INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

ENGENHARIA DE REDES E SISTEMAS DE COMPUTADORES

BASES DE DADOS

2020/2021

TRABALHO PRÁTICO - RELATÓRIO

Aluno:
Luís Oliveira - 24833

Docente:
Marco Lima

Instituto Politécnico
de Viana do Castelo

23 de janeiro de 2021

Conteúdo

Lista de Figuras

Capítulo 1

INTRODUÇÂO

1.1 Objetivos

Neste trabalho prático abordamos todos os tópicos dados na UC durante o semestre. Foi nos disponível 3 temas, tendo eu escolhido o tema nº2. Era pretendido que realizássemos o Modelo e Diagrama ER e a Normalização. Através do SQL PLUS devíamos realizar vários tipo de Scripts, sendo eles: Scripts de Criação da BD. Scripts de inserção, atualização e remoção com dados de exemplo. Scripts com Views, Stored Procedures, Triggers.

Capítulo 2

MODELAÇÃO

2.1 Modelo Entidade Relacionamento

ENTIDADES -
    PROFESSOR - (id_professor, nome_professor, idade_professor, telemovel_professor)
MODALIDADE - (nome_modalidade, id_professor, preco_mensal)
INSCRICAO - (id_inscrição, id_sócio, nome_modalidade, id_pagamento)
PAGAMENTO - (id_pagamento, id_inscrição, id_socio)
SOCIO - (id_socio, nome_socio, morada_socio, telemovel_socio)
TIPO PAGAMENTO - (tipo_pagamento, id_pagamento)
    RELACIONAMENTOS -
    INSCREVER - (MODALIDADE, PAGAMENTO, SOCIO)
ENSINAR - (PROFESSOR, MODALIDADE)

2.2 Diagrama Entidade Relacionamento

Realização do Diagrama ER na app Draw.io

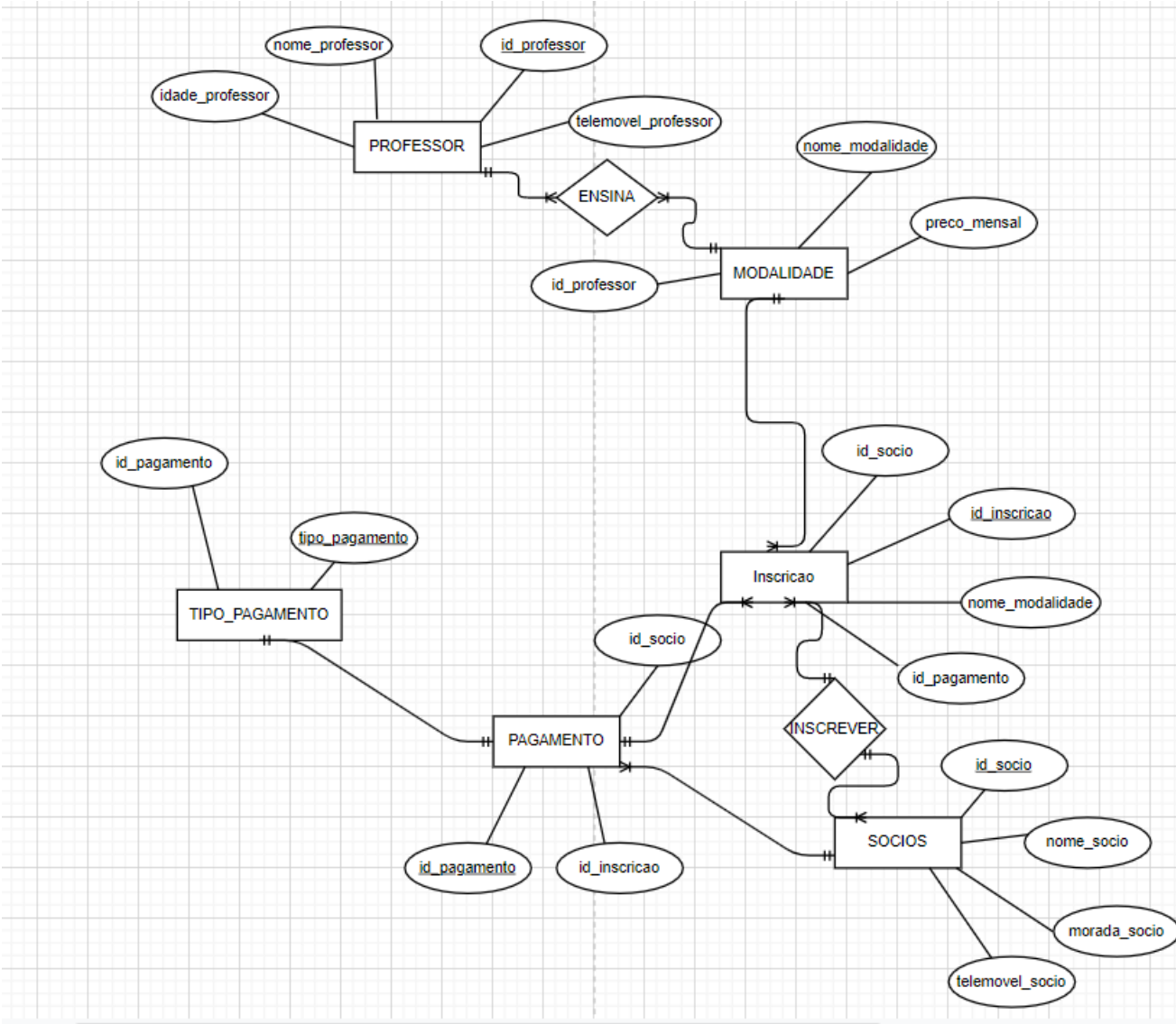

FIGURA 2.1: Diagrama ER

Realização do Modelo Relacional e Normalização na app Draw.io

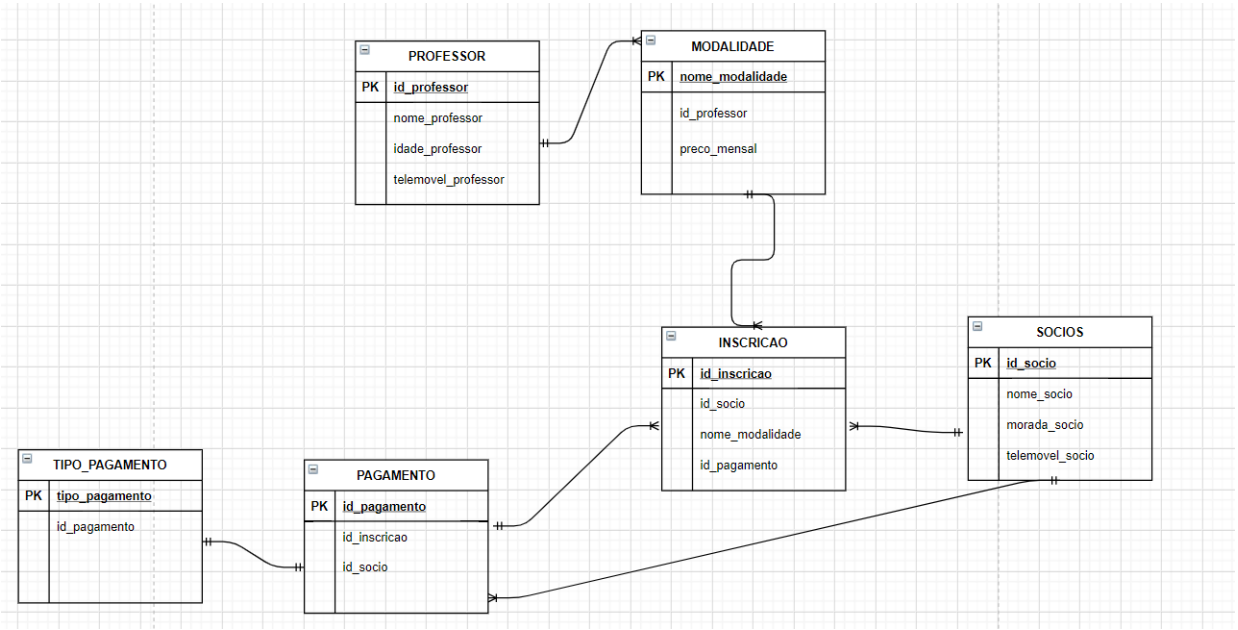

FIGURA 2.2: Modelo Relacional e Normalização

Capítulo 3

Base de Dados

3.1 Scripts DDL

Neste capítulo inseri scripts DDL, que permite criar, alterar, remover, tabelas e estruturas da base de dados.

```
SQL> CREATE TABLE PROFESSOR (
  2  id_professor INT NOT NULL,
  3  nome_professor VARCHAR(50) NOT NULL,
  4  idade_professor INT NOT NULL,
  5  telemovel_professor INT NOT NULL,
  6  PRIMARY KEY (id_professor)
  7  );

Table created.
```

FIGURA 3.1: Criação da tabela PROFESSOR

Criei a tabela PROFESSOR com 4 atributos,sendo id_nome a chave primária.

```
SQL> CREATE TABLE MODALIDADE (
  2  nome_modalidade VARCHAR(50) NOT NULL,
  3  id_professor INT NOT NULL,
  4  preco_mensal INT NOT NULL,
  5  PRIMARY KEY (nome_modalidade),
  6  FOREIGN KEY (id_professor) REFERENCES PROFESSOR (id_professor)
  7  );

Table created.
```

FIGURA 3.2: Criação da tabela MODALIDADE

Criei a tabela MODALIDADE com 3 atributos, sendo nome_modalidade chave primária e id_professor chave estrangeira.

```
SQL> CREATE TABLE INSCRICAO (
  2  id_inscricao INT NOT NULL,
  3  id_socio INT NOT NULL,
  4  nome_modalidade VARCHAR(50) NOT NULL,
  5  id_pagamento INT NOT NULL,
  6  PRIMARY KEY (id_inscricao),
  7  FOREIGN KEY (id_pagamento) REFERENCES PAGAMENTO (id_pagamento),
  8  FOREIGN KEY (id_socio) REFERENCES SOCIO (id_socio)
  9  );

Table created.
```

FIGURA 3.3: Criação da tabela INSCRICAO

Criei a tabela Inscricao com 4 atributos sendo id_inscricao chave primária e id_pagamento e id_socio chaves estrangeiras.

```
SQL> CREATE TABLE PAGAMENTO (
  2  id_pagamento INT NOT NULL,
  3  id_inscricao INT NOT NULL,
  4  id_socio INT NOT NULL,
  5  PRIMARY KEY (id_pagamento)
  6  );

Table created.
```

FIGURA 3.4: Criação da tabela PAGAMENTO

Criei a tabela PAGAMENTO com 3 atributos, sendo id_pagamento a chave primária.

```
SQL> CREATE TABLE SOCIO (
  2  id_socio INT NOT NULL,
  3  nome_socio VARCHAR(50) NOT NULL,
  4  morada_socio VARCHAR(50) NOT NULL,
  5  telemovel_socio INT NOT NULL,
  6  PRIMARY KEY (id_socio)
  7  );

Table created.
```

FIGURA 3.5: Criação da tabela SOCIO

Criei a tabela SOCIO com 4 atributos, sendo id_socio a chave primária.

```
SQL> CREATE TABLE TIPO_PAGAMENTO(
  2  tipo_pagamento VARCHAR(50) NOT NULL,
  3  id_pagamento INT NOT NULL,
  4  PRIMARY KEY (tipo_pagamento),
  5  FOREIGN KEY (id_pagamento) REFERENCES PAGAMENTO (id_pagamento)
  6  );

Table created.
```

FIGURA 3.6: Criação da tabela TIPO PAGAMENTO

Criei a tabela Tipo_Pagamento com 2 atributos, sendo tipo_pagamento chave primária, e id_pagamento chave estrangeira.

3.2 Scripts DML

Nesta parte inseri scripts DML, que permite alterar, adicionar, remover, ou consultar informação da base de dados.

Todos os dados inseridos aqui foram gerados pela app fornecia Mockaroo.

```
SQL> insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Carmelita', 1, 30, 942726694);

1 row created.

SQL> insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Brocky', 2, 36, 903181812);

1 row created.

SQL> insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Natividad', 2, 23, 910294560);
insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Natividad', 2, 23, 910294560)
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C007303) violated


SQL> insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Judas', 4, 35, 930321170);

1 row created.

SQL> insert into PROFESSOR (nome_professor, id_professor, idade_professor, telemovel_professor) values ('Sigismond', 5, 53, 926198098);

1 row created.
```

FIGURA 3.7: INSERT na tabela PROFESSOR

```
SQL> SELECT *FROM PROFESSOR;

ID_PROFESSOR NOME_PROFESSOR                                     IDADE_PROFESSOR
------------ ---------------------------------------------------- ---------------
TELEMOVEL_PROFESSOR
-------------------
           1 Carmelita                                                        30
          942726694

           2 Brocky                                                          36
          903181812

           4 Judas                                                           35
          930321170


ID_PROFESSOR NOME_PROFESSOR                                     IDADE_PROFESSOR
------------ ---------------------------------------------------- ---------------
TELEMOVEL_PROFESSOR
-------------------
           5 Sigismond                                                       53
          926198098
```

FIGURA 3.8: SELECT na tabela PROFESSOR

```
SQL> insert into MODALIDADE (nome_modalidade, id_professor, preco_mensal) values ('Luta', 1, 40);

1 row created.

SQL> insert into MODALIDADE (nome_modalidade, id_professor, preco_mensal) values ('Forca', 1, 18);

1 row created.
```

FIGURA 3.9: INSERT na tabela MODALIDADE

```
SQL> SELECT *FROM MODALIDADE;

NOME_MODALIDADE                                    ID_PROFESSOR PRECO_MENSAL
-------------------------------------------------- ------------ ------------
Dança                                                         4           32
Luta                                                          1           40
Forca                                                         1           18
```

FIGURA 3.10: SELECT na tabela MODALIDADE

```
SQL> insert into SOCIO (id_socio, nome_socio, morada_socio, telemovel_socio) values (3, 'Morey', 'Póvoa de Varzim', 977897723);

1 row created.

SQL>
SQL>
SQL> SELECT * FROM SOCIO
  2  SELECT * FROM SOCIO;
SELECT * FROM SOCIO
*
ERROR at line 2:
ORA-00933: SQL command not properly ended


SQL> SELECT * FROM SOCIO;

  ID_SOCIO NOME_SOCIO
---------- --------------------------------------------------
MORADA_SOCIO                                       TELEMOVEL_SOCIO
-------------------------------------------------- ---------------
         2 Samuele
Celorico de Basto                                       965012744

         5 Britteny
Fafe                                                    936427459

         3 Morey
Póvoa de Varzim                                         977897723


SQL>
```

FIGURA 3.11: Insert e SELECT na tabela SOCIO

```
SQL> select *from  TIPO_PAGAMENTO;

TIPO_PAGAMENTO                                     ID_PAGAMENTO
-------------------------------------------------- ------------
Dinheiro                                                      1
Multibanco                                                    3
Cheque                                                        1

SQL>
```

FIGURA 3.12: SELECT na tabela TIPO PAGAMENTO

```
SQL> SELECT * FROM PAGAMENTO;

ID_PAGAMENTO ID_INSCRICAO   ID_SOCIO
------------ ------------ ----------
           1            1          3
           3            1          1

SQL>
```

FIGURA 3.13: SELECT na tabela PAGAMENTO

3.3 Triggers, SP's e Views

```
SQL> CREATE OR REPLACE PROCEDURE GYM (id_util IN NUMBER)
  2  IS
  3  BEGIN
  4  INSERT INTO PROFESSOR VALUES ( id_util, 'Kevin', 45, 987654321);
  5  END;
  6  /

Procedure created.
```

FIGURA 3.14: Criação da SP

```
SQL> EXEC GYM (20);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM PROFESSOR;

ID_PROFESSOR NOME_PROFESSOR                                    IDADE_PROFESSOR
------------ ---------------------------------------------- ---------------
TELEMOVEL_PROFESSOR
-------------------
           1 Carmelita                                                   30
          942726694

           2 Brocky                                                      36
          903181812

           4 Judas                                                       35
          930321170


ID_PROFESSOR NOME_PROFESSOR                                    IDADE_PROFESSOR
------------ ---------------------------------------------- ---------------
TELEMOVEL_PROFESSOR
-------------------
           5 Sigismond                                                   53
          926198098

          20 Kevin                                                       45
          987654321
```

FIGURA 3.15: SELECT e EXECUTE DA SP

Criei uma SP que permite adicionar professores à tabela já existente.

```
SQL> CREATE VIEW professor_idade AS
  2  (SELECT id_professor,nome_professor, idade_professor, telemovel_professor
  3  FROM PROFESSOR
  4  WHERE idade_professor < 36);

View created.
```

FIGURA 3.16: Criação da view professor_idade

```
SQL> SELECT *FROM professor_idade;

ID_PROFESSOR NOME_PROFESSOR                                    IDADE_PROFESSOR
------------ -------------------------------------------------- ---------------
TELEMOVEL_PROFESSOR
-------------------
           1 Carmelita                                                       30
          942726694

           4 Judas                                                           35
          930321170
```

FIGURA 3.17: SELECT da view professor_idade

Criei uma view onde mostra a idade dos professores com menos de 36 anos. Neste caso só apareceram 2.

```
SQL> CREATE OR REPLACE VIEW modalidade_gym AS (SELECT COUNT(id_professor) AS ID_PROF, AVG(
AS PRECO_MENSAL FROM MODALIDADE GROUP BY id_professor);

View created.
```

FIGURA 3.18: Criação da view modalidade_gym

Criei outra view para adicionar modalidades.

```
SQL> INSERT INTO MODALIDADE (nome_modalidade, id_professor, preco_mensal) values ('TreinoPT', 2, 60);

1 row created.

SQL> SELECT *FROM MODALIDADE;

NOME_MODALIDADE                                    ID_PROFESSOR PRECO_MENSAL
-------------------------------------------------- ------------ ------------
Dança                                                         4           32
Luta                                                          1           40
Forca                                                         1           18
Mente                                                         5           45
TreinoPT                                                      2           60

SQL> SELECT *FROM modalidade_gym;

   ID_PROF PRECO_MENSAL
---------- ------------
         2           29
         1           60
         1           32
         1           45
```

FIGURA 3.19: Insert e SELECT da view modalidade_gym

```
SQL> update MODALIDADE SET preco_mensal = 50 WHERE id_professor = 2;

1 row updated.

SQL> select * from MODALIDADE
  2  select * from MODALIDADE;
select * from MODALIDADE
*
ERROR at line 2:
ORA-00933: SQL command not properly ended


SQL> select * from MODALIDADE;

NOME_MODALIDADE                                    ID_PROFESSOR PRECO_MENSAL
-------------------------------------------------- ------------ ------------
Dança                                                         4           32
Luta                                                          1           40
Forca                                                         1           18
Mente                                                         5           45
TreinoPT                                                      2           50

SQL>
```

FIGURA 3.20: UPDATE e SELECT da tabela MODALIDADE

Update feito onde o preço mensal do ginásio e o id do professor eram igual a 50 e a 2 respetivamente.

```
SQL> DELETE FROM MODALIDADE WHERE id_professor = 5;

1 row deleted.

SQL> select * from MODALIDADE;

NOME_MODALIDADE                                    ID_PROFESSOR PRECO_MENSAL
-------------------------------------------------- ------------ ------------
Dança                                                         4           32
Luta                                                          1           40
Forca                                                         1           18
TreinoPT                                                      2           50

SQL>
```

FIGURA 3.21: DELETE de um atributo da tabela MODALIDADE

Delete da Modalidade onde o professor com id = 5 exercia funções.

Capítulo 4

CONCLUSÕES

Por fim, este foi um trabalho onde dediquei bastante tempo, visto ser trabalhoso e demorado. Valeu a pena, visto que ajudou no estudo para o teste com a realização destes exercícios. Há algumas situações que ainda necessito de melhorar, como a criação de SPs e de Triggers. No global foi um trabalho importante para a aprendizagem de SQL e Modelação de Dados.