

Departamento de Ciência da Computação
UFLA - Universidade Federal de Lavras
GCC214 - Introdução a Sistemas de Banco de Dados Prof. Denilson
Alves Pereira
Trabalho Prático – Etapa 2 – Modelo Relacional

Tema: Banco de dados de uma Imobiliária

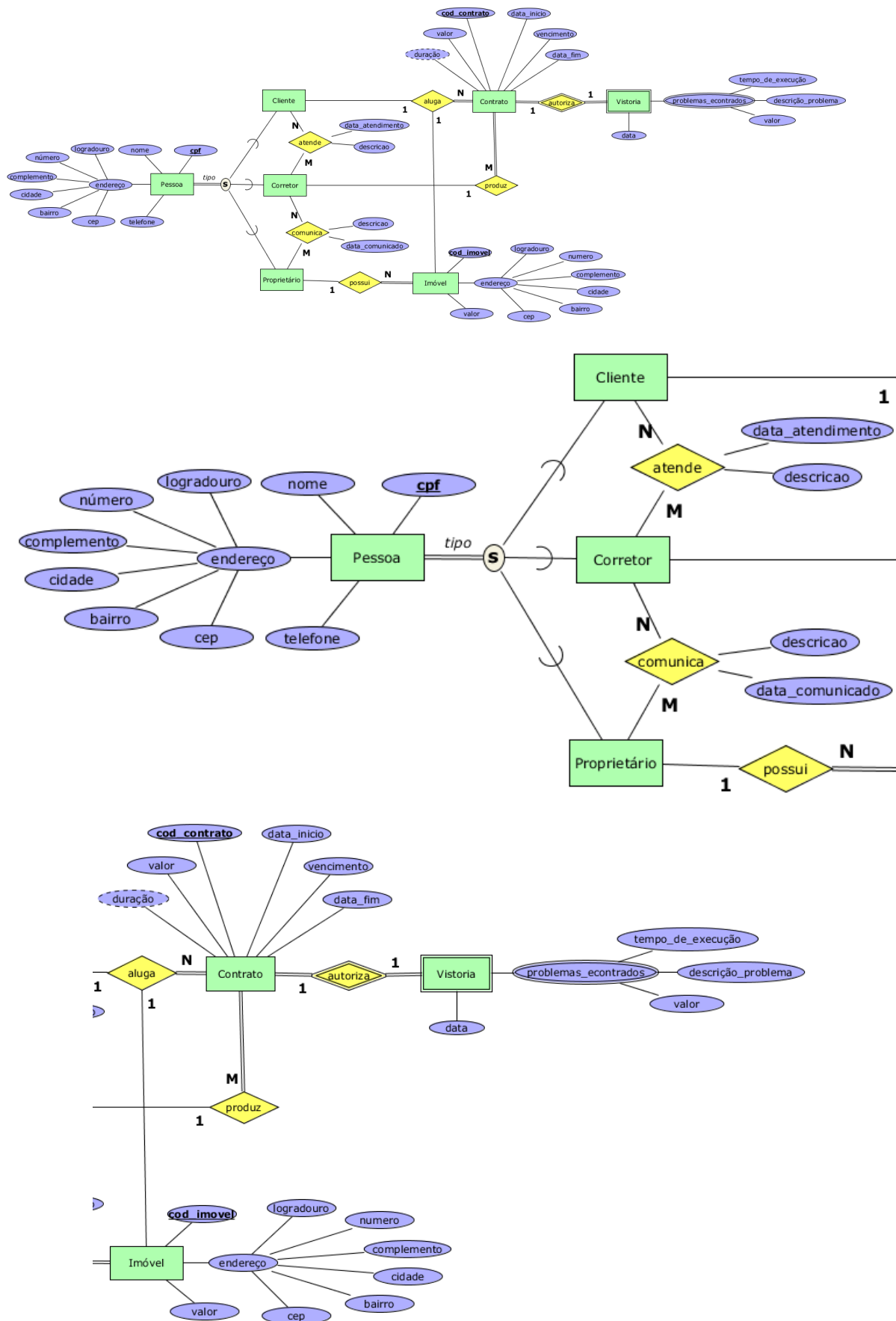
Alunos:

BRUNO DE ALMEIDA DE PAULA - 201920350
LUCAS NEVES SÁBER GABRIEL - 202020459
OTÁVIO AUGUSTO TRINDADE FONSECA - 202020551
THIAGO ODILON DE ALMEIDA - 202021025
TULIO OLIVEIRA CRUZ - 202020094

Descrição:

Levantamento de dados de uma imobiliária para um sistema de controle de aluguéis (com dados de corretores, contratos, clientes, proprietários, imóveis e vistorias). Dentro do sistema existem divisões: os corretores, clientes e proprietários possuem nome, cpf, telefone e endereço, sendo cpf de extrema importância para a identificação deles. Cada proprietário pode possuir diversos imóveis, cada imóvel tendo em seus dados um código, que serve para sua identificação, endereço e seu respectivo valor, o corretor comunica com o proprietário e atende o cliente, cada um desses atendimentos e comunicações tem uma data específica, gerando assim um contrato de aluguel, esse contrato possui um código, que é de extrema importância para a identificação de cada contrato, valor, duração, data de início e data de fim. Além disso, cada contrato possui uma vistoria, que depende da existência de um contrato para ser feita, nela consta, datas, problemas encontrados (podendo existir diversos problemas e as características de cada um), seu tempo de duração, valor e a descrição dos problemas, se encontrados.

Diagrama ER:

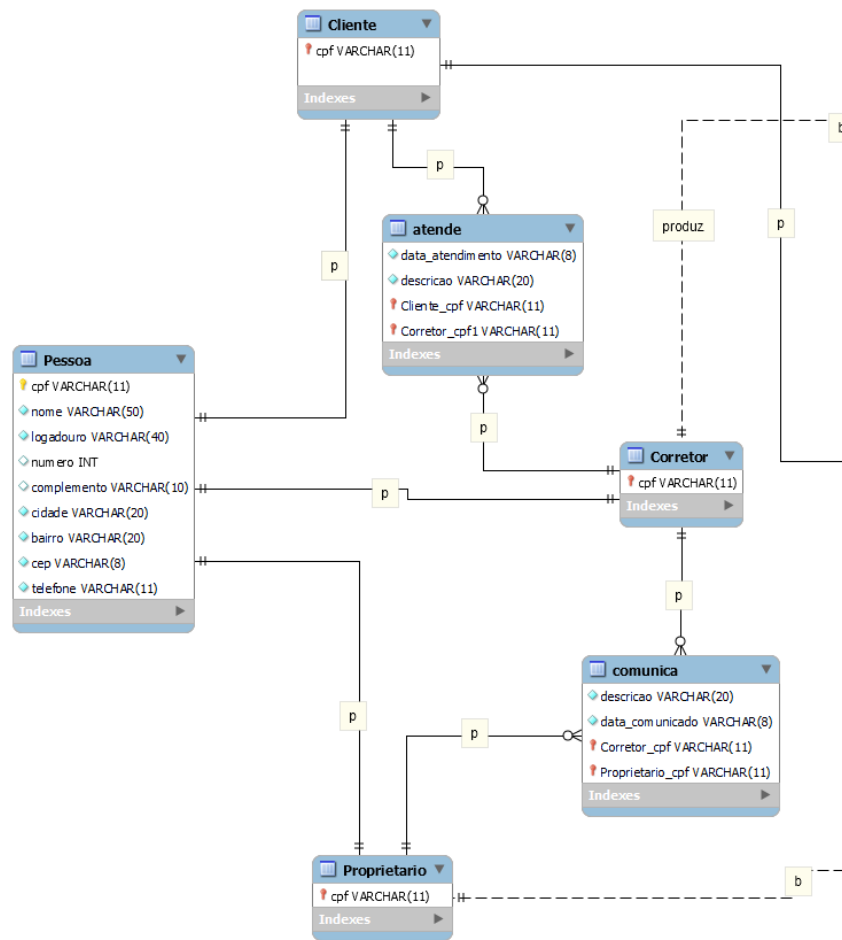


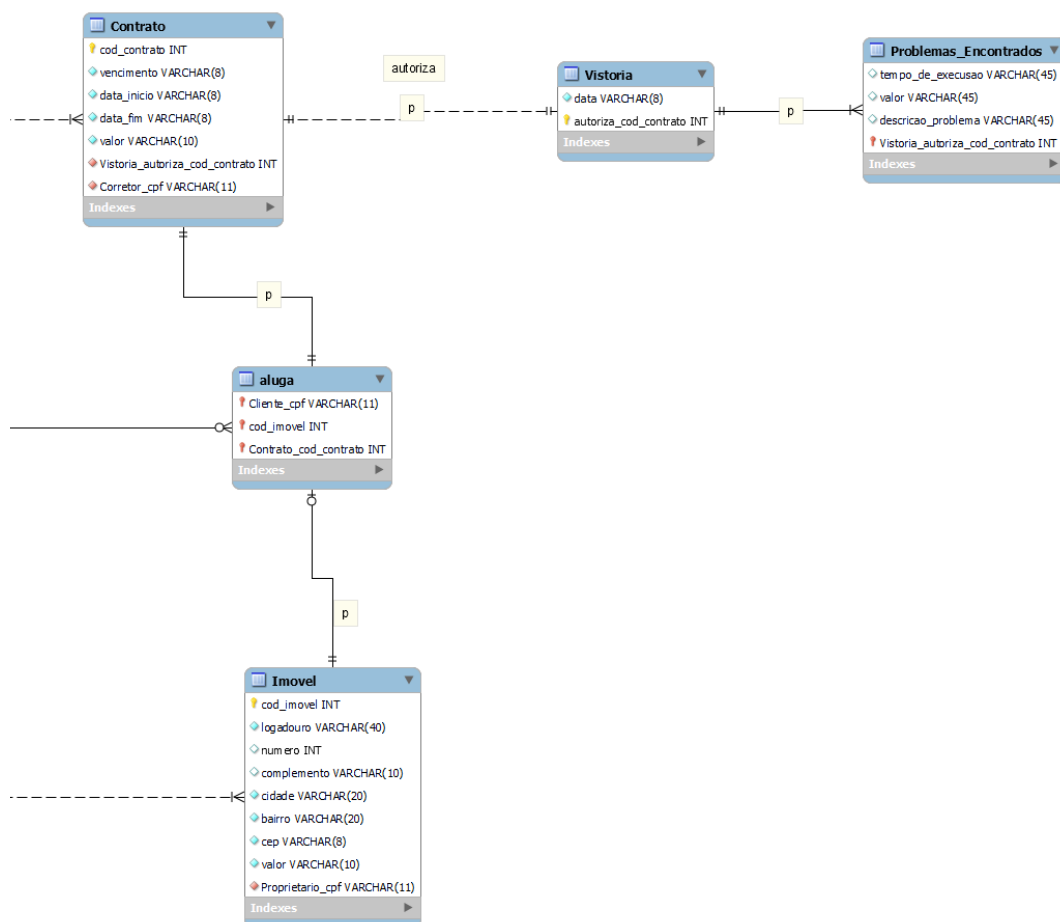
Dicionário de Dados:

Entidades			
Tipo Entidade	Pessoa		
Descrição	Conjunto de pessoas de uma imobiliária		
Atributos			
Nome	Descrição	Domínio	Permite nulo?
cpf	Cadastro de pessoa física	Texto (11) Formato: xxx.xxx.xxx-xx	Não
nome	Nome da pessoa	Texto (50)	Não
logradouro	Nome do logradouro (rua, avenida, alameda etc.)	Texto (40)	Não
número	Número da residência	Inteiro (4) positivo	Não
complemento	Tipo de residência	Texto (10)	Sim
cidade	Nome da cidade	Texto (20)	Não
bairro	Nome do bairro	Texto (20)	Sim
cep	Código de Endereçamento Postal	Texto (8) Formato: xx.xxx-xxx	Não
telefone	Número de telefone	Texto (11) Formato: (xx)xxxxx-xxxx	Não
Tipo Entidade	Contrato		
Descrição	Contrato entre corretora e cliente		
Atributos			
Nome	Descrição	Domínio	Permite nulo?
vencimento	Vencimento do Contrato	Texto (8) Formato: xx/xx/xxxx	Não
cod_contrato	código de informação do contrato	Inteiro (6) positivo	Não
data_inicio	Data de início de contrato	Texto (8) Formato: xx/xx/xxxx	Não
data_fim	Data final de contrato	Texto (8) Formato: xx/xx/xxxx	Não
valor	Valor mensal	Texto (10) Formato: R\$xxxx,xx	Não
Tipo Entidade	Imóvel		
Descrição	Imóvel do proprietário		

Atributos			
Nome	Descrição	Domínio	Permite nulo?
cod_imovel	Código de cadastro na imobiliária	Inteiro (6) positivo	Não
logradouro	Nome do logradouro (rua, avenida, alameda etc.)	Texto (40)	Não
número	Número da Residência	Inteiro (4) positivo	Não
complemento	Identificação Apartamento/Casa	Texto (10)	Sim
cidade	Nome da cidade	Texto (20)	Não
bairro	Nome do bairro	Texto 20)	Sim
cep	Código de Endereçamento Postal	Inteiro (8) positivo Formato: xx.xxx-xxx	Não
valor	Valor do aluguel	Texto (10) Formato: R\$xxxx,xx	Não
Tipo Entidade	Vistoria		
Descrição	Necessária para a produção do contrato		
Atributos			
Nome	Descrição	Domínio	Permite nulo?
data	Data da Vistoria	Texto (8) Formato: xx/xx/xxxx	Não
tempo_de_execução	Tempo de Execução da vistoria	Texto (4) Formato: xx:xx	Sim
valor	Valor da vistoria	Texto (10) Formato: R\$xxxx,xx	Sim
descrição_problema	Detalhes do que foi observado na vistoria	Texto (200)	Não

Relacionamento			
Tipo Relacionamento	atende		
Descrição	Relação entre Corretor e Cliente		
Atributos			
Nome	Descrição	Domínio	Permite nulo?
data_atendimento	Indica a data de atendimento	Data	Não
descricao	Descreve como foi o atendimento	Texto	Não
Tipo Relacionamento	comunica		





Dicionário de dados - Relacional

Tabela	Pessoa
Descrição	Indica o tipo de pessoa, se é cliente, corretor ou proprietário
Atributos	
Nome	Descrição
cpf	Cadastro de Pessoa Física
nome	Nome da Pessoa
logradouro	Nome do logradouro (rua, avenida, etc)
numero	Numero da Residencia
complemento	Tipo de Residência (casa, apartamento, etc)
cidade	Nome da Cidade
bairro	Nome do Bairro
cep	Código de endereçamento postal
telefone	Telefone pessoal

Tabela	Contrato
Descrição	Registra as informações sobre o contrato feito
Atributos	
Nome	Descrição
cod_contrato	Código para identifica o contrato
vencimento	Data que a validade do contrato expira
data_inicio	Data de inicio do contrato
data_fim	Data de fim do contrato
valor	Valor total do contrato
vistoria_autoriza_cod_contrato	Verifica se a vistoria autorizou ou não o fechamento do contrato

Tabela	Imóvel
Descrição	Propriedade que pertence ao locador
Atributos	
Nome	Descrição
cod_imovel	Código de cadastro na imobiliária
logradouro	Nome do logradouro (rua, avenida, alameda etc.)
número	Número da Residência
complemento	Identificação Apartamento/Casa
cidade	Nome da cidade
bairro	Nome do bairro
cep	Código de Endereçamento Postal
valor	Valor do aluguel

Tabela	Vistoria
Descrição	Avalia as condições do imóvel para fechar o contrato
Atributos	
Nome	Descrição
data	Data de realização da vistoria
tempo_de_execucao	Tempo que levou para realizar a vistoria
valor	Valor da vistoria
descricao_problema	Descreve qual problema precisa ser resolvido
autoriza_cod_contrato	Conclusão da vistoria para fechar o contrato

Tabela	Atende
Descrição	Relação entre Corretor e Cliente
Atributos	
Nome	Descrição
data_atendimento	Indica a data de atendimento
descricao	Descreve como foi o atendimento

Tabela	Comunica
Descrição	Relação entre Proprietário e Corretor
Atributos	
Nome	Descrição
data_comunicado	Indica a data que ocorreu o comunicado
descricao	Descreve como foi o comunicado

Tabela	Aluga
Descrição	Indica o tipo de contrato com o cliente, com o imóvel, e vice-versa

Tabela	possui
Descrição	Relação entre Proprietário e Corretor para indicar os imóveis que o locador possui

Tabela	produz
Descrição	Relação entre Corretora e Contrato para criar o contrato de locação

Tabela	autoriza
Descrição	Relação entre Contrato e Vistoria para aprovar a locação

Comandos SQL

A -Criação de todas as tabelas

-- CRIA TABELA PESSOA--

```
CREATE TABLE IF NOT EXISTS Pessoa (  
    cpf      INT(11)    NOT NULL PRIMARY KEY,  
    nome     VARCHAR(50) NOT NULL,  
    logadouro VARCHAR(40) NOT NULL,  
    numero   INT        NULL,  
    complemento VARCHAR(10) NULL,  
    cidade   VARCHAR(30) NOT NULL,  
    bairro   VARCHAR(20) NOT NULL,  
    cep      VARCHAR(8)  NOT NULL,  
    telefone VARCHAR(11)  NOT NULL,  
    CONSTRAINT uq_Telefone UNIQUE (nome)  
);
```

-- Cria tabela de Corretores--

```
CREATE TABLE IF NOT EXISTS Corretor (  
    cpf      INT(11)    NOT NULL,  
  
    CONSTRAINT pk_Corretor PRIMARY KEY (cpf),  
    CONSTRAINT cpf_corretor FOREIGN KEY (cpf)  
        REFERENCES Pessoa (cpf)  
    ON DELETE NO ACTION  
);
```

-- Cria tabela de Clientes--

```
CREATE TABLE IF NOT EXISTS Cliente (  
    cpf      INT(11)    NOT NULL,  
  
    CONSTRAINT pk_Cliente PRIMARY KEY (cpf),  
    CONSTRAINT cpf_cliente FOREIGN KEY (cpf)  
        REFERENCES Pessoa (cpf)  
    ON DELETE CASCADE  
);
```

-- Cria tabela de Proprietários--

```
CREATE TABLE IF NOT EXISTS Proprietario (  
    cpf      INT(11)    NOT NULL,  
  
    CONSTRAINT pk_Corretor PRIMARY KEY (cpf),
```

```

        CONSTRAINT cpf_proprietario FOREIGN KEY (cpf)
            REFERENCES Pessoa (cpf)
            ON DELETE NO ACTION
    );

-- Cria tabela da Relação Atende (Corretor -> Cliente)--
CREATE TABLE IF NOT EXISTS Atende (
    data_atendimento    VARCHAR(10)    NOT NULL,
    descricao            VARCHAR(40)    NOT NULL,
    Cliente_cpf          INT(11)        NOT NULL,
    Corretor_cpf         INT(11)        NOT NULL,

    CONSTRAINT fk_cpfcliente FOREIGN KEY (Cliente_cpf)
        REFERENCES Cliente (cpf)
        ON DELETE NO ACTION,
    CONSTRAINT fk_cpfcorretor FOREIGN KEY (Corretor_cpf)
        REFERENCES Corretor (cpf)
        ON DELETE NO ACTION
);

-- Cria tabela da Relação Comunica (Corretor -> Proprietario)--
CREATE TABLE IF NOT EXISTS Comunica (
    descricao_comunicado VARCHAR(30)    NOT NULL,
    data_comunicado       VARCHAR(10)    NOT NULL,
    Corretor_cpf           INT(11)        NOT NULL,
    Proprietario_cpf       INT(11)        NOT NULL,

    PRIMARY KEY (Corretor_cpf, Proprietario_cpf),

    CONSTRAINT cpfProp FOREIGN KEY (Proprietario_cpf)
        REFERENCES Proprietario (cpf)
        ON DELETE CASCADE,
    CONSTRAINT cpfCorre FOREIGN KEY (Corretor_cpf)
        REFERENCES Corretor (cpf)
        ON DELETE CASCADE
);

-- Cria tabela de Vistoria--
CREATE TABLE IF NOT EXISTS Vistoria (
    data_vistoria          VARCHAR(10)    NOT NULL,
    autoriza_cod_contrato  INT             NOT NULL,

    CONSTRAINT pk_vistoria PRIMARY KEY (autoriza_cod_contrato)
);

```

```
);

-- Cria tabela de Problemas_Encontrados--
CREATE TABLE IF NOT EXISTS Problemas_Encontrados (
    tempo_de_execucao          VARCHAR(45)
    NULL,
    valor                      VARCHAR(45)
    NULL,
    descricao_problema        VARCHAR(45)
    NULL,
    Vistoria_autoriza_cod_contrato INT          NOT NULL,

    CONSTRAINT Vistoria_autoriza_cod FOREIGN KEY
(Vistoria_autoriza_cod_contrato)
    REFERENCES Vistoria (autoriza_cod_contrato)
    ON DELETE CASCADE
);
```

```
-- Cria tabela de Contratos--
CREATE TABLE IF NOT EXISTS Contrato (
    cod_contrato      INT          NOT NULL AUTO_INCREMENT,
    vencimento        VARCHAR(10)  NOT NULL,
    data_inicio       VARCHAR(10)  NOT NULL,
    data_fim          VARCHAR(10)  NOT NULL,
    valor             VARCHAR(10)  NOT NULL,
    Vistoria_autoriza_cod_contrato INT NOT NULL,
    Corretor_cpf      INT(11)      NOT NULL,

    CONSTRAINT pk_contrato PRIMARY KEY (cod_contrato),

    CONSTRAINT cpfCorretor FOREIGN KEY (Corretor_cpf)
    REFERENCES Corretor (cpf)
    ON DELETE CASCADE
);
```

```
-- Cria tabela de Imóveis--
CREATE TABLE IF NOT EXISTS Imovel (
    cod_imovel      INT          NOT NULL,
    logadouro       VARCHAR(40)  NOT NULL,
    numero          INT          NOT NULL DEFAULT 0000,
    complemento     VARCHAR(10)  NULL,
    cidade          VARCHAR(30)  NOT NULL,
    bairro          VARCHAR(20)  NOT NULL,
```

```

cep                VARCHAR(8)                NOT NULL,
valor              VARCHAR(10)               NOT NULL,
Proprietario_cpf   INT(11)                   NOT NULL,

```

```

    CONSTRAINT pk_imovel PRIMARY KEY (cod_imovel),
    CONSTRAINT cpfProprietario FOREIGN KEY (Proprietario_cpf)
        REFERENCES Proprietario (cpf)
        ON DELETE CASCADE

```

```
);
```

-- Cria tabela da relação Aluga--

```
CREATE TABLE IF NOT EXISTS Aluga (
```

```

    Cliente_cpf        INT(11)                NOT NULL,
    cod_contrato        INT                    NOT NULL,
    cod_imovel          INT                    NOT NULL,

```

```

    CONSTRAINT pk_aluga PRIMARY KEY (Cliente_cpf, cod_contrato, cod_imovel),
    CONSTRAINT cpfcliente FOREIGN KEY (Cliente_cpf)

```

```
    REFERENCES Cliente (cpf)
```

```
    ON DELETE NO ACTION,
```

```
    CONSTRAINT codcontrato FOREIGN KEY (cod_contrato)
```

```
    REFERENCES Contrato (cod_contrato)
```

```
    ON DELETE NO ACTION,
```

```
    CONSTRAINT codimovel FOREIGN KEY (cod_imovel)
```

```
    REFERENCES Imovel (cod_imovel)
```

```
    ON DELETE NO ACTION

```

```
);
```

B -Exemplos de ALTER TABLE

```
CREATE TABLE IF NOT EXISTS cidadao(
```

```
    cpf varchar(11),
```

```
    nome varchar(50),
```

```
    telefone varchar(11),
```

```
    CONSTRAINT pk_Cidadao PRIMARY KEY (cpf)
```

```
);
```

```
INSERT INTO cidadao VALUE
```

```
('12345678900', 'Carlos', '32991205566'),
```

```
('12345678000', 'Sansao', '32991205580'),
```

```
('42345678900', 'Bene', '32991205589');
```

```
SELECT * FROM cidadao;
```

```
-- /1º exemplo alteracao/
ALTER TABLE cidadao
ADD COLUMN profissao varchar(10);

-- /2º exemplo alteracao/
ALTER TABLE cidadao
DROP COLUMN profissao;

-- /3º exemplo alteracao/
ALTER TABLE cidadao
ADD COLUMN graduacao varchar(10) after nome;

-- /4 º exemplo alteracao/
ALTER TABLE cidadao
ADD COLUMN codigo int first;

-- /Apagar tabela/
DROP TABLE IF EXISTS cidadao;
```

C -Exemplos de inserção de dados em cada uma das tabelas

```
INSERT INTO Pessoa
(cpf, nome, logadouro, numero, complemento, cidade, bairro, cep, telefone)
VALUES
('130979444', 'Joao', 'rua Joao Almeida', '155','Casa', 'Lavras', 'Jardim Gloria',
'37200001', '032990205689'),
('130794555', 'Pedro', 'rua Joao Gonçalves', '156','Apartamento 204', 'Lavras',
'Alterosa', '37200000', '032991205689'),
('130794666', 'Paulo', 'rua Olavio Bilac', '333','Sobrado', 'Lavras', 'Jardim Gloria',
'37202895', '032992205689'),
('130794777', 'Caio', 'rua Almeida silva', '', 'Casa S/N', 'Lavras', 'Barreiro', '3720865',
'032991305689'),
('130794888', 'Bene', 'rua Judite Terezinha', '111','Casa', 'Lavras', 'Centro',
'37200000', '032991405689'),
('130794999', 'Otavio', 'rua Albertop Valentim', '916','Apartamento 506', 'Lavras',
'Vila do Toco', '37200-000', '032991505689'),
('130794101', 'Rafael', 'rua Doutor Sales', '89', ' ', 'Lavras', 'Jardim Gloria',
'37202865', '032991265689'),
('130794111', 'Bruno', 'rua Elio Alves Vilela', '155','Casa', 'Lavras', 'Cardek',
'37200000', '033991205689'),
('130794121', 'Matue', 'rua do Direito', '666','Sobrado', 'Lavras', 'Jardim Gloria',
'37202867', '033991705689'),
('130794131', 'Snider', 'rua Padre Damiao', '158','Casa', 'Lavras', 'Multirao',
'37202867', '035991909689');
```

INSERT INTO Pessoa

(cpf, nome, logadouro, numero, complemento, cidade, bairro, cep, telefone)

VALUES

('130979141', 'Mari Clara', 'rua Joao Almeida', '135','Casa', 'Lavras', 'Jardim Gloria', '37200001', '031990205689'),

('130797555', 'Pedro Paulo', 'rua Carlos Alves', '89','Apartamento 204', 'Lavras', 'Belvedere', '37200005', '031991205689'),

('130797666', 'Antonio', 'rua Olavio Bento', '665','Sobrado', 'Lavras', 'Belvedere', '37200005', '033992205689'),

('130797777', 'Jailton', 'Alameda weed','89','Casa B', 'Lavras', 'Barreiro', '3720865', '035991305689'),

('130797888', 'Carlus', 'Avenida Doutor Silvio Meniccuci', '74','Casa', 'Lavras', 'Vila Ester', '37202865', '035995041689'),

('130797999', 'Hugo', 'rua Alberto Calos', '14','Apartamento 506', 'Lavras', 'Vila do Toco', '37200-000', '031991505689'),

('130797101', 'Rosa', 'rua Doutor Candido Mato', '159',' ', 'Lavras', 'Jardim Gloria', '37202865', '031991265689'),

('130797111', 'Gilberto', 'rua Elio Alves Vilela', '356','Casa', 'Lavras', 'Cardek', '37200000', '031991205689'),

('130797121', 'Idme', 'Avenida Carlos Chagas', '468','Sobrado', 'Lavras', 'Jardim Gloria', '37202867', '031991705689'),

('130797131', 'Gorete', 'rua Padre Miguel', '478','Casa', 'Lavras', 'Multirao', '37202867', '031991805689'),

('130978444', 'Cleonice', 'rua Joao Almeida', '568','Casa', 'Lavras', 'Jardim Gloria', '37200001', '035990205689'),

('130798555', 'Margarida', 'rua Joao Gonçalves', '12','Apartamento 204', 'Lavras', 'Alterosa', '37200000', '035991205689'),

('130798666', 'Diogo', 'rua Olavio Bilac', '25','Sobrado', 'Lavras', 'Jardim Gloria', '37202895', '035992205689'),

('130798777', 'Thiago', 'rua Almeida silva','65',' ', 'Lavras', 'Barreiro', '3720865', '035991305689'),

('130798888', 'Caputo', 'rua Judite Terezinha', '456','Casa', 'Lavras', 'Centro', '37200000', '035991405689'),

('130798999', 'Fernanda', 'rua Albertop Valentim', '4925','Apartamento 506', 'Lavras', 'Vila do Toco', '37200-000', '035991505689'),

('130798101', 'Maju', 'rua Doutor Sales', '32','Apartamento 506 ', 'Lavras', 'Jardim Gloria', '37202865', '035991265689'),

('130798111', 'Marina', 'rua Elio Alves Vilela', '153','Casa', 'Lavras', 'Cardek', '37200000', '035991205689'),

('130798121', 'Sofia', 'rua do Direito', '667','Casa', 'Lavras', 'Jardim Gloria', '37202867', '035991705689'),

('130798131', 'Isabela', 'rua Padre Damiao', '18','Casa', 'Lavras', 'Multirao', '37202867', '035991805689');

INSERT INTO Corretor

(cpf)

VALUES

('130979444'),

('130794555'),

('130794666'),

('130794777'),

('130794888'),

('130794999'),

('130794101'),

('130794111'),

('130794121'),

('130794131');

INSERT INTO Cliente

(cpf)

VALUES

('130979141'),

('130797555'),

('130797666'),

('130797777'),

('130797888'),

('130797999'),

('130797101'),

('130797111'),

('130797121'),

('130797131');

INSERT INTO Proprietario

(cpf)

VALUES

('130978444'),

('130798555'),

('130798666'),

('130798777'),

('130798888'),

('130798999'),

('130798101'),

('130798111'),

('130798121'),

('130798131');


```

INSERT INTO Atende
    ( data_atendimento, descricao, Cliente_cpf, Corretor_cpf)
VALUES
    ('31/01/2021','Cliente solicitou casa no Jardim Gloria'
, '130979141','130794555'),
    ('08/02/2022','Cliente solicitou casa no Alterosa'
, '130797555','130794999'),
    ('25/01/2020','Cliente solicitou casa no Jardim Gloria'
, '130797666','130794999'),
    ('04/10/2021','Cliente solicitou 1 suite'
, '130797666','130794131'),
    ('24/02/2022','Cliente solicitou casa no Centro'
, '130797777','130794555'),
    ('03/01/2022','Cliente solicitou casa Com 3 Quartos','130797888','130794999'),
    ('26/05/2021','Cliente solicitou casa no Jardim Gloria'
, '130797999','130794555'),
    ('16/11/2020','Cliente solicitou casa no Cardek'
, '130797101','130794111'),
    ('13/08/2021','Cliente solicitou casa no Jardim Gloria'
, '130797121','130794777'),
    ('19/04/2022','Cliente solicitou Aluguel até R$3.000' , '130797111','130794111');

```

```

INSERT INTO Comunica
    (descricao_comunicado, data_comunicado, Corretor_cpf, Proprietario_cpf)
VALUES
    ('Cliente com Problema'      , '01/03/2022','130794555','130978444'),
    ('Taxa da empresa'           , '05/03/2022','130794555','130798555'),
    ('Aprovação de comprovante'  , '21/09/2020','130794101','130798777'),
    ('Aprovado'
, '13/03/2020','130794555','130798888'),
    ('Aprovado'
, '15/05/2021','130794999','130798666'),
    ('Aprovado'
, '20/06/2021','130794101','130798101'),
    ('Aprovado'
, '31/08/2020','130979444','130798666'),
    ('Aprovado'
, '09/10/2022','130794101','130798121'),
    ('Problema com o fiador' , '10/07/2022','130794131','130798131'),
    ('Aprovado'
, '09/03/2022','130979444','130798111');

```

-- Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`imobiliaria`.`comunica`, CONSTRAINT `cpfCorre` FOREIGN KEY (`Corretor_cpf`) REFERENCES `corretor` (`cpf`) ON DELETE CASCADE)

INSERT INTO Vistoria

```
(data_vistoria, autoriza_cod_contrato)
VALUES
('05/03/2022','1'),
('21/09/2020','2'),
('13/03/2020','3'),
('15/05/2021','4'),
('20/06/2021','5'),
('31/08/2020','6'),
('09/10/2022','7'),
('10/07/2022','8'),
('09/03/2022','9'),
('18/05/2022','10');
```

INSERT INTO problemas_encontrados

```
(tempo_de_execucao,valor,descricao_problema,Vistoria_autoriza_cod_contrato)
VALUES
('Um mes'      , '500', 'Vazamento', '1'),
('Um mes'      , '1000', 'Piso Quebrado', '2'),
('Um mes'      , '100', 'Torneira Quebrada', '3'),
('Uma semana'  , '650', 'Goteira', '4'),
('Dois meses'  , '1250', 'Janela Danificada', '5'),
('Quatro dias' , '800', 'Problema com fiacao', '6'),
('90 disa'     , '1050', 'Trinca na parede', '7'),
('Um mes'      , '2958', 'Muro quebrado', '8'),
('Duas semanas', '2870', 'Problema com rede Hidraulica', '9'),
('',' ','10');
```

INSERT INTO Contrato

```
(cod_contrato, vencimento, data_inicio, data_fim, valor,
Vistoria_autoriza_cod_contrato, Corretor_cpf)
VALUES
('1','01/06/2023', '01/05/2022', '25/05/2023', '1500', '1', '130979444'),
('2','02/12/2023', '02/05/2023', '24/06/2023', '1600', '2', '130794555'),
('3','03/10/2023', '03/05/2023', '23/06/2023', '1650', '3', '130794666'),
('4','05/05/2023', '05/05/2023', '22/10/2023', '1500', '4', '130794777'),
('5','04/05/2023', '04/05/2023', '21/12/2023', '2600', '5', '130794888'),
('6','12/05/2023', '12/05/2023', '20/06/2023', '1350', '6', '130794999'),
('7','11/05/2023', '11/05/2023', '19/08/2023', '1500', '7', '130794101'),
```

```
('8','13/05/2023','13/05/2023','18/04/2025','20000','8'      , '130794111'),
('9','14/05/2023','14/05/2023','17/04/2024','1500','9'      , '130794121'),
('10','16/02/2024','15/05/2023','16/02/2024','3700','10','130794131');
```

INSERT INTO Imovel

(cod_imovel , logadouro, numero, complemento, cidade, bairro, cep, valor, Proprietario_cpf)

VALUES

```
('1', 'rua Capitao Valter Cunha', '741', 'Casa', 'Visconde do Rio Branco',
'Lagoinha', '36520000', '1500'      , '130978444'),
('2', 'rua Pedro Domiciano', '23', 'Casa', 'Visconde do Rio Branco', 'Barreiro',
'36520000', '1500'      , '130798555'),
('3', 'rua Domiciano Dutra', '320', 'Apartamento', 'Visconde do Rio Branco', 'Caxias',
'36720000', '1500'      , '130798666'),
('4', 'rua Marechal Deodoro', '600', 'Sobrado', 'Visconde do Rio Branco', 'Capibari',
'36520890', '1500'      , '130798777'),
('5', 'rua Treze de maio', '745', 'Casa', 'Visconde do Rio Branco',
'Barreiro', '36520000', '200000'      , '130798888'),
('6', 'rua Paulista' , '641', 'Casa', 'Visconde do Rio Branco', 'Chacara', '32720853',
'4500'      , '130798999'),
('7', 'rua Alberto Dr. Alberto Vargas', '51', 'Casa', 'Visconde do Rio
Branco', 'Chacara', '36520000', '1500'      , '130798101'),
('8', 'rua Antonio Maria', '563', 'Apartamento', 'Visconde do Rio Branco',
'Centro', '36520000', '1500'      , '130798111'),
('9', 'rua Valdemar Barbosa', '779', 'Casa', 'Visconde do Rio Branco', 'centro',
'36520200', '3000'      , '130798121'),
('10', 'rua Alzira Dutra', '235', 'Sobrado', 'Visconde do Rio Branco',
'Planalto', '3652030', '5000'      , '130798131');
```

INSERT INTO Aluga

(Cliente_cpf, cod_contrato, cod_imovel)

VALUES

```
('130979141', '1', '1'),
('130797555', '2', '2'),
('130797666', '3', '3'),
('130797777', '4', '4'),
('130797888', '5', '5'),
('130797999', '6', '6'),
('130797555', '7', '7'),
('130797111', '8', '8'),
('130797121', '9', '9'),
('130797666', '10', '10');
```

D -Exemplos de modificação de dados

1. Altere a cidade e o telefone da pessoa de nome 'Otavio'.

```
UPDATE Pessoa
SET cidade = 'São Paulo', telefone = '11999231257'
WHERE nome = 'Otavio';
```

```
-- TESTE DE FUNCIONAMENTO
select nome, telefone, cidade
from pessoa
where nome = 'Otavio';
```

2. Altere o valor do imóvel de código 5.

```
UPDATE Imovel
SET valor = '2300.00'
WHERE cod_imovel = 5;
```

```
-- TESTE DE FUNCIONAMENTO
select cod_imovel, valor
from Imovel
where cod_imovel = 5;
```

3. Altere o valor do contrato do imóvel de cod 2

```
UPDATE Contrato
SET valor = '1.567,90'
WHERE cod_contrato IN (SELECT cod_contrato
                        FROM Aluga NATURAL JOIN Imovel
                        WHERE cod_imovel= 23);
```

```
-- TESTE DE FUNCIONAMENTO
select cod_imovel, Proprietario_cpf
from Imovel
where cod_imovel = 2;
```

4. Aumente o valor do imóvel de código 10 em 15%.

```
UPDATE Imovel
SET valor = valor*1.15
WHERE cod_imovel = 10;
```

```
-- TESTE DE FUNCIONAMENTO
select cod_imovel, valor
from Imovel
where cod_imovel = 10;
```

5. Altere o vencimento do contrato de código 2 e de valor R\$2.540,00.

```
UPDATE Contrato
SET vencimento = '23/09/2022'
WHERE cod_contrato = 2 AND valor = '2540.00';
```

```
-- TESTE DE FUNCIONAMENTO
select cod_contrato, valor, vencimento
from Contrato
where cod_contrato =2;
```

E - Exemplos de exclusão de dados

#1. Excluir imóveis de cep "36520000".

```
DELETE FROM Imovel
WHERE cep = '36520000';
```

#2. Excluir pessoa com nome 'Jeovana Santos'.

```
DELETE FROM Pessoa
WHERE cpf = '130.797.111';
```

#3. Excluir contrato com a data inicial 'X' e de data final 'Y'.

```
DELETE FROM Contrato
WHERE data_inicio = '01/05/2022' AND data_fim = '25/04/2023';
```

#4. Excluir imóveis de número residencial entre 4501 até 5061

```
DELETE FROM Imovel
WHERE numero >= 600 AND numero <= 800;
```

#5. Excluir imóveis do proprietário de cpf '130798666'

```
DELETE FROM Imovel
WHERE Proprietario_cpf IN (SELECT cpf
                           FROM Proprietario
                           WHERE cpf = '130798666');
```

F - Exemplos de consultas

1- recupera o código de imóvel o preço dos imóveis, do mais barato ao mais caro

```
SELECT cod_imovel, valor
FROM Imovel
ORDER BY valor ASC;
```

2- recupera o nome do cliente, o código do imóvel e o valor pago

```
SELECT Pessoa.nome AS nomeCliente, Aluga.cod_imovel, valor
FROM Pessoa NATURAL JOIN Cliente JOIN Aluga JOIN Imovel
```

```
WHERE Aluga.Cliente_cpf = Cliente.cpf AND Aluga.cod_imovel =  
Imovel.cod_imovel;
```

```
# 3- recupera a media dos valores de contrato que o corretor '...' concluiu  
SELECT AVG(valor)  
FROM Pessoa P NATURAL JOIN Corretor C JOIN Contrato ct ON C.cpf =  
ct.Corretor_cpf  
WHERE P.nome = 'Joao';
```

```
# 4- retorna a cidade e o preço de cada imovel do proprietario ...  
SELECT I.cidade, valor  
FROM Pessoa NATURAL JOIN Proprietario P JOIN Imovel I on P.cpf =  
I.Proprietario_cpf  
WHERE Pessoa.nome = 'Cleonice';
```

-- ATE AQUI OK--

```
# 5- Recupera o nome do corretor, a quantidade de imoveis que ele vendeu e o  
valor total dos imoveis  
SELECT Pessoa.nome AS nomeCorretor, COUNT(*) AS qtdeImoveis, SUM(valor)  
AS valorTotal  
FROM Pessoa NATURAL JOIN Corretor C JOIN Contrato Ct ON C.cpf =  
Ct.Corretor_cpf  
GROUP BY nomeCorretor;
```

```
# 6- Recupera o nome do proprietario do imovel de código de contrato 1  
SELECT nome  
FROM Contrato C NATURAL JOIN Aluga NATURAL JOIN Imovel I JOIN Proprietario  
P JOIN Pessoa  
WHERE C.cod_contrato = 1 AND I.Proprietario_cpf = P.cpf AND P.cpf = Pessoa.cpf;
```

```
# 7- Recupera o nome do corretor e a quantidade de contratos que ele concluiu  
SELECT nome AS nomeCorretor, COUNT(*) AS qtdeContratos  
FROM Pessoa NATURAL JOIN Corretor C JOIN Contrato Ct ON C.cpf =  
Ct.Corretor_cpf  
GROUP BY cpf;
```

```
# 8- Recupera o nome do corretor e as datas de atendimento, da mais recente para  
a mais antiga que atendeu o cliente de cpf "..."  
SELECT P.nome AS NomeCorretor, A.data_atendimento  
FROM Atende A JOIN Pessoa P ON A.Corretor_cpf = P.cpf  
WHERE P.cpf = A.Corretor_cpf AND A.Cliente_cpf = '130979141'  
ORDER BY A.data_atendimento DESC;
```

9 - Recupera o código, o bairro e o valor dos imóveis que são mais caros do que qualquer imóvel do bairro "..."

```
SELECT cod_imovel, bairro, valor
FROM Imovel
WHERE valor > (SELECT MAX(valor)
                FROM Imovel
                WHERE bairro = "Barreiro");
```

10 - Recupera os problemas encontrados e o na vistoria do imóvel de código e o valor para resolve-los "..."

```
SELECT descricao_problema, P.valor
FROM Imovel I NATURAL JOIN Aluga NATURAL JOIN Contrato C JOIN
Problemas_Encontrados P
WHERE C.Vistoria_autoriza_cod_contrato = P.Vistoria_autoriza_cod_contrato AND
I.cod_imovel = 2;
```

11- Recupera o nome e o telefone dos proprietários que possuem um imóvel com aluguel de valor entre 2000 e 3000 reais

```
SELECT nome, telefone
FROM Pessoa NATURAL JOIN Proprietario P JOIN Imovel I
WHERE I.valor BETWEEN 1500 AND 2000 and P.cpf = I.Proprietario_cpf;
```

12- Recupera nome e telefone dos clientes com DDD 321

```
SELECT nome AS nomeCliente, telefone
FROM Pessoa NATURAL JOIN Cliente
WHERE telefone LIKE '035%';
```

13- Recupera o código do imóvel que não teve problemas encontrados na vistoria

```
SELECT cod_imovel
FROM Imovel NATURAL JOIN Aluga A JOIN Contrato c JOIN
Problemas_encontrados PE
WHERE PE.valor = " " AND C.Vistoria_autoriza_cod_contrato =
PE.Vistoria_autoriza_cod_contrato AND A.cod_contrato = C.cod_contrato;
```

G - Exemplos de visões (Views)

-- view que mostra o nome do corretor que está em um contrato x e o cod do imóvel

```
create view Corretor_de_Contrato as
select cod_contrato, Pessoa.nome AS Corretor, cod_imovel
from Corretor natural join Pessoa natural join Contrato natural join aluga
where corretor.cpf = Contrato.Corretor_cpf and aluga.cod_contrato =
Contrato.cod_contrato
group by cod_Contrato, cpf, cod_imovel;
```

-- EXEMPLO DE USO

```
select cod_contrato, nome as Corretor , cod_imovel  
from Corretor_de_Contrato  
where cod_imovel = 6;
```

-- Mostra o nome, do Corretor e a quantidade de clientes que ele ja atendeu ate o momento

```
create view QuantosClientes (NOME_CORRETOR, qntClientes) as  
select Pessoa.nome, count(*)  
from Pessoa natural join Corretor join Atende  
where corretor.cpf = atende.Corretor_cpf  
group by Corretor.cpf;
```

-- MOSTRA OS CORRETORES QUE TEM MAIS DE x CLIENTES

```
select NOME_CORRETOR, qntClientes  
from QuantosClientes  
where qntClientes > 1;
```

-- PROBLEMAS ENCONTRADOS NA VISTORIA DE CADA IMOVEL QUE JA FOI ALUGADO

```
create view ProblemasPorImovel as  
select aluga.cod_imovel, descricao_problema  
from imovel join aluga on imovel.cod_imovel = aluga.cod_imovel join contrato on  
contrato.cod_contrato = aluga.cod_contrato join  
vistoria on vistoria.autoriza_cod_contrato = contrato.cod_contrato join  
problemas_encontrados on  
problemas_encontrados.Vistoria_autoriza_cod_contrato =  
vistoria.autoriza_cod_contrato  
group by cod_imovel, descricao_problema;
```

-- CONTA QUANTOS PROBLEMAS AQUELE IMOVEL JA TEVE

```
select cod_imovel, count(*) as qntProblemas  
from ProblemasPorImovel  
where cod_imovel = 3;
```

H - Exemplos de criação de usuários, concessão e revogação de permissão de acesso


```

CREATE USER 'admin'@'localhost' IDENTIFIED BY '1234';
CREATE USER 'user'@'localhost' IDENTIFIED BY '4567';
CREATE USER 'manutencao'@'localhost' IDENTIFIED BY '7890';

GRANT ALL ON imobiliaria.* TO 'admin'@'localhost';
GRANT SELECT ON imobiliaria.* TO 'user'@'localhost';

GRANT ALL ON imobiliaria.Imovel TO 'manutencao'@'localhost';
GRANT ALL ON imobiliaria.Contrato TO 'manutencao'@'localhost';
GRANT ALL ON imobiliaria.Vistoria TO 'manutencao'@'localhost';
GRANT ALL ON imobiliaria.Pessoa TO 'manutencao'@'localhost';

REVOKE DELETE ON imobiliaria.Imovel FROM 'manutencao'@'localhost';

```

I - Exemplos de procedimentos/funções

-- 1 Formata o CPF

```

DELIMITER $$
CREATE FUNCTION FormataCPF(cpf VARCHAR(11))
RETURNS VARCHAR(14)
DETERMINISTIC
BEGIN
    DECLARE a CHAR(1) DEFAULT '.';
    DECLARE formatado VARCHAR(14);
    SET formatado = CONCAT(SUBSTRING(cpf, 1, 3), a,
                           SUBSTRING(cpf, 4, 3), a,
                           SUBSTRING(cpf, 7, 3), a,
                           SUBSTRING(cpf, 10, 2));

    RETURN formatado;
END$$
DELIMITER ;

-- chama função
SELECT FormataCPF('98745612300'); -- resultado: '987.456.123-00'

```

-- 2 Converte as iniciais da string para maiusculas

```

DELIMITER $$
CREATE FUNCTION Capitalize(str VARCHAR(1024)) -- recebe
string
RETURNS VARCHAR(1024)
DETERMINISTIC
BEGIN

```

```

DECLARE i INT DEFAULT 1;
DECLARE myc, pc CHAR(1);
DECLARE outstr VARCHAR(1024) DEFAULT str;
WHILE i <= CHAR_LENGTH(str) DO
enquanto i for menor que CHAR_LENGTH(str) -> retorna o tamanho da string str
    SET myc = SUBSTRING(str, i, 1);
SUBSTRING(str,i,1) retorna a substring de str começando na posicao i com 1
caracter)
    SET pc = CASE WHEN i = 1 THEN ''
    ELSE SUBSTRING(str, i - 1, 1)
    END;
    IF pc IN (' ', '&', '"', '_', '?', ',', ':', '!', ';', '-', '/', '(', ')') THEN
    SET outstr = INSERT(outstr, i, 1, UPPER(myc)); -- substitui o
caractere da posicao i de outstr por myc convertido para maiuscula
    END IF;
    SET i = i + 1;
END WHILE;
return outstr;
END$$
DELIMITER ;

-- chama função
SET @str = 'thiago odilon'; -- cria variavel
CALL Capitalize(@str); -- chama função
SELECT @str; -- resultado: 'Thiago Odilon'

-- 3 Lista os contratos que estão vencendo

DELIMITER //
CREATE PROCEDURE Vencimento()
BEGIN
    SELECT CASE EXTRACT(MONTH FROM vencimento) -- usa um case
dentro do select, que vai usar uma função extract
        WHEN 01 THEN 'JAN' -- formata
mes transformando em string
        WHEN 02 THEN 'FEV'
        WHEN 03 THEN 'MAR'
        WHEN 04 THEN 'ABR'
        WHEN 05 THEN 'MAI'
        WHEN 06 THEN 'JUN'
        WHEN 07 THEN 'JUL'
        WHEN 08 THEN 'AGO'
        WHEN 09 THEN 'SET'
        WHEN 10 THEN 'OUT'

```

```

        WHEN 11 THEN 'NOV'
        WHEN 12 THEN 'DEZ'
    END AS mes, GROUP_CONCAT(cod_contrato) AS contratos    --
concatena o id dos contratos
FROM Contrato
GROUP BY mes;          -- agrupa funcionarios com o mesmo mês
END //
DELIMITER ;

CALL Vencimento();

```

J - Exemplos de triggers

-- 1 Trigger que impede que algum proprietario seja excluído.

```

DELIMITER $$
CREATE TRIGGER impedeDelete BEFORE DELETE ON Proprietario
FOR EACH ROW BEGIN
    SIGNAL SQLSTATE '16444' SET message_text = 'Não é possível excluir
proprietarios!';
END;
$$
DELIMITER ;

```

-- Exemplo de como disparar o Trigger:

```

DELETE FROM Proprietario
WHERE cpf = '130978444';

```

-- 2 Trigger que verifica o valor do aluguel do imovel, pois acima de 10.000 o imóvel não é aceito

```

DELIMITER $$
CREATE TRIGGER beforeImovelInsert
BEFORE INSERT ON Imovel
FOR EACH ROW
BEGIN
    IF NEW.valor > 10000 THEN /* SIGNAL retorna um erro. Código 45000 é
uma exceção genérica definida pelo usuário. */
        SIGNAL SQLSTATE '69000' SET message_text = 'Não aceitamos
aluguéis acima de R$10.000!';
    END IF;
END $$
DELIMITER ;

```

```
-- Exemplo de como disparar o Trigger:
INSERT INTO Imovel
VALUES      (15,'Rua Tupinambas', 133, 'Casa', 'São Tiago', 'Cerrado',
'36350-000', 12000, '130978444');
```

-- 3 Trigger de auditoria de alterações no valor de Contrato

```
CREATE TABLE auditoriaValor (
idAuditoria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
cod_contrato INT NOT NULL,
valorAnterior DECIMAL(8,2) NOT NULL,
valorNovo    DECIMAL(8,2) NOT NULL,
user VARCHAR(20) NOT NULL,
dataHora datetime NOT NULL
);
```

```
DELIMITER $$
CREATE TRIGGER afterContratoUpdate
AFTER UPDATE ON Contrato
FOR EACH ROW
BEGIN
    IF OLD.valor != NEW.valor THEN
        INSERT INTO auditoriaValor (cod_contrato, valorAnterior, valorNovo,
user, dataHora)
        VALUE (NEW.cod_contrato, OLD.valor, NEW.valor, USER(), NOW());
    END IF;
END $$
DELIMITER ;
```

```
-- Exemplo de como disparar o Trigger:
UPDATE Contrato
SET valor = '5000'
WHERE cod_contrato = '2';
SELECT * FROM auditoriaValor;
```