Vistas en postgres

Jesús Reyes Carvajal

Definición de una vista

- Los usuarios que acceden a una base de datos relacional, lo hacen típicamente a través de vistas, de modo que diferentes usuarios tienen diferentes vistas.
- Una vista, en sí, es una “tabla virtual” derivada, con nombre. El término virtual significa que la tabla no existe como tal, pero para el usuario si parece existir.
- Las vistas no se sustentan en datos almacenados físicamente, solo se almacena su definición en el catálogo de sistema, y esta construida en base a otras tablas.
- Las vistas tienen la misma estructura que una tabla : filas y columnas. los datos se recuperan mediante una consulta SELECT y se presentarán igual que los de una tabla.

Sintaxis de una vista

La sintaxis de definición de una vista en SQL es:

CREATE [OR REPLACE] VIEW nombre_vista AS sentencia_SELECT

Ejemplo sencillo :

```
CREATE VIEW datos AS
    SELECT * FROM empleados
    WHERE apellido LIKE ‘A%’;
```

Ventajas

Seguridad: Cada vista proporciona un nivel de seguridad, cada persona puede tener acceso a determinada información.

Simplicidad: Las vistas permiten ocultar la complejidad de los datos, creando una vista como resultado de una combinación de tablas se puede ocultar la complejidad al usuario.

Diferentes perspectivas de la base de datos: Proporciona diversos resultados de información basados en los mismos datos, mostrar información desde distintos ángulos nos ayuda a crear ambientes de trabajo.

Ejecución y eliminación de vistas

Las vistas pueden tener diferentes llamados

Llamado sencillo:
SELECT * FROM nombre_de_la_vista;

Llamado condicionado:
SELECT * FROM nombre_de_la_vista
WHERE condición(es);

Eliminación de una vista

DROP VIEW nombre_de_la_vista;

Crear vistas con reglas

Las vistas en Postgres se implementan utilizando el sistema de reglas

1. Forma básica de crear una vista llamada team

```
CREATE OR REPLACE VIEW team AS SELECT * FROM equipos;
```

```
equipos:
cod_equ | director | cod_nac | nomb_equ
---------+----------+---------+----------
 e1      | juan     | n1      | Sky
 e2      | pedro    | n2      | Movistar
(2 rows)
```

2. Creación de la vista team con una regla

```
CREATE TABLE team(
        cod_equ  varchar(5),
        director varchar(30),
        cod_nac  varchar(5),
        nomb_equ varchar(30));
```

```
CREATE RULE "_RETURN" AS
ON SELECT TO team
DO INSTEAD SELECT * FROM equipos;
```

```
team;
 cod_equ | director | cod_nac | nomb_equ
---------+----------+---------+----------
 e1      | juan     | n1      | Sky
 e2      | pedro    | n2      | Movistar
(2 rows)
```

Crear una vista actualizable

```
CREATE or REPLACE RULE insertar_equipos AS ON insert TO team
      DO INSTEAD insert into equipos values (
         new.cod_equ,
         new.director,
         new.cod_nac,
         new.nomb_equ);
```

```
ciclismo=# insert into team (cod_equ,nomb_equ,director,cod_nac)
values('e6','Arkea','Carlos Ortyz','n23');
```

```
select * from team;
 cod_equ |   director   | cod_nac | nomb_equ
---------+--------------+---------+----------
 e1      | juan                   | n1          | Sky
 e2      | pedro          | n2                  | Movistar
 e6      | Carlos Ortyz | n23         | Arkea
```

```
select * from equipos;
 cod_equ |   director   | cod_nac | nomb_equ
---------+--------------+---------+----------
 e1      | juan                   | n1      | Sky
 e2      | pedro                  | n2      | Movistar
 e6      | Carlos Ortyz   | n23     | Arkea
```

Vistas materializables

Las vistas materializables son vistas actualizables, donde a través de la sentencia REFRESH MATERIALIZED VIEW nomb_vista; se refrescan los datos en la vista con base en los datos de la tabla.

CREATE TABLE ciudades (id int, nombre varchar(50));

INSERT INTO ciudades VALUES (1,'Bogotá'),(2,'Cali');
CREATE MATERIALIZED VIEW vciudades AS SELECT * FROM ciudades;

SELECT * FROM vciudades;

```
id | nombre
----+--------
  1 | Bogotá
  2 | Cali
```

INSERT INTO ciudades values (3,'Villavicencio');

SELECT * FROM vciudades;

```
id | nombre
----+--------
  1 | Bogotá
  2 | Cali
```

REFRESH MATERIALIZED VIEW vciudades;

SELECT * FROM vciudades;

```
id | nombre
----+--------
  1 | Bogota
  2 | Cali
  3 | Villavicencio
```