

HINTS

CONOCER LA SINTAXIS DE LAS DECLARACIONES DE POSTGRESQL

Es posible que le interese conocer la sintaxis de diferentes declaraciones de bases de datos, tales como: **SELECT**, **ALTER**, **DROP**, entre otros.

Si desea aprender más sobre la declaración **DROP**, **\h DROP TABLE** puede ayudarte con esto:

```
primera=# \h DROP TABLE
Orden:      DROP TABLE
Descripción: elimina una tabla
Sintaxis:
DROP TABLE [ IF EXISTS ] nombre [, ...] [ CASCADE | RESTRICT ]

URL: https://www.postgresql.org/docs/13/sql-droptable.html
```

Supongamos que desea analizar los tiempos de ejecución de las consultas. Veamos ahora cómo se puede hacer desde el shell psql.

CONOCER LOS TIEMPOS DE EJECUCIÓN DE CONSULTAS

Para ello, tendrá que dar el comando **\timing**. Si desea analizar el tiempo de ejecución de la siguiente consulta:

```
1 select * from directorio_telefonico limit 10;
```

Primero da el comando **\timing** -:

```
primera=# \timing
El despliegue de duración está activado.
primera=# _
```

Una vez habilitada la opción de tiempo, puede ejecutar su consulta:

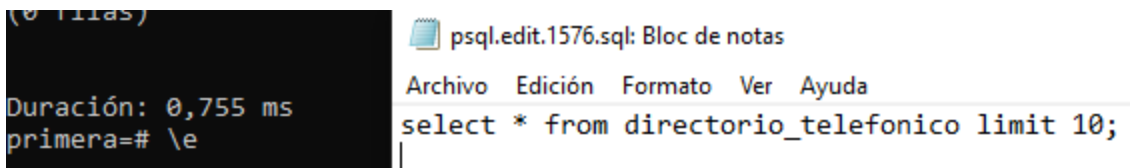
```
primera=# \timing
El despliegue de duración está activado.
primera=# select * from directorio_telefonico limit 10;
 nombre | apellido | numero_telefonico | direccion | edad
-----+-----+-----+-----+-----
(0 filas)

Duración: 0,755 ms
```

Como se puede observar, el tiempo que tardó en ejecutar la consulta es de 0,755 ms. Éste puede variar de una máquina a otra. Ahora verá el comando psql final, que proporciona su comando/consulta ejecutado previamente en un editor de texto.

PSQL + EDITOR DE TEXTO

Si ejecuta el comando `\e`, abre el último comando/consulta ejecutado escrito en un editor de texto, y esto también le permite editarlo y volver a ejecutarlo.



```
(0 filas)

Duración: 0,755 ms
primera=# \e
psql.edit.1576.sql: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
select * from directorio_telefonico limit 10;
```

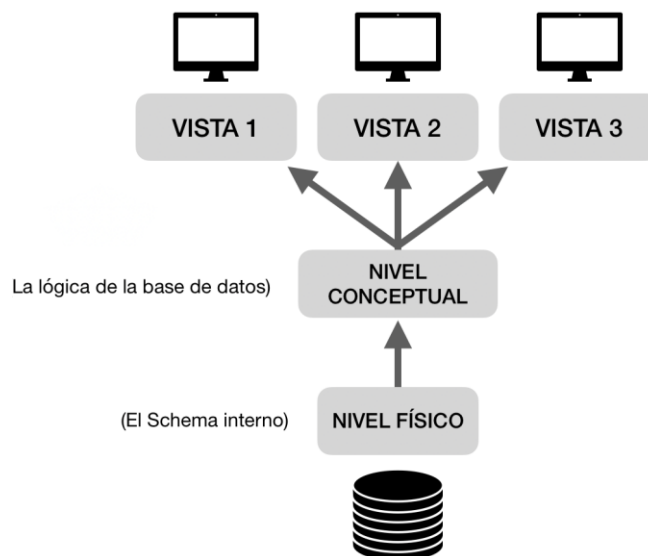
Como puede ver, el comando `\e` abre el editor de texto del Bloc de notas, con la consulta ejecutada anteriormente.

ABSTRACCIÓN DE DATOS

Los sistemas de bases de datos son compuestos de estructuras de datos complejas. Para hacer que el sistema sea eficiente en términos de recuperación de datos, y reducir la complejidad en términos de usabilidad de los usuarios, los desarrolladores utilizan la **abstracción**, que es el proceso de ocultar detalles irrelevantes para simplificar el diseño de la base de datos.

¿Qué entendemos por detalles irrelevantes? Veamos un ejemplo: si queremos acceder a un mensaje de nuestro correo electrónico, no se nos indicará dónde se almacenan físicamente sus datos, ya sea en un servidor en Chile o en Singapur. En realidad, no es relevante la ubicación del correo electrónico, sino conocer su contenido, por lo que, en este caso, la abstracción de datos reduce esa información, mostrándonos solamente la que realmente es importante.

EL PROCESO DE ABSTRACCIÓN



Éste consta de tres niveles:

Físico: es el nivel más bajo de abstracción de datos. Nos indica cómo se almacenan realmente los datos en la memoria. Supongamos que necesitamos almacenar los detalles de un empleado, aquí los bloques de almacenamiento, y la cantidad de memoria utilizada para estos fines se mantienen ocultos para el usuario.

Lógico: este nivel comprende la información que realmente se almacena en la base de datos en forma de tablas. Podemos almacenar los diversos atributos de un empleado, y sus relaciones con el administrador.

Vista: es el nivel más alto de abstracción. Los usuarios solo ven una parte de la base de datos real. Existe para facilitar el acceso a la base de datos por parte de un usuario individual. Dónde éstos ven los datos en forma de filas y columnas. Las tablas y las relaciones se utilizan para almacenar datos,

y pueden existir múltiples vistas de la misma base. Entonces, los usuarios pueden simplemente ver los datos e interactuar con éstos, pero los detalles de almacenamiento e implementación están ocultos.