

Base de datos relacional: Buenas prácticas



Utilizar el guión bajo “ _ ” en vez de un espacio al separar palabras

```
CREATE DATABASE mi_negocio;
```

```
CREATE TABLE mi_tabla(  
    id INT,  
    campo_uno VARCHAR(20)  
);
```

Nombrar las tablas en plural y los campos en singular

```
CREATE TABLE personas (  
    id INT,  
    nombre VARCHAR(20)  
);
```

Al crear la DB utilizar el CHARSET “utf8mb4” para que tome los caracteres especiales como la ñ (utf8mb4_general_ci)

```
CREATE DATABASE ejemplo DEFAULT CHARACTER SET utf8mb4  
COLLATE utf8mb4_general_ci;
```

Todas las tablas deben de tener su id propio (a menos que realmente no sea necesario, puede darse esto en relaciones de muchos a muchos)

```
CREATE TABLE mi_tabla1 (  
    id INT,  
    campo_1 VARCHAR(20)  
);
```

```
CREATE TABLE mi_tabla2 (  
    id INT,  
    campo_1 VARCHAR(20)  
);
```

- Al realizar consultas SELECT siempre nombrar la tabla y luego el campo ej: **tabla1.id**
- Siempre utiliza los alias “AS” para nombrar a la columna del resultado
- No utilizar MAYÚSCULAS! ni para nombres de Bases de datos, ni para tablas ni para nombrar los campos

Nombrar una “Restricción de llave foránea” **fk_nombre_de_la_tabla_id**, ejemplo:
fk_usuario_id

```
ALTER TABLE `aulas_alumnos`  
  ADD CONSTRAINT `fk_alumno_id`  
  FOREIGN KEY (`alumno_id`)  
  REFERENCES `alumnos` (`id`)  
  ON DELETE RESTRICT ON UPDATE CASCADE;
```

Utilizar campos de “estado” (siempre que sea necesario)

Utilizar valores por DEFAULT (siempre que sea necesario)

```
CREATE TABLE mi_tabla1 (  
    id INT,  
    campo_1 VARCHAR(20),  
    estado BOOLEAN NOT NULL DEFAULT TRUE,  
    fecha_de_alta TIMESTAMP NOT NULL DEFAULT  
    CURRENT_TIMESTAMP  
);
```


Aplicando las buenas prácticas siempre vamos a tener una base de datos entendible, mantenible y escalable en el tiempo, pero sobre todo va a hacer nuestro trabajo mucho más prolijo y fácil.