

UT4. DISEÑO FÍSICO DE LA BASE DE DATOS. RESTRICCIONES

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark



A2. DISEÑAR LAS REGLAS DE NEGOCIO PARA EL SGBD ESPECÍFICO

- Las actualizaciones que se realizan sobre las relaciones de la base de datos deben observar ciertas restricciones que imponen las reglas de negocio de la empresa.
- Algunos SGBD proporcionan mecanismos que permiten definir estas restricciones y vigilan que no se violen. Los motores de Bases de Datos relacionales sí contemplan estas restricciones.
- Hay restricciones que imponen las reglas de negocio que se resuelven con otras funcionalidades que aportan los SGBD que veremos en capítulos posteriores. Incluso hay reglas que no se pueden resolver con los SGBD, y se incluyen en programas de aplicación.



RESTRICCIONES (CONSTRAINT)

Puede tener nombre	Restricción	Significado
	NOT NULL	No puede haber valores vacíos en la columna.
	UNIQUE	Todos los valores de la columna tienen que ser únicos (no puede haber valores repetidos). Puede haber varias columnas UNIQUE.
	PRIMARY KEY	Indica la columna que identifica los elementos de la tabla. Sólo puede haber una PRIMARY KEY por tabla, pero puede estar formada por varias columnas.
	FOREIGN KEY	Crea un enlace entre dos tablas a través de una columna. Dicha columna tiene que ser clave en otra tabla.
	CHECK	Comprobaciones sobre una columna para comprobar si su valor es válido conforme a una expresión.
	DEFAULT	Cada columna debe contener un valor (incluido NULL). Cuando se rellena la tabla si no se especifica un valor para la columna ésta se pone a DEFAULT. Proporciona un valor por defecto para la columna a la que se aplica.



RESTRICCIONES NOT NULL Y DEFAULT

- Para cada columna, al crearla, después del tipo se indica si permite o no valores nulos. Si no se indica nada, permite valores nulos.
- También se puede indicar el valor por defecto.

Column_name data_type [NOT NULL | NULL] [DEFAULT default_value]

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid'  
);
```



RESTRICCIONES NOT NULL Y DEFAULT

No estándar

- Modificar Null a Not Null y viceversa:

ALTER TABLE nombreTabla MODIFY nombreColumna *data_type* [NOT NULL | NULL] ;

```
ALTER TABLE Personas MODIFY primerApellido varchar(50) NULL;
```

```
ALTER TABLE Personas MODIFY edad INT NOT NULL;
```

- Modificar restricción Default:

ALTER TABLE nombreTabla ALTER nombreColumna SET DAFAULT valor;

```
ALTER TABLE Personas ALTER edad SET DEFAULT 0;
```

- Eliminar restricción Default:

ALTER TABLE nombreTabla ALTER nombreColumna DROP DAFAULT;

```
ALTER TABLE Personas ALTER edad DROP DEFAULT;
```

Solo si la
columna no
tiene valores
nulos



RESTRICCIÓN CHECK

- Para cada columna, al crearla, se puede indicar las restricciones de tipo check.
- Si no se cumple la restricción de tipo check, no permite realizar la operación (insertar o actualizar).

Column_name data_type [NOT NULL | NULL] [DEFAULT default_value] [CHECK condición]

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL CHECK (edad >= 18),  
    población VARCHAR(50) DEFAULT 'Madrid'  
);
```



RESTRICCIÓN UNIQUE

- Para cada columna, al crearla, después de indicar que no permite valores nulos, se indica que la columna es UNIQUE. Los gestores permiten que se declare la columna como única, y a la vez, que permita valores nulos. Es decir, que en los casos en los que se conozca el valor, el valor no se pueda repetir.
- No tiene sentido aplicar el valor default en una tabla UNIQUE (pero hay gestores que lo permiten)

Column_name data_type [NOT NULL | NULL] [UNIQUE] [DEFAULT default_value]

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL UNIQUE,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid'  
);
```



RESTRICCIÓN UNIQUE

- UNIQUE también se puede aplicar a un conjunto de columnas
- Esta opción también sirve para una columna, y es equivalente a la solución anterior.

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid',  
    UNIQUE (id),  
    UNIQUE (primerApellido, nombre)  
);
```



RESTRICCIÓN PRIMARY KEY

- Funciona igual que UNIQUE y son incompatibles (solo puede aparecer una de las 2 en cada columna). Algunos SGBD permiten que aparezcan ambas restricciones.
- Por definición, una PRIMARY KEY es única, y no puede permitir valores nulos.

Column_name data_type [NOT NULL | NULL] [UNIQUE | PRIMARY KEY] [DEFAULT default_value]

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL PRIMARY KEY,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid'  
);
```



RESTRICCIÓN PRIMARY KEY

- PRIMARY KEY también se puede aplicar a un conjunto de columnas
- Esta opción también sirve para una columna, y es equivalente a la solución anterior.

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50) NOT NULL,  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid',  
    PRIMARY KEY (primerApellido, nombre)  
);
```



RESTRICCIÓN FOREIGN KEY

- Las restricciones FOREIGN KEY se aplican sobre una o varias columnas, pero no se puede indicar en la definición de la propia columna.
- La tabla referenciada (y las columnas referenciadas) debe existir previamente.

***FOREIGN KEY* [nombre] (columna [, columna]...) REFERENCES tablaReferenciada (columnaReferenciada [, columnaReferenciada]...)**

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL PRIMARY KEY,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid'  
    FOREIGN KEY nombreopcional (edad) REFERENCES edades(edad)  
);
```



RESTRICCIONES FOREIGN KEY

- **Integridad referencial o clave ajena (*foreign key*):** Sirve para indicar una clave externa. Cuando una clave se marca con integridad referencial, no se podrán introducir valores que no estén incluidos en los campos relacionados con esa clave. Esto último causa problemas en las operaciones de borrado y modificación de registros, ya que si se ejecutan esas operaciones sobre la tabla principal quedarán filas en la tabla secundaria con la clave externa sin integridad. Esto se puede manipular agregando las siguientes cláusulas:
 - **CASCADE:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado de clave, implica borrar las filas con el mismo valor de clave foránea o modificar los valores de esas claves foráneas.
 - **SET NULL:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado de clave, implica asignar el valor *NULL* a las claves foráneas con el mismo valor.
 - **NO ACTION o RESTRICT:** El borrado de las filas de la relación que contiene la clave primaria referenciada, o la modificación de dicha clave, sólo se permite si no existen filas con dicha clave en la relación que contiene la clave ajena. Es decir, no puedo borrar en la tabla de la clave primaria si en una tabla con clave foránea referencio a ese valor.
 - **SET DEFAULT:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado implica asignar el valor por defecto a las claves foráneas con el mismo valor.



RESTRICCIÓN FOREIGN KEY

- En una restricción FOREIGN KEY, se debe indicar el comportamiento en caso de borrar o actualizar en la tabla que referencia. Si no se indica nada, el comportamiento es RESTRICT o NO ACTION.
- El comportamiento para el borrado y la actualización puede ser distinto.

***FOREIGN KEY* [nombre] (columna [, columna]...) REFERENCES tablaReferenciada (columnaReferenciada [, columnaReferenciada]...) [ON DELETE comportamiento] [ON UPDATE comportamiento]**

Ejemplo:

```
CREATE TABLE Personas (  
    id INT NOT NULL PRIMARY KEY,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid',  
    FOREIGN KEY nombreopcional (edad) REFERENCES edades (edad)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);
```



AÑADIR RESTRICCIONES

- Las restricciones CHECK, PRIMARY KEY, UNIQUE y FOREIGN KEY pueden ser añadidas después de crear las tablas (siempre que no tengan valores que imposibiliten crearlas).
- Las FOREIGN KEY se recomienda hacerlo justo después de crear todas las tablas, y siempre después de crear las PRIMARY KEY.

***ALTER TABLE* nombreTabla ADD restricción**

```
ALTER TABLE personas ADD PRIMARY KEY (id);
```

```
ALTER TABLE personas ADD UNIQUE (primerApellido, nombre);
```

```
ALTER TABLE personas ADD CHECK (edad >= 18);
```

```
ALTER TABLE personas ADD FOREIGN KEY nombreopcional (edad) REFERENCES edades(edad) ON DELETE  
SET NULL ON UPDATE CASCADE;
```



RESTRICCIONES CON NOMBRES

- Las restricciones CHECK, PRIMARY KEY, UNIQUE y FOREIGN KEY pueden ser creadas con un nombre. Si no se indica nombre, el SGBD le asigna un nombre autogenerado.

***ALTER TABLE* nombreTabla ADD [CONSTRAINT [nombre]] restricción**

```
CREATE TABLE Personas (  
    id INT NOT NULL,  
    primerApellido VARCHAR(50) NOT NULL,  
    nombre VARCHAR(50),  
    edad INT NULL,  
    población VARCHAR(50) DEFAULT 'Madrid',  
    CONSTRAINT foranea FOREIGN KEY (edad) REFERENCES          edades(edad) ON  
DELETE SET NULL ON UPDATE CASCADE);
```

```
ALTER TABLE personas ADD CONSTRAINT PRIMARY KEY (id);  
ALTER TABLE personas ADD CONSTRAINT resUnica UNIQUE (primerApellido, nombre);;
```



RESTRICCIONES CON NOMBRES

- Se recomienda crear las restricciones con nombre para poder eliminarlas.

***ALTER TABLE* nombreTabla DROP CONSTRAINT nombre;**

```
ALTER TABLE personas DROP CONSTRAINT foranea;  
ALTER TABLE personas DROP CONSTRAINT resUnica;
```

- **En MySQL no funciona así, y debe hacerse como:**

```
ALTER TABLE personas DROP PRIMARY KEY;  
ALTER TABLE personas DROP FOREIGN KEY foranea;
```



UT4. DISEÑO FÍSICO DE LA BASE DE DATOS. RESTRICCIONES

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

