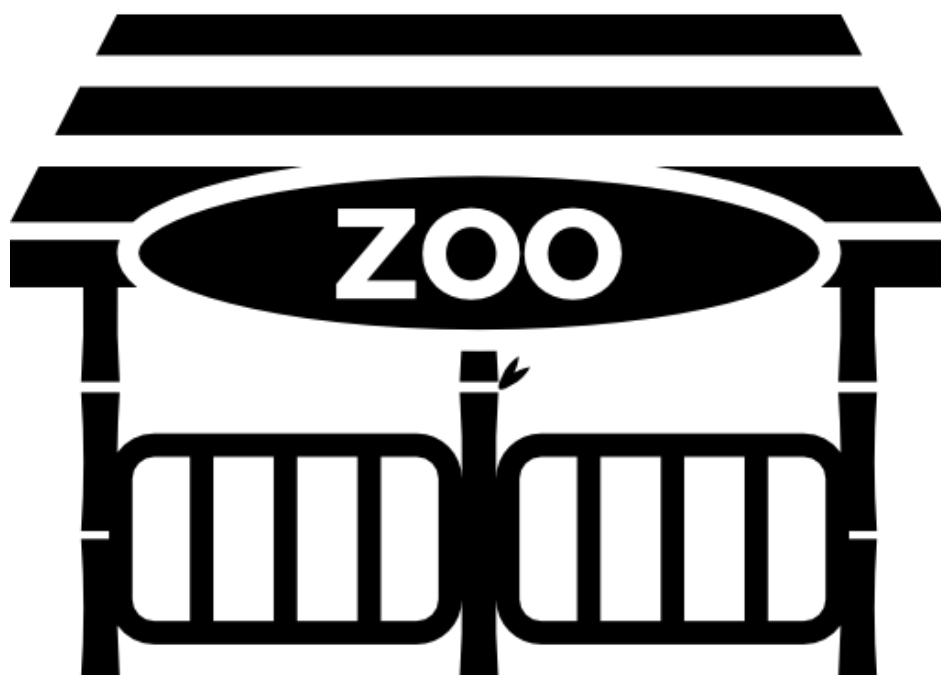


Relatório

Licenciatura em Engenharia Informática

Unidade Curricular Base de Dados



Discentes: Guilherme Grilo, nº48921

Hélder Godinho, nº42741

Docente: Prof.^ª Irene Pimenta Rodrigues

Conteúdo

| | |
|---|----|
| Introdução | 3 |
| Exercícios resolvidos | 4 |
| 1. Para o problema acima, construa um diagrama Entidade-Relação que descreva a informação. No diagrama não se esqueça de indicar as restrições das relações e as chaves primárias das entidades. | 4 |
| O diagrama E-R segue também em anexo no ficheiro “ER_diagram.png”. | 4 |
| 2. Transforme o modelo E-R em tabelas. Nas tabelas não se esqueça de indicar as chaves primárias e as chaves candidatas. | 4 |
| 3. Defina o conjunto de dependências funcionais que a base de dados deve verificar. | 5 |
| 4. Apresente a Base de Dados na forma normal de Boyce Codd, BCNF. | 6 |
| 5. Indique justificando se a sua base de dados tem informação redundante. | 6 |
| 6. Indique se a base de dados que apresentou na forma normal de Boyce-Codd preserva as dependências. | 6 |
| 7. Para cada relação da base de dados na BCNF indique as chaves primárias, candidatas e estrangeiras. | 6 |
| 8. Indique os comandos SQL para a criação das tabelas que constituem esta base de dados. E construa esta base de dados no Postgres. | 7 |
| 9. Indique as expressões em SQL para inserir a seguinte informação na sua base de dados e insira-a. | 10 |
| 10. Indique a expressão em SQL para responder às seguintes perguntas (no relatório indique também o resultado): | 14 |
| (a) Em que locais do zoo se podem visitar aves? | 14 |
| (b) Em que locais do zoo não há carnívoros? | 15 |
| (c) Indique os irmãos da Kilu. | 15 |
| (d) Indique os telefones do tratador responsável pela Kata. | 16 |
| (e) Indique os telefones do responsável pelo auxiliar responsável pela local onde está a Kata. | 16 |
| (f) Indique os tratamentos (data e tratamento) que a Mali já fez no zoo. | 16 |
| (g) Indique os nomes dos veterinários que já diagnosticaram uma gravidez a um carnívoro. | 17 |
| (h) Indique para cada família da ordem artiodáctilos quantos animais tem o zoo. | 17 |
| (j) Qual é a ordem com mais animais no zoo? | 17 |
| (k) Qual é a ordem dos animais que têm mais de 5 consultas por ano (diagnóstico ou tratamento). | 18 |
| (l) Indique o número de animais nascidos em cativeiro. | 18 |
| (m) Qual é o animal (nome e espécie) mais velho do zoo? | 18 |
| (n) Qual é o local húmido com mais mamíferos? | 19 |
| (o) Para cada tratador indique o número de mamíferos por que é responsável? | 19 |
| (p) Indique o nome dos animais que já foram tratados por todos os veterinários? | 20 |
| Conclusão | 21 |
| Referências bibliográficas | 22 |

Introdução

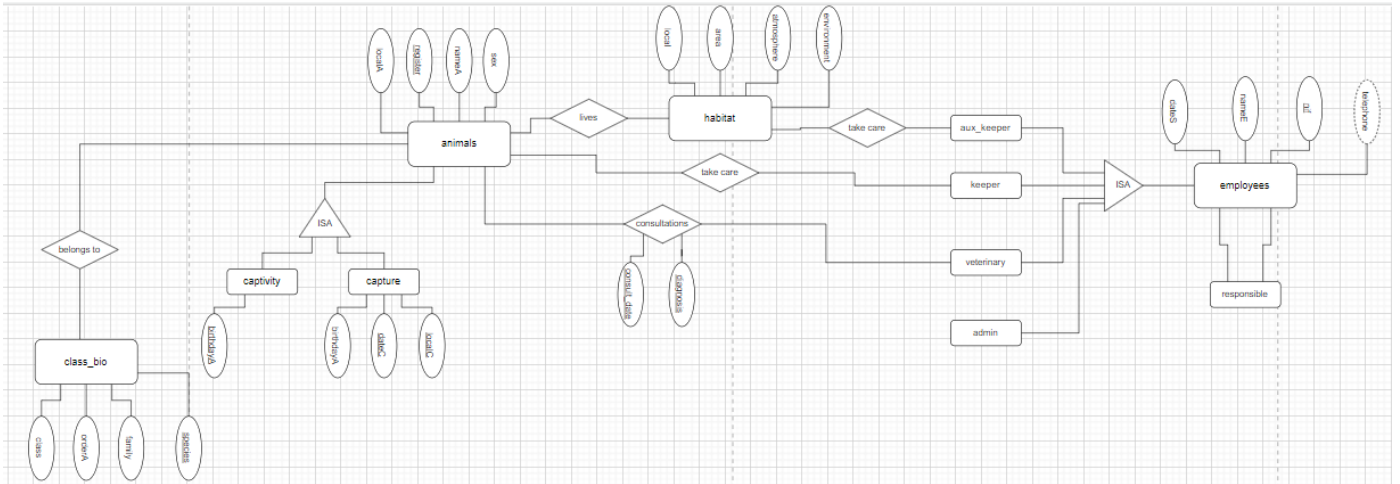
Neste trabalho pretendemos colocar em prática os nossos conhecimentos aprendidos ao longo do presente semestre na U.C. de Base de Dados tais como: desenvolver um diagrama E-R e transformá-lo em tabelas; definir dependências funcionais nas mesmas; criar tabelas, inserir e selecionar informação das mesmas em linguagem Postgresql; identificar chaves primárias e chaves candidatas de cada tabela; normalizar as tabelas e as relações entre as mesmas numa base de dados através da Forma Normal de Boyce-Codd.

Este trabalho está dividido em 10 partes/exercícios onde são abordados e postos em prática todos os conhecimentos anteriormente referidos.

A metodologia utilizada para a realização deste trabalho foi essencialmente pesquisa web gráfica, como por exemplo a visualização dos powerpoints desta U.C. ou recurso aos documentos oficiais da linguagem Postgresql.

Exercícios resolvidos

1. Para o problema acima, construa um diagrama Entidade-Relação que descreva a informação. No diagrama não se esqueça de indicar as restrições das relações e as chaves primárias das entidades.



O diagrama E-R segue também em anexo no ficheiro “ER_diagram.png”.

2. Transforme o modelo E-R em tabelas. Nas tabelas não se esqueça de indicar as chaves primárias e as chaves candidatas.

animals(register, nameA, sex, localA)

Chaves primárias- {register}

Chaves candidatas- {register}

habitat(local, area, atmosphere, environment)

Chaves primárias- {local}

Chaves candidatas- {local}

class_bio(species, registerA, class, orderA, family)

Chaves primárias- {species, registerA}

Chaves candidatas- {registerA}, {species, registerA}

captivity(birthdayA, registerA, registerDad, registerMom)

Chaves primárias- {birthdayA, registerA}

Chaves candidatas- {birthdayA, registerA}, {registerDad, registerMom, registerA}

capture(birthdayA, dateC, localC, registerA)

Chaves primárias- {dateC, registerA}

Chaves candidatas- {dateC, registerA}



employees(nameE, nif, dateS)

Chaves primárias- {nif}

Chaves candidatas- {nif}

keeper(nifE, registerA)

Chaves primárias- {nifE, registerA}

Chaves candidatas- {nifE, registerA}

aux_keeper(nifE, localH)

Chaves primárias- {nifE, localH}

Chaves candidatas- {nifE, localH}

veterinary(nifE)

Chaves primárias- {nifE}

Chaves candidatas- {nifE}

admin(nifE)

Chaves primárias- {nifE}

Chaves candidatas- {nifE}

responsible(nifresp, niftrab)

Chaves primárias- {niftrab}

Chaves candidatas- {niftrab}

telephone(nifE, numberTele)

Chaves primárias- {nifE, numberTele}

Chaves candidatas- {nifE, numberTele}

consultations(consult_date, nifE, registerA, diagnosis)

Chaves primárias- {consult_date, nifE, registerA, diagnosis}

Chaves candidatas- {consult_date, nifE, registerA, diagnosis }

3. Defina o conjunto de dependências funcionais que a base de dados deve verificar.

local -> aux_keeper

animals -> keeper

species -> family

family -> orderA



orderA -> class

consultations -> veterinary

consultations -> animals

employees -> nif

employees -> responsible

employees -> telephone

consultations -> consult_date

4. Apresente a Base de Dados na forma normal de Boyce Codd, BCNF.

register -> nameA, sex, localA

local -> area, atmosphere, environment

species, registerA -> family, orderA, class

birthdayA , registerA -> registerMom, registerDad

dateC, registerA -> birthdayA, localC

nif -> dateS, nameE

nifE -> registerA

registerA -> nifE

nifE -> localH

localH -> nifE

niftrab -> nifresp

numbertele -> nifE

5. Indique justificando se a sua base de dados tem informação redundante. Não, na minha perspetiva as minhas tabelas não possuem qualquer informação que possa ser considerada redundante.

6. Indique se a base de dados que apresentou na forma normal de Boyce-Codd preserva as dependências.

A minha base de dados, na forma normal de Boyce-Codd, preserva as dependências tal como foi provado no exercício 4.

7. Para cada relação da base de dados na BCNF indique as chaves primárias, candidatas e estrangeiras.

register -> nameA, sex, localA

Chaves primárias – {register}

Chaves candidatas – {register}, {nameA, sex, localA}

local -> area, atmosphere, environment



Chaves primárias- {local}

Chaves candidatas- {local}

species, registerA -> family, orderA, class

Chaves primárias- {species, registerA}

Chaves candidatas- {species, registerA}

Chaves estrangeiras- {registerA}

birthdayA , registerA-> registerMom, registerDad

Chaves primárias- {birthdayA, registerA}

Chaves candidatas- {birthdayA, registerA}

Chaves estrangeiras- {registerA}, {registerDad}, {registerMom}

dateC, registerA -> birthdayA, localC

Chaves primárias- {dateC, registerA}

Chaves candidatas- {dateC, registerA}

Chaves estrangeiras- {registerA}

nif -> dateS, nameE

Chaves primárias- {nif}

Chaves candidatas- {nif}, {dateS, name}

niftrab -> nifresp

Chaves primárias- {niftrab}

Chaves candidatas- {niftrab}

Chaves estrangeiras- {niftrab}, {nifresp}

8. Indique os comandos SQL para a criação das tabelas que constituem esta base de dados. E construa esta base de dados no Postgres.

Também disponível em anexo no ficheiro “create_tables.txt”.

animals(register, nameA, sex, localA)

DROP TABLE IF EXISTS animals CASCADE;

CREATE TABLE animals(

register DECIMAL PRIMARY KEY,

nameA varchar(100),

sex varchar(20),

localA VARCHAR(10)

);



habitat(local, area, atmosphere, environment)

```
DROP TABLE IF EXISTS habitat CASCADE;  
CREATE TABLE habitat(  
    local varchar(10) PRIMARY KEY,  
    area varchar(100),  
    atmosphere varchar(100),  
    environment varchar(100)  
);
```

class_bio(species, registerA, class, orderA, family)

```
DROP TABLE IF EXISTS class_bio CASCADE;  
CREATE TABLE class_bio(  
    species varchar(100),  
    registerA DECIMAL,  
    class varchar(100),  
    orderA varchar(100),  
    family varchar(100),  
    PRIMARY KEY (species, registerA),  
    FOREIGN KEY (registerA) REFERENCES animals ON DELETE RESTRICT  
);
```

captivity(birthdayA, registerA, registerDad, registerMom)

```
DROP TABLE IF EXISTS captivity CASCADE;  
CREATE TABLE captivity(  
    birthdayA DATE,  
    registerA DECIMAL,  
    registerDad DECIMAL,  
    registerMom DECIMAL,  
    PRIMARY KEY (birthdayA, registerA),  
    FOREIGN KEY (registerA) REFERENCES animals ON DELETE RESTRICT  
);
```

capture(birthdayA, dateC, localC, registerA)

```
DROP TABLE IF EXISTS capture CASCADE;  
CREATE TABLE capture(  
    birthdayA DATE,  
    dateC DATE,  
    localC varchar(100),  
    registerA DECIMAL,  
    PRIMARY KEY (dateC, registerA),  
    FOREIGN KEY (registerA) REFERENCES animals on DELETE RESTRICT  
);
```




employees(nameE, nif, dateS)

```
DROP TABLE IF EXISTS employees CASCADE;  
CREATE TABLE employees(  
    nameE varchar(100),  
    nif DECIMAL PRIMARY KEY,  
    dateS DATE  
);
```

keeper(nifE, registerA)

```
DROP TABLE IF EXISTS keeper CASCADE;  
CREATE TABLE keeper(  
    nifE DECIMAL,  
    registerA DECIMAL,  
    PRIMARY KEY (nifE, registerA),  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT,  
    FOREIGN KEY (registerA) REFERENCES animals ON DELETE RESTRICT  
);
```

aux_keeper(nifE, localH)

```
DROP TABLE IF EXISTS aux_keeper CASCADE;  
CREATE TABLE aux_keeper(  
    nifE DECIMAL,  
    localH varchar(10),  
    PRIMARY KEY (nifE, localH),  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT,  
    FOREIGN KEY (localH) REFERENCES habitat ON DELETE RESTRICT  
);
```

veterinary(nifE)

```
DROP TABLE IF EXISTS veterinary CASCADE;  
CREATE TABLE veterinary (  
    nifE DECIMAL PRIMARY KEY,  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT  
);
```

admin(nifE)

```
DROP TABLE IF EXISTS admin CASCADE;  
CREATE TABLE admin (  
    nifE DECIMAL PRIMARY KEY,  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT  
);
```



responsible(nifresp, niftrab)

```
DROP TABLE IF EXISTS responsible CASCADE;  
CREATE TABLE responsible(  
    nifresp DECIMAL,  
    niftrab DECIMAL PRIMARY KEY,  
    FOREIGN KEY (nifresp) REFERENCES employees ON DELETE RESTRICT,  
    FOREIGN KEY (niftrab) REFERENCES employees ON DELETE RESTRICT  
);
```

telephone(nifE, numberTele)

```
DROP TABLE IF EXISTS telephone CASCADE;  
CREATE TABLE telephone(  
    nifE DECIMAL,  
    numberTele DECIMAL,  
    PRIMARY KEY (nifE, numberTele),  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT  
);
```

consultations(consult_date, nifE, registerA, diagnosis)

```
DROP TABLE IF EXISTS consultations CASCADE;  
CREATE TABLE consultations(  
    consult_date DATE,  
    nifE DECIMAL,  
    registerA DECIMAL,  
    diagnosis varchar(100),  
    PRIMARY KEY (nifE, registerA, consult_date, diagnosis),  
    FOREIGN KEY (nifE) REFERENCES employees ON DELETE RESTRICT,  
    FOREIGN KEY (registerA) REFERENCES animals ON DELETE RESTRICT  
);
```

9. Indique as expressões em SQL para inserir a seguinte informação na sua base de dados e insira-a.

Disponível também em anexo no ficheiro “new_inserts.txt”.

```
INSERT INTO animals VALUES(123456, 'Taji', 'masculino', 'A3');  
INSERT INTO animals VALUES(222456, 'Mali', 'feminino', 'A3');  
INSERT INTO animals VALUES(322456, 'Aka', 'feminino', 'A3');  
INSERT INTO animals VALUES(422456, 'TaTa', 'masculino', 'A4');  
INSERT INTO animals VALUES(432456, 'Cáta', 'feminino', 'A5.1');  
INSERT INTO animals VALUES(522456, 'Kata', 'feminino', 'A5.1');  
INSERT INTO animals VALUES(622456, 'Mata', 'masculino', 'A4');  
INSERT INTO animals VALUES(123444, 'Hipo', 'masculino', 'A1');  
INSERT INTO animals VALUES(223444, 'Tapi', 'feminino', 'A1');  
INSERT INTO animals VALUES(323444, 'Hita', 'feminino', 'A1');  
INSERT INTO animals VALUES(123666, 'Kaki', 'masculino', 'A2');
```



```
INSERT INTO animals VALUES(223666, 'Kalu', 'feminino', 'A2');
INSERT INTO animals VALUES(323666, 'Kilu', 'feminino', 'A2');
INSERT INTO animals VALUES(423666, 'Luka', 'feminino', 'A2');
INSERT INTO animals VALUES(524666, 'Kuli', 'masculino', 'A2');
INSERT INTO animals VALUES(123555, 'Ará', 'masculino', 'A5.2');
INSERT INTO animals VALUES(133555, 'Zará', 'masculino', 'A5.2');
INSERT INTO animals VALUES(223555, 'Rará', 'feminino', 'A5.2');
INSERT INTO animals VALUES(323555, 'Rara', 'masculino', 'A5.2');
INSERT INTO animals VALUES(423555, 'Zula', 'feminino', 'A5.2');
INSERT INTO animals VALUES(523555, 'Zura', 'feminino', 'A5.2');
```

```
INSERT INTO habitat VALUES('A3', 1200, 'quente e húmida', 'terrestre');
INSERT INTO habitat VALUES('A4', 1100, 'quente e húmida', 'terrestre');
INSERT INTO habitat VALUES('A5.1', 1100, 'quente e húmida', 'terrestre');
INSERT INTO habitat VALUES('A1', 2000, 'quente e seca', 'misto');
INSERT INTO habitat VALUES('A2', 1500, 'fria e seca', 'terrestre');
INSERT INTO habitat VALUES('A5.2', 500, 'quente e húmida', 'terrestre');
```

```
INSERT INTO class_bio VALUES('tigre', 123456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 222456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 322456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 422456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 432456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 522456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('tigre', 622456, 'mamíferos', 'carnívoros', 'felinos');
INSERT INTO class_bio VALUES('hipópotamo comum', 123444, 'mamíferos', 'artiodáctilos',
'hipopótamos');
INSERT INTO class_bio VALUES('hipópotamo comum', 223444, 'mamíferos', 'artiodáctilos',
'hipopótamos');
INSERT INTO class_bio VALUES('hipópotamo comum', 323444, 'mamíferos', 'artiodáctilos',
'hipopótamos');
INSERT INTO class_bio VALUES('veado', 123666, 'mamíferos', 'artiodáctilos', 'cervídeos');
INSERT INTO class_bio VALUES('veado', 223666, 'mamíferos', 'artiodáctilos', 'cervídeos');
INSERT INTO class_bio VALUES('veado', 323666, 'mamíferos', 'artiodáctilos', 'cervídeos');
INSERT INTO class_bio VALUES('veado', 423666, 'mamíferos', 'artiodáctilos', 'cervídeos');
INSERT INTO class_bio VALUES('veado', 524666, 'mamíferos', 'artiodáctilos', 'cervídeos');
INSERT INTO class_bio VALUES('arara-azul-pequena', 123555, 'aves', 'psittaciformes',
'psittacidae');
INSERT INTO class_bio VALUES('arara-azul-pequena', 133555, 'aves', 'psittaciformes',
'psittacidae');
INSERT INTO class_bio VALUES('arara-azul-pequena', 223555, 'aves', 'psittaciformes',
'psittacidae');
INSERT INTO class_bio VALUES('arara-azul-pequena', 323555, 'aves', 'psittaciformes',
'psittacidae');
INSERT INTO class_bio VALUES('arara-azul-pequena', 423555, 'aves', 'psittaciformes',
'psittacidae');
INSERT INTO class_bio VALUES('arara-azul-pequena', 523555, 'aves', 'psittaciformes',
'psittacidae');
```



```
INSERT INTO captivity VALUES('2005/12/12', 322456, 123456, 222456);
INSERT INTO captivity VALUES('2006/01/20', 422456, 123456, 222456);
INSERT INTO captivity VALUES('2007/03/02', 522456, 422456, 432456);
INSERT INTO captivity VALUES('2008/02/02', 622456, 123456, 522456);
INSERT INTO captivity VALUES('2006/09/01', 323444, 123444, 223444);
INSERT INTO captivity VALUES('2008/04/03', 323666, 123666, 223666);
INSERT INTO captivity VALUES('2008/03/04', 524666, 123666, 423666);
INSERT INTO captivity VALUES('2009/05/07', 323555, 123555, 223555);
INSERT INTO captivity VALUES('2009/05/07', 423555, 123555, 223555);
INSERT INTO captivity VALUES('2009/05/07', 523555, 123555, 223555);
```

```
INSERT INTO capture VALUES('2003/05/24', '2004/06/02', 'India, Agra', 123456);
INSERT INTO capture VALUES('2003/02/27', '2004/03/16', 'India, Deli', 222456);
INSERT INTO capture VALUES('2004/09/01', '2005/01/01', 'India, Calcutá', 432456);
INSERT INTO capture VALUES('2003/05/30', '2004/06/06', 'África, Madagascar', 123444);
INSERT INTO capture VALUES('2003/12/02', '2004/06/06', 'África, Madagascar', 223444);
INSERT INTO capture VALUES('2005/11/11', '2006/05/14', 'Europa, Pirenéus', 123666);
INSERT INTO capture VALUES('2006/01/07', '2006/07/10', 'Europa, Ourense', 223666);
INSERT INTO capture VALUES('2006/10/01', '2007/02/08', 'Europa, Gerês', 423666);
INSERT INTO capture VALUES('2005/12/12', '2006/06/12', 'América do Sul, Paraná', 123555);
INSERT INTO capture VALUES('2006/02/11', '2006/06/12', 'América do Sul, Paraná', 133555);
INSERT INTO capture VALUES('2007/04/09', '2007/07/02', 'América do Sul, Uruguai', 223555);
```

```
INSERT INTO employees VALUES('Joaquim Silva', 123123123, '2003/02/01');
INSERT INTO employees VALUES('Manuel Santos', 123123124, '2003/04/01');
INSERT INTO employees VALUES('Maria Gomes', 123123125, '2003/01/01');
INSERT INTO employees VALUES('Mariana Silva', 123123126, '2004/02/01');
INSERT INTO employees VALUES('Jorge Gomes', 123123127, '2004/03/01');
INSERT INTO employees VALUES('Francisco Jorge', 123123128, '2004/03/01');
INSERT INTO employees VALUES('Manuel Ferreira', 123123129, '2004/02/01');
INSERT INTO employees VALUES('Manuela Torres', 123123130, '2004/04/01');
INSERT INTO employees VALUES('Pedro Vale', 123123131, '2004/05/01');
INSERT INTO employees VALUES('Isabel Soares', 123123132, '2004/06/01');
```

```
INSERT INTO keeper VALUES(123123123, 123456);
INSERT INTO keeper VALUES(123123123, 222456);
INSERT INTO keeper VALUES(123123123, 322456);
INSERT INTO keeper VALUES(123123123, 422456);
INSERT INTO keeper VALUES(123123123, 432456);
INSERT INTO keeper VALUES(123123123, 522456);
INSERT INTO keeper VALUES(123123123, 622456);
INSERT INTO keeper VALUES(123123124, 123444);
INSERT INTO keeper VALUES(123123124, 223444);
INSERT INTO keeper VALUES(123123124, 323444);
INSERT INTO keeper VALUES(123123124, 123666);
INSERT INTO keeper VALUES(123123124, 223666);
INSERT INTO keeper VALUES(123123124, 323666);
INSERT INTO keeper VALUES(123123124, 423666);
INSERT INTO keeper VALUES(123123124, 524666);
```



```
INSERT INTO keeper VALUES(123123125, 123555);  
INSERT INTO keeper VALUES(123123125, 133555);  
INSERT INTO keeper VALUES(123123125, 223555);  
INSERT INTO keeper VALUES(123123125, 323555);  
INSERT INTO keeper VALUES(123123125, 423555);  
INSERT INTO keeper VALUES(123123125, 523555);
```

```
INSERT INTO aux_keeper VALUES(123123126, 'A3');  
INSERT INTO aux_keeper VALUES(123123126, 'A4');  
INSERT INTO aux_keeper VALUES(123123126, 'A5.1');  
INSERT INTO aux_keeper VALUES(123123127, 'A1');  
INSERT INTO aux_keeper VALUES(123123128, 'A2');  
INSERT INTO aux_keeper VALUES(123123128, 'A5.2');
```

```
INSERT INTO veterinary VALUES(123123131);  
INSERT INTO veterinary VALUES(123123132);
```

```
INSERT INTO admin VALUES(123123129);  
INSERT INTO admin VALUES(123123130);
```

```
INSERT INTO responsible VALUES(123123125, 123123123);  
INSERT INTO responsible VALUES(123123125, 123123124);  
INSERT INTO responsible VALUES(123123130, 123123125);  
INSERT INTO responsible VALUES(123123130, 123123126);  
INSERT INTO responsible VALUES(123123130, 123123127);  
INSERT INTO responsible VALUES(123123130, 123123128);  
INSERT INTO responsible VALUES(123123130, 123123129);  
INSERT INTO responsible VALUES(123123129, 123123130);  
INSERT INTO responsible VALUES(123123129, 123123131);  
INSERT INTO responsible VALUES(123123131, 123123132);
```

```
INSERT INTO telephone VALUES(123123123, 919999999);  
INSERT INTO telephone VALUES(123123123, 266787809);  
INSERT INTO telephone VALUES(123123124, 919999998);  
INSERT INTO telephone VALUES(123123124, 266787808);  
INSERT INTO telephone VALUES(123123125, 919999997);  
INSERT INTO telephone VALUES(123123125, 266787807);  
INSERT INTO telephone VALUES(123123126, 919999996);  
INSERT INTO telephone VALUES(123123126, 266787806);  
INSERT INTO telephone VALUES(123123127, 919999995);  
INSERT INTO telephone VALUES(123123127, 266787806);  
INSERT INTO telephone VALUES(123123128, 919999994);  
INSERT INTO telephone VALUES(123123128, 266787806);  
INSERT INTO telephone VALUES(123123129, 919999996);  
INSERT INTO telephone VALUES(123123129, 266787806);
```



```
INSERT INTO telephone VALUES(123123130, 919999996);
INSERT INTO telephone VALUES(123123130, 266787806);
INSERT INTO telephone VALUES(123123131, 919999986);
INSERT INTO telephone VALUES(123123131, 266787816);
INSERT INTO telephone VALUES(123123132, 919999976);
INSERT INTO telephone VALUES(123123132, 266787826);
```

```
INSERT INTO consultations VALUES('2005/08/12', 123123131, 222456, 'grávida');
INSERT INTO consultations VALUES('2005/09/12', 123123131, 222456, 'cálcio injetado');
INSERT INTO consultations VALUES('2005/12/12', 123123131, 222456, 'parto');
INSERT INTO consultations VALUES('2006/07/12', 123123131, 222456, 'infecção');
INSERT INTO consultations VALUES('2006/07/12', 123123131, 222456, 'antibiótico injetado');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 123666, 'infecção');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 123666, 'antibiótico injetado');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 123555, 'infecção');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 123555, 'antibiótico injetado');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 423555, 'infecção');
INSERT INTO consultations VALUES('2009/05/12', 123123131, 423555, 'antibiótico injetado');
INSERT INTO consultations VALUES('2007/08/12', 123123131, 223444, 'infecção');
INSERT INTO consultations VALUES('2007/08/12', 123123131, 223444, 'antibiótico injetado');
INSERT INTO consultations VALUES('2006/07/12', 123123132, 223444, 'grávida');
INSERT INTO consultations VALUES('2006/07/12', 123123132, 223444, 'cálcio injetado');
INSERT INTO consultations VALUES('2006/09/12', 123123132, 223444, 'parto');
INSERT INTO consultations VALUES('2007/07/12', 123123132, 223444, 'infecção');
INSERT INTO consultations VALUES('2007/07/12', 123123132, 223444, 'antibiótico injetado');
INSERT INTO consultations VALUES('2007/07/12', 123123132, 223444, 'grávida');
INSERT INTO consultations VALUES('2007/07/12', 123123132, 223444, 'cálcio injetado');
INSERT INTO consultations VALUES('2007/09/12', 123123132, 223444, 'parto');
INSERT INTO consultations VALUES('2009/06/12', 123123132, 423555, 'infecção');
INSERT INTO consultations VALUES('2009/06/12', 123123132, 423555, 'antibiótico injetado');
```

10. Indique a expressão em SQL para responder às seguintes perguntas (no relatório indique também o resultado):

(a) Em que locais do zoo se podem visitar aves?

```
SELECT DISTINCT localA
FROM animals
JOIN class_bio ON animals.register = class_bio.registera
WHERE class like 'aves';
```

| localA |
|--------|
| A5.2 |



(b) Em que locais do zoo não há carnívoros?

```
SELECT DISTINCT locala
from animals
JOIN class_bio ON animals.register = class_bio.registera
WHERE ordera not like 'carnívoros';
```

| localA |
|--------|
| A1 |
| A2 |
| A5.2 |

(c) Indique os irmãos da Kilu.

```
SELECT namea
from animals
JOIN captivity ON animals.register = captivity.registera
WHERE registermom = (SELECT registermom
FROM captivity
JOIN animals ON captivity.registera = animals.register
WHERE animals.namea LIKE 'Kilu')
or
registerdad = (
SELECT registerdad
FROM captivity
JOIN animals ON captivity.registera = animals.register
WHERE animals.namea LIKE 'Kilu')
EXCEPT
SELECT namea
FROM animals
WHERE namea LIKE 'Kilu';
```

| namea |
|-------|
| Kuli |



(d) Indique os telefones do tratador responsável pela Kata.

```
SELECT numbertele
FROM telephone
JOIN keeper ON telephone.nife = keeper.nife
JOIN animals ON keeper.registera = animals.register
WHERE animals.namea LIKE 'Kata';
```

| numbertele |
|-------------------|
| 919999999 |
| 266787809 |

(e) Indique os telefones do responsável pelo auxiliar responsável pela local onde está a Kata.

```
WITH teleN as (select nife
from aux_keeper
JOIN animals ON aux_keeper.localh = animals.locala
where animals.namea LIKE 'Kata')
select numbertele
from telephone
join teleN on teleN.nife= telephone.nife
join responsible on responsible.niftrab = telephone.nife;
```

| numberTele |
|-------------------|
| 919999996 |
| 266787806 |

(f) Indique os tratamentos (data e tratamento) que a Mali já fez no zoo.

```
SELECT consult_date, diagnosis
FROM consultations
JOIN animals ON consultations.registera = animals.register
WHERE animals.namea LIKE 'Mali';
```

| consult_date | diagnosis |
|---------------------|----------------------|
| 2005-08-12 | grávida |
| 2005-09-12 | cálcio injetado |
| 2005-12-12 | parto |
| 2006-07-12 | infecção |
| 2006-07-12 | antibiótico injetado |



(g) Indique os nomes dos veterinários que já diagnosticaram uma gravidez a um carnívoro.

```
SELECT DISTINCT namee
FROM employees
JOIN consultations ON employees.nif = consultations.nife
NATURAL INNER JOIN class_bio
WHERE consultations.nife = employees.nif AND consultations.registera = class_bio.registera
AND class_bio.ordera LIKE 'carnívoros';
```

| namee |
|------------|
| Pedro Vale |

(h) Indique para cada família da ordem artiodáctilos quantos animais tem o ZOO.

```
SELECT family, COUNT(ordera) AS animalsN
FROM class_bio
WHERE ordera LIKE 'artiodáctilos'
GROUP BY family;
```

| family | ordera |
|-------------|--------|
| cervídeos | 5 |
| hipopótamos | 3 |

(j) Qual é a ordem com mais animais no zoo?

```
SELECT ordera, COUNT(ordera) AS orderN
FROM class_bio
GROUP by ordera
Order by orderN DESC
LIMIT 1;
```

| ordera | COUNT |
|---------------|-------|
| artiodáctilos | 8 |



(k) Qual é a ordem dos animais que têm mais de 5 consultas por ano (diagnóstico ou tratamento).

```
SELECT ordera, count(ordera)
from class_bio
natural INNER join consultations
where class_bio.registera = consultations.registera
group by ordera
HAVING COUNT(ordera) > 5;
```

| ordera | COUNT |
|----------------|-------|
| artiodáctilos | 12 |
| psittaciformes | 6 |

(l) Indique o número de animais nascidos em cativeiro.

```
WITH animalsN as (SELECT registera, count(registera) as captive
from captivity
group by registera)
select SUM(captive) from animalsN;
```

| SUM |
|-----|
| 10 |

(m) Qual é o animal (nome e espécie) mais velho do zoo?

```
WITH oldA AS (SELECT birthdaya AS age , registera
FROM captivity
UNION
SELECT birthdaya AS age , registera
FROM capture)
SELECT namea FROM animals
WHERE register =
(SELECT registera
FROM oldA
WHERE age = (SELECT MIN(age)
FROM oldA));
```

| namea |
|-------|
| Mali |



(n) Qual é o local húmido com mais mamíferos?

```
WITH mammals AS (SELECT register FROM animals
JOIN class_bio ON animals.register = class_bio.registera
WHERE class LIKE 'mamíferos'),
mammalszone AS (SELECT local, COUNT(animals) AS numanimals
FROM habitat
JOIN animals ON habitat.local = animals.locala
JOIN mammals ON animals.register = mammals.register
WHERE habitat.atmosphere LIKE '%húmida%'
group by local)
SELECT local FROM habitat
WHERE local = (SELECT local
FROM mammalszone
WHERE numanimals = (SELECT MAX(numanimals)
FROM mammalszone));
```

| local |
|-------|
| A3 |

(o) Para cada tratador indique o número de mamíferos por que é responsável?

```
WITH mammals AS (SELECT register FROM animals
JOIN class_bio ON animals.register = class_bio.registera
WHERE class LIKE 'mamíferos'),
mammalsPerKeeper AS (SELECT nife, registera
FROM keeper
join mammals ON keeper.registera = mammals.register)
SELECT namee, COUNT(mammalsPerKeeper.nife)
AS mammalkeeper
FROM employees
JOIN mammalsPerKeeper ON employees.nif = mammalsPerKeeper.nife
GROUP BY namee;
```

| namee | COUNT |
|---------------|-------|
| Joaquim Silva | 7 |
| Manuel Santos | 8 |



(p) Indique o nome dos animais que já foram tratados por todos os veterinários?

```
WITH vetIsabel as (Select registra
from consultations
NATURAL INNER JOIN veterinary
JOIN employees ON consultations.nife = employees.nif
WHERE employees.nameee LIKE 'Isabel Soares' ),
vetPedro as (Select registra from consultations
NATURAL INNER JOIN veterinary
JOIN employees ON consultations.nife = employees.nif
WHERE employees.nameee LIKE 'Pedro Vale')
select DISTINCT namea from animals
join vetPedro ON animals.register = vetPedro.registera
join vetIsabel ON animals.register = vetIsabel.registera;
```

| namea |
|-------|
| Tapi |
| Zula |



Conclusão

Neste trabalho abordámos todas as temáticas lecionadas na unidade curricular de Base de Dados e concluímos que conseguimos criar uma Base de Dados exceto normalizar a mesma na forma normal de Boyce-Codd (FNBC), onde revelámos muitas dificuldades acabando por não o conseguir fazer.

Conseguimos cumprir quase todos os objetivos delineados para este trabalho com exceção, tal como referido acima, de normalizar a nossa Base de Dados na FNBC, pois não entendemos como fazer.

Este trabalho foi bastante importante para a consolidação dos conhecimentos adquiridos nesta unidade curricular ao longo deste semestre visto que nos permitiu desenvolver e consolidar tudo o que foi lecionado na mesma.

Referências bibliográficas

Rodrigues, Irene. “Aulas de Base de Dados”, 2021. Universidade de Évora

Rodrigues, Irene. “trab2.pdf”. Universidade de Évora’s MOODLE

<https://www.postgresql.org/docs/>

<https://www.geeksforgeeks.org/postgresql-tutorial/>