

# **Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof. Jorge Luiz da Silva

## **TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

### **Projeto de banco de dados para um sistema de Aluguel de imóveis**

Dauane Neves Gerônimo - @dauaneneves

Erik Schneider - @marchingbeagle

Gabriel Willian - @GabrielWDuarte

Gustavo Valsechi - @gustavo-valsechi


Raul Fonseca - @castroderaul23

Renan Henriques - @renanhrqs

Criciúma, 05/12/2023

## URL do projeto no GitHub

<https://github.com/marchingbeagle/projeto-final-bd2-aluguel-imoveis>

 marchingbeagle / projeto-final-bd2-aluguel-imoveis

Code

Issues

4

Pull requests

Actions

Projects

Wiki

Security

Insights

Q

Type to search

>

+

🕒

🔗

📧

👤

projeto-final-bd2-aluguel-imoveis

Public

Watch 2

Fork 0

Star 1

main

1 branch

0 tags

Go to file

Add file

Code

castroderaul23 Update trigger.sql

411188e 24 minutes ago

29 commits

BACK

fix

15 hours ago

SQL

Update trigger.sql

24 minutes ago

README.md

Update README.md

14 hours ago

dicionario\_dados\_locadora\_imovel.xlsx

dicionário\_final

15 hours ago

README.md

📄

Projeto banco de dados: Aluguel de imóveis. 🏠

Integrantes

Erik Schneider - @marchingbeagle

Renan Henriques - @renanhrqs

Gustavo Valsechi - @gustavo-valsechi

Gabriel Duarte - @GabrielWDuarte

Dauane Neves - @dauaneneves

Raul Fonseca - @castroderaul23

Modelo Físico

Utilizamos a ferramenta de modelagem de dados MySQL Workbench para criação do modelo físico do banco de dados, para posterior exportação dos scripts DDL das tabelas e relacionamentos.

About

No description, website, or topics provided.

Readme

Activity

1 star

2 watching

0 forks

Report repository

Releases

No releases published







Create a new release

Packages

No packages published

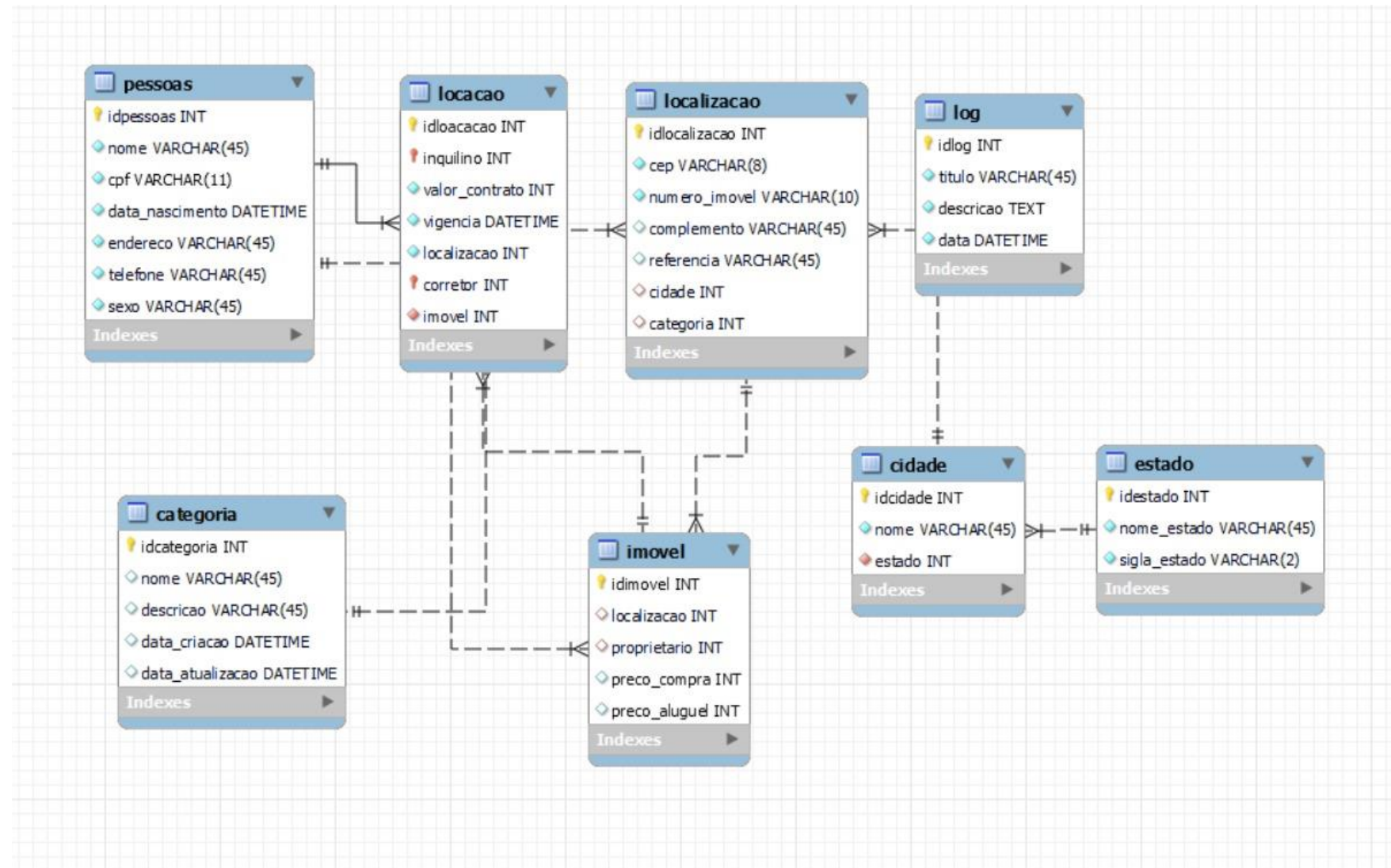
Publish your first package

Contributors 6



Languages

## Modelo ER Físico



## Dicionário de Dados

Tabela	Pessoas					
Descrição	Tabela responsável por armazenar os dados das pessoas inseridas do sistema					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idpessoas	int	1 – sem limite	NOT NULL	X		Código de identificação de pessoas
nome	varchar(45)	5 – 45	NOT NULL			Nome da pessoa
cpf	varchar(11)	11	NOT NULL			CPF pessoa
data_nascimento	datetime		NOT NULL			Data de nascimento da pessoa
endereco	varchar(45)	1 – 45	NOT NULL			endereço da pessoa
telefone	varchar(45)	1 – 45	NOT NULL			telefone da pessoa
sexo	varchar(45)	1 – 45	NOT NULL			sexo da pessoa
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idpessoas	x		x	idpessoas		

Tabela	Locação					
Descrição	Tabela responsável por armazenar os dados de locação criadas pelos corretores					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idlocacao	int	1 – sem limite	NOT NULL	X		Código de locação
inquilino	int	1 – sem limite	NOT NULL	X		Nome do inquilino
valor_contrato	int	1 – sem limite	NOT NULL			Valor do contrato
vigencia	int	1 – sem limite	NOT NULL			Vigência do contrato
localizacao	int	1 – sem limite	NOT NULL			localização do imóvel
corretor	int	1 – sem limite	NOT NULL	X		Qual corretor responsável
imovel	int	2 – sem limite	NOT NULL		X	
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
fk_idpessoas_idx		x		corretor, inquilino		
idlocacao	x		x	idlocacao		
inquilino	x		x	inquilino		

Tabela	Estado					
Descrição	Tabela responsável por armazenar os dados dos estados dos imóveis					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idestado	int	1 – sem limite	NOT NULL	X		Código de identidade do estado
nome_estado	varchar(45)	5 – 45	NOT NULL			Nome do estado
sigla_estado	varchar(2)	2	NOT NULL			Sigla do estado
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idestado	x		x	idestado		

Tabela	Cidade					
Descrição	Tabela responsável por armazenar os dados das cidades dos imóveis					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idcidade	int	1 – sem limite	NOT NULL	X		Código de identidade da cidade
nome	varchar(45)	5 – 45	NOT NULL			Nome da cidade
estado	int		NOT NULL		X	Estado correspondente
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
fk.idestado_idx		x		estado		
idcidade	x		x	idcidade		

Tabela	Categoria					
Descrição	Tabela responsável por armazenar os dados de categorias dos imóvel					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idcategoria	int	1 – sem limite	NOT NULL	X		Código de identidade da categoria
nome	varchar(45)	5 – 45	NULL			Nome da categoria
descricao	varchar(45)	5 – 45	NULL			Descrição dos dados
data_criacao	datetime		NULL			Data de criação
data_atualizacao	datetime		NULL			Data de atualização
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idcategoria	x		x	idcategoria		

Tabela	Localização					
Descrição	Tabela responsável por armazenar os dados da localização do imóvel					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idlocalizacao	int	1 – sem limite	NOT NULL	X		Código de identidade da localização
cep	varchar(8)	1 – 8	NOT NULL			CEP da localização
numero_imovel	varchar(10)	1 – 10	NOT NULL			Número do imóvel
complemento	varchar(45)	1 – 45	NULL			Complemento de localização
referencia	varchar(45)	1 – 45	NULL			Referência da localização
cidade	int	0 – sem limite	NULL		X	Cidade do imóvel
categoria	int	0 – sem limite	NULL		X	Categoria do imóvel
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
fk_idcidade_idx		x		cidade		
fk_idcategoria_idx		x		categoria		
idlocalizacao	x		x	idlocalizacao		

<b>Tabela</b>	Log
---------------	-----

Descrição	Tabela responsável por armazenar os logs					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idlog	int	1 – sem limite	NOT NULL	X		Código de identidade do log
titulo	varchar(45)	10 – 45	NOT NULL			Título do log
descricao	TEXT	0 – sem limite	NOT NULL			Descrição do log
data	datetime		NOT NULL			Data do log
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idlog	x		x	idlog		

Tabela	Imóvel					
Descrição	Tabela responsável por armazenar os dados principais do imóvel					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
idimovel	int	1 – sem limite	NOT NULL	X		Código de identidade do imóvel
localizacao	int	1 – sem limite	NULL		X	Localização do imóvel
proprietario	int	1 – sem limite	NULL		X	Nome do proprietário
preco_compra	int	1 – sem limite	NULL			Preço da compra
preco_aluguel	int	1 – sem limite	NULL			Preço do aluguel
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
fk_idlocalizacao_idx		x		localizacao		
fk_idproprietario_idx		x		proprietario		
idimovel	x		x	idimovel		

## Script dos comandos DDL para criação do Banco de dados

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`pessoas` (
  `idpessoas` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `cpf` VARCHAR(11) NOT NULL,
  `data_nascimento` DATETIME NOT NULL,
  `endereco` VARCHAR(45) NOT NULL,
  `telefone` VARCHAR(45) NOT NULL,
  `sexo` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idpessoas`))
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`log` (
  `idlog` INT NOT NULL AUTO_INCREMENT,
  `titulo` VARCHAR(45) NOT NULL,
  `descricao` TEXT NOT NULL,
```



```
`data` DATETIME NOT NULL,  
PRIMARY KEY (`idlog`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`estado` (  
  `idestado` INT NOT NULL,  
  `nome_estado` VARCHAR(45) NOT NULL,  
  `sigla_estado` VARCHAR(2) NOT NULL,  
  PRIMARY KEY (`idestado`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`cidade` (  
  `idcidade` INT NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `estado` INT NOT NULL,  
  PRIMARY KEY (`idcidade`),  
  INDEX `fk.idestado_idx` (`estado` ASC),  
  CONSTRAINT `fk.idestado`  
    FOREIGN KEY (`estado`)  
    REFERENCES `mydb`.`estado` (`idestado`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`categoria` (  
  `idcategoria` INT NOT NULL,  
  `nome_categoria` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idcategoria`))  
ENGINE = InnoDB;
```

```
`idcategoria` INT NOT NULL,  
`nome` VARCHAR(45) NULL,  
`descricao` VARCHAR(45) NULL,  
`data_criacao` DATETIME NULL,  
`data_atualizacao` DATETIME NULL,  
PRIMARY KEY (`idcategoria`))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`localizacao` (  
  `idlocalizacao` INT NOT NULL,  
  `cep` VARCHAR(8) NOT NULL,  
  `numero_imovel` VARCHAR(10) NOT NULL,  
  `complemento` VARCHAR(45) NULL,  
  `referencia` VARCHAR(45) NULL,  
  `cidade` INT NULL,  
  `categoria` INT NULL,  
  PRIMARY KEY (`idlocalizacao`),  
  INDEX `fk_idcidade_idx` (`cidade` ASC) ,  
  INDEX `fk_idcategoria_idx` (`categoria` ASC),  
  CONSTRAINT `fk_idcidade`  
    FOREIGN KEY (`cidade`)  
    REFERENCES `mydb`.`cidade` (`idcidade`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_idcategoria`
```

```
FOREIGN KEY (`categoria`)  
REFERENCES `mydb`.`categoria` (`idcategoria`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`imovel` (  
  `idimovel` INT NOT NULL,  
  `localizacao` INT NULL,  
  `proprietario` INT NULL,  
  `preco_compra` INT NULL,  
  `preco_aluguel` INT NULL,  
  PRIMARY KEY (`idimovel`),  
  INDEX `fk_idlocalizacao_idx` (`localizacao` ASC),  
  INDEX `fk_idproprietario_idx` (`proprietario` ASC),  
  CONSTRAINT `fk_idlocalizacao`  
    FOREIGN KEY (`localizacao`)  
    REFERENCES `mydb`.`localizacao` (`idlocalizacao`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_idproprietario`  
    FOREIGN KEY (`proprietario`)  
    REFERENCES `mydb`.`pessoas` (`idpessoas`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`locacao` (  
  `idlocacao` INT NOT NULL,  
  `inquilino` INT NOT NULL,  
  `valor_contrato` INT NOT NULL,  
  `vigencia` DATETIME NOT NULL,  
  `localizacao` INT NOT NULL,  
  `corretor` INT NOT NULL,  
  `imovel` INT NOT NULL,  
  PRIMARY KEY (`idlocacao`, `corretor`, `inquilino`),  
  INDEX `fk_idpessoas_idx` (`corretor` ASC, `inquilino` ASC),  
  CONSTRAINT `fk_idpessoas`  
    FOREIGN KEY (`corretor`)  
    REFERENCES `mydb`.`pessoas` (`idpessoas`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_idpessoas_inquilino`  
    FOREIGN KEY (`inquilino`)  
    REFERENCES `mydb`.`pessoas` (`idpessoas`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_idimovel`  
    FOREIGN KEY (`imovel`)  
    REFERENCES `mydb`.`imovel` (`idimovel`)
```

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

## Script que popula as tabelas do Banco de dados

-- INSERT statements for mydb.estado

INSERT INTO mydb.estado (idestado, nome\_estado, sigla\_estado)

VALUES

(1, 'São Paulo', 'SP'),

(2, 'Rio de Janeiro', 'RJ'),

(3, 'Minas Gerais', 'MG'),

(4, 'Bahia', 'BA'),

(5, 'Paraná', 'PR'),

(6, 'Santa Catarina', 'SC'),

(7, 'Rio Grande do Sul', 'RS'),

(8, 'Ceará', 'CE'),

(9, 'Pernambuco', 'PE'),

(10, 'Amazonas', 'AM');

-- INSERT statements for mydb.cidade

INSERT INTO mydb.cidade (idcidade, nome, estado)

VALUES

(1, 'São Paulo', 1),

(2, 'Rio de Janeiro', 2),

(3, 'Belo Horizonte', 3),  
(4, 'Salvador', 4),  
(5, 'Curitiba', 5),  
(6, 'Florianópolis', 6),  
(7, 'Porto Alegre', 7),  
(8, 'Fortaleza', 8),  
(9, 'Recife', 9),  
(10, 'Manaus', 10);

-- INSERT statements for mydb.categoria

INSERT INTO mydb.categoria (idcategoria, nome, descricao, data\_criacao, data\_atualizacao)

VALUES

(1, 'Residencial', 'Casas e apartamentos para moradia', '2023-01-01 00:00:00', '2023-01-01 00:00:00'),  
(2, 'Comercial', 'Imóveis para fins comerciais', '2023-01-02 00:00:00', '2023-01-02 00:00:00'),  
(3, 'Industrial', 'Áreas para indústrias', '2023-01-03 00:00:00', '2023-01-03 00:00:00'),  
(4, 'Rural', 'Terrenos e fazendas', '2023-01-04 00:00:00', '2023-01-04 00:00:00'),  
(5, 'Condomínio', 'Unidades em condomínios fechados', '2023-01-05 00:00:00', '2023-01-05 00:00:00'),  
(6, 'Compartilhado', 'Espaços compartilhados', '2023-01-06 00:00:00', '2023-01-06 00:00:00'),  
(7, 'Turismo', 'Imóveis para aluguel por temporada', '2023-01-07 00:00:00', '2023-01-07 00:00:00'),  
(8, 'Esportivo', 'Áreas para práticas esportivas', '2023-01-08 00:00:00', '2023-01-08 00:00:00'),  
(9, 'Educacional', 'Imóveis para instituições educacionais', '2023-01-09 00:00:00', '2023-01-09 00:00:00'),

```
(10, 'Saúde', 'Imóveis para instituições de saúde', '2023-01-10 00:00:00', '2023-01-10 00:00:00');
```

```
-- INSERT statements for mydb.localizacao
```

```
INSERT INTO mydb.localizacao (idlocalizacao, cep, numero_imovel, complemento, referencia, cidade, categoria)
```

```
VALUES
```

```
(1, '12345678', '101', 'Bloco A', 'Próximo ao shopping', 1, 1),
```

```
(2, '87654321', '202', 'Andar 3', 'Ao lado da praça', 2, 2),
```

```
(3, '54321876', '303', 'Perto da estação', 'Em frente à escola', 3, 3),
```

```
(4, '23456789', '404', 'Casa verde', 'Ao lado do parque', 4, 4),
```

```
(5, '87654321', '505', 'Edifício Sol', 'Próximo à academia', 5, 5),
```

```
(6, '34567890', '606', 'Apartamento Ouro', 'Perto do centro', 6, 6),
```

```
(7, '78901234', '707', 'Casa azul', 'Ao lado do mercado', 7, 7),
```

```
(8, '12345678', '808', 'Bloco B', 'Em frente à padaria', 8, 8),
```

```
(9, '98765432', '909', 'Cobertura Luxo', 'Próximo à praia', 9, 9),
```

```
(10, '54321876', '1010', 'Edifício Diamante', 'Perto do hospital', 10, 10);
```

```
-- INSERT statements for mydb.pessoas
```

```
INSERT INTO mydb.pessoas (idpessoas, nome, cpf, data_nascimento, endereco, telefone, sexo)
```

```
VALUES
```

```
(1, 'João Silva', '12345678901', '1990-05-15 00:00:00', 'Rua A, 123', '123456789', 'M'),
```

```
(2, 'Maria Oliveira', '98765432109', '1985-08-22 00:00:00', 'Avenida B, 456', '987654321', 'F'),
```



```
(3, 'Carlos Santos', '11122233344', '1988-12-10 00:00:00', 'Rua C, 789', '111222333', 'M'),  
(4, 'Ana Pereira', '55566677788', '1995-04-05 00:00:00', 'Avenida D, 1011', '555666777', 'F'),  
(5, 'Pedro Souza', '99988877766', '1980-09-30 00:00:00', 'Rua E, 1213', '999888777', 'M'),  
(6, 'Fernanda Lima', '44433322211', '1992-06-20 00:00:00', 'Avenida F, 1415', '444333222', 'F'),  
(7, 'Ricardo Rocha', '77788899900', '1983-11-25 00:00:00', 'Rua G, 1617', '777888999', 'M'),  
(8, 'Camila Oliveira', '22211144455', '1998-03-15 00:00:00', 'Avenida H, 1819', '222111444', 'F'),  
(9, 'Gabriel Silva', '66655544433', '1987-07-05 00:00:00', 'Rua I, 2021', '666555444', 'M'),  
(10, 'Mariana Santos', '33322211100', '1994-01-10 00:00:00', 'Avenida J, 2223', '333222111', 'F');
```

```
-- INSERT statements for mydb.imovel
```

```
INSERT INTO mydb.imovel (idimovel, localizacao, proprietario, preco_compra, preco_aluguel)
```

```
VALUES
```

```
(1, 1, 1, 300000, 2000),  
(2, 2, 2, 500000, 2500),  
(3, 3, 3, 150000, 1200),  
(4, 4, 4, 700000, 3000),  
(5, 5, 5, 250000, 1800),  
(6, 6, 6, 400000, 2200),  
(7, 7, 7, 180000, 1500),  
(8, 8, 8, 600000, 2800),  
(9, 9, 9, 350000, 2000),  
(10, 10, 10, 450000, 2300),
```

```
(11, 1, 1, 320000, 2100),  
(12, 1, 2, 280000, 1900),  
(13, 2, 3, 550000, 2700),  
(14, 2, 4, 480000, 2300),  
(15, 3, 5, 160000, 1300),  
(16, 3, 6, 140000, 1100),  
(17, 4, 7, 720000, 3200),  
(18, 4, 8, 680000, 3000),  
(19, 5, 9, 260000, 1900),  
(20, 5, 1, 280000, 2000);
```

```
-- INSERT statements for mydb.locacao
```

```
INSERT INTO mydb.locacao (idlocacao, inquilino, valor_contrato, vigencia, localizacao, corretor, imovel)
```

```
VALUES
```

```
(1,1,1500,'2023-01-01 00:00:00',1,2,1),  
(2,2,2000,'2023-02-01 00:00:00',2,3,2),  
(3,3,1200,'2023-03-01 00:00:00',3,4,2),  
(4,4,2500,'2023-04-01 00:00:00',4,5,2),  
(5,5,1800,'2023-05-01 00:00:00',5,6,3),  
(6,6,2200,'2023-06-01 00:00:00',6,7,5),  
(7,7,1500,'2023-07-01 00:00:00',7,8,5),  
(8,8,2800,'2023-08-01 00:00:00',8,9,6),
```

(9,9,2000,'2023-09-01 00:00:00',9,10,7),  
(11,1,2050,'2023-11-01 00:00:00',1,1,8),  
(12,2,1900,'2023-12-01 00:00:00',1,2,9),  
(13,3,2600,'2024-01-01 00:00:00',2,3,10),  
(14,4,2300,'2024-02-01 00:00:00',2,4,11),  
(15,5,1250,'2024-03-01 00:00:00',3,5,12),  
(16,6,1100,'2024-04-01 00:00:00',3,6,19),  
(17,7,3100,'2024-05-01 00:00:00',4,7,20),  
(18,8,2900,'2024-06-01 00:00:00',4,8,11),  
(19,9,1950,'2024-07-01 00:00:00',5,9,13),  
(20,10,2100,'2024-08-01 00:00:00',5,10,2);

## Principais consultas mapeadas baseadas em regras de negócio (mínimo 4)

-- Calcular a média de preços de aluguel por categoria:

```
SELECT
    c.nome AS categoria,
    AVG(i.preco_aluguel) AS media_preco_aluguel
FROM mydb.imovel i
INNER JOIN mydb.localizacao l ON i.localizacao = l.idlocalizacao
INNER JOIN mydb.categoria c ON l.categoria = c.idcategoria
GROUP BY c.nome;
```

-- Listar todos os contratos de locação com vigência expirada:

```
SELECT
    idlocacao,
    inquilino,
    valor_contrato,
    vigencia,
    CURRENT_DATE AS data_atual,
    imovel,
    corretor
FROM mydb.locacao
WHERE vigencia < CURRENT_DATE;
```

-- Listar imóveis disponíveis para aluguel

SELECT

i.idimovel,

l.cep,

l.numero\_imovel,

l.complemento,

l.referencia,

l.cidade,

c.nome AS categoria,

i.proprietario,

i.preco\_aluguel

FROM

imovel i

LEFT JOIN

locacao lo ON i.idimovel = lo.imovel

JOIN

localizacao l ON i.localizacao = l.idlocalizacao

JOIN

categoria c ON l.categoria = c.idcategoria

WHERE

lo.imovel IS NULL;

-- Distribuição dos valores de aluguel para cada categoria de imóvel.

SELECT c.nome AS categoria, AVG(imovel.preco\_aluguel) AS media\_aluguel, MIN(imovel.preco\_aluguel) AS menor\_aluguel, MAX(imovel.preco\_aluguel) AS maior\_aluguel

FROM mydb.imovel

JOIN mydb.localizacao ON imovel.localizacao = localizacao.idlocalizacao

JOIN mydb.categoria c ON localizacao.categoria = c.idcategoria

WHERE imovel.preco\_aluguel IS NOT NULL

GROUP BY c.nome;