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**

**2 DESENVOLVIMENTO**.................................................................**5**

**2.1 MODELO DE ENTIDADE E RELACIONAMENTO**......................**6**

**2.2 MODELO RELACIONAL**..............................................................**7**

**2.3 SCRIPT SQL DE CRIAÇÃO DA BASE DE DADOS** ...................**8**

**2.4 SCRIPT SQL DE CRIAÇÃO DAS STORED PROCEDURES**..**13**

**2.5 SCRIPT SQL DE CRIAÇÃO DOS INDÍCES**...............................**14**

**2.6 SCRIPT SQL DE CRIAÇÃO DAS VIEWS**..................................**15**

**3 CONCLUSÃO**..............................................................................**16**

# 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.

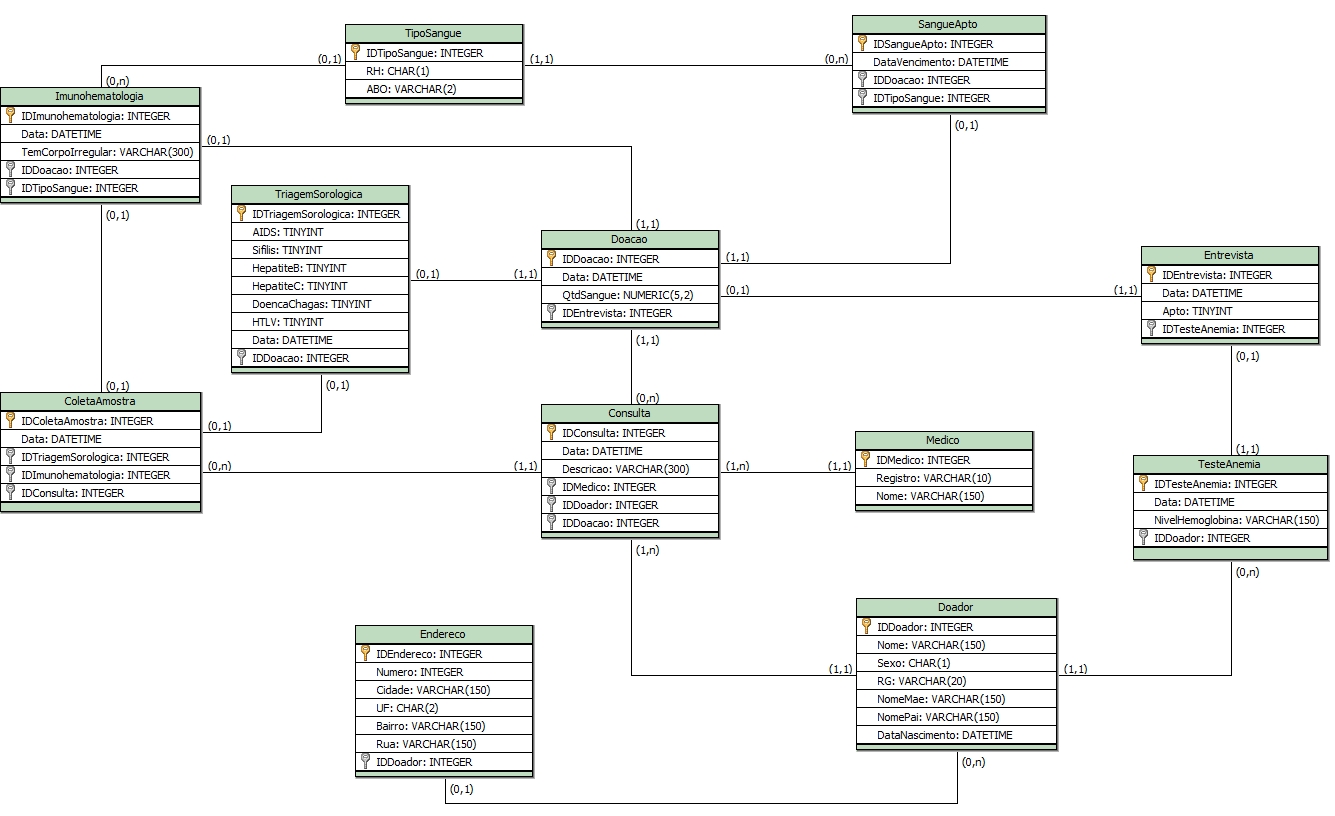
- A exclusão de dados apenas é permitida, se houver casos de tuplas apenas com chaves.

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.

## **C:\Users\Maths\Desktop\Conceitual (1).jpgMODELO DE ENTIDADE E RELACIONAMENTO**

## 

## **MODELO RELACIONAL**



## 

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

CREATE DATABASE BANCOSANGUE;

USE BANCOSANGUE;

CREATE TABLE Doador (

IDDoador INTEGER NOT NULL AUTO\_INCREMENT,

Nome VARCHAR(150),

Sexo CHAR(1),

RG VARCHAR(20),

NomeMae VARCHAR(150),

NomePai VARCHAR(150),

DataNascimento DATETIME,

PRIMARY KEY(IDDoador)

);

CREATE TABLE Endereco (

IDEndereco INTEGER NOT NULL AUTO\_INCREMENT,

Numero INTEGER,

Cidade VARCHAR(150),

UF CHAR(2),

Bairro VARCHAR(150),

Rua VARCHAR(150),

IDDoador INTEGER NOT NULL,

PRIMARY KEY(IDEndereco),

FOREIGN KEY(IDDoador) REFERENCES Doador (IDDoador)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE TesteAnemia (

IDTesteAnemia INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

NivelHemoglobina VARCHAR(150),

IDDoador INTEGER NOT NULL,

PRIMARY KEY(IDTesteAnemia),

FOREIGN KEY(IDDoador) REFERENCES Doador (IDDoador)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Entrevista (

IDEntrevista INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

Apto TINYINT,

IDTesteAnemia INTEGER NOT NULL,

PRIMARY KEY(IDEntrevista),

FOREIGN KEY(IDTesteAnemia) REFERENCES TesteAnemia(IDTesteAnemia)

ON UPDATE RESTRICT

ON DELETE RESTRICT

);

CREATE TABLE Doacao (

IDDoacao INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

QtdSangue NUMERIC(5,2),

IDEntrevista INTEGER NOT NULL,

PRIMARY KEY(IDDoacao),

FOREIGN KEY(IDEntrevista) REFERENCES Entrevista(IDEntrevista)

ON UPDATE RESTRICT

ON DELETE RESTRICT

);

CREATE TABLE TriagemSorologica (

IDTriagemSorologica INTEGER NOT NULL AUTO\_INCREMENT,

AIDS TINYINT,

Sifilis TINYINT,

HepatiteB TINYINT,

HepatiteC TINYINT,

DoencaChagas TINYINT,

HTLV TINYINT,

Data DATETIME,

IDDoacao INTEGER NOT NULL,

PRIMARY KEY(IDTriagemSorologica),

FOREIGN KEY(IDDoacao) REFERENCES Doacao (IDDoacao)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE TipoSangue (

IDTipoSangue INTEGER NOT NULL AUTO\_INCREMENT,

RH CHAR(1),

ABO VARCHAR(2),

PRIMARY KEY(IDTipoSangue)

);

CREATE TABLE Imunohematologia (

IDImunohematologia INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

TemCorpoIrregular VARCHAR(300),

IDDoacao INTEGER NOT NULL,

IDTipoSangue INTEGER NOT NULL,

PRIMARY KEY(IDImunohematologia),

FOREIGN KEY(IDTipoSangue) REFERENCES TipoSangue(IDTipoSangue)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDDoacao) REFERENCES Doacao (IDDoacao)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE SangueApto (

IDSangueApto INTEGER NOT NULL AUTO\_INCREMENT,

DataVencimento DATETIME,

IDDoacao INTEGER NOT NULL,

IDTipoSangue INTEGER NOT NULL,

PRIMARY KEY(IDSangueApto),

FOREIGN KEY(IDTipoSangue) REFERENCES TipoSangue(IDTipoSangue)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDDoacao) REFERENCES Doacao (IDDoacao)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Medico (

IDMedico INTEGER NOT NULL AUTO\_INCREMENT,

Registro VARCHAR(10),

Nome VARCHAR(150),

PRIMARY KEY(IDMedico)

);

CREATE TABLE Consulta (

IDConsulta INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

Descricao VARCHAR(300),

IDMedico INTEGER NOT NULL,

IDDoador INTEGER NOT NULL,

IDDoacao INTEGER NOT NULL,

PRIMARY KEY(IDConsulta),

FOREIGN KEY(IDMedico) REFERENCES Medico(IDMedico)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDDoador) REFERENCES Doador(IDDoador)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDDoacao) REFERENCES Doacao (IDDoacao)

ON UPDATE RESTRICT

ON DELETE RESTRICT

);

CREATE TABLE ColetaAmostra (

IDColetaAmostra INTEGER NOT NULL AUTO\_INCREMENT,

Data DATETIME,

IDTriagemSorologica INTEGER,

IDImunohematologia INTEGER,

IDConsulta INTEGER NOT NULL,

PRIMARY KEY(IDColetaAmostra),

FOREIGN KEY(IDImunohematologia) REFERENCES Imunohematologia(IDImunohematologia)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDTriagemSorologica) REFERENCES TriagemSorologica (IDTriagemSorologica)

ON UPDATE RESTRICT

ON DELETE RESTRICT,

FOREIGN KEY(IDConsulta) REFERENCES Consulta (IDConsulta)

ON UPDATE CASCADE

ON DELETE CASCADE

);

## 

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

## 

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

CREATE INDEX Index\_DoadorNome ON Doador(Nome);

CREATE INDEX Index\_DoadorRG ON Doador(RG);

CREATE INDEX Index\_SangueAptoDataVencimento ON SangueApto(DataVencimento);

CREATE INDEX Index\_EntrevistaApto ON Entrevista(Apto);

CREATE INDEX Index\_TesteAnemiaData ON TesteAnemia(Data);

CREATE INDEX Index\_EnderecoBairro ON Endereco(Bairro);

CREATE INDEX Index\_EnderecoRua ON Endereco(Rua);

CREATE INDEX Index\_TriagemSorologica ON TriagemSorologica(Data);

CREATE INDEX Index\_MedicoRegistro ON Medico(Registro);

CREATE INDEX Index\_ColetaAmostraData ON ColetaAmostra(Data);

CREATE INDEX Index\_ImunohematologiaData ON Imunohematologia(Data);

CREATE INDEX Index\_TipoSangueABO ON TipoSangue(ABO);

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

# 

# conclusão

A criação de projetos finais em quaisquer disciplinas que necessitam de experiência, tanto teórica quanto prática é de extrema importância para o aprendizado do aluno, verificação e fixação de conteúdos aprendidos durante o semestre letivo.

O trabalho feito foi de extrema importância, não só pelo conhecimento adquirido, mas também pela possibilidade de colocar em pratica tudo o que foi ministrado durante as aulas.

A resolução do problema que foi apresentada neste documento foi feita em conjunto, levando em conta a análise de cada membro do grupo, chegando a um consentimento pela equipe, tendo então, inúmeras maneiras de se resolver o mesmo problema.