

DI-FCT-NOVA

18 de junho de 2016

Bases de Dados (10640)

Exame de Recurso, 2015/16

Consulta limitada

Parte I

(correspondente à matéria do 1º teste)

Grupo I.1

Nota: O enunciado deste grupo é *intencionalmente* vago. Ao responder deve, sempre que necessário, dizer o que assumiu e que acha que não está suficientemente detalhado no enunciado.

[Exame] O portal de um canal de notícias decidiu fazer uma aplicação online sobre o **EURO2016**, para permitir aos seus utilizadores consultar informação sobre as várias equipas participantes, jogadores, arbitro, jogos, estádios, etc. A sua tarefa neste grupo é propor uma base de dados relacional que sirva de base a esta aplicação, e que permita armazenar toda essa informação.

Mais concretamente, a base de dados deve permitir armazenar informação sobre as várias seleções participantes na prova, todas elas pertencentes a um país europeu. De cada país, pretende-se apenas guardar o seu nome, e um código único.

Cada seleção tem uma sigla da respetiva federação, um treinador e vários jogadores. Do treinador, é guardado o seu nome, a sua data de nascimento e a sua nacionalidade, que pode ser diferente do país de que é selecionador, e que pode até não ser europeu (vide Scolari e seleção de Portugal, há uns anos). De cada jogador, pretende guardar-se o seu nome, data de nascimento, em que clube joga (e de que país é esse clube) e a posição onde joga (que só pode ser 'Guarda Redes', 'Defesa', 'Médio' ou 'Avançado'). Cada jogador tem um nº, que é único na seleção, mas que, obviamente, não é único para o conjunto das seleções. E.g. só pode haver um jogador nº 7 na seleção de Portugal, mas há vários nº 7 no conjunto de todas as seleções.

Quanto aos jogos, que têm sempre um identificador único, pretende saber-se entre que seleções é, em que data, hora e estádio se realiza, qual o resultado final e que jogadores participam no jogo.

Quanto aos estádios, deve guardar-se pelo menos o seu nome, lotação, o nome da localidade onde se situa, e o clube a que o estádio pertence.

1. [Exame]

- a) Esboce um diagrama de entidades e relações para a base de dados sobre o **EURO2016**.
[Cotação em exame: 2,5 valores; Cotação em substituição de teste: 4,5 valores]
- b) Proponha uma base de dados relacional para a armazenar a informação sobre o **EURO2016**. Para cada uma das relações que propuser deve indicar a chave primária, e todas as chaves estrangeiras. Deve ainda indicar qual o domínio de cada um dos atributos. Para definição da base de dados pode usar SQL, ou outra qualquer notação, desde que seja claro quais as relações, chaves, atributos e respetivos domínios. **[2,5; 4,5]**

2. [EXTRA] Olhando bem para a modelação da base de dados, chegou-se à conclusão que havia várias partes que estavam demasiadamente simplificadas. Uma delas tem a ver com a posição onde jogam os jogadores.

Se é verdade que cada jogador tem uma posição típica onde costuma jogar, em jogos diferentes um jogador pode jogar em posições diferentes. Aliás, isso pode até acontecer num mesmo jogo. Por exemplo, pode bem acontecer que um dado jogador num dado jogo, jogue parte do tempo como ponta de lança, e outra parte como médio centro. Além disso, há muito mais posições possíveis (defesa central, lateral esquerdo, direito, ala, etc). O melhor mesmo seria ter uma solução geral que permitisse armazenar tantas posições de um jogador num jogo, quanto se queira.

- a) Esboce um diagrama de entidades e relações que modele esta nova especificação de participação de jogadores em jogo, e suas posições **[0; 1,5]**
- b) Proponha um conjunto de relações para armazenar esta informação (incluindo chaves primárias candidatas e estrangeiras. Pode usar SQL, ou outra qualquer notação, desde que seja claro quais as relações, chaves, atributos e respetivos domínios. **[0; 1,5]**

Grupo I.2

Para representar uma pequena parte da informação sobre os jogadores e árbitros que participam nos jogos, alguém criou a seguinte relação:

$$R = (\text{Jogador, Jogo, Número, Seleção, Posição, Árbitro, Nacionalidade})$$

onde Jogador (resp. Jogo, Seleção, Árbitro) identifica um jogador (resp. um jogo, uma seleção, um árbitro), Número é o número de um jogador, Posição uma posição que o jogador pode assumir durante um jogo, e Nacionalidade uma nacionalidade de um árbitro.

Para garantir que em cada jogo cada jogador só pode jogar por uma seleção, que cada árbitro só tem uma nacionalidade, que em cada jogo só há um árbitro e de uma nacionalidade, e que cada jogador só tem um nº e uma seleção, definiram-se as seguintes dependências funcionais F:

Jogador, Jogo \rightarrow Seleção
Árbitro \rightarrow Nacionalidade
Jogo \rightarrow Árbitro, Nacionalidade
Jogador \rightarrow Número, Seleção

- a) **[Exame]** Qual (ou quais) a(s) chave(s) candidata(s) de R? **[0,5; 2]**
- b) **[Exame]** Apresente uma cobertura canónica do conjunto de dependências F. **[0,75; 2]**
- c) **[Exame]** Usando o algoritmo dado nas aulas, decomponha sem perdas o esquema R, por forma a obter um conjunto de esquemas na forma normal de Boyce-Codd. **[0,75; 2]**
- d) **[EXTRA]** O conjunto de esquemas a que chegou preserva as dependências?
Se **acha que sim**, mostre que de facto todas as dependências são preservadas.
Se **acha que não**, decomponha R por forma a obter um esquema na 3ª forma normal que preserve as dependências. **[0; 2]**

Parte II

(correspondente à matéria do 2º teste)

A base de dados da parte anterior foi estendida para passar também a armazenar informação sobre o decorrer dos jogos. Nesse sentido foram criadas as seguintes tabelas (onde os atributos que constituem a chave primária estão sublinhados):

seleções(<u>idSelecao</u> , nomePais) jogadores(<u>idJogador</u> , idSeleção, nomeJog) equipalncial(<u>idJogo</u> , <u>idSeleção</u> , <u>idJogador</u>) substituições(<u>idOcorr</u> , idJogSai, idJogEntra)	jogos(<u>idJogo</u> , data, estádio) selecaoJogo(<u>idJogo</u> , <u>idSeleção</u>) ocorrencias(<u>idOcorr</u> , idJogo, minuto, tipo, texto) golos(<u>idOcorr</u> , idEquipa)
---	---

A tabela seleções guarda informação sobre todas as seleções, cada uma com um identificador único e um nome. A de jogos, tem um identificador para cada cada jogo, a data e o estádio em que ocorre. Um tuplo na tabela selecaoJogo indica que a seleção com identificador idSeleção participou no jogo com idJogo (claro que para cada jogo só pode haver dois tuplos nesta tabela, mas já lá vamos).

A tabela de jogadores tem todos os jogadores do torneio, cada um com um identificador, um nome, e indicação de a que seleção pertencem (em idSeleção). A tabela equipalncial guarda a constituição da seleção no início do jogo, i.e. quais os jogadores (em idJogador) que alinharam de início num dado jogo (em idJogo) por uma dada seleção (em idSeleção).

As outras 3 tabelas destinam-se a guardar as várias ocorrências do jogo, para as ir mostrando numa aplicação online, como a que se encontra na figura. Em ocorrências encontram-se as várias ocorrências de um jogo, cada uma delas tem um identificador idOcorr, uma referência do jogo a que dizem respeito (em idJogo), o minuto em que aconteceram, um tipo (que serve para eventualmente colocar o icon como se vê na figura, e que pode ter valores como, 'Início do Jogo', 'Fim do jogo', 'Falta', 'Golo', 'Substituição', etc, ou null) e um campo de texto livre.

Existem mais duas tabelas para dois tipos particulares de ocorrências. Uma armazena os golos, em que idOcorr é, para além de chave primária, chave estrangeira que referencia idCorr em ocorrências, e que guarda também o identificador da equipa que marcou o golo. A outra, de substituições, que também se refere a uma ocorrência na tabela de ocorrências, que guarda informação sobre o (identificador do) jogador que saiu e o que entrou.

Nesta base de dados, para além das chaves primárias indicadas, estão ainda definidas as seguintes chaves estrangeiras: idSeleção em qualquer uma das tabelas referencia idSeleção em seleções; idJogo em qualquer uma das tabelas referencia idJogo em jogos; idOcorr em qualquer uma das tabelas referencia idCorr em ocorrências.

Grupo II.1

1. Apresente uma expressão em **álgebra relacional** e uma **consulta SQL** para cada uma das perguntas:
 - a) [Exame] Qual o nome de todos os jogadores que alinharam de início em jogos no dia 14/06/2016? [1; 3]
 - b) [Exame] Quais os nomes dos jogadores que nunca alinharam de início em qualque jogo? [1; 3]
 - c) [EXTRA] Quais os jogadores de Portugal que alinharam de início em todos os jogos (de Portugal, e para os quais já há equipa inicial, claro)? [0; 3]

55'	Cartão amarelo para B. Bjarnason.
50'	G0000000000000000LO de B. Bjarnason. Primeiro golo da Islândia em grandes competições. Aparece solto ao segundo poste, a encostar de pé direito, nas costas de Vieirinha. O lateral paga pela descoordenação com Pepe, perante uma troca de lugar entre os dois jogadores Islandeses que estavam na área.
47'	RONALDO ATIRA AO LADO! O capitão aproveita um corte de Ragnar Sigurdsson para a entrada da área e atira de primeira, de pé esquerdo, mas o remate sai enrolado e ao lado
46'	Recomeça o Portugal - Islândia
45'	Portugal fez cinco remates à baliza, três dos quais por Nani, o autor do golo.
45'	Intervalo no Portugal - Islândia Após algumas dificuldades iniciais para impor o seu jogo, a seleção portuguesa conseguiu assumir o domínio do encontro e criou oportunidades suficientes para justificar a vantagem, alcançada com um golo de Nani, ao minuto 31.

2. Apresente **uma consulta SQL** para cada uma das perguntas:
- a) **[Exame]** Apresentar a lista de todas as ocorrências do jogo Portugal-Islândia, cada uma com o tipo de ocorrência, o minuto, e o respetivo texto, por ordem inversa daquela em que foram ocorrendo (e.g. como aparece na figura). **[0,75; 1,5]**
 - b) **[Exame]** Quantas substituições houve em cada jogo? **[0,75; 1,5]**
 - c) **[EXTRA]** Quais as seleções com uma média de golos por jogo superior a 1? **[0; 1,5]**

Grupo II.2

1. A base de dados descrita acima não consegue garantir uma série de condições que é normal serem garantidas em jogos de futebol. Neste grupo vamos resolver alguns desses problemas. Imponha restrições de integridade sobre a base de dados acima, para:
- a) **[Exame]** Garantir que em cada jogo participam exatamente duas equipas. **[0,75; 1,5]**
 - b) **[Exame]** Garantir que os jogadores que alinham de início numa seleção são sempre jogadores dessa seleção. **[0,75; 1,5]**
 - c) **[EXTRA]** Garantir que só podem marcar golos as seleções que participam no jogo em que a ocorrência do golo está introduzida. **[0; 1,5]**
2. Com esta base de dados, não é muito fácil calcular o resultado final de cada jogo com base nas ocorrências do jogo. Como tal, decidiu-se ter mais uma tabela com os resultados finais dos jogos, com o seguinte esquema:

`golosFinal(idJogo, idSelecao, golos)`

Assim, para cada jogo terminado, deverão existir dois tuplos nesta tabela, ambos com o identificador do jogo, e cada um deles com o identificador de uma das seleções e o nº de golos que essa seleção marcou nesse jogo.

- a) **[Exame]** Comece por criar uma view `golosEquipaJogo` que retorna, para cada jogo e seleção, quantos golos marcou essa seleção nesse jogo **[0,75; 1]**
- b) **[Exame]** Usando a view `golosEquipaJogo` (se não a criou assuma que existe), crie agora os mecanismos necessário na base de dados para que esta tabela seja preenchida automaticamente. Ou seja, para que, sempre que é introduzida uma ocorrência de final de jogo, seja introduzida nesta nova tabela o resultado final desse jogo.
(Pode, se preferir, usar a linguagem PL/SQL) [0,75; 1]

Parte III

(correspondente à matéria do 3º teste)

Considere novamente a base de dados de eventos e resultados de jogos, descrita na Parte II, incluindo as restrições impostas no Grupo II.2.

Grupo III.1

1. Apresente um conjunto de regras em Datalog que permita obter resposta às seguintes perguntas:
 - a) **[Exame]** Qual o nome dos jogadores de Portugal que alinharam de início no jogo Portugal-Islândia **[0,75; 1]**
 - b) **[Exame]** Quais os nomes dos jogadores que nunca alinharam de início em qualquer jogo? **[0,75; 1,25]**
 - c) **[EXTRA]** Quais os jogadores que estavam em jogo no final do jogo Portugal-Islândia? **[0; 1,25]**

Grupo III.2

1. **[Exame]** Na definição das propriedades ACID, a consistência só é exigida no final de cada transação, e não no final de cada ação que constitui cada uma das transações. Mostre através de um exemplo de operação sobre a base de dados acima, que exigir a verificação de consistência após cada ação, para além de desnecessário, seria prejudicial. **[0,75; 1,5]**
2. Considere as seguintes transações concorrentes executadas numa situação em que inicialmente a tabela de golos está vazia. Não assuma, para já, nenhuma ordem entre as operações de uma e da outra transação:

Transação 1	Transação 2
begin transaction select * from golos; insert into golos values (1,...); select * from golos; commit;	begin transaction select * from golos; insert into golos values (2,...); select * from golos; commit;

- a) **[Exame]** Apresente um escalonamento das ações nestas duas transações em que o resultado seja diferente consoante se esteja em modo de isolamento **Read uncommitted** ou **Read committed**, mostrando qual o resultado para cada um destes 2 modos de isolamento. **[1; 1,5]**
- b) **[Exame]** Apresente um escalonamento das ações nestas duas transações em que o resultado seja diferente consoante se esteja em modo de isolamento **Read committed** ou **Serializable**, mostrando qual o resultado para cada um destes 2 modos de isolamento. **[1; 1,5]**

Grupo III.3

Para transferir dados de ocorrências de golos e substituições de um jogo, e para as mostrar posteriormente numa página Web, decidiu-se usar ficheiros XML de acordo com o seguinte DTD:

```

<!DOCTYPE jogo[
  <!Element jogo(equipaCasa, equipaFora, ocorrencia*)>
    <!ATTLIST jogo data #REQUIRED >
  <!Element equipaCasa(nomeEquipa)>
    <!ATTLIST jogo idEquipa ID #REQUIRED >
  <!Element equipaFora(nomeEquipa)>
    <!ATTLIST jogo idEquipa ID #REQUIRED >
  <!Element ocorrencia(golo | substituicao)>
  <!Element golo(marcador)>
    <!ATTLIST golo equipa IDREF #REQUIRED minuto>
  <!Element substituicao(sai,entra)>
    <!ATTLIST substituicao minuto>
  <!Element nomeEquipa(#PCDATA)>
  <!Element marcador(#PCDATA)>
  <!Element sai(#PCDATA)>
  <!Element entra(#PCDATA)>
]>

```

1. **[Exame]** Apresente o texto dum documento XML que use este DTD, e que contenha a seguinte informação sobre o jogo Islândia-Portugal, realizado a 14/06/2016: **[1; 2]**
 - Golo de Portugal, marcado por Nani ao minuto 31;
 - Substituição de João Mário por Quaresma ao minuto 75.
2. Apresente expressões XPATH sobre um ficheiro de acordo com o DTD acima que devolvam os resultados das seguintes perguntas:
 - a) **[Exame]** Quais os jogadores que marcaram golos? **[0,5; 1]**
 - b) **[Exame]** Quais os jogadores que entraram ainda na 1ª parte (i.e. até ao minuto 45)? **[0,75; 1]**
 - c) **[EXTRA]** Qual o nome das equipas dos jogadores que marcaram golos na 2ª parte? **[0; 1]**
3. **[EXTRA]** Como terá reparado, o DTD acima não tem nenhuma informação sobre os vários jogadores que participaram no jogo, e como tal não garante que os marcadores e os jogadores substituídos são das equipas em jogo. A sua tarefa nesta pergunta é alterar o DTD, por forma a que:
 - O ficheiro XML tenha a informação dos vários jogadores de cada uma das equipas (com informação sobre a que equipa pertence, claro);
 - O marcador de cada golo é um dos jogadores dessas equipas – não necessariamente da equipa que marcou o golo, pois há que prever golos na própria baliza;
 - A substituição deve ter informação sobre a equipa em que ocorre, e essa equipa deve ser uma das que participa no jogo. **[0; 3]**

Grupo III.4

Resolveu-se alterar a base de dados apresentada na Parte II usando as *features* objeto-relacional do SGBD. Nomeadamente, pensou-se que seria uma boa ideia ter uma tabela de participação de seleções em jogos, semelhante à *selecaoJogo*, mas armazenando logo aí o conjunto de todos os jogadores que alinharam de início por essa equipa nesse jogo, onde cada jogador tem um nome e um número. Ter-se-ia então, em vez das tabelas *selecaoJogo* e *equipalnicial*, uma só tabela, *participalnicial*, com um atributo *idJogo*, um atributo *idSeleção*, e um atributo *início* com um conjunto de nomes e números de jogadores.

1. **[EXTRA]** Apresente o código SQL completo de criação da nova tabela *participalnicial*. **[0; 2]**
2. **[EXTRA]** Considerando a base de dados com esta tabela em vez da originais *selecaoJogo* e *equipalnicial*, apresente o código SQL para a pergunta: **[0; 2]**

“Quais os jogos onde participou de início um jogador chamado Ronaldo e com o nº 7”