

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 4**

“Blazingly Fast SQL Querys with Index and Use Cases for Triggers”

**Informe Técnico de Laboratorio #2 – SBD2**

Autores: Joaquin Rivas Sanchez

Tutor: Luis Manuel Valera Perez

**La Habana, febrero 2024**

**“Año 69 de la Revolución”**

# Resumen

Este informe técnico aborda el uso de índices y funciones de disparo en bases de datos SQL, específicamente en el contexto de optimización de consultas y mantenimiento de integridad de datos en la Biblioteca Nacional José Martí. Se exploran casos de uso prácticos y se presentan resultados detallados.

En cuanto al uso de índices, se examinan tres métodos para mejorar la velocidad de las consultas de búsqueda de texto, consultas en rango y búsqueda directa. Se muestran análisis exhaustivos de rendimiento y se concluye que el uso adecuado de índices puede optimizar significativamente el rendimiento de las consultas.

En cuanto a las funciones trigger, se presentan diversos casos de uso para garantizar la integridad de los datos y automatizar tareas, como eliminar tuplas relacionadas, registrar auditorías y validar datos ingresados. Se incluyen detalles sobre la implementación de estas funciones y su impacto en la base de datos.

En resumen, el informe destaca la importancia de utilizar índices y funciones de disparo de manera efectiva para mejorar el rendimiento y la funcionalidad de las bases de datos relacionales, adaptándose a las necesidades específicas de la aplicación.

**Palabras clave**: Índices; Funciones de disparo; Bases de datos; Optimización; Integridad de datos; Consultas SQL.

# Introducción

En este trabajo se estudia el uso de índices SQL y funciones trigger en el contexto de la base de datos de la Biblioteca Nacional José Martí de Cuba, una institución que maneja una gran cantidad de información. Los índices SQL son estructuras que optimizan las consultas y búsquedas en la base de datos, mientras que las funciones trigger son procedimientos que se ejecutan automáticamente en respuesta a ciertos eventos o condiciones, garantizando la integridad, la eficiencia y la seguridad de la base de datos. Se analiza la importancia de estos elementos en el rendimiento, la seguridad y la integridad de la base de datos, así como su impacto en la eficiencia de las operaciones diarias de la institución. Se proporcionan recomendaciones y buenas prácticas para la implementación y gestión de índices SQL y funciones trigger en el entorno específico de una Biblioteca Nacional.

**Situación Problemática**: La Biblioteca Nacional José Martí de Cuba, como institución encargada de gestionar un vasto acervo bibliográfico y documental, se enfrenta a desafíos en la eficiencia y seguridad de su base de datos. La necesidad de optimizar la recuperación de información y garantizar la integridad de los datos se presenta como una problemática relevante en su operación diaria.

**Problema Científico**: ¿Cómo puede la Biblioteca Nacional José Martí de Cuba mejorar la eficiencia en la recuperación de información y garantizar la integridad de sus datos a través del uso adecuado de índices SQL y funciones trigger en su base de datos?

**Objeto de Estudio**: El uso de índices SQL y funciones trigger en la base de datos de la Biblioteca Nacional José Martí de Cuba, con el fin de analizar su impacto en la optimización del rendimiento, la seguridad y la integridad de la información almacenada.

**Campo de Acción**: El campo de acción de este estudio se sitúa en el ámbito de la gestión de bases de datos en instituciones bibliotecarias, centrándose específicamente en la implementación y gestión de índices SQL y funciones trigger en la Biblioteca Nacional José Martí de Cuba.

**Tareas de la Investigación**:

Diseñar e implementar estrategias para la optimización del rendimiento y la integridad de la base de datos mediante índices SQL y funciones trigger.

Evaluar el impacto de las mejoras implementadas en la eficiencia y seguridad de la base de datos de la Biblioteca Nacional José Martí.

Proponer recomendaciones y directrices para la gestión efectiva de índices SQL y funciones trigger en el contexto específico de una biblioteca nacional.

A través de estas tareas, se busca contribuir al fortalecimiento de la gestión de la información en la Biblioteca Nacional José Martí de Cuba, mejorando la accesibilidad y seguridad de su base de datos para beneficio de sus usuarios y la preservación del patrimonio bibliográfico y documental del país.

# Desarrollo:

**Contexto del Caso de Estudio:**

Para ilustrar la aplicación de index y trigger functions, consideramos el anterior caso de estudio de la base de datos para la Biblioteca Nacional José Martí. Esta base de datos contiene tablas relacionadas con documentos que entre ellos pueden ser libros, miembros asociados, prestamos, correo electrónico y número de teléfonos entre otras

# Capítulo 1: Uso de index

## Introducción

Los índices son estructuras de datos que se utilizan para mejorar la velocidad de recuperación de registros en una tabla de base de datos. Se pueden crear índices en una o varias columnas de una tabla para permitir búsquedas más eficientes. En nuestro caso de estudio, exploramos la creación de índices en columnas clave para mejorar el rendimiento de consultas frecuentes.

## Búsqueda de Texto

Problemática*:*

Se informó de quejas en la página web de la Biblioteca sobre la lentitud de las respuestas, luego de un análisis extenso se llegó a la conclusión que tiene que ver con el módulo de búsqueda de texto

Se podrá de ejemplo la tabla ‘author’ las consultas serán en este formato %substring%

Refiérase al anexo 1 y anexo 1.1, 1.2, 1.3 para el resultado del explain analyze

Como se aprecia el método 1 no arrojó resultados favorables por lo que se eliminó, el método 2 arrojó los mejores, pero aumenta la complejidad al usar el método gin y diccionarios para hacer las búsquedas, el método 3 hace uso del método gin y usa trigramas (refiérase a la bibliografía), tiene como requisitos el crear una extensión, pero es el más completo y fácil de los 3

## 1.2 Búsqueda de Texto con prefijo

Aquí como anotación se demuestra una tercera forma que es más rápida si se trata de búsquedas en la forma substring%

Refiérase al anexo 2, y anexo 2.1, 2.2 para el resultado del explain analyze

## 2. Consultas en Rango

Problemática*:*

La Biblioteca muy a menudo desea hacer censos sobre sus miembros asociados en cuanto a su edad, pero se cuenta con un enorme dataset que no es comúnmente actualizado, para optimizar este tipo de query se usó índices

Refiérase al anexo 3, y anexo 3.1, 3.2 para el resultado del explain analyze

## 3. Búsqueda directa

Problemática*:*

Se desea brindar una nueva funcionalidad en la página web a sus usuarios en la cual este pueda hacer una búsqueda por nombre exacto de la ubicación de una sala

Refiérase al anexo 4, y anexo 4.1, 4.2 para el resultado del explain analyze

## Conclusiones parciales

Como se pudo ver el uso de índices en todos los casos menos el primer método discutido, arrojaron buenos resultados.

Se recomienda el uso de indices en estos ejemplos utilizados, despues de valorar los posibles contras.

# Capítulo 2: Uso de trigger functions

## Introducción

Las trigger functions son funciones almacenadas en la base de datos que se ejecutan automáticamente cuando ocurren ciertos eventos, como la inserción, actualización o eliminación de registros en una tabla. Estas funciones pueden ser útiles para realizar acciones adicionales, como validar datos, mantener integridad referencial o registrar auditorías.

Entre las formas que puede adoptar un trigger están

Row-level triggers: Estos disparadores se invocan una vez para cada fila afectada por una declaración que disparó el trigger. Cuando se llama a una función PL/pgSQL como un trigger, se definen automáticamente varias variables especiales para describir la condición que provocó la llamada.

Statement-level triggers: A diferencia de los disparadores a nivel de fila, los disparadores a nivel de declaración se invocan una vez para cada declaración, independientemente del número de filas afectadas. Operan en todo el conjunto de filas afectadas por la declaración.

## 1. Triggers en cascada

Problemática*:*

Se desea garantizar que al eliminar una tupla de la tabla phone\_room se elimine también el respectivo teléfono

Revisar anexo 5

Problemática*:*

Se desea garantizar que al eliminar una sala sean también eliminas las respectivas conexiones service\_email, service\_phone y si existe algún miembro que este actualmente usando un servicio de esa sala esta no sea eliminada alertando al usuario de que no se puede llevar a cabo la operación

Revisar anexo 6

## 2. Triggers para auditoría

Problemática*:*

Se desea registrar los cambios hechos por un usuario al monto a pagar de una multa

Revisar anexo 7

Problemática*:*

Se desea registrar la actividad sobre los libros de cualquier operación insert/update/delete

Revisar anexo 11

## 3. Triggers para validación

Problemática*:*

Para asegurar que no sea posible imponer un monto de multa mayor a 1000 pesos con un mínimo de 10 pesos se creó un trigger function que valida la creación de una multa

Revisar anexo 8

Problemática*:*

Se quiere garantizar que no sea posible registrar un préstamo a un miembro trabajador profesional de un documento que sea patrimonio

Revisar anexo 10

Problemática*:*

Se quiere garantizar que al asociar un correo a una librería ambos existan, y así mantener la integridad referencial

Revisar anexo 12

## 4. Triggers para metadatos

Problemática*:*

Se quiere quitar complejidad del backend de la aplicación a la hora de garantizar la fecha de creación de los préstamos, además de garantizar una mayor coherencia del formato, para ello se usó un trigger

Revisar anexo 9

# Conclusiones parciales

El uso de funciones trigger aumentó significativamente la funcionalidad, características y seguridad ofreciendo una base de datos más robusta y mejor

# Conclusiones generales

El uso de índices y trigger functions puede tener un impacto significativo en el rendimiento y la funcionalidad de una base de datos relacional. Los índices permiten acelerar las consultas al proporcionar un acceso más rápido a los datos, especialmente en columnas clave o comúnmente consultadas. Por otro lado, las trigger functions ofrecen la capacidad de automatizar tareas y mantener la integridad y la consistencia de los datos.

Es importante tener en cuenta que el diseño adecuado de índices y trigger functions depende de las características específicas de la base de datos y de los requisitos de la aplicación. Un uso cuidadoso de estas técnicas puede mejorar significativamente el rendimiento y la eficiencia de las operaciones en la base de datos.

# Bibliografía Consultada

*39.1. Overview of Trigger Behavior*. (2024, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/trigger-definition.html>

*39.1. Overview of Trigger Behavior*. (2024b, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/trigger-definition.html>

*11.2. Index types*. (2024, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/indexes-types.html>

*Chapter 70. GIN Indexes*. (2024, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/gin.html>

Cockroach Labs Documentation Team. (n.d.). *Trigram indexes*. CockroachDB Docs. <https://www.cockroachlabs.com/docs/stable/trigram-indexes#:~:text=A%20trigram%20index%20is%20a%20type%20of%20inverted,large%20tables%20without%20providing%20an%20exact%20search%20term>.

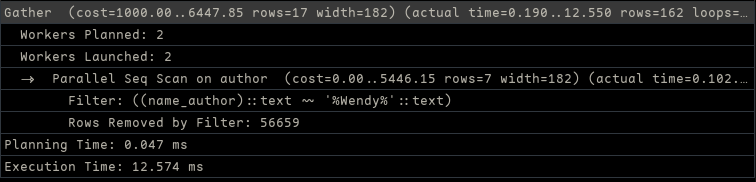
*CREATE INDEX*. (2024, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/sql-createindex.html>

*43.10. Trigger Functions*. (2024, February 8). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/plpgsql-trigger.html#:~:text=PL%2FpgSQL%20can%20be%20used%20to%20define%20trigger%20functions,change%20triggers%29%20or%20event_trigger%20%28for%20database%20event%20triggers%29>.

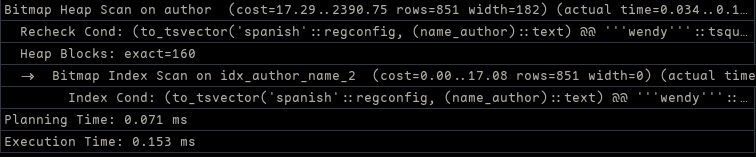
# Anexos

**

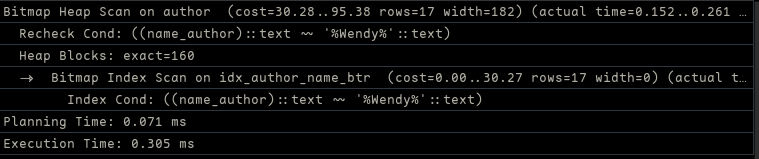
**Anexo 1**: Búsqueda de texto



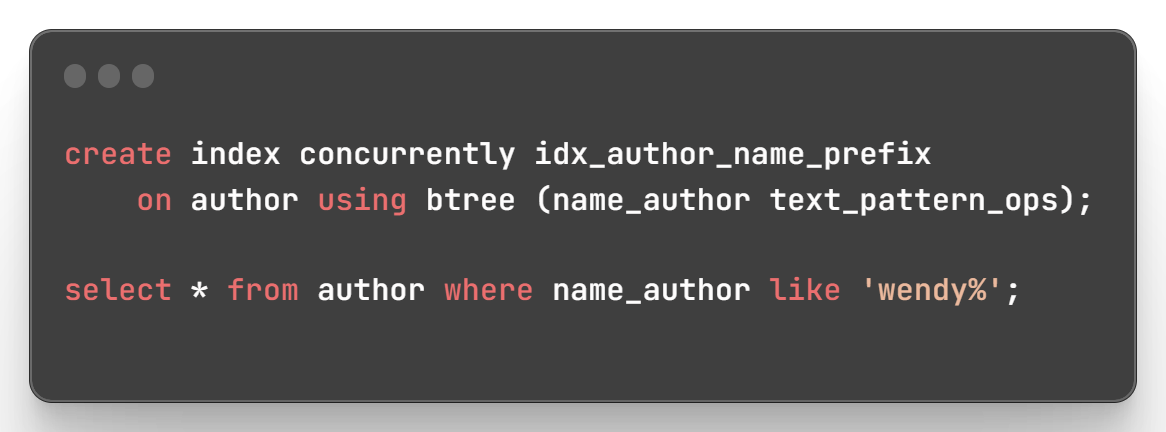
**Anexo 1.1**: explain analyze sin índices



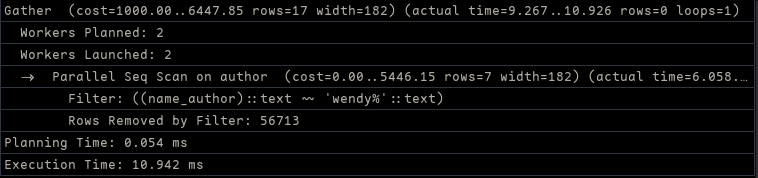
**Anexo 1.2**: explain analyze con índices método 2



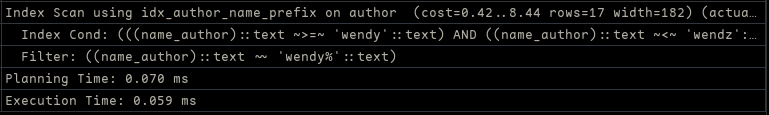
**Anexo 1.3**: explain analyze con índice método 3



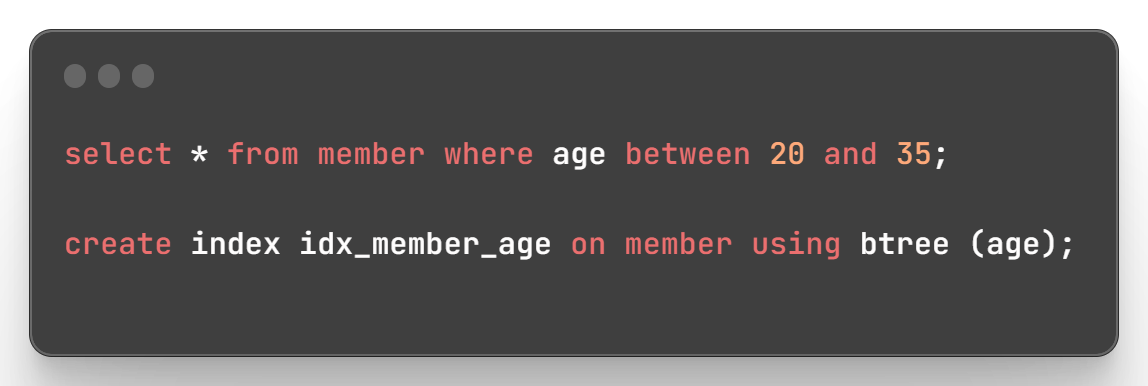
**Anexo 2**: Búsqueda de texto en la forma substring%

****

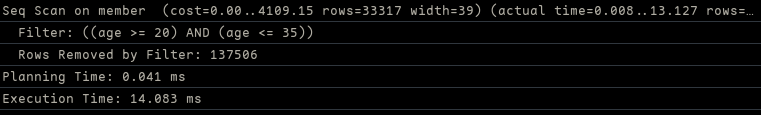
**Anexo 2.1**: Búsqueda de texto en la forma substring% sin índice

****

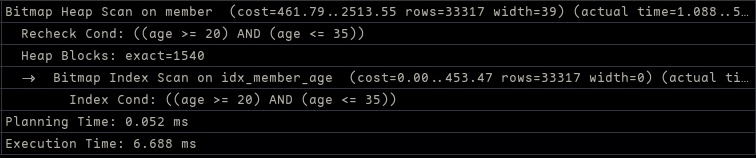
**Anexo 2.2**: Búsqueda de texto en la forma substring% con índice

****

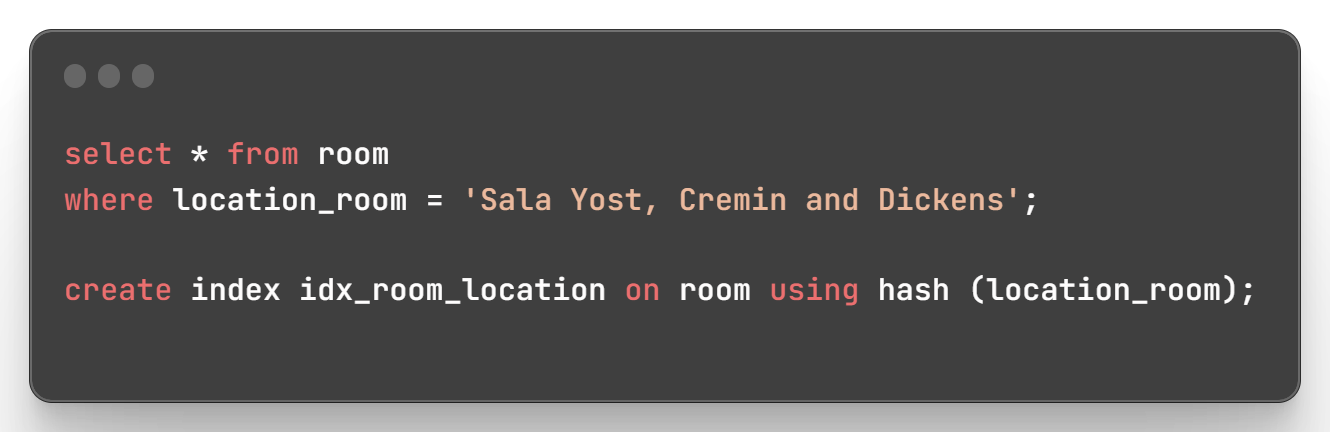
**Anexo 3**: Búsqueda de un intervalo de números

****

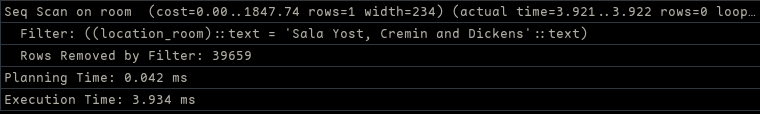
**Anexo 3.1**: Búsqueda de un intervalo de números sin índice



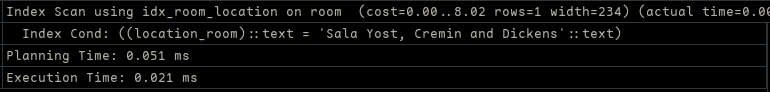
**Anexo 3.2**: Búsqueda de un intervalo de números con índice

****

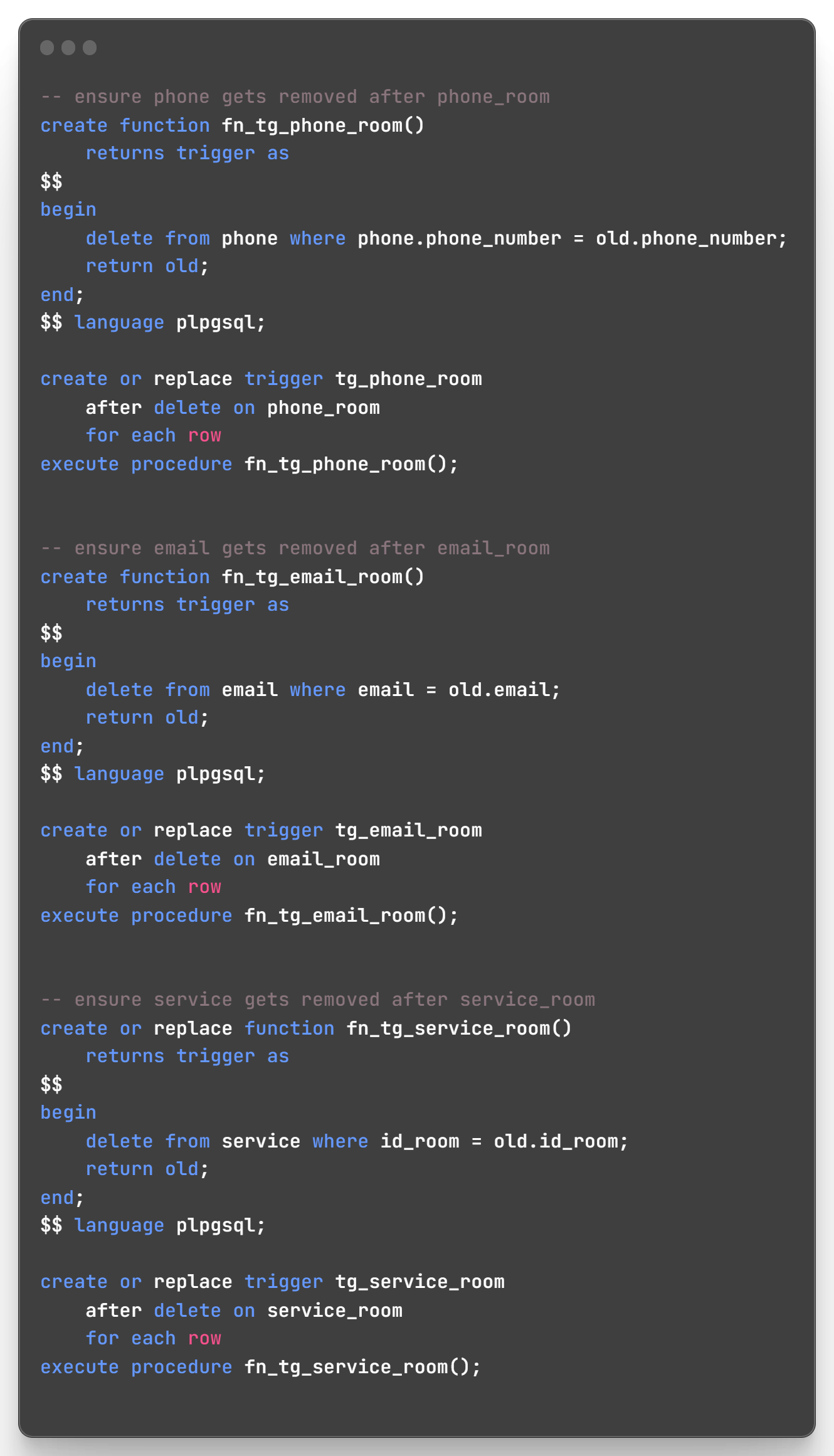
**Anexo 4**: Búsqueda de texto exacto

****

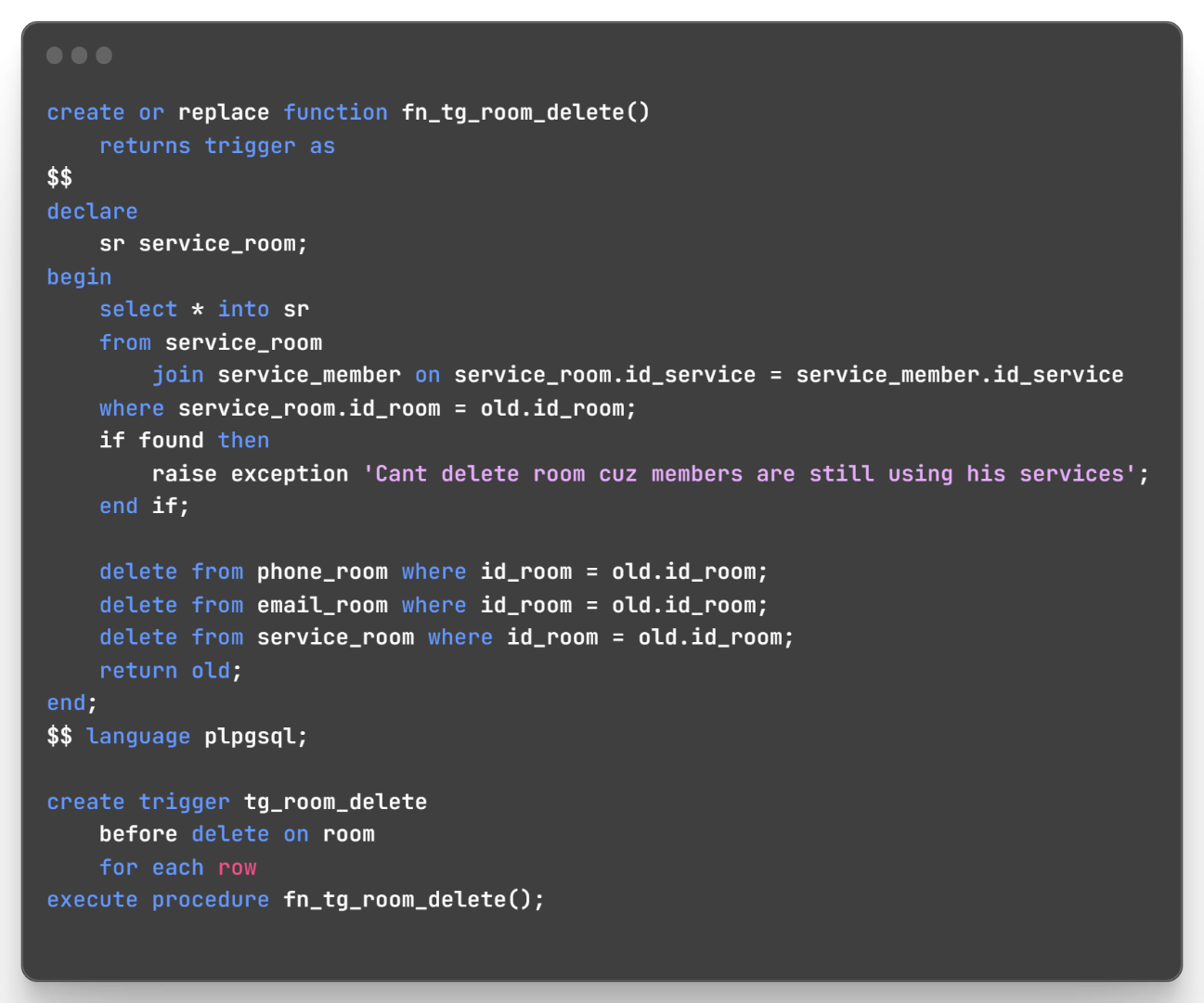
**Anexo 4.1**: Búsqueda de texto exacto sin índice



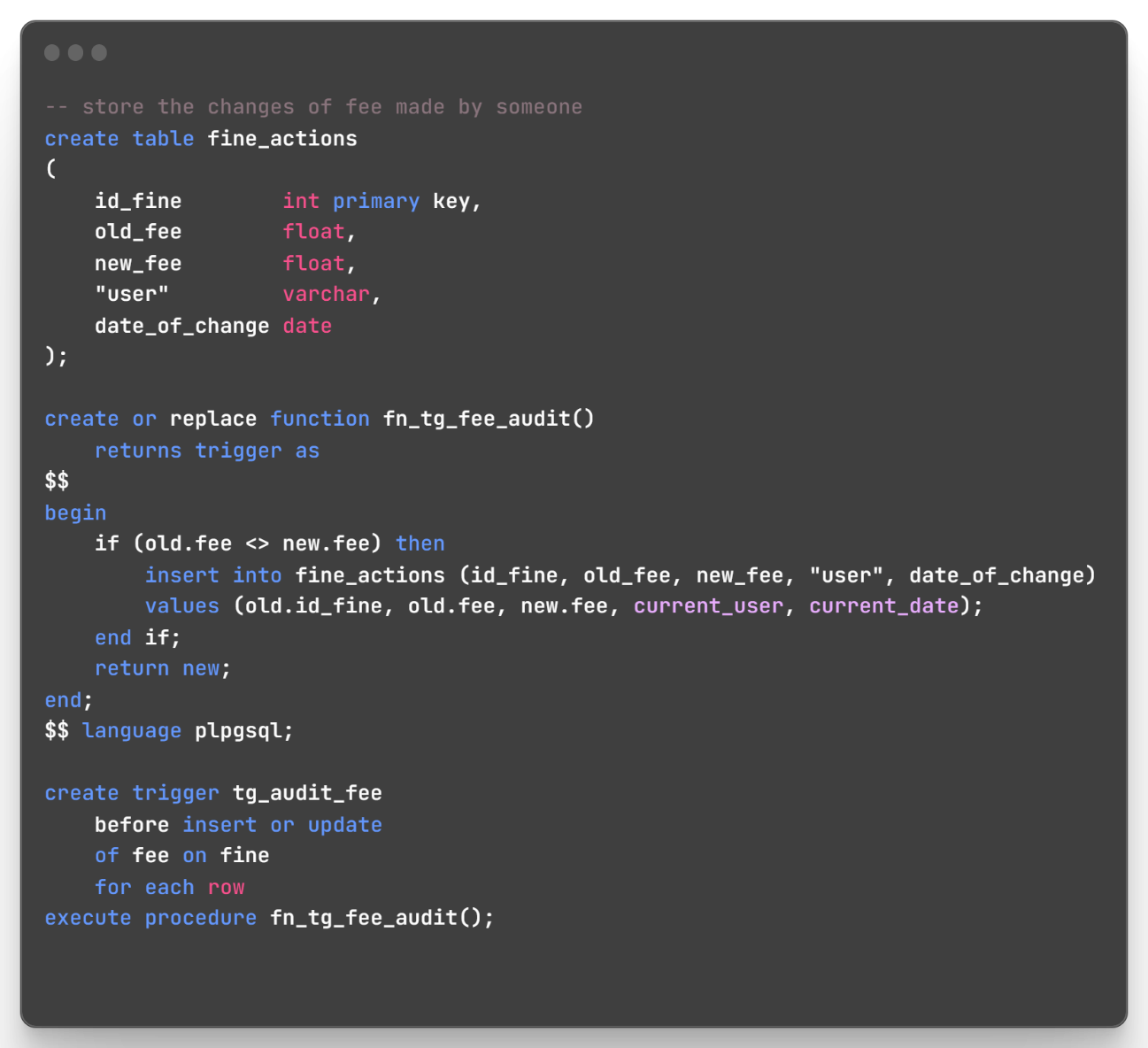
**Anexo 4.1**: Búsqueda de texto exacto con índice

****

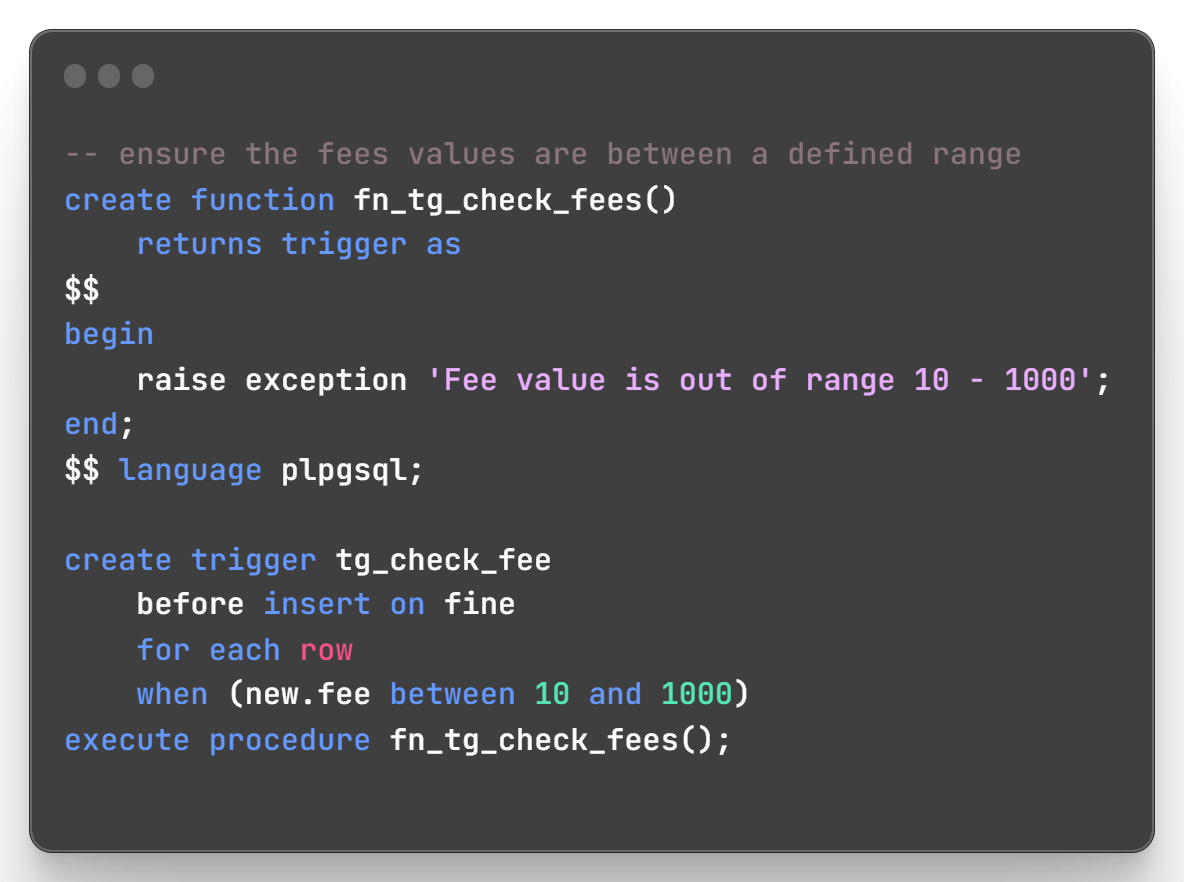
**Anexo 5**: Eliminar después

****

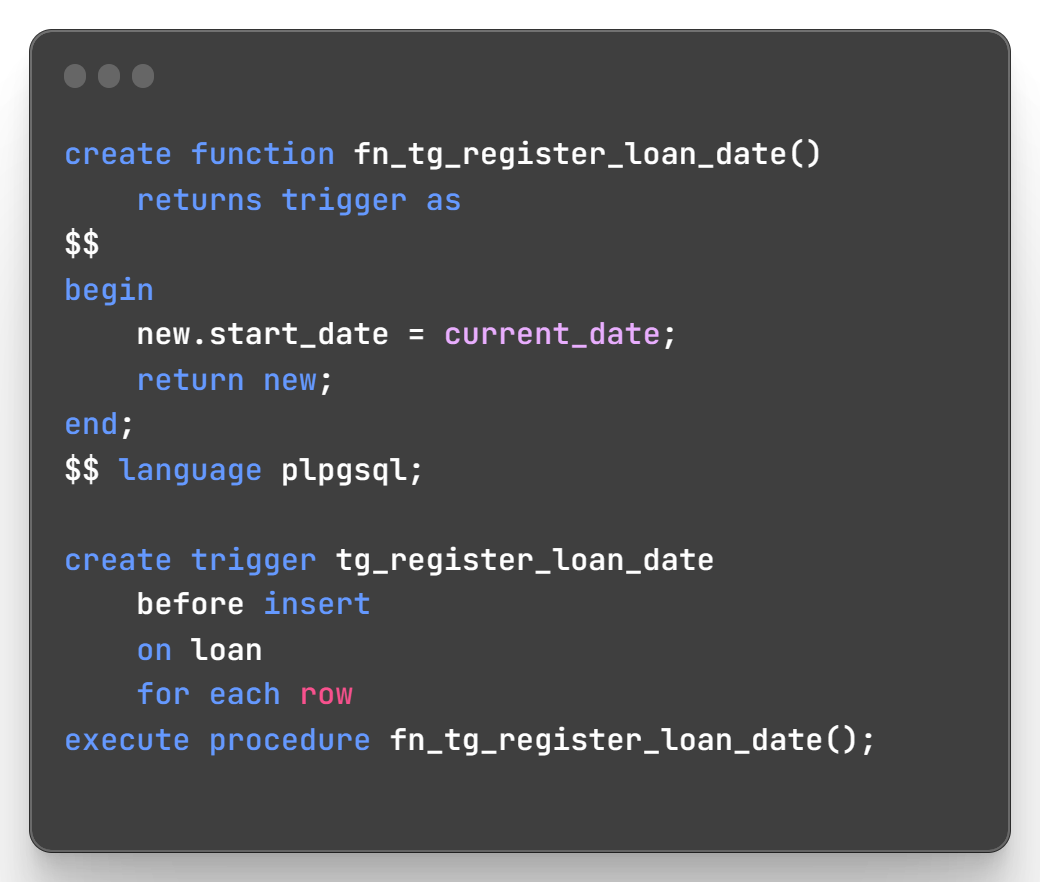
**Anexo 6**: Eliminar relacionadas

****

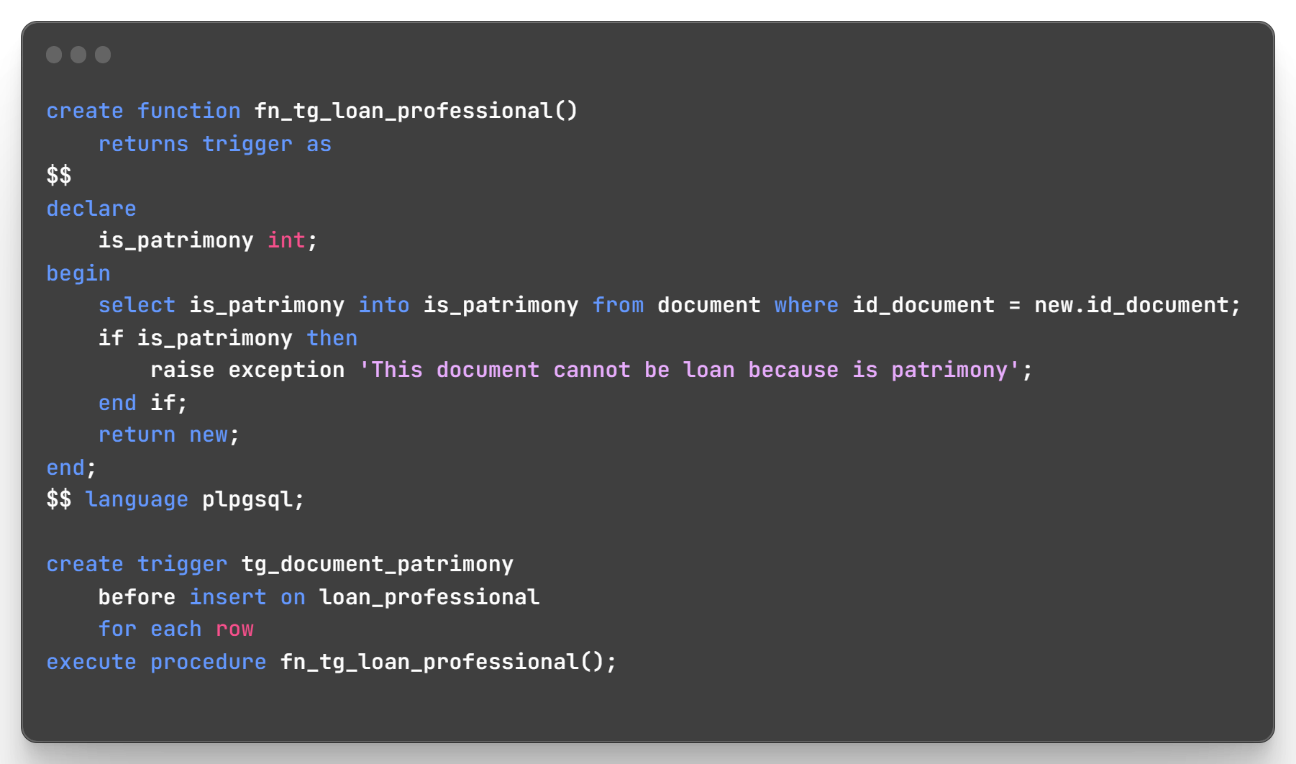
**Anexo 7**

****

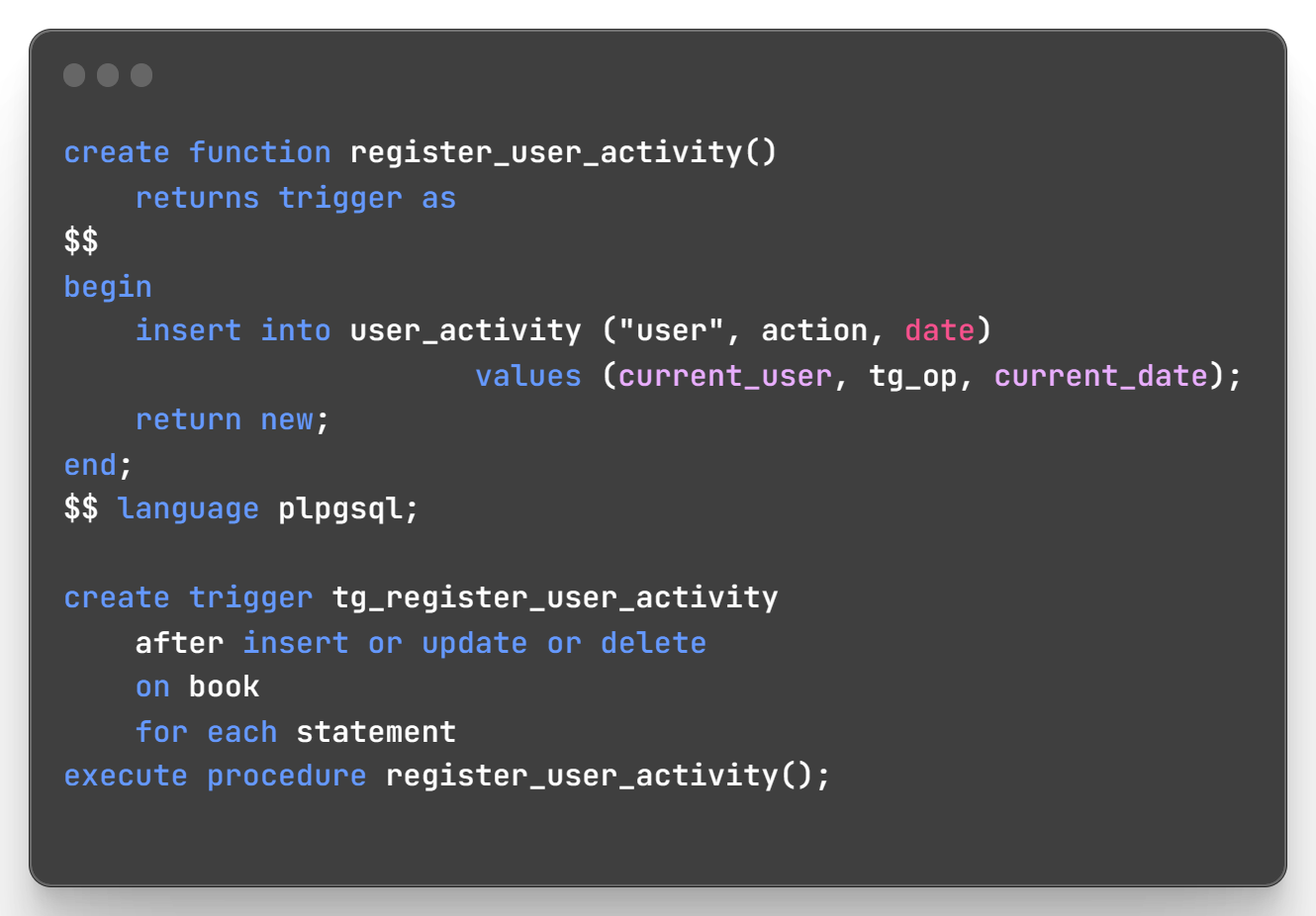
**Anexo 8**

****

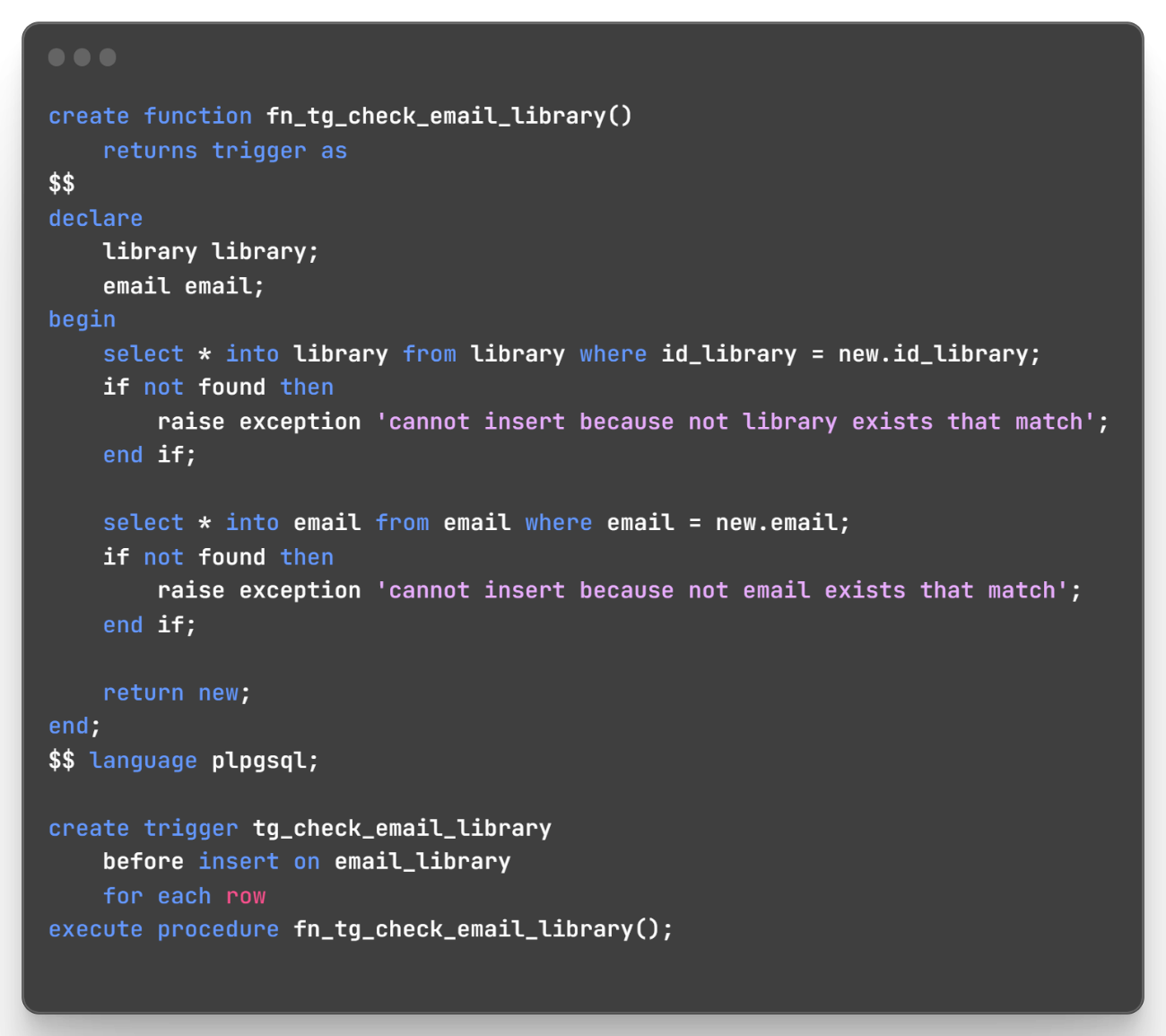
**Anexo 9**

****

**Anexo 10**

****

**Anexo 11**

****

**Anexo 12**