

# Trabalho Prático Aplicação de Base de dados

PARTE I

LICENCIATURA ENGENHARIA INFORMÁTICA

PÓS-LABORAL

António Santos | 30001686

Jorge Gonçalves | 30000621

Ricardo Costa | 30001464

Ricardo Ribeiro | 30000037

## Conteúdo

Introdução .....	3
Modelação .....	4
Modelo Conceitual .....	4
Diagrama Entidade-relacionamento (DER) .....	4
Identificação das Entidades .....	4
Identificação das Relações .....	6
DER Universidade .....	9
Modelo Lógico .....	9
Mapeamento DER para Modelo Relacional .....	10
Normalização .....	11
Modelo Relacional Universidade .....	12
Modelo Físico .....	13
Dicionário de dados .....	15
Aplicação – Parte I .....	16
Listagens .....	16

## Introdução

O presente trabalho é sobre o desenvolvimento de uma base de dados de uma universidade que tem como objetivo o armazenamento de alguns dados.

É objetivo deste trabalho a consolidação de matérias expostas anteriormente ou em andamento, e em particular o formalismo entidade-relacionamento, modelo relacional, linguagem SQL e o desenvolvimento de aplicações para aceder base de dados. As tarefas a realizar consistem em:

- Elaborar o modelo conceptual.
- Passagem do modelo conceptual para o modelo lógico.
- Levantamento de dependências funcionais e verificação da normalização do modelo lógico.
- Elaboração de Dicionário de Dados.
- Construção do modelo físico do sistema.
- Concepção de algumas consultas na linguagem SQL.

As ferramentas utilizadas foram as seguintes:

- Draw.io – Ferramenta que permite a construção de diagramas, fluxogramas, esquemas, etc...
- Oracle SQL Developer – IDE para trabalhar com SQL em base de dados Oracle.
- Oracle Application Express (APEX) – Ambiente para desenvolvimento de software baseado no SGBD da Oracle.

# 1. Modelação

## 1.1. MODELO CONCEPTUAL

Um modelo conceptual representa a informação de forma abstrata e não leva em conta o banco de dados em si, mas a forma como as estruturas são criadas para armazenar os dados. O modelo conceptual descreve o sistema de informação da organização identificando entidades, relações e atributos.

Iremos aplicar um modelo conceptual simples, o modelo entidade e relacionamento (MER). O MER foi apresentado, em 1976, por Peter Chen como alternativa a aplicação aos conceitos do Modelo Relacional. O MER é constituído por entidades e seus relacionamentos, conceitos que são representados graficamente e que podem ser apreendidos com facilidade.

### 1.1.1. Diagrama Entidade-relacionamento (DER)

O DER consiste na representação gráfica de um determinado modelo. Tem como objetivo apresentar rápida e claramente as entidades envolvidas e a forma que estas se relacionam entre si. Iremos, tal como proposto, utilizar a notação de Peter Chen para a construção do DER para o tema proposto.

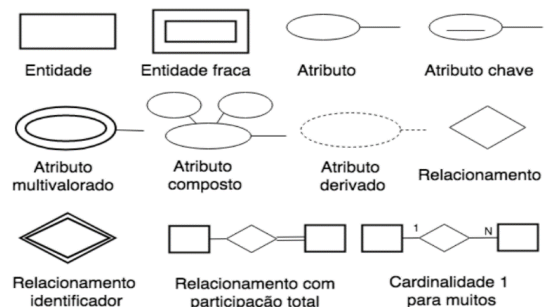


Figura 1- Notação de Peter Chen

### 1.1.2. Identificação das Entidades

Ao longo deste capítulo iremos descrever como foram identificadas cada uma das entidades e respetivos atributos.

*“A universidade é dividida em departamentos. Cada departamento tem um número, um nome e um escritório principal.”*

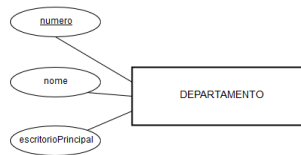


Figura 2 - Entidade DEPARTAMENTO e respectivos atributos

Na imagem acima, identificamos a entidade “Departamento” e respectivos atributos (numero, nome e escritorioPrincipal). O atributo “numero” foi definido como chave desta entidade.

*“Os professores têm um número de contribuinte, um nome próprio, um apelido, uma data de nascimento, uma categoria (professor assistente, auxiliar, associado ou catedrático), e uma ou mais especialidades de investigação (base de dados, engenharia de redes, inteligência artificial, etc.).”*

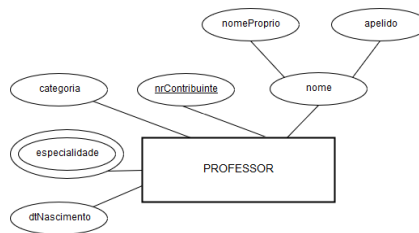


Figura 3 - Entidade DEPARTAMENTO e respectivos atributos

Nesta imagem, identificamos a entidade “Professor” e respectivos atributos (nrContribuinte, nome, categoria, especialidade e dtNascimento). O atributo “nrContribuinte” foi definido como chave desta entidade. O atributo “nome” é definido como atributo composto pois o seu conteúdo é formado por dois itens menores (nomeProprio e apelido). O atributo “especialidade” é definido como multivalorado pois o seu conteúdo é definido por ter mais de um valor.

*“Consideramos também que a universidade deseja guardar informação sobre projetos de investigação, organismo financiador, orçamento disponível e pessoas envolvidas. Assim, cada projeto têm um número de identificação, um organismo financiador, uma data de início, uma data de fim, e um orçamento.”*

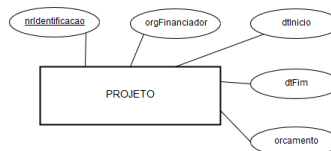


Figura 4 - Entidade PROJETO e respectivos atributos

Em cima, identificamos a entidade “Projeto” e respectivos atributos (nrIdentificacao, orgFinanciador, dtInicio, dtFim, orcamento). O atributo “nrIdentificacao” foi definido como chave desta entidade.

“Os estudantes de pós-graduação têm um número de contribuinte, um nome, uma data de nascimento, um ou mais contactos telefónicos, e um curso (ex. mestrado, doutoramento).”

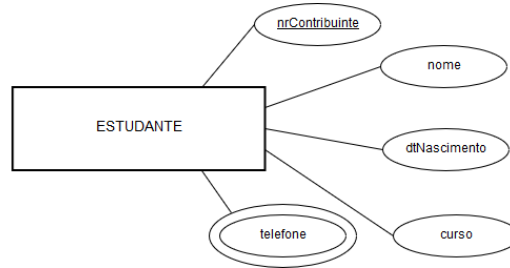


Figura 5 - Entidade ESTUDANTE e respetivos atributos

Na figura 5, identificamos a entidade “Estudante” e os respetivos atributos (nrContribuinte, nome, dtNascimento, curso, telefone). O atributo “nrContribuinte” foi definido como chave desta entidade. O atributo “telefone” é definido como multivalorado pois o seu conteúdo é definido por mais de um valor.

### 1.1.3. Identificação das Relações

Neste capítulo serão descritas cada uma das relações identificadas, seus graus, suas restrições e cardinalidade.

“Os departamentos são liderados por um professor.”

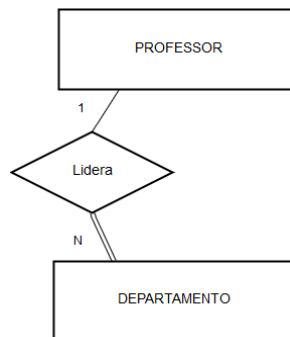


Figura 6- Relação PROFESSOR Lidera DEPARTAMENTO

Na figura 6, identificamos a relação binária “Lidera” entre as entidades “Professor” e “Departamento”. O “Professor” pode existir mesmo não existindo “Departamento”, portanto existe uma participação parcial. Já o “Departamento” apenas pode existir se estiver relacionado com um “Professor”, portanto a sua participação é total. Relativamente a cardinalidade é de uma relação 1:N (relação um-para-muitos).

*“Os professores podem trabalhar em um ou mais departamentos. Associada a cada um destas funções está uma percentagem do seu tempo.”*

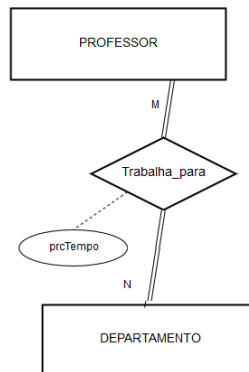


Figura 7 - Relação PROFESSOR Trabalha\_Para DEPARTAMENTO

Nesta imagem (figura 7) identificamos a relação binária “Trabalha\_para” entre as entidades “Professor” e “Departamento” e identificamos o respectivo atributo “prcTempo”. O “Departamento” devera estar relacionado obrigatoriamente com um ou mais “Professor” e vice-versa, portanto para ambos existe uma participação total. Relativamente a cardinalidade é de uma relação M:N (relação muitos-para-muitos).

*“Cada projeto é gerido por um professor (o investigador principal do projeto) e cada projeto tem a participação de um ou mais professores. Os professores podem gerir e/ou trabalhar em vários projetos.”*

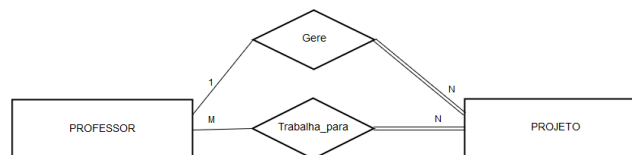


Figura 8 - Relações PROFESSOR Gere PROJETO e PROFESSOR Trabalha\_Para PROJETO

Na figura 8 apresentamos as relações binárias “Gere” e “Trabalha\_para” entre as entidades “Professor” e “Projeto”. Na relação “Gere” e “Trabalha\_para” o “Projeto” apenas pode existir se estiver relacionado com um “Professor”, portanto a sua participação é total, o “Professor” pode existir mesmo não existindo “Projeto”, portanto existe uma participação parcial. Na relação “Gere” a cardinalidade é de uma relação 1:N (relação um-para-muitos), na relação “Trabalha\_para” a cardinalidade é de uma relação M:N (relação muitos-para-muitos).

*“Cada projeto tem um ou mais estudantes de pós-graduação. Sempre que um estudante de pós-graduação trabalha em um projeto terá que existir um professor a supervisionar esse trabalho. Os estudantes podem trabalhar em vários projetos com supervisores eventualmente diferentes.”*

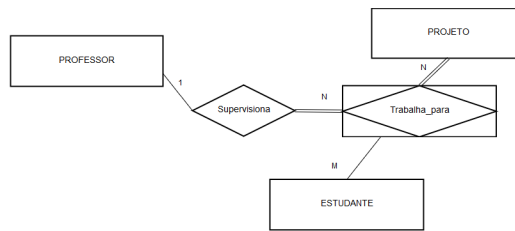


Figura 9 - Relação PROFESSOR Supervisiona (ESTUDANTE Trabalha\_Para PROJETO)

Figura 9 , a relação “Trabalha\_para” entre as entidades “Estudante” e “Projeto”. Na relação “Trabalha\_para” o “Projeto” apenas pode existir se estiver relacionado com pelo menos um “Estudante”, portanto a sua participação é total, o “Estudante” pode existir mesmo não existindo “Projeto”, portanto existe uma participação parcial. A cardinalidade é de uma relação M:N (relação muitos-para-muitos).

Devido a necessidade de associar mais uma entidade (“Professor”) a relação “Trabalha\_para” e no modelo MER não ser possível ter um relacionamento associado a outro relacionamento, tornamos a relação “Trabalha\_para” numa entidade associativa ,surge assim a relação “Supervisiona” entre a entidade “Professor” e a entidade associativa “Trabalha\_para”. Na relação “Supervisiona” a entidade associativa “Trabalha\_para” apenas pode existir se estiver relacionada com um “Professor”, portanto a sua participação é total, o “Professor” pode existir mesmo não existindo “Trabalha\_para”, portanto existe uma participação parcial. Relativamente a cardinalidade é a de uma relação 1:N (relação um-para-muitos).

Devido existirem três entidades envolvidas na relação descrita, então estamos perante uma relação ternária.

*“Os estudantes de pós-graduação estão associados a um departamento onde fazem o seu curso.”*

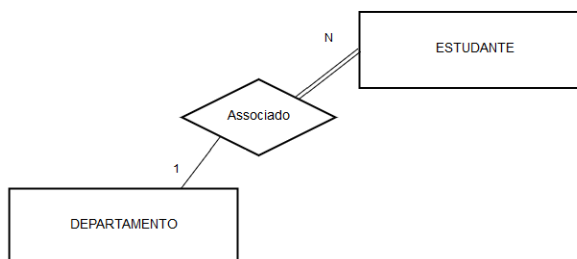


Figura 10 -Relação ESTUDANTE Associado DEPARTAMENTO

Em cima , identifica-mos a relação binária “Associado” entre as entidades “Estudante” e “Departamento”. O “Estudante” que apenas pode existir se estiver relacionado com um “Departamento”, portanto a sua participação é total. Já o “Departamento” pode existir mesmo não existindo “Estudante”, portanto existe uma participação parcial. Relativamente a cardinalidade é de uma relação 1:N (relação um-para-muitos).



“Cada estudante de pós-graduação tem um outro estudante mais velho que é o seu conselheiro.”

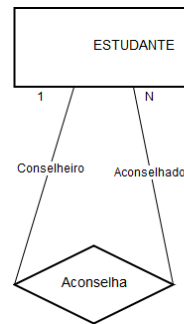


Figura 11 - Relação ESTUDANTE Aconselha ESTUDANTE

Neste último esquema , identificamos a relação unária “Aconselha” entre a entidade “Estudante” e ela própria. A cardinalidade é de uma relação 1:N (relação um-para-muitos).

#### 1.1.4.DER Universidade

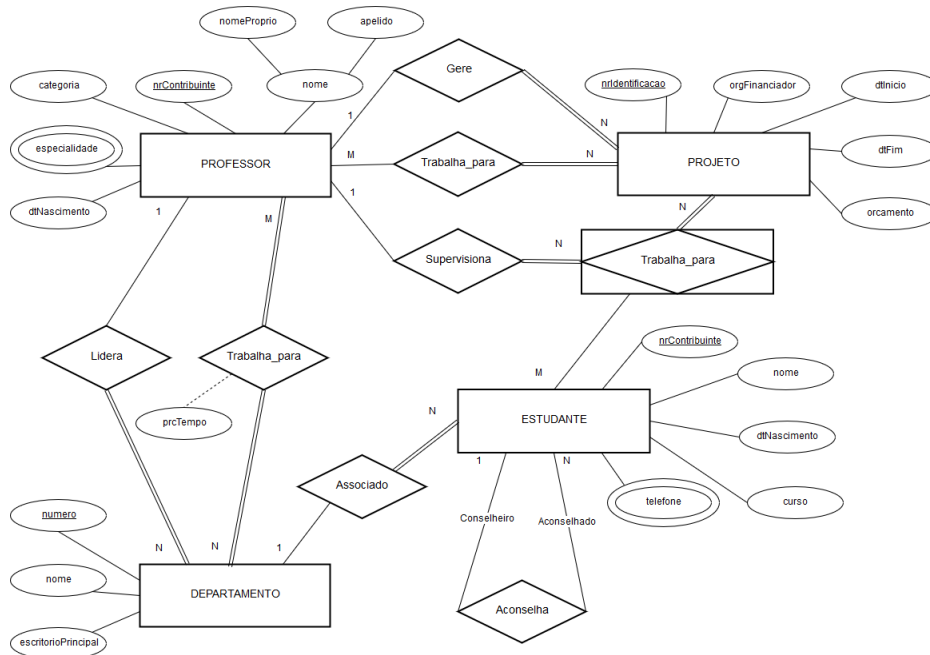


Figura 12 - Diagrama Entidade-Relacionamento Universidade

## 1.2. MODELO LÓGICO

O modelo lógico aplicado irá ser o Modelo Relacional. O Modelo Relacional de Dados foi proposto por Edgar Codd (1969) para responder as dificuldades colocadas pelos sistemas existentes, que obrigavam os programadores a conhecer a implementação física dos dados para os manipular. Uma das principais vantagens do modelo elaborado por

Codd foi ter-se baseado num ramo da matemática simples e poderoso – a teoria dos conjuntos.

### 1.2.1. Mapeamento DER para Modelo Relacional

As entidades e relações existentes no DER irão dar origem, de forma quase direta as tabelas que irão constituir a base de dados.

#### *Entidade PROFESSOR (Figura 3)*

Professor (nr\_contribuinte, categoria, data\_nascimento, nome\_proprio, apelido, especialidade)

#### *Entidade DEPARTAMENTO (Figura 2)*

Departamento (numero, nome, esc\_principal)

#### *Entidade ESTUDANTE (Figura 4)*

Estudante (nr\_contribuinte, nome, data\_nascimento, curso, telefone)

#### *Entidade PROJETO (Figura 5)*

Projeto (id, org\_financiador, data\_inicio, data\_fim, orçamento)

As relações “PROFESSOR Lidera DEPARTAMENTO” (Figura 6), “PROFESSOR Gere PROJETO” (Figura 8) e “ESTUDANTE Associado DEPARTAMENTO” (Figura 10) são relações entre duas entidades do tipo 1:N, portanto a entidade do lado N recebe como atributo a Chave Primária da entidade do lado 1.

#### *Relação PROFESSOR Lidera DEPARTAMENTO*

Departamento(numero, (...), **lider**)

#### *Relação PROFESSOR Gere PROJETO*

Projeto(id, (...), **inv\_princ**)

#### *Relação ESTUDANTE Associado DEPARTAMENTO*

Estudante(nr\_contribuinte, (...), **nr\_depart**)

As relações “PROFESSOR Trabalha\_para DEPARTAMENTO” (Figura 7), “PROFESSOR Trabalha\_para PROJETO” (Figura 8) e “ESTUDANTE Trabalha\_para PROJETO” (Figura 9) são relações entre duas entidades do tipo N:N, portanto devemos criar uma entidade associativa que permite associar as duas entidades originais através de duas relações 1:N. Desta forma a entidade associativa criada deves receber como atributo as chaves das duas entidades originais. A relação “PROFESSOR Trabalha\_para DEPARTAMENTO” contém um atributo próprio (prcTempo) que não vêm em nenhuma das entidades originais e que será necessário armazenar.

*Relação PROFESSOR Trabalha\_para DEPARTAMENTO*

Trabalha\_Departamento(nr\_prof, nr\_depart, prc\_tempo)

*Relação PROFESSOR Participa\_para PROJETO*

Participa\_Projeto(nr\_prof, id\_projeto)

*Relação ESTUDANTE Trabalha\_para PROJETO*

Trabalha\_Proj(nr\_estud, id\_projeto)

A relação “PROFESSOR Supervisiona (ESTUDANTE Trabalha\_Para PROJETO)” (Figura 9), trata-se de uma relação da entidade PROFESSOR com a entidade associativa ESTUDANTE Trabalha\_para PROJETO e que se trata de uma relação de duas entidades do tipo 1:N, portanto a entidade do lado N recebe como atributo a Chave Primária da entidade do lado 1. Para além disso optamos por mudar o nome da entidade associativa criada anteriormente.

*Relação PROFESSOR Supervisiona (ESTUDANTE Trabalha\_Para PROJETO)*

Trabalha\_Proj(nr\_estud, id\_projeto, nr\_prof)

A relação “ESTUDANTE Aconselha ESTUDANTE” (Figura 11), trata-se de uma relação recursiva pois a mesma entidade aparece nas duas extremidades da relação. Esta relação trata-se de uma relação de duas entidades do tipo 1:N, portanto a entidade do lado N recebe como atributo a Chave Primária da entidade do lado 1.

*Relação ESTUDANTE Aconselha ESTUDANTE*

Estudante(nr\_contribuinte, (...), conselheiro)

## 1.2.2.Normalização

A normalização é o processo que permite a simplificação da estrutura de uma base de dados de modo que esta se apresente num estado ótimo sem duplicação de informação. A terceira forma normal (3FN) será o nosso objetivo de normalização. As três primeiras formas normais baseiam-se no conceito de *Dependência Funcional* entre atributos de uma relação.

### 1.2.2.1.Primeira Forma Normal (1FN)

Uma relação diz-se que está na 1FN quando:

- Não contem atributos multivalor
- Não contem grupos repetitivos

No nosso DER definimos nas entidades PROFESSOR (especialidade) e ESTUDANTE (telefone) atributos multivalorados, portanto para garantirmos a 1FN adicionamos duas novas entidades.

#### *Entidade ESPECIALIDADE*

Professor(nr\_contribuinte, (...), especialidade)

Especialidade(descricao, nr\_prof)

#### *Entidade TELEFONE*

Estudante(nr\_contribuinte, (...), telefone)

Telefone(numero, nr\_estud)

#### *1.2.2.2.Segunda Forma Normal (2FN)*

Uma relação diz-se que está na 2FN quando:

- Esta na 1FN
- Todos atributos não chave dependem funcionalmente da totalidade da chave

Verificamos que, “Os estudantes de pós-graduação estão associados a um departamento onde fazem o seu curso.”, que um curso esta associado a um departamento, mas entendemos que um departamento poderá conter mais que um curso, portanto entendemos criar uma nova entidade afim de garantir a 2FN.

#### *Entidade CURSO*

Estudante (nr\_contribuinte, (...), curso, nr\_depart,)

Curso (descricao, nr\_depart)

#### *1.2.2.3.Terceira Forma Normal (3FN)*

Uma relação diz-se que está na 3FN quando:

- Esta na 2FN
- Todos atributos não chave não dependem funcionalmente uns dos outros.

Não verificamos nos atributos não chave qualquer dependência funcional uns dos outros.

### **1.2.3.Modelo Relacional Universidade**

### 1.3. MODELO FÍSICO

Com base no modelo lógico obtido elaboramos o modelo físico do sistema (SQL) contemplando todas as restrições de integridades que conseguimos garantir na forma declarativa.

```
CREATE TABLE PROFESSOR (  
    NR_CONTRIBUINTE NUMBER(9) NOT NULL,  
    NOME_PROPRIO VARCHAR2(25) NOT NULL,  
    APELIDO VARCHAR2(25) NOT NULL,  
    CATEGORIA VARCHAR2(50) NOT NULL,  
    DATA_NASCIMENTO DATE NOT NULL,  
    CONSTRAINT PK_PROFESSOR_NRCONTRIBUINTE PRIMARY KEY (NR_CONTRIBUINTE)  
);
```

```
CREATE TABLE PROJETO (  
    ID NUMBER(6) NOT NULL,  
    ORG_FINANC VARCHAR2(25) NOT NULL,  
    DATA_INICIO DATE NOT NULL,  
    DATA_FIM DATE NOT NULL,  
    ORCAMENTO DATE NOT NULL,  
    INV_PRINC NUMBER(9) NOT NULL,  
    CONSTRAINT PK_PROJETO_ID PRIMARY KEY (ID),  
    CONSTRAINT FK_PROJETO_GESTOR FOREIGN KEY (INV_PRINC) REFERENCES  
    PROFESSOR(NR_CONTRIBUINTE)  
);
```

```
CREATE TABLE DEPARTAMENTO(  

```

```
Professor(nr_contribuinte, categoria, data_nascimento, nome_propio, apelido)
```

```
Projeto(id, org_financ, data_inicio, data_fim, orcamento, inv_princ)
```

```
Estudante(nr_contribuinte, nome, data_nascimento, curso, nr_conselheiro)
```

```
Departamento(numero, nome, esc_principal, nr_lider)
```

```
Telefone(numero, nr_estud)
```

```
Especialidade(especialidade, nr_prof)
```

```
Trabalha_Departamento(nr_prof, nr_depart, prc_tempo)
```

```
Participa_Projeto(nr_prof, id_projeto)
```

```

CONSTRAINT FK_CURSO_DEPARTAMENTO FOREIGN KEY (NR_DEPART) REFERENCES
DEPARTAMENTO(NUMERO)
);

CREATE TABLE ESTUDANTE (
    NR_CONTRIBUINTE NUMBER(9) NOT NULL,
    NOME VARCHAR2(50) NOT NULL,
    DATA_NASCIMENTO DATE NOT NULL,
    CONSELHEIRO NUMBER(9),
    CURSO VARCHAR2(100) NOT NULL,
    CONSTRAINT PK_ESTUDANTE_NRCONTRIBUINTE PRIMARY KEY (NR_CONTRIBUINTE),
    CONSTRAINT FK_ESTUDANTE_CURSO FOREIGN KEY (CURSO) REFERENCES CURSO(CURSO)
);

CREATE TABLE TELEFONE (
    NUMERO VARCHAR2(30) NOT NULL,
    NR_ESTUD NUMBER(9) NOT NULL,
    CONSTRAINT PK_TELEFONE_NUMERO PRIMARY KEY (NUMERO),
    CONSTRAINT FK_TELEFONE_ESTUDANTE FOREIGN KEY (NR_ESTUD) REFERENCES
ESTUDANTE(NR_CONTRIBUINTE)
);

CREATE TABLE ESPECIALIDADE (
    DESCRICAO VARCHAR2(30) NOT NULL,
    NR_PROF NUMBER(9) NOT NULL,
    CONSTRAINT PK_ESPECIALIDADE_DESCRICAO PRIMARY KEY (DESCRICAO, NR_PROF),
    CONSTRAINT FK_ESPECIALIDADE_PROFESSOR FOREIGN KEY (NR_PROF) REFERENCES
PROFESSOR(NR_CONTRIBUINTE)
);

CREATE TABLE PARTICIPA_PROJETO (
    NR_PROF NUMBER(9) NOT NULL,
    ID_PROJETO NUMBER(6) NOT NULL,
    CONSTRAINT PK_PARTICIPA_PROJETO_NR_PROF_ID_PROJETO PRIMARY KEY (NR_PROF,
ID_PROJETO),
    CONSTRAINT FK_PARTICIPA_PROJETO_PROFESSOR FOREIGN KEY (NR_PROF) REFERENCES
PROFESSOR(NR_CONTRIBUINTE),
    CONSTRAINT FK_PARTICIPA_PROJETO_PROJETO FOREIGN KEY (ID_PROJETO) REFERENCES
PROJETO(ID)
);

CREATE TABLE TRABALHA_DEPARTAMENTO (
    NR_PROF NUMBER(9) NOT NULL,
    NR_DEPART NUMBER(3) NOT NULL,
    PRC_TEMPO NUMBER(3) NOT NULL,
    CONSTRAINT PK_TRABALHA_DEPARTAMENTO_NR_PROF_NR_DEPART PRIMARY KEY (NR_PROF,
NR_DEPART),
    CONSTRAINT FK_TRABALHA_DEPARTAMENTO_PROFESSOR FOREIGN KEY (NR_PROF) REFERENCES
PROFESSOR(NR_CONTRIBUINTE),
    CONSTRAINT FK_TRABALHA_DEPARTAMENTO_DEPARTAMENTO FOREIGN KEY (NR_DEPART)
REFERENCES DEPARTAMENTO(NUMERO)
);

CREATE TABLE TRABALHA_PROJETO (
    NR_ESTUD NUMBER(9) NOT NULL,
    ID_PROJETO NUMBER(6) NOT NULL,
    SUPERVISOR NUMBER(9) NOT NULL,
    CONSTRAINT PK_TRABALHA_PROJETO_NR_ESTUD_ID_PROJETO_SUPERVISOR PRIMARY KEY
(NR_ESTUD, ID_PROJETO, SUPERVISOR),
    CONSTRAINT FK_TRABALHA_PROJETO_PROFESSOR FOREIGN KEY (SUPERVISOR) REFERENCES
PROFESSOR(NR_CONTRIBUINTE),
    CONSTRAINT FK_TRABALHA_PROJETO_ESTUDANTE FOREIGN KEY (NR_ESTUD) REFERENCES
ESTUDANTE(NR_CONTRIBUINTE),
    CONSTRAINT FK_TRABALHA_PROJETO_PROJETO FOREIGN KEY (ID_PROJETO) REFERENCES
PROJETO(ID)
);

```

## 1.4. DICIONARIO DE DADOS

Relação: CATEGORIA		Categorias dos professores		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
CATEGORIA	VARCHAR2	20	Sim	Categoria (professor assistente, auxiliar, associado ou catedrático)

Relação: PROFESSOR		Dados dos professores da universidade		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NR_CONTRIBUINTE	NUMBER	9	Sim	Numero de contribuinte
NOME_PROPRIO	VARCHAR2	25		Nome proprio
APELIDO	VARCHAR2	25		Apelido
CATEGORIA	VARCHAR2	20	Estrangeira	Categoria (professor assistente, auxiliar, associado ou catedrático)
DATA_NASCIMENTO	DATE			Data de nascimento

Relação: CURSO		Dados dos cursos da universidade		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
DESCRICAO	VARCHAR2	100	Sim	Descricao do curso
NR_DEPART	NUMBER	3	Estrangeira	Identificação do departamento a que pertence o curso

Relação: ESPECIALIDADE		Dados dos Cursos		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
DESCRICAO	VARCHAR2	30	Sim	Descricao da especialidade
NR_PROF	NUMBER	9	Sim, Estrangeira	Identificação do professor

Relação: PROJETO		Dados dos projetos da universidade		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
ID	NUMBER	9	Sim	Identificação do Projeto
ORG_FINANC	VARCHAR2	25		Nome do organismo financiador
DATA_INICIO	DATE			Data inicio do projeto
DATA_FIM	DATE			Data fim do projeto
INV_PRINC	NUMBER	9	Estrangeira	Identificação do gestor/investigador principal do Projeto
ORCAMENTO	NUMBER	10,2		Orcamento do projeto

Relação: DEPARTAMENTO		Dados dos departamentos da universidade		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NUMERO	NUMBER	3	Sim	Numero do departamento
NOME	VARCHAR2	50		Nome do departamento
ESC_PRINCIPAL	NUMBER	3		Numero do escritorio principal
LIDER	NUMBER	9	Estrangeira	Identificação do lider do departamento

Relação: ESTUDANTE		Dados dos estudantes da universidade		
--------------------	--	--------------------------------------	--	--

Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NR_CONTRIBUINTE	NUMBER	9	Sim	Numero de contribuinte
NOME	VARCHAR2	50		Nome
DATA_NASCIMENTO	DATE			Data de nascimento
CONSELHEIRO	NUMBER	9	Estrangeira	Identificação do conselheiro do estudante
CURSO	VARCHAR2	100	Estrangeira	Curso do estudante

Relação: TELEFONE		Telefones dos estudantes		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NUMERO	VARCHAR2	30	Sim	Numero telefone do estudante
NR_ESTUD	NUMBER	9	Estrangeira	Identificação do estudante

Relação: TRABALHA_DEPARTAMENTO		Departamentos onde trabalham os professores		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NR_PROF	NUMBER	9	Sim, Estrangeira	Identificação do professor
NR_DEPART	NUMBER	3	Sim, Estrangeira	Identificação do departamento
PRC_TEMPO	NUMBER	3		Porcentagem de tempo que trabalha o professor

Relação: PARTICIPA_PROJETO		Projetos onde participam os professores		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NR_PROF	NUMBER	9	Sim, Estrangeira	Identificação do professor
ID_PROJETO	NUMBER	3	Sim, Estrangeira	Identificação do projeto

Relação: TRABALHA_PROJETO		Projetos onde trabalham os estudantes		
Atributo	Tipo de Dado	Tamanho	Chave	Descrição
NR_ESTUD	NUMBER	9	Sim, Estrangeira	Identificação do estudante
ID_PROJETO	NUMBER	3	Sim, Estrangeira	Identificação do projeto
SUPERVISOR	NUMBER	9	Sim, Estrangeira	Identificação do professor

## 2. Aplicação – Parte I

### 2.1. LISTAGENS

- Listar o nome completo (sem duplicações) dos professores que supervisionam projetos do aluno “30000621”.



```

select
    concat(concat(p.nome proprio, ' '), p.apelido) Nome_Completo_Supervisor
from
    estudante e
    inner join trabalha_projeto tp
        on e.nr_contribuinte = tp.nr_estud
    inner join professor p
        on p.nr_contribuinte = tp.supervisor
where
    e.nr_contribuinte = '30000621'
group by
    p.apelido, p.nome proprio;

```

	NOME_COMPLETO_SUPERVISOR
1	Rui Neves
2	Mário Marques da Silva
3	Marco Costa

- b. Listar o nome completo de todos os estudantes que não são conselheiros de outros estudantes

```

select
    e.nome Nome_Completo_Estudante
from
    estudante e
    left join estudante c
        on e.nr_contribuinte = c.conselheiro
where
    c.conselheiro is null;

```

	NOME_COMPLETO_ESTUDANTE
1	Ricardo Ribeiro
2	Jorge Gonçalves

- c. Listar o total de projetos cujo investigador principal é do departamento “Matemática”.

```

select
    count(*) Total_de_Projetos
from
    projeto p
    inner join trabalha_departamento td
        on p.inv_princ = td.nr_prof
    inner join departamento d
        on d.numero = td.nr_depart
where
    d.nome = 'Matemática';

```

	TOTAL_DE_PROJETOS
1	2

- d. Listar o nome completo de todos os professores e nome do respetivo departamento onde trabalha esses professores.

```
select
  concat(concat(p.nome_proprio, ' '), p.apelido) Nome_Completo,
  listagg(d.nome, ', ' ) within group(order by p.nome_proprio, p.apelido) Departamento
from
  professor p
  inner join trabalha_departamento td
    on td.nr_prof = p.nr_contribuinte
  inner join departamento d
    on td.nr_depart = d.numero
group by
  p.nome_proprio, p.apelido
order by
  p.nome_proprio, p.apelido;
```

	NOME_COMPLETO	DEPARTAMENTO
1	Enes Silveira	Direito, Matemática
2	Marco Costa	Matemática
3	Rui Neves	Informática, Matemática
4	Valeria Pequeno	Matemática

- e. Listar o nome completo e o papel que desempenha no projeto de todos os participantes do projeto “100”.

```
select
  Nome, listagg(Papel, ', ' ) within group(order by Nome) Papel
from
  (
    select
      p.id Id_Projeto, pr.nome_proprio || ' ' || pr.apelido Nome, 'Investigador Principal' Papel
    from
      projeto p
      inner join professor pr
        on p.inv_princ = pr.nr_contribuinte

    union

    select
      pp.id_projeto Id_Projeto, pr.nome_proprio || ' ' || pr.apelido Nome, 'Participante' Papel
    from
      participa_projeto pp
      inner join professor pr
        on pp.nr_prof = pr.nr_contribuinte

    union

    select
      tp.id_projeto Id_Projeto, pr.nome_proprio || ' ' || pr.apelido Nome, 'Supervisor Alunos'
  )
Papel
from
  trabalha_projeto tp
  inner join professor pr
    on tp.supervisor = pr.nr_contribuinte
```

```

union

select
    tp.id_projeto Id_Projeto, nome Nome, 'Aluno' Papel
from
    trabalha_projeto tp
    inner join estudante e
        on tp.nr_estud = e.nr_contribuinte
)
where
    Id_Projeto = 100
group by
    Nome
order by
    Nome;

```

NOME	PAPEL
1 António Santos	Aluno
2 Enes Silveira	Participante
3 João Vela Bastos	Participante
4 Jorge Gonçalves	Aluno
5 Marco Costa	Participante, Supervisor Alunos
6 Mário Marques da Silva	Investigador Principal, Participante, Supervisor Alunos
7 Ricardo Costa	Aluno
8 Rui Neves	Participante
9 Valeria Pequeno	Participante, Supervisor Alunos