**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 Estoque para E-commerce**

Arthur Ghizi - https://github.com/Arthu085

Cristoffer Figueiredo Martins - https://github.com/cristoffer21

Gabriel Jorge Lóh - https://github.com/gabrieljloh

Gabriel Custódio Boelter - https://github.com/GabrielBoelter

Jean Lucas Preis - https://github.com/JeanPrei

João Vitor Figueiredo Espindola - https://github.com/joaovfe

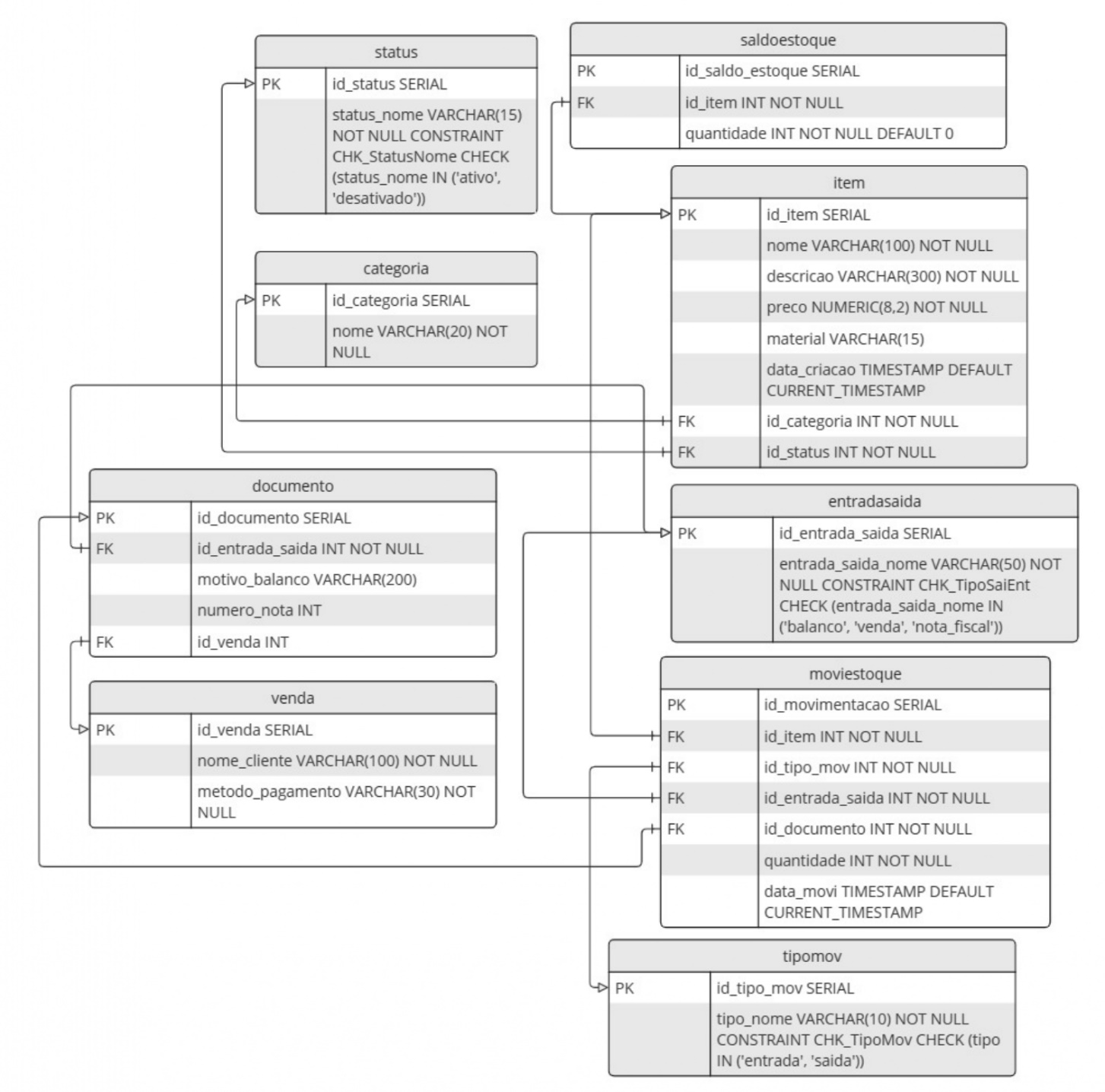
Criciúma, 13/11/2023

**URL do projeto no GitHub**

https://github.com/gabrieljloh/StoreStocker

****

**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 – sem limite | 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 – sem limite | 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**

-- Tabela: status

CREATE TABLE status (

id\_status SERIAL PRIMARY KEY,

status\_nome VARCHAR(15) NOT NULL CONSTRAINT CHK\_StatusNome CHECK (status\_nome IN ('ativo', 'desativado'))

);

-- Tabela: categoria

CREATE TABLE categoria (

id\_categoria SERIAL PRIMARY KEY,

nome VARCHAR(20) NOT NULL

);

-- Tabela: item

CREATE TABLE item (

id\_item SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

descricao VARCHAR(300) NOT NULL,

preco NUMERIC(8,2) NOT NULL,

material VARCHAR(15),

data\_criacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

id\_categoria INT NOT NULL REFERENCES categoria(id\_categoria),

id\_status INT NOT NULL REFERENCES status(id\_status)

);

-- Tabela: tipomov

CREATE TABLE tipomov (

id\_tipo\_mov SERIAL PRIMARY KEY,

tipo\_nome VARCHAR(10) NOT NULL CONSTRAINT CHK\_TipoMov CHECK (tipo\_nome IN ('entrada', 'saida'))

);

-- Tabela: entradasaida

CREATE TABLE entradasaida (

id\_entrada\_saida SERIAL PRIMARY KEY,

entrada\_saida\_nome VARCHAR(50) NOT NULL CONSTRAINT CHK\_TipoSaiEnt CHECK (entrada\_saida\_nome IN ('balanco', 'venda', 'nota\_fiscal'))

);

-- Tabela: venda

CREATE TABLE venda (

id\_venda SERIAL PRIMARY KEY,

nome\_cliente VARCHAR(100) NOT NULL,

metodo\_pagamento VARCHAR(30) NOT NULL,

);

-- Tabela: documento

CREATE TABLE documento (

id\_documento SERIAL PRIMARY KEY,

id\_entrada\_saida INT NOT NULL REFERENCES entradasaida(id\_entrada\_saida),

motivo\_balanco VARCHAR(200),

numero\_nota INT,

id\_venda INT REFERENCES venda(id\_venda)

);

-- Tabela: moviestoque

CREATE TABLE moviestoque (

id\_movimentacao SERIAL PRIMARY KEY,

id\_item INT NOT NULL REFERENCES item(id\_item),

id\_tipo\_mov INT NOT NULL REFERENCES tipomov(id\_tipo\_mov),

id\_entrada\_saida INT NOT NULL REFERENCES entradasaida(id\_entrada\_saida),

id\_documento INT NOT NULL REFERENCES documento(id\_documento),

quantidade INT NOT NULL,

data\_movi TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Tabela: saldoestoque

CREATE TABLE saldoestoque (

id\_saldo\_estoque SERIAL PRIMARY KEY,

id\_item INT NOT NULL REFERENCES item(id\_item),

quantidade INT NOT NULL DEFAULT 0

);

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

-- Inserções na tabela: status

INSERT INTO status (status\_nome) VALUES

('ativo'),

('desativado');

-- Inserções na tabela: categoria

INSERT INTO categoria (nome) VALUES

('Anéis'),

('Pulseiras'),

('Colares'),

('Brincos'),

('Relógios');

-- Inserções na tabela: tipomov

INSERT INTO tipomov (tipo\_nome) VALUES

('entrada'),

('saida');

-- Inserções na tabela: entradasaida

INSERT INTO entradasaida (entrada\_saida\_nome) VALUES

('balanco'),

('venda'),

('nota\_fiscal');

-- Insercões na tabela: item

INSERT INTO item (nome, descricao, preco, material, id\_categoria, id\_status) VALUES

('Anel de Ouro 18k', 'Anel delicado de ouro 18k com detalhes minimalistas.', 1500.00, 'Ouro', 1, 1),

('Pulseira de Prata', 'Pulseira de prata com fecho ajustável.', 250.00, 'Prata', 2, 1),

('Colar com Pingente de Pérola', 'Colar dourado com um pingente de pérola natural.', 900.00, 'Dourado', 3, 1),

('Brinco de Diamante', 'Brinco de ouro branco com diamantes lapidados.', 2500.00, 'Ouro Branco', 4, 1),

('Relógio de Aço Inoxidável', 'Relógio resistente com design moderno.', 300.00, 'Aço', 5, 1),

('Anel com Safira', 'Anel de ouro amarelo com uma safira azul.', 3500.00, 'Ouro Amarelo', 1, 1),

('Pulseira de Couro', 'Pulseira masculina de couro legítimo.', 120.00, 'Couro', 2, 1),

('Colar de Rubi', 'Colar de prata com um pingente de rubi.', 1200.00, 'Prata', 3, 1),

('Brinco de Pérola', 'Brinco clássico com pérolas naturais.', 800.00, 'Prata', 4, 1),

('Relógio Dourado', 'Relógio elegante com acabamento dourado.', 450.00, 'Aço Dourado', 5, 1),

('Anel de Prata com Cristal', 'Anel de prata com um cristal lapidado.', 200.00, 'Prata', 1, 1),

('Pulseira com Charme', 'Pulseira de prata com charms personalizados.', 320.00, 'Prata', 2, 1),

('Colar de Topázio', 'Colar de ouro branco com uma pedra de topázio.', 1800.00, 'Ouro Branco', 3, 1),

('Brinco de Esmeralda', 'Brinco de ouro amarelo com esmeraldas brilhantes.', 2800.00, 'Ouro Amarelo', 4, 1),

('Relógio Casual', 'Relógio casual com pulseira de couro.', 200.00, 'Couro', 5, 1),

('Anel de Noivado', 'Anel de platina com um diamante central.', 4500.00, 'Platina', 1, 1),

('Pulseira de Pérolas', 'Pulseira feita de pérolas cultivadas.', 1100.00, 'Pérola', 2, 1),

('Colar com Pingente de Coração', 'Colar dourado com um pingente de coração cravejado.', 950.00, 'Dourado', 3, 1),

('Brinco de Argola', 'Brinco de argola grande com design moderno.', 300.00, 'Aço', 4, 1),

('Relógio de Luxo', 'Relógio suíço com acabamento impecável.', 5000.00, 'Ouro', 5, 1),

('Anel de Rubi', 'Anel de ouro branco com uma pedra de rubi.', 3200.00, 'Ouro Branco', 1, 1),

('Pulseira com Pedras Naturais', 'Pulseira com mix de pedras naturais coloridas.', 450.00, 'Pedras Naturais', 2, 1),

('Colar de Safira', 'Colar de ouro amarelo com uma safira brilhante.', 2800.00, 'Ouro Amarelo', 3, 1),

('Brinco com Pedras Coloridas', 'Brinco moderno com pedras coloridas variadas.', 600.00, 'Prata', 4, 1),

('Relógio Digital', 'Relógio digital esportivo com diversas funções.', 180.00, 'Plástico', 5, 1),

('Anel de Turmalina Paraíba', 'Anel exclusivo com uma turmalina Paraíba rara.', 7200.00, 'Platina', 1, 1),

('Pulseira Minimalista', 'Pulseira dourada com design clean e moderno.', 290.00, 'Dourado', 2, 1),

('Colar de Esmeralda', 'Colar de prata com um pingente de esmeralda.', 2500.00, 'Prata', 3, 1),

('Brinco de Ouro Rosé', 'Brinco pequeno com acabamento em ouro rosé.', 400.00, 'Ouro Rosé', 4, 1),

('Relógio Clássico', 'Relógio com design clássico e pulseira de aço.', 350.00, 'Aço', 5, 1);

-- As inserções abaixo somente funcionam após a adição da trigger

-- Inserções de estoque por balanço

INSERT INTO documento (id\_entrada\_saida, motivo\_balanco) VALUES

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque'),

(1, 'Balanço de estoque');

INSERT INTO moviestoque (id\_item, id\_tipo\_mov, id\_entrada\_saida, id\_documento, quantidade) VALUES

(5, 1, 1, 1, 50),

(3, 1, 1, 2, 80),

(7, 1, 1, 3, 90),

(9, 1, 1, 4, 770),

(2, 1, 1, 5, 55),

(1, 1, 1, 6, 90),

(4, 1, 1, 7, 10);

-- Inserções de estoque por nota fiscal

INSERT INTO documento (id\_entrada\_saida, numero\_nota) VALUES

(3, 180639),

(3, 180640),

(3, 180641),

(3, 180642),

(3, 180643),

(3, 180644),

(3, 180645);

INSERT INTO moviestoque (id\_item, id\_tipo\_mov, id\_entrada\_saida, id\_documento, quantidade) VALUES

(5, 1, 3, 8, 10),

(3, 1, 3, 9, 20),

(7, 1, 3, 10, 30),

(9, 1, 3, 11, 70),

(2, 1, 3, 12, 95),

(1, 1, 3, 13, 98),

(4, 1, 3, 14, 17);

-- Inserções de saída de estoque por venda

INSERT INTO venda (nome\_cliente, metodo\_pagamento) VALUES

('José da Silva', 'PIX'),

('Pedro Souza', 'Débito'),

('João Pedro', 'Crédito');

INSERT INTO documento (id\_entrada\_saida, id\_venda) VALUES

(2, 1),

(2, 2),

(2, 3);

INSERT INTO moviestoque (id\_item, id\_tipo\_mov, id\_entrada\_saida, id\_documento, quantidade) VALUES

(5, 2, 2, 11, 5),

(3, 2, 2, 12, 4),

(7, 2, 2, 13, 20);

-- Essas inserções de estoque serão feitas dessa maneira por ser o script inicial com o banco, mas as próximas inserções de estoque serão feitas a partir da procedure criada.

**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)