|  |  |  |  |
| --- | --- | --- | --- |
| Desenho de personagem de desenhos animados com texto preto sobre fundo branco  Descrição gerada automaticamente com confiança média | Ano Letivo:  20251 | Turno:  Noturno | Local:  A definir |
| Curso:  ENGENHARIA DA COMPUTAÇÃO | | Turma:  3ª Fase |
| Disciplina:  BANCO DE DADOS II | | Professor(a):  JORGE LUIZ DA SILVA | |

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

**Projeto de banco de dados para um sistema de Aluguel de Equipamentos**

Vanessa Ugioni + [vanessaugioni](https://github.com/vanessaugioni)

Karine + [karine285](https://github.com/karine285)

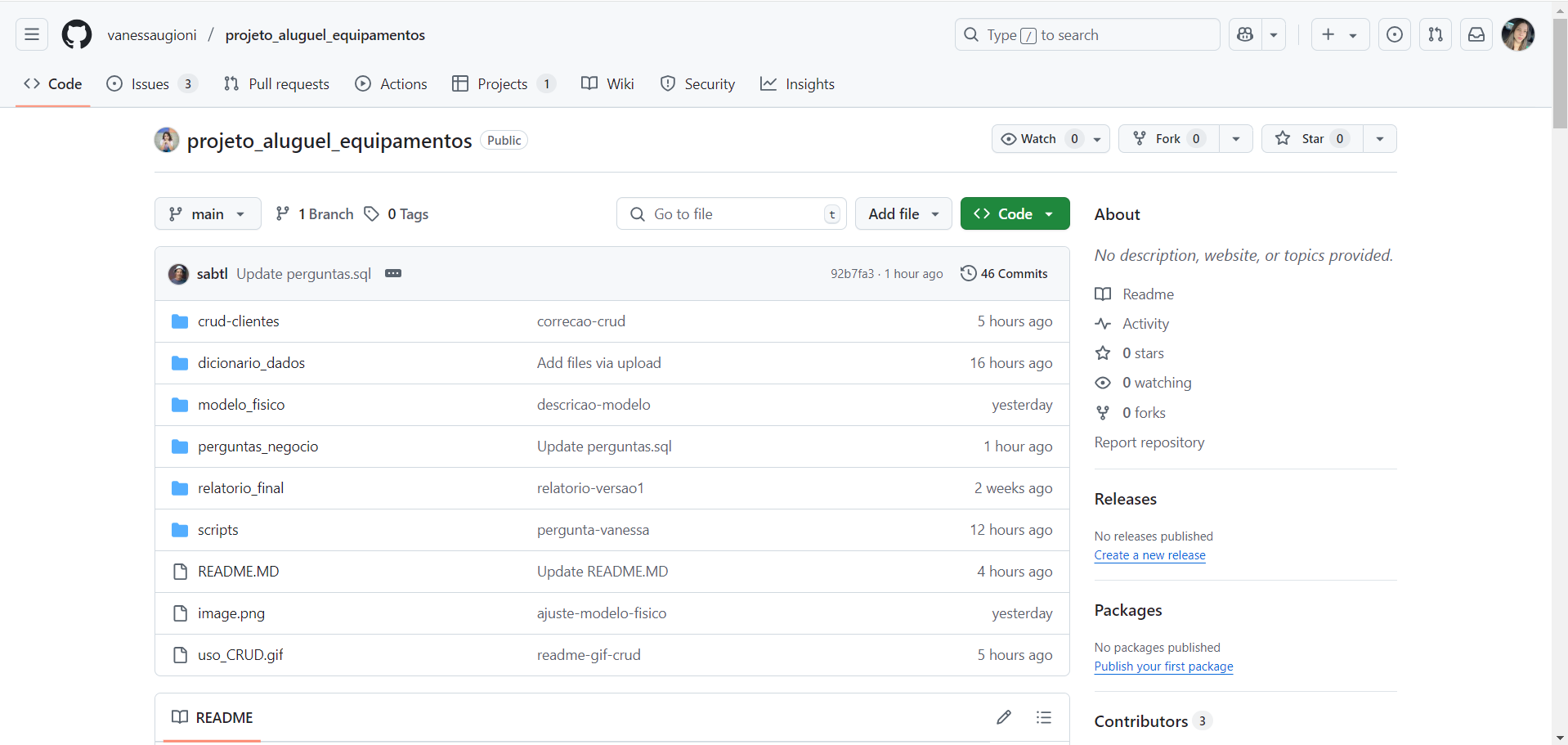
Sabrina + [sabtl](https://github.com/sabtl)

Ana Paula + [anapaulagesser](https://github.com/anapaulagesser)

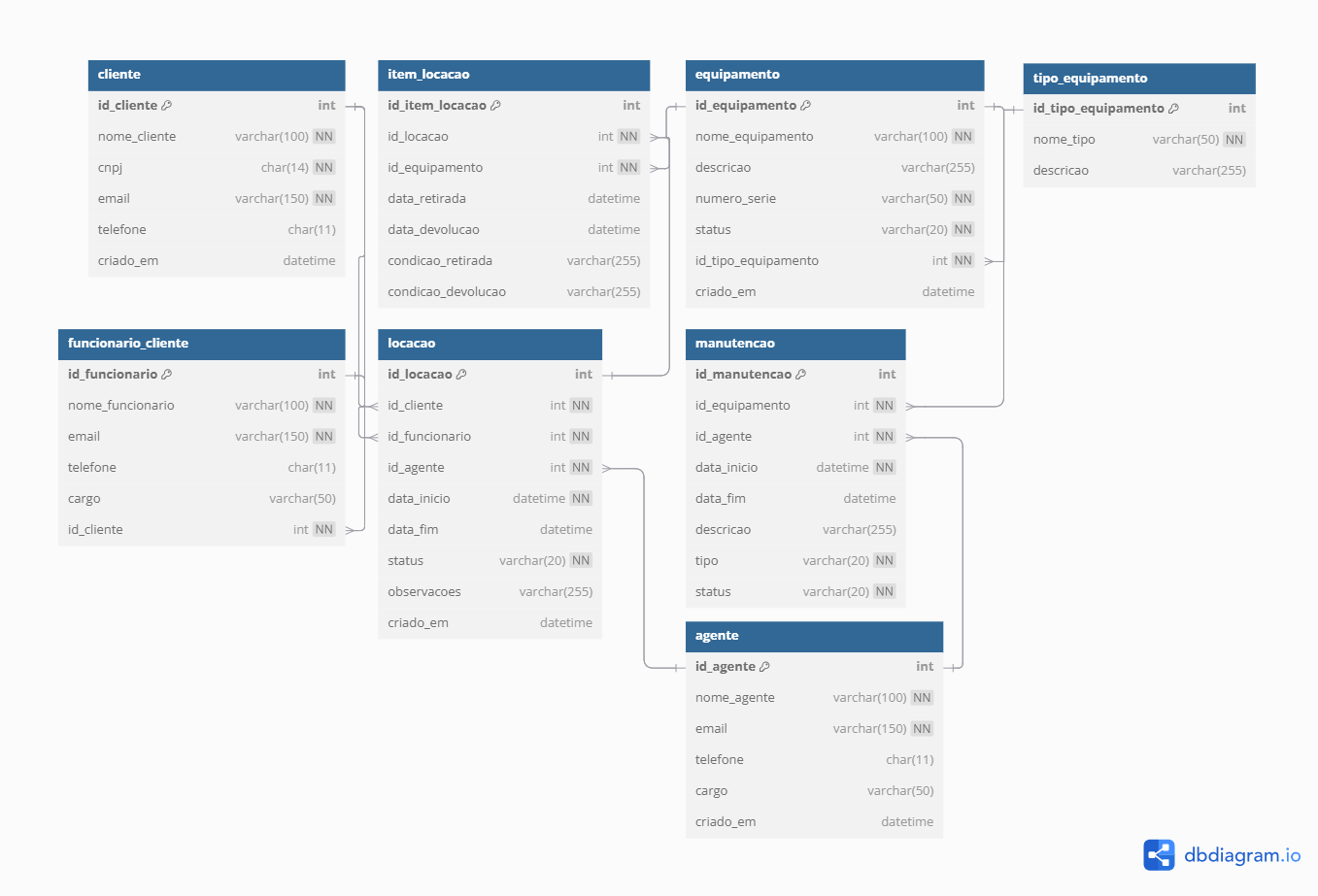
Criciúma, 23/06/2025

**URL do projeto no GitHub**

<https://github.com/vanessaugioni/projeto_aluguel_equipamentos>

****

**Modelo ER Físico**

****

**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Avaliação | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados das avaliações criadas pelos professores | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| cd\_avaliacao | | int | 1 – 2.147.483.647 | NOT NULL | X |  | Código de identificador da avaliação |
| nm\_avaliacao | | varchar(200) | 10 - 200 | NOT NULL |  |  | Nome da avaliação |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_dt\_avaliacao | | X | X | X | cd\_avaliacao, dt\_avaliacao | | |
| Idx\_tipo | |  | X |  | cd\_avaliacao, dt\_avaliacao | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Avaliação | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados das avaliações criadas pelos professores | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| cd\_avaliacao | | int | 1 – 2.147.483.647 | NOT NULL | X |  | Código de identificador da avaliação |
| nm\_avaliacao | | varchar(200) | 10 - 200 | NOT NULL |  |  | Nome da avaliação |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_dt\_avaliacao | | X | X | X | cd\_avaliacao, dt\_avaliacao | | |
| Idx\_tipo | |  | X |  | cd\_avaliacao, dt\_avaliacao | | |

**Script dos comandos DDL para criação do Banco de dados (mínimo 8 tabelas)**

-- ======================================

-- 1. Tabelas

-- ======================================

DROP TABLE IF EXISTS item\_locacao;

DROP TABLE IF EXISTS manutencao;

DROP TABLE IF EXISTS locacao;

DROP TABLE IF EXISTS equipamento;

DROP TABLE IF EXISTS funcionario\_cliente;

DROP TABLE IF EXISTS agente;

DROP TABLE IF EXISTS tipo\_equipamento;

DROP TABLE IF EXISTS cliente;

GO

CREATE TABLE cliente (

id\_cliente INT PRIMARY KEY IDENTITY(1,1),

nome\_cliente VARCHAR(100) NOT NULL,

cnpj CHAR(14) NOT NULL UNIQUE,

email VARCHAR(150) NOT NULL UNIQUE,

telefone CHAR(11),

criado\_em DATETIME

);

GO

CREATE TABLE funcionario\_cliente (

id\_funcionario INT PRIMARY KEY IDENTITY(1,1),

nome\_funcionario VARCHAR(100) NOT NULL,

email VARCHAR(150) NOT NULL UNIQUE,

telefone CHAR(11),

cargo VARCHAR(50),

id\_cliente INT NOT NULL FOREIGN KEY REFERENCES cliente(id\_cliente)

);

GO

CREATE TABLE agente (

id\_agente INT PRIMARY KEY IDENTITY(1,1),

nome\_agente VARCHAR(100) NOT NULL,

email VARCHAR(150) NOT NULL UNIQUE,

telefone CHAR(11),

cargo VARCHAR(50),

criado\_em DATETIME

);

GO

CREATE TABLE tipo\_equipamento (

id\_tipo\_equipamento INT PRIMARY KEY IDENTITY(1,1),

nome\_tipo VARCHAR(50) NOT NULL,

descricao VARCHAR(255)

);

GO

CREATE TABLE equipamento (

id\_equipamento INT PRIMARY KEY IDENTITY(1,1),

nome\_equipamento VARCHAR(100) NOT NULL,

descricao VARCHAR(255),

numero\_serie VARCHAR(50) NOT NULL UNIQUE,

status VARCHAR(20) NOT NULL DEFAULT 'Disponível',

id\_tipo\_equipamento INT NOT NULL FOREIGN KEY REFERENCES tipo\_equipamento(id\_tipo\_equipamento),

criado\_em DATETIME DEFAULT GETDATE(),

CONSTRAINT chk\_status\_equipamento CHECK (status IN ('Disponível', 'Indisponível'))

);

GO

CREATE TABLE locacao (

id\_locacao INT PRIMARY KEY IDENTITY(1,1),

id\_cliente INT NOT NULL FOREIGN KEY REFERENCES cliente(id\_cliente),

id\_funcionario INT NOT NULL FOREIGN KEY REFERENCES funcionario\_cliente(id\_funcionario),

id\_agente INT NOT NULL FOREIGN KEY REFERENCES agente(id\_agente),

data\_inicio DATETIME NOT NULL,

data\_fim DATETIME,

status VARCHAR(20) NOT NULL,

observacoes VARCHAR(255),

criado\_em DATETIME,

CONSTRAINT chk\_status\_locacao CHECK (status IN ('Ativa', 'Finalizada', 'Cancelada'))

);

GO

CREATE TABLE item\_locacao (

id\_item\_locacao INT PRIMARY KEY IDENTITY(1,1),

id\_locacao INT NOT NULL FOREIGN KEY REFERENCES locacao(id\_locacao),

id\_equipamento INT NOT NULL FOREIGN KEY REFERENCES equipamento(id\_equipamento),

data\_retirada DATETIME,

data\_devolucao DATETIME,

condicao\_retirada VARCHAR(255),

condicao\_devolucao VARCHAR(255)

);

GO

CREATE TABLE manutencao (

id\_manutencao INT PRIMARY KEY IDENTITY(1,1),

id\_equipamento INT NOT NULL FOREIGN KEY REFERENCES equipamento(id\_equipamento),

id\_agente INT NOT NULL FOREIGN KEY REFERENCES agente(id\_agente),

data\_inicio DATETIME NOT NULL,

data\_fim DATETIME,

descricao VARCHAR(255),

tipo VARCHAR(20) NOT NULL,

status VARCHAR(20) NOT NULL,

CONSTRAINT chk\_status\_manutencao CHECK (status IN ('Pendente', 'Concluída', 'Cancelada'))

);

GO

-- ======================================

-- 2. Índices (não únicos)

-- ======================================

CREATE NONCLUSTERED INDEX idx\_locacao\_id\_cliente ON locacao(id\_cliente);

CREATE NONCLUSTERED INDEX idx\_locacao\_id\_funcionario ON locacao(id\_funcionario);

CREATE NONCLUSTERED INDEX idx\_locacao\_status\_datafim ON locacao(status, data\_fim);

CREATE NONCLUSTERED INDEX idx\_locacao\_id\_locacao\_status ON locacao(id\_locacao, status);

CREATE NONCLUSTERED INDEX idx\_item\_locacao\_id\_locacao ON item\_locacao(id\_locacao);

CREATE NONCLUSTERED INDEX idx\_item\_locacao\_id\_equipamento ON item\_locacao(id\_equipamento);

CREATE NONCLUSTERED INDEX idx\_item\_locacao\_datas ON item\_locacao(data\_retirada, data\_devolucao);

CREATE NONCLUSTERED INDEX idx\_equipamento\_tipo ON equipamento(id\_tipo\_equipamento);

CREATE NONCLUSTERED INDEX idx\_manutencao\_data\_idequipamento ON manutencao (data\_inicio, id\_equipamento);

CREATE NONCLUSTERED INDEX idx\_manutencao\_status ON manutencao(status);

CREATE NONCLUSTERED INDEX idx\_funcionario\_cliente\_id\_cliente ON funcionario\_cliente(id\_cliente);

CREATE NONCLUSTERED INDEX idx\_itemlocacao\_idequipamento ON item\_locacao (id\_equipamento);

CREATE NONCLUSTERED INDEX idx\_locacao\_idcliente ON locacao (id\_cliente);

-- ======================================

-- 3. Função

-- ======================================

DROP FUNCTION IF EXISTS fn\_total\_locacoes\_por\_cliente\_com\_nome;

GO

CREATE FUNCTION fn\_total\_locacoes\_por\_cliente\_com\_nome(

@id\_cliente INT

)

RETURNS TABLE

AS

RETURN

(

SELECT

c.id\_cliente,

c.nome\_cliente,

COUNT(l.id\_locacao) AS total\_locacoes

FROM cliente c

LEFT JOIN locacao l ON l.id\_cliente = c.id\_cliente

WHERE c.id\_cliente = @id\_cliente

GROUP BY c.id\_cliente, c.nome\_cliente

);

GO

-- ======================================

-- 4. Gatilho

-- ======================================

DROP TRIGGER IF EXISTS trg\_valida\_data\_fim\_locacao;

GO

CREATE TRIGGER trg\_valida\_data\_fim\_locacao

ON locacao

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (

SELECT 1

FROM inserted

WHERE status IN ('Finalizada', 'Cancelada')

AND data\_fim IS NULL

)

BEGIN

THROW 50000, 'Data fim deve estar preenchida para status Finalizada ou Cancelada.', 1;

END;

IF EXISTS (

SELECT 1

FROM inserted

WHERE status = 'Ativa'

AND data\_fim IS NOT NULL

)

BEGIN

THROW 50000, 'Data fim deve ser nula para status Ativa.', 1;

END;

END;

GO

-- ======================================

-- 5. Procedimentos Armazenados

-- ======================================

DROP PROCEDURE IF EXISTS sp\_disponibilidade\_equipamento;

GO

CREATE PROCEDURE sp\_disponibilidade\_equipamento

@id\_equipamento INT,

@data\_inicio DATETIME,

@data\_fim DATETIME

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (

SELECT 1

FROM item\_locacao il

INNER JOIN locacao l ON l.id\_locacao = il.id\_locacao

WHERE il.id\_equipamento = @id\_equipamento

AND l.status = 'ativa'

AND (

(@data\_inicio BETWEEN il.data\_retirada AND ISNULL(il.data\_devolucao, GETDATE())) OR

(@data\_fim BETWEEN il.data\_retirada AND ISNULL(il.data\_devolucao, GETDATE())) OR

(il.data\_retirada BETWEEN @data\_inicio AND @data\_fim)

)

)

BEGIN

PRINT 'Equipamento indisponível no período informado.';

END

ELSE

BEGIN

PRINT 'Equipamento disponível.';

END

END;

GO

**Script que popula as tabelas do Banco de dados (mínimo 10 itens principais)**

-- Populando cliente

SET IDENTITY\_INSERT cliente ON

GO

INSERT cliente (id\_cliente, nome\_cliente, cnpj, email, telefone, criado\_em) VALUES

(1, 'Tech Solutions LTDA', '12345678000101', 'contato@techsolutionsltda.com', '11951939783', GETDATE()),

(2, 'Construtora Ápice', '12345678000102', 'contato@construtoraapice.com', '11978321457', GETDATE()),

(3, 'Hospital São Lucas', '12345678000103', 'contato@hospitalsaolucas.com', '11962349811', GETDATE()),

(4, 'Universidade Pioneira', '12345678000104', 'contato@universidadepioneira.com', '11945632187', GETDATE()),

(5, 'Distribuidora Alfa', '12345678000105', 'contato@distribuidoraalfa.com', '11980976543', GETDATE()),

(6, 'Clínica Vida', '12345678000106', 'contato@clinicavida.com', '11971564892', GETDATE()),

(7, 'Loja Inova', '12345678000107', 'contato@lojainova.com', '11939485721', GETDATE()),

(8, 'Transportadora Rápida', '12345678000108', 'contato@transportadorarapida.com', '11982736459', GETDATE()),

(9, 'Fazenda Bela Vista', '12345678000109', 'contato@fazendabv.com', '11956473829', GETDATE()),

(10, 'Consultoria Prisma', '12345678000110', 'contato@consultoriaprisma.com', '11998653218', GETDATE()),

(11, 'Indústria Vértice', '12345678000111', 'contato@industriavertice.com', '11983512796', GETDATE())

GO

SET IDENTITY\_INSERT cliente OFF

GO

-- Populando funcionario\_cliente

SET IDENTITY\_INSERT funcionario\_cliente ON

GO

INSERT funcionario\_cliente (id\_funcionario, nome\_funcionario, email, telefone, cargo, id\_cliente) VALUES

(1, 'Lucas Silva', 'lucas.s@empresa.com', '11958010319', 'Consultor Comercial', 1),

(2, 'Mariana Almeida', 'mariana.a@empresa.com', '11968274103', 'Gerente de Contas', 2),

(3, 'Carlos Pereira', 'carlos.p@empresa.com', '11997314592', 'Analista de Projetos', 3),

(4, 'Fernanda Costa', 'fernanda.c@empresa.com', '11984927364', 'Suporte Técnico', 4),

(5, 'Ricardo Oliveira', 'ricardo.o@empresa.com', '11975830247', 'Consultor Comercial', 5),

(6, 'Juliana Mendes', 'juliana.m@empresa.com', '11993476281', 'Coordenador de TI', 6),

(7, 'André Souza', 'andre.s@empresa.com', '11964718392', 'Suporte Técnico', 7),

(8, 'Patrícia Lima', 'patricia.l@empresa.com', '11992018473', 'Analista de Projetos', 8),

(9, 'Bruno Rocha', 'bruno.r@empresa.com', '11983620947', 'Consultor Comercial', 9),

(10, 'Camila Martins', 'camila.m@empresa.com', '11977482936', 'Gerente de Contas', 10),

(11, 'Diego Santana', 'diego.s@empresa.com', '11986479123', 'Coordenador de TI', 11)

GO

SET IDENTITY\_INSERT funcionario\_cliente OFF

GO

-- Populando tipo\_equipamento

SET IDENTITY\_INSERT tipo\_equipamento ON

GO

INSERT tipo\_equipamento (id\_tipo\_equipamento, nome\_tipo, descricao) VALUES

(1, 'Notebook', 'Computador portátil para tarefas gerais'),

(2, 'Monitor', 'Tela LCD/LED para computadores'),

(3, 'Impressora', 'Dispositivo para impressão de documentos'),

(4, 'Mouse', 'Periférico apontador'),

(5, 'Teclado', 'Periférico de entrada'),

(6, 'Switch de Rede', 'Equipamento de conexão em redes locais'),

(7, 'Servidor', 'Equipamento de alto desempenho para serviços'),

(8, 'Projetor', 'Dispositivo para projeção de imagem'),

(9, 'Nobreak', 'Fonte de energia ininterrupta'),

(10, 'Webcam', 'Câmera para videoconferência'),

(11, 'Scanner', 'Equipamento para digitalização de documentos')

GO

SET IDENTITY\_INSERT tipo\_equipamento OFF

GO

-- Populando agente

SET IDENTITY\_INSERT agente ON

GO

INSERT agente (id\_agente, nome\_agente, email, telefone, cargo, criado\_em) VALUES

(1, 'Carlos Silva', 'carlos.silva@empresa.com', '11987654321', 'Técnico de Suporte', GETDATE()),

(2, 'Fernanda Oliveira', 'fernanda.oliveira@empresa.com', '11987654322', 'Atendente', GETDATE()),

(3, 'Marcos Pereira', 'marcos.pereira@empresa.com', '11987654323', 'Gerente de TI', GETDATE()),

(4, 'Ana Souza', 'ana.souza@empresa.com', '11987654324', 'Técnico de Rede', GETDATE()),

(5, 'Lucas Almeida', 'lucas.almeida@empresa.com', '11987654325', 'Atendente', GETDATE()),

(6, 'Juliana Costa', 'juliana.costa@empresa.com', '11987654326', 'Gerente de Projetos', GETDATE()),

(7, 'Rafael Gomes', 'rafael.gomes@empresa.com', '11987654327', 'Técnico de Hardware', GETDATE()),

(8, 'Patrícia Lima', 'patricia.lima@empresa.com', '11987654328', 'Atendente', GETDATE()),

(9, 'Thiago Martins', 'thiago.martins@empresa.com', '11987654329', 'Gerente de Suporte', GETDATE()),

(10, 'Beatriz Fernandes', 'beatriz.fernandes@empresa.com', '11987654330', 'Técnico de Software', GETDATE()),

(11, 'Eduardo Rocha', 'eduardo.rocha@empresa.com', '11987654331', 'Atendente', GETDATE())

GO

SET IDENTITY\_INSERT agente OFF

GO

-- Populando equipamento

SET IDENTITY\_INSERT equipamento ON

GO

INSERT equipamento (id\_equipamento, nome\_equipamento, descricao, numero\_serie, status, id\_tipo\_equipamento, criado\_em) VALUES

(1, 'Lenovo ThinkPad X1', 'Notebook empresarial da Lenovo', 'SN0012LEN', 'Disponível', 1, '2025-06-11 10:00:00'),

(2, 'Asus VivoBook 15', 'Notebook Asus para uso geral', 'SN0013ASU', 'Disponível', 1, '2025-06-10 10:00:00'),

(3, 'HP EliteBook 840', 'Notebook de alta performance HP', 'SN0014HP', 'Disponível', 1, '2025-06-09 10:00:00'),

(4, 'MacBook Pro 14"', 'Notebook Apple para desenvolvedores', 'SN0015MAC', 'Disponível', 1, '2025-06-08 10:00:00'),

(5, 'Dell Latitude 7420', 'Notebook corporativo Dell', 'SN0016DEL', 'Disponível', 1, '2025-06-07 10:00:00'),

(6, 'Samsung Book i7', 'Notebook Samsung com bom custo-benefício', 'SN0017SAM', 'Disponível', 1, '2025-06-06 10:00:00'),

(7, 'Acer Aspire 5', 'Notebook Acer para tarefas diárias', 'SN0018ACE', 'Disponível', 1, '2025-06-05 10:00:00'),

(8, 'Positivo Motion Q', 'Notebook básico nacional', 'SN0019POS', 'Disponível', 1, '2025-06-04 10:00:00'),

(9, 'LG Gram 16', 'Notebook ultraleve LG', 'SN0020LG', 'Disponível', 1, '2025-06-03 10:00:00'),

(10, 'Microsoft Surface Laptop', 'Notebook Microsoft premium', 'SN0021MIC', 'Disponível', 1, '2025-06-02 10:00:00'),

(11, 'Avell LIV A52', 'Notebook gamer brasileiro', 'SN0022AVE', 'Disponível', 1, '2025-06-01 10:00:00')

GO

SET IDENTITY\_INSERT equipamento OFF

GO

-- Populando locacao

SET IDENTITY\_INSERT locacao ON

GO

INSERT locacao (id\_locacao, id\_cliente, id\_funcionario, id\_agente, data\_inicio, data\_fim, status, observacoes, criado\_em) VALUES

(1, 1, 1, 1, DATEADD(day, -10, GETDATE()), NULL, 'ativa', 'Locação para projeto temporário', GETDATE()),

(2, 2, 2, 2, DATEADD(day, -15, GETDATE()), DATEADD(day, -5, GETDATE()), 'finalizada', 'Equipamento devolvido em bom estado', GETDATE()),

(3, 3, 3, 3, DATEADD(day, -7, GETDATE()), NULL, 'ativa', 'Locação para evento corporativo', GETDATE()),

(4, 4, 4, 4, DATEADD(day, -30, GETDATE()), DATEADD(day, -1, GETDATE()), 'finalizada', 'Devolução com reparos necessários', GETDATE()),

(5, 5, 5, 5, DATEADD(day, -3, GETDATE()), NULL, 'ativa', 'Locação de equipamentos para home office', GETDATE()),

(6, 6, 6, 6, DATEADD(day, -20, GETDATE()), DATEADD(day, -10, GETDATE()), 'finalizada', 'Equipamento com defeito reportado', GETDATE()),

(7, 7, 7, 7, DATEADD(day, -5, GETDATE()), NULL, 'ativa', 'Locação para teste de software', GETDATE()),

(8, 8, 8, 8, DATEADD(day, -12, GETDATE()), DATEADD(day, -2, GETDATE()), 'finalizada', 'Locação concluída sem problemas', GETDATE()),

(9, 9, 9, 9, DATEADD(day, -1, GETDATE()), NULL, 'ativa', 'Equipamento para suporte remoto', GETDATE()),

(10, 10, 10, 10, DATEADD(day, -8, GETDATE()), NULL, 'ativa', 'Locação para treinamento interno', GETDATE()),

(11, 11, 11, 11, DATEADD(day, -18, GETDATE()), DATEADD(day, -3, GETDATE()), 'finalizada', 'Equipamento devolvido com acessórios', GETDATE())

GO

SET IDENTITY\_INSERT locacao OFF

GO

-- Populando item\_locacao

SET IDENTITY\_INSERT item\_locacao ON

GO

INSERT item\_locacao (id\_item\_locacao, id\_locacao, id\_equipamento, data\_retirada, data\_devolucao, condicao\_retirada, condicao\_devolucao) VALUES

(1, 1, 1, DATEADD(day, -10, GETDATE()), NULL, 'Equipamento em perfeito estado', NULL),

(2, 2, 2, DATEADD(day, -15, GETDATE()), DATEADD(day, -5, GETDATE()), 'Novo', 'Funcionando corretamente'),

(3, 3, 3, DATEADD(day, -7, GETDATE()), NULL, 'Sem avarias', NULL),

(4, 4, 4, DATEADD(day, -30, GETDATE()), DATEADD(day, -1, GETDATE()), 'Leve desgaste', 'Pequenos reparos realizados'),

(5, 5, 5, DATEADD(day, -3, GETDATE()), NULL, 'Perfeito estado', NULL),

(6, 6, 6, DATEADD(day, -20, GETDATE()), DATEADD(day, -10, GETDATE()), 'Problemas no toner', 'Toner substituído'),

(7, 7, 7, DATEADD(day, -5, GETDATE()), NULL, 'Equipamento funcionando', NULL),

(8, 8, 8, DATEADD(day, -12, GETDATE()), DATEADD(day, -2, GETDATE()), 'Sem problemas', 'Devolução sem avarias'),

(9, 9, 9, DATEADD(day, -1, GETDATE()), NULL, 'Equipamento usado', NULL),

(10, 10, 10, DATEADD(day, -8, GETDATE()), NULL, 'Equipamento com acessórios completos', NULL),

(11, 11, 11, DATEADD(day, -18, GETDATE()), DATEADD(day, -3, GETDATE()), 'Bom estado geral', 'Devolvido com pequenas marcas')

GO

SET IDENTITY\_INSERT item\_locacao OFF

GO

-- Populando manutencao

SET IDENTITY\_INSERT manutencao ON

GO

INSERT manutencao (id\_manutencao, id\_equipamento, id\_agente, data\_inicio, data\_fim, descricao, tipo, status) VALUES

(1, 1, 2, '2025-05-30 10:00:00', NULL, 'Troca de teclado', 'Corretiva', 'Pendente'),

(2, 2, 3, '2025-05-28 10:00:00', NULL, 'Atualização de sistema', 'Corretiva', 'Pendente'),

(3, 3, 4, '2025-05-26 10:00:00', NULL, 'Substituição de HD', 'Corretiva', 'Concluída'),

(4, 4, 5, '2025-05-24 10:00:00', NULL, 'Manutenção em porta USB', 'Corretiva', 'Pendente'),

(5, 5, 6, '2025-05-22 10:00:00', NULL, 'Reinstalação do Windows', 'Corretiva', 'Pendente'),

(6, 6, 7, '2025-05-20 10:00:00', NULL, 'Substituição de bateria', 'Corretiva', 'Pendente'),

(7, 7, 8, '2025-05-18 10:00:00', NULL, 'Manutenção de tela', 'Corretiva', 'Concluída'),

(8, 8, 9, '2025-05-16 10:00:00', NULL, 'Limpeza de cooler', 'Corretiva', 'Pendente'),

(9, 9, 10, '2025-05-14 10:00:00', NULL, 'Upgrade de memória RAM', 'Corretiva', 'Pendente'),

(10, 10, 11, '2025-05-12 10:00:00', NULL, 'Troca de pasta térmica', 'Corretiva', 'Pendente'),

(11, 11, 1, '2025-05-10 10:00:00', NULL, 'Verificação de desempenho', 'Corretiva', 'Concluída')

GO

SET IDENTITY\_INSERT manutencao OFF

GO

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 6)**

--13. (Aberta) Utilizando a linguagem SQL, execute consultas que possam responder a seguinte questão: Qual(is) Carro(s) possui(em) mais sinistros cadastrados?

select s.placa, COUNT(\*)

from sinistro s inner join carro c

on s.placa = c.placa

group by s.placa

order by COUNT(\*) desc

--14. (Aberta) Quantos clientes cadastrados possuem letra “a” no meio e não no final?

select \* --count(\*)

from cliente

where nome like '%a%'

and nome not like '%a'

--15. (Aberta) Quantos clientes não possuem apólice?

select nome, cod\_apolice --\* --count(\*)

from cliente c left join apolice a

on c.cod\_cliente = a.cod\_cliente

where cod\_apolice is null

select nome, cod\_apolice

from cliente c inner join apolice a on c.cod\_cliente = a.cod\_cliente

--16. (Aberta) Quais carros não possuem apólice?

select \* --count(\*)

from carro c left join apolice a on c.placa = a.placa

where cod\_apolice is null

select \*

from apolice

where placa not in (select placa from carro)