



# PROJETO - Entrega Intermédia

## Bases de Dados

FIFA World Cup Qatar 2022

GRUPO 307

Luís Jesus(202108683), Miguel Leitão(202108851) e Miguel Rocha(202108720)

Novembro, 2022

# Conteúdo

<b>1</b>	<b>Contexto</b>	<b>1</b>
<b>2</b>	<b>Diagrama UML</b>	<b>2</b>
<b>3</b>	<b>Modelo Relacional</b>	<b>3</b>
<b>4</b>	<b>Dependências Funcionais e Formas Normais</b>	<b>4</b>
<b>5</b>	<b>Restrições/Constraints</b>	<b>8</b>
<b>6</b>	<b>Interrogações/Queries</b>	<b>14</b>
<b>7</b>	<b>Gatilhos/Triggers</b>	<b>15</b>
7.1	Gatilho 1 - Dois cartões amarelos geram um cartão vermelho . . .	15
7.2	Gatilho 2 - Limite de cartões amarelos . . . . .	15
7.3	Gatilho 3 - Jogadores da mesma equipa com números repetidos .	15
<b>8</b>	<b>Contribuições, desempenho e dificuldades</b>	<b>16</b>

# 1 Contexto

O Qatar World Cup 2022 é um campeonato mundial no qual 32 equipas (países) se encontram a competir.

Durante este campeonato existem duas fases diferentes: a Fase de Grupos, que acontece de 20 de novembro a 2 de dezembro, e a Fase Eliminatória, que acontece de 3 de dezembro a 18 de dezembro. No final da Fase de Grupos, ficam selecionadas as 16 equipas que passam para a Fase Eliminatória, que disputam os lugares finais e, eventualmente, a taça.

A Fase de Grupos é composta, no total, por 8 grupos que são identificados por uma das seguintes letras: A, B, C, D, E, F, G ou H. Em cada grupo, ocorrem 3 jornadas - que são identificadas por um número - com 2 jogos cada. São apuradas duas equipas de cada grupo para a Fase Eliminatória.

Cada Fase Eliminatória corresponde a uma das seguintes eliminatórias: Oitavos-de-Final, Quartos-de-Final, Meias-Finais ou Final. Ao fim de cada uma das fases é possível saber a posição, no *ranking*, de cada uma das equipas de cada grupo que ficou apurada para as Fases Eliminatórias.

Relativamente a cada jogo, este é realizado numa data específica, tem um resultado e existem duas equipas envolvidas. Cada equipa tem o nome do seu respetivo país, é constituída por vários jogadores (não existem jogadores sem uma equipa associada) e possui um treinador. Em cada jogo, consideramos os seguintes 3 eventos principais: golo (normal, penalti, assistência ou auto-golo), cartão (vermelho ou amarelo) e substituição. Na substituição estão envolvidos dois jogadores: o que sai e o que entra. Todos os eventos possuem um minuto de acontecimento.

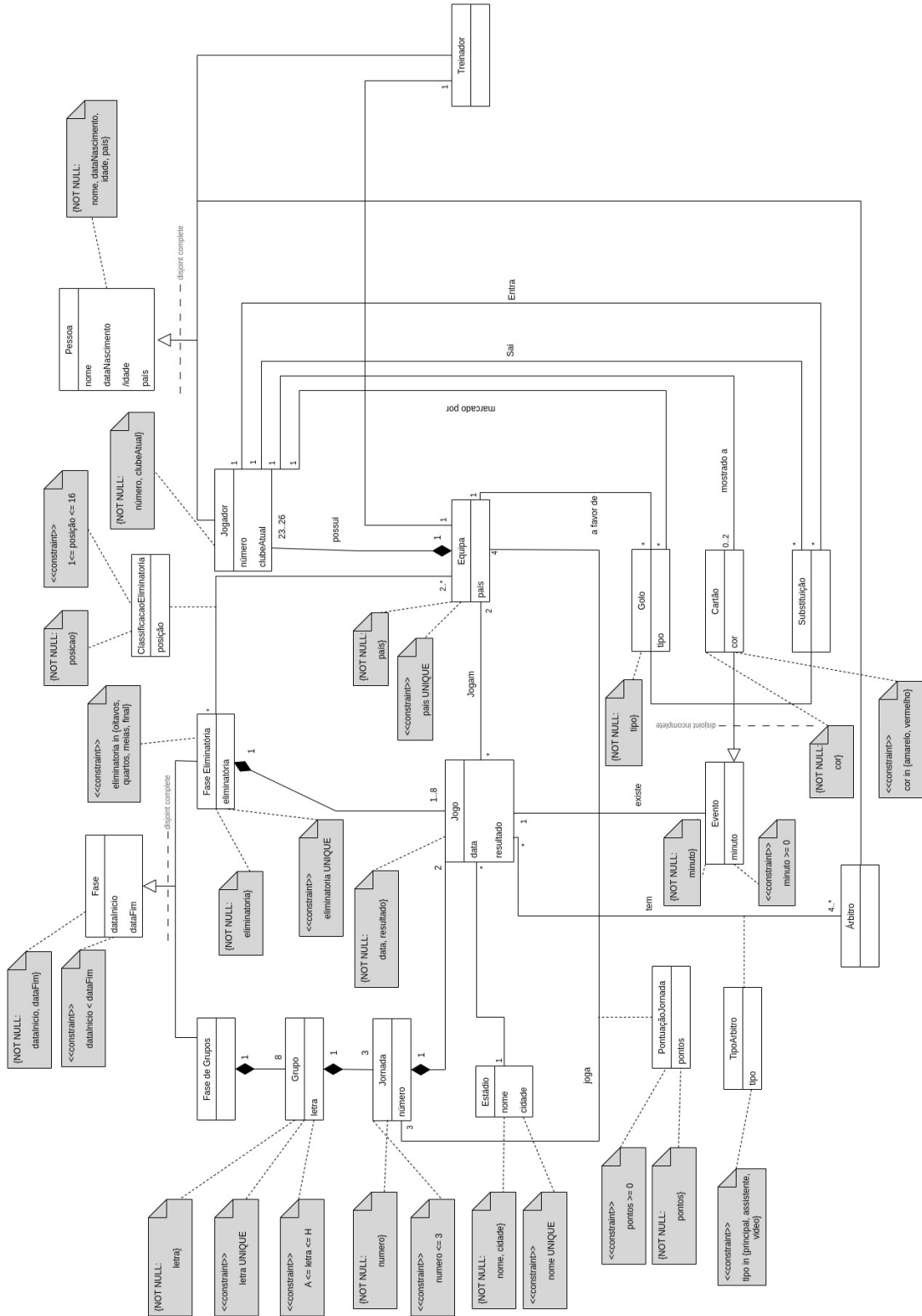
Cada jogo acontece num estádio, que é identificado por um nome e se localiza numa cidade específica do Qatar.

Relativamente a cada jogador, sabe-se que este é uma pessoa, e, portanto, é relevante saber o seu nome, data de nascimento, idade e país. Para além disso, interessa saber o número pelo qual são identificados na equipa e o clube para o qual jogam durante o resto do ano.

Em cada jogo, existe uma equipa de arbitragem composta por 4 ou mais árbitros (pessoas), sendo que cada um deles é de um dos seguintes tipos: principal, adjunto ou vídeo.

Esta base de dados permite saber a pontuação de cada equipa no fim de cada jornada. Relativamente à Fase Eliminatória, como não são contabilizados pontos, é possível obter a posição no *ranking*, de cada equipa, em cada uma das fases eliminatórias, permitindo, na final, obter o *ranking* completo e o vencedor do campeonato. Assim, se consultarmos, por exemplo, a tabela `ClassificacaoEliminatoria`, em que o atributo "eliminatoria" é "oitavos" (oitavos-de-final) teremos várias equipas na mesma posição. No final da competição fica definido o *ranking* final.

## 2 Diagrama UML



### 3 Modelo Relacional

- FaseDeGrupos(Id, dataInicio, dataFim)
- FaseEliminatoria(Id, dataInicio, dataFim, eliminatoria)
- Grupo(letra, faseDeGrupo  $\rightarrow$  FaseDeGrupos.Id)
- Jornada(Id, número, grupo  $\rightarrow$  Grupo.letra)
- Jogo(Id , data, resultado, jornada  $\rightarrow$  Jornada.Id, faseElim  $\rightarrow$  FaseEliminatoria.Id, estadio  $\rightarrow$  Estadio.Id)
- Estadio(Id, nome, cidade)
- Equipa(Id, país)
- Jogam(jogo  $\rightarrow$  Jogo.Id, equipa  $\rightarrow$  Equipa.Id)
- PontuacaoJornada(jornada  $\rightarrow$  Jornada.Id, equipa  $\rightarrow$  Equipa.Id, pontos)
- ClassificacaoEliminatoria(faseElim  $\rightarrow$  FaseEliminatoria.Id, equipa  $\rightarrow$  Equipa.Id, posição)
- Jogador(Id, nome, dataNascimento, idade, país, clubeAtual, número, equipa  $\rightarrow$  Equipa.Id)
- Treinador(Id, nome, dataNascimento, idade, país, equipa  $\rightarrow$  Equipa.id)
- Arbitro (Id, nome, dataNascimento, idade, país)
- TipoArbitro (jogo  $\rightarrow$  Jogo.Id, arbitro  $\rightarrow$  Arbitro.Id, tipo)
- Evento (Id, minuto, jogo  $\rightarrow$  Jogo.Id)
- Golo (evento  $\rightarrow$  Evento.Id, equipa  $\rightarrow$  Equipa.Id, jogador  $\rightarrow$  Jogador.Id, tipo)
- Cartao (evento  $\rightarrow$  Evento.Id, jogador  $\rightarrow$  Jogador.Id, cor)
- Substituicao (evento  $\rightarrow$  Evento.Id, jogadorEntra  $\rightarrow$  Jogador.Id, jogadorSai  $\rightarrow$  Jogador.Id)

## 4 Dependências Funcionais e Formas Normais

**FaseDeGrupos**(Id, dataInicio, dataFim)

- DFs:

- $\text{Id} \rightarrow \text{dataInicio}, \text{dataFim}$
- $\text{dataInicio}, \text{dataFim} \rightarrow \text{Id}$

- Formas Normais:

- BCNF: Sim
- 3NF: Sim

**FaseEliminatoria**(Id, dataInicio, dataFim, eliminatoria)

- DFs:

- $\text{Id} \rightarrow \text{dataInicio}, \text{dataFim}, \text{eliminatoria}$
- $\text{dataInicio}, \text{dataFim} \rightarrow \text{Id}, \text{eliminatoria}$
- $\text{eliminatoria} \rightarrow \text{Id}, \text{dataInicio}, \text{dataFim}$

- Formas Normais:

- BCNF: Sim
- 3NF: Sim

**Grupo**(letra, faseDeGrupo  $\rightarrow$  FaseDeGrupos.Id)

- DFs:

- $\text{letra} \rightarrow \text{faseDeGrupo}$

- Formas Normais:

- BCNF: Sim
- 3NF: Sim

**Jornada**(Id, número, grupo  $\rightarrow$  Grupo.letra)

- DFs:

- $\text{Id} \rightarrow \text{número}, \text{grupo}$

- Formas Normais:

- BCNF: Sim
- 3NF: Sim

**Jogo**(Id , data, resultado, jornada  $\rightarrow$  Jornada.Id, faseElim  $\rightarrow$  FaseEliminatoria.Id, estadio  $\rightarrow$  Estadio.Id)

- DFs:

- $\text{Id} \rightarrow \text{data}, \text{resultado}, \text{jornada}, \text{faseElim}, \text{estadio}$

- Formas Normais:

- BCNF: Sim
- 3NF: Sim

**Estadio**(Id, nome, cidade)

- DFs:
  - $\text{Id} \rightarrow \text{nome, cidade}$
  - $\text{nome} \rightarrow \text{Id, cidade}$

- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Equipa**(Id, país)

- DFs:
  - $\text{Id} \rightarrow \text{país}$
  - $\text{país} \rightarrow \text{Id}$

- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Jogam**(jogo  $\rightarrow$  Jogo.Id, equipa  $\rightarrow$  Equipa.Id)

- DFs: -

**PontuacaoJornada**(jornada  $\rightarrow$  Jornada.Id, equipa  $\rightarrow$  Equipa.Id, pontos)

- DFs:
  - $\text{jornada, equipa} \rightarrow \text{pontos}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**ClassificacaoEliminatoria**(faseElim  $\rightarrow$  FaseEliminatoria.Id, equipa  $\rightarrow$  Equipa.Id, posição)

- DFs:
  - $\text{faseElim, equipa} \rightarrow \text{pontos}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Jogador**(Id, nome, dataNascimento, idade, país, clubeAtual, número, equipa  $\rightarrow$  Equipa.Id)

- DFs:
  - $\text{Id} \rightarrow \text{nome, dataNascimento, idade, país, clubeAtual, número, equipa}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Treinador**(Id, nome, dataNascimento, idade, país, equipa  $\rightarrow$  Equipa.id)

- DFs:

- $\text{Id} \rightarrow \text{nome}, \text{dataNascimento}, \text{idade}, \text{país}, \text{equipa}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Arbitro**(Id, nome, dataNascimento, idade, país)

- DFs:
  - $\text{Id} \rightarrow \text{nome}, \text{dataNascimento}, \text{idade}, \text{país}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**TipoArbitro** (jogo  $\rightarrow$  Jogo.Id, arbitro  $\rightarrow$  Arbitro.Id, tipo)

- DFs:
  - $\text{jogo}, \text{arbitro} \rightarrow \text{tipo}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Evento**(Id, minuto, jogo  $\rightarrow$  Jogo.Id)

- DFs:
  - $\text{Id} \rightarrow \text{minuto}, \text{jogo}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Golo**(evento  $\rightarrow$  Evento.Id, equipa  $\rightarrow$  Equipa.Id, jogador  $\rightarrow$  Jogador.Id)

- DFs:
  - $\text{evento} \rightarrow \text{equipa}, \text{jogador}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Cartao**(evento  $\rightarrow$  Evento.Id, jogador  $\rightarrow$  Jogador.Id, cor)

- DFs:
  - $\text{evento} \rightarrow \text{jogador}, \text{cor}$
- Formas Normais:
  - BCNF: Sim
  - 3NF: Sim

**Substituicao**(evento  $\rightarrow$  Evento.Id, jogadorEntra  $\rightarrow$  Jogador(Id), jogadorSai  $\rightarrow$  Jogador(Id))

- DFs:
  - $\text{evento} \rightarrow \text{jogadorEntra}, \text{jogadorSai}$



- Formas Normais:

- BCNF: Sim
- 3NF: Sim

Uma relação está na Forma Normal de Boyce-Codd (BCNF) se o lado esquerdo de qualquer dependência funcional não trivial for uma superchave da relação (contém uma chave).

Por outro lado, uma relação  $X \rightarrow Y$  na 3ª Forma Normal (3NF) obedece a uma das seguintes “regras”:

- X é superchave;
- Y é um atributo primo (cada elemento de Y faz parte de alguma chave candidata).

No caso das relações da nossa base de dados, todas elas diziam respeito ao caso em que, numa relação do tipo  $X \rightarrow Y$ , X é superchave. Por essa razão, estão todas nas formas normais analisadas e, portanto, não é necessário efetuar decomposições.

## 5 Restrições/Constraints

### Fase de Grupos

- é identificada por um Id:
  - Id PRIMARY KEY;
- a data de início é menor do que a data de fim:
  - CHECK (dataInicio  $\leq$  dataFim);
  - data NOT NULL.

### Fase Eliminatória

- é identificada por um Id:
  - Id PRIMARY KEY.
- a data de início é menor do que a data de fim:
  - CHECK (dataInicio  $\leq$  dataFim);
  - data NOT NULL.
- o nome da eliminatória é unico, não nulo, e é um entre os seguintes: oitavos, quartos, meias e final.
  - eliminatória NOT NULL UNIQUE;
  - eliminatória CHECK (eliminatória IN ('oitavos', 'quartos', 'meias', 'final')).

### Grupo

- é identificado por uma letra:
  - letra PRIMARY KEY;
- a letra só pode ser A, B, C, D, E, F, G ou H, por isso, é única:
  - letra UNIQUE;
  - CHECK (letra IN ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H')).
- faseDeGrupo deverá corresponder ao Id da tabela FaseDeGrupos:
  - faseDeGrupo REFERENCES FaseDeGrupos(Id);
  - faseDeGrupo NOT NULL;
  - faseDeGrupo ON UPDATE CASCADE ON DELETE CASCADE.

### Jornada

- é identificado por um Id não nulo:
  - Id PRIMARY KEY.
- existem três jornadas em cada grupo, numeradas de 1 a 3:
  - numero NOT NULL CHECK (numero  $\geq$  0 AND numero  $\leq$  3).
- grupo deverá corresponder a um Id da tabela Grupo:
  - grupo REFERENCES Grupo(Id).
  - grupo NOT NULL;
  - grupo ON UPDATE CASCADE ON DELETE CASCADE.

### Jogo

- é identificado por um Id não nulo:
  - Id PRIMARY KEY;
- ou jornada é nulo ou faseElim é nulo e a que não for nula deve corresponder a um Id da tabela Jornada ou FaseEliminatoria, respetivamente:
  - CHECK ((jornada IS NOT NULL AND faseElim IS NULL) OR (jornada IS NULL AND faseElim IS NOT NULL));
  - jornada REFERENCES Jornada(Id);
  - jornada ON UPDATE CASCADE ON DELETE CASCADE;
  - faseElim REFERENCES FaseEliminatoria(Id);
  - faseElim ON UPDATE CASCADE ON DELETE CASCADE.
- estadio deve corresponder a um Id da tabela Estadio e é não nulo:
  - estadio REFERENCES Estadio(Id);
  - estadio ON UPDATE CASCADE ON DELETE CASCADE;
  - estadio NOT NULL.
- data não pode ser nula:
  - data NOT NULL;

### Estádio

- é identificado por um Id não nulo:
  - Id PRIMARY KEY;
- nome e cidade não podem ser nulos:
  - nome NOT NULL;
  - cidade NOT NULL.
- cidade é única:
  - cidade UNIQUE.

### Equipa

- é identificado por um Id não nulo:
  - Id PRIMARY KEY;
- país é único e não nulo:
  - pais UNIQUE;
  - pais NOT NULL.

### Jogam

- relação identificada por um jogo e uma equipa, que não podem ser nulos:
  - jogo, equipa PRIMARY KEY;
- jogo deve corresponder a um Id da tabela Jogo:
  - jogo REFERENCES Jogo(Id);
  - jogo ON UPDATE CASCADE ON DELETE CASCADE.
- equipa deve corresponder a um Id da tabela Equipa:
  - equipa REFERENCES Equipa(Id);

- equipa ON UPDATE CASCADE ON DELETE CASCADE.

#### PontuacaoJornada

- identificada por uma jornada e uma equipa não nulos:
  - jornada, equipa PRIMARY KEY;
- jornada deve corresponder a um Id da Jornada:
  - jornada REFERENCES Jornada(Id);
  - jornada ON UPDATE CASCADE ON DELETE CASCADE.
- equipa deve corresponder a um Id da tabela Equipa:
  - equipa REFERENCES Equipa(Id);
  - equipa ON UPDATE CASCADE ON DELETE CASCADE.
- pontos devem ser maiores ou iguais a 0:
  - CHECK (pontos  $\geq$  0);
  - pontos NOT NULL;

#### ClassificacaoEliminatoria

- identificada por uma fase eliminatória e uma equipa não nulos:
  - faseElim, equipa PRIMARY KEY;
- faseElim deve corresponder a um Id da tabela FaseEliminatoria:
  - faseElim REFERENCES FaseEliminatoria(Id);
  - faseElim ON UPDATE CASCADE ON DELETE CASCADE.
- equipa deve corresponder a um Id da tabela Equipa:
  - equipa REFERENCES Equipa(Id);
  - equipa ON UPDATE CASCADE ON DELETE CASCADE.
- posição deve estar entre 1 e 16 (número de equipas apuradas depois da fase de grupos):
  - CHECK (posicao  $\leq$  16 AND posicao  $\geq$  1);
  - posicao NOT NULL.

#### Jogador

- identificado por um Id não nulo:
  - Id PRIMARY KEY;
- nome, dataNascimento, país, clubeAtual, número e equipa não podem ser nulos:
  - nome NOT NULL;
  - dataNascimento NOT NULL;
  - idade NOT NULL;
  - país NOT NULL;
  - clubeAtual NOT NULL;
  - numero NOT NULL;
  - equipa NOT NULL.
- equipa deve corresponder a um Id da tabela Equipa:

- equipa REFERENCES Equipa(Id);
- equipa ON UPDATE CASCADE ON DELETE CASCADE.

#### Treinador

- identificada por um Id não nulo:
  - Id PRIMARY KEY;
- nome, dataNascimento, país e equipa não podem ser nulos:
  - nome NOT NULL;
  - dataNascimento NOT NULL;
  - idade NOT NULL;
  - país NOT NULL;
  - equipa NOT NULL.
- equipa deve corresponder a um Id da tabela Equipa:
  - equipa REFERENCES Equipa(Id);
  - equipa ON UPDATE CASCADE ON DELETE CASCADE.

#### Árbitro

- identificada por um Id não nulo:
  - Id PRIMARY KEY;
- nome, dataNascimento e país não podem ser nulos:
  - nome NOT NULL;
  - dataNascimento NOT NULL;
  - idade NOT NULL;
  - país NOT NULL.

#### TipoArbitro

- identificado por um jogo e um arbitro (Ids), ambos não nulos:
  - jogo, arbitro PRIMARY KEY;
- o tipo de árbitro pode ser principal, assistente ou vídeo:
  - CHECK (tipo ='principal' OR tipo ='adjunto' OR tipo ='video');
  - tipo NOT NULL.
- jogo e arbitro correspondem a um Id das tabelas Jogo e Arbitro, respetivamente:
  - jogo REFERENCES Jogo(Id);
  - jogo ON UPDATE CASCADE ON DELETE CASCADE;
  - arbitro REFERENCES Arbitro(Id);
  - arbitro ON UPDATE CASCADE ON DELETE CASCADE.

#### Evento

- identificado por um Id não nulo:
  - Id PRIMARY KEY;
- minuto é superior ou igual a 0 e não nulo:
  - CHECK (minuto  $\geq$  0).

- minuto NOT NULL.
- jogo não pode ser nulo e corresponde a um Id da tabela Jogo:
  - jogo REFERENCES Jogo(Id);
  - jogo NOT NULL;
  - jogo ON UPDATE CASCADE ON DELETE CASCADE.

#### Golo

- identificado por um evento:
  - evento PRIMARY KEY;
- evento corresponde a um Id da tabela Evento:
  - evento REFERENCES Evento(Id);
  - evento ON UPDATE CASCADE ON DELETE CASCADE.
- equipa e jogador são não nulos:
  - equipa NOT NULL;
  - jogador NOT NULL.
- tipo tem de ser normal, penalti, assistencia ou auto-golo:
  - CHECK (tipo IN ('normal', 'penalti', 'auto-golo', 'assistencia'));
  - tipo NOT NULL.
- equipa e jogador correspondem a um Id da tabela Equipa e Jogador, respetivamente:
  - equipa REFERENCES Equipa(Id);
  - equipa ON UPDATE CASCADE ON DELETE CASCADE;
  - jogador REFERENCES Jogador(Id).
  - jogador ON UPDATE CASCADE ON DELETE CASCADE.

#### Cartao

- identificado por um evento:
  - evento PRIMARY KEY;
- evento corresponde a um Id da tabela Evento:
  - evento REFERENCES Evento(Id);
  - evento ON UPDATE CASCADE ON DELETE CASCADE.
- cor deve ser amarelo ou vermelho, e jogador é não nulo:
  - CHECK (cor ='amarelo' OR cor ='vermelho');
  - cor NOT NULL;
  - jogador NOT NULL.
- jogador corresponde a um Id da tabela Jogador, respetivamente:
  - jogador REFERENCES Jogador(Id);
  - jogador ON UPDATE CASCADE ON DELETE CASCADE.

#### Substituicao

- identificado por um evento:
  - evento PRIMARY KEY;

- evento corresponde a um Id da tabela Evento:
  - evento REFERENCES Evento(Id).
  - evento ON UPDATE CASCADE ON DELETE CASCADE.
- jogadorEntra e jogadorSai são não nulos e correspondem cada um a um Id da tabela Jogador:
  - jogadorEntra NOT NULL;
  - jogadorSai NOT NULL;
  - jogadorEntra REFERENCES Jogador(Id);
  - jogadorEntra ON UPDATE CASCADE ON DELETE CASCADE;
  - jogadorSai REFERENCES Jogador(Id);
  - jogadorSai ON UPDATE CASCADE ON DELETE CASCADE.

## 6 Interrogações/Queries

Tendo em conta que a base de dados em questão, possui diferentes informações sobre diferentes tópicos, escolhemos as seguintes 10 interrogações que pensamos ser mais interessantes de forma a que, em simultâneo, fossem usadas diferentes keywords de SQL:

- 1) Listar nome, idade e número de todos os jogadores da equipa Portugal;
- 2) Listar todos os jogadores - nome, data de nascimento, idade e clube atual - com mais de 35 anos;
- 3) Listar todos os árbitros - nome, idade e tipo - que arbitraram jogos no estádio X;
- 4) Listar equipas (nome da equipa e jogador) que, em alguma altura da competição, viram um jogador seu a sair do campo porque teve um cartão vermelho;
- 5) Listar a primeira equipa de cada um dos grupos no final da fase de grupos (mostrar grupo, equipa e pontos);
- 6) Mostrar a média de golos por jogo de cada equipa ao longo da fase de grupos;
- 7) Indicar o nome do melhor marcador de cada equipa e o número de golos;
- 8) Listar todos os jogos (id e resultado) em que apenas uma das equipas marcou;
- 9) Listar todos os eventos do jogo com mais substituições;
- 10) Listar nome, idade e país dos treinadores cujas equipas registaram, em soma ao longo da competição, mais de 4 cartões.

**Nota:** Em algumas interrogações utilizaram-se Views, não só para as tornas mais diversas, como também para facilitar a manipulação dos dados.



## **7 Gatilhos/Triggers**

Tentamos implementar 3 gatilhos/triggers que consideramos bastante úteis e pertinentes de forma a "fortalecer" a base dados, permitindo uma melhor manutenção da base de dados:

### **7.1 Gatilho 1 - Dois cartões amarelos geram um cartão vermelho**

Decidimos implementar este gatilho por ser uma das regras fundamentais do futebol. Quando um mesmo jogador recebe, no mesmo jogo dois cartões amarelos, é-lhe imediatamente mostrado um cartão vermelho, ficando então com os 3 cartões (dois amarelos e um vermelho).

### **7.2 Gatilho 2 - Limite de cartões amarelos**

Assim que um jogador recebe dois cartões amarelos, ao ser-lhe mostrado um cartão vermelho, ele é expulso do jogo e portanto não poderá receber mais cartões.

Deste modo, a inserção de um novo cartão amarelo num mesmo jogo para o mesmo jogador, tendo ele já recebido dois cartões amarelos nesse jogo, deve ser inválida e, portanto, deverá ser impedido e sinalizado como uma restrição.

### **7.3 Gatilho 3 - Jogadores da mesma equipa com números repetidos**

Numa mesma equipa de futebol, que é constituída por 23 jogadores, dois futebolistas não podem ter o mesmo número.

Neste sentido, ativando o gatilho, sempre que se tentar povoar uma equipa com um jogador cujo número atribuído já existe na base de dados, esta inserção é imediatamente impedida e resulta numa mensagem de erro.

## 8 Contribuições, desempenho e dificuldades

O grupo conseguiu organizar-se bem e distribuir as diferentes o mais uniformemente possível de modo a que todos conseguissem fazer um pouco de tudo: UML, modelo relacional, análise de dependências funcionais, ficheiros .sql, etc...

Neste sentido, parece-nos justo que a percentagem da contribuição fique uniformemente distribuída pelos três uma vez que todos nos empenhamos de forma semelhante:

- Luís Jesus: 33.33%;
- Miguel Leitão: 33.33%;
- Miguel Rocha: 33.33%.

O trabalho de grupo, até ao momento, correu bem ainda que tenhamos encontrado algumas dificuldades, principalmente na construção do modelo UML e posterior criação das tabelas. Devido a algumas distrações, estávamos a obter erros de sintaxe no SQL e não é muito fácil fazer *debugging*. Para além disso, ao passar para o modelo relacional, foi necessário voltar atrás e melhorar o UML, pois verificamos incompatibilidades, incongruências e coisas que já não faziam muito sentido.

Para além disso, embora não tenha sido propriamente uma dificuldade, a parte de povoar a base de dados foi muito desgastante porque o volume de dados a colocar manualmente é relativamente grande e exige alguma concentração para evitar erros. Por essa razão, a base de dados não foi povoada na totalidade, mas sim com uma amostra suficiente, tal como sugerido pelos professores regentes.

Era um dos nossos objetivos fazer com que os pontos na tabela da pontuação das jornadas fossem gerados automaticamente, contudo, devido a falta de tempo, não chegou a ser possível fazê-lo.

**NOTA:** como não existem dados do mundial deste ano, uma vez que ainda não começou, a base de dados foi povoada com alguns dados do mundial de 2018. Para além disso, dado o grande volume de dados, não foi possível povoar totalmente a base de dados e, não existem, por exemplo, jogos das eliminatórias povoados. Ainda assim, isto não compromete a compreensão da base de dados.