

Licenciatura em Engenharia Informática e Computação

Bases de Dados

Índice

| | |
|--|---|
| 1. Descrição do projeto | 3 |
| 2. Diagrama UML | 4 |
| 3. Definição do Esquema Relacional | 5 |
| 4. Análise de Dependências Funcionais e Formas Normais | 6 |
| 5. Lista e forma de implementação das restrições | 8 |

1. Descrição do projeto

1ª Liga Portuguesa de Basquetebol

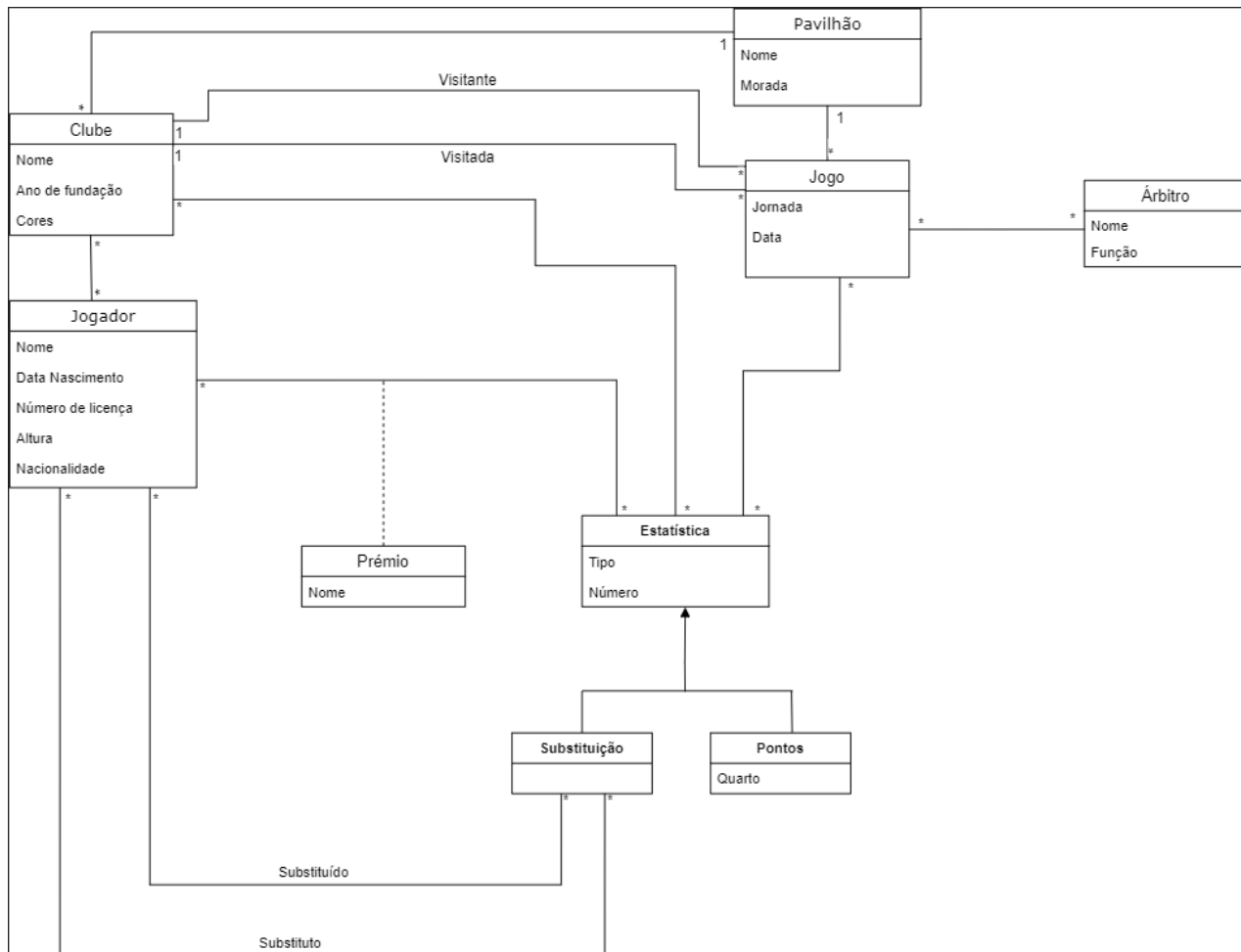
Neste projeto temos como objetivo armazenar numa base de dados várias informações relativas à época 2020/21 da 1ª Liga Portuguesa de Basquetebol.

A liga é composta por vários clubes, dos quais procuramos saber quais os jogadores inscritos para a época em questão, o seu total de vitórias e derrotas e a sua consequente classificação final. Dos jogadores de cada equipa é necessário saber qual o seu nome, número de licença, altura e nacionalidade.

Ao longo da época foram realizados diversos jogos sobre os quais guardamos a jornada e a data em que se realizou, quais as equipas o disputaram, qual a equipa de arbitragem envolvida e em que pavilhão se realizou. A todos os jogos está associada uma estatística que envolve pontos marcados, assistências, ressaltos, bloqueios, perdas e roubos de bola, faltas, valorização e tempo de jogo de cada jogador. Quanto aos pontos guardamos ainda para cada jogo o total de pontos em cada um dos quatro períodos da partida.

No final da temporada foram atribuídos prémios individuais tais como o prémio de melhor marcador, melhor defesa e jogador mais valioso.

2. Diagrama UML



3. Definição do Esquema Relacional

Primary key – laranja

Foreign key – amarelo

Jogador(id_Jogador, Nome, Data de nascimento, Número de Licença, Altura, Nacionalidade)

Clube(id_Clube, Nome, Ano de fundação, Cores, id_Pavilhão->Pavilhão)

Pavilhão(id_Pavilhão, Nome, Morada)

Jogo(id_Jogo, Jornada, Data, id_Pavilhão->Pavilhão, id_ClubeVisitado->Clube, id_ClubeVisitante->Clube)

Árbitro(id_Árbitro, Nome, Função)

Estatística(id_Estatística, Tipo, Número)

ÁrbitrosnoJogo(id_Árbitro->Árbitro, id_Jogo->Jogo)

JogadoresNoClube(id_Jogador->Jogador, id_Clube->Clube)

EstatísticaJogador(id_Jogador->Jogador, id_Estatística->Estatística)

EstatísticaClube(id_Estatística->Estatística, id_Clube->Clube)

Prémio(id_Jogador->Jogador, id_Estatística->Estatística, Nome)

EstatísticaSubstituição(id_Estatística->Estatística, idJogadorEntra->Jogador, idJogadorSai->Jogador)

EstatísticaPontos(id_Estatística->Estatística, Quarto)

EstatísticaJogo(id_Jogo->Jogo, id_Estatística->Estatística)

4. Análise de Dependências Funcionais e Formas Normais

Dependências Funcionais:

Jogador {id_Jogador -> Nome

id_Jogador -> Data de nascimento

id_Jogador -> Altura

id_Jogador -> Nacionalidade}

Clube {id_Clube -> Nome

id_Clube -> Data de fundação

id_Clube -> Cores

id_Clube -> id_Pavilhão}

Pavilhão {id_Pavilhão -> Nome

id_Pavilhão -> Morada}

Jogo {id_Jogo -> Jornada

id_Jogo -> Data

id_Jogo -> id_Pavilhão

id_Jogo -> id_Estatística

id_Jogo -> id_ClubeVisitado

id_Jogo -> id_ClubeVisitante

Jornada, Data -> id_Jogo}

Árbitro {id_Árbitro -> Nome

id_Árbitro -> Função}

Estatística {id_Estatística -> Tipo

id_Estatística -> Número}

Prémio {id_Jogador, id_Estatística -> Nome}

EstatísticaSubstituição {id_Estatística -> id_JogadorEntra

id_Estatística -> id_JogadorSai

id_JogadorEntra, id_JogadorSai -> id_Estatística}

EstatísticaPontos {id_Estatística -> Quarto}

ÁrbitrosnoJogo JogadoresNoClube EstatísticaJogador EstatísticaClube
não tem dependências funcionais.

Análise das Formas Normais:

Como seguimos um processo de criação básico, desde a criação do modelo UML até ao modelo relacional, as seguintes dependências não violam a Forma de Boyce-Codd nem a 3º Forma Normal:

- Jogador
- Clube
- Pavilhão
- Árbitro
- Estatística
- Prémio
- EstatísticaPontos

Porque as dependências são triviais ou têm uma chave primária do lado esquerdo.

As dependências funcionais *EstatísticaSubstituição* e *Jogo* violam a forma de Boyce-Codd, porque nem todas as dependências são triviais ou têm uma chave primária no lado esquerdo, mas não violam a 3º Forma Normal porque o lado direito consiste apenas em atributos primários.

5. Lista e forma de implementação das restrições

A relação Jogador contém um id que identifica cada jogador, logo não podem existir dois jogadores com o mesmo id, sendo aplicada a restrição de PRIMARY KEY, cada jogador tem também um nome, data de nascimento, altura e nacionalidade, sendo todos os atributos necessários para a tabela, todos os atributos são NOT NULL.

A relação Clube contém um id único para cada clube, tal como o seu nome que também é diferente dos outros clubes, sendo aplicada a restrição de PRIMARY KEY. Cada clube tem também uma data de fundação, as cores dos seus equipamentos, todos estes atributos são NOT NULL, e um id do pavilhão onde joga obtido por uma FOREIGN KEY.

A relação Pavilhão tem um id para os identificar dos outros pavilhões, pois um id só funciona para um único pavilhão, sendo aplicada a restrição de PRIMARY KEY, tem também o nome e a morada do mesmo, sendo que é obrigatório ter um nome e morada, estes atributos são NOT NULL.

A relação Jogo tem um id único para cada jogo sendo aplicada a restrição de PRIMARY KEY, tem também o número da jornada e a data em que se realizou o jogo, para definir esta relação estes atributos têm que ser NOT NULL. Tem também o id do pavilhão em que o jogo foi realizado e os id's dos clubes sendo um do visitado e o outro do visitante, obtidos por FOREIGN KEY.

A relação Árbitro contém um id para cada árbitro sendo PRIMARY KEY, tem um nome e uma em oito funções ('Comissário', 'Árbitro Principal', 'Árbitro Auxiliar 1', 'Árbitro Auxiliar 2', 'Marcador', 'Marcador Auxiliar', 'Cronometrista', 'Operador 24 segundos'), sendo verificada qual das oito funções é através de uma restrição CHECK.

A relação Estatística tem um id único através de PRIMARY KEY, tem um tipo verificado através de uma restrição CHECK e um número inteiro NOT NULL cujo DEFAULT é 0.

A relação ÁrbitrosnoJogo contém dois atributos id's, sendo um FOREIGN KEY da relação Árbitro e o outro FOREIGN KEY da relação Jogo, estes dois ids formam a PRIMARY KEY da relação.

A relação JogadoresnoClube tem dois atributos id's que são FOREIGN KEYS das relações Jogador e Clube que juntos formam a PRIMARY KEY da relação.

A relação EstatísticaJogador tem dois id's sendo um FOREIGN KEY de Jogador e outro FOREIGN KEY de Estatística formando os dois a PRIMARY KEY da relação.

A relação EstatísticaJogo tem uma PRIMARY KEY formada por duas FOREIGN KEYS das relações Jogo e Estatística.

A relação EstatísticaClube tem dois atributos id que são FOREIGN KEY das relações Clube e Estatística que juntos formam a PRIMARY KEY da relação.

A relação Prémio tem dois id's FOREIGN KEY das relações Jogador e Estatística que formam a PRIMARY KEY, tem também um atributo que indica o nome do prémio sendo verificado através da restrição CHECK.

A relação EstatísticaSubstituição tem um id PRIMARY KEY que é FOREIGN KEY da relação Estatística, tem também dois id's, sendo um id do Jogador que entra em jogo, conseguido através de uma FOREIGN KEY, e um id do jogador que sai, obtido também através de uma FOREIGN KEY.

A relação EstatísticaPontos tem um id PRIMARY KEY obtido por uma FOREIGN KEY na relação Estatística e um número inteiro que indica o quarto, logo este valor tem que ser NOT NULL.