

2. Lenguaje de definición de datos

Índice

[1. Tablas](#)

[2. Vistas](#)

[3. Índices](#)

1. Tablas

Creación, alteración y eliminación

En primer lugar se debe crear una base de datos (si no existe previamente):

```
CREATE DATABASE books_db;
```

Después hay que seleccionarla:

```
USE books_db;
```

A continuación crear las tablas:

```
CREATE TABLE authors (  
  id INT NOT NULL PRIMARY KEY,  
  name VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE books (  
  id INT NOT NULL PRIMARY KEY,  
  title VARCHAR(50) NOT NULL,  
  idAuthor INT NOT NULL DEFAULT 0  
);
```

Para modificar el esquema de una tabla, por ejemplo añadiendo una fecha:

```
ALTER TABLE books  
  ADD COLUMN publish_date DATE NULL;
```

Y para destruir una tabla o base de datos:

```
DROP TABLE books;  
DROP DATABASE books_db;
```

Integridad referencial

En SQL es posible definir las claves foráneas de una tabla cuando es creada hacia otra creada previamente, y además definir una dependencia de integridad referencial.

Estas dependencias automatizan tareas de escritura (**UPDATE** y **DELETE**) como por ejemplo si se elimina una entrada de una tabla, también se eliminan las entradas de una segunda tabla dependiente cuya clave foránea apunta (tiene una referencia) a la primera tabla.

Para definir la integridad referencial de este caso, se puede hacer por ejemplo del siguiente modo (para un borrado en cascada):

```
CREATE TABLE tablaReferenciada (  
    id INT NOT NULL PRIMARY KEY,  
    ...  
);  
  
CREATE TABLE tablaDependiente (  
    id INT NOT NULL PRIMARY KEY,  
    ...  
    idTablaReferenciada INT NOT NULL DEFAULT 0  
);  
  
ALTER TABLE tablaDependiente  
    ADD FOREIGN KEY (idTablaReferenciada) REFERENCES tablaReferenciada(id)  
    ON DELETE CASCADE;
```

En lugar de **ON DELETE** para borrado, se puede poner **ON UPDATE** para actualización.

En lugar de **CASCADE** para que borre las hijas si se borra el padre, se puede poner:

- **RESTRICT**: NO se borrarían las dependientes aunque se borre la referenciada.
- **SET NULL**: La FK de la dependiente se actualizaría al valor **NULL**.
- **SET DEFAULT**: La FK de la dependiente se actualizaría al valor por defecto (0 en el ejemplo).
- **NO ACTION**: No se haría nada y por tanto la dependiente apuntaría a una PK inexistente.

2. Vistas

[WIP]

3. Índices

[WIP]

Referencias

[Crear base de datos](#)
[Borrar base de datos](#)
[Crear tabla](#)
[Modificar tabla](#)
[Borrar tabla](#)
[Claves foráneas](#)