



Universidade de São Paulo - São Carlos
Instituto de Ciências Matemáticas e de Computação
SCC-541 Laboratório de Bases de Dados
Trabalho Prático T6 – Exercícios sobre Visões (*views*)

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

Data para entrega: 07 de junho

Data de entrega para **todos** os grupos: 28 de junho até as 18:00h (via ttdia)

Datasets

Base de dados da Fórmula 1 - FIA.

A base de dados a ser utilizada é a mesma que vem sendo preparada desde o **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. Os originais podem ser obtidos nos seguintes *sites*:

- Dados da Fórmula-1: [Ergast Developer API](#)
- Países e Cidades do planeta: [GeoNames](#)
- Aeroportos: [OurAirports](#)

Atividades do projeto

- O objetivo deste trabalho é integrar o uso de todos os conhecimentos estudados durante a disciplina, com a criação do protótipo de uma ferramenta que seja capaz de manipular os dados e gerar relatórios através de uma interface amigável.

Orientações gerais

No desenvolvimento da ferramenta, os seguintes pontos devem ser observados:

- As informações devem ser apresentadas de forma intuitiva. Por exemplo, os nomes das colunas nos relatórios devem estar inteligíveis na Língua Portuguesa.
- Algumas das informações solicitadas já foram trabalhadas de alguma forma em atividades anteriores.
- Os comandos SQL utilizados na aplicação devem estar explícitos no código. Não podem ser utilizadas ferramentas que automatizem ou ocultem do desenvolvedor os *scripts* executados.

1 Administrar Usuários

O acesso à ferramenta somente pode ser feito a partir do *login* que cada usuário deve efetuar para ter acesso à base de dados. Cada usuário com acesso deve ser de um dos seguintes tipos:

Admin: Pode acessar quaisquer informação da base.

Login: `admin`, *Senha:* `admin`

Escuderia: Pode acessar apenas informações relativas à sua Escuderia e aos pilotos que correm por ela.

Login: `<constructorref>_c`, *Senha:* `<constructorref>`.

Exemplo: `<driverref> = mclaren`, *Login:* `mclaren_c`, *Senha:* `mclaren`.

Piloto: Pode acessar informações relativas ao seu desempenho.

Login: `<driverref>_d`, *Senha:* `<driverref>`. Exemplo: `<driverref> = hamilton`, *Login:* `hamilton_d`, *Senha:* `hamilton`.

Pontos que devem ser tratados:

1. Deve ser criada uma tabela chamada `USERS` com os seguintes atributos: `Userid`, `Login`, `Password`, `Tipo`, `IdOriginal` (`id` na tabela de origem).
2. A *password* deve utilizar a função `md5()` do SGBD para armazenar os dados.
3. Cada usuário só pode ser de um dos tipos: `'Administrador'`, `'Escuderia'` ou `'Piloto'`.
4. Os pilotos e escuderias já cadastrados na base deverão ser cadastrado também na tabela `USERS`, automaticamente.
 - Deve se assegurar que sempre que um piloto ou escuderia é criado ou modificado na respectiva tabela, o registro correspondente na tabela `USERS` deve ser corrigido também.
5. Deve ser criada uma tabela chamada `Log-Table` para armazenar o *log* de todos os acessos ao sistema, onde cada registro deve constar de:
 - `userid` do usuário logado,
 - data e
 - hora do *login*.

2 Telas da ferramenta

A ferramenta deve estar centrada em três telas principais, cada uma correspondente a um modo de operação da ferramenta.

Tela 1: Tela de *login*.

Depois de confirmado o *login*, deve ser mostrada a **Tela 2**.

Tela 2: Tela de *overview*.

Essa tela deve apresentar:

- O nome do usuário logado, dependendo do seu tipo:
`Admin`, nome do construtor ou nome completo (forename+surname) do piloto.
- Informações de *overview* de acordo com o tipo do usuário (veja descrição abaixo) e *links* ou botões que permitam executar as `Ações Possíveis dos Usuários`.

- Caminho (botão ou *link*) para a **Tela 3**.

Tela 3: Tela de relatórios.

Deverão ser apresentados botões ou algo análogo para cada um dos relatórios, os quais varia de acordo com o usuário logado.

Sempre que um deles for ativado, a tela deve apresentar o resultado correspondente.

3 Ações possíveis dos usuários

Admin: • **Cadastrar Escuderias:** exibe uma janela que permite ao usuário inserir os dados para adicionar uma nova tupla na tabela **CONSTRUCTORS**. Os dados que deve ser inseridos são:

ConstructorRef, **Name**, **Nationality** e **URL**.

- **Cadastrar Pilotos:** exibe uma janela que permite ao usuário inserir os dados para adicionar um novo piloto na tabela **DRIVER**. Os dados a serem inseridos são:

- **Driverref**, **Number**, **Code**, **Forename**, **Surname**, **Date of Birth**, **Nationality**.

Quando houver um novo cadastro de Escuderia ou Piloto, o sistema deverá inserir automaticamente os novos usuários na tabela **USERS** usando **triggers** conforme os padrões de *login* e senha indicados anteriormente. Caso já haja algum usuário com o *login* informado, a **trigger** deve cancelar a inserção na tabela **USERS** e na tabela da Escuderia ou Piloto, conforme o caso.

Escuderias: • Consultar por **Forename**: exibe uma janela que permite a inserção de um novo nome. O programa deve verificar se há algum piloto com esse nome (**forename** igual ao nome recebido) que já tenha corrido pela Escuderia logada. Caso existam, a ferramenta deve apresentar o nome completo, data de nascimento e nacionalidade do pilotos já existente.

Dica: para verificar se um piloto já correu por uma Escuderia, verifique a tabela **RESULTS**.

Piloto: Não pode alterar nada na base, mas apenas visualizar seus relatórios e *overview*.

4 Definição da Tela de Overview

Para cada tipo de usuário, a tela de *overview* deve apresentar informações diferentes, da seguinte maneira:

- **Admin:**
 - Quantidade de pilotos cadastrados
 - quantidade de escuderias cadastradas;
 - quantidade de corridas cadastradas;
 - quantidade de temporadas (*seasons*) cadastradas.
- **Escuderia:** Devem ser criadas funções PL/SQL que recebam dados da Escuderia como parâmetros e retornem os seguintes dados:
 - quantidade de vitórias da escuderia;
 - quantidade de pilotos diferentes que já correram pela escuderia;
 - primeiro e último ano em que há dados da escuderia na base (pela tabela **RESULTS**).

- **Piloto** devem ser criadas funções PL/SQL que recebam os dados do Piloto como parâmetros e retornem os seguintes dados:
 - Quantidade de vitórias do piloto;
 - Primeiro e último ano em que há dados do piloto na base (pela tabela [RESULTS](#)).

5 Relatórios

Apresente os relatórios considerando que eles devem ser compreendidos por um usuário do respectivo tipo. É interessante, por exemplo, aplicar alguma ordenação que faça sentido em cada relatório.

- **Admin:**
 - **Relatório 1:** Indica a quantidade de resultados por cada *status*, apresentando o nome do *status* e sua contagem.
 - **Relatório 2:** Receber o nome de uma cidade e, para cada cidade que tenha esse nome, apresenta todos os aeroportos brasileiros que estejam a, no máximo, 100 Km da respectiva cidades e que sejam dos tipos '[medium airport](#)' ou '[large airport](#)'. é possível haver diversas cidade com o mesmo nome, assim poderá haver várias tuplas, cada uma com um aeroporto que atenda às condições da pesquisa. O relatório deve apresentar:
 - * O nome da cidade,
 - * O código IATA do aeroporto,
 - * Nome do aeroporto,
 - * Cidade do aeroporto (atributo [city](#) de [AIRPORTS](#)),
 - * Distância da cidade ao aeroporto e
 - * Tipo do Aeroporto.
 * Crie também um índice que ajude nessa consulta.
- **Escuderias:** Recomenda-se a criação de funções PL/SQL que recebam como parâmetro o id do construtor logado.
 - **Relatório 3:** Lista os pilotos da escuderia, bem como a quantidade de vezes em que cada um deles alcançou a primeira posição em uma corrida. Os pilotos são identificados por seu nome completo.
Dica: Para verificar se um piloto já correu por uma Escuderia (e também para verificar se houve vitória), consulte a tabela [RESULTS](#).
 Crie também um índice que ajude nessa consulta.
 - **Relatório 4:** Lista a quantidade de resultados por cada status, apresentando o status e sua contagem, limitadas ao escopo de sua escuderia.
- **Piloto:** Recomenda-se a criação de funções PL/SQL que recebam como parâmetro o id do piloto logado.
 - **Relatório 5:** Consultar a quantidade de vitórias obtidas, apresentando o ano e a corrida onde cada vitória foi alcançada.
 Esse relatório deverá utilizar o comando [ROLLUP](#) para permitir a visualização de vitórias por ano e corrida. Portanto, deve aparecer na tabela as sumarizações por ano, por ano e corrida e uma sumarização geral.

As informações devem estar restritas apenas ao Piloto logado. Crie também um índice que ajude nessa consulta.

- **Relatório 6:** lista a quantidade de resultados por cada *status*, apresentando o *status* e sua contagem, limitada ao escopo do piloto logado.

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**.
- Comentários no código para auxiliar no entendimento são importantes.
- Na apresentação, serão feitas perguntas individuais aos membros do grupo, o que comporá a nota individual. As perguntas poderão abordar qualquer parte do projeto.
- Todos os membros devem mostrar claramente sua contribuição no desenvolvimento do trabalho.
- Os principais critérios a serem avaliados são:
 - funcionalidades implementadas;
 - índices criados;
 - funções e *triggers* criados;
 - usabilidade do sistema;
 - correteza das soluções;
 - justificativas das decisões tomadas.
- Plágio será avaliado com nota zero.

ATENÇÃO: Todos os grupos devem entregar o projeto até às 18h00 do dia 28 de junho, independentemente da data de apresentação do grupo.

A postagem é feita somente pelo líder da equipe.

Bom Trabalho!