ÚNICOS)

Esta restricción también puede ponerse al final de la tabla:

Ejemplo:

```
CREATE TABLE EMPLEADOS
(
COD_EMP VARCHAR(9) NOT NULL,
DNI VARCHAR2(9) NOT NULL,
CONSTRAINT EMP_UK UNIQUE(dni)
);
```

Si la restricción se refiere a varios campos, ponemos:

```
CREATE TABLE EMPLEADOS
(
COD_EMP VARCHAR(9) NOT NULL,
DNI VARCHAR2(9) NOT NULL,
TELEFONO VARCHAR(9) NOT NULL,
CONSTRAINT EMP_UK UNIQUE(DNI,TELEFONO)
);
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN PRIMARY KEY (CLAVE PRIMARIA)

- La clave primaria está formada por los campos que identifican a cada registro de forma única dentro de la tabla.
- El campo o campos que forman la PK son campos obligatorios (NOT NULL) y no pueden estar repetidos (UNIQUE), estos dos atributos se asumen, no es necesario ponerlos.
- Si la clave está formada por un solo campo se acostumbra a poner la restricción a continuación de la definición del campo.

```
CREATE TABLE EMPLEADOS
(
COD_EMPLEADO INT  PRIMARY KEY,
DNI_EMPLEADO  VARCHAR(9) UNIQUE
);
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN PRIMARY KEY (CLAVE PRIMARIA)

Si la clave está formada por varios campos, es obligatorio poner la restricción al final de la tabla, después de definir todas las columnas:

```
CREATE TABLE VENTAS
 (
COD_CLIENTE INT  NOT NULL,
COD_PRODUCTO INT NOT NULL,
CONSTRAINT VEN_PK PRIMARY KEY (COD_CLIENTE,COD_PRODUCTO)
);
```


UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN FOREIGN KEY (CLAVE SECUNDARIA O FORÁNEA)

- Una clave secundaria está formada por uno más campos de la tabla que están relacionados con la clave primaria (y a veces con otro campo) de otra tabla.

- Las restricciones de tipo FOREIGN KEY provocan una restricción de integridad referencial, en la que no se pueden indicar datos en las claves secundarias que no existan en las claves principales relacionadas.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN FOREIGN KEY (CLAVE SECUNDARIA O FORÁNEA)

Ejemplo:

```
CREATE TABLE VENTAS
(
COD_CLIENTE INT  NOT NULL,
COD_PRODUCTO INT NOT NULL,
CONSTRAINT VEN_PK (COD_CLIENTE,COD_PRODUCTO),
CONSTRAINT VEN1_FK FOREIGN KEY(COD_CLIENTE) REFERENCES CLIENTES(COD_CLIENTE),
CONSTRAINT VEN2_FK FOREIGN KEY(COD_PRODUCTO) REFERENCES PRODUCTOS(COD_PRODUCTO)
);
```

Las dos FK creadas significan que el campo **COD_CLIENTE** se relaciona con el campo **COD_CLIENTE** de la tabla CLIENTES y que el campo **COD_PRODUCTO** se relaciona con el campo **COD_PRODUCTO** de la tabla PRODUCTOS. Los campos no tienen porque llamarse igual en las dos tablas, pero sí deben ser del mismo tipo o tipos compatibles.

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN FOREIGN KEY E INTEGRIDAD REFERENCIAL

- La Integridad Referencial es obligatoria en BD Relacionales, pero puede ocasionar problemas al realizar operaciones de modificación y borrado.
- Para evitarlos podemos añadir cláusulas detrás de REFERENTES, que implementan las opciones del borrado modificación:
  - **ON DELETE SET NULL:** Coloca a nulos todas las claves secundarias relacionadas con la que borramos.
  - **ON DELETE CASCADE:** Borra todas las filas relacionadas con aquella que hemos eliminado.
  - **ON DELETE SET DEFAULT**. Coloca en el registro relacionado el valor por defecto en la columna relacionada.
  - **ON DELETE NOT ACTION.** No permite el borrado o modificación si hay valores que aparecen en registros de la tabla relacionada.
- Estas opciones son válidas también para la modificación de tablas (UPDATE).

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN FOREIGN KEY E INTEGRIDAD REFERENCIAL

Ejemplo: La misma tabla del ejemplo anterior, pero añadiendo las opciones de borrado y modificación:

```
CREATE TABLE VENTAS
(
COD_CLIENTE INT,
COD_PRODUCTO INT,
FECHA DATETIME,
CONSTRAINT VEN_PK (COD_CLIENTE,COD_PRODUCTO),
CONSTRAINT VEN1_FK FOREIGN KEY(COD_CLIENTE) REFERENCES CLIENTES(COD_CLIENTE)
ON DELETE SET  NULL,
CONSTRAINT VEN2_FK FOREIGN KEY(COD_PRODUCTO) REFERENCES PRODUCTOS(COD_PRODUCTO)
ON DELETE CASCADE
);
```


UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN CHECK (RESTRICCIONES DE VALIDACIÓN)

- Marcan una condición que deben cumplir los valores de un campo.
- Un mismo campo puede tener varias restricciones CHECK.

Ejemplo:

```
CREATE TABLE  PRODUCTOS
(
    COD_PRODUCTO  INT  NOT NULL PRIMARY KEY
    NOMBRE VARCHAR (50) NOT NULL,
    PRECIO DECIMAL(2,0) CHECK  (PRECIO >1),
    STOCK_MINIMO INT CHECK (STOCK_MINIMO >3)
);
```

En este caso, el precio no puede ser 0 y el mínimo valor para el campo stock_mínimo será 3.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

RESTRICCIÓN CHECK (RESTRICCIONES DE VALIDACIÓN)

- Si las restricciones hacen referencia a otros campos de la tabla, deben aparecer al final de dicha tabla.

Ejemplo:

```
CREATE TABLE  PRODUCTOS
(
COD_PRODUCTO  INT  NOT NULL RPIMARY KEY
NOMBRE VARCHAR (50) NOT NULL,
PRECIO DECIMAL(2,0) CHECK  (PRECIO >1),
STOCK_MINIMO INT CHECK (STOCK_MINIMO >3),
STOCK_MÁXIMO INT,
CONSTRAINT Stock_máx CHECK (STOCK_MINIMO<STOCK_MAXIMO)
);
```

En este caso, el stock máximo será siempre mayor que el mínimo, que a su vez está obligado a ser mayor de 3.

UNION EUROPEA Fondo Social Europeo "Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

AÑADIR RESTRICCIONES A UNA TABLA QUE YA ESTÁ CREADA

- Para añadir restricciones después de tener creada la tabla:

  ALTER TABLE Nombre _tabla

  ADD CONSTRAINT [NOMBRE] TIPO (Campos)

  TIPO puede ser CHECK, PRIMARY KEY O FOREIGN KEY.

Ejemplo:

ALTER TABLE PRODUCTOS

ADD CONSTRAINT precio_max CHECK (PRECIO <1000) ;

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

BORRAR RESTRICCIONES

ALTER TABLE tabla

DROP {FOREIGN KEY | PRIMARY KEY | UNIQUE(listaColumnas) | CONSTRAINT nombreRestricción} [CASCADE]

Ejemplo:

ALTER TABLE PRODUCTOS DROP CONSTRAINT precio_max;

- La opción CONSTRAINT elimina la restricción cuyo nombre se indica.
- La opción CASCADE hace que se eliminen en cascada las restricciones de integridad que dependen de la restricción eliminada y que, de otro modo, no permitiría eliminar dicha restricción.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

ÍNDICES EN MYSQL

Un índice es una estructura de datos que mejora la velocidad de las operaciones en una tabla. En MySQL hay varios tipos de índices:

- **PRIMARY KEY:** Se crea al definir la clave primaria de la tabla. El campo no admite duplicados ni valores NULL.
- **KEY o INDEX:** Son usados indistintamente por MySQL, permite crear índices sobre una columna, sobre varias columnas o sobre partes de una columna.
- **UNIQUE**: Este tipo de índice no permite almacenar valores repetidos en el campo.
- **FULLTEXT:** Permiten realizar búsquedas de palabras. Sólo pueden usarse sobre columnas CHAR, VARCHAR o TEXT

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

CREAR ÍNDICES

Toda tabla que tenga definida su clave primaria tiene definido un índice sobre el campo o campos de la clave primaria.

Podemos añadir nuevos índices a la tabla con la instrucción:

CREATE INDEX Nombre _ Índice ON Nombre_Tabla (Nombre_Campo)

Ejemplo:

CREATE INDEX Precio_I ON PRODUCTOS (Precio);

CREATE INDEX Ejemplo2_I ON PRODUCTOS (Precio,fecha);

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"**Una manera de hacer Europa**". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.