|  |  |
| --- | --- |
|  | Trabajo grupal o individual |

## Identificación del trabajo

|  |  |
| --- | --- |
| 1. **Módulo:** | 4 |
| 1. **Asignatura:** | Bases de Datos |
| 1. **RA:** | Manejo de Procedimientos almacenados, logs y funciones en PostgreSQL |
| 1. **Docente Online:** | Jacob Felipe |
| 1. **Fecha de entrega**: | 5 de diciembre de 2024 |
|  |  |
|  |  |

## Identificación del/los estudiante/s

|  |  |
| --- | --- |
| **Nombre y apellido** | **Carrera** |
| Christian Aránguiz M. | Técnico en Programación |

## Introducción

Esta cuarta iteración en el desarrollo de la base de datos para la biblioteca de la Universidad PULS-AR se enfoca en consolidar la capacidad del sistema para gestionar, rastrear y analizar el estado de los libros prestados y devueltos, así como los lectores asociados a estos movimientos. A lo largo de esta actividad, se profundiza en el uso de procedimientos almacenados, triggers y funciones para automatizar tareas críticas y garantizar la integridad de los datos, elementos esenciales en cualquier sistema de bases de datos profesional.

El objetivo principal de esta iteración es perfeccionar las consultas que permiten verificar el estado de los libros y establecer una relación clara entre lectores y préstamos, tanto activos como históricos. Para ello, se implementó un procedimiento almacenado que registra las devoluciones actualizando el estado de los préstamos. Además, se creó una tabla de logs complementada con un trigger que registra automáticamente cada devolución, aportando trazabilidad y facilitando auditorías o consultas futuras. Finalmente, se desarrolló una función para calcular el número de libros actualmente prestados, una métrica clave que permite evaluar el uso de los recursos de la biblioteca en tiempo real.

A través de estas mejoras, el sistema no solo cumple con las exigencias planteadas, sino que también simula situaciones prácticas del ámbito profesional, como la necesidad de contar con datos fiables para la toma de decisiones y la automatización de procesos complejos. Este enfoque iterativo refuerza la capacidad de escalabilidad del sistema, asegurando que pueda adaptarse a futuros requerimientos y proporcionar un servicio eficiente a los usuarios.

## Desarrollo

1. Procedimiento almacenado de devolución de libros.

El procedimiento almacenado registrar\_devolucion permite registrar la devolución de libros modificando el estado del préstamo en la tabla prestamos. Tal como se explicó en la actividad anterior, en lugar de eliminar la relación entre el lector y el libro, se actualiza la columna estado a 'devuelto'. Esto asegura trazabilidad y mantiene un historial de préstamos, lo que es fundamental para consultas futuras y para cumplir con las mejores prácticas en bases de datos, creando así un sistema de bases de datos escalable y fácil de mantener en nuevas iteraciones.

La estructura del procedimiento utiliza parámetros (p\_lector y p\_libro) para identificar el préstamo asociado al lector y al libro. A través de un comando UPDATE, se modifica únicamente la fila correspondiente, garantizando precisión. Además, se incluye un mensaje de notificación (RAISE NOTICE) que confirma la operación realizada, útil tanto para depuración como para retroalimentación al usuario.

Este diseño es eficiente y flexible, ya que encapsula la lógica en un solo lugar, permite reusabilidad y evita la pérdida de datos históricos. Así, se prioriza la integridad y consistencia de la base de datos.

Por último, la línea LANGUAGE plpgsql indica que el procedimiento está escrito en el lenguaje procedural de PostgreSQL, conocido como PL/pgSQL. Este lenguaje permite realizar operaciones más complejas que las consultas SQL estándar, como control de flujo (condicionales y bucles), manejo de excepciones y manipulación de variables. Por su parte, los dobles signos de dólar ($$) delimitan el cuerpo del procedimiento, especificando dónde inicia y termina el código PL/pgSQL. Este delimitador es útil porque evita conflictos con comillas simples ('), permitiendo incluirlas dentro del procedimiento sin necesidad de escape, lo que mejora la legibilidad y facilita la escritura de código más limpio.

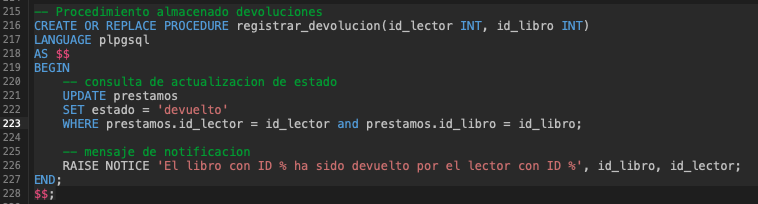


Ilustración 1:Consulta de creación (o modificación) de Procedimiento Almacenado

A screenshot of a computer

Description automatically generated

Ilustración 2: primer llamado de procedimiento almacenado + mensaje de NOTICE

A screen shot of a computer

Description automatically generated

Ilustración 3: segundo llamado de procedimiento almacenado + mensaje de NOTICE

A screenshot of a computer program

Description automatically generated

Ilustración 4: Prueba de actualización de estado en ids seleccionados en procedimiento almacenado

1. Tabla logs de devoluciones.

La tabla log\_devoluciones se creó para registrar automáticamente las devoluciones de libros realizadas en la biblioteca, garantizando trazabilidad y precisión. Incluye un identificador único (id\_log), los IDs del lector y del libro, y un campo de TIMESTAMP que registra la fecha y hora exacta de cada devolución. Esta estructura permite un registro claro y organizado de las devoluciones sin necesidad de intervención manual.

Se desarrolló la función registrar\_log\_devoluciones para ser utilizada por un trigger asociado a la tabla prestamos. La función verifica si el nuevo estado del préstamo es 'devuelto' y, de ser así, inserta automáticamente un registro en la tabla log\_devoluciones con los datos correspondientes. Esto asegura que todas las devoluciones queden registradas sin errores humanos.

El trigger trigger\_log\_devoluciones se activa después de cada UPDATE en la tabla prestamos, verificando fila por fila si el cambio corresponde a una devolución. Al ejecutar el procedimiento de devolución, tres registros fueron generados en la tabla de logs, confirmando su correcto funcionamiento. Además, se realizó una consulta que unió las tablas de logs, lectores y libros, mostrando el nombre del lector, el título del libro y la fecha de devolución, lo que demuestra la integridad y utilidad de los datos.

A screen shot of a computer

Description automatically generated

Ilustración 5: Creación tabla log\_devoluciones

A screenshot of a computer program

Description automatically generated

Ilustración 6: Creación de función que retorna el trigger que se activa si el estado de un prestamo cambia a devuelto

A screenshot of a computer

Description automatically generated

Ilustración 7: prueba de autollenado de tabla log\_devoluciones

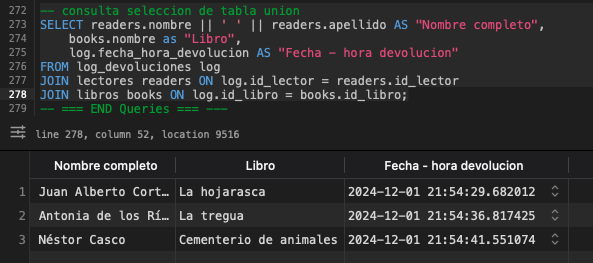


Ilustración 8: Consulta a tabla unión entre log\_devoluciones, libros y prestamos

1. Función libros\_prestados.

La función libros\_prestados se diseñó para contar cuántos libros están actualmente en estado 'prestado' en la tabla prestamos. Este requerimiento permite obtener una métrica clave del sistema de préstamos, que ayuda a monitorear el uso actual de los recursos de la biblioteca de manera precisa y eficiente.

La función utiliza un simple comando SELECT COUNT(\*) con una cláusula WHERE que filtra únicamente las filas cuyo estado sea 'prestado'. Este enfoque es eficiente y directo, ya que se enfoca exclusivamente en los datos relevantes. La función devuelve un entero, que representa la cantidad total de libros prestados en ese momento.

Para verificar su funcionamiento, se llamó a la función con una consulta SQL y se obtuvo el resultado esperado. Esto confirmó que la lógica y estructura eran correctas, proporcionando un resultado preciso y en línea con las expectativas del sistema. La función es reutilizable y puede integrarse fácilmente en reportes o métricas en tiempo real para la biblioteca.

A screen shot of a computer program

Description automatically generated

Ilustración 9: Se declara función para contar libros prestados.

A screenshot of a computer

Description automatically generated

Ilustración 10: resultado de conteo

## Conclusión

En esta actividad se aprendió a integrar herramientas avanzadas de PostgreSQL, como triggers, procedimientos almacenados y funciones, para automatizar procesos y garantizar la integridad de los datos en un sistema de bases de datos. Se comprendió cómo estas herramientas facilitan el manejo eficiente de información en tiempo real, permitiendo rastrear préstamos y devoluciones de libros con precisión.

En la vida real, estas habilidades son esenciales para diseñar y gestionar sistemas que requieren trazabilidad y automatización, como bibliotecas, inventarios o sistemas de gestión de clientes. La implementación de estos conceptos no solo optimiza el flujo de trabajo, sino que también asegura la escalabilidad del sistema para atender necesidades futuras. Esta iteración no solo refuerza los aprendizajes académicos, sino que también desarrolla competencias clave para resolver problemas complejos en entornos profesionales.

## Bibliografía

* **Procedimientos y funciones en PostgreSQL:** https://learn.microsoft.com/es-es/training/modules/procedures-functions-postgresql/