UNIVERSIDADE
AUTÓNOMA
DE LISBOA

Aplicação de Base de Dados

Unidade curricular: Aplicação de Bases de Dados

Curso: Engenharia Informática

Turno: Diurno

Autores: Bruno Saraiva nº 20160782

Diogo Palos nº 30001058

Francisco Monteiro nº 30001029

Miguel Nunes nº 30000814

Ricardo Melo nº 30000486

UNIVERSIDADE
AUTÓNOMA
DE LISBOA

Introdução

Este trabalho consiste na criação de uma Base de Dados sobre uma universidade e o seu funcionamento em termos de departamentos, os professores, os alunos e os seus projetos.

Antes da base de dados foram criados o Diagrama Entidade Relacionamento (DER) e o respetivo Modelo Relacional (MR) com a sua normalização até à terceira forma.

Na base de dados, foram criadas as tabelas de dados, com as respetivas informações associadas e as relações entre esses mesmos dados, sendo esta posteriormente populada com dados e feito as respetivas consultas.

Na realização deste trabalho, foram utilizados o draw.io para a criação do diagrama (DER), o Apex do Oracle para fazer a Base de Dados e foi utilizado o word para fazer o modelo relacional (MR) e o relatório.

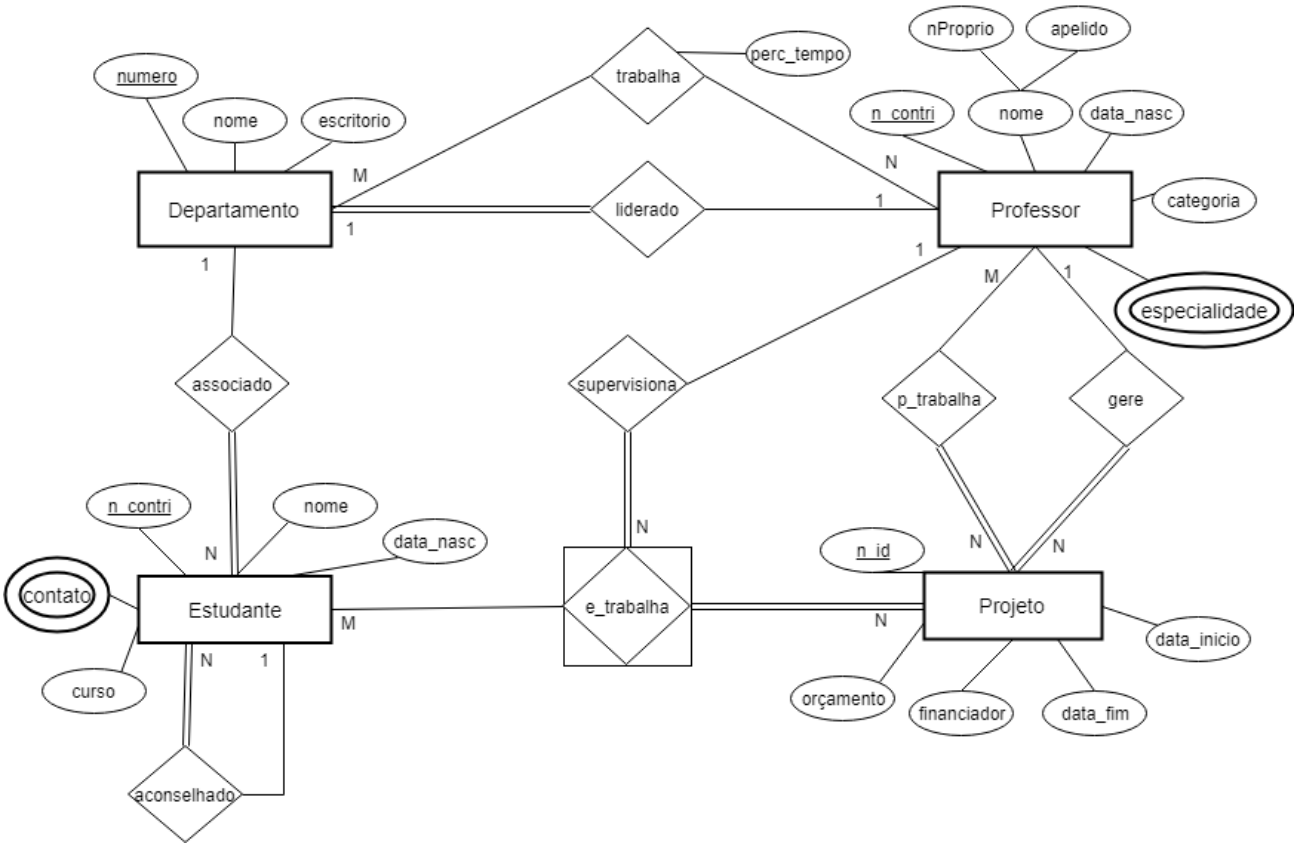
Para a segunda fase, foram feitas algumas alterações e correções no DER e no MR, assim como introduzidas restrições de integridade e ainda realizadas consultas com recurso a PL SQL através mais uma vez do Apex do Oracle onde foi feita a base de dados.

UNIVERSIDADE AUTÓNOMA DE LISBOA

Modelação

Alterações realizadas:

Diagrama Entidade Relacionamento (DER)



Quanto ao DER, foram feitas três alterações sendo estas:

1- O atributo especialidade da Entidade Professor foi alterado de atributo simples para atributo multivalorado, uma vez que é possível um professor ter várias especialidades.

UNIVERSIDADE
AUTÓNOMA
DE LISBOA

2- A cardinalidade da relação supervisiona foi alterada de 1:1 para 1:N uma vez que um professor pode supervisionar o trabalho de vários alunos.

3- A cardinalidade do auto-relacionamento da entidade estudante foi alterada de 1:1 para 1:N, pois um aluno mais velho pode aconselhar vários alunos mais novos.

UNIVERSIDADE AUTÓNOMA DE LISBOA

Modelo Relacional (MR):

DEPARTAMENTO(Numero Number, Nome Varchar2, Escritorio Varchar2, GerenteContri Number)

GerenteContri: fk(Professor)

PROFESSOR(N_Contri Number, NomeProprio Varchar2, Apelido Varchar2, Data_Nasc Date, Categoria Varchar2)

ESTUDANTE(N_Contri Number, Nome Varchar2, Data_Nasc Date, Curso Varchar2, ConselheiroContri Number, NumDepto Number)

ConselheiroContri: fk(Estudante)

NumDepto: fk(Departamento)

PROJETO(N_ID Number, Data_Inicio Date, Data_Fim Date, Financiador Varchar2, Orcamento Number, ProfContri Number)

ProfContri: fk(Professor)

E_TRABALHA(EstContri Number, Proj_ID Number, SupervisorContri Number)

EstContri: fk(Estudante)

Proj_ID: fk(Projeto)

SupervisorContri: fk(Professor)

TRABALHA(Perc_Tempo Number, NumDepto Number, ProfContri Number)

NumDepto: fk(Departamento)

ProfContri: fk(Professor)

P_TRABALHA(ProfContri Number, Proj_ID Number)

ProfContri: fk(Professor)

Proj_ID: fk(Projeto)

CONTATO_ESTUDANTE(EstContri Number, Contato Number)

EstContri: fk(Estudante)

ESPECIALIDADE_PROFESSOR(ProfContri Number, Especialidade Varchar2)

ProfContri: fk(Professor)

UNIVERSIDADE
AUTÓNOMA
DE LISBOA

Quanto ao MR foram feitas as seguintes alterações:

1- Foram adicionados os tipos de atributos em falta (ex: Number, Varchar2, Date).

2- Foi criada a relação Especialidade_Professor, uma vez que no DER o atributo especialidade passou a ser multivalorado e de acordo com as regras de transformação é necessária a nova relação com chave primária, neste caso composta pela chave primária de professor e pelo atributo multivalorado (especialidade).

3- Foram alterados os nomes de atributos repetidos para evitar certas complicações por exemplo o atributo ProfContri na tabela Departamento passou a ser GerenteContri, uma vez que se refere ao professor que gere este departamento, o atributo com o mesmo nome (ProfContri) na tabela E_Trabalha, passou a ser SupervisorContri, uma vez que se refere ao professor que supervisiona os trabalhos dos alunos.

UNIVERSIDADE AUTÓNOMA DE LISBOA

Restrições de Integridade

Não foram feitas alterações em relação às restrições de integridade da primeira parte.

Relações de Integridade a acrescentar na parte 2:

a) Solução 1:
ALTER TABLE departamento ADD CONSTRAINT dep_uniq UNIQUE (nome);

Ou

Solução 2:

```
CREATE OR REPLACE trigger checkndep
BEFORE INSERT OR UPDATE ON departamento
FOR EACH ROW
declare
v_cont number;
begin
select count(nome) into v_cont from departamento;
if (v_cont > 1) then
RAISE_APPLICATION_ERROR (-20508,
 'Já existe um departamento com esse nome.');
end if;
end;
```

Código para testar:

```
Insert into Departamento values (20,'Informática','Sala
49',323643178);
```

b)
```
alter table Departamento
  add constraint check_esc
  check (escritorio = null);
```

c)
```
alter table projeto
  add constraint check_date
  check (data_fim > data_inicio);
```

Código teste:

```
Insert into Projeto values
(1,to_date('20181202','YYYYMMDD'),to_date('20181201','YYYYMMDD'),'UAL',3590,246728918);
```

Resultado:

```
ORA-02290: check constraint (PROJETOABD2018.CHECK_DATE) violated
```

UNIVERSIDADE AUTÓNOMA DE LISBOA

d) CREATE OR REPLACE trigger checkcont

```
BEFORE INSERT OR UPDATE ON e_trabalha
FOR EACH ROW
DECLARE
v_cont number;
BEGIN
select count(Supervisorcontri) into v_cont from e_trabalha;
IF (v_cont > 3) THEN
RAISE_APPLICATION_ERROR (-20508,
 'O professor não pode supervisionar mais que 3 projetos.');
END IF;
END;
```

Código teste:

```
Insert into E_Trabalha values (293192532,21,235672834);
```

Resultado:

```
ORA-20508: O professor não pode supervisionar mais que 3 projetos.
ORA-06512: at "PROJETOABD2018.CHECKCONT", line 6
ORA-04088: error during execution of trigger 'PROJETOABD2018.CHECKCONT'
```

UNIVERSIDADE AUTÓNOMA DE LISBOA

Consultas PL/SQL

1 a) Listar o total de projetos cujo investigador principal é do departamento História.

Código:

```
DECLARE
d_nome departamento.nome%type := 'Historia';
p_count Number(1);
BEGIN
Select count(n_id)  into p_count FROM projeto,departamento WHERE projeto.profcontri = departamento.gerentecontri AND departamento.nome = d_nome;
dbms_output.put_line('O professor do curso de ' || d_nome || ' é investigador principal de ' || p_count || ' projetos. ');
END;
```

Results | Explain | Describe | Saved SQL | History

```
O professor do departamento de Historia é investigador principal de 2 projetos.

Statement processed.

0.02 seconds
```

UNIVERSIDADE AUTÓNOMA DE LISBOA

1 b) Listar o nome completo de todos os professores e nome do respetivo departamento onde trabalha esses professores.

Código:

```
DECLARE
cursor cur_c is select * from professor INNER JOIN departamento ON professor.n_contri = departamento.gerentecontri;
v_cur cur_c%rowtype;
BEGIN
open cur_c;
loop
fetch cur_c into v_cur;
exit when cur_c%notfound;
dbms_output.put_line('O professor: '|| v_cur.nomeproprio ||' '|| v_cur.apelido ||', trabalha no departamento: '|| v_cur.nome);
end loop;
close cur_c;
END;
```

Results | Explain | Describe | Saved SQL | History

```
O professor: José Martinho, trabalha no departamento: Informática
O professor: António Pina, trabalha no departamento: Matematica
O professor: Francisco Benoliel, trabalha no departamento: Eletrónica
O professor: Ana Balhau, trabalha no departamento: Direito
O professor: Carlos Lucas, trabalha no departamento: Quimica
O professor: João Silva, trabalha no departamento: Historia
O professor: Ana Balhau, trabalha no departamento: Gestão
O professor: Joaquim Silva, trabalha no departamento: Ciências Politicas
O professor: Maria Cruz, trabalha no departamento: Marketing
O professor: Ana Balhau, trabalha no departamento: Medicina

Statement processed.
```

UNIVERSIDADE AUTÓNOMA DE LISBOA

2) Crie um bloco PL/SQL (procedimento, função, trigger, etc., o que achar mais conveniente) chamado calcula_orcamento para calcular o orçamento disponível para um dado departamento (de acordo com os projetos que estão associados a ele).

Código:

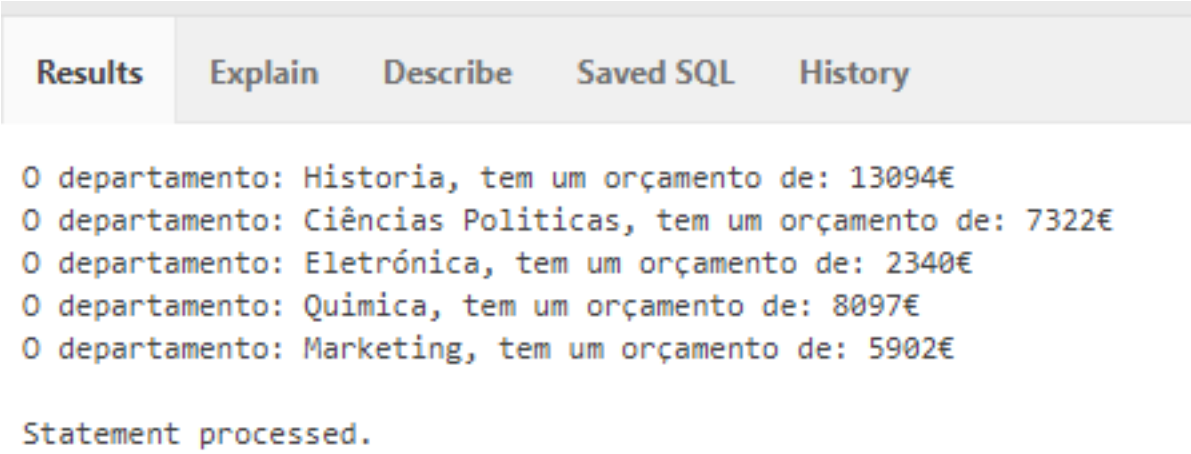
```
CREATE OR REPLACE PROCEDURE calcula_orcamento
AS
v_nome varchar2(30);
v_profcontri departamento.gerentecontri%type;
v_orc number;
BEGIN
for o in (select SUM(projeto.orcamento) orcamento,
departamento.nome from projeto INNER JOIN departamento ON
projeto.profcontri = departamento.gerentecontri GROUP BY
departamento.nome)
loop
dbms_output.put_line('O departamento: '|| o.nome ||', '|| 'tem um
orçamento de: '|| o.orcamento ||'€');
end loop;
END calcula_orcamento;
```

UNIVERSIDADE
AUTÓNOMA
DE LISBOA

3) Crie um bloco PL/SQL para testar o “calcula_orcamento”.

Código:

```
BEGIN
calcula_orcamento();
END;
```



UNIVERSIDADE AUTÓNOMA DE LISBOA

Solução extra:

Para a pergunta 2, para além da solução com o procedure, conseguimos ainda resolver este problema usando um cursor.

Código:

```
DECLARE
cursor calcula_orcamento is select SUM(projeto.orcamento) as orcamento, departamento.nome from projeto INNER JOIN departamento ON projeto.profcontri = departamento.gerentecontri GROUP BY departamento.nome;
v_orc calcula_orcamento%rowtype;
p_profcontri projeto.profcontri%type;
BEGIN
open calcula_orcamento;
loop
fetch calcula_orcamento into v_orc;
exit when calcula_orcamento%notfound;
dbms_output.put_line('O departamento: '|| v_orc.nome ||', '|| 'tem um orçamento de: '|| v_orc.orcamento ||'€');
end loop;
close calcula_orcamento;
END;
```

UNIVERSIDADE
AUTÓNOMA
DE LISBOA

Conclusão

Na realização deste projeto, conseguimos cumprir todos os objetivos propostos, desde o Diagrama entidade relacionamento e respetivo Modelo Relacional de acordo com as três formas normais à criação da base de dados e sua devida população e as consultas desejadas.

Quanto às dificuldades encontradas, a verificação das três formas normais do modelo relacional foi onde mais dúvidas surgiram. Em relação ao resto do trabalho, apesar de ter surgido também uma ou outra dúvida, foram cumpridas as tarefas a realizar.

Em relação à segunda parte, conseguimos corrigir os erros da primeira parte, sendo estes umas alterações ou correções ligeiras no DER e no MR e as respetivas alterações na base de dados.

Também foram cumpridos os objetivos da segunda parte, foram introduzidas as restrições de integridade e feitas as consultas pedidas que, apesar de algumas dificuldades ou erros, acabaram por ser realizadas corretamente, tendo em alguns casos específicos até mais do que uma solução disponível.