Universidade de São Paulo - São Carlos
Instituto de Ciências Matemáticas e de Computação

SCC-541 Laboratório de Bases de Dados

Trabalho Pratico T6 – Exercícios sobre Gatilhos (*Triggers*)

Dr. Caetano Traina Jr., Igor Alberte R. Eleutério
1'semestre de 2023

Data para entrega: 31 de maio

Datasets

Base de dados da Fórmula 1 - FIA.

A base de dados a ser utilizada é a mesma preparada no **Trabalho 1**. Caso necessário, os arquivos originais para a carga de dados encontra-se no Google Drive, e podem ser acessados no endereço este link. Para acessar os arquivos, você deve estar logado no Google Drive com sua conta USP. Cada arquivo corresponde a uma tabela, que deve ser carregada na base de dados.

Os dados originais foram ligeiramente modificados para facilitar o trabalho. Eles podem ser obtidos nos seguintes *sites*:

- Dados da Fórmula-1: Ergast Developer API
- Países e Cidades do planeta: GeoNames
- Aeroportos: OurAirports

Atividades da semana

- O objetivo deste trabalho é praticar o uso de **gatilhos** (*triggers*) nos SGBDs.
- É obrigatória a criação de *triggers* em todos os exercício, exceto se a questão apontar o contrário.

Exercício 1) Na tabela `AIRPORTS`, há vários Aeroportos cujo atributo `City` apresenta um nome que não aparece na tabela `GEOCITIES15K` no atributo `name`. A equipe responsável pelo projeto decidiu que, em relação aos dados já persistidos, nada irá ser feito. No entanto, para novos aeroportos inseridos na base, será obrigatório vinculá-los a uma cidade. Sua equipe foi designada para desenvolver uma *trigger* que avalia se o atributo `City` de um aeroporto sendo inserido (ou modificado) na tabela `AIRPORTS` não encontra correspondência com o atributo `Name` da tabela `GEOCITIES15K`. Quando isso aconteçer, a operação não deve ser concluída e uma exceção deve ser lançada.

(a) Nome da função: `VerificaAeroporto`

(b) Nome do *trigger*: `TR_Airports`

(c) Mensagem da exceção: `'Cidade não encontrada! Operação cancelada.'`

Faça ao menos um teste para inserção e um para atualização, e mostre-os no relatório junto do resultado.

Exercício 2) 2. Execute o SEGUINTE *script* e faça o que se pede.

```
CREATE TABLE Results_Status (
    StatusID INTEGER PRIMARY KEY,
    Contagem INTEGER,
    FOREIGN KEY (StatusID) REFERENCES Status(StatusID)
    );

INSERT INTO Results_Status
    SELECT S.StatusId , COUNT (*)
       FROM Status S JOIN Results R ON R.StatusID = S.StatusID
       GROUP BY S.StatusId , S.Status;
```

RESPONDA: O que esse *script* faz?

Depois, crie uma única *trigger* para as questões (a), (b) e (c) e outra para a questão (d). A *trigger* das três primeiras questões deve se chamar `TR_ResultsStatus`, e a função associada `AtualizaContagem`;
A trigger da questão (d) deve se chamar `TR_Results` e a função relacionada `VerificaStatus`:

(a) Ao inserir novas tuplas na tabela `Results`, incremente as quantidades inseridas nos respectivos `status` na tabela `Results_Status`. Além disso, mostre na tela a mensagem
`'StatusID: <status modificado>, Contagem: <nova contagem>.'`

(b) Ao remover tuplas da tabela `Results`, diminua as quantidades removidas nos respectivos `status` da tabela `Results_Status`. Além disso, mostre na tela a mensagem
`'StatusID: <status modificado>, Contagem: <nova contagem>.'`

(c) Ao atualizar tuplas da tabela `Results` (especificamente o atributo `statusid`), altere, na tabela `Results_Status` as quantidades removidas nas respectivas escuderias. Além disso, mostre na tela a mensagem
`'StatusID Anterior: <status Anterior>, Contagem: <nova contagem>'` e
`'StatusID Atual: <status atual>, Contagem: <nova contagem>`,
em que a primeira mensagem é o `Status` que teve a quantidade subtraída, e a segunda se refere à contagem que teve a quantidade aumentada.

(d) Antes de inserir novas tuplas na tabela `Results` ou atualizar o `Statusid` de alguma delas, verifique se o `StatusId` fica negativo. Caso fique, levante uma exceção com a mensagem
`'StatusID Negativo! Operação cancelada.'`
e não execute a operação.

Execute comandos para testar cada um dos casos e apresente os testes e resultados no relatório.

Entrega

Cada equipe deverá entregar dois arquivos no `Escaninho` do `Tidia`, sendo que apenas o líder de cada equipe deverá colocar, no seu *Escaninho*, em uma pasta com o nome `T5`:

- Um arquivo com o script no formato `.zip`, contendo um arquivo `.sql` com os comandos `SQL` utilizados para cada atividade.
- Um arquivo com o relatório SUCINTO no formato `.pdf`.
- O relatório deverá apresentar testes (pelo menos um por questão) para cada um dos procedimentos/funções criados em cada questão, com uma captura de tela do resultado para cada teste.

- Quando os resultados forem muito longos, o grupo deverá apresentar somente as primeiras tuplas (por exemplo, as 10 primeiras linhas de uma tabela resultante).

Como a atividade tem prazo de entrega máximo de uma hora antes da próxima aula, os arquivos devem ser submetidos

até às 18h00 do dia 31 de maio, *com postagem somente do lider da equipe*.

Não serão aceitos projetos feitos à mão e a organização clara das respostas também é um ponto avaliado.

Plágio será avaliado com nota zero.

Bom Trabalho!