

## 5. INDICES Y VISTAS

### 5.1. Vistas.

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto.

Entonces, una vista almacena una consulta como un objeto para utilizarse posteriormente. Las tablas consultadas en una vista se llaman tablas base. En general, se puede dar un nombre a cualquier consulta y almacenarla como una vista.

Una vista suele llamarse también tabla virtual porque los resultados que retorna y la manera de referenciarlas es la misma que para una tabla.

Las vistas permiten:

- **ocultar información:** permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario solo puede consultar la vista.
- **simplificar la administración de los permisos de usuario:** se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.
- **mejorar el rendimiento:** se puede evitar tipear instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

La sintaxis para crear una vista es la siguiente:

```
create view nombre_vista as
    select atributos
    from tabla
    where condición;
```

Consideraciones:

- Los campos y expresiones de la consulta que define una vista DEBEN tener un nombre. Se debe colocar nombre de campo cuando es un campo calculado o si hay 2 campos con el mismo nombre
- Los nombres de los campos y expresiones de la consulta que define una vista DEBEN ser únicos (no puede haber dos campos o encabezados con igual nombre).
- Podemos realizar consultas a una vista como si se tratara de una tabla.
- Al crear una vista, PostgreSQL verifica que existan las tablas a las que se hacen referencia en ella.
- Se aconseja probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el imaginado.

### 5.2. Índices.

La indexación es una técnica que optimiza el acceso a los datos, mejora el rendimiento acelerando las consultas y otras operaciones. Es útil cuando la tabla contiene miles de registros.

Los índices permiten:

- Para facilitar la obtención de información de una tabla se utilizan índices.
- El índice de una tabla desempeña la misma función que el índice de un libro: permite encontrar datos rápidamente; en el caso de las tablas, localiza registros.
- Una tabla se indexa por un campo (o varios).
- El índice es un tipo de archivo con 2 entradas: un dato (un valor de algún campo de la tabla) y un puntero.
- Un índice posibilita el acceso directo y rápido haciendo más eficiente las búsquedas. Sin índice, se debe recorrer secuencialmente toda la tabla para encontrar un registro.
- El objetivo de un índice es acelerar la recuperación de información.

#### Tipos de índice:

- "primary key": es el que definimos como clave primaria. Los valores indexados deben ser únicos y además no pueden ser nulos. PostgreSQL le da el nombre "PRIMARY". Una tabla solamente puede tener una clave primaria. El índice llamado primary se crea automáticamente cuando establecemos un campo como clave primaria.
- "index": crea un índice común, los valores no necesariamente son únicos y aceptan valores "null". Podemos darle un nombre, si no se lo damos, se coloca uno por defecto. "key" es sinónimo de "index". Puede haber varios por tabla.

**create index** nombre\_indice **on** tabla(atributos);

- "unique": crea un índice para los cuales los valores deben ser únicos y diferentes, aparece un mensaje de error si intentamos agregar un registro con un valor ya existente. Permite valores nulos y pueden definirse varios por tabla. Podemos darle un nombre, si no se lo damos, se coloca uno por defecto.

**create unique index** nombre\_indice **on** tabla(atributos);

#### Ejemplo:

```
create table libros(  
  codigo int not null,  
  titulo varchar(40),  
  autor varchar(30),  
  editorial varchar(15),  
  precio decimal(6,2)  
);  
  
create index I_libros_editorial on libros(editorial);  
  
create unique index I_libros_tituloeditorial on libros(titulo,editorial);
```

### 5.3. Secuencias.

Una secuencia (sequence) se emplea para generar valores enteros secuenciales únicos y asignárselos a campos numéricos; se utilizan generalmente para las claves primarias de las tablas garantizando que sus valores no se repitan (normalmente utilizamos la definición de un campo serial, este tiene asociado una secuencia en forma automática)

Una secuencia es una tabla con un campo numérico en el cual se almacena un valor y cada vez que se consulta, se incrementa tal valor para la próxima consulta.

**Sintaxis para crear una sentencia es la siguiente:**

**create sequence** NOMBRESECUENCIA

**start with** VALORENTERO

**increment by** VALORENTERO

**maxvalue** VALORENTERO

**minvalue** VALORENTERO

**cycle o nocycle;**

- La cláusula "start with" indica el valor desde el cual comenzará la generación de números secuenciales. Si no se especifica, se inicia con el valor que indique "minvalue".
- La cláusula "increment by" especifica el incremento, es decir, la diferencia entre los números de la secuencia; debe ser un valor numérico entero positivo o negativo diferente de 0. Si no se indica, por defecto es 1.
- "maxvalue" define el valor máximo para la secuencia. Si se omite, por defecto es 9223372036854775807.
- "minvalue" establece el valor mínimo de la secuencia. Si se omite será -9223372036854775808.
- La cláusula "cycle" indica que, cuando la secuencia llegue a máximo valor (valor de "maxvalue") se reinicie, comenzando con el mínimo valor ("minvalue") nuevamente, es decir, la secuencia vuelve a utilizar los números. Si se omite, por defecto la secuencia se crea "nocycle", lo que produce un error si supera el máximo valor.

Si no se especifica ninguna cláusula, excepto el nombre de la secuencia, por defecto, comenzará en 1, se incrementará en 1, el mínimo valor será -9223372036854775808, el máximo será 9223372036854775807 y "nocycle".

Usar una secuencia al llenar un valor en una columna:

```
Nextval('nombre_secuencia');
```