

# PostgreSQL – AutoAudit

## Manual de instalación y uso de auditoría automatizada de bases de datos

El siguiente manual tiene como objetivo explicar e ilustrar la instalación y utilización de la extensión para PostgreSQL llamada “AutoAudit”, la cual tiene como funcionamiento el registro de operaciones de inserción, modificación y eliminación de filas en todas las tablas de la base de datos en la que se instale.

Cabe destacar que el proceso aquí mostrado fue realizado en una computadora con Windows.

Este registro posee Identificador único del evento, Tipo de operación (INSERT, UPDATE, DELETE), Nombre de la tabla afectada, Fecha y hora exacta del evento, Usuario/rol ejecutor de la operación, Dirección IP del cliente conectado, Estado previo de los datos (antes de la modificación) y Estado posterior de los datos (después de la modificación)



Ejemplo visual del registro:

|   | event_id<br>[PK] bigint | operation_type<br>text | table_name<br>text | event_time<br>timestamp with time zone | executed_by<br>text | client_ip<br>inet | old_data<br>jsonb   | new_data<br>jsonb   |
|---|-------------------------|------------------------|--------------------|--|---------------------|-------------------|---|---|
| 1 | 1                       | INSERT                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | :::1              | [null]  | {}  |
| 2 | 2                       | UPDATE                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | :::1              | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 499.99, 'category': 'cellphone', 'product_id': 30} | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 459.99, 'category': 'cellphone', 'product_id': 30} |
| 3 | 4                       | INSERT                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | :::1              | [null]  | {}  |
| 4 | 5                       | UPDATE                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | :::1              | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 499.99, 'category': 'cellphone', 'product_id': 31} | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 459.99, 'category': 'cellphone', 'product_id': 31} |
| 5 | 6                       | DELETE                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | :::1              | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 459.99, 'category': 'cellphone', 'product_id': 31} | [null]  |

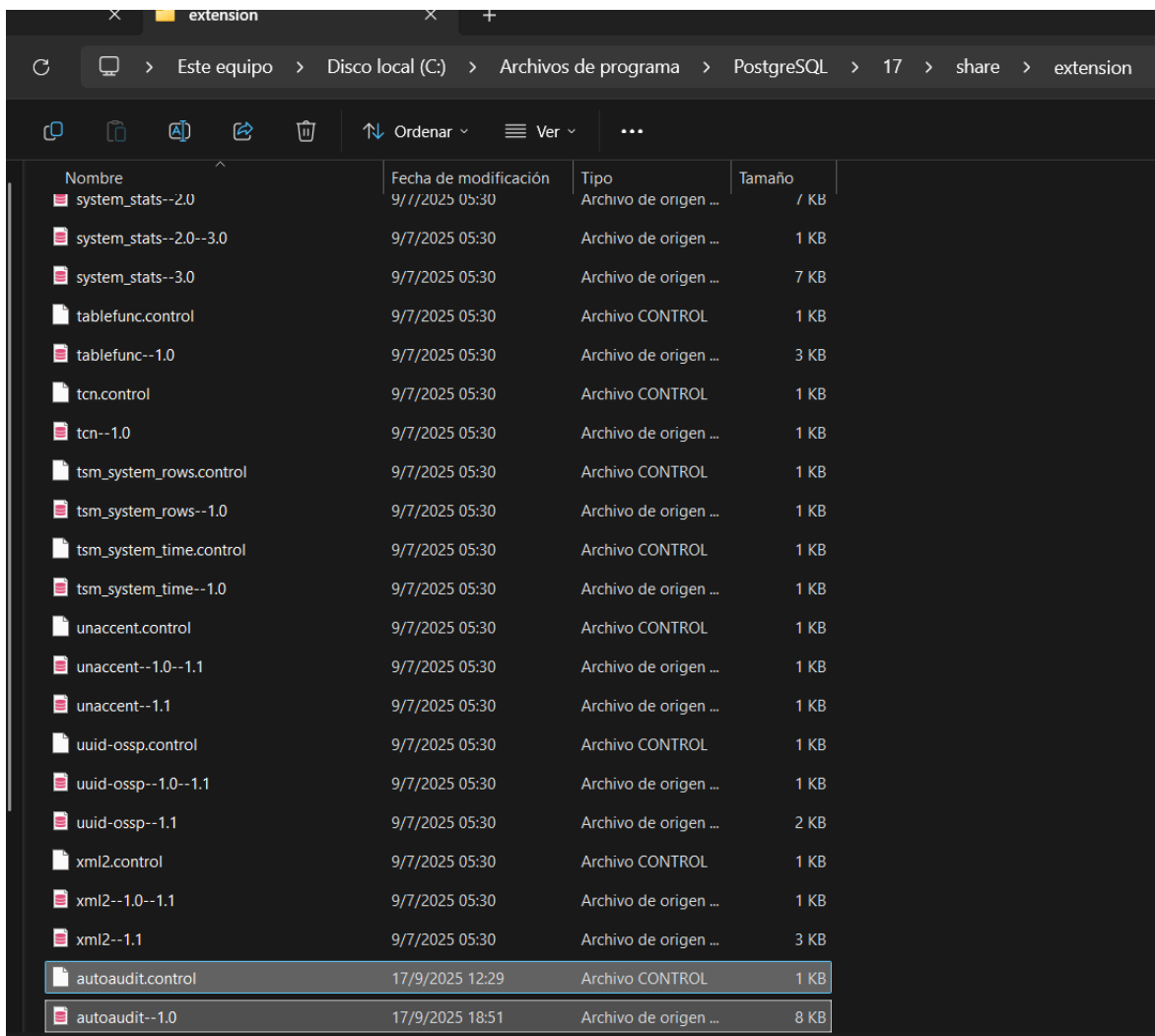
## Instalación

Se requiere tener los dos siguientes archivos (están en la carpeta autoaudit):

- autoaudit.control
- autoaudit--1.0

| Nombre  | Fecha de modificación | Tipo                  | Tamaño |
|---|-----------------------|-----------------------|--------|
|  autoaudit.control | 17/9/2025 12:29       | Archivo CONTROL       | 1 KB   |
|  autoaudit--1.0    | 17/9/2025 18:52       | Archivo de origen ... | 8 KB   |

Una vez descargados los copiamos y vamos a la ruta donde se almacenan las extensiones de PostgreSQL, que por defecto suele ser: C:\Program Files\PostgreSQL\17\share\extension

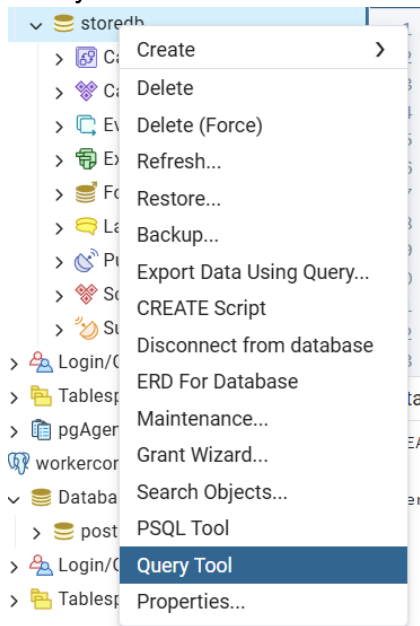


Nota: esta ruta puede variar si la cambiamos en el momento en el que instalamos PostgreSQL.

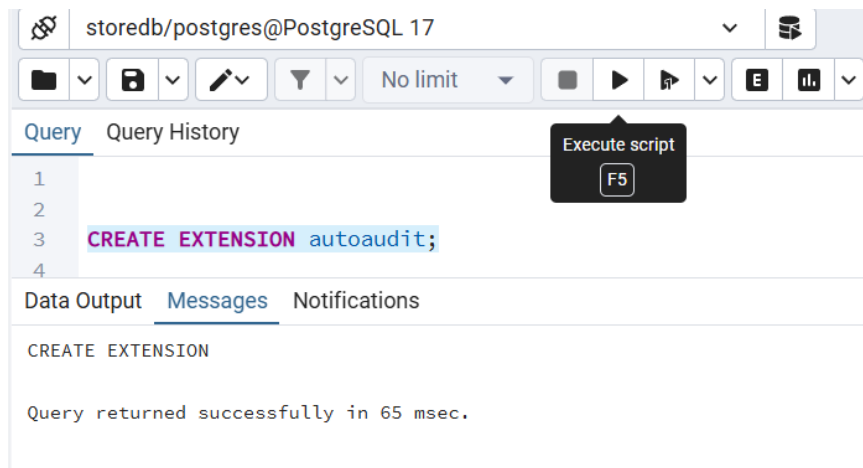
En esta ruta debemos pegar los archivos de autoaudit mencionados anteriormente.

Cabe destacar que a partir de aquí el proceso se debe repetir para cada base de datos con la que queramos usar la extensión, pero es un proceso bastante sencillo.

Una vez que estén almacenados en dicho directorio, desde PostgreSQL iniciados con el usuario postgres con los privilegios superiores de administración (que es el perfil de login que deseamos pueda acceder a los registros y ante todo caso el único que podrá borrarlos) y vamos a ir a la base de datos donde queremos utilizarla y vamos a abrir un Query Tool en la misma:



Dentro de este vamos a ejecutar el siguiente código: **CREATE EXTENSION autoaudit;**

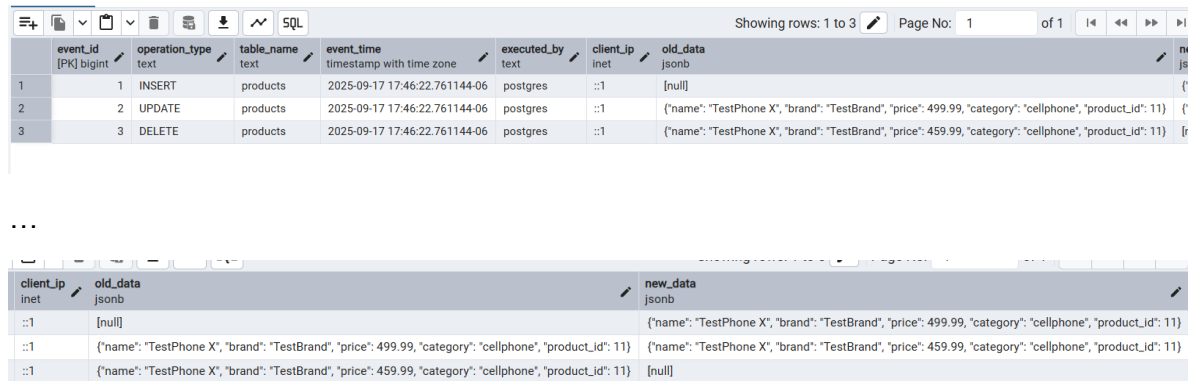


Con solo hacer esto ya quedará funcionando, a partir de ahora cada vez que un usuario inserte, elimine o modifique filas de la base de datos se añadirá al registro.

## USO

Para consultarlo solo es necesario ejecutar el siguiente comando, siempre desde el usuario postgres con privilegios de administrador donde instalamos la extensión:

**select \* from autoaudit.audit\_log**

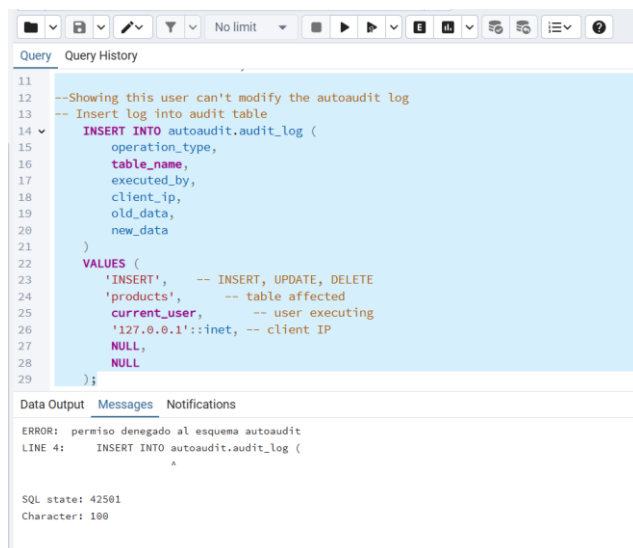


| event_id<br>[PK] bigint | operation_type<br>text | table_name<br>text | event_time<br>timestamp with time zone | executed_by<br>text           | client_ip<br>inet | old_data<br>jsonb | new_data<br>jsonb   |
|-------------------------|------------------------|--------------------|--|-------------------------------|-------------------|-------------------|---|
| 1                       | 1                      | INSERT             | products                               | 2025-09-17 17:46:22.761144-06 | postgres          | :::1              | [null]  |
| 2                       | 2                      | UPDATE             | products                               | 2025-09-17 17:46:22.761144-06 | postgres          | :::1              | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 499.99, 'category': 'cellphone', 'product_id': 11} |
| 3                       | 3                      | DELETE             | products                               | 2025-09-17 17:46:22.761144-06 | postgres          | :::1              | {'name': 'TestPhone X', 'brand': 'TestBrand', 'price': 459.99, 'category': 'cellphone', 'product_id': 11} |

Esta tabla esta en un esquema de la base de datos llamado autoaudit, el nombre de la tabla es audit\_log, solo el rol postgre con privilegios que la creó puede modificar/eliminar dicho esquema y tabla.

Naturalmente, los usuarios que vaya a trabajar con la base de datos tendrán sus propios permisos con restricciones para el uso de la base, ese manual no comprende dichos permisos pues son ajenos al tema.

Lo que sí se comprende es una muestra de que un usuario sin permisos de modificar autoaudit no puede hacer cambios manuales en la tabla (para estos ejemplos se creó un usuario Worker al que solo se le garantizaron permisos para modificar las tablas del esquema public):



```
11
12 --Showing this user can't modify the autoaudit log
13 -- Insert log into audit table
14 INSERT INTO autoaudit.audit_log (
15     operation_type,
16     table_name,
17     executed_by,
18     client_ip,
19     old_data,
20     new_data
21 )
22 VALUES (
23     'INSERT', -- INSERT, UPDATE, DELETE
24     'products', -- table affected
25     current_user, -- user executing
26     '127.0.0.1::inet', -- client IP
27     NULL,
28     NULL
29 );
```

ERROR: permiso denegado al esquema autoaudit  
LINE 4: INSERT INTO autoaudit.audit\_log (  
A

SQL state: 42501  
Character: 100

Por otro lado, se evidencia el funcionamiento cuando dichos roles/usuarios hacen los cambios monitoreados en las tablas base de datos:

+

📄

▼

📋

▼

🗑️

🔄

📥

📈

SQL

Showing

|   | event_id<br>[PK] bigint | operation_type<br>text | table_name<br>text | event_time<br>timestamp with time zone | executed_by<br>text | client_ip<br>inet | old_data<br>jsonb       |
|---|-------------------------|------------------------|--------------------|--|---------------------|-------------------|-------------------------|
| 1 | 1                       | INSERT                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | ::1               | [null]                  |
| 2 | 2                       | UPDATE                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | ::1               | {'name': 'TestPhone X', |
| 3 | 3                       | DELETE                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | ::1               | {'name': 'TestPhone X', |
| 4 | 4                       | INSERT                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | ::1               | [null]                  |
| 5 | 5                       | UPDATE                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | ::1               | {'name': 'TestPhone X', |
| 6 | 6                       | DELETE                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | ::1               | {'name': 'TestPhone X', |

Finalmente podemos ver como el usuario postgres, que es el que tiene los privilegios, sí puede eliminar o modificar datos del registro, mientras que como ya vimos otros usuarios sin permisos no:

```
57 --this user can delete things from the audit, other users can't
58 delete from autoaudit.audit_log where event_id = 3
59 select * from autoaudit.audit_log
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 125 msec.

|   | event_id<br>[PK] bigint | operation_type<br>text | table_name<br>text | event_time<br>timestamp with time zone | executed_by<br>text | client_ip<br>inet |
|---|-------------------------|------------------------|--------------------|--|---------------------|-------------------|
| 1 | 1                       | INSERT                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | ::1               |
| 2 | 2                       | UPDATE                 | products           | 2025-09-17 18:51:24.216113-06          | postgres            | ::1               |
| 3 | 4                       | INSERT                 | products           | 2025-09-17 18:51:30.655282-06          | Worker              | ::1               |

Como podemos ver el registro 3 ha desaparecido.

Al hacer esto aseguramos que otros usuarios no puedan modificar este importante registro.

Terminamos aquí con este manual de uso, ¡ya tienes todo lo que se necesita para instalar esta extensión y visualizar los registros de esta auditoría automatizada!