Base de Datos

2024-08-10

# Base de Datos

Siguiendo la estructura de aplicaciones y módulos que hemos visto, los nombres de las tablas de la base de datos tendrán la siguiente forma:

## Esquemas

Creación de un usuario/esquema de base de datos por cada uno de los módulos. El sentido de crear un esquema es por seguridad, para acotar el riesgo que pueda tener el entrar un usuario al resto de tablas de otro esquema. Ejemplo: sede, app, pd, etc.

A las tablas de otros esquemas se les dará solo acceso en modo consulta cuando sea necesario.

Para el resto de las operaciones CRUD que no sean consultas se harían directamente por peticiones a los servicios de su mantenimiento.

## Tablas

El nombre de las tablas irá siempre en minúscula, separando las distintas palabras que forman el nombre con guiones bajos, así pues, tendrá el formato ap\_modulo\_tabla:

* **2 caracteres para indicar la aplicación**, “pd” para el caso de plan director.
* **Módulo al que pertenece la tabla**.
* **Nombre de la tabla en singular**.

Las tablas que no se gestionen específicamente en un módulo comenzarán por “dpj” y en otro caso por el módulo donde se haga concretamente la gestión (sede, app, pd…).

Siempre que sea posible, la clave primaria de la tabla será una sola columna, de tipo SERIAL y tamaño de acuerdo con las necesidades estimadas de la tabla.

Para el caso de tablas que indican la relación N:N entre otras dos tablas, el formato será ap\_modulo\_tabla1\_tabla2.

A la hora de realizar el CREATE TABLE, excluiremos la definición de la clave primaria de la sentencia. Posteriormente crearemos el índice único y por último modificaremos la tabla para añadir la restricción de la clave primaria. Ejemplo:

create table "fcp".pd\_prueba\_indices2  
(  
 x\_indice serial not null,  
 campo1 smallint not null,  
 campo2 integer not null  
);  
  
create unique index "fcp".indice\_prueba on "fcp".pd\_prueba\_indices2 (x\_indice) in dbsindices;  
  
alter table "fcp".pd\_prueba\_indices2 add constraint primary key (x\_indice);

## Columnas

| **Categoría** | **Prefijo** | **Descripción** |
| --- | --- | --- |
| **Clave primaria** | x\_tabla | La clave primaria de la tabla. Comenzarán por x seguido del nombre de la tabla. (Ej. x\_area) |
| **Clave externa** | tab\_x\_tabla | Las columnas que contengan referencias a otras tablas comenzarán por tres caracteres identificadores de la tabla referenciada, seguido del nombre de la clave primaria de dicha tabla (Ej. are\_x\_area). |
| **Código** | c\_\_\_\_\_\_\_ | Las columnas cuyo contenido es un código comenzarán por c (Ej. c\_usuario) |
| **Descripción** | d\_\_\_\_\_\_\_ | Las columnas cuyo contenido es una descripción de hasta 20 caracteres comenzarán por d y tendrán tipo varchar (Ej. d\_email) |
| **Descripción larga** | l\_\_\_\_\_\_\_ | Las columnas cuyo contenido es una descripción de más de 20 caracteres comenzarán por l y tendrán tipo varchar (Ej. l\_pais) |
| **Fecha** | f\_\_\_\_\_\_\_\_ | Las columnas con fechas comenzarán por f y su tipo será datetime con el formato que corresponda. |
| **Número** | n\_\_\_\_\_\_\_\_ | Las columnas con números comenzarán por n (Ej. n\_anio\_objetivo) |
| **Importe** | v\_\_\_\_\_\_\_\_ | Las columnas con importes comenzarán por v y tendrán el formato decimal(12,2) |
| **Indicativos** | i\_\_\_\_\_\_\_\_ | Las columnas con indicadores booleanos comenzarán por i y usualmente tendrán formato varchar de un 1 carácter. |
| **Blob** | b\_\_\_\_\_\_\_ | Las columnas que almacenen objetos binarios de gran tamaño comenzarán por b. |
| **Bson** | j\_\_\_\_\_\_\_ | Columnas que contengan cadenas con objetos json (Ej. j\_datos) |
| **Xml** | m\_\_\_\_\_\_ | Columnas que contengan cadenas con formato xml (Ej. m\_definicion) |

## Columnas de auditoría

| **Columna** | **Descripción** |
| --- | --- |
| c\_usu\_alta | Usuario de creación - char(8) |
| f\_alta | Fecha de creación - datetime year to fraction(3) |
| c\_usu\_modificacion | Usuario de la última modificación - char(8) |
| f\_modificacion | Fecha de la última modificación - datetime year to fraction(3) |

En caso de borrado lógico, se debe usar un campo indicador de activo (i\_activo).

En caso de que la funcionalidad pueda depender de la UT con la que está accediendo el usuario, se debe incluir la unidad tramitadora (c\_utramit\_alta, c\_utramit\_modificacion).

Todas las tablas tendrán las columnas para registrar el usuario y la fecha que creó el registro, así como el usuario y la fecha de la última modificación.

### Bloqueo a Nivel de Filas

En tablas que tengan muchas operaciones de insert, update, y delete, se debe cambiar el bloqueo de nivel tabla a nivel de fila.

Si durante el desarrollo de cualquier nueva funcionalidad se detecta algún punto no contemplado aquí, se creará una propuesta para dar cobertura al caso. Esta propuesta deberá ser expuesta y aprobada por el grupo encargado de la definición de los estándares recogidos en este documento.