BASES DE DATOS UD-2

BASES DE DATOS RELACIONALES (III)

CREAR UNA BASE DE DATOS RELACIONAL

ÍNDICE

[DATABASE 3](#_Toc117447013)

[A. BASE DE DATOS, ESQUEMAS Y OBJETOS 3](#_Toc117447014)

[B. INSTRUCCIONES DATABASE 3](#_Toc117447015)

[TABLE 4](#_Toc117447016)

[A. SQL. CREACIÓN DE TABLAS 4](#_Toc117447017)

[B. TIPO DE DATOS 4](#_Toc117447018)

[1. ¿CÓMO SE ALMACENAN LOS STRINGS? DIFERENCIA ENTRE CHAR Y VARCHAR/VARCHAR2. 5](#_Toc117447019)

[2. NÚMEROS: PRECISIÓN Y ESCALA 6](#_Toc117447020)

[C. VALOR NULL O NULO 6](#_Toc117447021)

[D. RESTRICCIONES (CONSTRAINTS) 7](#_Toc117447022)

[1. ¿QUÉ ES UNA RESTRICCIÓN O CONSTRAINT? 7](#_Toc117447023)

[2. RESTRICCIÓN DE COLUMNA Y DE TABLA 7](#_Toc117447024)

[3. PRIMARY KEY 8](#_Toc117447025)

[4. FOREIGN KEY 8](#_Toc117447026)

[5. NOT NULL 9](#_Toc117447027)

[6. UNIQUE 9](#_Toc117447028)

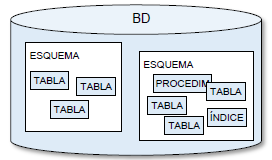
[7. CHECK 10](#_Toc117447029)

[8. NOMBRE DE RESTRICCIÓN 11](#_Toc117447030)

[9. VISUALIZAR RESTRICCIONES YA CREADAS 12](#_Toc117447031)

# DATABASE

## BASE DE DATOS, ESQUEMAS Y OBJETOS

Una BD dentro del contexto de un SGBD es un conjunto de objetos que se utilizarán para gestionar los datos. Esos objetos son tablas, vistas, procedimientos, enlaces… y están contenidos en esquemas. Por lo general un esquema está asociado a un usuario (salvo que ese usuario de permisos a otro para poder acceder al esquema).

## INSTRUCCIONES DATABASE

Existen varios tipos de instrucciones referidas a la BD como tal, por ejemplo:

Interfaz de usuario gráfica, Texto, Aplicación

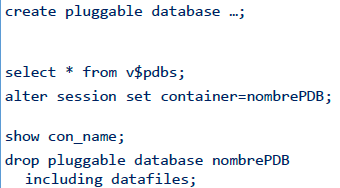
Descripción generada automáticamente

Aunque esos comandos son el estándar SQL puede haber diferencias en los SGBD:

* + **MySQL**. Para visualizar la BD actual se usa el comando:



* + **Oracle DataBase**. En Oracle las BBDD se denomina PBD y se usan varios comandos diferentes. Por ejemplo:



# TABLE

Es un objeto dentro de una BD que representa una tabla o relación del modelo relacional. Está compuestas por campos (columnas) cada uno de los cuales con un nombre y tipo de datos.

## SQL. CREACIÓN DE TABLAS

Formato de creación de tabla simple:

Texto

Descripción generada automáticamente

El nombre debe tener una longitud mínima de 1 carácter y máxima de 30 caracteres. No debe ser una palabra reservada. El primer carácter debe ser alfanumérico y el resto pueden ser letras, números y el carácter subrayado (\_).

Para ver la estructura de la tabla se usa el comando:



## TIPO DE DATOS

Establece de qué clase es el dato que guarda un campo en una tabla. Los principales son: texto, numérico y fecha.

En cada SGDB se establecen los distintos tipos:

|  |  |  |  |
| --- | --- | --- | --- |
|  | TIPO DE DATO | | DESCRIPCIÓN |
|  | **MySQL** | **OracleDB** |
| Numérico | TYNYINT  SMALLINT  MEDIUMINT  INT, INTEGER  BIGINT | NUMBER | Número entero con o sin signo. |
| FLOAT  DOUBLE | NUMBER  BINARY\_FLOAT  BINARY\_DOUBLE | Número decimal. |
| BIT |  | Número binario |
| Texto | CHAR | CHAR  NCHAR | Cadena con longitud fija. |
| VARCHAR | VARCHAR2  NVARCHAR2 | Cadena. |
| Fecha | DATE | DATE | Fecha. |
| DATETIME  TIMESTAMP | TIMESTAMP | Fecha y hora. |
| TIME |  | Hora. |
| YEAR |  | Año. |
| Datos binarios | TINYBLOB  MEDIUMBLOB  BLOB  LONGBLOB | BLOB  CLOB  NCLOB  BFILE  RAW | Datos de archivo (LOB: objeto grande). |

1. ¿CÓMO SE ALMACENAN LOS STRINGS? DIFERENCIA ENTRE CHAR Y VARCHAR/VARCHAR2.

Dentro de ellos van caracteres alfanuméricos y signos de puntuación. Los caracteres ocupan 1 byte de memoria.

* + En el CHAR(n) asignamos un espacio de memoria fijo de n bytes.
  + En el VARCHAR(n)/VARCHAR2(n) se ocupan los bytes del dato guardado más uno de fin de string.

|  |  |  |
| --- | --- | --- |
|  | TAMAÑO OCUPADO | |
| VALOR | **CHAR(4)** | **VARCHAR2(4)** |
| ‘’ (vacío) | 4 bytes | 1 byte |
| ‘ab’ | 4 bytes | 3 bytes |
| ‘abcd’ | 4 bytes | 5 bytes |
| ‘abcde’ | No lo guarda | No lo guarda |

1. NÚMEROS: PRECISIÓN Y ESCALA

El formato es **NUMBER(precisión, escala)**, siendo:

* **Precisión**: total de dígitos (enteros y decimales).
* **Escala**: número de decimales. Si la escala es un número negativo se realiza un redondeo a la unidad indicada (-1 decenas, -2 centenas…).

|  |  |  |  |
| --- | --- | --- | --- |
| Dato actual | Formato | Almacenamiento | Explicación |
| 7456123.89 | NUMBER | 7456123.89 | Se almacena todo el número. No hay un límite (lo marca la memoria). Desaconsejado. |
| 7456123.89 | NUMBER(9) | 7456123 | Sólo se almacena la parte entera redondeándolo. |
| 7456123.89 | NUMBER(9,2) | 7456123.89 | Se almacena todo el número. |
| 7456123.89 | NUMBER(9,1) | 7456123.9 | Se almacena todo el número entero y se redondea la parte decimal a 1 dígito. |
| 7456123.8 | NUMBER(6) | ERROR | El valor entero es mayor que la precisión. |
| 7456123.8 | NUMBER(15,1) | 7456123.8 | Se almacena todo el número. No se llega al límite de 15. |
| 7456123.89 | NUMBER(7,-2) | 7456100 | Se almacena el número entero redondeándolo a las centenas. |
| 7456123.89 | NUMBER(-7,2) | ERROR | El valor de precisión debe ser positivo. |

## VALOR NULL O NULO

Es un valor que se produce cuando en una columna no se introduce un dato. Esto se hace cuando no se conoce el dato o cuando el dato que hay que introducir no tiene sentido.

No es lo mismo que:

* **Un espacio en blanco**. Un espacio en blanco es un carácter que se trata como una letra, no es una ausencia de valor.
* **Un espacio en cero**. Un cero es un dígito que indica 0 unidades, no es una ausencia de valor.

Cuando se opera con valor null, hay que seguir reglas específicas.

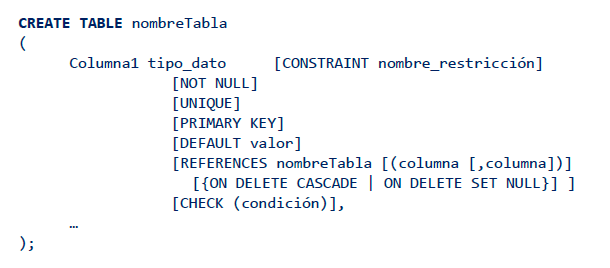
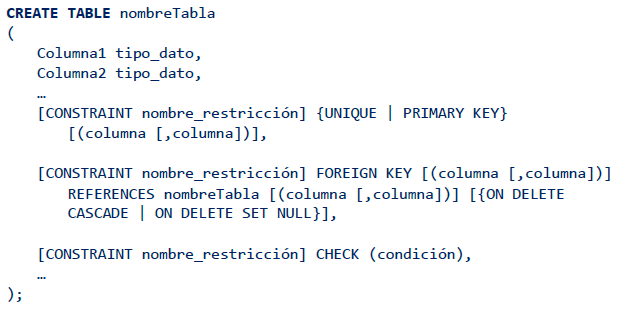
Se emplean dos tipos de operadores específicos IS NULL y IS NOT NULL para comprobar si un valor tiene el valor null. No se utilizan los operadores de comparación (= y !=).

## RESTRICCIONES (CONSTRAINTS)

1. ¿QUÉ ES UNA RESTRICCIÓN O CONSTRAINT?

Se puede definir como la condición que los datos de uno o varios campos de una tabla deben cumplir obligatoriamente. Se crea en el momento de crear una tabla (CREATE TABLE) aunque puede añadirse posteriormente (ALTER TABLE).

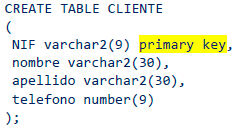
1. RESTRICCIÓN DE COLUMNA Y DE TABLA

* **Restricción de columna.** La restricción se define junto a la definición del campo. Sólo puede afectar a un campo.
* **Restricción de tabla.** La restricción se define de manera independiente a los campos. Puede afectar a más de un campo.

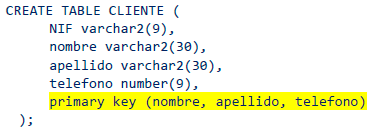
1. PRIMARY KEY

Define la clave primera de una tabla o relación. Por ejemplo:

* **Clave simple.**



* **Clave compuesta.**

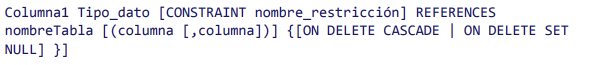


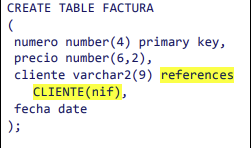
1. FOREIGN KEY

Define la clave externa de una tabla. Hace referencia a una PK de otra tabla o de la misma tabla.

Se puede realizar restricción de columna o de tabla.

Si es de columna se usará la estructura:



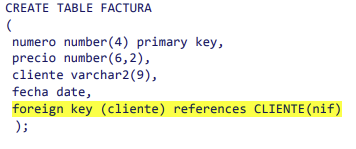
Es decir:

En este caso no se utiliza la clausula FOREIGN KEY y se hace referencia directamente en la línea de código correspondiente a la columna que queremos convertir en FK.

Si es tabla se usará la estructura:



Es decir:



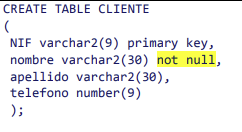
En este caso sí que haremos uso de la cláusula FOREIGN KEY después de las líneas de código correspondientes a las columnas y sus tipos.

Los tipos de borrado (ON DELETE) son la integridad referencia de la BD. Por defecto el tipo de borrado será restringido.

Hay que tener en cuenta que el tipo de datos y su dominio de un campo FK debe ser igual que el tipo de datos y dominio de la PK a la que hace referencia.

1. NOT NULL

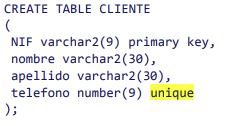
Convierte a un campo en obligatorio, es decir, no puede estar vacío, no puede tener valor NULL.

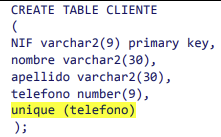
No tiene restricción de tabla, sólo puede ser restricción de columna y sigue la siguiente estructura:

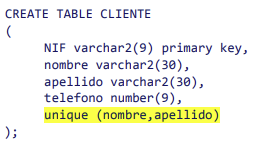
1. UNIQUE

Convierte al valor de un campo (o conjunto de campos) en único, es decir no pude repetirse.

Puede utilizarse como restricción de columna o como restricción de tabla.

En el caso de usarlo como restricción de columna:

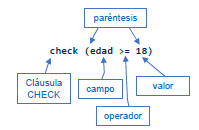
En el caso de usarlo como restricción de tabla:

También se puede usar de forma compuesta, es decir, afectando a varios campos:

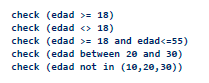
1. CHECK

Sirve para imponer una condición a un campo.

Su sintaxis es la siguiente:

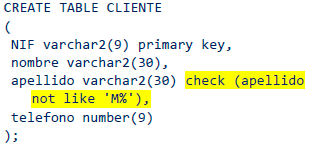


Ejemplos de operadores:

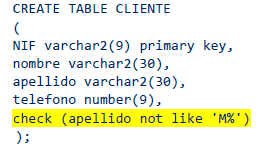


Sólo puede realizarse la validación de un campo con CHECK (no existe el check compuesto). Pero sí que existe como restricción de columna y de tabla.

* Restricción de columna:



* Restricción de tabla:



1. NOMBRE DE RESTRICCIÓN

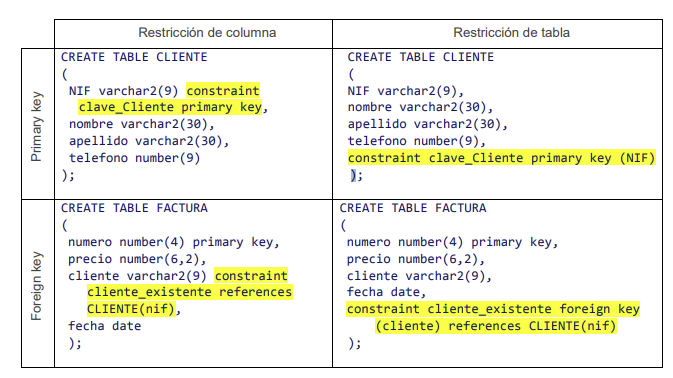
Las restricciones tendrán un nombre asociado que las defina. Es opcional pero muy recomendable para facilitar la detección y análisis de códigos de error que lance el SGBD.

Tienen la siguiente sintaxis:





El nombre de una constraint no puede repetirse en toda la base datos.

Pueden ser constraints de columna o de tabla, por ejemplo,

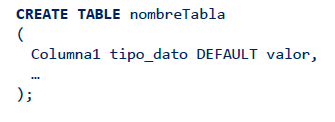
1. VISUALIZAR RESTRICCIONES YA CREADAS

Existen dos formas de ver las restricciones definidas:

* **USER\_CONSTRAINTS**: Restricciones en tablas propiedad del usuario. Si nos fijamos en el campo “constraint\_type” aparecerá una letra:
  + **C** 🡪 Restricciones CHECK y NOT NULL.
  + **P** 🡪 Restricciones PRIMARY KEY.
  + **R** 🡪 Restricciones FOREIGN KEY (references).
  + **U** 🡪 Restricciones UNIQUE.
* **USER\_CONS\_COLUMS**: Información sobre las restricciones, en especial se refleja el campo o campos a los que afectan.

## VALOR POR DEFECTO (DEFAULT)

Sirve para dar valores por defecto. Si no insertamos un valor en un registro, se incluirá ese valor.

La sintaxis es:

## SQL. BORRADO DE TABLAS

Podemos usar el TRUNCATE TABLE (del lenguaje LDD) borra el contenido de la tabla liberando la memoria donde están guardados los registros. No se puede volver atrás.

Otra forma es con DELETE FROM nombreTabla, este es por LMD. Se puede volver atrás.

DROP STORAGE libera todo el espacio (opción por defecto) mientras que REUSE STORAGE mantiene el espacio reservado para nuevas filas.

DROP TABLE borra el contenido y la tabla. Puede tener opcionalmente la cláusula CASADE CONSTRAINTS para permitir que se borren tablas que estén asociadas, para ello la base de datos de forma automáticas borrará la restricción FOREIGN KEY que relaciona ambas tablas.

## SQL. RENOMBRAR TABLAS

Para renombrar tablas se usa el comando RENAME nombreAnterior TO nombreNuevo.

# VISTAS

## DEFINICIÓN DE VISTA

Es una **tabla lógica** que permite acceder a la información de una o varias tablas. No contiene información por sí misma, sino que su información está basada en lo que contienen otras tablas, llamadas **tablas base**, y siempre refleja los datos de estas tablas.

Permite simplificar el acceso a los datos: a través de una consulta simple permite obtener información que, de otro modo, hubiese requerido una consulta compleja.

Tienen la misma estructura que una tabla y permite las mismas operaciones que estas.

## VISTAS PREDEFINIDAS

Son vistas que el SGBA tiene pre-creadas. Tienen tres prefijos DBA\_ (vistas creadas por administración), USER\_ (creadas por el usuario) y ALL\_ (creadas por todos los usuarios).

## CREACIÓN BORRADO Y CAMBIO DE NOMBRES DE VISTAS

Tienen la siguiente sintaxis:

Texto

Descripción generada automáticamente con confianza bajaWord

Descripción generada automáticamente con confianza media

# ÍNDICES

## DEFINICIÓN DE ÍNDICE

Es un objeto de la BD que se asocia a una tabla, la cual indexa a través de una tabla o varias columnas.

Proporciona un acceso rápido y directo a las filas de una tabla: es un elemento de optimización de la base de datos.

Por defecto, cuando se crea una PRIMARY KEY o una restricción UNIQUE, se crea un índice para el campo o campos. El nombre del índice será el nombre de la restricción. En la vista USER\_CONSTRAINTS puede verse que la restricción tiene un índice asociado.

## CREACIÓN Y BORRADO DE ÍNDICES

Para crear un índice:

Para borrar un índice: