UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

caio Schnepper

Eric Koiti Ueta

Evandro Alessi Correia Dos Santos

Matheus bezerra filho

Projeto final de fundamentos de banco de dados

PONTA GROSSA

2018

caio Schnepper

Eric Koiti Ueta

Evandro Alessi Correia Dos Santos

Matheus bezerra filho

Projeto final de fundamentos de banco de dados

Projeto final de Fundamentos de Banco de Dados apresentado como requisito parcial para obtenção da nota e conclusão da matéria, Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Professora: Simone de Almeida

PONTA GROSSA

2018

SUMÁRIO

[1 INTRODUÇÃO 4](#_Toc468489545)

[2 desenvolvimento 5](#_Toc468489546)

[**2.1** **MODELO DE ENTIDADE E RELACIONAMENTO** 7](#_Toc468489547)

[**2.2** **MODELO RELACIONAL** 8](#_Toc468489548)

[**2.3** **SCRIPT SQL DE CRIAÇÃO Da BAse DE DADOS** 8](#_Toc468489549)

[**2.4** **SCRIPT SQL DE CRIAÇão das STORED PROCEDUREs** 12](#_Toc468489550)

[**2.5** **SCRIPT sql DE criação dos indíces** 61](#_Toc468489551)

[**2.6** **SCRIPT sql DE criação das views** 62](#_Toc468489552)

[3 conclusão 66](#_Toc468489553)

# INTRODUÇÃO

Este projeto representa o processo que leva à criação de um banco de dados para a administração de um banco de sangue, partindo da criação do modelo entidade e relacionamento e finalizando no banco físico seguindo o que foi descrito no enunciado do projeto, utilizando views para realizar a consulta de dados mais avançados e stored procedure para manipulação desses dados. Este projeto será utilizado como método avaliativo da equipe a fim de obter nota parcial para a disciplina de Fundamentos de Banco de Dados, demandando a aplicação de conhecimentos e conceitos aprendidos durante o semestre.

A estrutura do projeto é: modelo de entidade e relacionamento, modelo relacional, scripts de criação do banco de dados, scripts de criação das stored procedures, scripts de criação dos índices e script de criação das views.

# desenvolvimento

O objetivo almejado com a criação do banco de dados foi de além do armazenamento dos dados referentes às doações de sangue dos doadores e detecção de possíveis doenças presentes do sangue, também gerenciar esses dados de maneira eficiente, e dessa forma viabilizar o armazenamento de doações ou exames passados, mantendo então a base de dados a mais conservadora possível em relação aos dados contidos nela. A modelagem foi desenvolvida com foco na optimização da base de dados, tendo o seguinte funcionamento:

- Para iniciar o processo de doação, o doador deve primeiramente fazer o exame de anemia, caso o nível de hemoglobina do doador esteja a baixo do estabelecido para a doação, o processo de doação é interrompido.

- Para casos de doadores frequentes, o sistema irá fazer a busca da última data da última doação realizada, e para realizar uma nova doação o período não poderá ser inferior a 90 dias para homens ou 120 dias para mulheres.

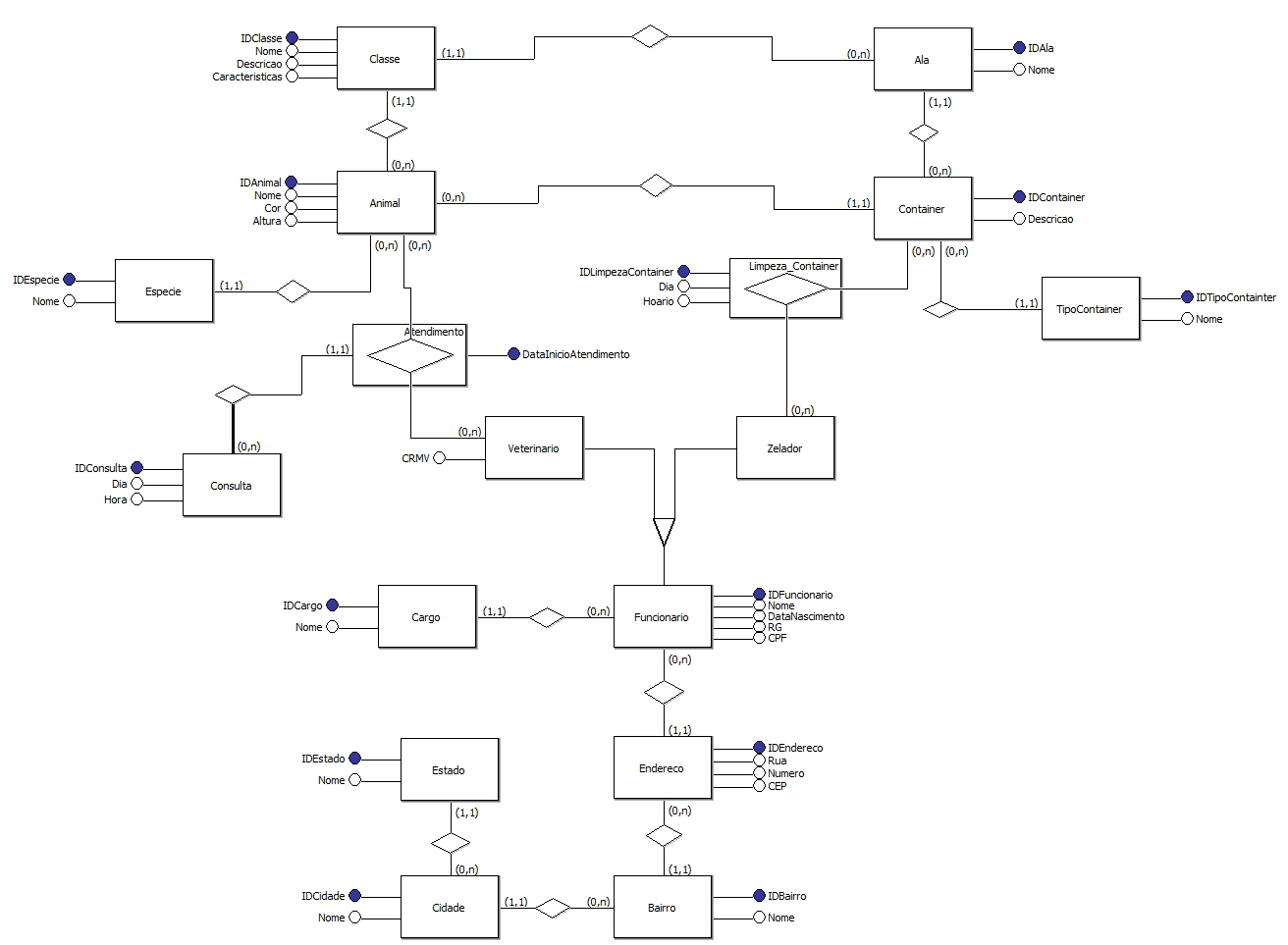
- Caso seja feita a constatação de que o doador é inapto a realizar a doação, seus dados ficam salvos na base de dados.

- Para realmente concluir uma doação, é necessário que o doador atenda a todos os requisitos da fundação pró-sangue, tanto em exames quanto na triagem clínica.

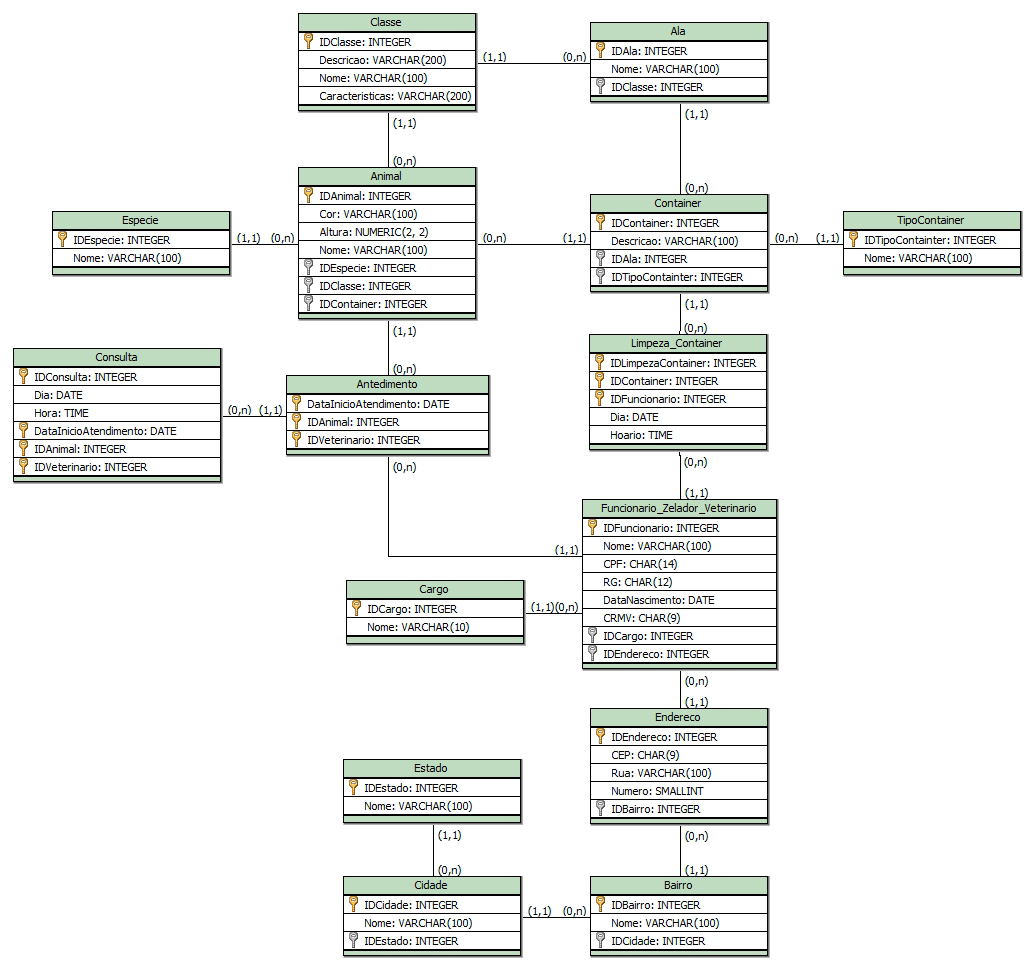
- A exclusão de dados apenas é permitida, se houver casos de tuplas apenas com chaves.

A análise do enunciado que resultou no modelo apresentado baseou-se no fato de que para finalizar o processo de doação de sangue, o doador deve passar por uma série de exames necessários que são feitos individualmente entre si para liberação de sua efetiva doação. Como também o controle de futuras novas doações feitas por um mesmo doador. Portanto, não há necessidade de preocupação caso o doador seja convocado para uma consulta médica ou para repetição de exame.

## **MODELO DE ENTIDADE E RELACIONAMENTO**



## **MODELO RELACIONAL**



## **SCRIPT SQL DE CRIAÇÃO Da BAse DE DADOS**

DROP DATABASE IF EXISTS Zoologico;

CREATE DATABASE Zoologico;

USE Zoologico;

CREATE TABLE Cargo (

IDCargo INTEGER auto\_increment,

Nome VARCHAR(100),

PRIMARY KEY (IDCargo) );

CREATE TABLE Estado (

IDEstado INTEGER auto\_increment,

Nome VARCHAR(100),

PRIMARY KEY (IDEstado));

CREATE TABLE Cidade (

IDCidade INTEGER auto\_increment,

Nome VARCHAR(100),

IDEstado INTEGER,

PRIMARY KEY (IDCIdade),

FOREIGN KEY(IDEstado) REFERENCES Estado (IDEstado) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Bairro (

IDBairro INTEGER auto\_increment,

Nome VARCHAR(100),

IDCidade INTEGER,

PRIMARY KEY (IDBairro),

FOREIGN KEY(IDCidade) REFERENCES Cidade (IDCidade) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Endereco (

IDEndereco INTEGER auto\_increment,

CEP CHAR(9),

Rua VARCHAR(100),

Numero SMALLINT,

IDBairro INTEGER,

PRIMARY KEY (IDEndereco),

FOREIGN KEY(IDBairro) REFERENCES Bairro (IDBairro) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Funcionario (

IDFuncionario INTEGER auto\_increment,

Nome VARCHAR(100),

CPF CHAR(14),

RG CHAR(12),

DataNascimento DATE,

CRMV CHAR(9),

IDCargo INTEGER,

IDEndereco INTEGER,

PRIMARY KEY (IDFuncionario),

FOREIGN KEY(IDCargo) REFERENCES Cargo (IDCargo) ON UPDATE RESTRICT ON DELETE RESTRICT,

FOREIGN KEY(IDEndereco) REFERENCES Endereco (IDEndereco) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Especie (

IDEspecie INTEGER auto\_increment,

Nome VARCHAR(100),

PRIMARY KEY (IDEspecie));

CREATE TABLE Classe (

IDClasse INTEGER auto\_increment,

Descricao VARCHAR(200),

Nome VARCHAR(100),

Caracteristicas VARCHAR(200),

PRIMARY KEY (IDClasse));

CREATE TABLE Ala (

IDAla INTEGER auto\_increment,

Nome VARCHAR(100),

IDClasse INTEGER,

PRIMARY KEY (IDAla),

FOREIGN KEY(IDClasse) REFERENCES Classe (IDClasse) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE TipoContainer (

IDTipoContainer INTEGER auto\_increment,

Nome VARCHAR(100),

PRIMARY KEY (IDTipoContainer));

CREATE TABLE Container (

IDContainer INTEGER auto\_increment,

Descricao VARCHAR(100),

IDAla INTEGER,

IDTipoContainer INTEGER,

PRIMARY KEY (IDContainer),

FOREIGN KEY(IDAla) REFERENCES Ala (IDAla) ON UPDATE RESTRICT ON DELETE RESTRICT,

FOREIGN KEY(IDTipoContainer) REFERENCES TipoContainer (IDTipoContainer) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Animal (

IDAnimal INTEGER auto\_increment,

Cor VARCHAR(100),

Altura NUMERIC(4, 2),

Nome VARCHAR(100),

IDEspecie INTEGER,

IDClasse INTEGER,

IDContainer INTEGER,

PRIMARY KEY (IDAnimal),

FOREIGN KEY(IDEspecie) REFERENCES Especie (IDEspecie) ON UPDATE RESTRICT ON DELETE RESTRICT,

FOREIGN KEY(IDClasse) REFERENCES Classe (IDClasse) ON UPDATE RESTRICT ON DELETE RESTRICT,

FOREIGN KEY(IDContainer) REFERENCES Container (IDContainer) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Atendimento (

DataInicioAtendimento DATE,

IDAnimal INTEGER,

IDVeterinario INTEGER,

PRIMARY KEY(DataInicioAtendimento,IDAnimal,IDVeterinario),

FOREIGN KEY(IDAnimal) REFERENCES Animal (IDAnimal) ON UPDATE RESTRICT ON DELETE RESTRICT,

FOREIGN KEY(IDVeterinario) REFERENCES Funcionario (IDFuncionario) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Consulta (

IDConsulta INTEGER auto\_increment,

Dia DATE,

Hora TIME,

DataInicioAtendimento DATE,

IDAnimal INTEGER,

IDVeterinario INTEGER,

PRIMARY KEY (IDConsulta),

FOREIGN KEY(DataInicioAtendimento, IDAnimal, IDVeterinario) REFERENCES Atendimento (DataInicioAtendimento,IDAnimal,IDVeterinario) ON UPDATE RESTRICT ON DELETE RESTRICT);

CREATE TABLE Limpeza\_Container (

IDLimpezaContainer INTEGER auto\_increment,

IDContainer INTEGER,

IDFuncionario INTEGER,

Dia DATE,

Horario TIME,

PRIMARY KEY(IDLimpezaContainer,IDContainer,IDFuncionario),

FOREIGN KEY(IDContainer) REFERENCES Container (IDContainer),

FOREIGN KEY(IDFuncionario) REFERENCES Funcionario (IDFuncionario) ON UPDATE RESTRICT ON DELETE RESTRICT);

## **SCRIPT SQL DE CRIAÇão das PROCEDUREs**

USE Zoologico;

**-- ESPECIE**

**-- PROCEDURE PARA INCLUIR UMA ESPECIE**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ESPECIE;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ESPECIE(IN PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Especie WHERE nome = PNOME)) THEN

BEGIN

INSERT INTO Especie(Nome) VALUES (PNOME);

SET MENSAGEM = CONCAT("A especie '", PNOME, "' foi incluida com sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. A especie '",PNOME,"' ja esta cadastrada.");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UMA ESPECIE**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_ESPECIE;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_ESPECIE(IN PIDESPECIE INTEGER, PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Especie WHERE Especie.IDEspecie = PIDESPECIE)) THEN

BEGIN

SET NOME = (SELECT Especie.Nome FROM Especie WHERE Especie.IDEspecie = PIDESPECIE);

IF (NOT EXISTS(SELECT 1 FROM Especie WHERE Especie.nome = PNOME AND Especie.IDEspecie <> PIDESPECIE)) THEN

BEGIN

UPDATE Especie SET Nome = PNOME WHERE Especie.IDEspecie = PIDESPECIE;

SET MENSAGEM = CONCAT("A especie '",NOME,"' foi alterada com sucesso para '",PNOME,"'.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da especie '",NOME,". A especie '",PNOME,"' ja esta cadastrada.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da especie. Nao existe especie cadastrada no banco de dados com ID = ",PIDESPECIE,".");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA ESPECIE**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ESPECIE;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ESPECIE(IN PIDESPECIE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Especie WHERE Especie.IDEspecie = PIDESPECIE)) THEN

BEGIN

SET NOME = (SELECT Especie.Nome FROM Especie WHERE Especie.IDEspecie = PIDESPECIE);

IF (NOT EXISTS(SELECT 1 FROM Especie INNER JOIN Animal ON Especie.IDEspecie = Animal.IDEspecie WHERE Especie.IDEspecie = PIDESPECIE)) THEN

BEGIN

DELETE FROM Especie WHERE Especie.IDEspecie = PIDESPECIE;

SET MENSAGEM = CONCAT("A especie '",NOME,"' foi excluida com Sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao da especie. A especie '",NOME,"' possui vinculos com algum animal.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao existe especie cadastrada no banco de dados com ID = ",PIDESPECIE,".");

END IF;

END$

DELIMITER ;

**-- CLASSE**

**-- PROCEDURE PARA INCLUIR UMA CLASSE**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_CLASSE;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_CLASSE (IN PNOME VARCHAR(100), PDESCRICAO VARCHAR(200), PCARACTERISTICAS VARCHAR(200), OUT MENSAGEM VARCHAR(200))

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Classe WHERE Classe.Nome = PNOME)) THEN

BEGIN

INSERT INTO Classe(Descricao, Nome, Caracteristicas) VALUES (PDESCRICAO, PNOME, PCARACTERISTICAS);

SET MENSAGEM = CONCAT("A classe '", PNOME, "' foi incluida com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. A classe '",PNOME,"' ja esta cadastrada.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UMA CLASSE**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_CLASSE;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_CLASSE (IN PIDCLASSE INTEGER, PNOME VARCHAR(100), PDESCRICAO VARCHAR(200), PCARACTERISTICAS VARCHAR(200), OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

SET NOME = (SELECT Classe.Nome FROM Classe WHERE Classe.IDClasse = PIDCLASSE);

IF (NOT EXISTS(SELECT 1 FROM Classe WHERE Classe.Nome = PNOME AND Classe.IDClasse <> PIDCLASSE)) THEN

BEGIN

UPDATE Classe SET Descricao = PDESCRICAO, Nome = PNOME, Caracteristicas = PCARACTERISTICAS WHERE IDClasse = PIDCLASSE;

SET MENSAGEM = CONCAT("A classe '",NOME,"' foi alterada com sucesso para '",PNOME,"'.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da classe '",NOME,"'. A classe '",PNOME,"' ja esta cadastrada.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da classe. Nao existe classe cadastrada no banco de dados com ID = ",PIDCLASSE,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA CLASSE**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_CLASSE;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_CLASSE (IN PIDCLASSE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

SET NOME = (SELECT Classe.Nome FROM Classe WHERE Classe.IDClasse = PIDCLASSE);

IF ((NOT EXISTS (SELECT 1 FROM Animal WHERE Animal.IDClasse = PIDCLASSE)) AND (NOT EXISTS (SELECT 1 FROM Ala WHERE IDClasse = PIDCLASSE))) THEN

BEGIN

DELETE FROM Classe WHERE IDClasse = PIDCLASSE;

SET MENSAGEM = CONCAT("A classe '",NOME,"' foi excluida com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao da classe. A classe '",NOME,"' possui vinculos com algum animal ou alguma ala.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe classe cadastrada no banco de dados com ID = ",PIDCLASSE,".");

END;

END IF;

END$

DELIMITER ;

**-- ANIMAL**

**-- PROCEDURE PARA INCLUIR UM ANIMAL**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ANIMAL;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ANIMAL (IN PNOME VARCHAR(100), PCOR VARCHAR(100), PALTURA NUMERIC(4,2), PIDESPECIE INTEGER, PIDCLASSE INTEGER, PIDCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Especie WHERE Especie.IDEspecie = PIDESPECIE)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Container WHERE Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

INSERT INTO Animal(Nome, Cor, Altura, IDEspecie, IDClasse, IDContainer) VALUES (PNOME, PCOR, PALTURA, PIDESPECIE, PIDCLASSE, PIDCONTAINER);

SET MENSAGEM = CONCAT("O animal '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha container com ID = ",PIDCONTAINER," cadastrado no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha classe com ID = ",PIDCLASSE," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha especie com ID = ",PIDESPECIE," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM ANIMAL**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_ANIMAL;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_ANIMAL (IN PIDANIMAL INTEGER, PNOME VARCHAR(100), PCOR VARCHAR(100), PALTURA NUMERIC(4,2), PIDESPECIE INTEGER, PIDCLASSE INTEGER, PIDCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Animal WHERE Animal.IDAnimal = PIDANIMAL)) THEN

BEGIN

SET NOME = (SELECT Animal.Nome FROM Animal WHERE Animal.IDAnimal = PIDANIMAL);

IF (EXISTS (SELECT 1 FROM Especie WHERE Especie.IDEspecie = PIDESPECIE)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Container WHERE Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

UPDATE Animal SET Nome = PNOME, Cor = PCOR, Altura = PALTURA, IDEspecie = PIDESPECIE, IDClasse = PIDCLASSE, IDContainer = PIDCONTAINER WHERE IDAnimal = PIDANIMAL;

SET MENSAGEM = CONCAT("O animal '",NOME,"' teve seus dados alterados com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do animal '",NOME,"'. Nao ha container com ID = ",PIDCONTAINER," cadastrado no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do animal '",NOME,"'. Nao ha classe com ID = ",PIDCLASSE," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do animal '",NOME,"'. Nao ha especie com ID = ",PIDESPECIE," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do animal do animal. Nao existe animal cadastrado no banco de dados com ID = ",PIDANIMAL,".");

END;

END IF;

END $

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM ANIMAL**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ANIMAL;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ANIMAL (IN PIDANIMAL INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Animal WHERE Animal.IDAnimal = PIDANIMAL)) THEN

BEGIN

SET NOME = (SELECT Animal.Nome FROM Animal WHERE Animal.IDAnimal = PIDANIMAL);

IF (NOT EXISTS (SELECT 1 FROM Consulta WHERE Consulta.IDAnimal = PIDANIMAL)) THEN

BEGIN

DELETE FROM Animal WHERE IDAnimal = PIDANIMAL;

SET MENSAGEM = CONCAT("O animal '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do animal. O animal '",NOME,"' possui registros de consultas.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe animal cadastrado no banco de dados com ID = ",PID,".");

END;

END IF;

END$

DELIMITER ;

**-- ALA**

**-- PROCEDURE PARA INCLUIR UMA ALA**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ALA;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ALA (IN PNOME VARCHAR(100), PIDCLASSE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Ala WHERE Ala.Nome = PNOME)) THEN

BEGIN

INSERT INTO Ala(Nome, IDClasse) VALUES (PNOME, PIDCLASSE);

SET MENSAGEM = CONCAT("A ala '", PNOME, "' foi incluida com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. A ala '",PNOME,"' ja esta cadastrada.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha classe com ID = ",PIDCLASSE," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_ALA;

**-- PROCEDURE PARA ALTERAR UMA ALA**

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_ALA (IN PIDALA INTEGER, PNOME VARCHAR(100), PIDCLASSE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Ala WHERE Ala.IDAla = PIDALA)) THEN

BEGIN

SET NOME = (SELECT Ala.Nome FROM Ala WHERE Ala.IDAla = PIDALA);

IF (EXISTS (SELECT 1 FROM Classe WHERE Classe.IDClasse = PIDCLASSE)) THEN

BEGIN

IF (NOT EXISTS(SELECT 1 FROM Ala WHERE Ala.Nome = PNOME AND Ala.IDAla <> PIDALA)) THEN

BEGIN

UPDATE Ala SET Nome = PNOME, IDClasse = PIDCLASSE WHERE IDAla = PIDALA;

SET MENSAGEM = CONCAT("A ala '",NOME,"' foi alterada com sucesso para '",PNOME,"'.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da ala '",NOME,"'. A ala '",PNOME,"' ja esta cadastrada.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da ala '",NOME,"'. Nao ha classe com ID = ",PIDCLASSE," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da ala. Nao existe ala cadastrada no banco de dados com ID = ",PIDALA,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA ALA**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ALA;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ALA (IN PIDALA INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Ala WHERE Ala.IDAla = PIDALA)) THEN

BEGIN

SET NOME = (SELECT Ala.Nome FROM Ala WHERE Ala.IDAla = PIDALA);

IF (NOT EXISTS (SELECT 1 FROM Container WHERE Container.IDAla = PIDALA)) THEN

BEGIN

DELETE FROM Ala WHERE IDAla = PIDALA;

SET MENSAGEM = CONCAT("A ala '",NOME,"' foi excluida com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao da ala. A ala '",NOME,"' possui vinculos com algum container.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe ala cadastrada no banco de dados com ID = ",PIDALA,".");

END;

END IF;

END$

DELIMITER ;

**-- TIPOCONTAINER**

**-- PROCEDURE PARA INCLUIR UM TIPOCONTAINER**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_TIPOCONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_TIPOCONTAINER(IN PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

IF (NOT EXISTS (SELECT 1 FROM TipoContainer WHERE nome = PNOME)) THEN

BEGIN

INSERT INTO TipoContainer(Nome) VALUES (PNOME);

SET MENSAGEM = CONCAT("O tipo de container '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O tipo de container '",PNOME,"' ja esta cadastrado.");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM TIPOCONTAINER**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_TIPOCONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_TIPOCONTAINER(IN PIDTIPOCONTAINER INTEGER, PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER)) THEN

BEGIN

SET NOME = (SELECT TipoContainer.Nome FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER);

IF (NOT EXISTS(SELECT 1 FROM TipoContainer WHERE TipoContainer.nome = PNOME AND TipoContainer.IDTipoContainer <> PIDTIPOCONTAINER)) THEN

BEGIN

UPDATE TipoContainer SET Nome = PNOME WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER;

SET MENSAGEM = CONCAT("O tipo de container '",NOME,"' foi alterado com sucesso para '",PNOME,"'.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do tipo de container '",NOME,". O tipo de container '",PNOME,"' ja esta cadastrado.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do tipo de container. Nao existe tipo de container cadastrado no banco de dados com ID = ",PIDTIPOCONTAINER,".");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM TIPOCONTAINER**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_TIPOCONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_TIPOCONTAINER(IN PIDTIPOCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER)) THEN

BEGIN

SET NOME = (SELECT TipoContainer.Nome FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER);

IF (NOT EXISTS(SELECT 1 FROM Container WHERE Container.IDTipoContainer = PIDTIPOCONTAINER)) THEN

BEGIN

DELETE FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER;

SET MENSAGEM = CONCAT("O tipo de container '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do tipo de container. O tipo de container '",NOME,"' possui vinculos com algum container.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao existe tipo de container cadastrado no banco de dados com ID = ",PIDTIPOCONTAINER,".");

END IF;

END$

DELIMITER ;

**-- CONTAINER**

**-- PROCEDURE PARA INCLUIR UM CONTAINER**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_CONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_CONTAINER (IN PDESCRICAO VARCHAR(100), PIDALA INTEGER, PIDTIPOCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Ala WHERE Ala.IDAla = PIDALA)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER)) THEN

BEGIN

INSERT INTO Container(Descricao, IDAla, IDTipoContainer) VALUES (PDESCRICAO, PIDALA, PIDTIPOCONTAINER);

SET MENSAGEM = CONCAT("A container foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha tipo de cotainer com ID = ",PIDTIPOCONTAINER," cadastrado no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha ala com ID = ",PIDALA," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_CONTAINER;

**-- PROCEDURE PARA ALTERAR UM CONTAINER**

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_CONTAINER (IN PIDCONTAINER INTEGER, PDESCRICAO VARCHAR(100), PIDALA INTEGER, PIDTIPOCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS(SELECT 1 FROM Container WHERE Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Ala WHERE Ala.IDAla = PIDALA)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM TipoContainer WHERE TipoContainer.IDTipoContainer = PIDTIPOCONTAINER)) THEN

BEGIN

UPDATE Container SET Descricao = PDESCRICAO, IDAla = PIDALA, IDTipoContainer = PIDTIPOCONTAINER WHERE IDContainer = PIDCONTAINER;

SET MENSAGEM = CONCAT("A container '",PIDCONTAINER,"' teve seus dados alterados com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da container '",PIDCONTAINER,"'. Nao ha tipo de container com ID = ",PIDALA," cadastrado no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da container '",PIDCONTAINER,"'. Nao ha ala com ID = ",PIDALA," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do container. Nao existe container cadastrada no banco de dados com ID = ",PIDCONTAINER,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM CONTAINER**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_CONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_CONTAINER (IN PIDCONTAINER INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Container WHERE Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Animal WHERE Animal.IDContainer = PIDCONTAINER)) THEN

BEGIN

DELETE FROM Container WHERE IDContainer = PIDCONTAINER;

SET MENSAGEM = CONCAT("A container '",PIDCONTAINER,"' foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao da container. A container '",PIDCONTAINER,"' possui vinculos com algum animal.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe container cadastrado no banco de dados com ID = ",PIDCONTAINER,".");

END;

END IF;

END$

DELIMITER ;

**-- ESTADO**

**-- PROCEDURE PARA INCLUIR UM ESTADO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ESTADO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ESTADO(IN PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Estado WHERE nome = PNOME)) THEN

BEGIN

INSERT INTO Estado(Nome) VALUES (PNOME);

SET MENSAGEM = CONCAT("O estado '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O estado '",PNOME,"' ja esta cadastrado.");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM ESTADO**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_ESTADO;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_ESTADO(IN PIDESTADO INTEGER, PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Estado WHERE Estado.IDEstado = PIDESTADO)) THEN

BEGIN

SET NOME = (SELECT Estado.Nome FROM Estado WHERE Estado.IDEstado = PIDESTADO);

IF (NOT EXISTS(SELECT 1 FROM Estado WHERE Estado.nome = PNOME AND Estado.IDEstado <> PIDESTADO)) THEN

BEGIN

UPDATE Estado SET Nome = PNOME WHERE Estado.IDEstado = PIDESTADO;

SET MENSAGEM = CONCAT("O estado '",NOME,"' foi alterado com sucesso para '",PNOME,"'.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do estado '",NOME,". O estado '",PNOME,"' ja esta cadastrado.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do estado. Nao existe estado cadastrado no banco de dados com ID = ",PIDESTADO,".");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA ESTADO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ESTADO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ESTADO(IN PIDESTADO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Estado WHERE Estado.IDEstado = PIDESTADO)) THEN

BEGIN

SET NOME = (SELECT Estado.Nome FROM Estado WHERE Estado.IDEstado = PIDESTADO);

IF (NOT EXISTS(SELECT 1 FROM Cidade WHERE Cidade.IDEstado = PIDESTADO)) THEN

BEGIN

DELETE FROM Estado WHERE Estado.IDEstado = PIDESTADO;

SET MENSAGEM = CONCAT("O estado '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do estado. O estado '",NOME,"' possui vinculos com alguma cidade.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao existe estado cadastrado no banco de dados com ID = ",PIDESTADO,".");

END IF;

END$

DELIMITER ;

**-- CIDADE**

**-- PROCEDURE PARA INCLUIR UMA CIDADE**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_CIDADE;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_CIDADE (IN PNOME VARCHAR(100), PIDESTADO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Estado WHERE Estado.IDEstado = PIDESTADO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Cidade WHERE Cidade.Nome = PNOME AND Cidade.IDEstado = PIDESTADO)) THEN

BEGIN

INSERT INTO Cidade(Nome, IDEstado) VALUES (PNOME, PIDESTADO);

SET MENSAGEM = CONCAT("A cidade '", PNOME, "' foi incluida com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. A cidade '",PNOME,"' ja esta cadastrada para o estado ",(SELECT Estado.Nome FROM Estado WHERE Estado.IDEstado = PIDESTADO),".");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha estado com ID = ",PIDESTADO," cadastrado no banco de dados.");

END;

END IF;

END$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_CIDADE;

**-- PROCEDURE PARA ALTERAR UMA CIDADE**

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_CIDADE (IN PIDCIDADE INTEGER, PNOME VARCHAR(100), PIDESTADO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE)) THEN

BEGIN

SET NOME = (SELECT Cidade.Nome FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE);

IF (EXISTS (SELECT 1 FROM Estado WHERE Estado.IDEstado = PIDESTADO)) THEN

BEGIN

IF (NOT EXISTS(SELECT 1 FROM Cidade WHERE Cidade.Nome = PNOME AND Cidade.IDEstado = PIDESTADO AND Cidade.IDCidade <> PIDCIDADE)) THEN

BEGIN

UPDATE Cidade SET Nome = PNOME, IDEstado = PIDESTADO WHERE IDCidade = PIDCIDADE;

SET MENSAGEM = CONCAT("A cidade '",NOME,"' foi alterada com sucesso para '",PNOME,"'.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da cidade '",NOME,"'. A cidade '",PNOME,"' ja esta cadastrada para o estado ",(SELECT Estado.Nome FROM Estado WHERE Estado.IDEstado = PIDESTADO),".");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da cidade '",NOME,"'. Nao ha estado com ID = ",PIDESTADO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da cidade. Nao existe cidade cadastrada no banco de dados com ID = ",PIDCIDADE,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA CIDADE**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_CIDADE;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_CIDADE (IN PIDCIDADE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE)) THEN

BEGIN

SET NOME = (SELECT Cidade.Nome FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE);

IF (NOT EXISTS (SELECT 1 FROM Bairro WHERE Bairro.IDCidade = PIDCIDADE)) THEN

BEGIN

DELETE FROM Cidade WHERE IDCidade = PIDCIDADE;

SET MENSAGEM = CONCAT("A cidade '",NOME,"' foi excluida com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao da cidade. A cidade '",NOME,"' possui vinculos com algum bairro.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe cidade cadastrada no banco de dados com ID = ",PIDCIDADE,".");

END;

END IF;

END$

DELIMITER ;

**-- BAIRRO**

**-- PROCEDURE PARA INCLUIR UM BAIRRO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_BAIRRO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_BAIRRO (IN PNOME VARCHAR(100), PIDCIDADE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Bairro WHERE Bairro.Nome = PNOME AND Bairro.IDCidade = PIDCIDADE)) THEN

BEGIN

INSERT INTO Bairro(Nome, IDCidade) VALUES (PNOME, PIDCIDADE);

SET MENSAGEM = CONCAT("O bairro '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O bairro '",PNOME,"' ja esta cadastrada para o cidade ",(SELECT Cidade.Nome FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE),".");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha cidade com ID = ",PIDCIDADE," cadastrado no banco de dados.");

END;

END IF;

END$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_BAIRRO;

**-- PROCEDURE PARA ALTERAR UM BAIRRO**

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_BAIRRO (IN PIDBAIRRO INTEGER, PNOME VARCHAR(100), PIDCIDADE INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO)) THEN

BEGIN

SET NOME = (SELECT Bairro.Nome FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO);

IF (EXISTS (SELECT 1 FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE)) THEN

BEGIN

IF (NOT EXISTS(SELECT 1 FROM Bairro WHERE Bairro.Nome = PNOME AND Bairro.IDCidade = PIDCIDADE AND Bairro.IDBairro <> PIDBAIRRO)) THEN

BEGIN

UPDATE Bairro SET Nome = PNOME, IDCidade = PIDCIDADE WHERE IDBairro = PIDBAIRRO;

SET MENSAGEM = CONCAT("O bairro '",NOME,"' foi alterado com sucesso para '",PNOME,"'.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da bairro '",NOME,"'. A bairro '",PNOME,"' ja esta cadastrada para o cidade ",(SELECT Cidade.Nome FROM Cidade WHERE Cidade.IDCidade = PIDCIDADE),".");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da bairro '",NOME,"'. Nao ha cidade com ID = ",PIDCIDADE," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da bairro. Nao existe bairro cadastrado no banco de dados com ID = ",PIDBAIRRO,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM BAIRRO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_BAIRRO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_BAIRRO (IN PIDBAIRRO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO)) THEN

BEGIN

SET NOME = (SELECT Bairro.Nome FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO);

IF (NOT EXISTS (SELECT 1 FROM Endereco WHERE Endereco.IDBairro = PIDBAIRRO)) THEN

BEGIN

DELETE FROM Bairro WHERE IDBairro = PIDBAIRRO;

SET MENSAGEM = CONCAT("O bairro '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do bairro. O bairro '",NOME,"' possui vinculos com algum endereco.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe bairro cadastrado no banco de dados com ID = ",PIDBAIRRO,".");

END;

END IF;

END$

DELIMITER ;

**-- ENDERECO**

**-- PROCEDURE PARA INCLUIR UM ENDERECO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ENDERECO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ENDERECO (IN PCEP CHAR(9), PRUA VARCHAR(100), PNUMERO SMALLINT, PIDBAIRRO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Endereco WHERE Endereco.Rua = PRUA AND Endereco.Numero = PNUMERO AND Endereco.CEP = PCEP AND Endereco.IDBairro = PIDBAIRRO)) THEN

BEGIN

INSERT INTO Endereco(CEP, Rua, Numero, IDBairro) VALUES (PCEP, PRUA, PNUMERO, PIDBAIRRO);

SET MENSAGEM = CONCAT("O endereco foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Esse endereco ja esta cadastrado.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha bairro com ID = ",PIDBAIRRO," cadastrado no banco de dados.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM ENDERECO**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_ENDERECO;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_ENDERECO (IN PIDENDERECO INTEGER,PCEP CHAR(9), PRUA VARCHAR(100), PNUMERO SMALLINT, PIDBAIRRO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS(SELECT 1 FROM Endereco WHERE Endereco.IDEndereco = PIDENDERECO)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Bairro WHERE Bairro.IDBairro = PIDBAIRRO)) THEN

BEGIN

IF (NOT EXISTS(SELECT 1 FROM Endereco WHERE Endereco.CEP = PCEP AND Endereco.Rua = PRUA AND Endereco.Numero = PNUMERO AND Endereco.IDBairro = PIDBAIRRO AND Endereco.IDEndereco <> PIDENDERECO)) THEN

BEGIN

UPDATE Endereco SET CEP = PCEP, Rua = PRUA, Numero = PNUMERO, IDBairro = PIDBAIRRO WHERE IDEndereco = PIDENDERECO;

SET MENSAGEM = CONCAT("O endereco foi alterado com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da endereco. Esse endereco ja esta cadastrado.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da endereco. Nao ha bairro com ID = ",PIDBAIRRO," cadastrado no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da endereco. Nao existe endereco cadastrado no banco de dados com ID = ",PIDENDERECO,".");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM ENDERECO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ENDERECO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ENDERECO (IN PIDENDERECO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Endereco WHERE Endereco.IDEndereco = PIDENDERECO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.IDEndereco = PIDENDERECO)) THEN

BEGIN

DELETE FROM Endereco WHERE IDEndereco = PIDENDERECO;

SET MENSAGEM = CONCAT("O endereco foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do endereco. O endereco possui vinculos com algum funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe endereco cadastrado no banco de dados com ID = ",PIDENDERECO,".");

END;

END IF;

END$

DELIMITER ;

**-- CARGO**

**-- PROCEDURE PARA INCLUIR UM CARGO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_CARGO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_CARGO(IN PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Cargo WHERE nome = PNOME)) THEN

BEGIN

INSERT INTO Cargo(Nome) VALUES (PNOME);

SET MENSAGEM = CONCAT("O cargo '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O cargo '",PNOME,"' ja esta cadastrado.");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM CARGO**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_CARGO;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_CARGO(IN PIDCARGO INTEGER, PNOME VARCHAR(100), OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Cargo WHERE Cargo.IDCargo = PIDCARGO)) THEN

BEGIN

SET NOME = (SELECT Cargo.Nome FROM Cargo WHERE Cargo.IDCargo = PIDCARGO);

IF (NOT EXISTS(SELECT 1 FROM Cargo WHERE Cargo.nome = PNOME AND Cargo.IDCargo <> PIDCARGO)) THEN

BEGIN

UPDATE Cargo SET Nome = PNOME WHERE Cargo.IDCargo = PIDCARGO;

SET MENSAGEM = CONCAT("O cargo '",NOME,"' foi alterado com sucesso para '",PNOME,"'.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do cargo '",NOME,". O cargo '",PNOME,"' ja esta cadastrado.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do cargo. Nao existe cargo cadastrado no banco de dados com ID = ",PIDCARGO,".");

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA CARGO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_CARGO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_CARGO(IN PIDCARGO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Cargo WHERE Cargo.IDCargo = PIDCARGO)) THEN

BEGIN

SET NOME = (SELECT Cargo.Nome FROM Cargo WHERE Cargo.IDCargo = PIDCARGO);

IF (NOT EXISTS(SELECT 1 FROM Funcionario WHERE Funcionario.IDCargo = PIDCARGO)) THEN

BEGIN

DELETE FROM Cargo WHERE Cargo.IDCargo = PIDCARGO;

SET MENSAGEM = CONCAT("O cargo '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do cargo. O cargo '",NOME,"' possui vinculos com algum funcionario.");

END IF;

END;

ELSE

SET MENSAGEM = CONCAT("Nao existe cargo cadastrado no banco de dados com ID = ",PIDCARGO,".");

END IF;

END$

DELIMITER ;

**-- FUNCIONARIO**

**-- PROCEDURE PARA INCLUIR UM FUNCIONARIO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_FUNCIONARIO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_FUNCIONARIO (IN PNOME VARCHAR(100), PCPF CHAR(14), PRG CHAR(12), PDATANASCIMENTO DATE, PCRMV CHAR(9), PIDCARGO INTEGER, PIDENDERECO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Cargo WHERE Cargo.IDCargo = PIDCARGO)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Endereco WHERE Endereco.IDEndereco = PIDENDERECO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.CPF = PCPF)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.RG = PRG)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.CRMV IS NOT NULL AND Funcionario.CRMV = PCRMV)) THEN

BEGIN

INSERT INTO Funcionario(Nome, CPF, RG, DataNascimento, CRMV, IDCargo, IDEndereco) VALUES (PNOME, PCPF, PRG, PDATANASCIMENTO, PCRMV, PIDCARGO, PIDENDERECO);

SET MENSAGEM = CONCAT("O funcionario '", PNOME, "' foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O CRMV '",PCRMV,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O RG '",PRG,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O CPF '",PCPF,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha endereco com ID = ",PIDENDERECO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha cargo com ID = ",PIDCARGO," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UM FUNCIONARIO**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_FUNCIONARIO;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_FUNCIONARIO (IN PIDFUNCIONARIO INTEGER, PNOME VARCHAR(100), PCPF VARCHAR(100), PRG NUMERIC(4,2), PDATANASCIMENTO DATE, PCRMV CHAR(9), PIDCARGO INTEGER, PIDENDERECO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS(SELECT 1 FROM Funcionario WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO)) THEN

BEGIN

SET NOME = (SELECT Funcionario.Nome FROM Funcionario WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO);

IF (EXISTS (SELECT 1 FROM Cargo WHERE Cargo.IDCargo = PIDCARGO)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Endereco WHERE Endereco.IDEndereco = PIDENDERECO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.CPF = PCPF AND Funcionario.IDFuncionario <> PIDFUNCIONARIO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.RG = PRG AND Funcionario.IDFuncionario <> PIDFUNCIONARIO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.CRMV IS NOT NULL AND Funcionario.CRMV = PCRMV AND Funcionario.IDFuncionario <> PIDFUNCIONARIO)) THEN

BEGIN

UPDATE Funcionario SET Nome = PNOME, CPF = PCPF, RG = PRG, DataNascimento = PDATANASCIMENTO, CRMV = PCRMV, IDCargo = PIDCARGO, IDEndereco = PIDENDERECO;

SET MENSAGEM = CONCAT("O funcionario '", PNOME, "' foi alterado com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario '",NOME,"'. O CRMV '",PCRMV,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario '",NOME,"'. O RG '",PRG,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario '",NOME,"'. O CPF '",PCPF,"' ja esta cadastrado para outro funcionario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario '",NOME,"'. Nao ha endereco com ID = ",PIDENDERECO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario '",NOME,"'. Nao ha cargo com ID = ",PIDCARGO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao do funcionario do funcionario. Nao existe funcionario cadastrado no banco de dados com ID = ",PIDFUNCIONARIO,".");

END;

END IF;

END $

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM FUNCIONARIO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_FUNCIONARIO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_FUNCIONARIO (IN PIDFUNCIONARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

DECLARE NOME VARCHAR(100);

IF (EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO)) THEN

BEGIN

SET NOME = (SELECT Funcionario.Nome FROM Funcionario WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO);

IF (NOT EXISTS (SELECT 1 FROM Consulta WHERE Consulta.IDVeterinario = PIDFUNCIONARIO)) THEN

BEGIN

DELETE FROM Funcionario WHERE IDFuncionario = PIDFUNCIONARIO;

SET MENSAGEM = CONCAT("O funcionario '",NOME,"' foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do funcionario. O funcionario '",NOME,"' possui registros de consultas.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe funcionario cadastrado no banco de dados com ID = ",PID,".");

END;

END IF;

END$

DELIMITER ;

**-- LIMPEZA\_CONTAINER**

**-- PROCEDURE PARA INCLUIR UMA LIMPEZA\_CONTAINER**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_LIMPEZA\_CONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_LIMPEZA\_CONTAINER (IN PIDCONTAINER INTEGER, PIDFUNCIONARIO INTEGER, PDIA DATE, PHORARIO TIME, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Container WHERE Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Funcionario INNER JOIN Cargo ON Cargo.IDCargo = Fucionario.IDCargo WHERE Funcionario.IDFuncionario = PIDFUNCIONARIO AND Cargo.Nome = 'Zelador')) THEN

BEGIN

INSERT INTO Limpeza\_Container(IDContainer, IDFuncionario, Dia, Horario) VALUES (PIDCONTAINER, PIDFUNCIONARIO, PDIA, PHORARIO);

SET MENSAGEM = CONCAT("A limpeza de container foi incluida com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O funcionario com ID ", PID," nao e um zelador.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha funcionario com ID = ",PIDFUNCIONARIO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha container com ID = ",PIDCONTAINER," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA ALTERAR UMA LIMPEZA\_CONTAINER**

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_LIMPEZA\_CONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_LIMPEZA\_CONTAINER (IN PIDLIMPEZACONTAINER INTEGER, PIDCONTAINER INTEGER, PIDFUNCIONARIO INTEGER, PDIA DATE, PHORARIO TIME, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS(SELECT 1 FROM Limpeza\_Container WHERE Limpeza\_Container.IDLimpezaContainer = PIDLIMPEZACONTAINER AND Limpeza\_Container.IDFuncionario = PIDFUNCIONARIO AND Limpeza\_Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

UPDATE Limpeza\_Container SET Dia = PDIA, Horario = PHORARIO WHERE Limpeza\_Container.IDLimpezaContainer = PIDLIMPEZACONTAINER AND Limpeza\_Container.IDFuncionario = PIDFUNCIONARIO AND Limpeza\_Container.IDContainer = PIDCONTAINER;

SET MENSAGEM = CONCAT("A limpeza de container foi alterada com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da limpeza de container da limpeza de container. Nao existe limpeza de container cadastrada no banco de dados com os dados inseridos.");

END;

END IF;

END $

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA LIMPEZA\_CONTAINER**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_LIMPEZA\_CONTAINER;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_LIMPEZA\_CONTAINER (IN PIDLIMPEZACONTAINER INTEGER, PIDCONTAINER INTEGER, PIDFUNCIONARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Limpeza\_Container WHERE Limpeza\_Container.IDLimpezaContainer = PIDLIMPEZACONTAINER AND Limpeza\_Container.IDFuncionario = PIDFUNCIONARIO AND Limpeza\_Container.IDContainer = PIDCONTAINER)) THEN

BEGIN

DELETE FROM Limpeza\_Container WHERE Limpeza\_Container.IDLimpezaContainer = PIDLIMPEZACONTAINER AND Limpeza\_Container.IDFuncionario = PIDFUNCIONARIO AND Limpeza\_Container.IDContainer = PIDCONTAINER;

SET MENSAGEM = CONCAT("A limpeza de container foi excluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe limpeza de container cadastrada no banco de dados com dos dados inseridos");

END;

END IF;

END$

DELIMITER ;

**-- ATENDIMENTO**

**-- PROCEDURE PARA INICIAR UM ATENDIMENTO**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_ATENDIMENTO;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_ATENDIMENTO (IN PDATAINICIOATENDIMENTO DATE, PIDANIMAL INTEGER, PIDVETERINARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Animal WHERE Animal.IDAnimal = PIDANIMAL)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Funcionario WHERE Funcionario.IDFuncionario = PIDVETERINARIO)) THEN

BEGIN

IF (EXISTS (SELECT 1 FROM Funcionario INNER JOIN Cargo ON Cargo.IDCargo = Funcionario.IDCargo WHERE Funcionario.IDFuncionario = PIDVETERINARIO AND Cargo.Nome = 'Veterinario')) THEN

BEGIN

IF (NOT EXISTS (SELECT COUNT(Atendimento.IDVeterinario) FROM Atendimento INNER JOIN (SELECT IDAnimal AS IDAnimal, MAX(DataInicioAtendimento) AS DataInicioAtendimento FROM Atendimento GROUP BY IDAnimal) AS Nova ON Nova.DataInicioAtendimento = Atendimento.DataInicioAtendimento AND Nova.IDAnimal = Atendimento.IDAnimal WHERE IDVeterinario = PIDVETERINARIO GROUP BY Atendimento.IDVeterinario HAVING COUNT(Atendimento.IDVeterinario) >= 15)) THEN

BEGIN

INSERT INTO Atendimento(DataInicioAtendimento, IDAnimal, IDVeterinario) VALUES (PDATAINICIOATENDIMENTO, PIDANIMAL, PIDVETERINARIO);

SET MENSAGEM = CONCAT("O atendimento foi incluido com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. A veterinario atingiu o limite de atendimento que eh 15 animais");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. O funcionario com ID = ",PIDVETERINARIO," nao e um veterinario.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha funcionario com ID = ",PIDVETERINARIO," cadastrada no banco de dados.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Nao ha animal com ID = ",PIDANIMAL," cadastrada no banco de dados.");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UM ATENDIMENTO**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_ATENDIMENTO;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_ATENDIMENTO (IN PDATAINICIOATENDIMENTO DATE, PIDANIMAL INTEGER, PIDVETERINARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Atendimento WHERE Atendimento.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Atendimento.IDAnimal = PIDANIMAL AND Atendimento.IDVeterinario = PIDVETERINARIO)) THEN

BEGIN

IF (NOT EXISTS (SELECT 1 FROM Consulta WHERE Consulta.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Consulta.IDAnimal = PIDANIMAL AND Consulta.IDVeterinario = PIDVETERINARIO)) THEN

BEGIN

DELETE FROM Atendimento WHERE Atendimento.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Atendimento.IDAnimal = PIDANIMAL AND Atendimento.IDVeterinario = PIDVETERINARIO;

SET MENSAGEM = CONCAT("O atendimento foi excluido com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a exclusao do atendimento. O atendimento possui registros de consultas.");

END;

END IF;

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe atendimento cadastrado no banco de dados informados");

END;

END IF;

END$

DELIMITER ;

**-- CONSULTA**

**-- PROCEDURE PARA INCLUIR UMA CONSULTA**

DROP PROCEDURE IF EXISTS SP\_INCLUIR\_CONSULTA;

DELIMITER $

CREATE PROCEDURE SP\_INCLUIR\_CONSULTA (IN PDIA DATE, PHORA TIME, PDATAINICIOATENDIMENTO DATE, PIDANIMAL INTEGER, PIDVETERINARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS (SELECT 1 FROM Atendimento WHERE Atendimento.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Atendimento.IDAnimal = PIDANIMAL AND Atendimento.IDVeterinario = PIDVETERINARIO AND DataInicioAtendimento = (SELECT MAX(DataInicioAtendimento) FROM Atendimento WHERE IDAnimal = PIDANIMAL))) THEN

BEGIN

INSERT INTO Consulta(Dia, Hora, DataInicioAtendimento, IDAnimal, IDVeterinario) VALUES (PDIA, PHORA, PDATAINICIOATENDIMENTO, PIDANIMAL, PIDVETERINARIO);

SET MENSAGEM = CONCAT("A consulta foi incluida com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a inclusao. Este veterinario nao atende atualmente esse animal.");

END;

END IF;

END$

DELIMITER ;

DROP PROCEDURE IF EXISTS SP\_ALTERAR\_CONSULTA;

**-- PROCEDURE PARA ALTERAR UMA CONSULTA**

DELIMITER $

CREATE PROCEDURE SP\_ALTERAR\_CONSULTA (IN PIDCONSULTA INTEGER, PDATAINICIOATENDIMENTO DATE, PIDANIMAL INTEGER, PIDVETERINARIO INTEGER, PDIA DATE, PHORA TIME, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS(SELECT 1 FROM Consulta WHERE Consulta.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Consulta.IDAnimal = PIDANIMAL AND Consulta.IDVeterinario = PIDVETERINARIO AND Contulta.IDConsulta = PIDCONSULTA)) THEN

BEGIN

UPDATE Consulta SET Dia = PDIA, Hora = PHORA WHERE DataInicioAtendimento = PDATAINICIOATENDIMENTO AND IDAnimal = PIDANIMAL AND IDVeterinario = PIDVETERINARIO AND IDConsulta = PIDCONSULTA;

SET MENSAGEM = CONCAT("A consulta foi alterada com sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao foi possivel realizar a alteracao da consulta. Nao existe consulta cadastrada no banco de dados informados");

END;

END IF;

END$

DELIMITER ;

**-- PROCEDURE PARA EXCLUIR UMA CONSULTA**

DROP PROCEDURE IF EXISTS SP\_EXCLUIR\_CONSULTA;

DELIMITER $

CREATE PROCEDURE SP\_EXCLUIR\_CONSULTA (IN PIDCONSULTA INTEGER, PDATAINICIOATENDIMENTO DATE, PIDANIMAL INTEGER, PIDVETERINARIO INTEGER, OUT MENSAGEM VARCHAR(200))

BEGIN

IF (EXISTS(SELECT 1 FROM Consulta WHERE Consulta.DataInicioAtendimento = PDATAINICIOATENDIMENTO AND Consulta.IDAnimal = PIDANIMAL AND Consulta.IDVeterinario = PIDVETERINARIO AND Consulta.IDConsulta = PIDCONSULTA)) THEN

BEGIN

DELETE FROM Consulta WHERE DataInicioAtendimento = PDATAINICIOATENDIMENTO AND IDAnimal = PIDANIMAL AND IDVeterinario = PIDVETERINARIO AND IDConsulta = PIDCONSULTA;

SET MENSAGEM = CONCAT("A consulta foi excluida com Sucesso.");

END;

ELSE

BEGIN

SET MENSAGEM = CONCAT("Nao existe consulta cadastrada no banco de dados com ID = ",PIDCONSULTA,".");

END;

END IF;

END$

DELIMITER ;

## **SCRIPT sql DE criação dos indíces**

CREATE INDEX INDEX\_NOME\_ESTADO ON Estado(Nome);

CREATE INDEX INDEX\_NOME\_CIDADE ON Cidade(Nome);

CREATE INDEX INDEX\_NOME\_BAIRRO ON Bairro(Nome);

CREATE INDEX INDEX\_RUA\_ENDERECO ON Endereco(Rua);

CREATE INDEX INDEX\_NOME\_FUNCIONARIO ON Funcionario(Nome);

CREATE INDEX INDEX\_NOME\_CARGO ON Cargo(Nome);

CREATE INDEX INDEX\_NOME\_CLASSE ON Classe(Nome);

CREATE INDEX INDEX\_NOME\_ESPECIE ON Especie(Nome);

CREATE INDEX INDEX\_NOME\_ANIMAL ON Animal(Nome);

CREATE INDEX INDEX\_NOME\_ALA ON Ala(Nome);

## **SCRIPT sql DE criação das views**

USE Zoologico;

**-- VIEW QUE RETORNA A QUANTIDADE DE CONSULTAS POR ANIMAL E VETERINÁRIO**

DROP VIEW IF EXISTS QTDE\_CONSULTAS;

CREATE VIEW QTDE\_CONSULTAS AS (

SELECT Funcionario.Nome AS Veterinario, Animal.Nome AS Animal, COUNT(\*) AS Consultas

FROM Consulta

INNER JOIN Funcionario ON Consulta.IDVeterinario = Funcionario.IDFuncionario

INNER JOIN Animal ON Consulta.IDAnimal = Animal.IDAnimal

GROUP BY Funcionario.IDFuncionario, Animal.IDAnimal

ORDER BY Funcionario.Nome, Animal.Nome ASC

);

**-- VIEW QUE RETORNA INFORMAÇÃO COMPLETA DAS CARACTERÍSTICAS E LOCALIZAÇÃO DO ANIMAIS**

DROP VIEW IF EXISTS INFO\_ANIMAIS;

CREATE VIEW INFO\_ANIMAIS AS (

SELECT Especie.Nome AS Especie, Classe.Nome AS Classe, Animal.Nome as Animal, Animal.Cor as Cor, Animal.Altura AS Altura, Container.IDContainer AS Container, Ala.Nome as Ala

FROM Animal

INNER JOIN Container ON Container.IDContainer = Animal.IDContainer

INNER JOIN Ala ON Container.IDAla = Ala.IDAla

INNER JOIN Classe ON Animal.IDClasse = Classe.IDClasse

INNER JOIN Especie ON Especie.IDEspecie = Animal.IDEspecie

ORDER BY Especie.Nome ASC

);

**-- VIEW QUE RETORNA A QUANTIDADE DE ANIMAIS POR ESPÉCIE**

DROP VIEW IF EXISTS QTDE\_ANIMAIS\_ESPECIE;

CREATE VIEW QTDE\_ANIMAIS\_ESPECIE AS (

SELECT Especie.Nome AS Especie, COUNT(Especie.IDEspecie) AS Animais

FROM Animal

INNER JOIN Especie ON Especie.IDEspecie = Animal.IDEspecie

GROUP BY Especie.IDEspecie

ORDER BY Especie.Nome ASC

);

**-- VIEW QUE RETORNA A QUANTIDADE DE ANIMAIS POR ALA**

DROP VIEW IF EXISTS QTDE\_ANIMAIS\_ALA;

CREATE VIEW QTDE\_ANIMAIS\_ALA AS (

SELECT Ala.Nome AS Ala, COUNT(Ala.IDAla) AS Animais

FROM Ala

INNER JOIN Container ON Container.IDAla = Ala.IDAla

INNER JOIN Animal ON Animal.IDContainer = Container.IDContainer

GROUP BY Ala.IDAla

ORDER BY Ala.Nome ASC

);

**-- VIEW QUE RETORNA A QUANTIDADE DE ANIMAIS POR CONTAINER**

DROP VIEW IF EXISTS QTDE\_ANIMAIS\_CONTAINER;

CREATE VIEW QTDE\_ANIMAIS\_CONTAINER AS (

SELECT Ala.Nome AS Ala, Container.IDContainer AS Container, COUNT(Container.IDContainer) AS Animais

FROM Container

INNER JOIN Ala ON Container.IDAla = Ala.IDAla

INNER JOIN Animal ON Animal.IDContainer = Container.IDContainer

GROUP BY Container.IDContainer

ORDER BY Ala.Nome, Container.IDContainer ASC

);

**-- VIEW QUE RETORNA OS FUNCIONÁRIOS E SEUS CARGOS**

DROP VIEW IF EXISTS FUNCIONARIOS\_CARGO;

CREATE VIEW FUNCIONARIOS\_CARGO AS (

SELECT Funcionario.Nome AS Funcionario, Cargo.Nome AS Cargo

FROM Funcionario

INNER JOIN Cargo ON Cargo.IDCargo = Funcionario.IDCargo

);

**-- VIEW QUE RETORNA A QUANTIDADE DE FUNCIONÁRIOS POR CARGO**

DROP VIEW IF EXISTS QTDE\_FUNCIONARIOS\_CARGO;

CREATE VIEW QTDE\_FUNCIONARIOS\_CARGO AS (

SELECT Cargo.Nome AS Cargo, COUNT(Cargo.IDCargo) AS Funcionarios

FROM Cargo

INNER JOIN Funcionario ON Cargo.IDCargo = Funcionario.IDCargo

GROUP BY Cargo.IDCargo

ORDER BY Cargo.Nome ASC

);

**-- VIEW QUE RETORNA A QUANTIDADE DE LIMPEZAS REALIZADAS NOS CONTAINERS POR FUNCIONÁRIO E CONTAINER**

DROP VIEW IF EXISTS QTDE\_LIMPEZA\_CONTAINER;

CREATE VIEW QTDE\_LIMPEZA\_CONTAINER AS (

SELECT Container.IDContainer AS Container, Funcionario.Nome AS Zelador, COUNT(\*)

FROM Limpeza\_Container

INNER JOIN Funcionario ON Funcionario.IDFuncionario = Limpeza\_Container.IDFuncionario

INNER JOIN Container ON Container.IDContainer = Limpeza\_Container.IDContainer

GROUP BY Container.IDContainer, Funcionario.IDFuncionario

ORDER BY Container.IDContainer, Funcionario.Nome

);

**-- VIEW QUE RETORNA A QUANTIDADE DE ANIMAIS QUE OS VETERINÁRIOS ESTÃO ATENDENDO NO MOMENTO**

DROP VIEW IF EXISTS QTDE\_ATENDIMENTO\_VETERINARIO;

CREATE VIEW QTDE\_ATENDIMENTO\_VETERINARIO AS (

SELECT Funcionario.Nome AS Veterinario, COUNT(Funcionario.IDFuncionario) AS Animais

FROM Funcionario

INNER JOIN Atendimento ON Atendimento.IDVeterinario = Funcionario.IDFuncionario

INNER JOIN Animal ON Animal.IDAnimal = Atendimento.IDAnimal

WHERE Atendimento.DataInicioAtendimento = (SELECT MAX(Atendimento.DataInicioAtendimento) FROM Atendimento WHERE Atendimento.IDAnimal = Animal.IDAnimal)

GROUP BY Funcionario.IDFuncionario

ORDER BY Funcionario.Nome ASC

);

# conclusão

A elaboração de trabalhos e projetos finais em disciplinas de áreas práticas é de extrema importância para revisão, verificação e fixação de conteúdos estudados.

O projeto apresentado além de permitir exercitar e praticar o conteúdo apresentado na disciplina durante o semestre, também permitiu o exercício da capacidade de análise e resolução de problemas, sendo ambos fatores de extrema importância para a atuação no mercado de trabalho.

A resolução do problema apresentada neste documento foi definida em conjunto, de acordo com análise individual e em equipe, podendo haver diferentes maneiras de se resolver o mesmo problema.