



Instituto Tecnológico De Costa Rica

Bases de Datos II

**I Proyecto Base de Datos**

Aaron González, Daniel Porras, y Anthony Jiménez

Leonardo Víquez

San Carlos

Abril

4 de abril de 2024

I Proyecto Base de Datos

Índice

Introducción	3
Análisis del problema	3
Solución del problema	4
Análisis de resultados	5
Conclusiones	6
Recomendaciones	6
Bibliografía	7

## Introducción

En el ámbito de las bases de datos, la auditoría juega un papel fundamental en la garantía de la integridad, seguridad y trazabilidad de la información. En este proyecto, se aborda la implementación de un sistema de auditoría en **PostgreSQL**, una de las bases de datos relacionales más utilizadas en la actualidad.

La necesidad de implementar un sistema de auditoría en una base de datos surge de la creciente importancia de la seguridad de la información y el cumplimiento de regulaciones y normativas en materia de protección de datos. La auditoría permite registrar y monitorizar las actividades realizadas en la base de datos, proporcionando un registro detallado de quién, cuándo y qué cambios se han realizado en los datos.

En este contexto, se ha desarrollado un sistema de auditoría en **PostgreSQL** que permite registrar las operaciones de inserción, actualización y eliminación en las tablas de la base de datos. Este proyecto se enmarca en las tendencias actuales de seguridad y trazabilidad de la información, donde la implementación de sistemas de auditoría se considera una práctica recomendada para garantizar la integridad y la confidencialidad de los datos.

## Análisis del problema

El problema identificado consiste en la necesidad de implementar un sistema de auditoría en una base de datos PostgreSQL. Este sistema debe ser capaz de registrar de manera automática y precisa las operaciones realizadas en las tablas de la base de datos, incluyendo inserciones, actualizaciones y eliminaciones, así como también información relevante como el usuario que realizó la operación, la dirección IP desde la que se realizó, la tabla afectada, el tipo de operación y los datos antes y después de la operación.

Para abordar este problema, es necesario considerar los siguientes aspectos:

### 1. Requerimientos Funcionales:

- El sistema debe ser capaz de registrar las operaciones de inserción, actualización y

eliminación en todas las tablas de la base de datos.

- Debe registrar la fecha y hora de la operación, el usuario que realizó la operación y la dirección IP desde la que se realizó.
- Debe almacenar los datos antes y después de la operación en formato JSONB para su posterior consulta.

## 2. Requerimientos No Funcionales:

- El sistema debe ser eficiente en términos de consumo de recursos y no afectar el rendimiento de la base de datos.
- Debe ser seguro y garantizar la integridad de los datos registrados.
- Debe ser escalable para adaptarse al crecimiento de la base de datos y la cantidad de operaciones registradas.

## Solución del problema

Para dar solución al problema planteado, se propone la implementación de un sistema de auditoría basado en triggers y funciones en PostgreSQL. Este sistema consta de los siguientes elementos principales:

1. Esquema y Tabla de Auditoría: Se crea un esquema llamado `auditoria` que contiene una tabla `registro_auditoria` para almacenar la información de auditoría.
2. Función `registrar_actividad`: Se define una función que registra una actividad en la tabla de auditoría, incluyendo información como el usuario, la dirección IP, la tabla afectada, el tipo de operación y los datos antes y después de la operación.
3. Trigger `trig_registrar_actividad`: Se crea un `trigger` que se ejecuta antes de una operación en una tabla y registra la actividad correspondiente en la tabla de auditoría.
4. Procedimientos `registrar_auditores` y `registrar_en_todo_lado`: Se crean procedimientos que facilitan la creación de `triggers` en una tabla específica o en todas las tablas de la base de datos, respectivamente.
5. Función `registrar_en_tablas_nuevas`: Se define una función llamada `registrar_en_tablas_nuevas`

la cual se encarga de registrar los **triggers** en toda tabla que se cree después de la creación de la extensión, esta se invoca desde un **event trigger** el cual corre en el evento **ddl\_command\_end** solamente cuando la etiqueta es **CREATE TABLE**.

La implementación de este sistema de auditoría proporciona una solución robusta y eficiente para el registro de actividades en una base de datos PostgreSQL, cumpliendo con los requerimientos funcionales y no funcionales identificados en el análisis del problema.

### Análisis de resultados

Requerimiento	Estado	Observaciones
Implementar sistema de auditoría en PostgreSQL	Concluido	Se implementó utilizando funciones y triggers en PostgreSQL según lo especificado en la documentación.
Registrar operaciones de inserción, actualización y eliminación	Concluido	Se logró registrar todas las operaciones correctamente.
Almacenar información de usuario, IP y datos afectados	Concluido	Se almacenó la información requerida en el formato adecuado.
Utilizar formato JSONB para almacenar datos antes y después	Concluido	Se utilizó el formato JSONB para almacenar los datos antes y después de las operaciones.

## Conclusiones

En este proyecto se logró implementar de manera satisfactoria un sistema de auditoría en PostgreSQL que cumple con los objetivos planteados. Se ha demostrado que es posible registrar de manera automática y precisa las operaciones realizadas en las tablas de la base de datos, garantizando la integridad y seguridad de la información.

Durante el desarrollo del proyecto, se pudo constatar que la utilización de funciones y triggers en PostgreSQL es una estrategia eficiente para la implementación de sistemas de auditoría en bases de datos, permitiendo registrar información relevante como el usuario que realizó la operación, la dirección IP desde la que se realizó y los datos antes y después de la operación.

## Recomendaciones

Para ampliar el alcance del proyecto y mejorar aún más el sistema de auditoría implementado, se sugiere considerar las siguientes recomendaciones:

1. Implementación de Alertas: Agregar funcionalidades para generar alertas automáticas en caso de detectar operaciones inusuales o sospechosas en la base de datos.
2. Mejora en la Visualización de los Datos: Desarrollar interfaces gráficas para facilitar la visualización y análisis de los datos registrados en la tabla de auditoría.
3. Automatización de Procesos de Mantenimiento: Implementar procesos automatizados para el mantenimiento y limpieza de la tabla de auditoría, evitando la acumulación de datos innecesarios.
4. Integración con Herramientas de Monitoreo: Integrar el sistema de auditoría con herramientas de monitoreo de bases de datos para obtener métricas y estadísticas sobre el rendimiento y la seguridad de la base de datos.

## Bibliografía

Manuel Kniep. 2015. «Writing Postgres Extensions - the Basics». 2015. <https://big-elephants.com/2015-10/writing-postgres-extensions-part-i/> .

The PostgreSQL Global Development Group. 2022a. «PostgreSQL Documentation». 2022. <https://www.postgresql.org/docs/>.

———. 2022b. «PostgreSQL Documentation». 2022. <https://www.postgresql.org/docs/current/sql-createextension.html>.