**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 Seguradora de Imóveis**

Ângelo José da Rosa (@angelum23)

Bruno Frelo Venturini (@Bruno-Venturini)

Eduardo Fraga de Freitas (@dufrtss)

Gabriel Ferreira Guinzani

Guilherme Sávio (@guilherme-savio)

Gustavo Thomé de Jesus (@guga2905)

Higor Goulart Massiroli (@higorgoulart)

Sofia Martins Silva (@SofiaMartinslv)

Criciúma, 13/06/2023

**Modelo ER Físico**

**Interface gráfica do usuário

Descrição gerada automaticamente**

**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Apolice | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados das apolices de seguro | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_apolice | | int | 1 – sem limite | NOT NULL | X |  | Código identificador da apólice |
| id\_imovel | | int | 1 – sem limite | NOT NULL |  | X | Código identificador do imóvel vinculado |
| dt\_inicio | | date | 1 – sem limite | NOT NULL |  |  | Data de início da apólice |
| dt\_termino | | date | 1 – sem limite | NOT NULL |  |  | Data de término da apólice |
| valor\_apolice | | decimal | 1 - sem limite | NOT NULL |  |  | Valor coberto pela apólice |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Imovel | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados do imóvel assegurado | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_imovel | | int | 1 – sem limite | NOT NULL | X |  | Código identificador do imóvel |
| id\_proprietario | | int | 1 – sem limite | NOT NULL |  | X | Código identificador do proprietário vinculado |
| id\_inquilino | | int | 1 – sem limite | NULL |  | X | Código identificador do inquilino vinculado |
| endereco | | varchar | 1 – sem limite | NOT NULL |  |  | Endereço onde o imóvel se encontra |
| tipo\_imovel | | varchar | 1 – sem limite | NOT NULL |  |  | Tipologia da construção do imóvel (Casa, Apt.) |
| valor\_imovel | | decimal | 1 – sem limite | NOT NULL |  |  | Valor de compra e venda do imóvel |
| area\_imovel | | decimal | 1 - sem limite | NOT NULL |  |  | Área ocupada pelo imóvel |
| ano\_construcao | | int | 1 a 4 | NOT NULL |  |  | Ano de construção do Imóvel |
| **Índice** |  |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
| **Tabela** | Avaliacao | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados de avaliação de imóveis assegurados | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_avaliacao | | int | 1 – sem limite | NOT NULL | X |  | Código identificador da avaliacao |
| id\_imovel | | int | 1 – sem limite | NOT NULL |  | X | Código identificador do imóvel |
| dt\_avaliacao | | date | 1 – sem limite | NOT NULL |  |  | data da avaliação |
| valor\_avaliado | | decimal | 1 – sem limite | NOT NULL |  |  | Valor avaliado de cobertura do imóvel |
| **Índice** |  |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
| **Tabela** | Sinistro | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados de sinistros | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_sinistro | | int | 1 – sem limite | NOT NULL | X |  | Código identificador do sinistro |
| id\_apolice | | int | 1 – sem limite | NOT NULL |  | X | Código identificador da apólice |
| dt\_pagamento | | date | 1 – sem limite | NOT NULL |  |  | Data de pagamento vinculada ao sinistro |
| valor\_pagamento | | decimal | 1 - sem limite | NOT NULL |  |  | Valor de pagamento vinculado ao sinistro |
| **Índice** | |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Tabela** | Apolice\_Cobertura | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados necessários para atribuir coberturas a apólices | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_sinistro | | int | 1 – sem limite | NOT NULL |  | X | Código identificador do sinistro |
| id\_cobertura | | int | 1 – sem limite | NOT NULL |  | X | Código identificador da cobertura |
| **Índice** |  |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
| **Tabela** | Cobertura | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados de cobertura | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_cobertura | | int | 1 – sem limite | NOT NULL | X |  | Código identificador da cobertura |
| descricao | | varchar | 1 – sem limite | NOT NULL |  |  | Descrição da Cobertura |
| valor | | decimal | 1 - sem limite | NOT NULL |  |  | Valor de pagamento vinculado ao sinistro |
| **Índice** | |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Tabela** | Cliente | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos clientes | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_cliente | | int | 1 – sem limite | NOT NULL | X |  | Código identificador do cliente |
| nome | | varchar | 1 – sem limite | NOT NULL |  |  | Nome do cliente |
| dt\_nasc | | date | 1 - sem limite | NOT NULL |  |  | Data de nascimento do cliente |
| endereco | | varchar | 1 - sem limite | NOT NULL |  |  | Endereço do cliente |
| telefone | | varchar | 1 - sem limite | NOT NULL |  |  | Telefone do cliente |
| email | | varchar | 1 - sem limite | NOT NULL |  |  | E-mail do cliente |
| **Índice** | |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
|  |  |  |  |  |  |  |  |
| **Tabela** | Pagamento | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados de pagamentos | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_pagamento | | int | 1 – sem limite | NOT NULL | X |  | Código identificador do pagamento |
| id\_apolice | | int | 1 – sem limite | NOT NULL |  | X | Código de identificador da apólice |
| dt\_pagamento | | date | 1 - sem limite | NOT NULL |  |  | Data do pagamento |
| valor\_pagamento | | decimal | 1 - sem limite | NOT NULL |  |  | Valor de pagamento |
| **Índice** | |  |  |  |  |  |  |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |

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

CREATE TABLE IF NOT EXISTS `cliente` (

`id\_cliente` int AUTO\_INCREMENT PRIMARY KEY,

`nome` varchar(255),

`dt\_nasc` date,

`endereco` varchar(255),

`telefone` varchar(255),

`email` varchar(255)

);

CREATE TABLE IF NOT EXISTS `imovel` (

`id\_imovel` int AUTO\_INCREMENT PRIMARY KEY,

`id\_proprietario` int,

`id\_inquilino` int,

`endereco` varchar(255),

`tipo\_imovel` varchar(255),

`valor\_imovel` decimal(12,2),

`area\_imovel` decimal(12,2),

`ano\_construcao` int

);

CREATE TABLE IF NOT EXISTS `apolice` (

`id\_apolice` int AUTO\_INCREMENT PRIMARY KEY,

`id\_imovel` int,

`dt\_inicio` date,

`dt\_termino` date,

`valor\_apolice` decimal(12,2)

);

CREATE TABLE IF NOT EXISTS `cobertura` (

`id\_cobertura` int AUTO\_INCREMENT PRIMARY KEY,

`descricao` varchar(255),

`valor` decimal(12,2)

);

CREATE TABLE IF NOT EXISTS `apolice\_cobertura` (

`id\_apolice` int,

`id\_cobertura` int

);

CREATE TABLE IF NOT EXISTS `sinistro` (

`id\_sinistro` int AUTO\_INCREMENT PRIMARY KEY,

`id\_apolice` int,

`dt\_sinistro` date,

`descricao` varchar(255),

`valor\_sinistro` decimal(12,2)

);

CREATE TABLE IF NOT EXISTS `pagamento` (

`id\_pagamento` int AUTO\_INCREMENT PRIMARY KEY,

`id\_apolice` int,

`dt\_pagamento` date,

`valor\_pagamento` decimal(12,2)

);

CREATE TABLE IF NOT EXISTS `avaliacao` (

`id\_avaliacao` int AUTO\_INCREMENT PRIMARY KEY,

`id\_imovel` int,

`dt\_avaliacao` date,

`valor\_avaliado` decimal(12,2)

);

ALTER TABLE `imovel` ADD FOREIGN KEY (`id\_proprietario`) REFERENCES `cliente` (`id\_cliente`);

ALTER TABLE `imovel` ADD FOREIGN KEY (`id\_inquilino`) REFERENCES `cliente` (`id\_cliente`);

ALTER TABLE `apolice` ADD FOREIGN KEY (`id\_imovel`) REFERENCES `imovel` (`id\_imovel`);

ALTER TABLE `apolice\_cobertura` ADD FOREIGN KEY (`id\_apolice`) REFERENCES `apolice` (`id\_apolice`);

ALTER TABLE `apolice\_cobertura` ADD FOREIGN KEY (`id\_cobertura`) REFERENCES `cobertura` (`id\_cobertura`);

ALTER TABLE `sinistro` ADD FOREIGN KEY (`id\_apolice`) REFERENCES `apolice` (`id\_apolice`);

ALTER TABLE `pagamento` ADD FOREIGN KEY (`id\_apolice`) REFERENCES `apolice` (`id\_apolice`);

ALTER TABLE `avaliacao` ADD FOREIGN KEY (`id\_imovel`) REFERENCES `imovel` (`id\_imovel`);

**Script que popula as tabelas do Banco de dados**

-- Inserção de dados na tabela cliente

INSERT INTO cliente (id\_cliente, nome, dt\_nasc, endereco, telefone, email)

VALUES

(1, 'João', '1990-05-15', 'Rua A', '123456789', 'joao.silva@example.com'),

(2, 'Maria', '1985-09-10', 'Rua B', '987654321', 'maria.santos@example.com'),

(3, 'Pedro', '1992-02-28', 'Rua C', '555555555', 'pedro.almeida@example.com'),

(4, 'Ana', '1998-11-03', 'Rua D', '111111111', 'ana.ferreira@example.com'),

(5, 'Carlos', '1982-07-20', 'Rua E', '222222222', 'carlos.oliveira@example.com'),

(6, 'Mariana', '1995-04-12', 'Rua F', '333333333', 'mariana.costa@example.com'),

(7, 'Lucas', '1991-08-25', 'Rua G', '444444444', 'lucas.ribeiro@example.com'),

(8, 'Fernanda', '1987-12-18', 'Rua H', '666666666', 'fernanda.gomes@example.com'),

(9, 'Ricardo', '1997-03-08', 'Rua I', '777777777', 'ricardo.martins@example.com'),

(10, 'Isabela', '1994-06-30', 'Rua J', '999999999', 'isabela.lima@example.com');

-- Inserção de dados na tabela imovel

INSERT INTO imovel (id\_imovel, id\_proprietario, id\_inquilino, endereco, tipo\_imovel, valor\_imovel, area\_imovel, ano\_construcao)

VALUES

(1, 1, 2, 'Rua X', 'Casa', 250000.00, 200.00, 2000),

(2, 3, NULL, 'Rua Y', 'Apartamento', 150000.00, 100.00, 2010),

(3, 4, 5, 'Rua Z', 'Casa', 180000.00, 180.50, 1995),

(4, 6, NULL, 'Rua W', 'Apartamento', 200000.00, 120.00, 2005),

(5, 7, 8, 'Rua V', 'Casa', 300000.00, 300.00, 2015),

(6, 9, NULL, 'Rua U', 'Apartamento', 220000.00, 150.00, 2008),

(7, 10, NULL, 'Rua T', 'Casa', 280000.00, 220.00, 2003),

(8, 2, NULL, 'Rua S', 'Apartamento', 190000.00, 110.00, 2012),

(9, 5, 9, 'Rua R', 'Casa', 260000.00, 250.00, 2006),

(10, 8, NULL, 'Rua Q', 'Apartamento', 230000.00, 130.00, 2018);

-- Inserção de dados na tabela apolice

INSERT INTO apolice (id\_apolice, id\_imovel, dt\_inicio, dt\_termino, valor\_apolice)

VALUES

(1, 1, '2021-01-01', '2023-12-31', 1500.00),

(2, 2, '2021-02-01', '2023-07-31', 1200.00),

(3, 3, '2021-03-01', '2023-11-30', 1350.00),

(4, 4, '2021-04-01', '2023-09-30', 1400.00),

(5, 5, '2021-05-01', '2023-10-31', 1600.00),

(6, 6, '2021-06-01', '2023-05-31', 1300.00),

(7, 7, '2021-07-01', '2023-12-31', 1700.00),

(8, 8, '2021-08-01', '2023-10-31', 1250.00),

(9, 9, '2021-09-01', '2023-11-30', 1450.00),

(10, 10, '2021-10-01', '2023-12-31', 1550.00),

(11, 9, '2023-06-28', '2023-06-30', 1600.00);

-- Inserção de dados na tabela cobertura

INSERT INTO cobertura (id\_cobertura, descricao, valor)

VALUES

(1, 'Incêndio', 50000.00),

(2, 'Roubo', 10000.00),

(3, 'Danos Elétricos', 15000.00),

(4, 'Vendaval', 20000.00),

(5, 'Desmoronamento', 30000.00),

(6, 'Responsabilidade Civil', 25000.00),

(7, 'Perda de Aluguel', 1000.00),

(8, 'Inundação', 35000.00),

(9, 'Furto', 8000.00),

(10, 'Desastres Naturais', 40000.00);

-- Inserção de dados na tabela apolice\_cobertura

INSERT INTO apolice\_cobertura (id\_apolice, id\_cobertura)

VALUES

(1, 1),

(1, 2),

(1, 3),

(2, 3),

(3, 4),

(3, 5),

(4, 6),

(4, 7),

(5, 2),

(5, 8),

(5, 9),

(6, 10);

-- Inserção de dados na tabela sinistro

INSERT INTO sinistro (id\_sinistro, id\_apolice, dt\_sinistro, descricao, valor\_sinistro)

VALUES

(1, 1, '2022-03-15', 'Incêndio no imóvel', 20000.00),

(2, 3, '2022-06-20', 'Desmoronamento parcial', 15000.00),

(3, 4, '2022-07-10', 'Roubo de pertences', 5000.00),

(4, 7, '2022-09-05', 'Inundação causada por enchente', 10000.00),

(5, 8, '2022-11-18', 'Danos elétricos', 8000.00),

(6, 10, '2022-12-24', 'Vendaval destruiu telhado', 12000.00),

(7, 10, '2022-12-27', 'Perda de aluguel devido a sinistro', 500.00),

(8, 2, '2022-05-10', 'Furto de objetos valiosos', 7000.00),

(9, 5, '2022-08-15', 'Desastre natural causou danos', 18000.00),

(10, 6, '2022-09-30', 'Roubo de aparelhos eletrônicos', 3000.00);

-- Inserção de dados na tabela pagamento

INSERT INTO pagamento (id\_pagamento, id\_apolice, dt\_pagamento, valor\_pagamento)

VALUES

(1, 1, '2023-02-05', 300.00),

(2, 2, '2023-03-10', 250.00),

(3, 3, '2023-04-15', 275.00),

(4, 4, '2023-05-20', 280.00),

(5, 5, '2023-06-25', 320.00),

(6, 6, '2023-02-12', 260.00),

(7, 7, '2023-08-05', 340.00),

(8, 8, '2023-09-10', 225.00),

(9, 9, '2023-10-15', 290.00),

(10, 10, '2023-11-20', 310.00),

(11, 11, '2023-11-22', 310);

-- Inserção de dados na tabela avaliacao

INSERT INTO avaliacao (id\_avaliacao, id\_imovel, dt\_avaliacao, valor\_avaliado)

VALUES

(1, 1, '2021-02-20', 260000.00),

(2, 2, '2021-03-25', 170000.00),

(3, 3, '2021-04-30', 190000.00),

(4, 4, '2021-05-05', 210000.00),

(5, 5, '2021-06-10', 320000.00),

(6, 6, '2021-07-15', 230000.00),

(7, 7, '2021-08-20', 270000.00),

(8, 8, '2021-09-25', 180000.00),

(9, 9, '2021-10-30', 200000.00),

(10, 10, '2021-11-05', 240000.00);

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

--1) Consulta de sinistros por tipo de imóvel e valor médio de sinistro:

SELECT i.tipo\_imovel, COUNT(s.id\_sinistro) AS total\_sinistros, ROUND(AVG(s.valor\_sinistro),2) AS valor\_medio\_sinistro

FROM imovel i

LEFT JOIN apolice a ON i.id\_imovel = a.id\_imovel

LEFT JOIN sinistro s ON a.id\_apolice = s.id\_apolice

GROUP BY i.tipo\_imovel

ORDER BY total\_sinistros DESC;

--2) Consulta de pagamentos atrasados por apólice:

--Essa consulta lista os pagamentos que foram realizados após a data de término da apólice, indicando possíveis pagamentos atrasados. Os resultados são ordenados pelo ID da apólice e pela data de pagamento.

SELECT a.id\_apolice, a.dt\_inicio, a.dt\_termino, p.dt\_pagamento, p.valor\_pagamento

FROM apolice a

JOIN pagamento p ON a.id\_apolice = p.id\_apolice

WHERE p.dt\_pagamento > a.dt\_termino

ORDER BY a.id\_apolice, p.dt\_pagamento;

--3) Consulta de apólices com pelo menos três tipos de coberturas vinculadas:

SELECT a.id\_apolice, a.dt\_inicio, a.dt\_termino, GROUP\_CONCAT(c.descricao SEPARATOR ', ') AS coberturas

FROM apolice a

JOIN apolice\_cobertura ac ON a.id\_apolice = ac.id\_apolice

JOIN cobertura c ON ac.id\_cobertura = c.id\_cobertura

where (select count(\*) from apolice\_cobertura ac2 where ac2.id\_apolice = a.id\_apolice) > 2

GROUP BY a.id\_apolice, a.dt\_inicio, a.dt\_termino;

--4) Consulta de média de avaliação por tipo de imóvel:

SELECT i.tipo\_imovel, AVG(a.valor\_avaliado) AS media\_avaliacao

FROM imovel i

LEFT JOIN avaliacao a ON i.id\_imovel = a.id\_imovel

GROUP BY i.tipo\_imovel;

--5) Consulta de valor total de sinistros por mês:

SELECT MONTH(s.dt\_sinistro) AS mes, YEAR(s.dt\_sinistro) AS ano, SUM(s.valor\_sinistro) AS valor\_total\_sinistros

FROM sinistro s

GROUP BY mes, ano

ORDER BY ano, mes;

--6) Consulta demonstrando funcionamento da função:

SELECT c.nome as nome\_prop, i.id\_imovel, verificar\_apolice\_ativa(i.id\_imovel) as is\_ativa

FROM cliente c

INNER JOIN imovel i ON id\_cliente = id\_proprietario;