	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 SEGURO PARA CELULARES

Ana Julia Vieira Lidorio — anajulidorio

Ana Laura Vicenzi Dordete — anaavicenzi

João Gustavo Rodovanski Vitali — gutinjao22

Lara da Rosa Dondossola — laradondossola

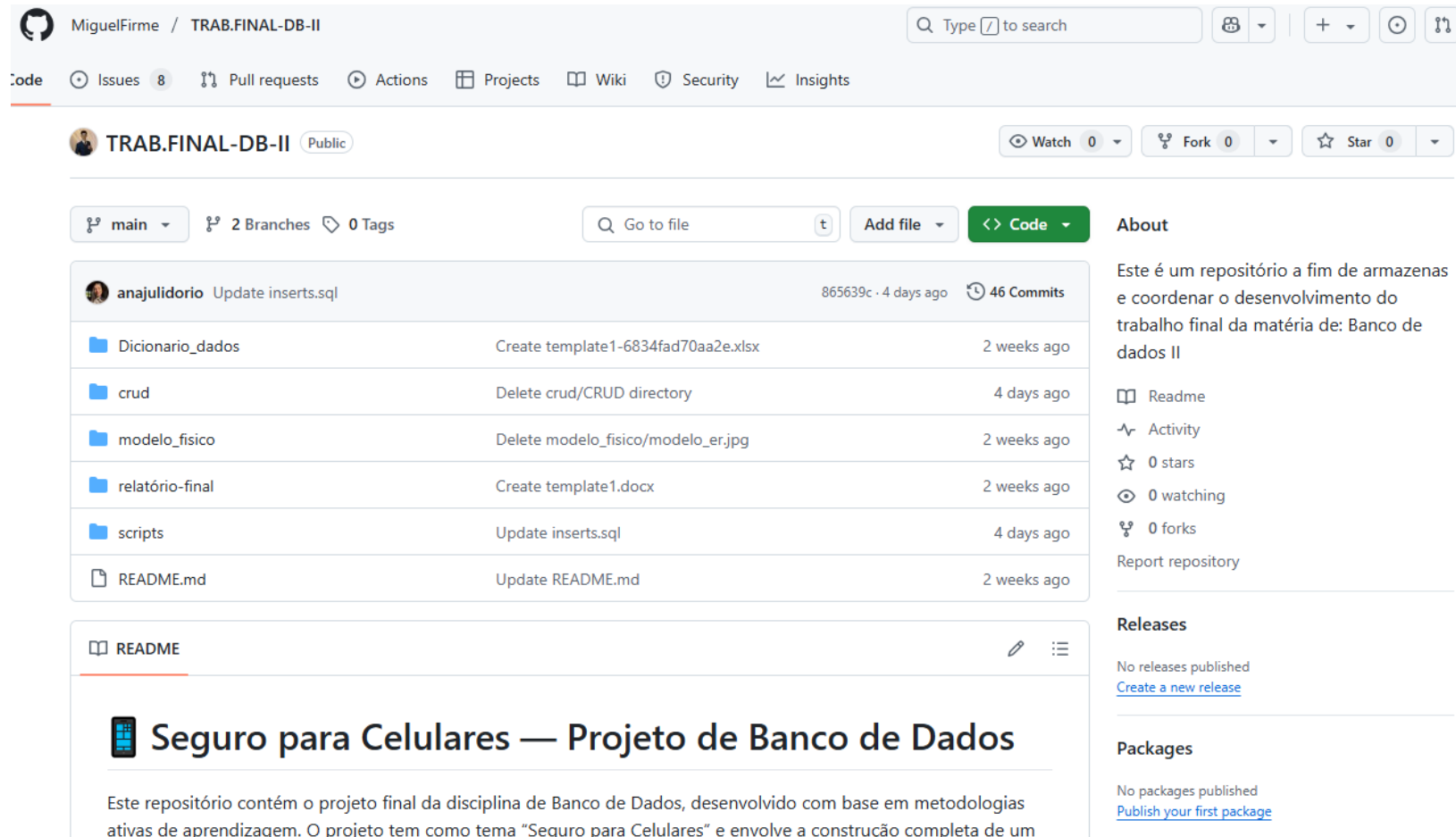
Lorenzo Viero Sartori — loren1z9o

Miguel Antônio Gregório Firme — miguelfirme

Criciúma, 26/05/2025

## URL do projeto no GitHub

<https://github.com/MiguelFirme/TRAB.FINAL-DB-II>



MiguelFirme / TRAB.FINAL-DB-II

Type  to search

code Issues 8 Pull requests Actions Projects Wiki Security Insights

TRAB.FINAL-DB-II Public

Watch 0 Fork 0 Star 0

main 2 Branches 0 Tags

Go to file Add file Code

anajulidorio	Update inserts.sql	865639c · 4 days ago	46 Commits
Dicionario_dados	Create template1-6834fad70aa2e.xlsx	2 weeks ago	
crud	Delete crud/CRUD directory	4 days ago	
modelo_fisico	Delete modelo_fisico/modelo_er.jpg	2 weeks ago	
relatório-final	Create template1.docx	2 weeks ago	
scripts	Update inserts.sql	4 days ago	
README.md	Update README.md	2 weeks ago	

README

### Seguro para Celulares — Projeto de Banco de Dados

Este repositório contém o projeto final da disciplina de Banco de Dados, desenvolvido com base em metodologias ativas de aprendizagem. O projeto tem como tema "Seguro para Celulares" e envolve a construção completa de um

About

Este é um repositório a fim de armazenar e coordenar o desenvolvimento do trabalho final da matéria de: Banco de dados II

Readme Activity 0 stars 0 watching 0 forks Report repository

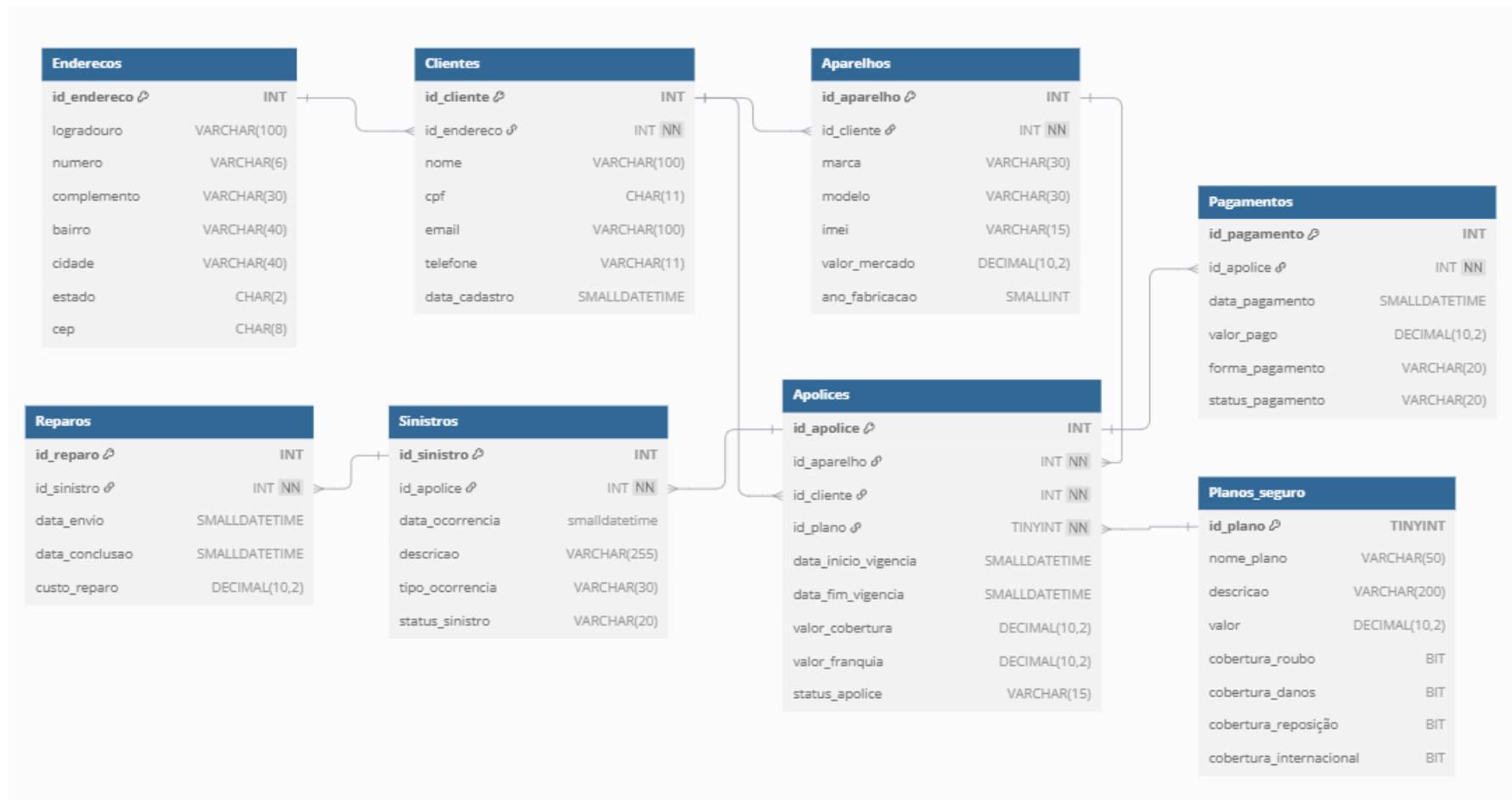
Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

## Modelo ER Físico



## Dicionário de Dados

Dicionário de Dados						
Tabela	Endereços					
Descrição	Tabela que armazena a localização					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_endereco	INT	(-2,147,483,648) até (2,147,483,647)		X		Código para o endereço
logradouro	VARCHAR	1–100	NOT NULL			Armazena a rua, avenida ou circulação urbana
numero	VARCHAR	1–6	NOT NULL			Armazena o número do endereço
complemento	VARCHAR	1–30	NOT NULL			informação extra para localização do endereço
bairro	VARCHAR	1–40	NOT NULL			Região onde o endereço está localizado
cidade	VARCHAR	1–40	NOT NULL			cidade onde o endereço se encontra
estado	CHAR	1–2	NOT NULL			Estado onde é localizado o endereço
cep	CHAR	1–8	NOT NULL			Código de Endereçamento Postal
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		

Tabela	Clientes					
Descrição	Tabela que armazena o cadastro do cliente					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_cliente	INT	(-2,147,483,648) até (2,147,483,647)		X		codigo único do cliente
id_endereco	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único do endereço
nome	VARCHAR	1-100	NOT NULL			nome do cliente
cpf	CHAR	1-11	NOT NULL, UNIQUE			cadastro de pessoa física
email	VARCHAR	1-100	NOT NULL			e-mail utilizado pelo cliente
telefone	VARCHAR	1-11	NOT NULL			guarda o telefone para contato do cliente
data_cadastro	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			Armazena a data a qual o cadastro foi feito
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_clientes_id		X	X	id_cliente		
idx_clientes_id_endereco		X		id_endereco		

Tabela	Aparelhos					
Descrição	Armazena os dados do aparelho do cliente					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_aparelho	INT	(-2,147,483,648) até (2,147,483,647)		X		Código único para o aparelho
id_cliente	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	Código único para o cliente
marca	VARCHAR	1--30	NOT NULL			marca do celular
modelo	VARCHAR	1--50	NOT NULL			modelo do aparelho telefonico
imei	VARCHAR	1--15	NOT NULL, UNIQUE			número único de identificação de aparelhos celulares.
ano_fabricacao	SMALLINT	(-32,768) até (32,767)	NOT NULL			ano o qual foi feito o celular
valor_mercado	DECIMAL	10,2	NOT NULL			valor de mercado atual do aparelho
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_aparelhos_cliente		X		id_cliente		
idx_aparelhos_modelo		X		modelo		

Tabela	Apolices					
Descrição	Armazena as informações importantes referentes ao contrato do seguro					
Atributos						
Nome da Coluna	Tipo do Dado	Valor mín e max	Nulidade	PK	FK	Descrição
id_apolice	INT	(-2,147,483,648) até (2,147,483,647)		X		código único para a apolice
id_aparelho	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único para o aparelho
id_cliente	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único para o cliente
id_plano	TINYINT(NN)	0 a 255	NOT NULL		X	código único para o plano
data_inicio_vigencia	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			data do início da análise do problema
data_fim_vigencia	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			data a qual a cobertura inicia
valor_cobertura	DECIMAL	10,2	NOT NULL			data a qual a cobertura finaliza
valor_franquia	DECIMAL	10,2	NOT NULL			valor da franquia do celular
status_apolice	VARCHAR	1--15	NOT NULL			situação do aparelho
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_apolices_aparelho		X		id_aparelho		
idx_apolices_status		X		status_apolice		

Tabela	Plano_seguro					
Descrição	Armazena as informações do plano de seguro					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_plano	TINYINT	0 a 255		X		código único do aparelho
nome_plano	VARCHAR	1–50	NOT NULL			nome do plano de seguro do celular
descricao	VARCHAR	1–200	NOT NULL			descrição do plano de seguro
valor_mensal	DECIMAL	10,2	NOT NULL			valor mensal do plano
cobertura_roubo	BIT	0 - 1	NOT NULL			cobertura em que o plano oferece
cobertura_danos	BIT	0 - 1	NOT NULL			cobertura em danos a qual o plano oferece
cobertura_perda	BIT	0 - 1	NOT NULL			cobertura de perda do celular que o plano disponibiliza
franquia	DECIMAL	10,2	NOT NULL			quantidade de dados que você pode usar no plano sem custos adicionais.
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_planos_id		X		id_plano		
idx_planos_nome_id		X				

Tabela	Sinistros					
Descrição	Armazena os dados de sinistros ocorridos nos aparelhos					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_sinistro	INT	(-2,147,483,648) até (2,147,483,647)		X		código único para o sinistro
id_apolice	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único para apólice
data_ocorrencia	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			data a qual foi solicitado o seguro
tipo_ocorrencia	VARCHAR	1--30	NOT NULL			sinistro relatado pelo cliente
descricao	VARCHAR	1--100	NOT NULL			situação descrita pelo cliente
status_sinistro	VARCHAR	1--20	NOT NULL			situação atual do sinistro
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_sinistros_apolice		X		id_apolice		
idx_sinistros_data		X		data_ocorrencia		

Tabela	Reparos					
Descrição	Armazena os dados sobre os reparos feitos em aparelhos celulares					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_reparo	INT	(-2,147,483,648) até (2,147,483,647)		X		código único de reparo
id_sinistro	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único de sinistro
data_envio	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			dia o qual o aparelho foi enviado
data_conclusao	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			dia o qual o reparo foi concluído
custo_reparo	DECIMAL	10,2	NOT NULL			valor o qual foi cobrado no reparo
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_reparos_sinistro		X		id_sinistro		

Tabela	Pagamentos					
Descrição	Armazena os pagamentos dos clientes					
			Atributos			
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
id_pagamento	INT	(-2,147,483,648) até (2,147,483,647)		X		código único de pagamento
id_apolice	INT(NN)	(-2,147,483,648) até (2,147,483,647)	NOT NULL		X	código único da apólice
data_pagamento	SMALLDATETIME	1900-01-01 até 2079-06-06	NOT NULL			data o qual fora realizado o pagamento
valor_pago	DECIMAL	10,2	NOT NULL			valor que foi pago no serviço
forma_pagamento	VARCHAR	1-20	NOT NULL			maneira o qual foi realizado o pagamento
status_pagamento	VARCHAR	1-20	NOT NULL			situação atual do pagamento
			Índice			
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_pagamentos_apolice		X		id_apolice		
idx_pagamentos_status		X		status_pagamento		

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

```
CREATE TABLE Enderecos (  
    id_endereco INT IDENTITY PRIMARY KEY,  
    logradouro VARCHAR(100) NOT NULL,  
    numero VARCHAR(6) NULL,  
    complemento VARCHAR(30) NULL,  
    bairro VARCHAR(40) NOT NULL,  
    cidade VARCHAR(40) NOT NULL,  
    estado CHAR(2) NOT NULL,  
    cep CHAR(8) NOT NULL  
);
```

```
CREATE TABLE Clientes (  
    id_cliente INT IDENTITY PRIMARY KEY,  
    id_endereco INT NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    cpf CHAR(11) NOT NULL UNIQUE,  
    email VARCHAR(100) NOT NULL,  
    telefone VARCHAR(11) NOT NULL,  
    data_cadastro SMALLDATETIME NOT NULL,  
  
    CONSTRAINT fk_clientes_endereco FOREIGN KEY (id_endereco)  
        REFERENCES Enderecos (id_endereco)  
);
```



```
CREATE TABLE Aparelhos (  
    id_aparelho INT IDENTITY PRIMARY KEY,  
    id_cliente INT NOT NULL,  
    marca VARCHAR(30) NOT NULL,  
    modelo VARCHAR(30) NOT NULL,  
    imei VARCHAR(15) NOT NULL UNIQUE,  
    valor_mercado DECIMAL(10,2) NOT NULL,  
    ano_fabricacao SMALLINT NULL,  
  
    CONSTRAINT fk_aparelho_cliente FOREIGN KEY (id_cliente)  
        REFERENCES Clientes (id_cliente)  
);
```

```
CREATE TABLE Planos_Seguro (  
    id_plano TINYINT IDENTITY PRIMARY KEY,  
    nome_plano VARCHAR(50) NOT NULL,  
    descricao VARCHAR(200) NOT NULL,  
    valor DECIMAL(10,2) NOT NULL,  
    cobertura_roubo BIT NOT NULL,  
    cobertura_danos BIT NOT NULL,  
    cobertura_reposicao BIT NOT NULL,  
    cobertura_internacional BIT NOT NULL  
);
```

```
CREATE TABLE Apolices (  
    id_apolice INT IDENTITY PRIMARY KEY,  
    id_aparelho INT NOT NULL,  
    id_cliente INT NOT NULL,  
    id_plano TINYINT NOT NULL,  
    data_inicio_vigencia SMALLDATETIME NOT NULL,  
    data_fim_vigencia SMALLDATETIME NOT NULL,  
    valor_cobertura DECIMAL(10,2) NOT NULL,  
    valor_franquia DECIMAL(10,2) NOT NULL,  
    status_apolice VARCHAR(15) NOT NULL,  
  
    CONSTRAINT fk_apolice_aparelho FOREIGN KEY (id_aparelho)  
        REFERENCES Aparelhos (id_aparelho),  
  
    CONSTRAINT fk_apolice_cliente FOREIGN KEY (id_cliente)  
        REFERENCES Clientes (id_cliente),  
  
    CONSTRAINT fk_apolice_plano FOREIGN KEY (id_plano)  
        REFERENCES Planos_Seguro (id_plano)  
);
```

```
CREATE TABLE Sinistros (  
    id_sinistro INT IDENTITY PRIMARY KEY,  
    id_apolice INT NOT NULL,  
    data_ocorrencia SMALLDATETIME NOT NULL,
```

```
descricao VARCHAR(255) NOT NULL,  
tipo_ocorrencia VARCHAR(30) NOT NULL,  
status_sinistro VARCHAR(20) NOT NULL,  
  
CONSTRAINT fk_sinistro_apolice FOREIGN KEY (id_apolice)  
REFERENCES Apolices (id_apolice)  
);
```

```
CREATE TABLE Reparos (  
    id_reparo INT IDENTITY PRIMARY KEY,  
    id_sinistro INT NOT NULL,  
    data_envio SMALLDATETIME NOT NULL,  
    data_conclusao SMALLDATETIME NOT NULL,  
    custo_reparo DECIMAL(10,2) NOT NULL,  
  
    CONSTRAINT fk_reparo_sinistro FOREIGN KEY (id_sinistro)  
    REFERENCES Sinistros (id_sinistro)  
);
```

```
CREATE TABLE Pagamentos (  
    id_pagamento INT IDENTITY PRIMARY KEY,  
    id_apolice INT NOT NULL,  
    data_pagamento SMALLDATETIME NOT NULL,  
    valor_pago DECIMAL(10,2) NOT NULL,  
    forma_pagamento VARCHAR(20) NOT NULL,
```

```
status_pagamento VARCHAR(20) NOT NULL,
```

```
CONSTRAINT fk_pagamento_apolice FOREIGN KEY (id_apolice)
```

```
REFERENCES Apolices (id_apolice)
```

```
);
```

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

```
INSERT INTO Enderecos (logradouro, numero, complemento, bairro, cidade, estado, cep) VALUES
('Rua das Flores', '100', 'Apto 1', 'Centro', 'Curitiba', 'PR', '80010000'),
('Av. Paulista', '2000', NULL, 'Bela Vista', 'São Paulo', 'SP', '01310000'),
('Rua da Aurora', '300', NULL, 'Boa Vista', 'Recife', 'PE', '50050000'),
('Rua Goiás', '400', 'Casa', 'Savassi', 'Belo Horizonte', 'MG', '30140000'),
('Av. Sete de Setembro', '500', NULL, 'Campo Grande', 'Salvador', 'BA', '40060000'),
('Rua João Pessoa', '600', NULL, 'Centro', 'Porto Alegre', 'RS', '90010000'),
('Rua 24 de Outubro', '700', 'Fundos', 'Vila Isabel', 'Rio de Janeiro', 'RJ', '20560000'),
('Rua das Acácias', '800', NULL, 'Meireles', 'Fortaleza', 'CE', '60160000'),
('Rua Afonso Pena', '900', NULL, 'Centro', 'Campo Grande', 'MS', '79002000'),
('Rua Pará', '1100', NULL, 'Marco', 'Belém', 'PA', '66093000'),
('Av. Frei Serafim', '1200', 'Sala 202', 'Centro', 'Teresina', 'PI', '64001020'),
('Rua XV de Novembro', '1500', 'Fundos', 'Centro', 'Florianópolis', 'SC', '88010000');
```

```
INSERT INTO Clientes (id_endereco, nome, cpf, email, telefone, data_cadastro) VALUES
(1, 'João Silva', '12345678901', 'joao.silva@gmail.com', '41999990001', GETDATE()),
(2, 'Maria Souza', '23456789012', 'maria.souza@gmail.com', '11999990002', GETDATE()),
(3, 'Pedro Santos', '34567890123', 'pedro.santos@gmail.com', '81999990003', GETDATE()),
(4, 'Ana Costa', '45678901234', 'ana.costa@gmail.com', '31999990004', GETDATE()),
(5, 'Lucas Lima', '56789012345', 'lucas.lima@gmail.com', '71999990005', GETDATE()),
(6, 'Carla Rocha', '67890123456', 'carla.rocha@gmail.com', '51999990006', GETDATE()),
(7, 'Paulo Mendes', '78901234567', 'paulo.mendes@gmail.com', '21999990007', GETDATE()),
(8, 'Juliana Alves', '89012345678', 'juliana.alves@gmail.com', '85999990008', GETDATE()),
```

```
(9, 'Fernando Pires', '90123456789', 'fernando.pires@gmail.com', '67999990009', GETDATE()),
(10, 'Rafael Cardoso', '11223344556', 'rafael.cardoso@gmail.com', '91999990011', GETDATE()),
(11, 'Camila Nascimento', '22334455667', 'camila.nascimento@gmail.com', '86999990012', GETDATE()),
(12, 'Marcos Pereira', '55667788990', 'marcos.pereira@gmail.com', '48999990015', GETDATE());
```

```
INSERT INTO Aparelhos (id_cliente, marca, modelo, imei, valor_mercado, ano_fabricacao) VALUES
```

```
(1, 'Samsung', 'Galaxy S21', '359881234567890', 3500.00, 2021),
(2, 'Apple', 'iPhone 13', '353456789012345', 5000.00, 2021),
(3, 'Motorola', 'Edge 20', '351234567890123', 2500.00, 2022),
(4, 'Xiaomi', 'Mi 11', '354567890123456', 3000.00, 2021),
(5, 'LG', 'Velvet', '356789012345678', 2200.00, 2020),
(6, 'Asus', 'ROG Phone 5', '358901234567890', 4000.00, 2021),
(7, 'OnePlus', '9 Pro', '352345678901234', 4200.00, 2021),
(8, 'Realme', 'GT Master', '355678901234567', 2000.00, 2022),
(9, 'Nokia', '8.3', '357890123456789', 1800.00, 2020),
(10, 'Sony', 'Xperia 5', '359012345678901', 3500.00, 2021),
(11, 'Samsung', 'Galaxy A54', '351234567891011', 1800.00, 2023),
(12, 'Apple', 'iPhone 12', '353456789013012', 4000.00, 2020);
```

```
INSERT INTO Planos_Seguro (nome_plano, descricao, valor, cobertura_roubo, cobertura_danos, cobertura_reposicao, cobertura_internacional)
```

```
VALUES
```

```
('Bronze', 'Quebra acidental e roubo/furto mediante arrombamento', 29.90, 1, 1, 0, 0),
('Prata', 'Bronze + furto simples, danos elétricos e bloqueio remoto', 49.90, 1, 1, 0, 0),
('Ouro', 'Prata + reposição emergencial', 69.90, 1, 1, 1, 0),
('Diamante', 'Ouro + cobertura internacional', 89.90, 1, 1, 1, 1);
```

```
INSERT INTO Apolices (id_aparelho, id_cliente, id_plano, data_inicio_vigencia, data_fim_vigencia, valor_cobertura, valor_franquia, status_apolice) VALUES
```

```
(1, 1, 1, GETDATE(), DATEADD(year, 1, GETDATE()), 3000.00, 300.00, 'Ativa'),  
(2, 2, 2, GETDATE(), DATEADD(year, 1, GETDATE()), 4500.00, 400.00, 'Ativa'),  
(3, 3, 3, GETDATE(), DATEADD(year, 1, GETDATE()), 2500.00, 250.00, 'Ativa'),  
(4, 4, 4, GETDATE(), DATEADD(year, 1, GETDATE()), 2800.00, 280.00, 'Ativa'),  
(5, 5, 1, GETDATE(), DATEADD(year, 1, GETDATE()), 2000.00, 200.00, 'Ativa'),  
(6, 6, 2, GETDATE(), DATEADD(year, 1, GETDATE()), 3500.00, 300.00, 'Ativa'),  
(7, 7, 3, GETDATE(), DATEADD(year, 1, GETDATE()), 4000.00, 400.00, 'Ativa'),  
(8, 8, 4, GETDATE(), DATEADD(year, 1, GETDATE()), 2000.00, 200.00, 'Ativa'),  
(9, 9, 3, GETDATE(), DATEADD(year, 1, GETDATE()), 1800.00, 180.00, 'Ativa'),  
(10, 10, 2, GETDATE(), DATEADD(year, 1, GETDATE()), 3500.00, 350.00, 'Ativa'),  
(11, 11, 2, GETDATE(), DATEADD(YEAR, 1, GETDATE()), 1600.00, 160.00, 'Ativa'),  
(12, 12, 3, GETDATE(), DATEADD(YEAR, 1, GETDATE()), 3500.00, 350.00, 'Ativa');
```

```
INSERT INTO Sinistros (id_apolice, data_ocorrencia, descricao, tipo_ocorrencia, status_sinistro) VALUES
```

```
(1, '2024-01-15', 'Roubo em via pública', 'Roubo', 'Em análise'),  
(2, '2025-02-20', 'Queda do aparelho', 'Dano', 'Concluído'),  
(3, '2025-03-05', 'Perda do aparelho', 'Perda', 'Em análise'),  
(4, '2024-02-27', 'Roubo na residência', 'Roubo', 'Concluído'),  
(4, '2025-03-08', 'Perda em viagem internacional', 'Perda', 'Em análise'),  
(4, '2023-12-15', 'Dano por água', 'Dano', 'Concluído'),  
(5, '2025-04-10', 'Dano na tela', 'Dano', 'Em reparo'),  
(5, '2024-06-25', 'Roubo na saída da escola', 'Roubo', 'Em análise'),
```

```

(6, '2024-04-20', 'Perda em viagem', 'Perda', 'Negado'),
(7, '2025-04-25', 'Roubo em transporte público', 'Roubo', 'Em análise'),
(8, '2023-05-01', 'Dano por água', 'Dano', 'Em análise'),
(8, '2024-10-12', 'Roubo no ônibus', 'Roubo', 'Concluído'),
(9, '2025-05-10', 'Perda no shopping', 'Perda', 'Em análise'),
(10, '2023-05-15', 'Roubo na saída do trabalho', 'Roubo', 'Concluído'),
(2, '2025-04-15', 'Queda novamente', 'Dano', 'Em análise'),
(3, '2025-05-10', 'Tentativa de roubo', 'Roubo', 'Concluído'),
(4, '2025-06-01', 'Dano acidental em viagem', 'Dano', 'Em análise'),
(6, '2025-05-18', 'Dano elétrico', 'Dano', 'Concluído'),
(11, '2025-06-10', 'Queda grave', 'Dano', 'Concluído'),
(12, '2025-06-12', 'Tela destruída', 'Dano', 'Concluído'),
(6, '2025-06-14', 'Bateria inchada', 'Dano', 'Concluído');

```

```

INSERT INTO Reparos (id_sinistro, data_envio, data_conclusao, custo_reparo) VALUES

```

```

(2, '2025-02-21', '2025-03-01', 500.00),
(4, '2025-03-16', '2025-03-25', 700.00),
(5, '2025-04-11', '2025-04-20', 400.00),
(8, '2025-05-02', '2025-05-12', 600.00),
(10, '2025-05-16', '2025-05-25', 750.00),
(4, '2025-06-11', '2025-06-15', 30000.00),
(5, '2025-06-13', '2025-06-18', 40000.00),
(6, '2025-06-15', '2025-06-20', 35000.00);

```

```

INSERT INTO Pagamentos (id_apolice, data_pagamento, valor_pago, forma_pagamento, status_pagamento) VALUES

```



```
(1, GETDATE(), 29.90, 'Cartão', 'Pago'),  
(2, GETDATE(), 49.90, 'Boleto', 'Pago'),  
(3, GETDATE(), 69.90, 'Pix', 'Pago'),  
(4, GETDATE(), 89.90, 'Cartão', 'Pago'),  
(5, GETDATE(), 29.90, 'Boleto', 'Pendente'),  
(6, GETDATE(), 49.90, 'Cartão', 'Pago'),  
(7, GETDATE(), 69.90, 'Pix', 'Pago'),  
(8, GETDATE(), 89.90, 'Cartão', 'Pago'),  
(9, GETDATE(), 29.90, 'Boleto', 'Pendente'),  
(10, GETDATE(), 49.90, 'Pix', 'Pago'),  
(11, GETDATE(), 49.90, 'Pix', 'Pago'),  
(12, GETDATE(), 69.90, 'Cartão', 'Pago');
```

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

---1-Quais clientes, que possuem apólices de plano Prata ou superior, tiveram mais de um sinistro do tipo 'Roubo' ou 'Dano', com status diferente de 'Negado', ocorridos nos últimos 12 meses?

```
SELECT
    Clientes.nome AS NomeCliente,
    COUNT(Sinistros.id_sinistro) AS QuantidadeSinistros
FROM Clientes
INNER JOIN Apolices ON Clientes.id_cliente = Apolices.id_cliente
INNER JOIN Planos_Seguro ON Apolices.id_plano = Planos_Seguro.id_plano
INNER JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice
WHERE
    Planos_Seguro.nome_plano IN ('Prata', 'Ouro', 'Diamante')
    AND Sinistros.tipo_ocorrencia IN ('Roubo', 'Dano')
    AND Sinistros.status_sinistro <> 'Negado'
    AND Sinistros.data_ocorrencia >= DATEADD(MONTH, -12, GETDATE())
GROUP BY Clientes.nome
HAVING COUNT(Sinistros.id_sinistro) > 1
ORDER BY QuantidadeSinistros DESC;
```

--2-Quais os 5 modelos de aparelhos possuem o maior custo médio de reparo acima de 2000.00 e qual a posição de cada um no ranking?

```
WITH CTE AS (
    SELECT
        Aparelhos.modelo,
        AVG(Reparos.custo_reparo) AS custo_medio
    FROM Aparelhos
    INNER JOIN Apolices ON Aparelhos.id_aparelho = Apolices.id_aparelho
    INNER JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice
    INNER JOIN Reparos ON Sinistros.id_sinistro = Reparos.id_sinistro
    GROUP BY Aparelhos.modelo
),
CTE_RANK AS (
    SELECT
        modelo,
        custo_medio,
        RANK() OVER (ORDER BY custo_medio DESC) AS posicao
    FROM CTE
    WHERE custo_medio > 2000
)
SELECT *
FROM CTE_RANK
```

```
WHERE posicao <= 5;
```

--3-Qual a média de tempo (em dias) entre a ocorrência do sinistro e a conclusão do reparo, segmentada por modelo do aparelho e plano de seguro contratado?

```
WITH TempoReparo AS (  
    SELECT  
        Aparelhos.modelo,  
        Planos_Seguro.nome_plano,  
        DATEDIFF(DAY, Sinistros.data_ocorrencia, Reparos.data_conclusao) AS dias_para_reparo  
    FROM Sinistros  
    INNER JOIN Reparos ON Sinistros.id_sinistro = Reparos.id_sinistro  
    INNER JOIN Apolices ON Sinistros.id_apolice = Apolices.id_apolice  
    INNER JOIN Aparelhos ON Apolices.id_aparelho = Aparelhos.id_aparelho  
    INNER JOIN Planos_Seguro ON Apolices.id_plano = Planos_Seguro.id_plano  
    WHERE Reparos.data_conclusao IS NOT NULL  
    AND Reparos.data_conclusao >= Sinistros.data_ocorrencia  
)  
SELECT  
    modelo,  
    nome_plano,  
    CAST(AVG(dias_para_reparo) AS DECIMAL(5,2)) AS media_dias_reparo  
FROM TempoReparo  
GROUP BY modelo, nome_plano  
ORDER BY media_dias_reparo DESC;
```

--4-Qual o plano de seguro com maior índice de sinistros por apólices ativas? Trate dos casos de planos que não tenham apólices ativas.

```
SELECT  
    Planos_Seguro.nome_plano,  
    COALESCE(CAST(COALESCE(COUNT(DISTINCT Sinistros.id_sinistro), 0) * 1.0 / NULLIF(COUNT(DISTINCT Apolices.id_apolice), 0) AS  
DECIMAL(5,2)), 0) AS indice_sinistros  
FROM Planos_Seguro  
LEFT JOIN Apolices ON Planos_Seguro.id_plano = Apolices.id_plano AND Apolices.status_apolice = 'Ativa'  
LEFT JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice  
GROUP BY Planos_Seguro.id_plano, Planos_Seguro.nome_plano  
ORDER BY indice_sinistros DESC;
```

--5-Quais clientes, com apólices ativas, que tiveram sinistros concluídos nos últimos 6 meses e cujo tipo de ocorrência foi 'Roubo' ou 'Dano', geraram mais despesas para a seguradora, considerando apenas os custos de reparos?

```
SELECT  
    Clientes.nome AS NomeCliente,  
    SUM(ISNULL(Reparos.custo_reparo, 0)) AS TotalDespesas  
FROM Clientes
```

```

INNER JOIN Apolices ON Clientes.id_cliente = Apolices.id_cliente
INNER JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice
LEFT JOIN Reparos ON Sinistros.id_sinistro = Reparos.id_sinistro
WHERE
    Apolices.status_apolice = 'Ativa'
    AND Sinistros.status_sinistro = 'Concluído'
    AND Sinistros.data_ocorrencia >= DATEADD(MONTH, -6, GETDATE())
    AND Sinistros.tipo_ocorrencia IN ('Roubo', 'Dano')
GROUP BY Clientes.nome
ORDER BY TotalDespesas DESC;

```

--6-Faça um consulta por ID de um cliente e retorne: Dados cadastrais, Total de apólices, Total de sinistros, Valor total pago, Valor total de reparos. Ou seja, a ficha de um cliente.

```

CREATE OR ALTER PROCEDURE sp_resumo_cliente @id_cliente INT AS
BEGIN
    DECLARE @total_apolices INT,
            @total_sinistros INT,
            @total_pago DECIMAL(10,2),
            @total_reparos DECIMAL(10,2);

    SELECT @total_apolices = COUNT(*)
    FROM Apolices
    WHERE id_cliente = @id_cliente;

    SELECT @total_sinistros = COUNT(*)
    FROM Apolices
    INNER JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice
    WHERE Apolices.id_cliente = @id_cliente;

    SELECT @total_pago = SUM(valor_pago)
    FROM Apolices
    INNER JOIN Pagamentos ON Apolices.id_apolice = Pagamentos.id_apolice
    WHERE Apolices.id_cliente = @id_cliente AND Pagamentos.status_pagamento = 'Pago';

    SELECT @total_reparos = SUM(Reparos.custo_reparo)
    FROM Apolices
    INNER JOIN Sinistros ON Apolices.id_apolice = Sinistros.id_apolice
    INNER JOIN Reparos ON Sinistros.id_sinistro = Reparos.id_sinistro
    WHERE Apolices.id_cliente = @id_cliente;

    SELECT
        Clientes.*,
        @total_apolices AS total_apolices,
        @total_sinistros AS total_sinistros,

```

```
        @total_pago AS total_pago,  
        @total_reparos AS total_reparos  
FROM Clientes  
WHERE id_cliente = @id_cliente;  
END;  
  
EXEC sp_resumo_cliente @id_cliente = 1;
```