

# BD - FIRST REPORT

## Campeonato Hoquei em Patins da 1ª Divisao

Valter Castro **up201706546@up.pt**

Rodrigo Martins **up202008868@up.pt**

Maria Carneiro **up202107798@up.pt**

### Descricao

Este projeto baseia-se na gestao de um servico para um **campeonato de hoquei em patins**.

O **campeonato** de hoquei em patins tem um nome, uma **epoca** e uma divisao. E constituido por uma fase **regular** e uma fase de **playoff** de apuramento para decidir o campeao. A fase regular tem 26 jornadas, cada uma com 7 jogos entre uma equipa visitante e uma equipa visitada. Em cada jornada sao registados os marcadores dos golos. Uma epoca tem uma data de abertura e uma data final.

Cada **equipa** tem um nome, os respetivos pontos do campeonato, um **clube** e a sua composicao pode variar em cada epoca. Todos os clubes pertencem a uma liga e teem um nome e uma morada. As 8 equipas com mais pontos na fase regular disputam o playoff. Das restantes 6 equipas, as 3 melhores posicionadas mantem-se no campeonato da 1ª divisao e as 3 com menor pontuacao descem para o campeonato da segunda divisao. No playoff, as equipas defrontam-se em jogos de varias maos, sendo que nos quartos de final, os jogos sao disputados num esquema “melhor de 3 jogos”(vence a primeira equipa a ganhar 2 jogos). As meias-finais e final, disputam-se num esquema “melhor de 5 jogos”(vence a primeira equipa a ganhar 3 jogos).

Normalmente, uma equipa e constituída por 1 **treinador** principal, 1 **treinador adjunto**, 1 **delegado** e respetivos jogadores. Cada treinador pode convocar para a ficha de jogo 10 jogadores. Nesta e sabido a titularidade de cada um e respetivo numero de camisola. Em pista, podem estar simultaneamente 5 jogadores em cada equipa(4 jogadores em campo e 1 guarda redes).

Os **jogadores** teem nome, nacionalidade e idade. O periodo de inscricao de jogadores e feito ate final de dezembro. Se um jogador mudar de equipa durante este periodo so pode voltar a competir oficialmente a partir de janeiro.

Um **Arbitro** tem um nome, uma associacao e tem uma categoria de arbitro internacional ou A1.

Cada **jogo** tem uma equipa visitante e uma equipa visitada. O jogo tem tambem um identificador e e referente a um campeonato. O mesmo e arbitrado por 2



**Epoca**(*ID*, *dataInicio*, *dataFim*)

1. {*ID*}->{*dataInicio*, *dataFim*}

**Equipa**(*ID*, *epocaID*->*Epoca*, *clubeID*->*Clube*, *nome*, *pontos*)

1. {*ID*}->{*epocaID*->*Epoca*, *clubeID*->*Clube*, *nome*, *pontos*}
2. {*clubeID*->*Clube*}->{*ID*, *nome*, *epocaID*->*Epoca*, *pontos*}

**Campeonato**(*ID*, *epocaID*->*Epoca*, *divisao*, *nome*)

1. {*ID*}->{*epocaID*->*Epoca*, *divisao*, *nome*}

**Jogo**(*ID*, *visitanteID*->*Equipa*, *visitadaID*->*Equipa*, *vencedorID*->*Equipa*, *campeonatoID*->*Campeonato*, *pavilhaoID*->*Pavilhao*, *numeroJogo*, *tempoInicio*, *tempoFim*, *categoria*)

1. {*ID*}->{*visitanteID*->*Equipa*, *visitadaID*->*Equipa*, *vencedorID*->*Equipa*, *campeonatoID*->*Campeonato*, *pavilhaoID*->*Pavilhao*, *numeroJogo*, *tempoInicio*, *tempoFim*, *categoria*}

**Regular**(*campeonatoID*->*Campeonato*)

1. {*campeonatoID*->*Campeonato*}->{}

**PlayOff**(*campeonatoID*->*Campeonato*)

1. {*campeonatoID*->*Campeonato*}->{}

**Pessoa**(*ID*, *nome*)

1. {*ID*}->{*nome*}

**Cronometrista**(*pessoaID*->*Pessoa*)

1. {*pessoaID*->*Pessoa*}->{}

**TreinadorAdjunto**(*pessoaID*->*Pessoa*, *equipaID*->*Equipa*)

1. {*pessoaID*->*Pessoa*}->{*equipaID*->*Equipa*}

**Treinador**(*pessoaID*->*Pessoa*, *equipaID*->*Equipa*)

1. {*pessoaID*->*Pessoa*}->{*equipaID*->*Equipa*}

**DelegadoTecnico**(*pessoaID*->*Pessoa*)

1. {*pessoaID*->*Pessoa*}->{}

**Delegado**(*pessoaID*->*Pessoa*, *equipaID*->*Equipa*)

1. {*pessoaID*->*Pessoa*}->{*equipaID*->*Equipa*}

**Jogador**(*pessoaID*->*Pessoa*, *equipaID*->*Equipa*, *epocaID*->*Epoca*, *nacionalidade*, *idade*)

1. {*pessoaID*->*Pessoa*}->{*equipaID*->*Equipa*, *epocaID*->*Epoca*, *nacionalidade*, *idade*}

**Arbitro**(*peessoaID*->*Pessoa*, associacao, categoria)

1. {*peessoaID*->*Pessoa*}->{associacao, categoria}

**PessoaJogo**(*jogoID*->*Jogo*, *arbitroID*->*Arbitro*, *cronometristaID*->*Cronometrista*, *delegadoTecnicoID*->*DelegadoTecnico*)

**Pavilhao**(*ID*, nome, codigoPostal, cidade)

1. {*ID*}->{nome, codigoPostal, cidade}

**FichaJogo**(*ID*, *jogoID*->*Jogo*, *jogadorID*->*Jogador*, titularidade, numero-Camisola)

1. {*ID*}->{*jogoID*->*Jogo*, *jogadorID*->*Jogador*, titularidade, numero-Camisola}

**Evento**(*ID*, tempo, *jogoID*->*Jogo*)

1. {*ID*}->{tempo, *jogoID*->*Jogo*}

**Golo**(*eventoID*->*Evento*, *jogadorID*->*Jogador*)

1. {*eventoID*->*Evento*, *jogadorID*->*Jogador*}->{}

**Falta**(*eventoID*->*Evento*, *arbitroID*->*arbitro*, *jogadorID*->*Jogador*, gravidade)

1. {*eventoID*->*Evento*}->{*arbitroID*->*arbitro*, *jogadorID*->*Jogador*, gravidade}

## Formas Normais

Para identificar a 3ª **Forma Normal**, sera necessario assegurar o cumprimento da regra de **nao-transitividade**. Caso esta regra seja quebrada, tambem sera a Forma Normal de Boyce-Codd, visto esta se tratar de uma versao ligeiramente mais restrita da anterior. No modelo proposto, não existira quebra da 3ª Forma Normal, nem da Forma Normal de Boyce-Codd na medida em que o lado esquerdo de cada dependencia e uma **super-key** do esquema relacional – condicao suficiente para cumprir ambas.

## Restricoes

Para assegurar uma boa **manutencao** da base de dados, assim como fornecer **seguranca** adicional ao utilizador, recorreu-se ao uso de restricoes na producao de varias classes, nomeadamente do tipo chave, de **integridade referencial**, **CHECK** E **NOT NULL**.

Nas ocasioes onde a restricao **NOT NULL** e associada a um atributo, manifesta-se a obrigatoriedade da existencia deste mesmo atributo para a formacao minima da classe. No seguinte modelo a restricao **NOT NULL** e utilizada:

- O atributo nome de **Clube** e importante para a sua identificacao.
- O atributo data tem uma profunda ligacao na existencia de uma **Epoca** sendo necessario a sua implementacao.
- O atributo nome de **Equipa** e importante para a sua identificacao.
- O atributo nome de **Campeonato** e importante para a sua identificacao.
- Os atributos tempoInicio e tempoFim de **Jogo** determinam a sua duracao sendo necessarios. O atributo categoria permite saber que campeonato se trata.
- Os atributos titularidade e numeroCamisola sao essenciais para a **Ficha-Jogo** de um jogador.
- O atributo tempo de **Evento** carateriza este mesmo.
- O atributo gravidade permite saber que tipo de **Falta** e dada.
- O nome de **Pessoa** e necessario para a identificacao da mesma.
- A nacionalidade e idade de um **Jogador** sao necessarios para fins contratuais , sendo que, estes nao estando presentes no esquema , terao integracao facilitada numa futura extensao do mesmo.
- A categoria de um **Arbitro** permite saber num jogo qual o tipo do arbitro.
- O nome, codigo postal e cidade, dao a saber a localizacao de um **Pavilhao**, sendo assim, necessarios.
- Todos os atributos designados para foreign keys das tabelas são NOT NULL com excecao de vencedor em **Jogo**, dado que pode haver um empate.

A restricao **CHECK**:

- O atributo pontos de **Equipa** tem um limite entre 0 e 72, sendo absurdo qualquer outro valor no contexto abordado.
- O atributo categoria de **Jogo** so podera ser jornada, quartos-final, meia-final ou final.
- O atributo titularidade de **FichaJogo** so pode tomar como valor banco ou titular.
- O atributo gravidade de **Falta** so podera ser falta de equipa, azul ou vermelho.
- O atributo categoria de **Arbitro** so sera internacional ou A1.
- O atributo codigoPostal de **Pavilhao** esta limitado a 8 carateres que e o limite de um codigo postal.

A restricao **UNIQUE** neste modelo relacional nao foi determinada, como necessaria, para alem das PKs.

Relativamente a restricoes de **integridade referencial**, foram aplicadas chaves estrangeiras a classes intrinsecamente relacionadas com outras. Como exemplo , um **Campeonato** e **Regular** ou **PlayOff** , sendo necessario uma chave estrangeira apontada nos IDs de regular e playoff para campeonato.

Este raciocinio foi aplicado para as restantes classes que dispoem destas restricoes, resultando num modelo bastante trivial de interpretar.

## Interrogacoes

Foram adicionadas 10 interrogacoes com o objetivo de demonstrar a versatilidade das interrogacoes em SQL.

### 1. Qual é o top 10 de jogadores que marcaram mais golos?

Nesta interrogacao, é usado a funcao **COUNT** para saber o numero de golos marcados pelos 10 melhores marcadores.

### 2. Jogadores que marcaram no minimo um golo e que nao tiveram faltas?

Nesta interrogacao e de salientar o uso de **NATURAL JOIN** para identificar quais os nomes dos jogadores, e do **EXISTS**, para saber quais deles marcaram golos e nao cometeram faltas no campeonato.

### 3. Qual o clube com mais faltas?

Nesta interrogacao e sabido qual o clube que cometeu mais faltas durante o campeonato.

### 4. Quais os jogadores mais novos que marcaram golos nesta epoca?

Nesta interrogacao e identificado quantos golos marcou o jogador mais novo da epoca.

### 5. Qual o jogador espanhol com mais faltas?

Nesta interrogacao e de salientar o uso do operador **LIKE** para identificar qual o jogador espanhol que cometeu mais faltas.

### 6. Quantos eventos houve no jogo com id = 1?

Nesta interrogacao foi utilizado o **HAVING** para filtrar quantos eventos ocorreram no jogo com id = 1.

### 7. Media de golos marcados por jogador?

Nesta interrogacao é calculada a media(**AVG**) de golos por jogador no campeonato com um **ROUND** de 2 casas decimais.

### 8. Qual o golo mais tardio da equipa que marcou mais golos?

Nesta interrogacao é demonstrado qual o golo mais tardio da equipa que marcou mais golos do campeonato.

**9. Qual é o tempo fim e tempo início do jogo que teve mais faltas no Pavilhao Casablanca?**

Nesta interrogacao é usado o **REPLACE** para saber qual o tempo de inicio/fim do jogo com mais faltas no pavilhao Casablanca.

**10. Qual e o resultado do jogo com id = 2?**

Nesta interrogacao é usado o **UNION** para saber qual resultado do jogo com id = 2 no campeonato.

## **Gatilhos**

Foram adicionados 3 gatilhos. Dado que a complexidade da base de dados nao permite a uma diversidade consideravel de gatilhos, a sua producao foi feita para fins academicos.

**1. IdadeJogador**

Antes de ser criado um Jogador(**BEFORE INSERT**), e verificado se a idade do mesmo e inferior a 18. Caso esta condicao se verifique, é abortado o insert lancando um aviso “**Invalid age**”.

**2. AdicionaPontosVencedor**

Apos ser criado um Jogo(**AFTER INSERT**), e verificado se houve um vencedor. Caso se verifique, sao adicionados 3 pontos a respetiva equipa.

**3. RemoveEquipa**

Apos ser eliminado um Clube(**AFTER DELETE**), sao eliminadas as respetivas equipas. Para fins academicos, este trigger foi feito para simular um **ON DELETE CASCADE**, para tal foi necessario retirar temporariamente a verificacao de integridade referencial **PRAGMA foreign\_keys=off;**.

## **Instruções de execução**

Estes passos deverão ser seguidos escrupulosamente, para evitar comportamento inesperado por parte da base de dados. \* Executar o ficheiro criar.sql. \* Executar o ficheiro povoar.sql.

Correr as interrogações e gatilhos como pretendido, que se encontram nas pastas queries e triggers, respetivamente, no diretório principal da entrega.