



UNIVERSIDADE ESTADUAL DE CAMPINAS
[ST767B] Banco de Dados II
Gisele Busichia Baioco

PROJETO - PARTE 2

Jonatas Goes da Silva - 237948
Matheus Eduardo da Silva - 230719

**LIMEIRA -SP
2021**

SUMÁRIO

1. INTRODUÇÃO	3
2. DESCRIÇÃO	4
3. DESCRIÇÃO DAS PRINCIPAIS FUNCIONALIDADES DO SISTEMA E DOS DADOS A SEREM ARMAZENADOS	5
4. MER	8
5. PROJETO LÓGICO DO BANCO DE DADOS	9
6. TABELAS	11
7. PROCEDIMENTOS	15
8. TRIGGERS	33
9. SCRIPTS	34
10. VIEWS	37

1. INTRODUÇÃO

Com base em pesquisas desenvolvidas pelo grupo, conseguimos informações importantes sobre o funcionamento de uma empresa de aluguel de carros, visando com isso a construção de um modelo conceitual de um banco de dados com sistemas de controle de vendas, locação e estoque, para que, seja utilizado para criação de um banco de dados.

2. DESCRIÇÃO

O banco de dados de aluguel de um produto de valores extremamente altos deve conter diversas informações, principalmente relacionadas ao cliente que irá alugar o veículo, cujo cadastro deverá ser aprovado, onde após todas as informações do mesmo serem verificadas para enfim, poder alugar um veículo. Além disso, as informações relacionadas aos carros, auto-mecânicas para reparos e registro de aluguéis são necessárias para uma gestão completa do processo.

3. DESCRIÇÃO DAS PRINCIPAIS FUNCIONALIDADES DO SISTEMA E DOS DADOS A SEREM ARMAZENADOS

3.1 Principais funcionalidades:

- Verificar disponibilidade do veículo
- Organizar os carros pelas informações(valor, ano, quilometragem, etc)
- Verificar os dados do cliente(cpf, cartão de crédito, etc)
- Verificar os dados do atendente(carteira de trabalho, salário)
- Impedir a locação do veículo caso o seguro não esteja em conformidade
- Multa(taxa) por cada dia de atraso de entrega(prazo) do veículo
- Multar(taxa) se o tanque não estiver com no mínimo 50% da capacidade
- Registrar carros na auto-mecânica correspondente
- Registrar o aluguel do carro
- Registrar os reparos efetuados nos carros

3.2 Dados armazenados:

Para melhor visualização, itens sublinhados são definidos como chaves primárias, e itens em itálico são chaves únicas.

3.2.1 Dados do carro:

- Placa
- Modelo
- Marca
- Ano
- Quilometragem
- Seguro
- Tanque de combustível
- Manutenção
- Valor
- Disponibilidade

3.2.2 Dados da Pessoa:

- ID
- Nome
- CPF
- Email
- Telefone
- Endereço

3.2.3 Dados do cliente

- ID da pessoa (chave estrangeira para pessoa)

- Idade(acima de 21)
- *Carteira de habilitação*
- Cartão de crédito
- *RG*

3.2.4 Dados do atendente

- *ID da pessoa* (chave estrangeira para pessoa)
- *Carteira de trabalho*
- Salário

3.2.5 Dados da Pessoa Jurídica:

- *CNPJ*
- Nome
- Telefone
- Email
- Endereço

3.2.6 Dados da Auto-mecânica:

- *CNPJ da p. jurídica* (chave estrangeira para pessoa jurídica)
- Especialidade

3.2.7 Dados da seguradora:

- *CNPJ da p. jurídica* (chave estrangeira para pessoa jurídica)
- Cobertura

3.2.8 Dados da concessionária:

- *CNPJ da p. jurídica* (chave estrangeira para pessoa jurídica)
- Marca

3.2.9 Dados do estacionamento:

- *CNPJ da p. jurídica* (chave estrangeira para pessoa jurídica)
- Avaliação

3.2.10 Dados do Aluguel:

- *Código*
- *ID do atendente* (chave estrangeira para atendente)
- *Placa do carro* (chave estrangeira para Carro)
- *ID do cliente* (chave estrangeira para Cliente)
- Data início
- Data entrega
- Valor aluguel
- Prazo
- Taxas
- Valor final

- Status do aluguel

3.2.11 Dados de Reparo:

- Código
- Placa do carro(chave estrangeira para carro)
- CNPJ da seguradora(chave estrangeira para seguradora)
- CNPJ da auto-mecânica(chave estrangeira para auto-mecânica)
- Data início
- Data prevista de entrega
- Valor
- Status
- Informações

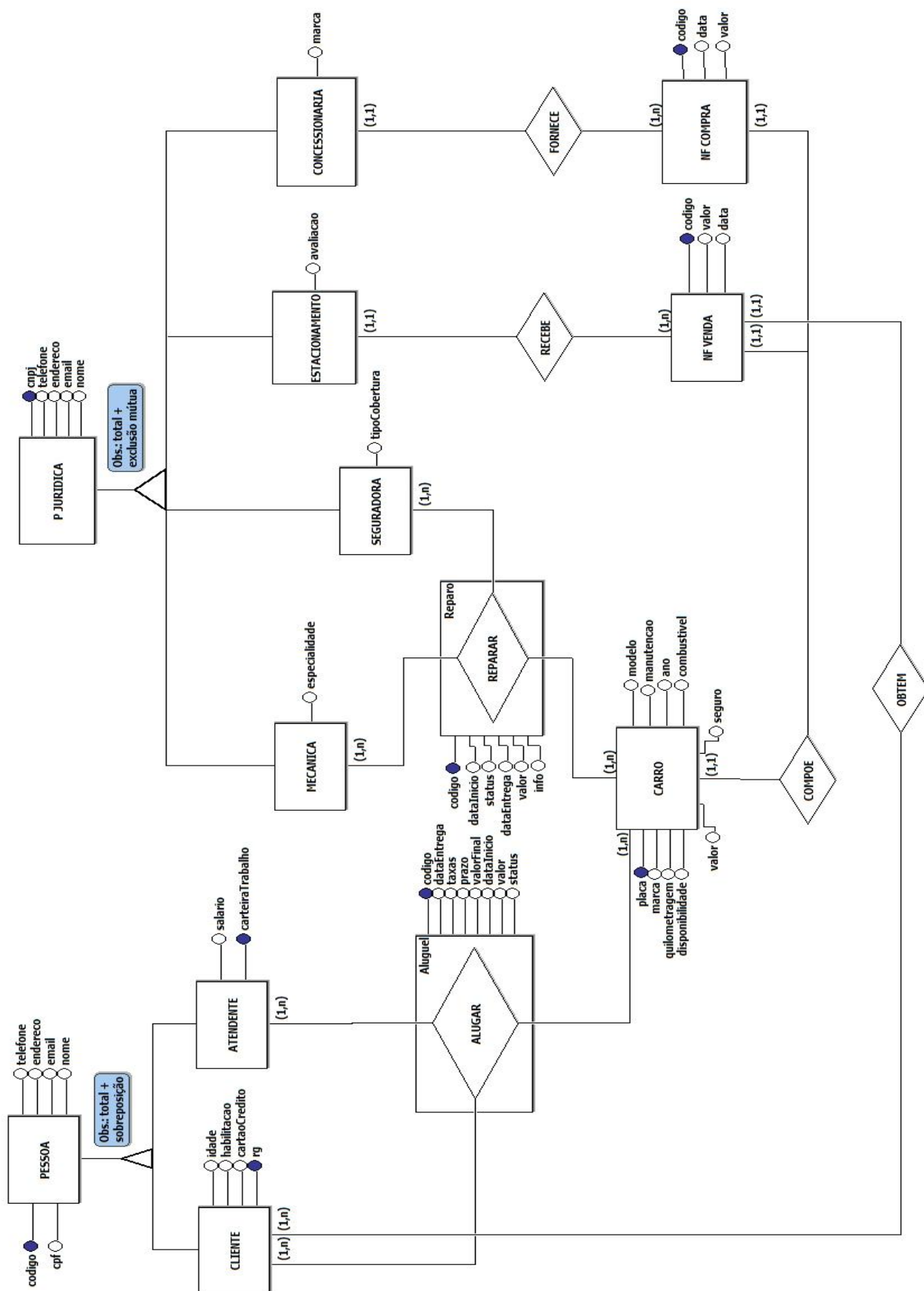
3.2.11 Dados da Compra veicular(“Nota fiscal de compra”):

- Código
- Placa do carro(chave estrangeira para carro)
- CNPJ da concessionária(chave estrangeira para concessionária)
- Data
- Valor

3.2.12 Dados da Venda veicular(“Nota fiscal de venda”):

- Código
- Placa do carro(chave estrangeira para carro)
- CNPJ do estacionamento(chave estrangeira para estacionamento)
- ID do cliente (chave estrangeira para cliente)
- Data
- Valor

4. MER



5. PROJETO LÓGICO DO BANCO DE DADOS

Pessoa = { ID, Nome, CPF, Email, Telefone, Endereço }

cliente = { ID, Idade, Carteira de habilitação, Cartão de crédito ,RG}
ID - chave estrangeira referenciando pessoa

atendente = { ID, Idade ,Carteira de trabalho ,Salário}
ID - chave estrangeira referenciando pessoa

Pessoa Jurídica = { CNPJ, Nome, Telefone, Email, Endereço }

Auto-mecânica = { CNPJ , Especialidade)
CNPJ - chave estrangeira referenciando pessoa jurídica

seguradora = { CNPJ , Cobertura}
CNPJ - chave estrangeira referenciando pessoa jurídica

concessionária = { CNPJ , Marca}
CNPJ - chave estrangeira referenciando pessoa jurídica

estacionamento = { CNPJ , Avaliação}
CNPJ - chave estrangeira referenciando pessoa jurídica

Aluguel = {Código, ID do atendente, Placa do carro, ID do cliente ,Data início, Data entrega, Valor aluguel, Prazo,Taxas, Valor final, Status do aluguel }

ID do atendente - chave estrangeira referenciando atendente

Placa do carro - chave estrangeira referenciando Carro

ID do cliente - chave estrangeira referenciando Cliente

Reparo = {Código, Placa do carro, CNPJ da seguradora, CNPJ da auto-mecânica ,Data prevista de entrega, Valor ,Status ,Informações}

Placa do carro - chave estrangeira referenciando carro

CNPJ da seguradora - chave estrangeira referenciando seguradora

CNPJ da auto-mecânica - chave estrangeira referenciando auto-mecânica

carro = { Placa, Modelo, Marca, Ano, Quilometragem, Seguro, Tanque de combustível, Manutenção, Valor, Disponibilidade}

Reparo = {Código, Placa, CNPJ da seguradora, CNPJ da auto-mecânica, Data início, Data prevista de entrega, Valor, Status, Informações}

Placa do carro - chave estrangeira referenciando carro

CNPJ da seguradora - chave estrangeira referenciando seguradora

CNPJ da auto-mecânica - chave estrangeira referenciando auto-mecânica

Compra veicular = {Código, Placa do carro, CNPJ da concessionária, Data, Valor }

Placa - chave estrangeira referenciando carro

CNPJ da concessionária - chave estrangeira referenciando concessionária

Venda veicular = { Código, Placa do carro, CNPJ do estacionamento, ID do cliente, Data, Valor }

Placa do carro - chave estrangeira para carro

CNPJ do estacionamento - chave estrangeira referenciando estacionamento

ID do cliente - chave estrangeira referenciando cliente

6. TABELAS

```
USE trabPt2_BD2
```

```
/*TABELAS:*/
```

```
create table pessoa
```

```
(  
id int not null,  
cpf char(11) not null,  
telefone char(10) null,  
endereco char(50) not null,  
email char(25) not null,  
nome char(35) not null,  
primary key(id),  
unique(id),  
unique(cpf)  
)
```

```
create table cliente
```

```
(  
id int not null,  
rg char(10) not null,  
idade smallint not null,  
cnh char(11) not null,  
cartao char(25) not null,  
primary key(id),  
unique(rg),  
unique(cnh),  
foreign key (id) references pessoa(id)  
)
```

```
create table atendente
```

```
(  
id int not null,  
cartTrab char(20) not null,  
salario smallmoney not null,  
primary key(id),  
unique(cartTrab),  
foreign key (id) references pessoa(id)  
)
```

```
create table carro
```

```
(  
placa char(10) not null,  
marca char(15) not null,  
modelo char(20) not null,  
ano smallint not null,
```

```
km int not null,  
combustivel int not null,  
valor smallmoney not null,  
manutencao bit not null,  
seguro bit not null,  
disponivel bit not null,  
primary key(placa),  
unique(placa),  
)
```

```
create table Pjuridica  
(  
  cnpj char(14) not null,  
  telefone char(10) null,  
  endereco char(50) not null,  
  email char(25) not null,  
  nome char(35) not null,  
  primary key(cnpj),  
  unique(cnpj),  
)
```

```
create table mecanica  
(  
  cnpj char(14) not null,  
  espec char(20) not null,  
  primary key(cnpj),  
  foreign key(cnpj) references Pjuridica(cnpj)  
)
```

```
create table seguradora  
(  
  cnpj char(14) not null,  
  cobertura char(20) not null,  
  primary key(cnpj),  
  foreign key(cnpj) references Pjuridica(cnpj)  
)
```

```
create table estacionamento  
(  
  cnpj char(14) not null,  
  avaliacao int not null,  
  primary key(cnpj),  
  foreign key(cnpj) references Pjuridica(cnpj)  
)
```

```
create table concessionaria  
(  
  cnpj char(14) not null,
```

```
marca char(20) not null,  
primary key(cnpj),  
foreign key(cnpj) references Pjuridica(cnpj)  
)
```

```
create table NFcompra  
(  
codigo int not null,  
valor smallmoney not null,  
dataCompra smalldatetime not null,  
cnpj_conc char(14) not null,  
placa_carro char(10) not null,  
primary key(codigo),  
unique(codigo),  
foreign key (cnpj_conc) references concessionaria(cnpj),  
foreign key (placa_carro) references carro(placa),  
)
```

```
create table NFvenda  
(  
codigo int not null,  
valor smallmoney not null,  
dataCompra smalldatetime not null,  
cnpj_est char(14) null,  
idCliente int null,  
placa_carro char(10) not null,  
primary key(codigo),  
unique(codigo),  
foreign key (cnpj_est) references estacionamento(cnpj),  
foreign key (idCliente) references cliente,  
)
```

```
create table aluguel  
(  
codigo int not null,  
dataInicio datetime not null,  
dataEntrega datetime null,  
prazo datetime not null,  
valor smallmoney not null,  
taxas smallmoney null,  
valorFinal smallmoney null,  
statusAluguel bit not null,  
placaCarro char(10) not null,  
idCliente int not null,  
idAtendente int not null,  
primary key(codigo),  
unique(codigo),  
foreign key (placaCarro) references carro,
```

```
foreign key (idCliente) references cliente,
foreign key (idAtendente) references atendente,
)
```

```
create table reparo
(
codigo int not null,
dataInicio smalldatetime not null,
dataEntrega smalldatetime null,
valor smallmoney not null,
info char(30) null,
statusReparo bit not null,
placaCarro char(10) not null,
cnpj_mec char(14) not null,
cnpj_seg char(14) not null,
primary key(codigo),
unique(codigo),
foreign key (placaCarro) references carro,
foreign key (cnpj_mec) references mecanica,
foreign key (cnpj_seg) references seguradora
)
```

```
/*criando index para aluguel do atendente e cliente e carros + reparo:*/
create index ix_alguel_cli on aluguel(idCliente)
create index ix_alguel_aten on aluguel(idAtendente)
create index ix_alguel_car on aluguel(placaCarro)
create index ix_reparo_car on reparo(placaCarro)
```

7. PROCEDIMENTOS

```
USE trabPt2_BD2
```

```
/*CRIANDO PROCEDIMENTOS:*/
```

```
/*Inserindo clientes:*/
```

```
create procedure cria_cliente
```

```
@id int,
```

```
@cpf char(11),
```

```
@telefone char(10),
```

```
@endereco char(50),
```

```
@email char(25),
```

```
@nome char(35),
```

```
@rg char(10),
```

```
@idade smallint,
```

```
@cnh char(11),
```

```
@cartao char(25)
```

```
as
```

```
begin transaction
```

```

insert into pessoa values(@id, @cpf, @telefone, @endereco, @email, @nome)
if @@rowcount > 0 /* insercao de pessoa bem sucedida */
begin
    insert into cliente values(@id, @rg, @idade, @cnh, @cartao)
    if @@rowcount > 0 /* insercao de cliente bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

```

```

/*Inserindo atendentes:*/
create procedure cria_atendente
@id int,
@cpf char(11),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@cartTrab char(20),
@salario smallmoney
as
begin transaction
insert into pessoa values(@id, @cpf, @telefone, @endereco, @email, @nome)
if @@rowcount > 0 /* insercao de pessoa bem sucedida */
begin
    insert into atendente values(@id, @cartTrab, @salario)
    if @@rowcount > 0 /* insercao de atendente bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end

```

```

else
begin
    rollback transaction
    return 0
end

/*Inserindo seguradoras:*/
create procedure cria_seguradora
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@cobertura char(20)
as
begin transaction
insert into Pjuridica values(@cnpj, @telefone, @endereco, @email, @nome)
if @@rowcount > 0 /* insercao de p. juridica bem sucedida */
begin
    insert into seguradora values(@cnpj, @cobertura)
    if @@rowcount > 0 /* insercao de seguradora bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else
begin
    rollback transaction
    return 0
end
end

```

```

/*Inserindo auto-mecanicas:*/
create procedure cria_mecanica
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@espec char(20)
as
begin transaction
insert into Pjuridica values(@cnpj, @telefone, @endereco, @email, @nome)

```



```

if @@rowcount > 0 /* insercao de p. juridica bem sucedida */
begin
    insert into mecanica values(@cnpj, @espec)
    if @@rowcount > 0 /* insercao de mecanica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

/*Inserindo concessionarias:*/
create procedure cria_concessionaria
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@marca char(20)
as
begin transaction
insert into Pjuridica values(@cnpj, @telefone, @endereco, @email, @nome)
if @@rowcount > 0 /* insercao de p. juridica bem sucedida */
begin
    insert into concessionaria values(@cnpj, @marca)
    if @@rowcount > 0 /* insercao de concessionaria bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction

```

```

        return 0
    end

/*Inserindo estacionamentos:*/
create procedure cria_estacionamento
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@avaliacao int
as
begin transaction
insert into Pjuridica values(@cnpj, @telefone, @endereco, @email, @nome)
if @@rowcount > 0 /* insercao de p. juridica bem sucedida */
begin
    insert into estacionamento values(@cnpj, @avaliacao)
    if @@rowcount > 0 /* insercao de concessionaria bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end
end

```

```

/*ALTERACOES:*/
/*Alterando clientes:*/
create procedure alter_cliente
@id int,
@cpf char(11),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@rg char(10),
@idade smallint,
@cnh char(11),
@cartao char(25)
as

```

```

begin transaction
update pessoa
set cpf = @cpf, telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where
id = @id
if @@rowcount > 0
begin
    update cliente
    set rg = @rg, idade = @idade, cnh = @cnh, cartao = @cartao where id = @id
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end
end

```

```

/*Alterando atendentes:*/
create procedure alter_atendente
@id int,
@cpf char(11),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@cartTrab char(20),
@salario smallmoney
as
begin transaction
update pessoa
set cpf = @cpf, telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where
id = @id
if @@rowcount > 0
begin
    update atendente
    set cartTrab = @cartTrab, salario = @salario where id = @id
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
end

```

```

        end
    else
        begin
            rollback transaction
            return 0
        end
    end
end
else
begin
    rollback transaction
    return 0
end
end

```

/*Alterando seguradora:*/

create procedure alter_seguradora

@cnpj char(14),

@telefone char(10),

@endereco char(50),

@email char(25),

@nome char(35),

@cobertura char(20)

as

begin transaction

update Pjuridica

set telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where cnpj = @cnpj

if @@rowcount > 0

begin

update seguradora

set cobertura = @cobertura where cnpj= @cnpj

if @@rowcount > 0

begin

commit transaction

return 1

end

else

begin

rollback transaction

return 0

end

end

else

begin

rollback transaction

return 0

end

/*Alterando auto-mecanica:*/

```

create procedure alter_mecanica
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@espec char(20)
as
begin transaction
update Pjuridica
set telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where cnpj =
@cnpj
if @@rowcount > 0
begin
    update mecanica
    set espec = @espec where cnpj= @cnpj
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end
end

```

/*Alterando concessionária:*/

```

create procedure alter_concessionaria
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@marca char(20)
as
begin transaction
update Pjuridica
set telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where cnpj =
@cnpj
if @@rowcount > 0
begin

```

```

update concessionaria
set marca = @marca where cnpj= @cnpj
if @@rowcount > 0
begin
    commit transaction
    return 1
end
else
begin
    rollback transaction
    return 0
end
end
else
begin
    rollback transaction
    return 0
end
end

```

/*Alterando estacionamento:*/

```

create procedure alter_estacionamento
@cnpj char(14),
@telefone char(10),
@endereco char(50),
@email char(25),
@nome char(35),
@avaliacao int
as
begin transaction
update Pjuridica
set telefone = @telefone, endereco = @endereco, email = @email, nome = @nome where cnpj =
@cnpj
if @@rowcount > 0
begin
    update estacionamento
    set avaliacao = @avaliacao where cnpj= @cnpj
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else

```

```

begin
    rollback transaction
    return 0
end

/*EXCLUSÃO:*/
/*excluindo cliente:*/
create procedure del_cliente
    @id int,
    @rg char(10)
as
begin transaction
delete cliente where rg = @rg
if @@rowcount > 0 /* exclusão de cliente bem sucedida */
begin
    delete pessoa
    where id = @id
    if @@rowcount > 0 /* exclusão de pessoa_fisica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else
begin
    rollback transaction
    return 0
end

/*excluindo atendente:*/
create procedure del_atendente
    @id int,
    @cartTrab char(10)
as
begin transaction
delete atendente where cartTrab = @cartTrab
if @@rowcount > 0 /* exclusão de atendente bem sucedida */
begin
    delete pessoa
    where id = @id
    if @@rowcount > 0 /* exclusão de pessoa_fisica bem sucedida */
    begin
        commit transaction

```

```

        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

/*excluindo seguradora:*/
create procedure del_seguradora
@cnpj char(14)
as
begin transaction
delete seguradora where cnpj = @cnpj
if @@rowcount > 0 /* exclusão de seguradora bem sucedida */
begin
    delete Pjuridica
    where cnpj = @cnpj
    if @@rowcount > 0 /* exclusão de Pjuridica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else
begin
    rollback transaction
    return 0
end

/*excluindo auto-mecanica:*/
create procedure del_mecanica
@cnpj char(14)
as
begin transaction
delete mecanica where cnpj = @cnpj
if @@rowcount > 0 /* exclusão de mecanica bem sucedida */

```



```

begin
    delete Pjuridica
    where cnpj = @cnpj
    if @@rowcount > 0 /* exclusão de Pjuridica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

```

```

/*excluindo concessionaria:*/
create procedure del_concessionaria
@cnpj char(14)
as
begin transaction
delete concessionaria where cnpj = @cnpj
if @@rowcount > 0 /* exclusão de concessionaria bem sucedida */
begin
    delete Pjuridica
    where cnpj = @cnpj
    if @@rowcount > 0 /* exclusão de Pjuridica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else
begin
    rollback transaction
    return 0
end

```

```

/*excluindo estacionamento:*/

```

```

create procedure del_estacionamento
@cnpj char(14)
as
begin transaction
delete estacionamento where cnpj = @cnpj
if @@rowcount > 0 /* exclusão de estacionamento bem sucedida */
begin
    delete Pjuridica
    where cnpj = @cnpj
    if @@rowcount > 0 /* exclusão de Pjuridica bem sucedida */
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

/*PROCEDIMENTOS ESPECIAIS:*/
/*Comprando um novo carro:*/
create procedure compra_carro
@placa char(10),
@marca char(15),
@modelo char(20),
@ano smallint,
@km int,
@combustivel int,
@valor smallmoney,
@manutencao bit,
@seguro bit,
@disponivel bit,
@codigo int,
@valorCarro smallmoney,
@dataCompra smalldatetime,
@cnpj_conc char(14)
as
begin transaction
insert into carro values(@placa, @marca, @modelo, @ano, @km, @combustivel, @valor,
@manutencao, @seguro, @disponivel)
if @@rowcount > 0

```

```

begin
    insert into NFcompra values(@codigo, @valorCarro, @dataCompra, @cnpj_conc, @placa)
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

/*Vendendo um carro:*/
create procedure vende_carro
@placa char(10),
@codigo int,
@valor smallmoney,
@dataCompra smalldatetime,
@cnpj_est char(14),
@idCliente int
as
begin transaction
delete reparo where placaCarro = @placa
if @@rowcount > 0
begin
    delete aluguel where placaCarro = @placa
    if @@rowcount > 0
    begin
        delete NFcompra where placa_carro = @placa
        if @@rowcount > 0
        begin
            delete carro where placa = @placa
            if @@rowcount > 0
            begin
                insert into NFvenda values(@codigo, @valor, @dataCompra, @cnpj_est,
@idCliente, @placa)
                if @@rowcount > 0
                begin
                    commit transaction
                    return 1
                end
            end
        end
    end
end

```



```

        begin
            commit transaction
            return 1
        end
    else
        begin
            rollback transaction
            return 0
        end
    end
end
else
begin
    rollback transaction
    return 0
end

/*alterando aluguel:*/
/*CHAMA TRIGGER PARA CALCULAR O VALOR*/
create procedure alter_aluguel_pago
@codigo int,
@dataEntrega smalldatetime,
@placaCarro char(10),
@taxas smallmoney,
@km int,
@combustivel int
as
begin transaction
update aluguel
set dataEntrega = @dataEntrega, taxas = @taxas where codigo = @codigo
if @@rowcount > 0
begin
    declare @dataDif int
    set @dataDif = DATEDIFF(day, (select dataInicio from aluguel where codigo = @codigo),
(select dataEntrega from aluguel where codigo = @codigo))
    declare @dataDifPrazo int
    set @dataDifPrazo = DATEDIFF(day,(select dataInicio from aluguel where codigo = @codigo),
(select prazo from aluguel where codigo = @codigo))
    update carro
    set disponivel = 1, km = km + @km, combustivel = @combustivel where placa = @placaCarro
    if @@rowcount > 0
    begin
        if @dataDif > @dataDifPrazo
        begin
            update aluguel
            set taxas = valor * (0.20*(@dataDif-@dataDifPrazo)) where codigo = @codigo
            if @@rowcount > 0
            begin
                commit transaction
            end
        end
    end
end
end

```

```

        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    update aluguel
    set taxas = 0 where codigo = @codigo
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
end
else
begin
    rollback transaction
    return 0
end
end

```

```

/*inserindo um novo reparo:*/
create procedure novo_reparo
@codigo int,
@dataInicio datetime,
@dataEntrega datetime,
@valor smallmoney,
@info char(30),
@placaCarro char(10),
@cnpj_mec char(14),
@cnpj_seg char(14),
@manutencao bit,
@seguro bit

```

```

as
begin transaction
insert into reparo values(@codigo, @dataInicio, @dataEntrega, @valor, @info, 1, @placaCarro,
@cnpj_mec, @cnpj_seg)
if @@rowcount > 0
begin
    update carro
    set manutencao = @manutencao, seguro = @seguro, disponivel = 0
    where placa = @placaCarro AND disponivel != 0 AND @manutencao = 1
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction
        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

/*alterando um reparo:*/
create procedure alter_reparo
@codigo int,
@dataEntrega datetime,
@placaCarro char(10)
as
begin transaction
update reparo
set dataEntrega = @dataEntrega, statusReparo = 0 where codigo = @codigo
if @@rowcount > 0
begin
    update carro
    set manutencao = 0, disponivel = 1
    where placa = @placaCarro AND disponivel = 0
    if @@rowcount > 0
    begin
        commit transaction
        return 1
    end
    else
    begin
        rollback transaction

```

```

        return 0
    end
end
else
begin
    rollback transaction
    return 0
end

```

8. TRIGGERS

USE trabPt2_BD2

/*GATILHOS:*/

/*Gatilhos para aluguel:*/

/*insert - novo aluguel*/

create trigger new_aluguel

on aluguel for insert

as

begin

update aluguel

set statusAluguel = 1 where codigo = (select codigo from inserted)

if @@rowcount = 0

rollback transaction

end

/*update - atualizar status aluguel e seu valor final*/

create trigger alteration_aluguel

on aluguel for update

as

begin

update aluguel

set statusAluguel = 0, valorFinal = (valor + taxas) where codigo = (select codigo from inserted)

if @@rowcount = 0

rollback transaction

end

/*Gatilhos para reparo:*/

/*insert - novo reparo*/

create trigger new_reparo

on reparo for insert

as

begin

update reparo

set statusReparo = 1 where codigo = (select codigo from inserted)

if @@rowcount = 0


```
        rollback transaction
    end
```

```
/*update - atualizar status reparo*/
create trigger alteration_reparo
on reparo for update
as
begin
    update reparo
    set statusReparo = 0 where codigo = (select codigo from inserted)
    if @@rowcount = 0
        rollback transaction
end
```

9. SCRIPTS

USE trabPt2_BD2

```
/*teste de tabelas*/
select * from carro
select * from cliente
select * from atendente
select * from pessoa
select * from Pjuridica
select * from mecanica
select * from seguradora
select * from concessionaria
select * from estacionamento
select * from NFcompra
select * from NFvenda
select *from aluguel
select *from reparo
```

```
/*inserindo valores nas tabelas base:*/
```

```
/*PESSOA:*/
```

```
insert into pessoa values (1, '23071923007', 40028922, 'R. das araras 22 Limeira-SP',
'math@unicamp', 'Matheus Silva')
```

```
insert into pessoa values (2, '23794823794', 23237070, 'R. leão 3 Salvador-BA',
'jon@unicamp', 'Jonatas Goes')
```

```
/*CLIENTE:*/
```

```
insert into cliente values(1, '2307192307', 20, '8093120200X', '70706633-909055-5888-9090')
```

```
/*ATENDENTE:*/
```

```
insert into atendente values(2, '3535358822', 1500.00)
```

/*P.JURIDICA:*/

insert into Pjuridica values(101010, 0800777, 'Av. maré alta 300 Limeira-SP',

'sac@seguroauto', 'Seguros Auto')

insert into Pjuridica values(202020, 0800666, 'Rua Caiçaras 149 Limeira-SP', 'sac@safework',
'SafeWork seguros')

insert into Pjuridica values(303030, 0800555, 'Rua Capitão Clemente 300 Limeira-SP',
'jonas@mecanica', 'Jonas auto-mecânica')

insert into Pjuridica values(313131, 0800555, 'Av. Senador Vergueiro 80 Limeira-SP',
'motor@mecanica', 'Motor auto-mecânica')

insert into Pjuridica values(404040, 9999111, 'Rua Leão Santos 34 Limeira-SP', 'sac@honda',
'Honda Aversa')

insert into Pjuridica values(505050, 8888111, 'Rua Senador Vergueiro 77 Limeira-SP',
'sac@renault', 'Renault Andretta')

insert into Pjuridica values(606060, 8888111, 'Av. Fabrício Vampré 1090 Limeira-SP',
'senna@estacionamento', 'Senna multimarcas')

/*SEGURADORAS:*/

insert into seguradora values(101010, 'Completo')

insert into seguradora values(202020, 'Batidas')

/*MECANICAS:*/

insert into mecanica values(303030, 'Funilaria e pintura')

insert into mecanica values(313131, 'Mecânica geral')

/*CONCESSIONARIAS:*/

insert into concessionaria values(404040, 'Honda')

insert into concessionaria values(505050, 'Renault')

/*ESTACIONAMENTOS:*/

insert into estacionamento values(606060, 4)

/*CARROS:*/

insert into carro values('WQE2B22', 'Honda', 'City', 2014, 140000, 89, 85.00, 0, 1, 1)

insert into carro values('DMM6U89', 'Renault', 'Sander RS', 2019, 87000, 65, 90.00, 0, 1, 1)

insert into carro values('JGS3K00', 'Hyundai', 'Hb20', 2016, 160000, 50, 65.00, 0, 1, 1)

insert into carro values('EMM8A77', 'Chevrolet', 'Onix', 2016, 164500, 100, 60.00, 0, 1, 1)

insert into carro values('GGW0P52', 'Ford', 'Fusion', 2020, 30000, 70, 250.00, 0, 1, 1)

/*Testando procedimentos:*/

/*teste cria_cliente:*/

declare @ret int

exec @ret = cria_cliente 3, '23093012903', '3405940022', 'Rua Jair Marques 45 Limeira-SP ',
'antonio@bol', 'Antonio Herrera', '239872323100', 34, '103390901W',
'50508833-000055-5222-8888'

print @ret

/*teste altera_cliente:*/

declare @ret2 int

```

exec @ret2 = alter_cliente 3, '23093012900', '3405940020', 'Rua Jair Marques 40 Limeira-SP',
,
'antonio@uol', 'Antonio Herrera', '239872323100', 34, '103390900W',
'50508833-000055-5222-8880'
print @ret2

```

```

/*teste del_cliente:*/
declare @ret3 int
exec @ret3 = del_cliente 3, '239872323100'
print @ret3

```

```

/*teste compra_carro(cria NFcompra):*/
declare @ret4 int
exec @ret4 = compra_carro 'FGG8A99', Volkswagen, Polo, 2020, 0, 100, 95, 0, 1, 1,1,
100000, '20210502 01:14:34:00 PM', '4356400'
print @ret4

```

```

/*teste vende_carro(cria NFvenda):*/
declare @ret5 int
exec @ret5 = vende_carro 'FGG8A99', 1, 92000, '20210902 05:19:34:00 PM', '606060', null
print @ret5

```

```

delete reparo where placaCarro = 'FGG8A99'
delete aluguel where placaCarro = 'FGG8A99'
delete NFcompra where placa_carro = 'FGG8A99'
delete carro where placa = 'FGG8A99'
insert into NFvenda values(1, 92000, '20210902 05:19:34:00 PM', '606060', null, 'FGG8A99')

```

```

/*ALUGUEIS - criando um novo e atualizando:*/
declare @ret6 int
exec @ret6 = novo_aluguel 1, '20210908 03:14:34:00 PM', null, '20210910 03:14:34:00 PM', 1,
'GGW0P52', 1, 2, 0, 1
print @ret6

```

```

/*teste alter_aluguel_pago:*/
declare @ret7 int
exec @ret7 = alter_aluguel_pago 1, '20210911 03:14:34:00 PM', 'GGW0P52', 0, 200, 60
print @ret7

```

```

/*REPAROS - criando um novo e atualizando:*/
/*teste novo_reparo:*/
declare @ret8 int
exec @ret8 = novo_reparo 1, '20210915 03:14:34:00 PM', null, 600, 'ajuste de embreagem',
'JGS3K00', 313131, 101010, 1, 1

```

```
print @ret8
```

```
/*teste alter_reparo:*/  
declare @ret9 int  
exec @ret9 = alter_reparo 1, '20210917 04:24:34:00 PM', 'JGS3K00'  
print @ret9
```

10.VIEWS

```
create view topcarros as
```

```
SELECT  
modelo,  
marca,  
placa,  
km  
FROM carro;
```

```
CREATE VIEW carrosdisponiveis  
as  
select *  
from carro  
where disponivel =1;
```

```
CREATE VIEW carrosindisponiveis  
as  
select *  
from carro  
where disponivel =0 ;
```

```
select * from carrosindisponiveis;
```

```
CREATE VIEW Carromanutenções  
as  
select  
carro.marca,  
carro.modelo,  
reparo.dataInicio,  
reparo.dataEntrega,  
reparo.statusReparo,  
reparo.placaCarro
```

```
from repara
```

```
inner join carro on  
reparo.placaCarro = carro.placa;
```