

Bases de Dados II

2018-2019

1. Descrição do sistema pretendido

O objetivo da plataforma é implementar uma base de dados, as respetivas vistas, funções procedimentos e *triggers* para suporte a um sistema *web*, que permita analisar o desenrolar do **campeonato português de futebol** para a primeira e segunda liga.

Na plataforma têm de estar registado, pelos menos os detalhes das equipas, jogadores, treinadores (principais e adjuntos), sanções disciplinares, resultados, classificação, melhores marcadores, guarda redes com mais golos sofridos, e outras informações que ache pertinente.

Deverá dividir e organizar os resultados por época futebolística e por primeira e segunda liga.

O sistema a projetar deve dar resposta às necessidades operacionais típicas de um sistema de gestão futebolista, mas também servir de apoio aos clubes, através da disponibilização de informação detalhada e agregada da informação de jogadores, equipas, sanções disciplinares, resultados, avisos disciplinares e prémios. Pode e deve acrescentar outros parâmetros.

Deve ler todo o enunciado de forma a perceber os requisitos de informação da base de dados.

Todas as melhorias ao sistema devem ser discutidas com o docente, no momento do desenho da base de dados.

2. Entregas

Pretende-se que o desenvolvimento do trabalho subjacente a este projeto seja efetuado de forma faseada ao longo do semestre. Deste modo, definiram-se quatro momentos para entregas parciais dos diversos elementos pedidos, bem como um momento adicional para defesa individual do trabalho, de acordo com o seguinte.

2.1. Análise de dados

Nesta fase deve ser feita a análise de dados do sistema proposto de modo a obter:

- Diagrama do modelo E/R (utilize designações sugestivas para as entidades, relacionamento e atributos).
- Diagrama do modelo físico de tabelas do SGBD Oracle.
- Dicionário de dados

- *Script* SQL de criação da base de dados.

2.2. Elaboração de pesquisas à base de dados

Nesta fase, a base de dados deve ser carregada com dados para que seja possível testar as suas funcionalidades. (Pode utilizar o <https://mockaroo.com/>)

Assim, devem ser entregues os seguintes elementos:

- **Scripts de inserção de dados** nas diversas tabelas.
- **Implementação de consultas** à base de dados, incluindo uma amostra dos respetivos resultados para verificação da correção das mesmas. Estas consultas devem ser obtidas através de comandos SELECT, e quando aplicável, guardadas como vistas. Podem ser criadas vistas auxiliares para simplificar esses comandos. Serão desvalorizados os comandos que não sejam corretamente comentados. As vistas criadas devem ter um nome do tipo VISTA_nome_da_vista (ex. VISTA_A, VISTA_B, ...). Algumas das consultas a implementar são:
 - a) VISTA_A, que no que respeita às equipas, apresente:
 - A equipa que ganhou o penúltimo campeonato;
 - A equipa que teve mais cartões amarelos (jogadores e treinadores) nesta época;
 - A equipa que sofreu mais golos, em casa, no ultimo mês.
 - b) VISTA_B, que mostre, quantos jogos foram realizados por equipas com treinadores de nacionalidade portuguesa, na presente época.
 - c) VISTA_C que apresente a lista de jogadores portugueses que jogam em equipas da segunda divisão esta época e que nunca receberam uma sanção disciplinar.
 - d) VISTA_D lista de jogadores, que jogam na posição “extremo”, que no último mês marcaram pelo menos dois golos.
 - e) VISTA_E que apresente o numero total de golos marcados por equipas de Lisboa e do Porto na ultima época.
 - f) VISTA_F que apresente o clube e os nomes dos jogadores que receberam cartões vermelhos nos últimos três meses. Considere apenas os jogadores a jogar na posição “guarda-redes” e que tenham sido utilizados em mais do que três jogos. (apresente o resultado por ordem alfabética)

- g) VISTA_G que apresente os quatros melhores goleadores, fora de casa, das ultimas duas épocas.
- h) VISTA_H que obtenha uma lista de treinadores que tenham ganho mais do que três jogos consecutivos e menos do que duas derrotas no campeonato. Ordene por ordem decrescente pelo número de vitórias.
- i) VISTA_I que para a equipa “AAC”, apresente uma listagem referindo quem foram os jogadores que estiveram presentes nas ultimas três jornadas. Ordene pela data da jornada.
- j) VISTA_J que para todos os jogos da ultima jornada, apresente uma listagem com o nome do estádio onde se realizou o jogo, nome do treinador, resultado final, quantidade de golos sofridos e marcados. Ordene decrescentemente pelo nome da equipa.
- k) VISTA_K que apresente uma lista de treinadores que trocaram de equipa nos últimos dois anos.
- l) VISTA_L que, para cada clube, apresente a lista de resultados por jogo. Apenas considere jogos com mais de 4 golos, realizados entre as 17:00 e as 23:00. Ordene decrescentemente pelo número de golos.
- m) Pelo menos duas consultas adicionais, que o grupo considere relevantes, justificando a sua relevância. A relevância e o nível de complexidade das mesmas influenciará fortemente a sua avaliação:
 - a. VISTA_M que inclua um SELECT com GROUP BY.
 - b. VISTA_N que inclua um SELECT encadeado.

2.3. Procedimentos, funções e triggers

O trabalho a desenvolver nesta fase inclui a construção de procedimentos/funções em PL/SQL que facilitem a manipulação dos dados da base de dados (BD), a utilização de *triggers* para garantia da integridade dos dados (ou outras tarefas onde façam sentido) e a aplicação do conceito de transação. Pretende-se também que seja feito o tratamento das exceções.

Assim, pretende-se que sejam entregues os elementos necessários à implementação dos seguintes pontos:

- a) Implementar os mecanismos necessários a garantir a integridade dos dados que não seja assegurada por restrições da BD, incluindo as restrições do numero de cartões por jogo/jogador (garantir que um jogador só pode ter 2 cartões amarelos ou/ e um vermelho), que nenhum jogador/treinador pode pertencer em simultâneo a duas equipas e que uma equipa não pode ter mais do que 5 jogadores estrangeiros.

- b) Criar a função **numero_golos** que recebe como argumento um código de uma posição (ex: avançado, extremo) e retorna o numero de golos realizados por todos os jogadores dessa posição, nos últimos 3 meses. A função deve lançar exceções (ver exceções abaixo).
- c) Criar a função **numero_amarelos** que recebe como argumento o nome de um jogador, uma data inicial e uma data final. Esta função deve retornar o somatório do número de cartões amarelos, recebidos apenas em jogos em casa, nesse intervalo de tempo. A função deve lançar exceções (ver exceções abaixo).
- d) Criar o trigger **golos_marcados** que, após cada golo marcado, atualize uma tabela com os 10 melhores goleadores e atualize a tabela com os guarda-redes que mais golos sofreram. Deve ainda enviar um aviso de parabéns à equipa e ao treinador.
- e) Criar uma função **ultimo_jogo** que recebe o nome de uma equipa e retorna a data do último jogo que tenham perdido fora de casa. A função deve lançar exceções (ver exceções abaixo).
- f) Crie a função **primeira_substituição** que para um determinado jogo, determine quantos minutos jogou o jogador que foi substituído. A função deve retornar os minutos ou, NULL se nenhum potencial jogador tiver sido substituído.
- g) Implemente um **trigger cartões** que, ao inserir um cartão vermelho na ficha de um dado jogador na base de dados, submeta um aviso à equipa desse mesmo jogador a informar que não pode jogar na próxima jornada. Ao receber um cartão vermelho deve ainda garantir que ele se encontra no estado suspenso por 2 jogos.
- h) Criar a função **clubes_cidade** que recebe como argumento um Concelho (Ex Lisboa) e deve retornar a posição (avançados, guarda redes, etc) que mais jogadores, nacionais, tiveram inscritos na penúltima época desportiva. A função deve lançar exceções (ver exceções abaixo).
- i) Criar o procedimento **época_desportiva** que ao receber uma época desportiva, que introduza numa tabela auxiliar, três melhores equipas e as três piores equipas.
- j) Criar uma função **idade** que recebe uma idade, uma data inicial e uma data final. A função deve devolver o numero de golos marcados por esses jogadores, apenas na presente época. Deve excluir jogadores que não estão no clube à menos de duas épocas. A função deve lançar exceções (ver exceções abaixo).
- k) Implementar o procedimento **alertas_treinador** que recebe como argumento um id de um treinador e de seguida deve calcular, o numero de jogos realizados, o numero de

jogos ganhos. o numero de jogos perdidos, o numero de golos marcados, por clube onde treinou e por época desportiva

- l) Criar o ***trigger treinador*** garantindo sempre que for recebida uma sanção disciplinar a um treinador, com um cartão vermelho, este é automaticamente suspenso, por dois jogos. Também deve passar o treinador adjunto para o papel de treinador principal, durante o período da suspensão.
- m) Crie a função ***tempo_medio_venda*** que recebe como argumento o id de uma equipa, e retorna a média, em minutos, que essa equipa necessitou para marcar o primeiro golo com o penúltimo treinador. Considere apenas jogos em casa e que não tenham cartões vermelhos associados nesse jogo.
- n) Criar um **procedimento** que se considerem importantes para um futuro funcionamento do sistema.
- o) Criar uma **função** que considerem importante para um futuro funcionamento do sistema.
- p) Criar um **trigger** que considerem importante para um futuro funcionamento do sistema.

Tratamento de exceções:

Sempre que for necessário devem ser lançadas exceções, nomeadamente as seguintes:

Código	Mensagem
-20100	Código de posição YYY inválido
-20101	Nome YYY de jogador Invalido
-20102	Data YYY invalida
-20103	Nome YYY de treinador Invalido
-20104	Nome YYY de Concelho invalido
-20105	Idade Invalida
-20106	Nome/ID YYY de equipa Invalido
-20107	Época Invalida
-20108	Não existem Cartões
-20109	Não existem substituições
-20110	Não existem jogos
-20111	Não foram marcados golos

Serão desvalorizados os procedimentos, funções e *triggers* que não forem corretamente comentados.

Pode acrescentar novas exceções

2.4. Relatório

Devem acompanhar todo o trabalho realizado com um relatório que explique as decisões, opções e dificuldades do seu trabalho.

2.5. Defesa individual do trabalho

As defesas dos trabalhos serão realizadas na última aula prática. Espera-se que cada um dos elementos do grupo consiga, individualmente, explicar e/ou alterar qualquer parte do trabalho entregue.

Os scripts entregues que não funcionem não serão considerados para avaliação.

Considera-se que os alunos que não façam a defesa não realizaram o trabalho.

2.6. Notas Importantes

O trabalho terá de ser realizado por grupos de máximo de dois alunos

Prazos a cumprir:

13-10-2018 – *Definição dos grupos*

29-10-2018 – *CheckPoint 1* (serão alvo de análise na aula prática seguinte).

26-11-2018 – *CheckPoint 2* (serão alvo de análise na aula prática seguinte).

15-01-2019 – *CheckPoint 3* e entrega de relatório.

Defesa individual na ultima aula prática

Semanalmente: Até às 17h00 de todas as sextas, entrega obrigatória através do moodle da versão atual do que fizeram. Para o CheckPoint 2 e 3 o docente irá apresentar um plano de respostas mínimas semanais.