Instituto de Ciências Matemáticas e de Computação – USP

Projeto Final (PF)

Disciplina: SCC541 - Laboratório de Bases de Dados

Prof. Dr. Caetano Traina Jr. - caetano@icmc.usp.br
PAE: Igor Alberte R. Eleutério - igoralberte@usp.br

22/06/2022

Data de entrega (todos os grupos): 06/07 até às 18h55 (via Tidia)

As apresentações dos grupos 1 ao 7 serão no dia 06/07. O restante dos grupos se apresentará no dia 13/07.

Para realizar este trabalho, o grupo deve criar o banco de dados sobre a Fórmula 1 conforme procedimentos disponibilizados junto ao Trabalho Prático 1 (T1). Recomenda-se a recarga dos dados para limpar o que foi feito anteriormente.

O objetivo deste trabalho é praticar os conhecimentos estudados durante a disciplina com a criação de um protótipo que seja capaz de manipular os dados e gerar relatórios através de uma interface amigável.

Orientações gerais

No desenvolvimento do sistema, os seguintes pontos devem ser observados:

- As informações devem ser apresentadas de forma intuitiva. Por exemplo, os nomes de colunas dos 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. Ou seja, não devem ser utilizadas ferramentas que automatizem ou ocultem do desenvolvedor os *scripts* executados.

Usuários

Deve ser criada uma tabela chamada `USERS` com os seguintes atributos: `Userid`, `Login`, `Password`, `Tipo`, `IdOriginal` (`id` na tabela de origem). O `Password` deve utilizar a função

`md5` do SGBD para armazenar os dados. O tipo deve ser: `'Administrador'`, `'Escuderia'` ou `'Piloto'`, conforme o caso.

Os pilotos e escuderias já cadastrados na base deverão ser cadastrado na tabela `USERS`.

Além disso, **deve ser criada uma tabela chamada** `log_table` para armazenar o *log* de acessos ao sistema: `userid` do usuário logado, data e hora do *login*.

Os tipos de usuários são:

- **Admin:** pode acessar quaisquer informações da base.
  ***Login*:** `admin`, **Senha:** `admin`.
- **Escuderia:** pode acessar apenas informações relativas à sua Escuderia.
  ***Login*:** `<constructorref>_c`, **Senha:** `<constructorref>`.
  Exemplo: `<driverref> = mclaren`, ***Login***: `mclaren_c`, **Senha:** `mclaren`.
- **Piloto:** informações relativas ao seu desempenho.
  ***Login*:** `<driverref>_d`, **Senha:** `<driverref>`.
  Exemplo: `<driverref> = hamilton`, ***Login***: `hamilton_d`, **Senha:** `hamilton`.

Telas

- **Tela 1:** tela de *login*. Após feito o *login*, a Tela 2 deverá ser mostrada.
- **Tela 2:** tela de *overview*. Deve apresentar o nome do usuário logado: Admin, nome do construtor ou nome completo (*forename*+*surname*) do piloto, dependendo do usuário logado. Além disso, devem ser apresentadas as **informações de *overview*** de acordo com o usuário (descrição abaixo) e *links* ou botões que permitam executar as **Ações Possíveis dos Usuários**. Também deve haver um caminho para a Tela 3.
- **Tela 3:** tela de relatórios. Deverão ser apresentados botões ou algo análogo para os **relatórios possíveis**, de acordo com o usuário logado, para que quando um deles for escolhido, a tela apresente seu resultado.

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 a serem inseridos são: `ConsctructorRef`, `Name`, `Nationality` e `URL`.
  - **cadastrar Pilotos:** exibe uma janela que permite ao usuário inserir os dados para adicionar uma nova tupla 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, automaticamente, inserir os novos usuários na tabela de `USERS` **através de *triggers*** conforme os padrões de *login* e senha mencionados anteriormente. Caso já haja

algum usuário com o *login* informado o *trigger* deve cancelar a inserção na tabela `USERS` e na tabela de Escuderia ou Piloto, conforme o caso.

- **Escuderias:**
  - **consultar por *Forename*:** exibe uma janela que permite a inserção de um nome. O programa deve verificar se há algum piloto com esse nome (*forename* igual ao nome recebido) que já tenha corrido pela Escuderia logada. Apresentar nome completo, data de nascimento e nacionalidade dos pilotos, caso existam. **Dica:** para verificar se um piloto já correu por uma Escuderia, verifique a tabela `RESULTS`.
- **Piloto:** não pode alterar nada na base. Apenas visualizar seus relatórios e *overview*.

Tela de *Overview*

Para cada tipo de usuário, a tela de *overview* apresentará informações diferentes. As informações são:

- **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 como parâmetros os dados da Escuderia 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 como parâmetros os dados do Piloto 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`).

Relatórios

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

- **Admin:**
  - **Relatório 1:** quantidade de resultados por cada *status*, apresentando o *status* e sua contagem.
  - **Relatório 2:** receber o nome de uma cidade e, para cada uma das cidades com esse nome, apresentar todos os aeroportos **brasileiros** que estejam a, no máximo, 100 km dessas cidades e que sejam dos tipos `'medium_airport'` ou `'large_airport'`. Para cada cidade encontrada, poderá haver várias tuplas, cada uma com um aeroporto que atenda às condições. Apresentar nome da cidade, 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:** listagem dos seus pilotos, bem como a quantidade de vezes que alcançaram a primeira posição em corridas. Apresentar os pilotos por nome completo. **Dica:** para verificar se um piloto já correu por uma Escuderia (e também verificar se houve vitória), consulte a tabela `RESULTS`. **Crie, também, um índice que ajude nessa consulta.**
  - **Relatório 4:** 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 obtida apresentando ano e corrida. Esse relatório deverá utilizar o comando `ROLLUP` para permitir a visualização de vitórias por ano e corrida. Ou seja, aparecerá na tabela sumarizações por ano; por ano e corrida; e uma geral. As informações devem estar restritas apenas ao Piloto logado. **Crie, também, um índice que ajude nessa consulta.**
  - **Relatório 6:** quantidade de resultados por cada *status*, apresentando o *status* e sua contagem limitada ao escopo do piloto logado.

Entrega

As atividades deste trabalho devem ser entregues via Atividade do Tidia, em um **arquivo no formato *.zip*** contendo o código fonte do sistema. Comentários no código para auxiliar no entendimento também são bem vistos.

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; usabilidade do sistema; corretude das soluções; justificativas das decisões tomadas.

Plágio será avaliado com nota zero.

Bom Trabalho!