

# UD 04



SQL. Lenguaje de definición de datos

# SQL

# Contenido del tema

- **Introducción**
  - Características SQL
  - Reglas y sentencias
- Operadores
  - Aritméticos
  - Comparación
  - Algebra de bool
- Tipos de datos
- Sentencias SQL para el DDL
  - Creación y selección de bases de datos
  - Creación de tablas y columnas
  - Restricciones de integridad y reglas de negocio
- Otras sentencias

# Introducción

SQL (Structured Query Language) es un Lenguaje de Consulta Estructurado. Pero es algo más que un lenguaje de consulta, puesto que no solo ofrece funciones de recuperación, sino también se permite operaciones de actualización y manipulación de información y estructura de los datos.

Inventado por IBM a mediados de los 70

En 1976 Oracle Corporation introduce la primera implementación comercial

SQL se considera estándar de la industria como lenguaje de acceso a base de datos.

# Características

De entre todas podemos recopilar las siguientes características que mejor definen al lenguaje:

- **Unificado:** Con un mismo lenguaje podemos hacer cualquier tarea en base de datos, desde la creación y configuración de la base de datos hasta la consulta e introducción de datos.
- **Declarativo:** Sólo importa conocer qué queremos obtener, sin tener en cuenta el cómo. Podemos obtener o procesar varias filas a la vez y utiliza de forma transparente al usuario rutinas de acceso optimizadas.
- **Relacionalmente completo:** Puede realizar cualquier consulta de datos
- **Sencillo** y similar a la expresión escrita. Similar al inglés
- Tiene carácter **estándar**. Aunque cada fabricante refleja en él sus propias peculiaridades.

# Reglas y sentencias

Para familiarizarnos con el lenguajes establecemos a partir de ahora las siguientes reglas sintácticas en relación a términos vistos en contenidos anteriores.

- A las relaciones las llamaremos **Tablas**
- A los atributos llamaremos **Columnas**
- A las tuplas de datos llamaremos **Registros**
- Cada fila contiene un registro

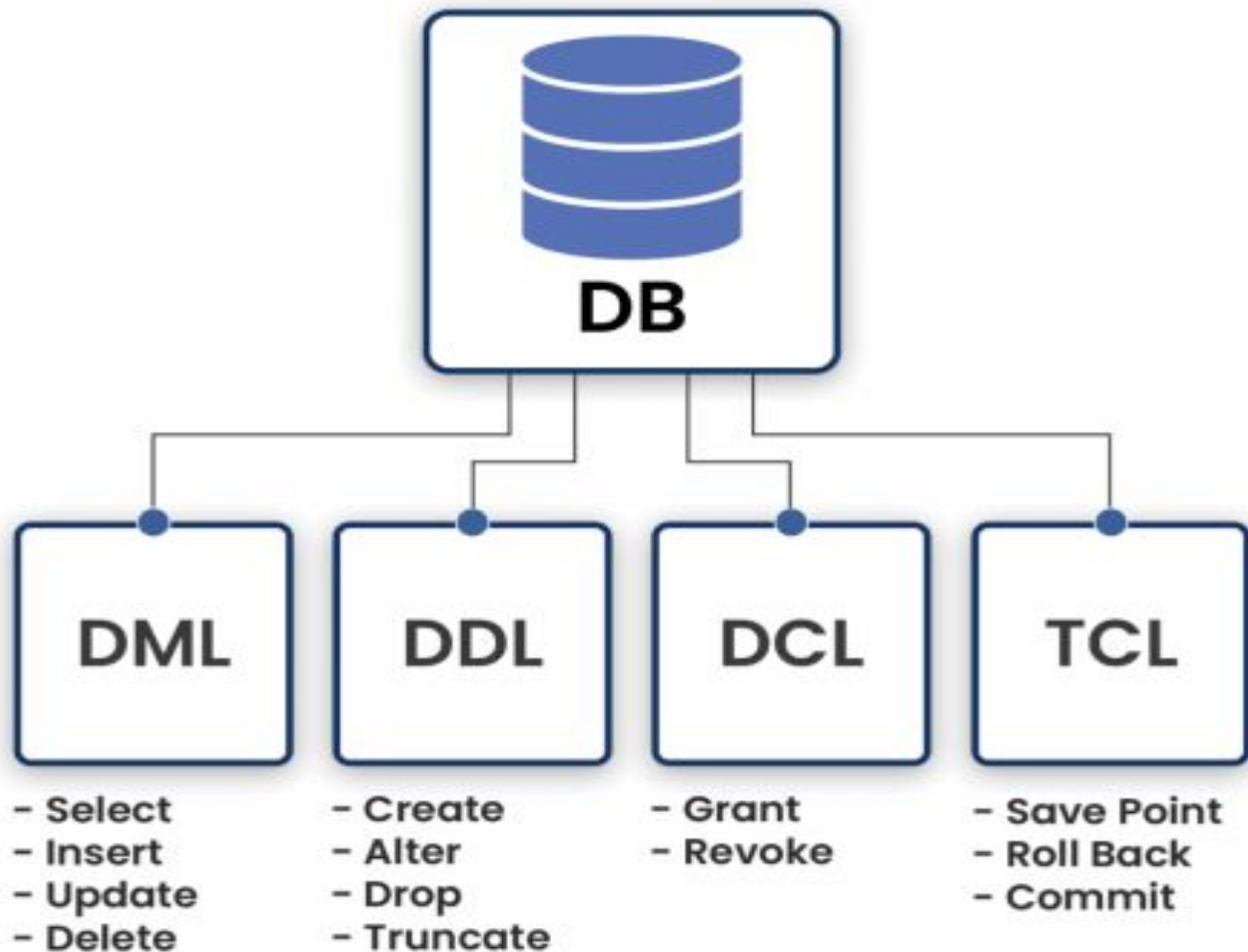
The diagram illustrates a database table structure. A central table is shown with five columns: DNI, Nombre, Apellidos, Sexo, and Edad. The table contains ten rows of data. Labels with arrows point to the table components: 'Columnas' points to the header row, 'Filas' points to the first data row, and 'Registro' points to the first data row. A label 'Atributos ó campos.' points to the first column.

DNI	Nombre	Apellidos	Sexo	Edad
309854	Pepe	Solano	V	32
205487	Lola	Sanchez	F	60
602548	Rosa	Perez	F	21
652314	Antonio	Cardenas	V	18
506981	Carlos	Lozano	V	51
559850	Fermin	Martin	V	54
203261	Laura	Mendez	F	69
363636	Carlos	Sanchez	V	22
696969	Felisa	Carmona	F	88

# Reglas y sentencias

Dependiendo de la necesidad podemos distinguir entre tres principales tipos de sentencia SQL:

- **DDL:** Lenguaje de definición de datos (Data definition language). Específica como vamos a almacenar la información y se compone de sentencias de creación y modificación de la estructura de datos
- **DML:** Lenguaje de manipulación de datos (Data manipulation language). Específica qué vamos a almacenar y consultar. Se compone de sentencias de inserción, consulta, modificación y borrado de datos.
- **DCL:** Lenguaje de configuración de datos (Data configuration language). Algunos autores integran DCL y TCL. Define la configuración del sistema, privilegios de usuarios y el control sobre las transacciones (TCL)





# Contenido del tema

- Introducción
  - Características SQL
  - Reglas y sentencias
- **Operadores**
  - Aritméticos
  - Comparación
  - Algebra de bool
- Tipos de datos
- Sentencias SQL para el DDL
  - Creación y selección de bases de datos
  - Creación de tablas y columnas
  - Restricciones de integridad y reglas de negocio
- Otras sentencias

# Operadores

En primer lugar debemos saber de qué se tratan estos operadores y para qué casos concretos se utilizan:

- Aunque existen tanto para la definición como para la manipulación de la información, serán mayormente utilizados a la hora de aplicar las condiciones de filtrado en las consultas, pero también se pueden utilizar por ejemplo a la hora de implementar una regla de negocio.
- Como todo lenguaje de programación, dentro de su sintaxis existen una serie de operadores entre los que encontramos, aritméticos, de comparación y del álgebra de bool.

# Operadores aritméticos

Los operadores aritméticos sirven para realizar operaciones sobre datos constantes o valores almacenados en base de datos. Se tratan de los operadores matemáticos más usuales. Estos se encuentran definidos en el propio lenguaje sin necesidad de invocar ni realizar ninguna función. Los más comunes son:

Operador aritmético	Operación
+	Suma
-	Resta
*	Multiplicación
/	División

# Operadores de comparación

Realizan operaciones de comparación entre dos elementos constantes o almacenados.

Hay que tener en cuenta que siempre devuelven un resultado booleano, es decir, true o false. Algunos sistemas gestores de bases de datos interpretan estos true o false como 0 y 1.

Un ejemplo de ello podría ser la restricción de sí una edad es mayor de 18 años.

“CHECK edad >= 18”

Operador	Función
=	Igual
> o >=	Mayor o mayor igual
< o <=	Menor o menor igual
!= <>	Distinto

# Álgebra de Boole

El álgebra de bool es la utilizada para formalizar expresiones que pueden concatenar operaciones lógicas y se utilizan en SQL para realizar comprobaciones o simplemente devolver un valor.

- Se rigen por las leyes de la lógica computacional.
- Los operadores lógicos usados serán: AND (&&), OR (||), NOT (!)
- Se pueden combinar entre ellos
- Se deben tener en cuenta las siguientes reglas algebraicas:

NOMBRE DE LA LEY	LOGICA DE PROPOSICIONES
1. Idempotencia	<ul style="list-style-type: none"><li>• <math>x \cdot x = x</math></li><li>• <math>x + x = x</math></li></ul>
2. Identidad	<ul style="list-style-type: none"><li>• <math>x \cdot 1 = x</math></li><li>• <math>x + 0 = x</math></li></ul>
3. Dominación	<ul style="list-style-type: none"><li>• <math>x \cdot 0 = 0</math></li><li>• <math>x + 1 = 1</math></li></ul>
4. Conmutativa	<ul style="list-style-type: none"><li>• <math>x \cdot y = y \cdot x</math></li><li>• <math>x + y = y + x</math></li></ul>
5. Asociativa	<ul style="list-style-type: none"><li>• <math>x \cdot y \cdot z \Leftrightarrow (x \cdot y) \cdot z \Leftrightarrow x \cdot (y \cdot z)</math></li><li>• <math>x + y + z = (x + y) + z = x + (y + z)</math></li></ul>
6. Distributiva	<ul style="list-style-type: none"><li>• <math>x \cdot (y + z) = (x \cdot y) + (x \cdot z)</math></li><li>• <math>x + (y \cdot z) = (x + y) \cdot (x + z)</math></li></ul>
7. Complementación:	
• Contradicción	• $x \cdot \bar{x} = 0$
• Tercero excluido	• $x + \bar{x} = 1$
8. Involución	<ul style="list-style-type: none"><li>• <math>\bar{\bar{x}} = x</math>      doble negación</li><li>• <math>\bar{\bar{\bar{x}}} = \bar{x}</math>      triple negación</li></ul>
9. D' Morgan	<ul style="list-style-type: none"><li>• <math>\overline{x \cdot y} = \bar{x} + \bar{y}</math></li><li>• <math>\overline{x + y} = \bar{x} \cdot \bar{y}</math></li></ul>
10. Absorción	<ul style="list-style-type: none"><li>• <math>x \cdot (x + y) = x</math></li><li>• <math>x + (x \cdot y) = x</math></li></ul>
12. Booleana	<ul style="list-style-type: none"><li>• <math>x \cdot (\bar{x} + y) = x \cdot y</math></li><li>• <math>x + (\bar{x} \cdot y) = x + y</math></li></ul>

# Contenido del tema

- Introducción
  - Características SQL
  - Reglas y sentencias
- Operadores
  - Aritméticos
  - Comparación
  - Algebra de bool
- **Tipos de datos**
- Sentencias SQL para el DDL
  - Creación y selección de bases de datos
  - Creación de tablas y columnas
  - Restricciones de integridad y reglas de negocio
- Otras sentencias

# Tipos de datos

Existe en SQL gran cantidad de datos que podemos asociar a las columnas de nuestras tablas ([https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)). Algunos de los más comunes que podemos encontrar son:

Tipo	Características	Ejemplo
CHAR(n)	Cadenas de caracteres de longitud fija (n)	Teléfono, DNI, Código Postal...
VARCHAR(n)	Cadenas de caracteres de longitud variable y longitud máxima (n)	Nombre, apellidos, dirección...
INT(n)	Números de tamaño (n)	Autonuméricos para PK, edad, cantidad...
DECIMAL(n,d)	Numero decimal de tamaño (n) con (d) decimales	Longitud, peso y otras unidades de medida
DATE	Formato para fechas	Instantes de tiempo y fechas
BLOB	Almacena objetos binarios	Ficheros, objetos pesados

# Contenido del tema

- Introducción
  - Características SQL
  - Reglas y sentencias
- Operadores
  - Aritméticos
  - Comparación
  - Algebra de bool
- Tipos de datos
- **Sentencias SQL para el DDL**
  - Creación y selección de bases de datos
  - Creación de tablas y columnas
  - Restricciones de integridad y reglas de negocio
- Otras sentencias



# Sentencias SQL para el DDL

La base de datos será el elemento principal, y dentro de esta crearemos el resto de elementos. Las operaciones que podremos hacer con las bases de datos fundamentalmente serán:

- Sobre bases de datos (Creación, Eliminación, Uso y Consulta)
- Sobre tablas (Mostrar su DDL, Creación, Modificación o Eliminación)
- Sobre las columnas (Cambiar su tipo, añadir o eliminar)
- Sobre las restricciones (Creación o eliminación)

# Creación y selección de bases de datos

Sintaxis de creación y eliminación:

```
CREATE DATABASE databasename;
```

```
DROP DATABASE testDB;
```

## Importante!

Si no usamos el comando `USE databasename` tendremos que acceder a los elementos de la base de datos indicando el prefijo con el nombre de la base de datos.

# Creación de tablas y columnas

La manipulación de las tablas y columnas permite su creación, modificación o borrado.

Además, desde la misma sintaxis se pueden indicar restricciones de nulidad y unicidad además de definir las claves primarias y foráneas de la propia tabla.

Para ello se sigue la siguiente sintaxis:

```
CREATE TABLE nombre_tabla
(
  Columna1 TIPO_DE_DATO
    [CONSTRAINT nombre_restricción]
    [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor]
    [REFERENCES Nombre_tabla [(columna [, columna])]
      [ON DELETE CASCADE]]
    [CHECK (condición)],
  Columna2 TIPO_DE_DATO
    [CONSTRAINT nombre_restricción]
    [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor]
    [REFERENCES Nombre_tabla [(columna [, columna])]
      [ON DELETE CASCADE]]
    [CHECK (condición)],
  ...
) [TABLESPACE espacio_de_tabla];
```

# Creación de tablas y columnas

Además de la creación, sobre las tablas podremos realizar las siguientes operaciones:

- Borrado:

```
DROP TABLE table_name;
```

- Modificación:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
ALTER TABLE table_name  
MODIFY COLUMN column_name datatype;
```

```
ALTER TABLE Persons  
DROP COLUMN DateOfBirth;
```

# Restricciones de integridad y negocio

Las restricciones o reglas de negocio son características especiales asociadas a los campos de una tabla:

- Se definen **directamente en la columna** o **al final** de la estructura de la sentencia CREATE TABLE o bien realizando una modificación de la tabla **tras ser creada**.
- Estas restricciones se almacenan con un **nombre asignado**, que bien, podemos definirlo nosotros o podemos dar la potestad al motor de bases de datos a que especifique de forma automática uno.
- El **nombre de la restricción debe ser único** en toda la base de datos.
- Las **foreign key** son un tipo de restricciones con mayor complejidad que estudiaremos en la siguiente diapositiva.

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255),  
    CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Sandnes')  
);
```

# Restricciones de integridad y negocio

Los valores almacenados en los archivos deben cumplir con restricciones de consistencia. Por ejemplo, no se puede insertar una nota de un alumno en una asignatura si previamente esa asignatura no está creada. A esto llamamos **integridad referencial**. La principal función de las FOREIGN KEY es precisamente asegurar la integridad referencial a la hora de establecer y representar las relaciones existentes entre tablas. Hay varios tipos de restricciones:

- **CASCADE**: Al modificar o eliminar un dato referenciado desde otra tabla en la base de datos, éste se actualiza en todas sus referencias.
- **SET NULL**: Al modificar o eliminar un dato referenciado desde otra tabla en la base de datos, éste se establece como NULL
- **DEFAULT**: Al modificar o eliminar un dato referenciado desde otra tabla en la base de datos, éste toma el valor por defecto de su columna
- **RESTRICT**: Al intentar modificar o eliminar un dato referenciado desde otra tabla en la base de datos, el servidor SQL nos da un error avisando que no puede realizarse dicha operación por existir referencias.

```
51 CREATE TABLE alumnos(
52     DNI CHAR(9),
53     Nombre VARCHAR(50),
54     Apellido1 VARCHAR(50),
55     Apellido2 VARCHAR(50),
56     Direccion VARCHAR(100),
57     Sexo CHAR(1),
58     Fecha_Nacimiento DATE,
59     Curso INT(4) NOT NULL,
60     CONSTRAINT pk_alumnos PRIMARY KEY (dni),
61     CONSTRAINT fk_cursos FOREIGN KEY (Curso)
62         REFERENCES cursos(Cod_curso) ON UPDATE CASCADE ON DELETE NULL,
63     CHECK (Sexo = 'H' || Sexo = 'M')
64 );
```

# Contenido del tema

- Introducción
  - Características SQL
  - Reglas y sentencias
- Operadores
  - Aritméticos
  - Comparación
  - Algebra de bool
- Tipos de datos
- Sentencias SQL para el DDL
  - Creación y selección de bases de datos
  - Creación de tablas y columnas
  - Restricciones de integridad y reglas de negocio
- **Otras sentencias**



# Otras sentencias

Un **script** es un fichero que contiene una serie de sentencias o procedimientos a ser ejecutados por nuestro servidor de bases de datos.

- Para su creación solo necesitaremos un editor de texto (Atom, Notepad++, Visual Studio... )
- Es recomendable usar algunos de los indicados por la ayuda que nos ofrece al autocompletar y resaltar palabras detectadas.
- Su extensión es .sql
- Será referenciado a través del comando `sql > source ruta/script.sql`

Por último utilizaremos una serie de sentencias que nos ayudarán a comprobar la efectividad de nuestra estructura de datos y sus restricciones. Estas sentencias serán:

- La sentencia de consulta SELECT
- La sentencia de eliminación DELETE
- La sentencia de modificación UPDATE
- La sentencia de inserción INSERT