**Trilha de Auditoria (Log) com utilização de Triggers**

1. **Introdução e contexto**

Em sistemas de grandes corporações, muitas vezes a modificação ou exclusão de registros em um banco de dados de forma indevida pode acarretar problemas sérios em relação aos dados.

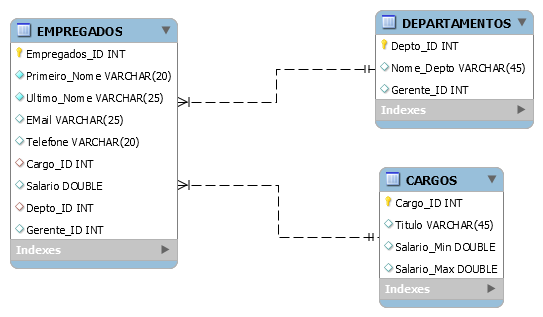
Apesar de existirem meios de recuperar as informações, identificar quando e como aconteceram os problemas pode facilitar muito o trabalho de identificação de responsáveis e escolha das ações a serem tomadas para recuperar os dados.

Em um banco de dados, uma das formas mais comuns de manter este controle é manter, em uma tabela à parte, todo o registro de operações que envolvem inserção, alteração ou exclusão de dados. Ou seja, cada vez um registro é modificado em uma tabela, grava-se nesta tabela separada a data, a hora da operação, quem fez a operação e quais os dados alterados.

Um caminho para implementar este recurso de forma automática seria através do uso de Triggers de banco, que a cada vez que uma operação seja executada em uma tabela, o registro seja armazenado na tabela de auditoria.

1. **Objetivo do Exercício**

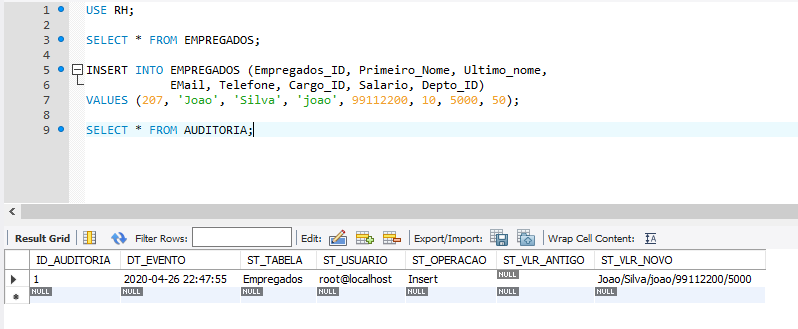
Utilizando o banco de dados RH, criar uma trilha de auditoria para a tabela “Empregados”, possibilitando rastrear todas as operações de inclusão, alteração e exclusão da tabela.



As informações a serem armazenadas são:

* Data
* Hora
* Usuário do banco
* Operação (I - Inclusão, A - Alteração ou E - Exclusão)
* Nome da Tabela
* Valores anteriormente contidos na tabela
* Valores novos modificados na tabela

Por exemplo, após criada a trigger, ao inserir um registro na tabela Empregados, a tabela Auditoria ficaria da seguinte forma;



1. **Detalhamento das Atividades a serem executadas**
   1. **Criar a tabela para armazenar as informações**

Criar, no schema RH, a tabela de auditoria conforme a especificação abaixo:

**TABELA:** AUDITORIA

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| *ID\_AUDITORIA* | *INT AUTO\_INCREMENT* |
| DT\_EVENTO | DATETIME |
| ST\_USUARIO | VARCHAR(60) |
| ST\_TABELA | VARCHAR(60) |
| ST\_OPERACAO | VARCHAR(10) |
| ST\_VLR\_ANTIGO | VARCHAR(1000) |
| ST\_VLR\_NOVO | VARCHAR(1000) |

* 1. **Criar uma Trigger para armazenar os dados inseridos na tabela**
* A trigger será disparada após a operação de Insert na tabela
* Criar uma variável para receber os valores que foram inseridos
* Concatenar os valores de inserção e jogar na variável criada, separados por barras. Ex.: “200/JOAO DA SILVA/joao@gmail.com/6199221002”
* Capturar as informações adicionais (Data e Hora atual, Usuário Corrente)
* O nome da tabela (EMPREGADOS) e a operação (INSERT) podem ser fixas dentro dessa trigger.
* Inserir o registro na tabela auditoria (campo ST\_VLR\_ANTIGO ficará vazio)
  1. **Criar uma Trigger para armazenar os dados alterados na tabela**
* A trigger será disparada após a operação de Update na tabela
* Criar duas variáveis, uma para receber os valores que foram alterados e outra para receber os valores que existiam anteriormente
* Concatenar os valores de alterados e jogar na variável criada, separados por barras. Ex.: “200/JOAO DA SILVA/**joao@gmail.com**/6199221002”
* Concatenar os valores antes de terem sido alterados e jogar na variável correspondente criada, separados por barras. Ex.: “200/JOAO DA SILVA/**joao@hotmail.com**/6199221002”
* Capturar as informações adicionais (Data e Hora atual, Usuário Corrente)
* O nome da tabela (EMPREGADOS) e a operação (UPDATE) podem ser fixas dentro dessa trigger.
* Inserir o registro na tabela auditoria
  1. **Criar uma Trigger para armazenar os dados excluídos da tabela**
* A trigger será disparada após a operação de Delete na tabela
* Criar uma variável para receber os valores que foram excluídos
* Concatenar os valores excluídos e jogar na variável criada, separados por barras. Ex.: “200/JOAO DA SILVA/**joao@gmail.com**/6199221002”
* Capturar as informações adicionais (Data e Hora atual, Usuário Corrente)
* O nome da tabela (EMPREGADOS) e a operação (DELETE) podem ser fixas dentro dessa trigger.
* Inserir o registro na tabela auditoria
  1. **Criar um novo usuário no banco de dados para testar a trigger**
* Crie um usuário chamado “operador” com a senha “operador” com acesso de SELECT, INSERT, UPDATE e DELETE na tabela de EMPREGADOS e AUDITORIA
* Insira 2 empregados na tabela de empregados
* Altere o salário de 1 dos empregados
* Exclua o outro empregado que não foi alterado
* Visualize a tabela de Auditoria

1. **Referência e comandos PL/MySQL**
2. Declaração de variáveis e manipulação de valores

DECLARE <variável> <tipo>

SET <variável> = <valor>

1. Concatenar valores

CONCAT(Valor1, Valor2, Valor3, ...)

Ex.: CONCAT ( NEW.Primeiro\_Nome, ‘ / ‘, NEW.Ultimo\_Nome)

Resulta em: “Joao/Silva”

1. Data e hora atual do servidor

NOW()

1. Usuário Atual

SELECT CURRENT\_USER()

1. **Formato da Entrega**

Gerar um arquivo .SQL no MySQL Workbench com as respostas do exercícios, separados por comentários, como no exemplo abaixo. O nome do arquivo deverá ser o primeiro e ultimo nome do aluno.

Ex.:

**Arquivo: HEDSON\_LIMA.SQL**

**-- Exercicio Aluno: Hedson Rodrigues Lima**

**-- Questão 3.1**

**USE RH;**

**CREATE…..**

**-- Questão 3.2**

**DELIMITER…**

**..**

**-- Questão 3.3**

**CREATE USER ….**