

Actividad de investigación sobre procedimientos almacenados en PostgreSQL

1- ¿Qué es un procedimiento almacenado?

R/. estos son bloques de código SQL los cuales me permiten utilizar lógica, reutilizar código y optimizar el rendimiento, es decir, es una función diseñada para realizar una acción específica. También nos permite escribir procedimientos más complejos a diferencia de escribir un SQL solo

Estos bloques se guardan en la base de datos, a diferencia de las funciones estos no devuelven un valor directamente, es decir en la consola no se mostrara un valor o una tabla como con las funciones, a menos que . A parte de que son muy útiles para automatizar tareas repetitivas.

2- ¿Cuál es la diferencia entre una función y un procedimiento en postgresql?

R/ la diferencia entre estas dos radica en su propósito y en cómo se utilizan. Una función siempre devuelve un valor (escalares, registros o tablas) y puede ser usada dentro de consultas SQL, como en un `SELECT`. En cambio, un procedimiento no devuelve valores directamente, sino que ejecuta un bloque de instrucciones y se llama con `CALL`, siendo útil para tareas que modifican el estado de la base de datos. Las funciones permiten ser usadas dentro de triggers, vistas y expresiones, mientras que los procedimientos no. Además, las funciones deben tener una cláusula `RETURN`, pero los procedimientos no la requieren.

3- Sintaxis general para crear, ejecutar y eliminar procedimientos.

R/.

Para ejecutar un procedimiento se utiliza `CALL` de esta manera;

`CALL nombre_procedimiento;`

Y para eliminarlos; `DROP PROCEDURE nombre_procedimiento;`

```
1  create or replace procedure nombre_procedimiento(parámetro tipo [, ...])
2  language plpgsql
3  as $$
4  begin
5
6  end;
7  $$;
```

4- ¿Qué son los parámetros de entrada y salida?

R/. Los parámetros de entrada y salida en PL/pgSQL permiten enviar y recibir información desde funciones o procedimientos almacenados. Los parámetros de entrada `IN` son usados

para proporcionar datos al procedimiento; son definidos por el usuario al momento de la llamada y su valor no cambia dentro del procedimiento. Por otro lado, los parámetros de entrada OUT permiten devolver resultados hacia quien llamó al procedimiento, como mensajes, cálculos o estados. También existen los parámetros INOUT, que funcionan tanto para recibir como para retornar información.

5- ¿Cómo se manejan transacciones dentro de procedimientos?

R/. Dentro de los procedimientos almacenados, el manejo de transacciones permite controlar de forma precisa cómo se ejecutan y confirman los cambios en la base de datos. Una transacción se inicia con la instrucción `BEGIN`, se finaliza correctamente con `COMMIT` y se revierte completamente con `ROLLBACK` si ocurre un error. En procedimientos, también se puede utilizar `SAVEPOINT` para establecer puntos intermedios dentro de una transacción, permitiendo deshacer parcialmente sin afectar toda la operación. Además, se puede usar un bloque `EXCEPTION` para capturar y manejar errores, lo que brinda mayor control y seguridad sobre la ejecución del procedimiento.

```
1  CREATE OR REPLACE PROCEDURE transferir_dinero(  
2      origen_id INT,  
3      destino_id INT,  
4      monto NUMERIC  
5  )  
6      LANGUAGE plpgsql  
7      AS $$  
8      BEGIN  
9          BEGIN  
10             IF (SELECT saldo FROM cuentas WHERE id = origen_id) < monto THEN  
11                 RAISE EXCEPTION 'Saldo insuficiente';  
12             END IF;  
13  
14             UPDATE cuentas SET saldo = saldo - monto WHERE id = origen_id;  
15             INSERT INTO movimientos(cuenta_id, tipo, monto) VALUES (origen_id, 'salida', monto);  
16  
17             UPDATE cuentas SET saldo = saldo + monto WHERE id = destino_id;  
18             INSERT INTO movimientos(cuenta_id, tipo, monto) VALUES (destino_id, 'entrada', monto);  
19  
20             COMMIT;  
21  EXCEPTION WHEN OTHERS THEN  
22      ROLLBACK;  
23      RAISE NOTICE 'Transacción cancelada: %', SQLERRM;  
24  END;  
25  END;  
26  $$;  
27
```

6- Ventajas y desventajas de usar procedimientos almacenados:

R/.

sus principales ventajas se destacan la mejora en el rendimiento, ya que el código se ejecuta directamente en el servidor; la reutilización del código, lo que evita

repetir instrucciones en distintas partes de la aplicación; y una mayor seguridad, al permitir controlar el acceso a la lógica de negocio desde la base de datos. También facilitan el mantenimiento, ya que cualquier cambio se realiza en un solo lugar. Sin embargo, entre sus desventajas está la dificultad de depuración, ya que puede ser más complejo encontrar errores en el código dentro del servidor; la portabilidad limitada, ya que cada sistema gestor de bases de datos tiene su propia sintaxis; y la sobrecarga en la base de datos si se abusa de su uso, lo que puede impactar en el rendimiento general del sistema.

7- Ejemplos simples de procedimientos.

R/.

```
1  --ejemplo simple para insertar un cliente
2
3
4  CREATE OR REPLACE PROCEDURE INSERTAR_CLIENTE(
5      IN P_NOMBRE TEXT,
6      IN P_CORREO TEXT
7  )
8      LANGUAGE PLPGSQL
9      AS $$
10     BEGIN
11         INSERT INTO CLIENTES(NOMBRE, CORREO) VALUES (P_NOMBRE, P_CORREO);
12     END;
13     $$;
14
```

```
1  --procedimiento simple para eliminar un usuario
2
3  CREATE OR REPLACE PROCEDURE ELIMINAR_USUARIO(
4      IN P_ID_USUARIO INT
5  )
6      LANGUAGE PLPGSQL
7      AS $$
8      BEGIN
9          DELETE FROM USUARIOS WHERE ID_USUARIO = P_ID_USUARIO;
10     END;
11     $$;
12
```

```
1  -- procedimiento simple para guardar el stock e un producto
2
3  CREATE OR REPLACE PROCEDURE ACTUALIZAR_STOCK(
4      IN P_ID_PRODUCTO INT,
5      IN P_NUEVO_STOCK INT
6  )
7      LANGUAGE PLPGSQL
8      AS $$
9      BEGIN
10         UPDATE PRODUCTOS SET STOCK = P_NUEVO_STOCK WHERE ID_PRODUCTO = P_ID_PRODUCTO;
11     END;
12     $$;
13
```

❖ Investigación Sobre Oracle

1- ¿Qué es Oracle Database?

R/. Oracle database es un sistema gestor de bases de datos reconocido mundialmente por su potencia, fiabilidad y escalabilidad, esta se ha convertido en una de las plataformas mas robustas en cuanto al manejo de grandes volúmenes de información estructurada, el núcleo de Oracle se basa en el modelo relacional en el cual los datos se relacionan o conectan mediante claves primarias y foráneas. Esta estructura permite realizar consultas complejas, mantener la integridad de los datos y facilitar el acceso concurrente por múltiples usuarios sin comprometer la consistencia. Además, Oracle ofrece soporte para procedimientos almacenados, triggers, vistas, particionamiento de datos y replicación, lo que lo convierte en una herramienta versátil para entornos exigentes.

Una de sus grandes fortalezas es la seguridad. Oracle incluye mecanismos avanzados como cifrado de datos, auditoría, control de acceso basado en roles y protección contra ataques SQL. También destaca por su alta disponibilidad, gracias a tecnologías como Real Application Clusters (RAC) y Data Guard, que permiten mantener la continuidad del servicio incluso ante fallos del sistema.

2- Diferencias clave entre Oracle y PostgreSQL.

R/. Estos dos cumplen funciones similares, presentan diferencias fundamentales en términos de arquitectura, licenciamiento, rendimiento y escalabilidad. Oracle es un sistema de gestión de bases de datos comercial desarrollado por Oracle Corporation. Es ampliamente utilizado en entornos empresariales debido a su robustez, seguridad avanzada y soporte para grandes volúmenes de datos. PostgreSQL, por otro lado, es una base de datos open-source lo que significa que es gratuita y cuenta con una comunidad activa que contribuye a su desarrollo y mejora continua. Oracle requiere una licencia comercial, lo que implica costos elevados para su implementación y mantenimiento. PostgreSQL, en cambio, es completamente gratuito y permite a los usuarios modificar su código fuente según sus necesidades, Oracle esta optimizado para manejar grandes cargas de trabajo en cambio PostgreSQL depende de herramientas externas para lograr una escalabilidad similar, en cuanto a SQL Oracle usa sus propias extensiones personalizadas a diferencia de PostgreSQL esta estricto a los estándares de SQL

3- ¿Qué son los procedimientos almacenados en Oracle?

R/. Los procedimientos almacenados en Oracle son bloques de código PL/SQL que se guardan en la base de datos y pueden ejecutarse repetidamente para realizar tareas específicas. A diferencia de los scripts SQL tradicionales, los procedimientos almacenados permiten encapsular lógica de negocio, mejorar el rendimiento y reducir la redundancia en la ejecución de consultas.

Un procedimiento almacenado en Oracle se define con la instrucción CREATE PROCEDURE, y puede incluir parámetros de entrada (IN), salida (OUT) o ambos (INOUT).

4- ¿Cómo se declara un procedimiento en PL/SQL?

R/. Los procedimientos almacenados pueden recibir parámetros, ejecutar consultas, manipular datos y devolver resultados indirectos a través de variables de salida. Se declaran con la instrucción CREATE PROCEDURE y se invocan con CALL o EXECUTE.

```
1  -- Creación o reemplazo del procedimiento almacenado llamado 'actualizar_saldo'
2  ✓ CREATE OR REPLACE PROCEDURE actualizar_saldo(
3      p_usuario_id IN NUMBER,      -- Parámetro de entrada: ID del usuario cuya cuenta se va a actualizar
4      p_nuevo_saldo IN NUMBER      -- Parámetro de entrada: nuevo saldo que se desea asignar
5  ) AS
6  BEGIN
7      -- Actualiza el campo 'saldo' en la tabla 'cuentas' donde el 'usuario_id' coincida con el dado
8      UPDATE cuentas
9      SET saldo = p_nuevo_saldo
10     WHERE usuario_id = p_usuario_id;
11  END;
12  ✓ /
13
14  -- Declaración del procedimiento para poder utilizarlo desde PL/SQL o herramientas como SQL*Plus
15  -- Esta declaración sirve para llamarlo en un bloque anónimo o desde otro procedimiento
16  DECLARE
17  BEGIN
18
19  END;
20
21  |- Llamada al procedimiento con valores de ejemplo (por ejemplo, ID de usuario 1 y nuevo saldo 500.00)
22
23  call actualizar_saldo(1, 500.00);
```

5- ¿Qué es un bloque anónimo en Oracle?

R/. En Oracle Database, un bloque anónimo es un fragmento de código pl/sql que se ejecuta directamente sin necesidad de ser almacenado en la base de datos. A diferencia de los procedimientos y funciones, los bloques anónimos no tienen un nombre y no se guardan para su reutilización, lo que los hace ideales para pruebas rápidas, ejecución de consultas temporales y automatización de tareas simples. Es también bueno para pruebas rápidas y automatización de tareas y la manipulación temporal de datos.

6- Herramientas gratuitas para practicar (ej. Oracle Live SQL).

R/. - Oracle Live SQ

- Oracle Online Editor – OneCompile
- SQL-Practice.com
- Oracle Dev Gym

7- ¿En qué casos un procedimiento almacenado puede generar más problemas que soluciones?

R/. Los procedimientos almacenados en Oracle son bloques de código PL/SQL que automatizan tareas dentro de la base de datos, mejorando la eficiencia y seguridad. Son útiles para realizar operaciones repetitivas sin escribir consultas manualmente. Sin embargo, pueden generar problemas si no se optimizan bien, afectando el rendimiento del servidor. También pueden complicar el mantenimiento y la portabilidad entre distintos sistemas de bases de datos. Su depuración es difícil, y al estar almacenados internamente, requieren acceso de administrador para modificaciones. Además, su falta de integración con herramientas de versionado puede hacer que los cambios sean difíciles de rastrear. Un bloque anónimo en Oracle, en cambio, es un fragmento de código PL/SQL que se ejecuta sin almacenarse en la base de datos, ideal para pruebas rápidas. Para practicar, herramientas como Oracle live SQL permiten ejecutar consultas en línea sin instalación. Oracle y PostgreSQL tienen diferencias clave en licenciamiento, rendimiento, seguridad y soporte, haciendo que cada uno sea más adecuado para distintos proyectos.

8- ¿Es buena práctica tener mucha lógica en procedimientos dentro de la base de datos?

Justifica.

R/. No siempre es buena práctica incluir demasiada lógica en procedimientos almacenados dentro de la base de datos. Aunque pueden mejorar el rendimiento al reducir la comunicación entre la aplicación y el servidor, también pueden generar problemas de mantenimiento y escalabilidad. La lógica de negocio suele ser más fácil de gestionar en el código de la aplicación, donde se pueden usar herramientas de control de versiones y pruebas automatizadas. Además, los procedimientos almacenados pueden hacer que el sistema sea menos flexible, dificultando la migración a otros motores de bases de datos. En algunos casos, pueden sobrecargar el servidor, afectando el rendimiento general. Por eso, es recomendable usarlos para tareas específicas como validaciones o cálculos críticos, pero no para manejar toda la lógica de negocio.

9- ¿Cómo podría afectar el rendimiento del sistema el uso excesivo de procedimientos?

R/. El uso excesivo de procedimientos almacenados puede afectar el rendimiento del sistema de varias maneras. Si los procedimientos no están bien optimizados, pueden generar cargas innecesarias en el servidor, aumentando el consumo de CPU y memoria. Además, los procedimientos almacenados pueden dificultar la escalabilidad, ya que concentran la lógica en la base de datos en lugar de distribuirla entre la aplicación y otros servicios. También pueden provocar bloqueos en la concurrencia, afectando el acceso simultáneo a los datos. Otro problema es la falta de flexibilidad, ya que modificar procedimientos requiere cambios en la base de datos, lo que puede ralentizar el desarrollo. Finalmente, si los procedimientos almacenados no están bien diseñados, pueden generar consultas ineficientes, afectando el tiempo de respuesta de la aplicación.

➤ Comparativa entre Oracle y PostgreSQL en el manejo de procedimientos.

Oracle y PostgreSQL manejan procedimientos almacenados de manera diferente debido a sus arquitecturas y enfoques de desarrollo. Oracle usa PL/SQL, un lenguaje procedural avanzado con soporte para parámetros IN, OUT y INOUT, mientras que PostgreSQL usa PL/pgSQL, con una sintaxis más estricta y sin INOUT. En Oracle, los procedimientos permiten transacciones inter3nas y pueden ejecutar autonomous transactions, mientras que en PostgreSQL, las transacciones solo se pueden usar dentro de procedimientos, pero no en funciones. En cuanto al manejo de excepciones, Oracle usa EXCEPTION con WHEN OTHERS THEN SQLERRM, mientras que PostgreSQL emplea RAISE EXCEPTION. Oracle ofrece herramientas avanzadas como packages y Data Guard para garantizar estabilidad, pero PostgreSQL sigue estándares abiertos, facilitando la portabilidad.

➤ Ejemplo de procedimiento en Oracle usando pl/SQL.

```
1  CREATE OR REPLACE PROCEDURE actualizar_saldo(
2      p_usuario_id IN NUMBER, -- Recibe el ID del usuario
3      p_nuevo_saldo IN NUMBER -- Recibe el nuevo saldo
4  ) AS
5      v_filas_afectadas NUMBER; -- Variable para verificar si se actualizó algún registro
6  BEGIN
7      -- Actualiza el saldo del usuario en la tabla 'cuentas'
8      UPDATE cuentas SET saldo = p_nuevo_saldo WHERE usuario_id = p_usuario_id;
9
10     -- Verifica cuántas filas fueron afectadas
11     v_filas_afectadas := SQL%ROWCOUNT;
12
13     -- Si no se actualizó ninguna fila, muestra un mensaje de advertencia
14     IF v_filas_afectadas = 0 THEN
15         DBMS_OUTPUT.PUT_LINE('Advertencia: No se encontró el usuario con ID ' || p_usuario_id);
16     ELSE
17         DBMS_OUTPUT.PUT_LINE('Saldo actualizado correctamente para el usuario ' || p_usuario_id);
18     END IF;
19 END;
```

