**Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof.Luciano B. Fernandes

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

**Projeto de banco de dados para um sistema de Suprimento (Almoxarifado/Compras)**

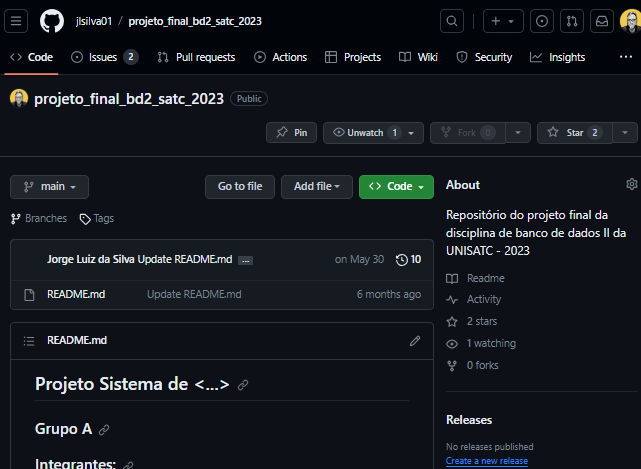
**RAFAEL RONSONI GAIDZINSKI**

**WILLIAM ESPINDOLA CARDOSO**

Criciúma, 24/06/2024

**URL do projeto no GitHub (complementar)**

https://github.com/jlsilva01/projeto\_final\_bd2\_satc\_2023

****

**Introdução ao Tema Cadeia de Suprimentos**

**Diagrama

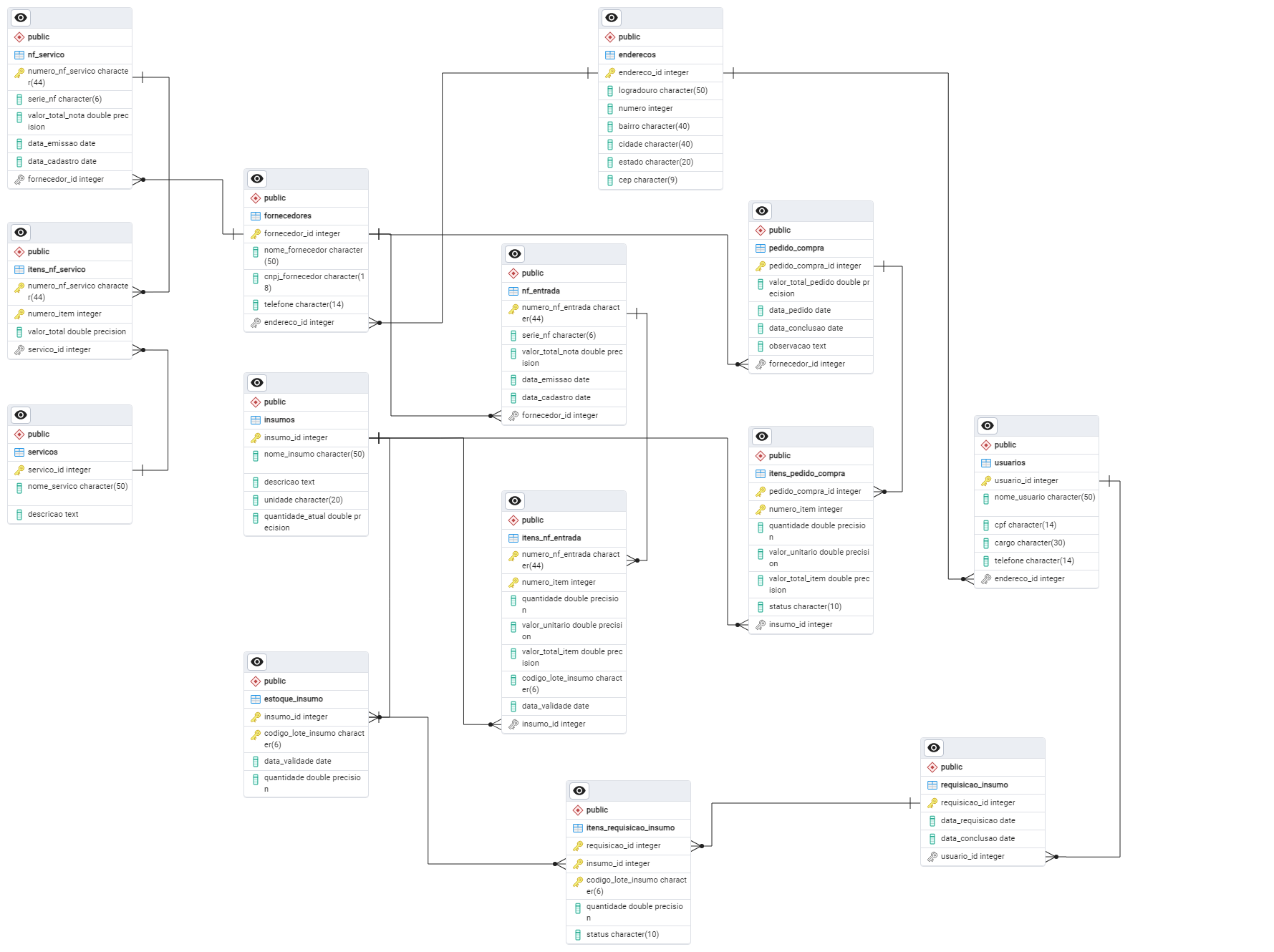
Descrição gerada automaticamente**

**Ciclo do Setor de Compras**

**Tela de celular

Descrição gerada automaticamente com confiança média**

**Modelo Físico do Banco de Dados**



**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Usuários | | | | | | |
| **Descrição** | Tabela responsável por armazenar as informações dos usuários do sistema | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| usuario\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do usuário |
| nome\_usuario | | character(50) | 1 - 50 | NOT NULL |  |  | Nome do usuário |
| cpf | | character(14) | 1 - 14 | NOT NULL |  |  | Número do CPF do usuário |
| cargo | | character(30) | 1 - 30 | NOT NULL |  |  | Nome do cargo do usuário |
| telefone | | character(14) | 1 - 14 | NOT NULL |  |  | Número de telefone do usuário |
| endereco\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código de identificador do endereço |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Endereços | | | | | | |
| **Descrição** | Tabela responsável por armazenar os endereços cadastrados | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| endereco\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do endereço |
| logradouro | | character(50) | 1 - 50 | NOT NULL |  |  | Nome da rua, avenida, rodovia, ... |
| numero | | integer | 1 – sem limite |  |  |  | Número do imóvel |
| bairro | | character(40) | 1 - 40 | NOT NULL |  |  | Nome do bairro |
| cidade | | character(40) | 1 - 40 | NOT NULL |  |  | Nome da cidade |
| estado | | character(20) | 1 - 20 | NOT NULL |  |  | Nome do estado |
| cep | | character(9) | 1 - 9 | NOT NULL |  |  | Número do código de endereço postal |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Fornecedores | | | | | | |
| **Descrição** | Tabela responsável por armazenar os dados dos fornecedores | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| fornecedor\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do fornecedor |
| nome\_fornecedor | | character(50) | 1 - 50 | NOT NULL |  |  | Nome do fornecedor |
| cnpj\_fornecedor | | character(18) | 1 - 18 | NOT NULL |  |  | Número do CNPJ do fornecedor |
| telefone | | character(14) | 1 - 14 | NOT NULL |  |  | Número do telefone do fornecedor |
| endereco\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código de identificador do endereço |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Insumos | | | | | | |
| **Descrição** | Tabela responsável por armazenar as informações dos insumos | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| insumo\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do insumo |
| nome\_insumo | | character(50) | 1 - 50 | NOT NULL |  |  | Nome do insumo |
| descricao | | text | 1 – sem limite |  |  |  | Informações sobre o insumo |
| unidade | | character(20) | 1 - 20 | NOT NULL |  |  | Informação sobre o tipo de medida (kg ou litros) |
| quantidade\_atual | | double precision | 15 casas decimais | NOT NULL |  |  | Informação sobre a quantidade atual do insumo |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Estoque Insumo | | | | | | |
| **Descrição** | Tabela responsável por armazenar o estoque dos insumos | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| insumo\_id | | integer | 1 – sem limite | NOT NULL | X | X | Código identificador do insumo |
| codigo\_lote\_insumo | | character(6) | 1 - 6 | NOT NULL | X |  | Número do código do lote do insumo |
| data\_validade | | date | 2024-06-30 | NOT NULL |  |  | Data de validade do insumo |
| quantidade | | double precision | 15 casas decimais | NOT NULL |  |  | Quantidade do insumo em estoque |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Pedido de Compra | | | | | | |
| **Descrição** | Tabela responsável por armazenar as informações do pedido de compra | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| pedido\_compra\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código de identificador do pedido de compra |
| valor\_total\_pedido | | double precision | 15 casas decimais | NOT NULL |  |  | Valor total do pedido |
| data\_pedido | | date | 2024-06-30 | NOT NULL |  |  | Data de realização do pedido |
| data\_conclusao | | date | 2024-06-30 |  |  |  | Data de fechamento do pedido |
| observação | | text | 1 – sem limite |  |  |  | Observações sobre o pedido de compra |
| fornecedor\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do fornecedor |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Itens do Pedido de Compra | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados no pedido de compra | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| pedido\_compra\_id | | integer | 1 – sem limite | NOT NULL | X | X | Código identificador do pedido de compra |
| numero\_item | | integer | 1 – sem limite | NOT NULL | X |  | Número do item registrado no pedido |
| quantidade | | double precision | 15 casas decimais | NOT NULL |  |  | Quantidade solicitada |
| valor\_unitario | | double precision | 15 casas decimais | NOT NULL |  |  | Preço unitário do item |
| valor\_total\_item | | double precision | 15 casas decimais | NOT NULL |  |  | Valor total do item |
| status | | character(10) | 1 - 10 | NOT NULL |  |  | Informação sobre o status do item pedido (Aberto, Atendido, Cancelado) |
| insumo\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do insumo |
|  |  |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| idx\_itens\_pedido\_compra\_insumo\_id | |  | X |  | insumo\_id | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Nota Fiscal de Entrada | | | | | | |
| **Descrição** | Tabela responsável por armazenar os registros das notas fiscais de entrada | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| numero\_nf\_entrada | | character(44) | 1 - 44 | NOT NULL | X |  | Número de identificação da nota fiscal |
| serie\_nf | | character(6) | 1 - 6 | NOT NULL |  |  | Número de série da nota fiscal |
| valor\_total\_nota | | double precision | 15 casas decimais | NOT NULL |  |  | Valor total da nota fiscal |
| data\_emissao | | date | 2024-06-30 | NOT NULL |  |  | Data de emissão da nota fiscal |
| data\_cadastro | | date | 2024-06-30 | NOT NULL |  |  | Data de cadastro da nota fiscal no sistema |
| fornecedor\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do fornecedor |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Itens da Nota Fiscal de Entrada | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados na nota fiscal de entrada | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| numero\_nf\_entrada | | character(44) | 1 – 44 | NOT NULL | X | X | Número de identificação da nota fiscal |
| numero\_item | | integer | 1 – sem limite | NOT NULL | X |  | Número do item registrado no pedido |
| quantidade | | double precision | 15 casas decimais | NOT NULL |  |  | Quantidade de pedido do item |
| valor\_unitario | | double precision | 15 casas decimais | NOT NULL |  |  | Preço unitário do item |
| valor\_total\_item | | double precision | 15 casas decimais | NOT NULL |  |  | Valor total do item |
| codigo\_lote\_insumo | | character(6) | 1 – 6 | NOT NULL |  |  | Número do código do lote do insumo |
| data\_validade | | date | 2024-06-30 | NOT NULL |  |  | Data de validade do insumo |
| insumo\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do insumo |
|  | | | | | | | |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| idx\_itens\_nf\_entrada\_insumo\_id | |  | X |  | insumo\_id | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Requisição de Insumo | | | | | | |
| **Descrição** | Tabela responsável por armazenar as informações da requisição de insumos | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| requisicao\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador da requisição |
| data\_requisicao | | date | 2024-06-30 | NOT NULL |  |  | Data de registro da requisição |
| data\_conclusao | | date | 2024-06-30 |  |  |  | Data de fechamento da requisição |
| usuario\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do usuário |
|  | |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Itens da Requisição de Insumo | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados na requisição de insumo | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| requisicao\_id | | integer | 1 – sem limite | NOT NULL | X | X | Código identificador da requisição |
| insumo\_id | | integer | 1 – sem limite | NOT NULL | X | X | Código identificador do insumo |
| codigo\_lote\_insumo | | character(6) | 1 - 6 | NOT NULL | X | X | Número do código do lote do insumo |
| quantidade | | double precision | 15 casas decimais | NOT NULL |  |  | Quantidade do insumo solicitada |
| status | | character(10) | 1 - 10 | NOT NULL |  |  | Informação sobre o status do item da requisição (Aberto, Atendido, Cancelado) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Índice** | | | | |
| **Nome do índice** | **Clustered** | **NonClustered** | **Unique** | **Colunas** |
| idx\_itens\_requisicao\_insumo\_insumo\_id |  | X |  | insumo\_id |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Servicos | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados na requisição | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| servico\_id | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do serviço |
| nome\_servico | | character(50) | 1 - 50 | NOT NULL |  |  | Nome do serviço |
| descricao | | text | 1 - sem limite | NOT NULL |  |  | Informações sobre o serviço |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Nota Fiscal de Serviço | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados na requisição | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| numero\_nf\_servico | | character(44) | 1 - 44 | NOT NULL | X |  | Número de identificação da nota fiscal |
| serie\_nf | | character(6) | 1 - 6 | NOT NULL |  |  | Número de série da nota fiscal |
| valor\_total\_nota | | double precision | 15 casas decimais | NOT NULL |  |  | Valor total da nota fiscal |
| data\_emissao | | date | 2024-06-30 | NOT NULL |  |  | Data de emissão da nota fiscal |
| data\_cadastro | | date | 2024-06-30 | NOT NULL |  |  | Data de cadastro da nota fiscal no sistema |
| fornecedor\_id | | integer | 1 – sem limite | NOT NULL |  | X | Código identificador do fornecedor |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | Itens da Nota Fiscal de Serviço | | | | | | |
| **Descrição** | Tabela responsável por armazenar os itens registrados na requisição | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| numero\_nf\_servico | | character(44) | 1 - 44 | NOT NULL | X | X | Código identificador da requisição |
| numero\_item | | integer | 1 – sem limite | NOT NULL | X |  | Código identificador do insumo |
| valor\_total | | double precision | 15 casas decimais | NOT NULL |  |  | Número do código do lote da produção |
| servico\_id | | integer | 1 – sem limite | NOT NULL |  | X | Quantidade solicitada |

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

BEGIN;

CREATE TABLE IF NOT EXISTS public.enderecos

(

    endereco\_id integer NOT NULL,

    logradouro character(50) COLLATE pg\_catalog."default" NOT NULL,

    numero integer,

    bairro character(40) COLLATE pg\_catalog."default" NOT NULL,

    cidade character(40) COLLATE pg\_catalog."default" NOT NULL,

    estado character(20) COLLATE pg\_catalog."default" NOT NULL,

    cep character(9) COLLATE pg\_catalog."default" NOT NULL,

    CONSTRAINT endereco\_pkey PRIMARY KEY (endereco\_id)

);

CREATE TABLE IF NOT EXISTS public.estoque\_insumo

(

    insumo\_id integer NOT NULL,

    codigo\_lote\_insumo character(6) COLLATE pg\_catalog."default" NOT NULL,

    data\_validade date NOT NULL,

    quantidade double precision NOT NULL,

    CONSTRAINT estoque\_insumo\_pkey PRIMARY KEY (insumo\_id, codigo\_lote\_insumo)

);

CREATE TABLE IF NOT EXISTS public.fornecedores

(

    fornecedor\_id integer NOT NULL,

    nome\_fornecedor character(50) COLLATE pg\_catalog."default" NOT NULL,

    cnpj\_fornecedor character(18) COLLATE pg\_catalog."default" NOT NULL,

    telefone character(14) COLLATE pg\_catalog."default" NOT NULL,

    endereco\_id integer NOT NULL,

    CONSTRAINT fornecedor\_pkey PRIMARY KEY (fornecedor\_id)

);

CREATE TABLE IF NOT EXISTS public.insumos

(

    insumo\_id integer NOT NULL,

    nome\_insumo character(50) COLLATE pg\_catalog."default" NOT NULL,

    descricao text COLLATE pg\_catalog."default",

    unidade character(20) COLLATE pg\_catalog."default" NOT NULL,

    quantidade\_atual double precision NOT NULL DEFAULT 0,

    CONSTRAINT insumos\_pkey PRIMARY KEY (insumo\_id)

);

CREATE TABLE IF NOT EXISTS public.itens\_nf\_entrada

(

    numero\_nf\_entrada character(44) COLLATE pg\_catalog."default" NOT NULL,

    numero\_item integer NOT NULL,

    quantidade double precision NOT NULL,

    valor\_unitario double precision NOT NULL,

    valor\_total\_item double precision NOT NULL,

    codigo\_lote\_insumo character(6) COLLATE pg\_catalog."default" NOT NULL,

    data\_validade date NOT NULL,

    insumo\_id integer NOT NULL,

    CONSTRAINT itens\_nf\_entrada\_pkey PRIMARY KEY (numero\_nf\_entrada, numero\_item)

);

CREATE TABLE IF NOT EXISTS public.itens\_nf\_servico

(

    numero\_nf\_servico character(44) COLLATE pg\_catalog."default" NOT NULL,

    numero\_item integer NOT NULL,

    valor\_total double precision NOT NULL,

    servico\_id integer NOT NULL,

    CONSTRAINT itens\_nf\_servico\_pkey PRIMARY KEY (numero\_nf\_servico, numero\_item)

);

CREATE TABLE IF NOT EXISTS public.itens\_pedido\_compra

(

    pedido\_compra\_id integer NOT NULL,

    numero\_item integer NOT NULL,

    quantidade double precision NOT NULL,

    valor\_unitario double precision NOT NULL,

    valor\_total\_item double precision NOT NULL,

    status character(10) COLLATE pg\_catalog."default" NOT NULL,

    insumo\_id integer NOT NULL,

    CONSTRAINT itens\_pedido\_compra\_pkey PRIMARY KEY (pedido\_compra\_id, numero\_item)

);

CREATE TABLE IF NOT EXISTS public.itens\_requisicao\_insumo

(

    requisicao\_id integer NOT NULL,

    insumo\_id integer NOT NULL,

    codigo\_lote\_insumo character(6) COLLATE pg\_catalog."default" NOT NULL,

    quantidade double precision NOT NULL,

    status character(10) COLLATE pg\_catalog."default" NOT NULL,

    CONSTRAINT itens\_requisicao\_pkey PRIMARY KEY (requisicao\_id, insumo\_id, codigo\_lote\_insumo)

);

CREATE TABLE IF NOT EXISTS public.nf\_entrada

(

    numero\_nf\_entrada character(44) COLLATE pg\_catalog."default" NOT NULL,

    serie\_nf character(6) COLLATE pg\_catalog."default" NOT NULL,

    valor\_total\_nota double precision NOT NULL DEFAULT 0,

    data\_emissao date NOT NULL,

    data\_cadastro date NOT NULL,

    fornecedor\_id integer NOT NULL,

    CONSTRAINT nf\_entrada\_pkey PRIMARY KEY (numero\_nf\_entrada)

);

CREATE TABLE IF NOT EXISTS public.nf\_servico

(

    numero\_nf\_servico character(44) COLLATE pg\_catalog."default" NOT NULL,

    serie\_nf character(6) COLLATE pg\_catalog."default" NOT NULL,

    valor\_total\_nota double precision NOT NULL DEFAULT 0,

    data\_emissao date NOT NULL,

    data\_cadastro date NOT NULL,

    fornecedor\_id integer NOT NULL,

    CONSTRAINT nf\_servico\_pkey PRIMARY KEY (numero\_nf\_servico)

);

CREATE TABLE IF NOT EXISTS public.pedido\_compra

(

    pedido\_compra\_id integer NOT NULL,

    valor\_total\_pedido double precision NOT NULL DEFAULT 0,

    data\_pedido date NOT NULL,

    data\_conclusao date,

    observacao text COLLATE pg\_catalog."default",

    fornecedor\_id integer NOT NULL,

    CONSTRAINT pedido\_compra\_pkey PRIMARY KEY (pedido\_compra\_id)

);

CREATE TABLE IF NOT EXISTS public.requisicao\_insumo

(

    requisicao\_id integer NOT NULL,

    data\_requisicao date NOT NULL,

    data\_conclusao date,

    usuario\_id integer NOT NULL,

    CONSTRAINT requisicao\_insumo\_pkey PRIMARY KEY (requisicao\_id)

);

CREATE TABLE IF NOT EXISTS public.servicos

(

    servico\_id integer NOT NULL,

    nome\_servico character(50) COLLATE pg\_catalog."default" NOT NULL,

    descricao text COLLATE pg\_catalog."default" NOT NULL,

    CONSTRAINT servicos\_pkey PRIMARY KEY (servico\_id)

);

CREATE TABLE IF NOT EXISTS public.usuarios

(

    usuario\_id integer NOT NULL,

    nome\_usuario character(50) COLLATE pg\_catalog."default" NOT NULL,

    cpf character(14) COLLATE pg\_catalog."default" NOT NULL,

    cargo character(30) COLLATE pg\_catalog."default" NOT NULL,

    telefone character(14) COLLATE pg\_catalog."default" NOT NULL,

    endereco\_id integer NOT NULL,

    CONSTRAINT usuario\_pkey PRIMARY KEY (usuario\_id)

);

ALTER TABLE IF EXISTS public.estoque\_insumo

    ADD CONSTRAINT fk\_estoque\_insumo\_insumo FOREIGN KEY (insumo\_id)

    REFERENCES public.insumos (insumo\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.fornecedores

    ADD CONSTRAINT fk\_fornecedor\_endereco FOREIGN KEY (endereco\_id)

    REFERENCES public.enderecos (endereco\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.insumos

    ADD CONSTRAINT ck\_unidade CHECK (unidade in ('kg','litros'));

ALTER TABLE IF EXISTS public.itens\_nf\_entrada

    ADD CONSTRAINT fk\_itens\_nf\_entrada\_insumo FOREIGN KEY (insumo\_id)

    REFERENCES public.insumos (insumo\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.itens\_nf\_entrada

    ADD CONSTRAINT fk\_itens\_nf\_entrada\_nf\_entrada FOREIGN KEY (numero\_nf\_entrada)

    REFERENCES public.nf\_entrada (numero\_nf\_entrada) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.itens\_nf\_servico

    ADD CONSTRAINT fk\_itens\_nf\_servico\_nf\_servico FOREIGN KEY (numero\_nf\_servico)

    REFERENCES public.nf\_servico (numero\_nf\_servico) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.itens\_nf\_servico

    ADD CONSTRAINT fk\_itens\_nf\_servico\_servico FOREIGN KEY (servico\_id)

    REFERENCES public.servicos (servico\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.itens\_pedido\_compra

    ADD CONSTRAINT fk\_itens\_pedido\_compra\_insumo FOREIGN KEY (insumo\_id)

    REFERENCES public.insumos (insumo\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.itens\_pedido\_compra

    ADD CONSTRAINT fk\_itens\_pedido\_compra\_pedido\_compra FOREIGN KEY (pedido\_compra\_id)

    REFERENCES public.pedido\_compra (pedido\_compra\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.itens\_pedido\_compra

    ADD CONSTRAINT ck\_status CHECK (status in ('Aberto','Atendido','Cancelado'));

ALTER TABLE IF EXISTS public.itens\_requisicao\_insumo

    ADD CONSTRAINT fk\_itens\_requisicao\_insumo\_estoque\_insumo FOREIGN KEY (insumo\_id, codigo\_lote\_insumo)

    REFERENCES public.estoque\_insumo (insumo\_id, codigo\_lote\_insumo) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.itens\_requisicao\_insumo

    ADD CONSTRAINT fk\_itens\_requisicao\_insumo\_requisicao\_insumo FOREIGN KEY (requisicao\_id)

    REFERENCES public.requisicao\_insumo (requisicao\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.itens\_requisicao\_insumo

    ADD CONSTRAINT ck\_status CHECK (status in ('Aberto','Atendido','Cancelado'));

ALTER TABLE IF EXISTS public.nf\_entrada

    ADD CONSTRAINT fk\_nf\_entrada\_fornecedor FOREIGN KEY (fornecedor\_id)

    REFERENCES public.fornecedores (fornecedor\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.nf\_servico

    ADD CONSTRAINT fk\_nf\_servico\_fornecedor FOREIGN KEY (fornecedor\_id)

    REFERENCES public.fornecedores (fornecedor\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.pedido\_compra

    ADD CONSTRAINT fk\_pedido\_compra\_fornecedor FOREIGN KEY (fornecedor\_id)

    REFERENCES public.fornecedores (fornecedor\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.requisicao\_insumo

    ADD CONSTRAINT fk\_requisicao\_insumo\_usuario FOREIGN KEY (usuario\_id)

    REFERENCES public.usuarios (usuario\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.usuarios

    ADD CONSTRAINT fk\_usuario\_endereco FOREIGN KEY (endereco\_id)

    REFERENCES public.enderecos (endereco\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

END;

**DML para o INSERT de dados nas tabelas do Banco de dados**

INSERT INTO public.enderecos (endereco\_id, logradouro, numero, bairro, cidade, estado, cep)

VALUES

(1, 'Rua das Flores', 100, 'Centro', 'Florianópolis', 'SC', '88000-001'),

(2, 'Avenida Brasil', 500, 'Jardins', 'Joinville', 'SC', '89200-000'),

(3, 'Rua do Comércio', 300, 'Centro', 'Blumenau', 'SC', '89010-000'),

(4, 'Rua das Palmeiras', 200, 'Boa Vista', 'Chapecó', 'SC', '89810-000'),

(5, 'Avenida Principal', 700, 'Industrial', 'Criciúma', 'SC', '88800-000'),

(6, 'Rua das Acácias', 50, 'Jardim América', 'Lages', 'SC', '88500-000'),

(7, 'Rua dos Ipês', 150, 'Centro', 'Itajaí', 'SC', '88300-000'),

(8, 'Avenida Central', 800, 'Centro', 'Jaraguá do Sul', 'SC', '89250-000'),

(9, 'Travessa das Oliveiras', 20, 'Mangabeira', 'Balneário Camboriú', 'SC', '88330-000'),

(10, 'Alameda das Rosas', 300, 'Aldeota', 'São José', 'SC', '88101-000'),

(11, 'Rua das Violetas', 80, 'Jardim das Flores', 'Palhoça', 'SC', '88130-000'),

(12, 'Avenida dos Eucaliptos', 600, 'Jardim Europa', 'Itapema', 'SC', '88220-000'),

(13, 'Rua da Praia', 1, 'Centro', 'Florianópolis', 'SC', '88015-120'),

(14, 'Rua das Gaivotas', 221, 'Barra Sul', 'Balneário Camboriú', 'SC', '88330-560'),

(15, 'Avenida das Palmeiras', 1500, 'Pioneiros', 'Itajaí', 'SC', '88301-800'),

(16, 'Rua dos Girassóis', 700, 'Centro', 'Joinville', 'SC', '89201-001'),

(17, 'Alameda das Acácias', 123, 'Lagoa', 'Florianópolis', 'SC', '88062-210'),

(18, 'Avenida das Flores', 800, 'Centro', 'Chapecó', 'SC', '89801-000'),

(19, 'Travessa das Hortênsias', 50, 'Estrela do Mar', 'Bombinhas', 'SC', '88215-000'),

(20, 'Rua das Magnólias', 300, 'Jurerê Internacional', 'Florianópolis', 'SC', '88053-000');

INSERT INTO public.usuarios (usuario\_id, nome\_usuario, cpf, cargo, telefone, endereco\_id) VALUES

(1, 'Ana Silva', '123.456.789-00', 'Analista de Compras', '(11)9876-5432', 1),

(2, 'Carlos Santos', '987.654.321-00', 'Assistente de Compras', '(22)8765-4321', 2),

(3, 'Maria Oliveira', '234.567.890-00', 'Coordenador de Compras', '(33)7654-3210', 3),

(4, 'Pedro Souza', '876.543.210-00', 'Analista Sênior de Compras', '(44)6543-2109', 4),

(5, 'Juliana Lima', '345.678.901-00', 'Estagiário de Compras', '(55)5432-1098', 5),

(6, 'Fernanda Pereira', '765.432.109-00', 'Gerente de Compras', '(66)4321-0987', 6),

(7, 'Rafaela Costa', '456.789.012-00', 'Analista de Suprimentos', '(77)3210-9876', 7),

(8, 'Lucas Mendes', '987.012.345-00', 'Comprador Técnico', '(88)2109-8765', 8),

(9, 'Gustavo Almeida', '678.901.234-00', 'Auxiliar de Compras', '(99)1098-7654', 9),

(10, 'Patrícia Santos', '012.345.678-00', 'Analista de Planejamento', '(00)0987-6543', 10);

INSERT INTO public.fornecedores (fornecedor\_id, nome\_fornecedor, cnpj\_fornecedor, telefone, endereco\_id) VALUES

(1, 'Pharma Distribuidora Ltda', '12.345.678/0001-90', '(51)9999-8888', 11),

(2, 'Central Ltda', '98.765.432/0001-21', '(48)9876-5432', 12),

(3, 'Medicamentos Expresso S.A.', '11.223.344/0001-55', '(47)8765-4321', 13),

(4, 'Remédios do Sul Eireli', '55.666.777/0001-33', '(49)7654-3210', 14),

(5, 'Drogaria Popular Ltda', '33.444.555/0001-99', '(48)6543-2109', 15),

(6, 'FarmaVida Distribuidora', '88.999.000/0001-12', '(47)5432-1098', 16),

(7, 'Bom Preço S.A.', '66.777.888/0001-44', '(48)4321-0987', 17),

(8, 'Distribuidora de Medicamentos do Norte', '09.876.543/0001-76', '(51)2109-8765', 18),

(9, 'PharmaMais Ltda', '12.345.678/0002-90', '(47)9876-5432', 19),

(10, 'São Lucas Eireli', '98.765.432/0002-21', '(48)8765-4321', 20);

INSERT INTO public.insumos (insumo\_id, nome\_insumo, descricao, unidade, quantidade\_atual) VALUES

(1, 'Paracetamol', 'Analgésico e antipirético', 'kg', 12),

(2, 'Ácido Acetilsalicílico', 'Anti-inflamatório e analgésico', 'kg', 8),

(3, 'Hidróxido de Alumínio', 'Antiácido', 'kg', 0),

(4, 'Sulfato de Neomicina', 'Antibiótico tópico', 'kg', 1),

(5, 'Citrato de Sódio', 'Anticoagulante', 'kg', 2),

(6, 'Ácido Cítrico', 'Acidulante', 'kg', 0),

(7, 'Cloridrato de Propranolol', 'Anti-hipertensivo', 'kg', 1),

(8, 'Cloridrato de Ciprofloxacino', 'Antibiótico', 'kg', 2),

(9, 'Fluoreto de Sódio', 'Agente tópico para prevenção de cáries', 'kg', 2),

(10, 'Cloreto de Sódio', 'Saline solution', 'kg', 0),

(11, 'Ácido Fólico', 'Suplemento vitamínico', 'kg', 0),

(12, 'Vitamina C', 'Ácido ascórbico', 'kg', 0),

(13, 'Ferro Sulfato', 'Suplemento de ferro', 'kg', 0),

(14, 'Cloridrato de Lidocaína', 'Anestésico local', 'kg', 0),

(15, 'Metilcelulose', 'Agente espessante', 'kg', 0);

INSERT INTO public.servicos (servico\_id, nome\_servico, descricao) VALUES

(1, 'Análise de Controle de Qualidade', 'Realização de testes físico-químicos e microbiológicos para garantir a qualidade dos produtos.'),

(2, 'Estudos de Estabilidade', 'Avaliação da estabilidade dos medicamentos em diferentes condições de armazenamento.'),

(3, 'Inspeção e Auditoria', 'Inspeção e auditoria interna e externa para garantir conformidade com boas práticas de fabricação (GMP).'),

(4, 'Monitoramento Ambiental', 'Monitoramento de condições ambientais nas áreas de produção.'),

(5, 'Treinamento em Boas Práticas de Fabricação (GMP)', 'Capacitação de pessoal em práticas adequadas de fabricação de medicamentos.'),

(6, 'Gerenciamento de Resíduos', 'Manuseio adequado e descarte de resíduos gerados durante a produção.'),

(7, 'Limpeza e Sanitização', 'Procedimentos de limpeza e sanitização de equipamentos e instalações.'),

(8, 'Controle de Documentação', 'Gerenciamento e controle de documentação técnica e regulatória.'),

(9, 'Manutenção de Equipamentos', 'Manutenção preventiva e corretiva de equipamentos utilizados na produção.'),

(10, 'Calibração de Equipamentos', 'Calibração regular de equipamentos de medição e controle.');

INSERT INTO public.pedido\_compra (pedido\_compra\_id, valor\_total\_pedido, data\_pedido, data\_conclusao, observacao, fornecedor\_id)

VALUES

(1, 1700.00, '2024-06-20', '2024-06-25', 'Urgente: entrega até dia 25/06', 1),

(2, 2810.50, '2024-06-21', NULL, 'Itens adicionais: ver disponibilidade', 2),

(3, 500.00, '2024-06-22', '2024-06-26', 'Pagamento antecipado realizado', 3),

(4, 1225.75, '2024-06-23', NULL, 'Confirmar disponibilidade de cores', 4),

(5, 3500.00, '2024-06-24', NULL, 'Embarque marítimo preferencial', 5),

(6, 815.25, '2024-06-25', '2024-06-28', 'Entrega parcial autorizada', 6),

(7, 3000.80, '2024-06-26', NULL, 'Acompanhar prazo de entrega', 7),

(8, 600.00, '2024-06-27', '2024-07-02', 'Necessário fatura', 8),

(9, 3500.00, '2024-06-28', '2024-07-03', 'Incluir brindes adicionais', 9),

(10, 71.00, '2024-06-29', NULL, 'Pedido regular mensal', 10);

INSERT INTO public.itens\_pedido\_compra (pedido\_compra\_id, numero\_item, quantidade, valor\_unitario, valor\_total\_item, status, insumo\_id)

VALUES

(1, 1, 5, 300.00, 1500.00, 'Atendido', 1),

(1, 2, 2, 100.00, 200.00, 'Atendido', 2),

(2, 1, 4, 700.00, 2800.00, 'Aberto', 3),

(2, 2, 1, 10.50, 10.50, 'Atendido', 4),

(3, 1, 10, 50.00, 500.00, 'Atendido', 5),

(4, 1, 3, 400.00, 1200.00, 'Aberto', 6),

(4, 2, 1, 25.75, 25.75, 'Atendido', 7),

(5, 1, 7, 500.00, 3500.00, 'Aberto', 8),

(6, 1, 2, 350.00, 700.00, 'Atendido', 9),

(6, 2, 1, 115.25, 115.25, 'Cancelado', 10),

(7, 1, 10, 300.00, 3000.00, 'Aberto', 1),

(8, 1, 6, 100.00, 600.00, 'Atendido', 2),

(9, 1, 5, 700.00, 3500.00, 'Cancelado', 3),

(10, 1, 2, 10.50, 21.00, 'Atendido', 4),

(10, 2, 1, 50.00, 50.00, 'Aberto', 5);

INSERT INTO public.nf\_entrada (numero\_nf\_entrada, serie\_nf, valor\_total\_nota, data\_emissao, data\_cadastro, fornecedor\_id)

VALUES

('NF00000001', 'S001', 1700.00, '2024-06-25', CURRENT\_DATE, 1),

('NF00000002', 'S012', 10.50, CURRENT\_DATE, CURRENT\_DATE, 2),

('NF00000003', 'S015', 500.00, '2024-06-26', CURRENT\_DATE, 3),

('NF00000004', 'S007', 25.75, CURRENT\_DATE, CURRENT\_DATE, 4),

('NF00000005', 'S002', 3500.00, CURRENT\_DATE, CURRENT\_DATE, 5),

('NF00000006', 'S014', 700.00, '2024-06-28', CURRENT\_DATE, 6),

('NF00000007', 'S006', 3000.00, CURRENT\_DATE, CURRENT\_DATE, 7),

('NF00000008', 'S005', 600.00, '2024-07-02', CURRENT\_DATE, 8),

('NF00000010', 'S003', 21.00, CURRENT\_DATE, CURRENT\_DATE, 10);

INSERT INTO public.itens\_nf\_entrada (numero\_nf\_entrada, numero\_item, quantidade, valor\_unitario, valor\_total\_item, codigo\_lote\_insumo, data\_validade, insumo\_id)

VALUES

('NF00000001', 1, 5, 300.00, 1500.00, 'L12345', '2024-06-30', 1),

('NF00000001', 2, 2, 100.00, 200.00, 'L23456', '2023-12-31', 2),

('NF00000002', 2, 1, 10.50, 10.50, 'L45678', '2022-07-15', 4),

('NF00000003', 1, 10, 50.00, 500.00, 'L56789', '2023-09-30', 5),

('NF00000004', 2, 1, 25.75, 25.75, 'L78901', '2023-08-10', 7),

('NF00000005', 1, 7, 500.00, 3500.00, 'L89012', '2024-11-28', 8),

('NF00000006', 1, 2, 350.00, 700.00, 'L90123', '2023-10-05', 9),

('NF00000007', 1, 10, 300.00, 3000.00, 'L12345', '2023-07-20', 1),

('NF00000008', 1, 6, 100.00, 600.00, 'L23451', '2024-02-15', 2),

('NF00000010', 1, 2, 10.50, 21.00, 'L45678', '2024-03-01', 4);

INSERT INTO public.estoque\_insumo(insumo\_id, codigo\_lote\_insumo, data\_validade,quantidade)

VALUES

(1, 'L12345', '2024-10-05', 12),

(2, 'L23456', '2024-12-15', 2),

(2, 'L23451', '2024-09-25', 6),

(4, 'L45678', '2024-11-07', 1),

(5, 'L56789', '2024-10-23', 2),

(7, 'L78901', '2024-09-17', 1),

(8, 'L89012', '2024-11-30', 2),

(9, 'L90123', '2024-12-14', 2);

INSERT INTO public.requisicao\_insumo (requisicao\_id, data\_requisicao, data\_conclusao, usuario\_id)

VALUES

(1, '2023-01-15', NULL, 1),

(2, '2023-02-20', '2023-02-25', 2),

(3, '2023-03-10', NULL, 3),

(4, '2023-04-05', NULL, 4),

(5, '2023-05-12', '2023-05-15', 5),

(6, '2023-06-08', NULL, 6),

(7, '2023-07-22', '2023-07-25', 7),

(8, '2023-08-17', NULL, 8),

(9, '2023-09-30', NULL, 9),

(10, '2023-10-25', '2023-10-28', 10);

INSERT INTO public.itens\_requisicao\_insumo (requisicao\_id, insumo\_id, codigo\_lote\_insumo, quantidade, status)

VALUES

(1, 1, 'L12345', 3, 'Atendido'),

(1, 2, 'L23456', 1, 'Aberto'),

(2, 4, 'L45678', 1, 'Atendido'),

(3, 5, 'L56789', 8, 'Aberto'),

(4, 7, 'L78901', 1, 'Aberto'),

(5, 8, 'L89012', 5, 'Atendido'),

(6, 9, 'L90123', 2, 'Aberto'),

(7, 1, 'L12345', 5, 'Cancelado');

**Views**

CREATE OR REPLACE VIEW vw\_quantidade\_entrada\_insumos AS

SELECT

    insumos.nome\_insumo,

    COALESCE(SUM(ipc.quantidade), 0)  AS quantidade\_entrada,

    COALESCE(SUM(ipc.valor\_total\_item), 0)  AS valor\_total\_insumo

FROM

    itens\_pedido\_compra as ipc

    INNER JOIN insumos ON insumos.insumo\_id = ipc.insumo\_id

GROUP BY

    ipc.insumo\_id, insumos.nome\_insumo

ORDER BY

    COALESCE(SUM(ipc.valor\_total\_item), 0) DESC;

select \* from vw\_quantidade\_entrada\_insumos;

CREATE OR REPLACE VIEW vw\_quantidade\_valor\_insumos\_fornecedor AS

SELECT

    f.nome\_fornecedor,

    SUM(i.quantidade) AS quantidade\_total,

    SUM(i.valor\_total\_item) AS valor\_total

FROM

    fornecedores f

    INNER JOIN nf\_entrada nf ON f.fornecedor\_id = nf.fornecedor\_id

    INNER JOIN itens\_nf\_entrada i ON nf.numero\_nf\_entrada = i.numero\_nf\_entrada

GROUP BY

    f.nome\_fornecedor

ORDER BY

    valor\_total DESC;

select \* from vw\_quantidade\_valor\_insumos\_fornecedor;

CREATE OR REPLACE VIEW vw\_quantidade\_saida\_insumos AS

SELECT

    ins.nome\_insumo,

    COALESCE(SUM(iri.quantidade), 0) AS quantidade\_saida,

    COALESCE(SUM(iri.quantidade \* inf.valor\_unitario), 0) AS valor\_total\_insumo

FROM

    itens\_requisicao\_insumo iri

    INNER JOIN insumos ins ON ins.insumo\_id = iri.insumo\_id

    LEFT JOIN (

        SELECT

            inf.codigo\_lote\_insumo,

            inf.valor\_unitario

        FROM

            itens\_nf\_entrada inf

        GROUP BY

            inf.codigo\_lote\_insumo, inf.valor\_unitario

    ) AS inf ON inf.codigo\_lote\_insumo = iri.codigo\_lote\_insumo

GROUP BY

    ins.nome\_insumo

ORDER BY

    valor\_total\_insumo DESC;

select \* from vw\_quantidade\_saida\_insumos;

**Functions**

CREATE OR REPLACE FUNCTION alterar\_status\_item\_requisicao\_para\_atendido(id\_requisicao integer,

                                                                   id\_insumo integer,

                                                                   lote\_insumo "varchar")

RETURNS VOID AS $$

DECLARE status\_item varchar(10);

DECLARE quantidade\_item float;

BEGIN

    SELECT status INTO status\_item FROM itens\_requisicao\_insumo

    WHERE requisicao\_id = id\_requisicao AND insumo\_id = id\_insumo AND codigo\_lote\_insumo = lote\_insumo;

    if (status\_item = 'Aberto') then

        UPDATE itens\_requisicao\_insumo SET status = 'Atendido'

        WHERE requisicao\_id = id\_requisicao AND insumo\_id = id\_insumo AND codigo\_lote\_insumo = lote\_insumo;

        SELECT quantidade INTO quantidade\_item FROM itens\_requisicao\_insumo

        WHERE requisicao\_id = id\_requisicao AND insumo\_id = id\_insumo AND codigo\_lote\_insumo = lote\_insumo;

        UPDATE estoque\_insumo SET quantidade = quantidade - quantidade\_item

        WHERE insumo\_id = id\_insumo AND codigo\_lote\_insumo = lote\_insumo;

    end if;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION alterar\_status\_item\_pedido\_compra\_para\_atendido(id\_pedido\_compra integer,

                                                                   item\_numero integer,

                                                                   lote\_insumo "varchar",

                                                                   data\_validade date)

RETURNS VOID AS $$

DECLARE status\_item varchar(10);

DECLARE quantidade\_item float;

DECLARE id\_insumo integer;

DECLARE quantidade\_estoque float;

BEGIN

    SELECT status INTO status\_item FROM itens\_pedido\_compra

    WHERE pedido\_compra\_id = id\_pedido\_compra AND numero\_item = item\_numero;

    if (status\_item = 'Aberto') then

        UPDATE itens\_pedido\_compra SET status = 'Atendido'

        WHERE pedido\_compra\_id = id\_pedido\_compra AND numero\_item = item\_numero;

        SELECT quantidade INTO quantidade\_item FROM itens\_pedido\_compra

        WHERE pedido\_compra\_id = id\_pedido\_compra AND numero\_item = item\_numero;

        SELECT insumo\_id INTO id\_insumo FROM itens\_pedido\_compra

        WHERE pedido\_compra\_id = id\_pedido\_compra AND numero\_item = item\_numero;

        SELECT quantidade INTO quantidade\_estoque FROM estoque\_insumo

        WHERE insumo\_id = id\_insumo and codigo\_lote\_insumo = lote\_insumo;

        if (quantidade\_estoque > 0) then

            UPDATE estoque\_insumo SET quantidade = quantidade + quantidade\_item

            WHERE insumo\_id = id\_insumo AND codigo\_lote\_insumo = lote\_insumo;

        else

            INSERT INTO estoque\_insumo VALUES (id\_insumo, lote\_insumo, data\_validade, quantidade\_item);

        end if;

        PERFORM concluir\_pedido(id\_pedido\_compra);

    end if;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION concluir\_pedido(id\_pedido\_compra integer)

RETURNS VOID AS $$

DECLARE

    tem\_item\_aberto boolean;

BEGIN

    SELECT EXISTS (

        SELECT 1

        FROM itens\_pedido\_compra

        WHERE pedido\_compra\_id = id\_pedido\_compra

          AND status = 'Aberto'

    ) INTO tem\_item\_aberto;

    if (NOT tem\_item\_aberto) then

        UPDATE pedido\_compra

        SET data\_conclusao = current\_date

        WHERE pedido\_compra\_id = id\_pedido\_compra;

    end if;

END;

$$ LANGUAGE plpgsql;

**Triggers**

CREATE OR REPLACE FUNCTION atualizar\_quantidade\_insumos()

RETURNS TRIGGER AS $$

BEGIN

    UPDATE insumos SET quantidade\_atual = quantidade\_atual + NEW.quantidade

    WHERE insumo\_id = NEW.insumo\_id;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_atualizar\_quantidade\_insumos

AFTER INSERT ON estoque\_insumo

FOR EACH ROW

EXECUTE FUNCTION atualizar\_quantidade\_insumos();

**Consultas realizadas baseadas em regras de negócio**

-- 1) Qual o fornecedor que provisionou a maior variedade de insumos?

SELECT fo.fornecedor\_id,

    fo.nome\_fornecedor,

    COUNT(DISTINCT inf.insumo\_id) as quantidade\_itens

    FROM nf\_entrada AS nf

    INNER JOIN fornecedores AS fo ON fo.fornecedor\_id = nf.fornecedor\_id

    INNER JOIN itens\_nf\_entrada AS inf ON inf.numero\_nf\_entrada = nf.numero\_nf\_entrada

    GROUP BY fo.fornecedor\_id, fo.nome\_fornecedor

    ORDER BY quantidade\_itens DESC;

-- 2) Qual foi o insumo menos atendido dos pedidos de compras?

SELECT ins.insumo\_id,

    ins.nome\_insumo,

    SUM(ipc.quantidade) as total

    FROM itens\_pedido\_compra as ipc

    INNER JOIN insumos as ins ON ins.insumo\_id = ipc.insumo\_id

    WHERE ipc.status NOT IN ('Atendido')

    GROUP BY ins.insumo\_id, ins.nome\_insumo, ipc.insumo\_id

    ORDER BY total DESC;

-- 3) Qual usuário faz mais requisições de insumos?

SELECT us.usuario\_id,

    us.nome\_usuario,

    SUM(iri.quantidade) as quantidade\_insumo

    FROM public.requisicao\_insumo AS ri

    INNER JOIN usuarios as us ON us.usuario\_id = ri.usuario\_id

    INNER JOIN itens\_requisicao\_insumo as iri ON iri.requisicao\_id = ri.requisicao\_id

    GROUP BY us.usuario\_id

    ORDER BY SUM(iri.quantidade) DESC;

-- 4) Teve algum insumo do mesmo lote que foi comprado em notas fiscais diferentes?

SELECT infe.insumo\_id,

    ins.nome\_insumo,

    infe.codigo\_lote\_insumo,

    COUNT(DISTINCT infe.numero\_nf\_entrada) as quantidade\_notas

    FROM itens\_nf\_entrada infe

    INNER JOIN insumos AS ins ON ins.insumo\_id = infe.insumo\_id

    GROUP BY infe.insumo\_id, ins.nome\_insumo, infe.codigo\_lote\_insumo

    HAVING COUNT(DISTINCT infe.numero\_nf\_entrada) > 1

    ORDER BY COUNT(DISTINCT infe.numero\_nf\_entrada) DESC;