

Base de Datos

Actividad: Creación de Triggers de Auditoría en BD clubangloviejo

Alumno: Quesada Santina

Curso: 7mo

Año: 2024

Profesor: González Luis

Resolución, explicación y comprensión de la actividad:

1. Introducción

Para esta actividad me plantee varias preguntas y me surgieron demasiados problemas al intentar resolverlas, pero como en la gran mayoría de las ocasiones, la perseverancia y la satisfacción de lograrlo, al final vencen por completo al sentimiento de rendirse.

A lo largo de todo este proceso, comprendí principalmente para qué, cómo y en que momentos son necesarios utilizar los Triggers.

Y además de analizar el procedimiento, agregué capturas de pantalla con claras evidencias de los resultados obtenidos en cada paso, que seguro le serán de ayuda para una mayor comprensión del tema al lector.

2. Análisis de las tablas "usuarios" y "socios":

Antes de realizar las tablas de auditoria necesarias, analicé las tablas ya existentes de "usuarios" y "socios", sobre las cuales debería basarme luego. Entendí como era el manejo de claves primarias y foráneas y memoricé los campos que iba a utilizar en mis tablas de auditoría. Además, probando y revisando las mismas me di cuenta que algunos valores de los campos no eran necesarios al momento de crear dichas tablas. Datos como NOT NULL y COLLATE 'latin1_swedish_ci' se agregaban automáticamente luego de la creación de la tabla, por lo cual, no incluirlos al principio, puede facilitar y mejorar el tiempo en que realizamos dichas tablas.

3. Comprensión de la utilidad de las tablas de auditoria

Entender para que se realizan estos procedimientos, así como cualquier otro, es clave para desenvolvernos eficazmente en nuestro día a día como programadores. Considero que antes de realizar las tareas y modificaciones es necesario tratar de analizar desde nuestro punto de vista lo que deberíamos hacer.

Dicho todo esto, luego del correspondiente análisis, pude entender que principalmente las tablas de auditoria son de ayuda para realizar respaldos y corroboraciones de información específica. Además, al probar crearlas en el proceso de verificación, caí en cuenta que lógicamente estas tablas si las realizamos correctamente, pueden mantener como un historial bastante detallado de las operaciones (inserciones, actualizaciones y eliminaciones) que se realicen en las tablas principales de nuestra base de datos. Si utilizo de ejemplo a un banco; que busca cumplir con la seguridad, trasparencia, recuperación y cumplimiento de cada dato e información de sus clientes, entonces podríamos decir que son de suma importancia estas tablas de auditoría en el día a día del banco.

4. Creación de las tablas de "auditoria_usuarios" y "auditoria_socios"

Para poder crear las tablas correctamente debí revisar varias veces los datos de las tablas principales (usuarios y socios) de las que iba a tomar los campos que necesitaba.

En la tabla "auditoria_usuarios" necesitaba los datos de: dirección, telefono y email. Mientras que en la tabla "auditoria_socios" necesitaba los datos de: flnscripcion, fRenovacion, fBaja, emmac y el activo (estado) del usuario. Aunque como se logra apreciar en las imágenes, también incluí dos campos más: operación (que especificará luego la operación que realicemos en la tabla, si es de inserción, actualización u otra) y fecha (que nos dirá en qué momento preciso se han realizado y registrado los cambios que hagamos u haga otra persona en nuestras tablas) Además, al principio de la consulta como extra, agregué una sentencia para eliminar las tablas en caso de que existan y prevenir posibles errores.

```
🥊 gestion-imdf 📗 Base de datos: squesada236 📘 Tabla: auditoria_socios 🔠 Datos 🕨 gestion-imdf 📗 Base de datos: squesada236 📗 Tabla: auditoria_socios 🖽 Datos 🕨 tabla_audit
  1 DROP TABLE IF EXISTS `auditoria usuarios`;
                                                             1 DROP TABLE IF EXISTS `auditoria socios`;
                                                              3 CREATE TABLE `auditoria_socios` (
  3 CREATE TABLE `auditoria usuarios` (
                                                                    `operacion` VARCHAR (80),
`fInscripcion_anterior` DATE,
        `operacion` VARCHAR(80),
`direccion_anterior` VARCHAR(80),
  4
  5
                                                                      `fInscripcion_nueva` DATE,
  6
          `direccion nueva` VARCHAR(80),
                                                                     `fRenovacion_anterior` DATE,
         `telefono_anterior` VARCHAR(80),
  7
                                                                      `fRenovacion_nueva` DATE,
         `telefono_nuevo` VARCHAR(80),
`email_anterior` VARCHAR(50),
                                                              9
10
                                                                      `fBaja_anterior` DATE,
  9
                                                                      `fBaja_nueva` DATE,
           `email_nuevo` VARCHAR(50),
  10
                                                              11
                                                                      `emmac_anterior` DATE,
         fecha TIMESTAMP
  11
                                                              12
                                                                       `emmac_nueva` DATE,
 12 );
                                                                      `activo_anterior` INT(11) NOT NULL DEFAULT '1',
                                                              13
                                                                      `activo_nuevo` INT(11) NOT NULL DEFAULT '1',
                                                                      `fecha` TIMESTAMP
                                                              15
                                                              16 );
```

5. Comprensión de la funcionalidad de los Triggers de Inserción (INSERT)
Al finalizar correctamente con la creación de las tablas de auditoría, debía comenzar a implementar los Triggers... pero me surgieron varias dudas y antes que realizar algo mal y frustrarme, busque entender cual era la principal funcionalidad de un Trigger y en que momento debía utilizarlo.

Para mí un **Trigger** es un **mecanismo** que permite realizar diferentes **operaciones** (insert, update, delete) de forma **automática** en nuestra base de datos o tabla específica, además los Triggers se diferencian principalmente y a mi entender porque solo **responden** a un **evento** específico que suceda en la tabla que esté asociada al mismo.

Por ejemplo, Retomemos el caso del **banco** e imaginamos que hay una tabla de **cuentas**, en la que queremos **insertar** automáticamente un registro en la tabla de **auditoría** (con un id, saldo anterior, saldo nuevo, fecha del cambio, etc.), esto se hará cada vez que se actualice el saldo de la cuenta. Entonces lo que necesitamos es un **Trigger** de **actualización** que se **active** cada vez que se actualiza el saldo en la tabla de cuentas.

6. Creación y lógica de implementación de los Triggers de usuarios y socios Como pasos fundamentales para una correcta implementación de los Triggers, se debe verificar y pensar que tipo de operación queremos realizar y en que momento debemos realizarla. Esto es primordial, ya que es el principal objetivo por lo cual lo realizamos. En el caso de los Triggers de usuarios y socios, en ambos se realizo un trigger AFTER INSERT y uno BEFORE UPDATE.

Luego debemos declarar los datos que queremos insertar como respaldo en nuestra tabla de auditoría, acá es importante revisar que los escribamos bien y en orden como los declaramos en nuestra tabla de auditoría, sino ocasionaremos un error.

Por último, debemos especificar si el dato va ser nuevo o es anterior y deseamos conservarlo, es crucial en operaciones de actualización y eliminación ya que distingue cada información. Básicamente se declara con NEW. Los datos que deseemos declarar como nuevos y OLD. los datos que deseemos declarar como ya ingresados y a conservar.

Triggers de usuarios:

Triggers de socios:

7. Verificación de la funcionalidad de los Triggers:

Por último, debemos corroborar que nuestros Triggers realicen sus tareas correspondientes, por lo cual, me dirigí a una nueva consulta e ingresé nuevos usuarios y socios para demostrar que todo ha salido correctamente.

Tabla de usuarios antes de insertar un usuario nuevo:



Tabla de **auditoria**_usuarios **antes** de insertar un nuevo usuario en la tabla usuario:

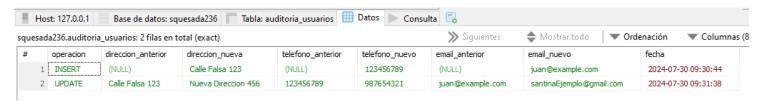


Tabla de usuarios después de agregar un nuevo usuario:



Tabla de **auditoria**_usuarios **después** de insertar un nuevo usuario en la tabla de usuarios:

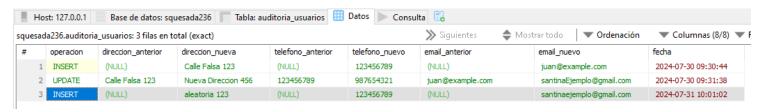


Tabla de usuarios con un usuario modificado:



Tabla de **auditoria**_usuarios **después** de **modificar** un usuario en la tabla de usuarios:



Como se ve en las anteriores capturas de tablas de usuarios al realizar inserción de usuarios y modificación de los existentes aparece en la tabla de auditoria_usuarios un reporte de lo que hemos realizado, lo que significa que los Trigger realizados para usuarios funcionan correctamente.

Tabla de socios antes de insertar un nuevo socio:

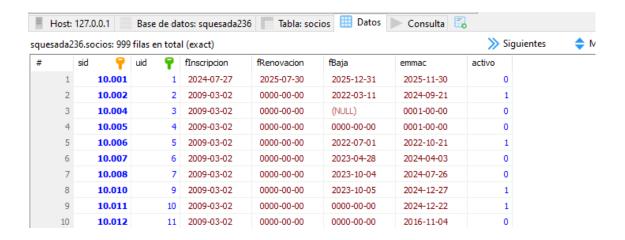


Tabla de **auditoria**_socios **antes** de **insertar** un nuevo socio en la tabla de socios:

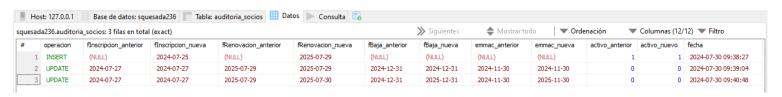


Tabla de socios después de insertar un nuevo socio:



Tabla de **auditoria**_socios **después** de **insertar** un nuevo socio en la tabla de socios:



Tabla de socios con un socio modificado:



Tabla de **auditoria**_socios **después** de haber **modificado** un socio en la tabla de socios:



Como se ve en las anteriores capturas de tablas de socios al realizar inserción de socios y modificación de los existentes aparece en la tabla de auditoria_socios un reporte de lo que hemos realizado, lo que significa que los Trigger realizados para socios funcionan correctamente.