

UNIDAD 4

DDL

Contenidos teóricos



Unidad 4. Lenguaje de definición de datos

1. Introducción a SQL (Structured Query Language)

- SQL es el lenguaje fundamental de los SGBD relacionales
- Es un lenguaje declarativo en el que se establece qué se quiere hacer y no cómo hacerlo.
- SQL es un lenguaje diseñado para el acceso y manejo de datos en bases de datos relacionales

Unidad 4. Lenguaje de definición de datos

1. Introducción a SQL (Structured Query Language)

1970	Codd propone el modelo relacional y, asociado a éste, un sublenguaje de acceso a los datos basado en el cálculo de predicados.
1974	Teniendo en cuenta los aportes hechos por Codd, donald Chamberlin y otras personas trabajaron en los laboratorios de investigación de IBM, con el fin de desarrollar un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional, al cual llamaron SEQUEL (Lenguaje de consulta estructurado Inglés).
1974-1977	Se implementan prototipos como el SEQUEL-XRM llevando así a una revisión del lenguaje SEQUEL, cambio el nombre por motivos legales a SQL.
1977	IBM desarrolla el Sistema de gestión de bases de datos(SGBD) experimental System R.
1979	Fue oracle quien introdujo SQL por primera vez en un Programa comercial.

Unidad 4. Lenguaje de definición de datos

1. Introducción a SQL (Structured Query Language)

1986	El ANSI adoptó SQL como estándar para los lenguajes relacionales, este estándar se conoce como "SQL-86" o "SQL1"
1987	SQL se transformó en estándar ISO, a nivel normativo.
	Este primer estándar no cubre todas las necesidades de los desarrolladores por esto:
1992	Se lanza un nuevo estándar ampliado y revisado del SQL llamado "SQL-92" o "SQL2".
	Desde entonces se han hecho diversas revisiones al estándar "SQL2" incluyendo diversas mejoras. 1999- incorpora mejoras que incluyen triggers, procedimientos, funciones... este estándar se conoce como "SQL99" 2011- se ha realizado la última actualización "SQL2011"

Unidad 4. Lenguaje de definición de datos

1. Introducción a SQL (Structured Query Language)

El lenguaje SQL se divide en 4 grandes categorías

Lenguaje de definición de datos - DDL (Data Definition Language)

- Permite crear y modificar la estructura de las tablas de la base de datos.
- Sentencias: Create, alter, drop...

Lenguaje de manipulación de datos - DML (Data Manipulation Language)

- Permite crear y modificar registros en las tablas de la base de datos.
- Sentencias: Insert, Update, Delete...

Unidad 4. Lenguaje de definición de datos

1. Introducción a SQL (Structured Query Language)

El lenguaje SQL se divide en 4 grandes categorías

Lenguaje de consulta de datos - DQL (Data Query Language)

- Permite realizar consultas sobre los datos de la Base de Datos.
- Sentencias: Select...

Lenguaje de control de datos - DCL (Data Control Language)

- Permite gestionar los permisos y restricciones de los usuarios de la base de datos.
- Sentencias: Grant, Revoke...

Unidad 4. Lenguaje de definición de datos

Instalación de un Sistema Gestor de Bases de Datos

- Para poder trabajar con el diseño físico de nuestra base de datos necesitamos instalar el SGBD con el que vamos a trabajar. En clase utilizaremos: PostgreSQL. Podéis descargarlo para instalarlo desde:

<https://www.postgresql.org/>

Durante la instalación pedirá una contraseña para el superusuario de la base de datos postgres. Para evitar problemas indicaremos que la contraseña es root.


El resto de características las dejaremos por defecto.

Unidad 4. Lenguaje de definición de datos

SQL Shell

Al ejecutar SQL Shell se nos abre una línea de comandos donde nos pide los datos de conexión a la base de datos, que son:

- Servidor (por defecto en localhost).
- Nombre de la base de datos (por defecto postgres).
- Puerto (por defecto 5432).
- Nombre de usuario (por defecto postgres).
- Contraseña (la indicada durante la instalación, en nuestro caso root).

 Selecccionar SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (12.1)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=#
```


Unidad 4. Lenguaje de definición de datos

SQL Shell

Desde SQL Shell podemos utilizar los siguientes comandos:

\l : para listar nuestras bases de datos.

\c : para saber con qué base de datos estamos trabajando.

\c basename : para modificar la base de datos con la que trabajamos.

\d : lista las tablas de la base de datos con la que trabajamos.

\d table: muestra las características de la tabla indicada.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

- DDL es la abreviatura del lenguaje de definición de datos de SQL.
- El DDL es la parte del lenguaje SQL que realiza la función de definición de datos del SGBD. Fundamentalmente se encarga de la creación, modificación y eliminación de los objetos de la base de datos .

Creación de una Base de Datos

```
CREATE DATABASE NombreBaseDatos;
```

Ejemplo: CREATE DATABASE empresa;

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Creación de una Tabla

El nombre de las tablas deben cumplir las siguientes restricciones:

- Deben comenzar con una letra.
- No deben tener más de 30 caracteres.
- No puede haber dos tablas con el mismo nombre para el mismo esquema (pueden coincidir los nombres si están en distintos esquemas).
- No puede coincidir con el nombre de una palabra reservada SQL (por ejemplo no se puede llamar SELECT a una tabla).

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Creación de una Tabla

CREATE TABLE nombreDeTabla

(nombreDeLaColumna1 tipoDeDatos [DEFAULT valor] [restricciones] [, ...]);

Ejemplo:

- `CREATE TABLE Cliente (nombre VARCHAR(25));`
- `CREATE TABLE Proveedor (
nombre VARCHAR(25),
localidad VARCHAR(30) DEFAULT 'Palencia');`

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Tipos de Datos

ALFANUMÉRICOS: Van encerrados entre comilla simple: 'Alfanumérico'

- CHAR (b): Texto de anchura fija. Si se introduce un valor de 10 caracteres en un campo de CHAR(100), el valor ocupará igualmente 100 caracteres.
- VARCHAR (b): Texto de anchura variable. Si se define una columna de longitud 100 caracteres y se introduce en ella un valor de 10 caracteres, la columna ocupará 10 y no 100 como hacía con el tipo de dato CHAR.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Tipos de Datos

NUMÉRICOS: son números simples: 123

- **NUMBER (p,s):** Almacena valores numéricos con decimales. Su tamaño es variable. El parámetro p indica la precisión (número de dígitos contando los decimales) que contendrá el número como máximo. El parámetro s indica la escala, esto es, el máximo de dígitos decimales.
- **INTEGER:** Número entero que ocupa 4 bytes. Es capaz de almacenar un número en el rango -2147483648 to +2147483647.
- **REAL:** Número decimal que ocupa 4 bytes. El número puede tener hasta 6 decimales.
- **SERIAL:** Tipo entero que ocupa 4 bytes. Su valor es autonumérico y alcanza el rango 1 a 2147483647.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Tipos de Datos

FECHAS (DATE): Las fechas van encerradas entre comillas simples '18/09/2017'

El formato por defecto de las fechas es: 'DD-MON-YYYY' Esto es:

- Dos dígitos para el día.
- Tres primeras letras del mes (depende del idioma instalado).
- Cuatro dígitos para el año.

Por ejemplo: '11-NOV-2011' '12-APR-1999'

Internamente una fecha se almacena como el número de días desde cierto punto de inicio (por ejemplo el año 0). Esto permite que las fechas puedan ser tratadas en operaciones aritméticas normales:

- '11-NOV-2011' + 10 = '21-NOV-2011'
- '12-APR-1999' - 1 = '11-APR-1999'
- '24-APR-1992' - '1-APR-1992' = 23

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Tipos de Datos

BINARIOS

Permiten almacenar información en formato "crudo", valores binarios tal y como se almacenan en el disco duro o como residen en memoria. Estas columnas se pueden utilizar tanto para almacenar grandes cantidades de datos (hasta 4Gb.), como para almacenar directamente cualquier tipo de fichero (ejecutables, sonidos, vídeos, fotos, documentos Word, DLLs...) o para transportar datos de una base de datos a otra, ya que el formato binario es el único formato común entre cualquier sistema informático.

Unidad 4. Lenguaje de definición de datos

ACTIVIDAD 1

Nos disponemos a crear nuestra primera base de datos. Se pide abrir el gestor de bases de datos previamente instalado y proceder con el diseño físico de la base de datos correspondiente al siguiente modelo relacional:

Libro (código, título, género, sinopsis)

Autor (código, nombre, apellidos, lugarNac, fechNac)

Editorial (nombre, teléfono, web, e-mail)

Edición (ISBN, libro, nEdicion, fechEdicion, editorial)

Escribe (libro, autor)

De momento no utilizaremos ningún tipo de clave primaria ni ajena.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Borrado de una Tabla

Al borrar una tabla:

- Desaparecen todos los datos
- El borrado de una tabla es irreversible, no hay ninguna petición de confirmación, por lo que conviene ser muy cuidadoso con esta operación.
- El borrado podrá realizarse si el usuario tiene permisos suficientes y si no hay restricciones de integridad que lo impidan.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Borrado de una Tabla

DROP TABLE Cliente;

Si la tabla "Cliente" tiene restricciones de integridad referencial con otra tabla que impide el borrado, dará un error al ejecutar esta sentencia.

Unidad 4. Lenguaje de definición de datos

ACTIVIDAD 2

En esta actividad borraremos todas las tablas creadas anteriormente.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Creación de dominios

En SQL estándar existe la posibilidad de crear dominios.

```
CREATE DOMAIN name [AS] data_type  
[ DEFAULT expression ] [ restricciones [ ... ] ]
```

Un dominio permite asignar a un nombre de dominio un tipo de dato, una restricción, un conjunto de valores..

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Creación de dominios

Ejemplo: creamos el dominio Tdireccion

```
CREATE DOMAIN Tdireccion AS VARCHAR(3);
```

Gracias a esa instrucción podemos crear la siguiente tabla:

```
CREATE TABLE personal (  
    cod_pers SMALLINT,  
    nombre VARCHAR(30),  
    direccion Tdireccion );
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

Se pueden realizar las siguientes operaciones:

- Cambiar el nombre de la tabla
- Añadir columnas
- Borrar columnas
- Renombrar columnas
- Cambiar el tipo de dato de una columna
- Añadir una restricción
- Eliminar una restricción

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

- Cambiar el nombre de la tabla

ALTER TABLE nombreViejo RENAME TO nombreNuevo;

ALTER TABLE personal RENAME TO persona;

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

- Añadir nuevas columnas a la tabla

```
ALTER TABLE nombreTabla ADD COLUMN nombreColumna TipoDatos  
[Propiedades] , ADD COLUMN nombreColumna TipoDatos  
[Propiedades], ...;
```

```
ALTER TABLE Proveedor ADD COLUMN fecha_nacimiento DATE;
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

- Borrar columnas de la tabla

```
ALTER TABLE nombreTabla DROP COLUMN columna1, DROP COLUMN  
columna2, ...;
```

Elimina la columna de manera irreversible, incluyendo los datos que contenía. No se puede eliminar la última columna de una tabla, habrá que usar DROP TABLE.

```
ALTER TABLE Proveedor DROP COLUMN fecha_nacimiento;
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

- Renombrar columnas de la tabla

**ALTER TABLE nombreTabla
RENAME COLUMN nombreAntiguo TO nombreNuevo**

ALTER TABLE Proveedor RENAME COLUMN nombre TO nombre_prov;

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

- Restricciones: Una restricción es una condición de obligado cumplimiento para una o más columnas de la tabla. A cada restricción se le puede poner un nombre. Restricciones:
 - Valores no nulos NOT NULL (NN)
 - Clave primaria PRIMARY KEY (PK)
 - Clave foránea FOREIGN KEY (FK)
 - Clave candidata UNIQUE (UK)
 - Restricciones de validación CHECK (CK)

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

- A cada restricción se le puede dar un nombre, para mayor claridad es conveniente seguir unas normas a la hora de nombrar las restricciones. tabla_columna_restricción.
- Por ejemplo para poner una restricción de clave primaria a la columna código del la tabla autor, podríamos crear una restricción con el siguiente nombre: autor_codigo_pk

Autor (código, dni, nombre)

FK

Libro(ISBN, titulo, autor)

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

Tabla sin nombrar las restricciones

```
CREATE TABLE autor(  
  cod_autor VARCHAR (5) PRIMARY KEY,  
  dni VARCHAR (9) UNIQUE,  
  nombre varchar(30) NOT NULL);
```

Tabla nombrando las restricciones

```
CREATE TABLE autor(  
  cod_autor VARCHAR (5) CONSTRAINT aut_cod_pk PRIMARY KEY,  
  dni VARCHAR (9) CONSTRAINT aut_dni_uk UNIQUE,  
  nombre varchar (30) NOT NULL);
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

Tabla estableciendo las restricciones después de la definición de los campos

SI SE CREAN LAS RESTRICCIONES DESPUÉS DE DEFINIR LOS CAMPOS
ES NECESARIO ESPECIFICAR A QUÉ CAMPO PERTENECE CADA RESTRICCIÓN

```
CREATE TABLE autor (  
  cod_autor VARCHAR (5),  
  nombre NOT NULL,  
  CONSTRAINT aut_cod_pk PRIMARY KEY (cod_autor),  
  CONSTRAINT aut_dni_uk UNIQUE (DNI));
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

En las restricciones de clave foránea se establecen la tabla y el campo al que hacen referencia.

Tabla sin nombrar las restricciones

```
CREATE TABLE libro (  
    isbn VARCHAR(25) PRIMARY KEY,  
    cod_autor VARCHAR(9) REFERENCES autor(cod_autor));
```

Tabla nombrando las restricciones

```
CREATE TABLE libro (  
    isbn VARCHAR(25) CONSTRAINT lib_isbn_pk PRIMARY KEY,  
    cod_autor VARCHAR (9) CONSTRAINT lib_cod_fk REFERENCES autor(cod_autor));
```


Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

Tabla estableciendo las restricciones después de la definición de los campos

```
CREATE TABLE libro (  
    isbn varchar (25),  
    cod_autor VARCHAR (9),  
    CONSTRAINT lib_isbn_pk PRIMARY KEY (isbn),  
    CONSTRAINT lib_cod_fk FOREIGN KEY (cod_autor) REFERENCES autor2 (cod_autor));  
);
```

Si se crean las restricciones después de definir los campos es necesario especificar a qué campo pertenece cada restricción y en caso de las claves foráneas se incluye la palabra FOREIGN KEY.

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

La integridad referencial es una herramienta imprescindible de las bases de datos relacionales, por eso es necesario establecer cómo se comporta una clave foránea si desaparece o se modifica ese valor en la clave primaria.

- **ON DELETE SET NULL.** Coloca nulos todas las claves secundarias relacionadas con la borrada. (SET NULL)
- **ON DELETE CASCADE.** Borra todos los registros cuya clave foránea es igual que la clave del registro borrado. (CASCADE)
- **ON DELETE SET DEFAULT.** Coloca en el registro relacionado el valor por defecto en la columna relacionada. (DEFAULT)
- **ON DELETE RESTRICT.** Produce un error indicando que se producirá una violación de la clave foránea definida. (RESTRICT)

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

```
CREATE TABLE libro (  
    isbn VARCHAR (25),  
    cod_autor VARCHAR (9),  
    CONSTRAINT lib_isbn_pk PRIMARY KEY (isbn),  
    CONSTRAINT lib_cod_fk FOREIGN KEY (cod_autor) REFERENCES autor2 (cod_autor)  
        ON DELETE RESTRICT ON UPDATE CASCADE);
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones

Si una clave primaria o foránea es compuesta por más de una campo es obligatorio definir las restricciones después de definir los atributos.

```
CREATE TABLE matricula (  
    cod_alumno VARCHAR (5),  
    cod_asignatura VARCHAR (10),  
    CONSTRAINT mat_cod_alum_cod_asig_pk PRIMARY KEY (cod_alumno, cod_asignatura));
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Restricciones CHECK

Son restricciones que establecen una condición que deben cumplir los datos de una columna. Un campo puede tener varias restricciones check.

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto varchar (40) not null,  
  importe INT CONSTRAINT ing_imp_ck_min CHECK (importe>0)  
    CONSTRAINT ing_imp_ck_max CHECK (importe<8000));
```

La restricción CHECK prohíbe añadir datos cuyo importe no esté entre 0 y 8000

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

Para poder hacer referencia a otras columnas en la restricción check, hay que crear la restricción después de definir los atributos.

```
CREATE TABLE ingresos1 (  
    cod INT PRIMARY KEY,  
    concepto VARCHAR (40) NOT NULL,  
    importe_max INT,  
    importe INT,  
    CONSTRAINT ing_ck_maximo CHECK (importe<importe_max));
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

Una vez creada una tabla puede añadirse una restricción.

```
CREATE TABLE proyecto(  
codigo VARCHAR (5),  
nombre VARCHAR (50) CONSTRAINT pro_nom_uk UNIQUE);
```

Ejemplo. Añadir la restricción de clave primaria al campo código de la tabla proyecto.

```
ALTER TABLE proyecto  
ADD CONSTRAINT pro_cod_pk PRIMARY KEY (codigo)
```

Unidad 4. Lenguaje de definición de datos

2. DDL (Data Definition Language)

Modificar Tablas

Ejemplo. Borrar la restricción unique del campo nombre de la tabla proyecto.

```
ALTER TABLE proyecto  
DROP CONSTRAINT pro_nom_uk;
```