

## Restricciones de Integridad en las BBDD. Nomenclatura

### Restricciones

Las restricciones son reglas, que normalmente se establecen en el momento de crear una tabla, para garantizar la integridad de los datos.

Básicamente, las restricciones obligan a cumplirse ciertas reglas cuando una fila es insertada, borrada o modificada, de forma que la operación se llevará a efecto sólo si se cumplen las restricciones definidas en la tabla.

Podemos contemplar los siguientes tipos de restricciones de integridad de datos:

**NOT NULL:** especifica que la columna no puede contener un valor nulo.

**PRIMARY KEY:** identifica de manera única a cada fila de la tabla mediante una o varias columnas, estas columnas que forman la clave primaria no pueden tener valores nulos.

**FOREIGN KEY:** establece una relación entre una(s) columna(s) de la tabla y otra(s) columna(s) de la tabla referenciada, siendo esta última(s) columna(s) la PRIMARY KEY.

**UNIQUE:** no permite duplicados, pero sí nulos; combinado con NOT NULL es la forma de definir una **clave alternativa**.

**CHECK:** especifica una condición que se debe evaluar a "cierto".

### Nomenclatura

En el momento de crear una restricción, además de especificar las reglas que se deben cumplir, podemos dar un nombre a la misma.

Para establecer los nombres de las restricciones, se puede seguir el siguiente convenio, utilizar nombres descriptivos que empiecen por

<b>PK_</b> NombreTabla	si se trata de una Primary Key
<b>UK_</b> NombreTabla	para una clave única con valores nulos
<b>AK_</b> NombreTabla	para clave única sin valores nulos
<b>FK_</b> Tabla1_ Tabla2	clave ajena de la Tabla1 (hijo) relacionada con la Tabla2 (padre). Las columnas referenciadas en la tabla padre deben ser PK o AK
<b>CK_</b> Tabla_ Columna	una CHECK sobre una Columna de esa Tabla

- Toda clave PRIMARY o UNIQUE genera automáticamente un índice.

- Debemos saber que si no especificamos ningún nombre, el sistema le asignará un nombre único a cada restricción con un formato propio del tipo "SYS\_C001209"

## Ejemplo de ejecución de sentencias

```
CREATE TABLE pieza (  
    numpieza number(10) PRIMARY KEY,  
    nompieza varchar(30),  
    preciovent number(6) CONSTRAINT CK_PIEZA_PRECIOVENT CHECK (preciovent >0)  
);
```

```
CREATE TABLE vendedor (  
    numvend number(5),  
    nomvend varchar(30),  
    nombrecomer varchar(30),  
    direccion varchar(30),  
    telefono varchar(15),  
    poblacion varchar(20),  
    provincia varchar(20),  
    constraint PK_VENDEDOR PRIMARY KEY(numvend)  
);
```

```
CREATE TABLE preciosum (  
    numvend NUMBER(5),  
    numpieza NUMBER(10) CONSTRAINT FK_PRECIOSUM_PIEZA REFERENCES pieza,  
    preciounit number(6),  
    diassum number(3) DEFAULT 2 CONSTRAINT CK_PRECIOSUM_DIASSUM CHECK (diassum>0),  
    CONSTRAINT FK_PRECIOSUM_VENDEDOR FOREIGN KEY (numvend) REFERENCES vendedor,  
    CONSTRAINT PK_PRECIOSUM PRIMARY KEY (numpieza, numvend)  
);
```

## Cómo obtener información sin usar el entorno gráfico

- 1) Obtener información de una tabla

```
DESC nombreTabla          DESC user_tables
```

- 2) Obtener información de las tablas de un usuario

```
SELECT table_name FROM user_tables
```

- 3) Obtener información de las restricciones.

--nombre de todas las restricciones una tabla

```
SELECT * FROM user_constraints  
WHERE table_name = 'NOMBRETABLA'
```

--nombre de todas las restricciones de un determinado tipo, por ejemplo las claves ajenas

```
SELECT * FROM user_constraints  
WHERE constraint_type = 'R'
```

- 4) Obtener lo mismo utilizando nuestra nomenclatura

--nombre de todas las restricciones una tabla

```
SELECT * FROM user_constraints  
WHERE constraint_name like '%NOMBRETABLA%'
```

--nombre de todas las restricciones de un determinado tipo, por ejemplo las claves ajenas

```
SELECT * FROM user_constraints  
WHERE constraint_name like 'FK%'
```