Faculdade de Engenharia da Universidade do Porto

Taça dos Libertadores

Base de Dados

Grupo 1109 / 1ª Entrega:

up202108742 Luís Contreiras

up202108728 Domingos Neto

up202108750 Rodrigo Resende

Resumo

Criação de uma base de dados relativa à Taça dos Libertadores e implementação da mesma em SQLite.

Palavras-Chave

UML; Modelo Conceptual; Esquema Relacional; Formas Normais; SQLite; Base de Dados; FEUP; Taça dos Libertadores.

Agradecimentos

Este projeto foi resultado de diversas contribuições e colaborações, dada de forma direta e indireta, mas todas elas essenciais à sua realização. Gostaríamos assim de expressar os nossos sinceros agradecimentos a todos os que tornaram possível este trabalho. Ao professor António Sá Pinto (asa.pinto@fe.up.pt) pela orientação dada e valioso acompanhamento constante durante o desenvolvimento do projeto.

Índice

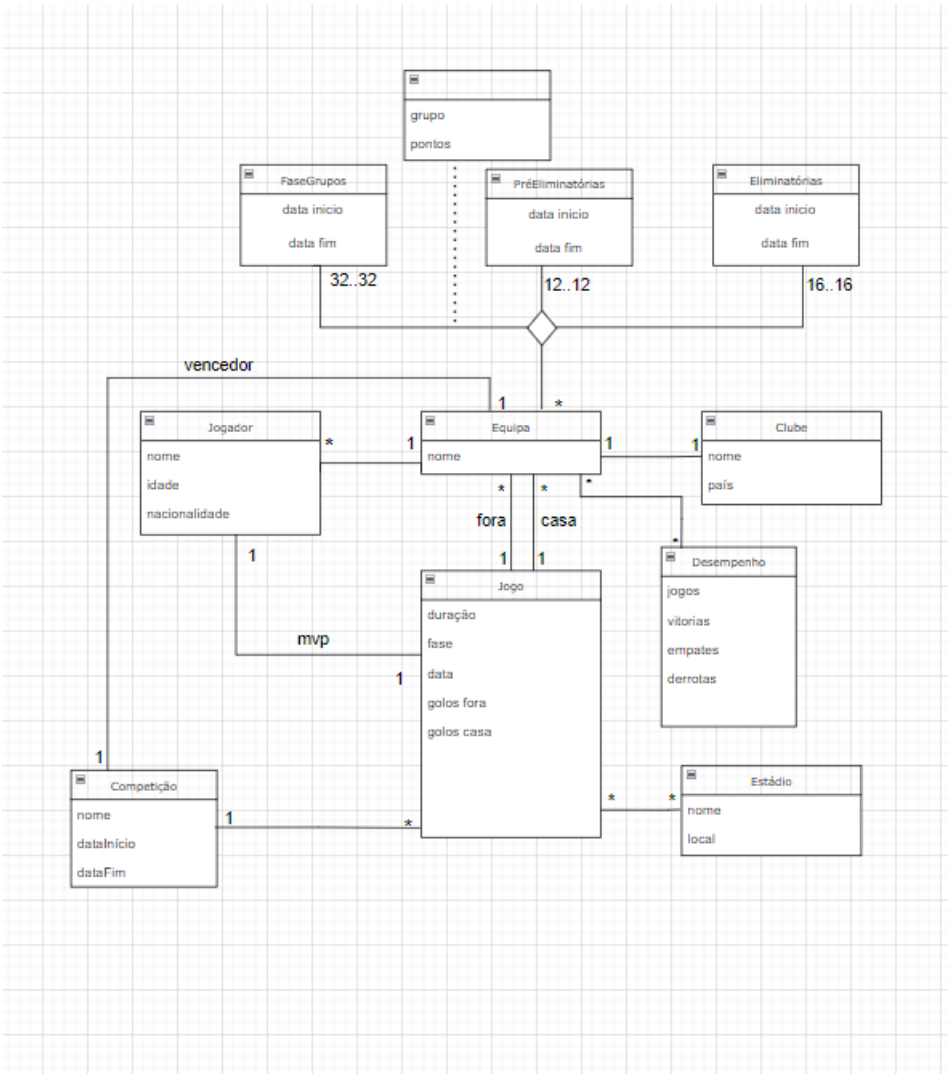
1. Introdução

Este projeto foi realizado no âmbito da unidade curricular Base de Dados, do 2º ano do Licenciatura em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

O objetivo deste projeto é implementar uma base de dados que contenha informação fundamental sobre a Taça Libertadores. Para tal será necessário criar o modelo conceptual, mapear esse modelo para um esquema relacional, implementar esse esquema numa base de dados SQLite, introduzir dados meios.

O presente relatório servirá para explorar a primeira parte desta implementação.

2. Diagrama UML



3. Esquema Relacional

Jogador(idJogador, nome, idade, nacionalidade, idEquipa -> Equipa, mvp -> Jogo)

Equipa(idEquipa, nome, vencedor -> Competição, casa -> Jogo, fora -> Jogo)

Clube(idClube, nome, país, idEquipa -> Equipa)

Jogo(idJogo, duração, jornada, data, golos fora, golos dentro, idCompetição -> Competição)

Estádio(idEstádio, nome, local)

JogoEstádio(idJogo -> Jogo, idEstádio -> Estádio)

Competição(idCompetição, nome, dataInício, nome, dataFim)

Desempenho(idDesempenho, jogos, vitórias, empates, derrotas)

EquipaDesempenho(idEquipa -> Equipa, idDesempenho -> Desempenho)

FaseGrupos(idFaseGrupos, dataInício, dataFim)

PréEliminatórias(idPréEliminatórias, dataInício, dataFim)

Eliminatórias(idEliminatórias, dataInício, dataFim)

EquipaFraseGrupos(idEquipa -> Equipa, idFaseGrupos -> FaseGrupos, grupos, pontos)

EquipaPréEliminatórias(idEquipa -> Equipa, idPréEliminatórias -> PréEliminatórias)

EquipaEliminatórias(idEquipa -> Equipa, idEliminatórias -> Eliminatórias)

4. Análise de Dependências Funcionais e Formas Normais

4.1 Dependências Funcionais

Jogador:

idJogador -> nome, idade, nacionalidade

Equipa:

idEquipa -> nome

Clube:

idClube -> nome, país

Jogo:

idJogo -> duração, jornada, data, golos fora, golos casa

Estádio:

idEstádio -> nome, local

Clube:

idClube -> nome, país

Competição:

idCompetição -> dataInício, dataFim

4.2 Violações

Jogador(idJogador, nome, idade, nacionalidade)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Equipa(idEquipa, nome)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Clube(idClube, nome, país)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Jogo(idJogo, duração, jornada, data)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Estádio(idEstádio, nome, local)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Competição(idCompetição,dataInício -> dataFim)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

FaseGrupos(idFaseGrupos -> dataInício, dataFim)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

PréEliminatórias(idPréEliminatórias -> dataInício, dataFim)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que

nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

Eliminatórias(idEliminatórias -> dataInício, dataFim)

A relação encontra-se na Forma Normal Boyce-Codd e na 3ª Forma Normal visto que nenhuma em dependência funcional não trivial o lado esquerdo é uma superchave.

5. Implementação em SQL

5.1 criar.sql

5.2 povoar.sql

6. Interrogações à Base de Dados

1. Qual foi a equipa vencedora do torneio? int1.int1.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int1.sql

```
1  .mode   columns
2  .headers    on
3  .nullvalue  NULL
4
5  --Qual foi a esquipa vencedora do torneio?
6  select e.nome as Campeão
7  from equipa e
8  join competicao cmp on cmp.vencedor = e.idEquipa
9  where cmp.vencedor = e.idEquipa;
```

2. Listagem de países de origem dos clubes participantes. int2.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int2.sql

```
1  .mode   columns
2  .headers    on
3  .nullvalue  NULL
4
5  --Listagem de países de origem dos clubes participantes
6  select c.pais as Países, count(*)
7  from clube c
8  group by c.pais;
```

3. Listagem de estádios ordenados por ordem decrescente de jogos disputados. int3.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int3.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem de estádios ordenados por ordem decrescente de jogos disputados
select est.nome, count(*)
from estadio est
join jogo jg on jg.estadio = est.idEstadio
where jg.estadio = est.idEstadio
group by est.nome
order by count(*) desc;
```

4. Quem foram os melhores jogadores em campo (mvp) na fase eliminatória?
Indicar idJogador, o nome do jogador, a equipa a que pertencem e a quantidade de jogos por ordem descendente. int4.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int4.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Quem foram os melhores jogadores em campo (mvp) na fase eliminatória?
--Indicar idJogador, o nome do jogador, a equipa a que pertencem e a quantidade de jogos por ordem descendente
select j.idJogador, j.nome as Jogador, e.nome as Equipa, count(*)
from jogador j
join equipa e on e.idEquipa = j.equipa
join jogo jg on jg.mvp = j.idJogador
where jg.mvp = j.idJogador and jg.fase != "Fase de Grupos" and jg.fase != "Pre Eliminatorias"
group by j.nome
order by count(*) desc;

```

5. Listagem de resultados na fase de grupos e na fase eliminatória.
Indicar a o nome da equipa em casa, os golos da equipa em casa, os golos da equipa de fora, o nome da equipa de fora. int5.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int5.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem de resultados na fase de grupos e fase eliminatoria
--Indicar a o nome da equipa em casa, os golos da equipa em casa, os golos da equipa de fora, o nome da equipa de fora
select ec.nome as Casa, jg.golosCasa, jg.golosFora, ef.nome as Fora
from jogo jg
join equipa ec on ec.idEquipa = jg.equipaCasa
join equipa ef on ef.idEquipa = jg.equipaFora
where fase != "Pre Eliminatorias";
```

6. Qual foi o melhor jogador do torneio?

Indicar o idJogador, nome, nacionalidade, idade, equipa. int6.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int6.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Qual foi o melhor jogador do torneio?
--Indicar o idJogador, nome, nacionalidade, idade, equipa
select j.idJogador, j.nome, j.nacionalidade, j.idade, e.nome as equipa
from jogador j
join equipa e on e.idEquipa = j.equipa
join jogo jg on jg.mvp = j.idJogador
group by j.nome
order by count(*) desc
limit 1;
```

7. Listagem de equipas que não passaram da fase pré-eliminatória. int7.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int7.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem de equipas que não passaram da fase pré eliminatória
select e.idEquipa, e.nome
from equipa e
join desempenho d on d.equipa = e.idEquipa
where d.jogos = 1;
```

8. Listagem de equipas que venceram jogos na fase eliminatória. int8.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int8.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem de equipas que venceram jogos na fase eliminatória
select e.idEquipa, e.nome
from equipa e
join jogo jg on e.idEquipa = jg.vencedor
where jg.fase != "Pre Eliminatorias" and jg.fase != "Fase de Grupos"
group by e.nome;
```

9. Listagem de equipas com mais vitórias do que derrotas por ordem decrescente de vitorias.

Indicar idEquipa, nome da equipa, vitorias e derrotas. int9.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int9.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem de equipas com mais vitórias do que derrotas por ordem decrescente de vitorias
--Indicar idEquipa, nome da equipa, vitorias e derrotas
select e.idEquipa, e.nome, d.vitorias, d.derrotas
from equipa e
join desempenho d on d.equipa = e.idEquipa
where d.vitorias > d. derrotas
order by d.vitorias desc;
```

10. Listagem dos jogos disputados em abril que tenha havido golos.

Indicar o idJogo, o nome da equipa em casa, os golos da equipa em casa, os golos da equipa de fora, o nome da equipa de fora, data do jogo. int10.sql

C: > Users > domin > Faculdade > BD > P_BD_G1109 > int10.sql

```
.mode   columns
.headers    on
.nullvalue  NULL

--Listagem dos jogos disputados em abril que tenha havido golos
--Indicar o idJogo, o nome da equipa em casa, os golos da equipa em casa, os golos da equipa de fora, o nome da equipa de fora, data do jogo
select jg.idJogo, ec.nome as Casa, jg.golosCasa, jg.golosFora, ef.nome as Fora, jg.data
from jogo jg
join equipa ec on ec.idEquipa = jg.equipaCasa
join equipa ef on ef.idEquipa = jg.equipaFora
where (jg.golosCasa > 0 or jg.golosFora > 0) and jg.data like "%/04/%";
```

7. Adição de gatilhos à base de dados

1. Gatilho que altera os pontos de uma equipa mediante um resultado modificado.

C: > Users > domin > Faculdade > BD > P_BD_G1109 > gatilho1_adiciona.sql

```
.mode columns
.header on
.nullvalue NULL

PRAGMA foreign_keys = ON;

-- Gatilho que altera os pontos de uma equipa mediante um resultado modificado
CREATE TRIGGER PontosVitoria
AFTER UPDATE OF vencedor ON JOGO
FOR EACH ROW
BEGIN
    UPDATE GRUPOEQUIPA
    SET pontos = pontos + 3
    WHERE GRUPOEQUIPA.equipa = NEW.vencedor;
END;

CREATE TRIGGER PontosDerrota
AFTER UPDATE OF vencedor ON JOGO
FOR EACH ROW
BEGIN
    UPDATE GRUPOEQUIPA
    SET pontos = pontos - 3
    WHERE GRUPOEQUIPA.equipa = OLD.vencedor;
END;
```

gatilho1_adiciona.sql

gatilho1_remove.sql

gatilho1_verifica.sql

2. Gatilho que permite a transferência (compra) de um jogador de uma equipa não participante.

C: > Users > domin > Faculdade > BD > P_BD_G1109 > gatilho2_adiciona.sql

```
.mode columns
.header on
.nullvalue NULL

PRAGMA foreign_keys = ON;

-- Gatilho que permite a transferência (compra) de um jogador de uma equipa não participante
CREATE TRIGGER PodeComprar
BEFORE INSERT ON JOGADOR
FOR EACH ROW
WHEN ( select capacidade
       from EQUIPA
       where EQUIPA.idEquipa = NEW.equipa ) >= 23
BEGIN
    SELECT RAISE(ABORT, 'Esta equipa tem demasiados jogadores');
END;

CREATE TRIGGER Compra
AFTER INSERT ON JOGADOR
FOR EACH ROW
BEGIN
    UPDATE EQUIPA
    SET capacidade = capacidade + 1
    WHERE EQUIPA.idEquipa = NEW.equipa;
END;
```

gatilho2_adiciona.sql

gatilho2_remove.sql

gatilho2_verifica.sql

3. Gatilho que permite a transferência (venda) de um jogador de uma equipa não participante.

C: > Users > domin > Faculdade > BD > P_BD_G1109 > gatilho3_adiciona.sql

```
.mode columns
.header on
.nullvalue NULL

PRAGMA foreign_keys = ON;

-- Gatilho que permite a transferência (venda) de um jogador
CREATE TRIGGER PodeVender
BEFORE DELETE ON JOGADOR
FOR EACH ROW
WHEN ( select capacidade
       from EQUIPA
       where EQUIPA.idEquipa = OLD.equipa ) <= 19
BEGIN
    SELECT RAISE(ABORT, 'Esta equipa não tem jogadores suficientes');
END;

CREATE TRIGGER Venda
AFTER DELETE ON JOGADOR
FOR EACH ROW
BEGIN
    UPDATE EQUIPA
    SET capacidade = capacidade - 1
    WHERE EQUIPA.idEquipa = OLD.equipa;
END;
```

gatilho3_adiciona.sql

gatilho3_remove.sql

gatilho3_verifica.sql

8. Avaliação da participação dos elementos do grupo

A contribuição para a realização do trabalho foi dividida pelos 3 elementos do grupo de uma forma não equitativa, sendo que dois dos membros (Domingos Neto: 202108728 e Luís Contreiras: 202108742) tiveram uma participação ativa.