

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






https://github.com/rafaelrgaidzinski/cadeia_suprimentos_banco_de_dados




 rafaelfgaidzinski / cadeia_suprimentos_banco_de_dados






Q Type  to search | >_ | + ▾ |  |  |  

<> Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

 cadeia_suprimentos_banco_de_dados Public Pin Unwatch 1 ▾ Fork 0 ▾ Star 0 ▾


 master ▾  1 Branch  0 Tags Q Go to file  Add file ▾ <> Code ▾ About 


 **rafaelfgaidzinski** Adicionado todos os itens do trabalho de banco de dados - cadeia de s...  b17284b · 2 minutes ago  1 Commit


 Apresentação	Adicionado todos os itens do trabalho de banco de dados - ...	2 minutes ago
 Backup	Adicionado todos os itens do trabalho de banco de dados - ...	2 minutes ago
 Dicionário de Dados	Adicionado todos os itens do trabalho de banco de dados - ...	2 minutes ago
 Modelo Físico	Adicionado todos os itens do trabalho de banco de dados - ...	2 minutes ago
 Scripts	Adicionado todos os itens do trabalho de banco de dados - ...	2 minutes ago


About

Trabalho final da matéria de banco de dados II para modelar um banco de dados de uma cadeia de suprimentos (Almoxarifado / Compras)

 Activity

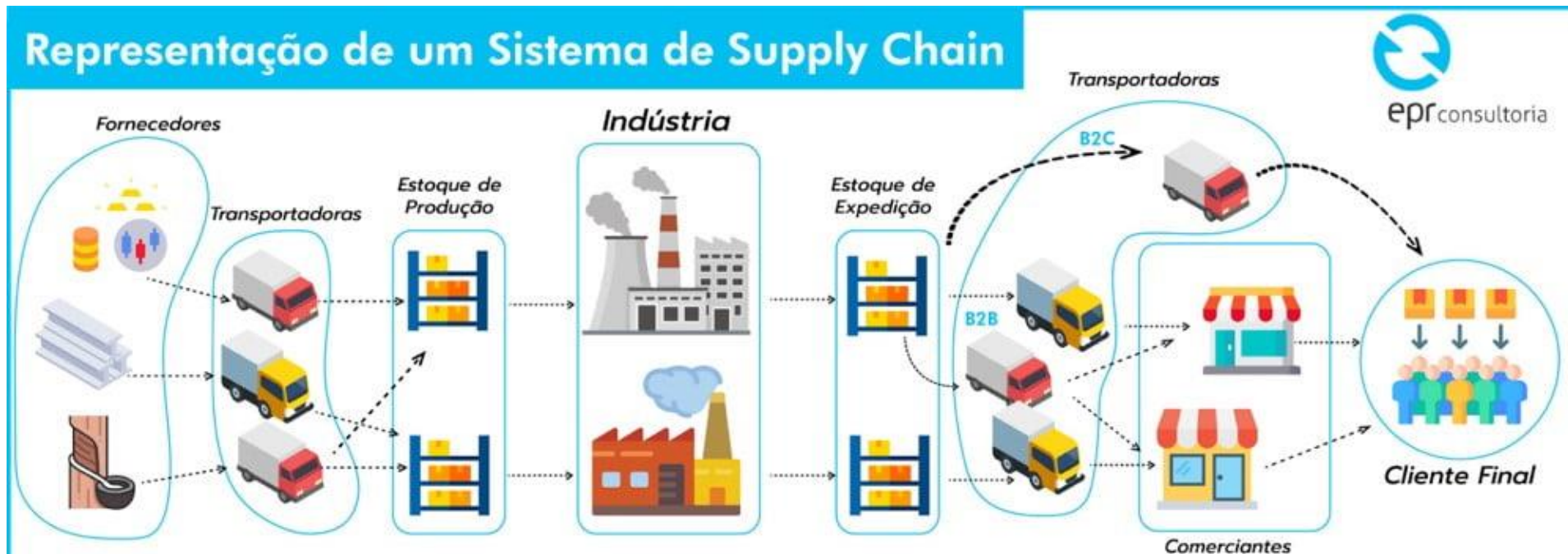
 0 stars

 1 watching

 0 forks

Releases

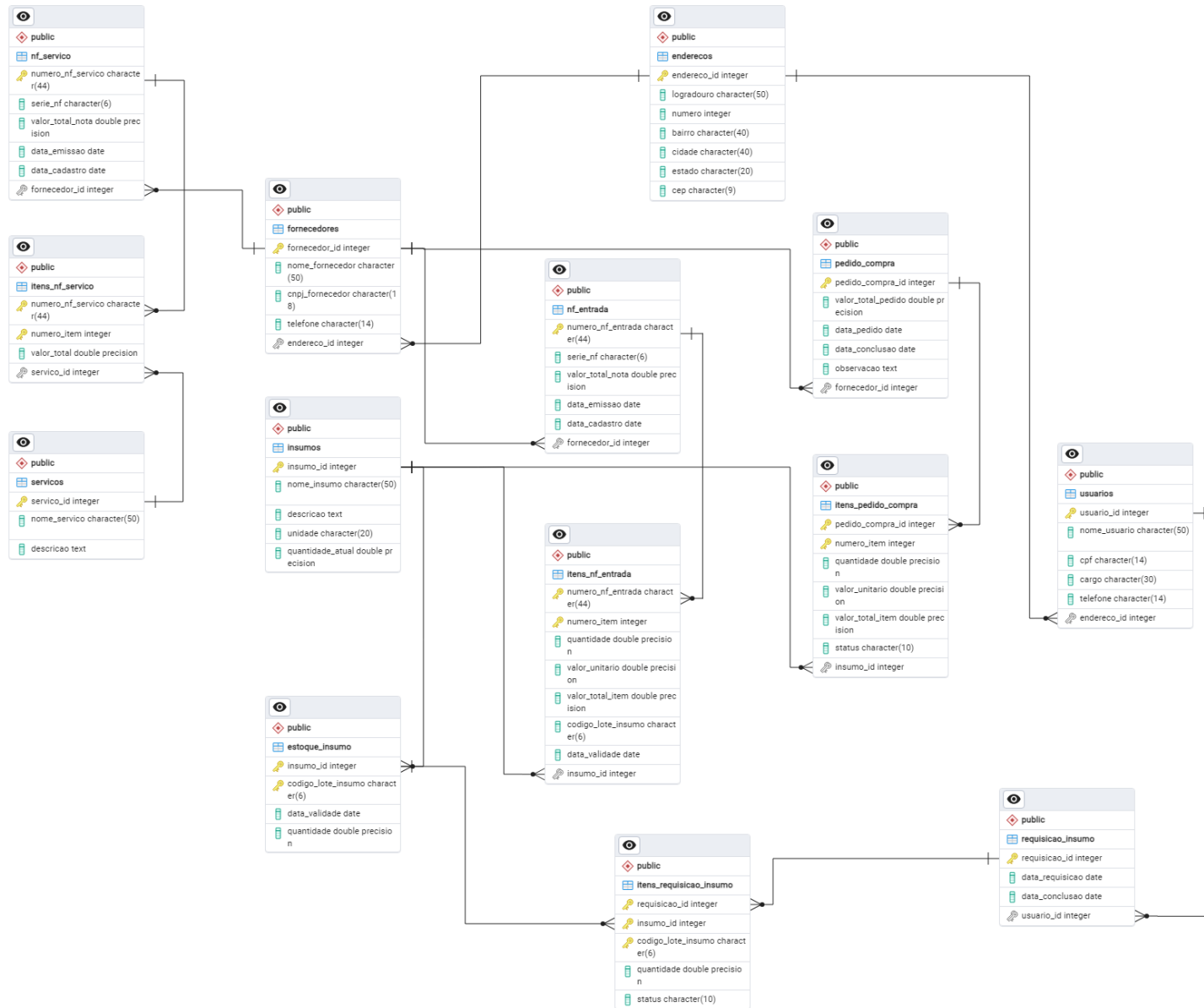
Introdução ao Tema Cadeia de Suprimentos



Ciclo do Setor de Compras



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	Serviços					
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;
```