**ÓRDENES DDL EN SQL**

El lenguaje DDL (Data Definition Language) se usa para definir los datos de la base de datos, como las tablas y su estructura.

Las órdenes más comunes son:

* **CREATE DATABASE** *Nombre;* → Crear una bbdd.
* **USE DATABASE** *Nombre;* → Establecer la bbdd que vamos a usar.
* **SHOW DATABASES/TABLES**; → Mostrar las bbdd o tablas que hay creadas.
* **CREATE TABLE** *Nombre(NombreCampo1 TipoCampo1 [Restricciones1],*

*NombreCampo2 TipoCampo2 [Restricciones2],*

*NombreCampoN TipoCampoN [RestriccionesN]);*

* **CREATE TABLE** OtraDatabase.NombreTabla; → Crear una tabla en una bbdd distinta a la activa.

**TIPOS DE DATOS**

Al igual que en programación, las bases de datos tienen restricciones de tipos de datos. Los principales y más importantes de SQL son:

* **CHAR**: Cadena de caracteres alfanuméricos
* **VARCHAR**: Cadena de caracteres de longitud variable
* **LONG**: Hasta 2GB en forma de cadena de caracteres
* **NUMBER**: Tanto enteros como punto flotante
* **FLOAT**: Decimales en punto flotante
* **DATE**: Año (incluyendo el siglo), mes, día, horas, minutos y segundos
* **ENUM**: Enumeración (‘Grande’, ‘Pequeño’, ‘Mediano’)

**RESTRICCIONES**

Las restricciones en bases de datos sirven para que un usuario final no introduzca datos erróneos, que la estructura de la propia base de datos sea íntegra y fiable, y que no se cometan errores de parte de quienes consultan o modifican la base de datos.

Las restricciones más importantes son:

* **[CONSTRAINT nombre restricción]**:
* **[NOT NULL]**: El campo no puede estar vacío.
* **[UNIQUE]**: No se puede repetir.
* **[FOREIGN KEY]**: Clave foránea o ajena.
* **[PRIMARY KEY]**: Clave primaria.
* **[DEFAULT valor]**: Establece un valor por defecto para un campo.
* **[REFERENCES nombretabla[(columna[,columna])]]**: Establecen las claves foráneas, referencian a la tabla de la que proceden y a sus columnas concretas.
* **[ON DELETE CASCADE]**: Si el campo es borrado, se borran sus hijos.
* **[CHECK condición]**: Comprobar si el campo cumple unos requisitos. Si la condición no se cumple, salta un error avisando de que es imposible introducir los datos en la base de datos.

**MODIFICACIÓN DE DATOS**

Una vez creados e introducidos los datos en la base de datos, se pueden modificar, eliminar y demás con las siguientes instrucciones:

* **RENAME Nombreviejo TO Nombrenuevo**: Se usa para renombrar tablas.
* **ALTER TABLE Nombretabla ADD**(*NombreCampo1 TipoCampo1 [Restricciones1],*

*NombreCampoN TipoCampoN [RestriccionesN])*

: Se usa para añadir columnas.

* **ALTER TABLE Nombretabla DROP COLUMN**(*Campo1 Tipo1 [Restricciones1],*

*CampoN CampoN [RestriccionesN])*

: Se usa para eliminar columnas.

* **ALTER TABLE Nomtabla DROP CONSTRAINT Nomrestricc**: Borrar restricción.
* **ALTER TABLE Nomtabla RENAME CONSTRAINT Nomviejo TO Nomnuevo**: Renombrar restricción.
* **ALTER TABLE Nomtabla DISABLE CONSTRAINT Nombrerestricc[CASCADE]**: Desactivar restricción.
* **ALTER TABLE Nomtabla ENABLE CONSTRAINT Nombrerestricc[CASCADE]**: Activar restricción.
* **SELECT \* FROM all\_Constraints**: Ver todas las restricciones.

**CONSTRAINTS - RESTRICCIONES**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Columna** | **Tabla** | **Nombrar** |
| ***PK*** | Si es PK *simple* | Si es *compuesta* | Siempre |
| ***NOT NULL*** | Sí, es más sencillo | Es posible | - |
| ***UNIQUE*** | Sí, es más sencillo | Es posible | - |
| ***CHECK*** | Es posible | Es más conveniente | Siempre |
| ***FK*** | Es posible | Muy recomendable | Siempre |

**EJERCICIO CLASE**

CREATE TABLE Usuario(Credito NUMBER CHECK(Credito between 0 and 2000);

ALTER TABLE Usuario ADD(Usuario varchar(20) PRIMARY KEY);

ALTER TABLE Usuario RENAME Usuario TO Login;

CREATE TABLE Empleado(Cod\_Cliente varchar(5) PRIMARY KEY,

Nom char(20),

Ape char(40),

Sueldo int);

ALTER TABLE Empleado Modify(Sueldo int CHECK(Sueldo between 1000 and 2000));