LDD→ Definir datos

LMD→ Manipulación de datos

LCD→ Control de datos

**FORMAS DE USAR SQL**

* **Interactiva**
* **Programada (PL/SQL)**
* **Desde un programa de 3ª generación**
* **Fichero SQL**

**PASOS A SEGUIR**

* **Iniciar MySQL (XAMPP)**
* **Iniciamos el entorno de consola de SQL (Dentro de la carpeta de XAMPP)**

**COMANDOS**

* **CREATE DATABASE *nombre*;→** Se crea una nueva base de datos
* **USE *nombre\_BD*;→** Nos posicionamos en una BD
* **SHOW TABLES;→** Se visualizan las tablas
* **DESC *nombre\_Tabla*;→** Se ven los registros
* **SELECT \* FROM *TABLA*;→** Selecciona los registros de una columna
* **SOURCE *FICHERO*;→** Ejecuta un fichero sql
* **DROP DATABASE *nombre \_BD*;** → Borra una BD

**TIPOS DE DATOS**

Los principales son:

* **Numéricos**
  + **INT**
  + **FLOAT**
* **Fecha**
  + **DATE**
* **Cadena**
  + **CHAR** (Tamaño fijo, 4 bytes)
  + **VARCHAR** (Tamaño variable, de 1 a 5 bytes)

**LENGUAJE DE DEFINICIÓN DE DATOS (LDD)**

**DEFINICIÓN DE LA TABLA**

* Nombre de la tabla
* Nombre de las columnas
* Tipos de datos almacenados
* Tamaño de cada columna

**RESTRICCIONES A NIVEL DE COLUMNA**

* NOT\_NULL | NULL
* DEFAULT
* PRIMARY KEY
* FOREIGN KEY
* UNIQUE
* CHECK: Control (La edad de los alumnos sea mayor que x)

**RESTRICCIONES A NIVEL DE TABLA**

Estructura:

*TABLE*

*COLUMNAS*

*…*

*RESTRICCIÓN*

Se escribe: CONSTRAINT *nombreRestricción* TipoDeRestricción *nombreColumna [, nombreColumna2]…*};

Este método viene bien para, en caso de un error a la hora de manipular la tabla, podremos saber si es una restricción nuestra.

* PRIMARY KEY (PK\_NombreTabla)
* UNIQUE (UQ\_NombreTabla\_NombreAtributo)
* CHECK (CK\_NombreTabla\_NombreAtributo) (Expresion Ej: (Edad > 18 AND Edad < 99))
* FOREIGN KEY (FK\_Tabla1\_Tabla2)

HAY CASOS EN LOS QUE NO PUEDEN IR ALGUNAS RESTRICCIONES. SON DOS CASOS:

* Cuando la clave primaria es compuesta, no puede ir a nivel de columna, tiene que ir a nivel de tabla. CONSTRAINT PK\_COL PRIMARY KEY (ATR, ATR2)
* En el caso de que no queramos que se repita la combinación de varias columnas, en vez de poner UNIQUE en las columnas, lo ponemos a nivel de tabla. (No es lo mismo poner UNIQUE a 2 columnas que a nivel de tabla. En el primer caso, no se puede repetir a y/o b. En el segundo caso, no se puede repetir a y b)

Una restricción puede ir a nivel de columna o de tabla **solo si afecta a una sola columna**

**CREAR TABLA**

CREATE TABLE [IF NOT EXISTS] *Nombre*

(

Col1 TIPO;

Col2 TIPO;

);

Con IF NOT EXISTS lo que hacemos es evitar que nos salga un error a la hora de crear una tabla ya existente.

**INTEGRIDAD REFERENCIAL**

Relación entre tablas. Los datos relacionados deben encontrarse existentes.

* **Actualización en cascada:** Actualiza los valores relacionados, si no, quedarían valores colgando.
* **Borrado en cascada**: Al borrar un registro, borra aquellos que estuviesen relacionados a él.
* **CASCADE**
* **SET NULL**
* **SET DEFAULT**
* **NO ACTION**
* **RESTRICT:** No poder borrar el valor si está relacionado con otros.

RESTRICT y NO ACTION tienen el mismo efecto, pero NO ACTION realiza la comprobación después de tratar de modificar la tabla, mientras que RESTRICT lo hace en la misma sentencia de actualización o borrado.

SET NULL hace referencia a la cardinalidad mínima (0, 1).

CASCADE hace referencia a una entidad débil, los alumnos dependen de la existencia de profesores.

**CREAR UNA TABLA A PARTIR DE OTRA**

Hacer una copia de otra tabla.

CREATE TABLE *NOMBRE*(*COLUMNAS*) AS

SELECT \* FROM *TABLA\_DE\_REFERENCIA;*

**MODIFICACIÓN DE TABLAS**

ALTER TABLE *NombreTabla*

Se puede hacer lo siguiente:

* **Añadir columna:** Si hay datos se pondrá en NULL los valores obligatoriamente
  + **ADD COLUMN \_\_\_**
* **Añadir constraint**
  + **ADD CONSTRAINT \_\_\_**
* **Borrar columna, constraint**
  + **DROP COLUMN \_\_\_**
  + **DROP *TIPOCONSTRAINT* [NombreConstraint]**
  + Borrar clave foránea e índice
  + Para borrar una clave foránea es recomendable ponerle nombre a la restricción. Al crear una clave foránea se crea un índice, por lo que también hay que borrarlo. Un índice permite la búsqueda de valores repetidos a la hora de crear nuevas inserciones. A la hora de borrar un dato, se comprueba que con su índice que no está asociado a ninguna otra tabla.
* **Modificar columna sin cambiar su nombre**
  + **MODIFY**
* **Modificar una columna y su nombre**
  + **CHANGE**
* **Renombrar tabla**
  + **RENAME TABLE *NombreAntiguo* TO *NombreNuevo***

En una clave foránea y primaria a la vez, solo existe un índice, por ello no se puede borrar la clave primaria primero.