UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CIÊNCIA DA COMPUTAÇÃO

ANDREY NALIGATSKI DIAS
JEFFERSON MICHAEL DE AZEVEDO JUNIOR
NICHOLAS DAMASCENO PINTO

PROJETO DE BANCO DE DADOS: MODELAGEM E DESENVOLVIMENTO

PONTA GROSSA
2019

ANDREY NALIGATSKI DIAS
JEFFERSON MICHAEL
NICHOLAS DAMASCENO

PROJETO DE BANCO DE DADOS: MODELAGEM E DESENVOLVIMENTO

Trabalho de modelagem e desenvolvimento, apresentado à disciplina de Banco de Dados, do curso de Ciência da computação da Universidade Tecnológica Federal do Paraná – UTFPR.

PONTA GROSSA
2019

SUMÁRIO

1. INTRODUÇÃO

Este trabalho tem como objetivo simular um banco de dados de uma loja contendo dentro deste informações básicas sobre a mesma e uma FAQ(Frequently Asked Questions, “Perguntas frequentes”) para os consumidores contendo respostas já predefinido para fácil acesso e rápido autoentendimento. Para questões ou problemas mais complexos, existe uma área de atendimento ao cliente, onde o mesmo especificará de seu problema com detalhes e um funcionário responsável atenderá e responder o cliente de forma clara.

2. DESENVOLVIMENTO

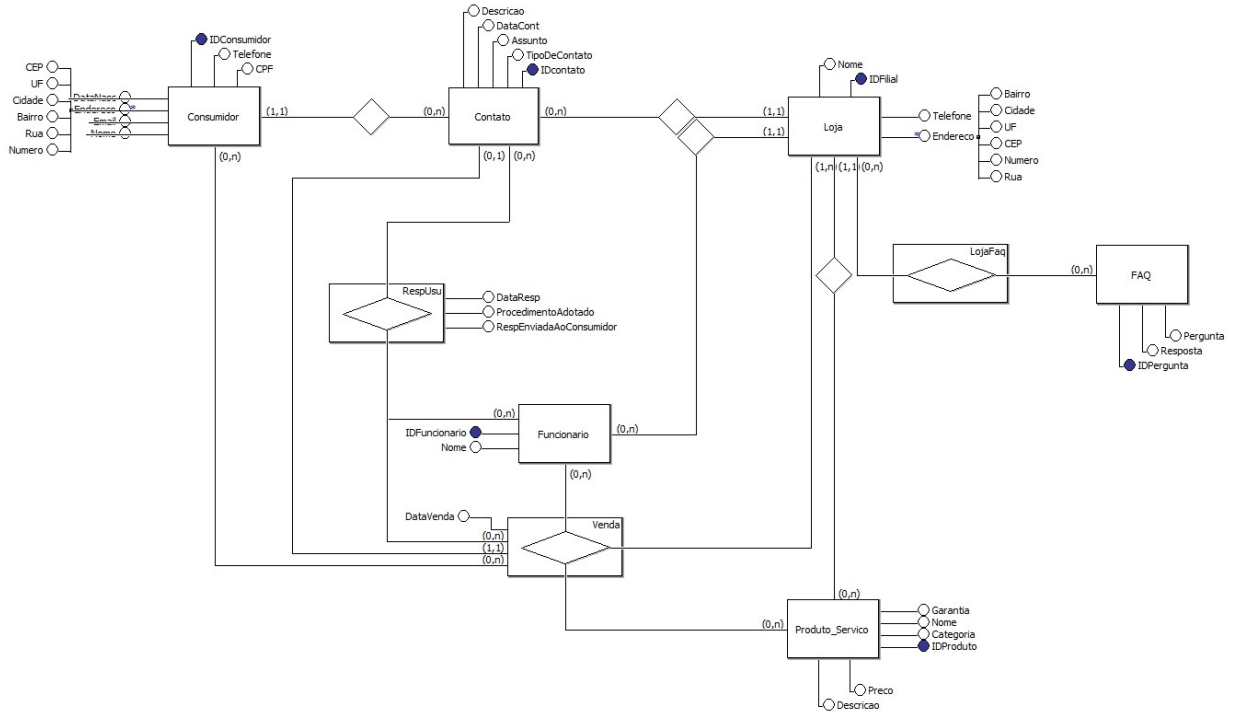
Com base na proposta apresentada do Banco de Dados do SAC(Sistema de Atendimento ao Cliente) desenvolvemos o Modelo de Entidade Relacional(MER), o Modelo Relacional Lógico(MEL) e o Script do MySQL, usando como embasamento as aulas até então aplicadas.

2.1. ELABORAÇÃO DO MODELO DE ENTIDADE E RELACIONAMENTO

Foi realizado um levantamento de entidades e atributos retirados do texto base, uma vez com estes já definidos, elaboramos as relações entre entidades e suas cardinalidades. Foi também definido a posição de cada atributo e a sua relação com cada entidade, totalizando 9, dentre estas 3 são associativas.

2.2. CONFECÇÃO DO MODELO DE ENTIDADE E RELACIONAMENTO

Com o auxílio do software indicado pela professora(Br Modelo) foi possível confeccionar um modelo relacional, conforme demonstrado na figura abaixo:

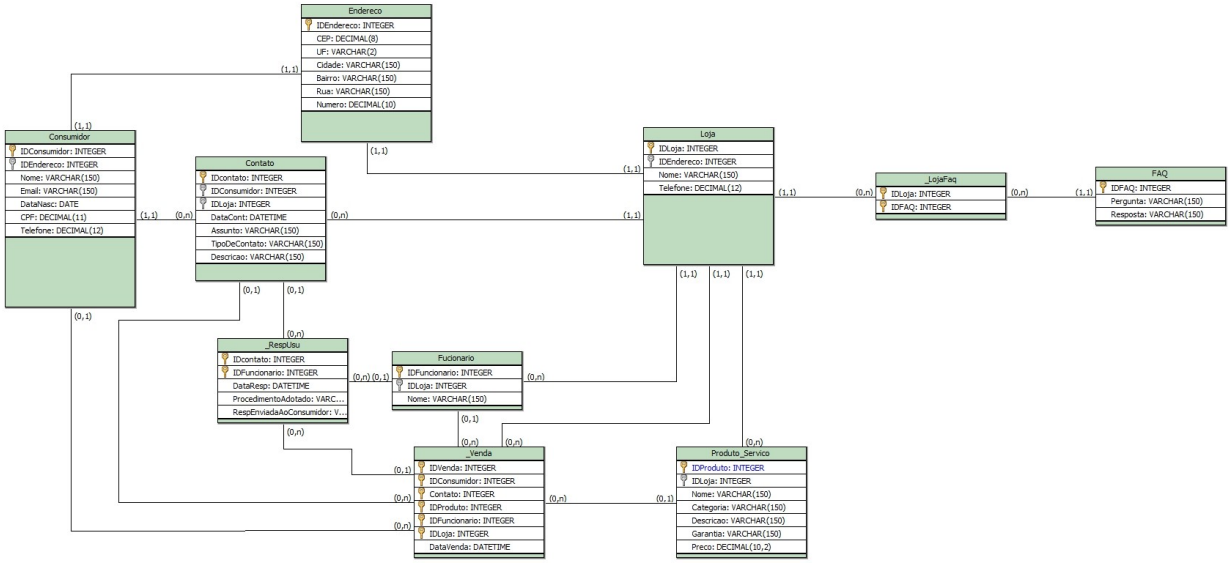


2.3. ELABORAÇÃO DO MODELO LÓGICO

Com o MER já em mãos, o MEL foi elaborado com base no modelo descrito anteriormente, adaptando apenas um atributo composto(Endereco) em uma tabela relacional, pois a mesma estava presente em duas entidades.

2.4. CONFECÇÃO DO MODELO LÓGICO

Novamente com o auxílio do Br Modelo, foi possível gerar o MEL, como podemos ver na imagem anexada abaixo:



2.5. GERANDO O BANCO

```sql
drop database if exists faq;

create database faq
default character set utf8
default collate utf8_general_ci;

use faq;

create table endereco(
    IDendereco int not null auto_increment,
    cep numeric(9) not null,
    uf varchar(2) not null,
    cidade varchar(150) not null,
    bairro varchar(150) not null,
    rua varchar(150) not null,
    numero numeric(10) not null,
    primary key (IDendereco)
) default charset = utf8;

create table consumidor(
    IDconsumidor int not null auto_increment,
    nome varchar(150) not null,
    telefone decimal(12),
    cpf numeric(11) not null,
    dataNasc date,
    email varchar(150),

    IDendereco int,
    foreign key (IDendereco) references endereco (IDendereco) on update restrict on delete restrict,

    primary key (IDconsumidor)
)default charset = utf8;

```

```sql
create table loja(
    IDloja int not null auto_increment,
    nome varchar(150),
    telefone decimal(12),

    IDendereco int,
    foreign key (IDendereco) references endereco (IDendereco) on update restrict on delete restrict,

    primary key (IDloja)
) default charset = utf8;

create table produto_servico(
    IDproduto_servico int not null auto_increment,
    nome varchar(150),
    garantia varchar(150),
    preco decimal(10,2),
    descricao varchar(150),
    categoria varchar(150),

    IDloja int,
    foreign key (IDloja) references loja (IDloja) on update restrict on delete restrict,

    primary key (IDproduto_servico)
) default charset = utf8;


create table faq (
    IDfaq int not null auto_increment,
    pergunta varchar(150),
    resposta varchar(150),
    primary key (IDfaq)
) default charset = utf8;
```

```sql
create table funcionario(
    IDfuncionario int not null auto_increment,
    nome varchar(150),

    IDloja int,
    foreign key (IDloja) references loja (IDloja) on update restrict on delete restrict,
    primary key (IDfuncionario)
) default charset = utf8;

create table lojafaq(
    IDfaq int,
    IDloja int,
    primary key (IDfaq,IDloja),
    foreign key (IDfaq) references faq (IDfaq) on update restrict on delete restrict,
    foreign key (IDloja) references loja (IDloja) on update restrict on delete restrict
) default charset = utf8;

create table venda(
    idvenda int not null auto_increment,
    dataVenda datetime not null,
    IDfuncionario int,
    IDproduto_servico int,
    IDloja int,
    IDconsumidor int,
    primary key (idvenda),
    foreign key (IDfuncionario) references funcionario (IDfuncionario) on update restrict on delete restrict,
    foreign key (IDproduto_servico) references produto_servico (IDproduto_servico) on update restrict on delete restrict,
    foreign key (IDloja) references loja (IDloja) on update restrict on delete restrict,
    foreign key (IDconsumidor) references consumidor (IDconsumidor) on update restrict on delete restrict

) default charset = utf8;
```

```
create table contato(
    IDcontato int not null auto_increment,
    dataCont datetime,
    assunto varchar(150) not null,
    tipodecontato enum('Telefone','Internet','Loja') default 'Internet',
    descricao varchar (150) not null,

    IDconsumidor int,
    foreign key (IDconsumidor) references consumidor (IDconsumidor) on update restrict on delete restrict,
    IDloja int,
    foreign key (IDloja) references loja (IDloja) on update restrict on delete restrict,
    IDvenda int,
    foreign key (IDvenda) references venda (IDvenda) on update restrict on delete restrict,

    primary key (IDcontato)
)default charset = utf8;

create table respusu(
    procedimentoAdotado varchar(150),
    dataResp datetime not null,
    respEnviadaAoConsumidor varchar(150) not null,
    IDcontato int,
    IDvenda int,
    IDfuncionario int,

    foreign key (IDfuncionario) references funcionario (IDfuncionario) on update restrict on delete restrict,
    foreign key (IDvenda) references venda (IDvenda) on update restrict on delete restrict,
    foreign key (IDcontato) references contato (IDcontato) on update restrict on delete restrict,
    primary key (IDcontato, IDvenda, IDfuncionario)
) default charset = utf8;
```

2.6. VIEW

```
#primeira view abaixo
create view reclamacoes (Produto, qtdReclamacoes) as
select produto_servico.nome, count(*) as qtdReclamacoes
from produto_servico, contato, venda
where contato.IDvenda = venda.IDvenda and
produto_servico.IDproduto_servico = venda.IDproduto_servico
group by produto_servico.IDproduto_servico
order by qtdReclamacoes desc;

#segunda view
create view semreclamacao (Nome, Descricao, Categoria) as
select p.nome, p.descricao, p.categoria
from venda v, produto_servico p
where v.IDproduto_servico = p.IDproduto_servico and
 v.IDproduto_servico not in(
select v.IDproduto_servico
    from venda v , contato c, produto_servico p
    where v.IDvenda = c.IDvenda and
    p.IDproduto_servico = v.IDproduto_servico and
    year(v.dataVenda) = '2018')
group by p.IDproduto_servico
order by p.nome,p.categoria
;
```

2.7. STORED PROCEDURE

```
#### endereco
DELIMITER //
create procedure sp_insendereco (in pcep numeric(9),puf varchar(2),pcidade varchar(150),
pbairro varchar(150),prua varchar(150),pnumero numeric(10), out mensagem varchar(150))
begin
    if( (pcep is not null) and
    (puf is not null) and
    (pcidade is not null) and
    (pbairro is not null) and
    (prua is not null) and
    (pnumero is not null) and
    (not exists(select * from endereco where cidade like pcidade and
    bairro like pbairro and rua like prua and numero = pnumero and cep = pcep))) then
        begin
            insert into endereco
            values (default, pcep, puf, pcidade, pbairro, prua, pnumero);
            set mensagem = 'Endereco inserido com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altendereco (in pid int,pcep numeric(9),puf varchar(2),
pcidade varchar(150),pbairro varchar(150),prua varchar(150),pnumero numeric(10), out mensagem varchar(150))
begin
    if( (pcep is not null) and
    (puf is not null) and
    (pcidade is not null) and
    (pbairro is not null) and
    (prua is not null) and
    (pnumero is not null) and
    (exists(select * from endereco where IDendereco = pid))) then
        begin
            update endereco
            set cep = pcep, uf = puf, cidade = pcidade, bairro = pbairro, rua = prua, numero = pnumero
            where IDendereco = pid;
            set mensagem = 'Endereco alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
#### consumidor
DELIMITER //
create procedure sp_insconsumidor (in pnome varchar(150),ptelefone numeric(12),pcpf numeric(11),
pdataNasc date,pemail varchar(150),pidendereco int, out mensagem varchar(150))
begin
    if( (pnome is not null) and
    (pcpf is not null) and
    ((pemail is not null) or (ptelefone)) and
    (not exists(select * from consumidor where cpf = pcpf)) and
    (exists(select * from endereco where IDendereco = pidendereco))) then
        begin
            insert into consumidor
            values (default, pnome, ptelefone, pcpf, pdataNasc, pemail, pidendereco);
            set mensagem = 'Consumidor registrado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altconsumidor (in pIDconsumidor int, pnome varchar(150),ptelefone numeric(12),
pcpf numeric(11),pdataNasc date,pemail varchar(150),pidendereco int, out mensagem varchar(150))
begin
    if( (pnome is not null) and
    (pcpf is not null) and
    ((pemail is not null) or (ptelefone)) and
    (exists(select * from endereco where IDendereco = pidendereco)) and
    (exists(select * from consumidor where IDconsumidor = pIDconsumidor))) then
        begin
            update consumidor
            set nome = pnome, telefone = ptelefone, cpf = pcpf, dataNasc = pdataNasc, email = pemail,
            IDendereco = pidendereco
            where IDconsumidor = pIDconsumidor;
            set mensagem = 'Consumidor registrado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
#### funcionario
DELIMITER //
create procedure sp_insfuncionario (in pnome varchar(150),pIDloja int, out mensagem varchar(150))
DELIMITER //
create procedure sp_altfuncionario (in pIDFuncionario int, pnome varchar(150), pIDloja int, out mensagem varchar(150))
begin
    if( (pnome is not null) and
    (exists(select * from funcionario where IDfuncionario = pIDfuncionario))and
    (exists(select * from loja where IDloja = pIDloja))) then
        begin
            update funcionario
            set nome = pnome, IDloja = pIDloja
            where IDfuncionario = pIDfuncionario;
            set mensagem = 'Funcionario registrado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;



DELIMITER //
create procedure sp_altfuncionario (in pIDFuncionario int, pnome varchar(150), pIDloja int, out mensagem varchar(150))
begin
    if( (pnome is not null) and
    (exists(select * from funcionario where IDfuncionario = pIDfuncionario))and
    (exists(select * from loja where IDloja = pIDloja))) then
        begin
            update funcionario
            set nome = pnome, IDloja = pIDloja
            where IDfuncionario = pIDfuncionario;
            set mensagem = 'Funcionario registrado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
#### faq
DELIMITER //
create procedure sp_insfaq(in ppergunta varchar(150), presposta varchar(150), out mensagem varchar(150))
begin
    if( (ppergunta is not null) and
    (presposta is not null) and
    (not exists(select * from faq where pergunta like ppergunta and resposta like presposta))) then
        begin
            insert into faq
            values (default, ppergunta, presposta);
            set mensagem = 'FAQ inserido com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altfaq (in pid int, ppergunta varchar(150), presposta varchar(150), out mensagem varchar(150))
begin
    if( (ppergunta is not null) and
    (presposta is not null) and
    (exists(select * from faq where IDfaq = pid))) then
        begin
            update faq
            set pergunta = ppergunta, resposta = presposta
            where IDfaq = pid;
            set mensagem = 'FAQ alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
#### contato
DELIMITER //
create procedure sp_inscontato (in pdatacont datetime, passunto varchar(150), ptipocontato varchar(150),
 pdescricao varchar(150), pIDconsumidor int, pIDloja int, pIDvenda int, out mensagem varchar(150))
begin
    if ((pdatacont is not null) and
        (passunto is not null) and
        (ptipocontato is not null) and
        (pdescricao is not null) and
        (exists(select * from consumidor where IDconsumidor = pIDconsumidor)) and
        (exists(select * from venda where IDvenda = pIDvenda)) and
        (exists(select * from loja where IDloja = pIDloja)))
        then
        begin
            insert into contato
            values (default, pdatacont, passunto, ptipocontato, pdescricao,pIDconsumidor, pIDloja, pIDvenda);
            set mensagem = 'Contato registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;


DELIMITER //
create procedure sp_altcontato (in pid int, pdatacont datetime, passunto varchar(150),
 ptipocontato varchar(150), pdescricao varchar(150), pIDconsumidor int, pIDloja int,
 pIDvenda int, out mensagem varchar(150))
begin
    if ((pdatacont is not null) and
        (passunto is not null) and
        (ptipocontato is not null) and
        (pdescricao is not null) and
        (exists(select * from consumidor where IDconsumidor = pIDconsumidor)) and
        (exists(select * from contato where IDcontato = pid)) and
        (exists(select * from venda where IDvenda = pIDvenda)) and
        (exists(select * from loja where IDloja = pIDloja)))
        then
        begin
            update contato
            set dataCont = pdatacont, assunto=passunto, tipodecontato=ptipocontato,
            descricao=pdescricao, IDconsumidor=pIDconsumidor, IDloja=pIDloja, IDvenda=pIDvenda
            where IDcontato = pid;
            set mensagem = 'Contato alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

```
#### produto
DELIMITER //
create procedure sp_insproduto (in pnome varchar(150), pgarantia varchar(150), ppreco decimal(10,2),
 pdescricao varchar(150), pcategoria varchar(150), pIDloja int, out mensagem varchar(150))
begin
    if ((pnome is not null) and
        (pgarantia is not null) and
        (ppreco is not null) and
        (pdescricao is not null) and
        (pcategoria is not null) and
        (exists(select * from loja where IDloja = pIDloja)))
        then
        begin
            insert into produto_servico
            values (default, pnome, pgarantia, ppreco, pdescricao,pcategoria, pIDloja);
            set mensagem = 'Produto_Servico registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altproduto (in pid int,  pnome varchar(150), pgarantia varchar(150),
 ppreco decimal(10,2), pdescricao varchar(150), pcategoria varchar(150), pIDloja int,
 out mensagem varchar(150))
begin
    if ((pnome is not null) and
        (pgarantia is not null) and
        (ppreco is not null) and
        (pdescricao is not null) and
        (pcategoria is not null) and
        (exists(select * from loja where IDloja = pIDloja))and
        (exists(select * from produto_servico where IDproduto_servico = pid)))
        then
        begin
            update produto_servico
            set nome = pnome, garantia=pgarantia, preco=ppreco, descricao=pdescricao,
            categoria=pcategoria, IDloja=pIDloja
            where IDproduto_servico = pid;
            set mensagem = 'Produto_servico alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

```
#### loja
DELIMITER //
create procedure sp_insloja (in pnome varchar(150), ptelefone decimal(12),  pIDendereco int, out mensagem varchar(150))
begin
    if ((pnome is not null) and
        (ptelefone is not null) and
        (exists(select * from endereco where IDendereco = pIDendereco)))
        then
        begin
            insert into loja
            values (default, pnome, ptelefone , pIDendereco);
            set mensagem = 'loja registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altloja(in pid int,  pnome varchar(150), ptelefone decimal(12),
 pIDendereco int, out mensagem varchar(150))
begin
    if ((pnome is not null) and
        (ptelefone is not null) and
        (exists(select * from endereco where IDendereco = pIDendereco))and
        (exists(select * from loja where IDloja = pid)))
        then
        begin
            update loja
            set nome = pnome, telefone=ptelefone, IDendereco=pIDendereco
            where IDloja= pid;
            set mensagem = 'Endereco alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

```
#### lojafaq
DELIMITER //
create procedure sp_inslojafaq(in pIDloja int, pIDfaq int, out mensagem varchar(150))
begin
    if ((exists(select * from faq where IDfaq = pIDfaq)) and
        (exists(select * from loja where IDloja = pIDloja)) and
        (not exists(select * from lojafaq where IDloja = pIDloja and IDfaq = pIDfaq)))
        then
        begin
            insert into lojafaq
            values (pIDfaq,pIDloja);
            set mensagem = 'Lojafaq registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;


DELIMITER //
create procedure sp_altlojafaq (in pIDloja int, pIDfaq int,npIDloja int, npIDfaq int, out mensagem varchar(150))
begin
    if ((exists(select * from faq where IDfaq = npIDfaq)) and
        (exists(select * from loja where IDloja = npIDloja)) and
        (not exists(select * from lojafaq where IDloja = pIDloja and IDfaq = pIDfaq)))
        then
        begin
            update lojafaq
            set IDfaq=npIDfaq , IDloja=npIDloja
            where IDloja = pIDloja and IDfaq = pIDfaq;
            set mensagem = 'Lojafaq alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

```
#### vendas
DELIMITER //
create procedure sp_insvenda(in pdatavenda datetime, pIDfuncionario int, pIDproduto_servico int,
 pIDloja int, pIDconsumidor int, out mensagem varchar(150))
begin
    if ((pdatavenda is not null) and
        (pIDfuncionario is not null) and
        (pIDproduto_servico is not null) and
        (pIDloja is not null) and
        (pIDconsumidor is not null) and
        (exists(select * from funcionario where IDfuncionario = pIDfuncionario)) and
        (exists(select * from produto_servico where IDproduto_servico = pIDproduto_servico)) and
        (exists(select * from loja where IDloja = pIDloja)) and
        (exists(select * from consumidor where IDconsumidor)))
        then
        begin
            insert into venda
            values (default, pdatavenda, pIDfuncionario, pIDproduto_servico, pIDloja, pIDconsumidor);
            set mensagem = 'Venda registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altvenda(in pIDvenda int, pdatavenda datetime, pIDfuncionario int, pIDproduto_servico int,
 pIDloja int, pIDconsumidor int, out mensagem varchar(150))
begin
    if ((pIDvenda is not null) and
        (pdatavenda is not null) and
        (pIDfuncionario is not null) and
        (pIDproduto_servico is not null) and
        (pIDloja is not null) and
        (pIDconsumidor is not null) and
        (exists(select * from venda where IDvenda = pIDvenda)) and
        (exists(select * from funcionario where IDfuncionario = pIDfuncionario)) and
        (exists(select * from produto_servico where IDproduto_servico = pIDproduto_servico)) and
        (exists(select * from loja where IDloja = pIDloja)) and
        (exists(select * from consumidor where IDconsumidor)))
        then
        begin
            update venda
            set dataVenda = pdatavenda, IDfuncionario = pIDfuncionario, IDproduto_servico = pIDproduto_servico,
            IDloja = pIDloja, IDconsumidor = pIDconsumidor
            where IDvenda= pIDvenda;
            set mensagem = 'Venda alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_insrespusu(in pprocedimentoAdotado varchar(150), pdataResp datetime,
 prespEnviadaAoConsumidor varchar(150), pIDcontato int, pIDvenda int, pIDfuncionario int, out mensagem varchar(150))
begin
    if ((pdataresp is not null) and
        (prespEnviadaAoConsumidor is not null) and
        (pIDcontato is not null) and
        (pIDvenda is not null) and
        (pIDfuncionario is not null) and
        (exists(select * from contato where IDcontato = pIDcontato)) and
        (exists(select * from venda where IDvenda = pIDvenda)) and
        (exists(select * from funcionario where IDfuncionario = pIDfuncionario)))
        then
        begin
            insert into respusu
            values (pprocedimentoAdotado, pdataresp, prespEnviadaAoConsumidor, pIDcontato, pIDvenda, pIDfuncionario);
            set mensagem = 'Respusu registrado com sucesso';
        end;

    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end
//
DELIMITER ;
```

```
DELIMITER //
create procedure sp_altrespusu(in pprocedimentoAdotado varchar(150), pdataResp datetime,
 prespEnviadaAoConsumidor varchar(150), pIDcontato int, pIDvenda int, pIDfuncionario int,
 npIDcontato int, npIDvenda int, npIDfuncionario int, out mensagem varchar(150))
begin
    if ((pdataresp is not null) and
        (prespEnviadaAoConsumidor is not null) and
        (npIDcontato is not null) and
        (npIDvenda is not null) and
        (npIDfuncionario is not null) and
        (exists(select * from contato where IDcontato = npIDcontato)) and
        (exists(select * from venda where IDvenda = npIDvenda)) and
        (exists(select * from funcionario where IDfuncionario = npIDfuncionario)) and
        (exists(select * from respusu where IDcontato = pIDcontato and IDvenda = pIDvenda and
        IDfuncionario = pIDfuncionario)))
        then
        begin
            update respusu
            set IDcontato=npIDcontato , IDvenda=npIDvenda, IDfuncionario=npIDfuncionario,
            dataResp = pdataResp, respEnviadaAoConsumidor = prespEnviadaAoConsumidor,
            procedimentoAdotado = pprocedimentoAdotado
            where IDcontato = pIDcontato and IDvenda = pIDvenda and IDfuncionario = pIDfuncionario;
            set mensagem = 'Respusu alterado com sucesso';
        end;
    else
        set mensagem = 'Operacao nao realizada, erro';
    end if;
end

//
DELIMITER ;
```

2.8. ÍNDICES

```
create index idx_cpf
on consumidor(cpf);
create index idx_assuntocontato
on contato(assunto);
create index idx_nomeloja
on loja(nome);
create index idx_pergunta
on faq(pergunta);
create index idx_nomeProdServ
on produto_servico(nome);
create index idx_vendaCodBarras
on venda(idvenda);
create index idx_dataresposta
on respusu(respEnviadaAoConsumidor);
```

3. CONCLUSÃO

Ao finalizarmos este projeto, concluímos que o trabalho em suma não apresentava tanta dificuldade, porém exigia uma alta dedicação e tempo, causando assim uma série de problemas entre os integrantes do grupo, contudo, o grupo conseguiu se organizar e cumprir tal tarefa no prazo esperado. O projeto nos trouxe experiência tanto em desenvolvimento de trabalhos acadêmicos quanto em desenvolvimento de bancos de dados. Acreditamos que os trabalhos práticos são sempre bem-vindos. Utilizando e aplicando o que foi aprendido em sala de forma eficiente e sem a pressão da prova.