

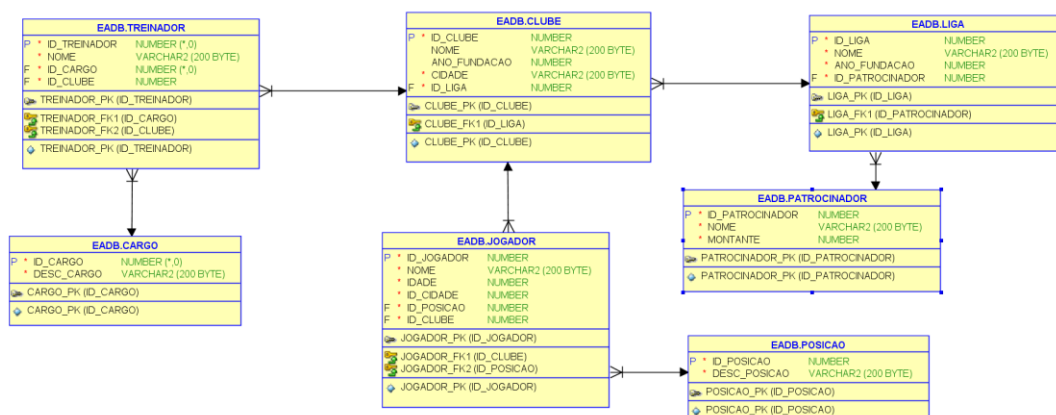


Curso: Mestrado Integrado em Informática
U.C.: Administração e Exploração de Bases de Dados

Folha de Exercício FE05	
Docente	Hugo Peixoto
Tema:	Criação de tablespaces, datafiles e users; Modelo Físico
Turma:	PL
Ano Letivo:	2017-18 – 1º Semestre
Duração da aula:	2 horas

1. Enunciado

- [1] Criar um Tablespace e respetivo Datafile com as opções por defeito para alocar a base de dados:
O nome do tablespace será: aedb_tables;
A localização do datafile deverá seguir os restantes da Base de Dados \u01\app\oracle\oradata\orcl12\orcl\...;
O nome do datafile será: aedb_tables_01.dbf;
Com o tamanho total de 100M;
Os restantes parâmetros serão os por defeito.
- [2] Criar um Tablespace temporário e respetivo Datafile com as opções por defeito para alocar as transações temporárias:
O nome do tablespace será: aedb_temp;
A localização do datafile deverá seguir os restantes da Base de Dados (\u01\app\oracle\oradata\orcl12\orcl\...)
O nome do datafile será: aedb_temp_01.dbf
Com o tamanho total de 50M;
O autoextend deverá ser ativado;
Os restantes parâmetros serão os por defeito.
- [3] Através da view de system *dba_tablespaces*, confirmar a correta execução dos passos anteriores;
- [4] Criar um user para a base de dados;
- [5] Conceder o Grant de acesso à base de dados ao user criado
- [6] Através da view de system *dba_users*, confirmar a correta execução do passo anterior;
- [7] Efetue a ligação ao user através do SQLDeveloper.
- [8] Recorrendo ao SQLDeveloper e ao modelo relacional abaixo crie a base de dados relacional.





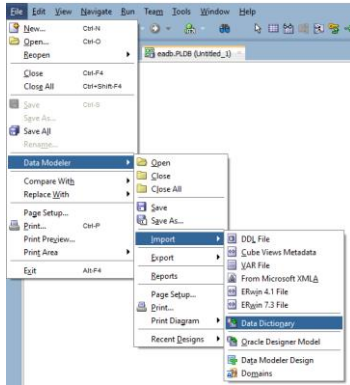
[9] Criar as tabelas com as respetivas relações (Primary Keys, Foreign Keys e índices).

[a] Deverá utilizar o comando *create table* para a tabela *Patrocinador*.

[b] As restantes tabelas poderão ser usadas através da interface do SQLDeveloper.

Nota: Atente que os índices serão criados de forma automática em cada uma das tabelas se uma chave primária for criada.

[10] Utilizando a função Data Modeler do SQL developer faça a criação do modelo relacional idêntico ao da figura acima descrita:



[11] Faça o download dos ficheiros que têm os scripts com os dados para cada tabela (scripts.sql.zip) na página da UC #Aula6.

[a] Corra o script de importação dos dados através do SQLDeveloper para cada uma das tabelas.

Tenha em atenção a ordem de importação dado que a relação entre as tabelas pode afetar a execução dos scripts!

[b] De cada vez que corre o script deverá confirmar a importação “Commit”.

[c] Deverá abrir a tabela no SQLDeveloper no separador “Data” e confirmar a importação dos dados.

[12] Construir uma *view* denominada JOGADOR_NEW que contenha a seguinte informação:

ID_JOGADOR; NOME; IDADE; NOME_CLUBE; DESCRICAO_POSICAO, NOME_LIGA; CIDADE_CLUBE; ANO DE FUNDACAO

[13] Listar todos os defesas direitos da Segunda Liga.

[14] Listar todos os jogadores com menos de 27 anos cuja posição é Trinco e que não jogam na II Liga.

[15] Construir uma *view* denominada TREINADOR_NEW que contenha a seguinte informação:

ID_TREINADOR; NOME_TREINADOR; NOME_CLUBE; DESCRICAO_DO_CARGO

[16] Listar todos o nome do treinador, nome do clube, cargo do treinador, cidade do clube e ano de fundação de todos os clubes fundados após 1945.

[17] Criar uma sequência denominada JOGID_SQ através da interface do *SQLDeveloper* que seja incremental. Comece em 1000, aumente apenas uma unidade e o limite deverá ser 99999.

[18] Criar um *trigger* denominado JOGID_TRIG que para antes de cada *insert* na tabela de JOGADORES crie o JOGADOR_ID de forma automática utilizando a sequência criada anteriormente.

[19] Criar o script de introdução de 3 Jogadores com nomes, clubes, posições e idades aleatórios, mas cujo o ID seja criado utilizando o *trigger* e sequências criados.