

# Práctica 22

## Sentencias Preparadas

### Objetivo

*En esta práctica se analiza la forma de crear y ejecutar sentencias preparadas con varios ejemplos y se solicita al alumno que realice varios ejercicios para demostrar su comprensión del tema.*

### Introducción

La sentencia PREPARE permite almacenar un comando SQL y asignarle un nombre al cual nos podemos referir después para su ejecución, su sintaxis básica es:

**PREPARE nombre AS sentencia**

En la cual puede definirse una *sentencia* SELECT, INSERT, UPDATE, DELETE a ejecutar. Solo se procesan sentencias simples, no se aceptan sentencias múltiples.

Dentro de la sentencia se pueden usar parámetros, que se identifican como \$1, \$2, etc., dependiendo de orden en que se deben proporcionar los valores que los reemplacen. Esto permite ejecutar varias veces la misma sentencia, pero proporcionando diferentes valores en cada caso, ya sea usando diferentes variables o asignando distintos valores a las mismas variables.

Si defino una sentencia preparada con un nombre existente, la versión previa es descartada y reemplazada por la nueva. Si la nueva tiene error, entonces ninguna versión se conserva.

El alcance de las sentencias preparadas solo es durante la sesión en la que se crea.

La sintaxis de la sentencia EXECUTE es:

**EXECUTE nombre [USING @var1 [, @var2] ...]**

Para ejecutar una sentencia que previamente fue preparada, solamente debemos hacer referencia a su nombre, pero si contiene parámetros, debemos emplear la cláusula USING, y una listade variables de usuario que contengan los valores que se deben utilizar. La cantidad de variables a enlistar debe coincidir exactamente con la cantidad de signos de interrogación en la sentencia preparada.

La única forma de proporcionar estos valores es mediante variables de usuario.

Para eliminar una sentencia preparada se utiliza la sentencia DEALLOCATE PREPARE que equivale a DROP PREPARE, cuya sintaxis es:

```
{DEALLOCATE | DROP} PREPARE nombre
```

Algunas razones para utilizar sentencias preparadas son:

- Probar cómo funcionan las sentencias antes de incluirlas dentro de un código.
- Cuando no tenemos acceso a un lenguaje de programación de procedimientos.
- Para detectar y resolver fallas en alguna aplicación.
- Crear casos de prueba para identificar problemas.

## Equipo necesario

Una computadora con sistema operativo **Windows** que cuente con **psql**.

## Metodología

1. Ejecute la herramienta **SQL Shell (psql)**.
2. Deje el usuario **postgres** y proporcione la contraseña definida en la primera práctica.
3. Mediante el comando **PREPARE**, se puede definir una determinada operación que se va a ejecutar posteriormente, y asignarle el nombre con el que se va a identificar, por ejemplo, la siguiente sentencia **SELECT**, permite calcular la hipotenusa de un triángulo rectángulo:

```
postgres=# PREPARE Hipo AS SELECT SQRT(POW($1,2) + POW($2,2))  
AS hipotenusa;
```

4. Luego, mediante el comando **EXECUTE** se puede calcular la hipotenusa de diversos triángulos indicando sus catetos:

```
postgres=# EXECUTE Hipo (3, 4);  
postgres=# EXECUTE Hipo (5, 12);  
postgres=# EXECUTE Hipo (8, 15);  
postgres=# EXECUTE Hipo (7, 24);
```

5. Obtenga una imagen completa de la pantalla, mostrando el resultado de las ejecuciones de esta sentencia preparada y consérvela como evidencia.
6. La sentencia se puede ejecutar tantas veces como desee, y cuando una sentencia que ya no se necesita se emplea el comando DEALLOCATE PREPARE para descartarla.  

```
postgres=# DEALLOCATE PREPARE Hipo;
```
7. Un comando preparado solo perdura mientras la sesión esté activa, si es que no se elimina antes y solamente el usuario en la conexión donde lo define es el que lo puede usar.



8. Ahora cambie a la base de datos **banco**.

```
postgres=# \c banco
```

9. Liste las tablas contenidas en esta base de datos:

```
banco=# \d
```

10. Con la siguiente sentencia, se listan los datos del cliente con clave 16:

```
banco=# SELECT * FROM Cliente WHERE Clavecte=16;
```

11. A continuación, al alumno deberá escribir una sentencia preparada, con el nombre de **UNO**, en la que se ejecute una sentencia **SELECT** similar a la anterior, reemplazando el **16** por **\$1**, para poder listar los datos de un cliente proporcionando su clave mediante el parámetro de entrada.

12. A continuación, realice las siguientes pruebas:

```
banco=# EXECUTE UNO (12);
```

```
banco=# EXECUTE UNO (41);
```

```
banco=# EXECUTE UNO (35);
```

13. Obtenga una imagen completa de la pantalla, mostrando la sentencia preparada y el resultado de las ejecuciones de esta sentencia y consérvela como evidencia.

14. Como segundo ejercicio, el alumno deberá escribir una sentencia preparada, con el nombre de **DOS**, en la que se ejecute una sentencia **UPDATE**, en la que se reemplace el nombre del cliente con el valor **\$1**, dada una determinada clave **\$2**.

15. Los valores el nuevo nombre y el de la clave se van a proporcionar como argumentos en el momento de la ejecución de la sentencia.

16. Pruebe la sentencia para cambiar a **Rita**, el nombre del cliente con clave **41** y el nombre del cliente **12** a **Jorge**.

```
banco=# EXECUTE DOS ('Rita', 41);
```

```
banco=# EXECUTE DOS ('Jorge', 12);
```

17. Ejecute la siguiente consulta para confirmar que se hicieron los cambios:

```
banco=# SELECT * FROM Cliente;
```

18. Obtenga una imagen completa de la pantalla, mostrando la sentencia preparada y el resultado de las ejecuciones de esta sentencia y consérvela como evidencia.

19. El tercer ejercicio consiste en que el alumno debe crear una sentencia preparada, con el nombre de **TRES**, que borre un registro de la tabla Cliente, mediante una sentencia **DELETE**, donde se indica la clave del cliente a eliminar mediante el parámetro **\$1**.

20. Y a continuación, pruebe su sentencia eliminando al cliente **16**:

```
banco=# EXECUTE TRES (16);
```

```
banco=# SELECT * FROM Cliente;
```

21. Obtenga una imagen completa de la pantalla, mostrando la sentencia preparada y el resultado de las ejecuciones de esta sentencia y consérvela como evidencia.
22. Como ejercicio final, se pide al alumno que escriba una sentencia preparada, con el nombre de **CUATRO**, que ejecute una sentencia **INSERT** donde agregue los datos de un nuevo cliente, que serán proporcionados como parámetros \$1, \$2, \$3 y \$4.
23. Pruebe su sentencia insertando los datos de estos dos clientes:  

```
banco=# EXECUTE CUATRO (86, 'Max', 'Smart', '443 107 1664');  
banco=# EXECUTE CUATRO (99, 'Gina', 'Torre', '443 376 5693');
```
24. Y ejecute la consulta:  

```
banco=# SELECT * FROM Cliente;
```
25. Obtenga una imagen completa de la pantalla, mostrando la sentencia preparada y el resultado de las ejecuciones de esta sentencia y consérvela como evidencia.
26. Cierre la consola de **psql**.
27. Fin de la Práctica.

## Evidencias

El alumno deberá enviar al instructor las evidencias requeridas durante la realización de la práctica.

## Sugerencias didácticas

El instructor deberá atender a los alumnos que tengan dificultades en la interpretación y la realización de las instrucciones de la práctica.

## Resultados

Se aprendió a crear y ejecutar sentencias preparadas mediante la consola **psql**, mediante diversos ejercicios ilustrativos con las sentencias SQL más comunes.

## Bibliografía

- <https://www.postgresql.org/docs/15/sql-prepare.html>