

Vistas

Antonio Espín Herranz

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.
- La vista se ejecuta cada que se ejecuta, y se pueden utilizar dentro de otras consultas.

Vistas

- 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.
- Podemos crear vistas con: un subconjunto de registros y campos de una tabla; una unión de varias tablas; una combinación de varias tablas; un resumen estadístico de una tabla; un subconjunto de otra vista, combinación de vistas y tablas.
- Una vista se define usando un "select".

Vistas

- Sintaxis:

```
create view NOMBREVISTA as  
SENTENCIAS SELECT  
from TABLA;
```

- Se puede utilizar: `select * from NOMBREVISTA;`

Vistas

- Ejemplo:

```
create view vista_empleados as
select (apellido || ' ' || e.nombre) as nombre, sexo,
s.nombre as seccion, cantidadhijos
from empleados as e
join secciones as s
on codigo=seccion
```

Vistas

- Vistas con encabezados:
- Son como los alias de las consultas
create view vista2(**idcat,nombrecat**) as
select id, nombre
from tbcategorias order by 2 desc;
- Para **borrar: drop view** nombreVista;

Vistas

- Para modificar una vista:

create or replace view vista2(idcat,nombrecat) as

select id, nombre

from tbcategorias order by 2 desc limit 5;

- Si hemos modificado los alias de las columnas no nos dejará actualizar.
- Podemos hacer ***drop view nombre_vista*** y luego volver a lanzar el comando anterior

Vistas sintaxis

Create [or replace] [temp | temporary] [recursive]

View nombre_vista [(col1, col2, ...)]

As SQL

[with [cascaded | local] | check option]

- **Temp** o **temporary**: Cuando se crea la vista de forma temporal y esta se borra cuando termina la sesión.
- **Recursive**: un SQL recursivo
- **Col1, col2, ...** : nombres de las columnas.
- **Cascaded / local**: si tenemos vistas anidadas se continúan o no comprobando las condiciones de las vistas. Cascaded es por defecto
- **With check option**: Cuando activamos esta opción y hacemos inserciones o actualizaciones se comprueba que los datos que se insertan o actualizan cumplen las condiciones de la vista.
 - Si este comando no se especifica e insertamos filas a través de los comandos insert o actualizamos con update puede crear o actualizar filas que no sean visibles en la vista.

Vistas actualizables

- Insert into a través de una vista, **debe de cumplir:**
 - La vista debe tener exactamente una entrada en su FROM lista, que debe ser una tabla u otra vista actualizable.
 - La definición de vista no debe contener cláusulas WITH, DISTINCT, ni en el nivel superior. GROUP BY HAVING LIMIT OFFSET
 - La definición de vista no debe contener operaciones de conjunto (UNION o INTERSECT) EXCEPT en el nivel superior.
 - La lista de selección de la vista no debe contener agregados, funciones de ventana o funciones de devolución de conjuntos.

Vistas actualizables

- With check option → Valida la condición de la vista
create view insertarCat as
select id, nombre from tbcategorias **where id > 15**
with check option;
- insert into insertarCat(id, nombre) values(**12**, 'xxx'); → **ERROR**

Vistas vs Vistas Materializadas

- Las **vistas** se almacenan en **memoria** y las vistas materializadas se almacenan físicamente en una tabla.
- **Create view** → vista
 - Cuando se ejecuta el SQL los datos se actualizan en la vista en memoria.
- **Create materialized view** → vista materializada.
 - Para que los datos de una vista materializada se actualicen se utiliza:
 - **Refresh materialized view.**

Vistas materializadas

Create **materialized** view nombre_tabla [(col1, .., colN)]
[tablespace nombre_tablespace]

As SQL

- Se puede especificar **un tablespace** para ver donde tenemos que colocar la vista materializada.
- La vista materializada se comporta como una tabla.
- Para **actualizar** los datos:
 - **Refresh** materialized view nombre-vista
- **Ventajas:**
 - Si tenemos muchos datos y la información nos puede servir del día anterior. Podemos elegir cuando queremos actualizarla, si es una vista normal cada vez que se ejecute tiene que volver a calcular todo. La vista materializa no afecta al rendimiento porque es como consultar una tabla.

Vistas del sistema

- Postgre proporciona un conjunto de vistas para obtener información:
 - Empiezan por **pg_XXXX**
- Consultar en:
 - <https://www.postgresql.org/docs/current/views.html>
- Por ejemplo, mostrar información de las tablas:
 - Select * from **pg_tables**;
 - Select * from **pg_views**;
- Para ver los campos de la vista desde psql lanzamos el comando: **\d nombre_vista**
- Y si queremos ver el SQL asociado a la vista: **\d+ nombre_vista**