

# Sistemas de Informação e Bases de Dados

Enunciado da Etapa 3 do Projeto de 2019/2020

## Ponto de partida

Considere para esta etapa do projeto o seguinte esquema relacional sobre a gestão da liga de futebol feminino, especificado em SQL-DDL, e correspondente a uma *simplificação* dos conceitos tidos em conta nas etapas 1 e 2.

```
-- -----  
DROP TABLE participa;  
DROP TABLE jogadora;  
DROP TABLE jogo;  
DROP TABLE clube;  
  
-- -----  
  
CREATE TABLE clube (  
    sigla      CHAR(3),          -- Para simplificar a referência de um clube.  
    nome       VARCHAR2(80) CONSTRAINT nn_clube_nome      NOT NULL,  
    localidade VARCHAR2(80) CONSTRAINT nn_clube_distrito NOT NULL,  
--  
    CONSTRAINT pk_clube  
        PRIMARY KEY (sigla)  
--  
    CONSTRAINT un_clube_nome -- Nome é chave candidata.  
        UNIQUE (nome)  
);  
  
-- -----  
  
CREATE TABLE jogo (  
    casa,                -- Clube que joga em casa.  
    visitante,           -- Clube visitante.  
    jornada  NUMBER(2) CONSTRAINT nn_jogo_jornada NOT NULL,  
    espetadores NUMBER(5), -- Pode ser NULL se não existirem dados.  
--  
    CONSTRAINT pk_jogo  
        PRIMARY KEY (casa, visitante),  
--  
    CONSTRAINT fk_jogo_casa  
        FOREIGN KEY (casa)  
        REFERENCES clube(sigla),  
    CONSTRAINT fk_jogo_visitante  
        FOREIGN KEY (visitante)  
        REFERENCES clube(sigla),  
--  
    CONSTRAINT ck_jogo_clubes  
        CHECK (casa <> visitante),  
    CONSTRAINT ck_jogo_jornada  
        CHECK (jornada BETWEEN 1 AND 30),
```

```

CONSTRAINT ck_jogo_espetadores          -- Restrição é satisfeita mesmo
CHECK (espetadores BETWEEN 0 AND 65000) -- que espetadores seja NULL.
);

```

---

```

CREATE TABLE jogadora (
  nif          NUMBER(9),    -- Número de identificação fiscal.
  nome         VARCHAR2(80) CONSTRAINT nn_jogadora_nome      NOT NULL,
  nascimento   NUMBER(4)     CONSTRAINT nn_jogadora_nascimento NOT NULL,
  clube        CONSTRAINT nn_jogadora_clube                  NOT NULL,
  --
  CONSTRAINT pk_jogadora
    PRIMARY KEY (nif),
  --
  CONSTRAINT fk_jogadora_clube
    FOREIGN KEY (clube)
    REFERENCES clube(sigla),
  --
  CONSTRAINT ck_jogadora_nif
    CHECK (nif > 0),
  CONSTRAINT ck_jogadora_nascimento
    CHECK (nascimento BETWEEN 1970 AND 2010)    -- Só o ano de nascimento.
);

```

---

```

CREATE TABLE participa (
  jogadora,
  casa,          -- Clube da casa no jogo em que a jogadora participa.
  visitante,     -- Clube visitante no mesmo jogo.
  golos          NUMBER(2) CONSTRAINT nn_participa_golos NOT NULL,
  --
  CONSTRAINT pk_participa
    PRIMARY KEY (jogadora, casa, visitante),
  --
  CONSTRAINT fk_participa_jogadora
    FOREIGN KEY (jogadora)
    REFERENCES jogadora(nif),
  CONSTRAINT fk_participa_jogo
    FOREIGN KEY (casa, visitante)
    REFERENCES jogo(casa, visitante),
  --
  CONSTRAINT ck_participa_golos
    CHECK (golos >= 0)    -- Só golos na outra baliza (sem autogolos).
);

```

---

As simplificações incluem: a) a remoção do conceito de *época*, passando a assumir-se que todos os jogos acontecem numa mesma época, que continua a ter jornadas; b) a eliminação do conceito de *estádio*; c) a simplificação do conceito de *morada*, passando apenas a interessar a localidade da sede de um clube; d) a redução da hierarquia de *pessoa* a apenas jogadora, com um subconjunto dos atributos, nomeadamente NIF, nome, e ano de nascimento; e) a supressão

de atributos em *clube*, que deixa de ter a data de fundação, e *jogo*, que passa a considerar apenas o número de espetadores; e f) a junção dos conceitos de *participação* em jogo e *golo*, deixando de se saber em que minutos entrou e saiu uma jogadora do jogo, quando foi marcado um golo, e se foi autogolo. Por fim, foi acrescentado um atributo em *clube*, que passou a poder ser identificado pela sua sigla, para além do nome, para simplificar a sua referência.

## Objetivo

Traduzir para *interrogações SQL* os seguintes pedidos de dados:

1. NIF, nome, e idade das jogadoras com apelido Costa, que marcaram pelo menos um golo, pertencentes a clubes em Lisboa, sobre os quais interessa mostrar apenas o nome. O resultado deve vir ordenado pelo nome do clube de forma ascendente, idade da jogadora de forma descendente, e nome e NIF da jogadora de forma ascendente.
2. NIF e nome das jogadoras que *nunca* jogaram em casa, e que jogaram fora de casa (como visitantes) menos de cinco jogos. Inclua no resultado o nome dos respetivos clubes e tenha em conta o caso das jogadoras que ainda não participaram em jogos fora de casa. O resultado deve vir ordenado de forma ascendente pelo nome do clube e NIF da jogadora.
3. NIF e nome das jogadoras nascidas antes do ano 2000 e que marcaram golos em *todos* os jogos que tiveram mais espetadores do que a média de espetadores dos jogos realizados. O resultado deve vir ordenado pelo nome e NIF da jogadora de forma ascendente.
4. Nome do clube com o maior número de golos sofridos para cada jornada. No caso de haver mais do que um clube nessa condição numa jornada, devem ser mostrados todos esses clubes. O valor do maior número de golos sofridos por clubes em cada jornada também deve aparecer no resultado, o qual deve ser ordenado pelo número da jornada e pelo nome do clube de forma ascendente. Dica: comece por uma interrogação que, para cada jogo, mostra duas linhas, uma para cada clube envolvido nesse jogo.
5. Todas as colunas da tabela jogo, com colunas adicionais para os golos marcados pelo clube da casa e pelo clube visitante. A interrogação deve ficar associada a uma *vista* chamada *jogo\_com\_golos* e as colunas extra devem designar-se *golos\_casa* e *golos\_visitante*. Dica: a cláusula FROM pode ter subinterrogações auxiliares, como no seguinte exemplo:  
`SELECT AUX.* FROM (SELECT * FROM jogadora) AUX WHERE (AUX.nif = 123456789).`

## A ter em conta

- Em caso de dificuldade em cumprir a totalidade de um pedido de dados, deve ser escrita uma descrição textual da parte do pedido que pode ser satisfeita e a *interrogação apresentada deve estar coerente com essa redução do pedido original*.
- O resultado de qualquer interrogação *não deve ter linhas repetidas*. Contudo, só deve ser usada a cláusula de remoção de duplicados se esta for mesmo necessária.
- As interrogações devem ser *inteligíveis* e estar bem alinhadas, de acordo com a apresentação dos exemplos das aulas teóricas, e não devem ter mudanças bruscas de linha.

- As interrogações devem ser as mais *simples* possíveis, evitando, por exemplo, a inclusão de tabelas desnecessárias na cláusula FROM ou o uso excessivo de subinterrogações.
- Com exceção do último pedido de dados, não devem ser colocadas subinterrogações auxiliares na cláusula FROM, isto é, só devem ser aí incluídos nomes de tabelas.
- A título meramente sugestivo, podem ser usadas as seguintes *iniciais* para referir de forma abreviada cada uma das tabelas fornecidas: clube C, jogo G, jogadora J, e participa P.

### **Estrutura do relatório**

- O relatório desta etapa do projeto deve ser um *script* SQL contendo as interrogações SQL devidamente anotadas com o número e texto do pedido de dados. É fundamental que o *script* possa ser executado de uma só vez e **sem erros**.
- No início do *script* SQL deve estar a sigla da unidade curricular e o ano letivo, a etapa do projeto, o número do grupo, e o nome, número, e turma dos alunos.

### **Entrega do relatório**

- O relatório deve ser entregue num ficheiro de texto simples com nome SIBD-1920-GXX-E3.SQL, onde XX representa o número do grupo, com dois dígitos (ex. 01).
- A entrega do relatório é feita através da atividade disponível na página de entrada do Moodle da unidade curricular.
- O prazo de entrega do relatório é **29 de novembro de 2019** (sexta-feira).

Bom trabalho na etapa 3 do projeto!