#AdegaLovers

Adega Lovers

Módulo Estoque

Projeto Integrador I

Versão<2.3>

BRENO JUAN DA FONSECA

JULIANA DE FATIMA PINTO GAMA

YASMIN APARECIDA SILVA DE SOUZA

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 17/08/2021 | 0.0 | Emissão Inicial | Grupo 2 |
| 18/08/2021 | 0.1 | Revisão 0.1 | Ana Paula |
| 07/09/2021 | 1.0 | Correção | Grupo 2 |
| 10/09/2021 | 2.0 | Revisão 0.1 | Ana Paula |
| 27/09/2021 | 2.1 | Correção | Grupo 2 |
| 09/11/2021 | 2.2 | Correção | Grupo 2 |
| 13/12/2021 | 2.3 | Entrega Final | Grupo 2 |

Índice Analítico

1. Introdução 4

1.1 Referências 4

2. Posicionamento 4

2.1 Descrição do Problema 4

2.2 Sentença de Posição do Produto 4

3. Descrições dos Envolvidos e Usuários 4

3.1 Resumo dos Envolvidos 5

3.2 Resumo dos Usuários 5

4. Visão Geral do Produto 5

5. Posicionamento da Estrutura Organizacional 5

6. Planejamento Estratégico 5

7. Definição dos Requisitos 5

8. Modelagem do Processo de Negócio 6

9. Modelo Entidade Relacionamento 6

10. Modelo Lógico-Relacional 6

11. Modelo Físico Construção Banco de Dados Relacional 6

12. Implementação de Consultas SQL 6

13. Extração de Dados Operacionais com Phyton 6

14. Extração de Dados Consolidados com Power BI 6

Documento do Projeto Integrador I

# Introdução

Pretende-se a partir desse projeto, analisar as condições operacionais e estratégicas da Adega Almeida, com o propósito de desenvolver ferramentas que possibilitem a melhor gestão das atividades de estoque.

Identificamos na Adega Almeida a inexistência de um banco de dados informatizado, pois a mesma tem sua operação ainda não informatizada desde sua abertura, deste modo não colhendo dados de entrada e saída de mercadorias, por exemplo, além de não contar com um catálogo de produtos também. Percebemos que com as atividades no formato atual (não-informatizadas) o proprietário não utiliza em sua amplitude todas as capacidades do negócio, pois não consegue mensurar os itens mais vendidos e menos vendidos, essa lacuna gera um impacto significativo em relação ao giro de estoque, que pode se tornar insuficiente, assim criando uma série de problemas operacionais como a armazenagem inadequada de produtos e também insuficiente, a reação em cadeia desse problema é claro, na área de vendas que consideravelmente sofrerá com a falta de produtos, o excesso de outros e a falha de comunicação pela inexistência de um banco de dados.

Com esse panorama, compreendemos que a criação e implementação de um banco de dados na Adega Almeida, pode gerar uma melhor gestão de estoque e dos produtos armazenados, e com a geração de relatórios uma melhor compreensão de quais decisões são as melhores para o negócio.

Este documento tem como objetivo identificar o estado atual da Adega Almeida sob o ponto de vista de negócio para a elaboração de uma proposta de solução para o estoque da adega contemplando a modelagem de dados, inserção de dados, consultas e relatórios.

**1.1 Referências**

MAXIMIANO, A. C. A. **Administração para empreendedores:** fundamentos da criação e da gestão de novos negócios. 2ª ed. São Paulo: Pearson Prentice Hall, 2011.

SERRA, Ana Paula. Projeto Integrador I: Tema e Escopo. *In*: **Projeto Integrador I - Tema e Escopo.pdf**. [*S. l.*], 22 set. 2021. Disponível em: file:///C:/Users/Dell/Downloads/Aula%202%20-%20PI%20-%20Tema%20e%20Escopo.pdf. Acesso em: 22 set. 2021.

CONTA, Azul. Software Conta Azul.Joinville, 2013. Disponível em: <

https://contaazul.com>. Acesso em: 25 de setembro de 2021

FERNANDES, Daniela. Adegas de cerveja são alternativas para o consumidor, São Paulo, 2020. Disponível em <https://www.jj.com.br/jundiai/2020/12/110867-adegas-de-cerveja-sao-alternativas-para-o-consumidor.html>. Acesso em 12 de setembro de 2021.

NAKAGAW, Marcelo. Ferramenta: ANÁLISE SWOT (CLÁSSICO), São Paulo, 2011. Disponível em < www.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/ME\_Analise-Swot.PDF>. Acesso em 22 de setembro de 2021.

ALVARENGA, Darlan. Consumo de cerveja 'migra' para dentro de casa e volume de vendas no Brasil é o maior desde 2014, São Paulo, 2021. Disponível em <https://g1.globo.com/economia/noticia/2021/05/23/consumo-de-cerveja-migra-para-dentro-de-casa-e-volume-de-vendas-no-brasil-e-o-maior-desde-2014.ghtml> Acesso em 15 de setembro de 2021.

# Posicionamento

## Descrição do Problema

A partir da análise de negócio da Adega Almeida, identificou-se que a não informatização do processo de controle de estoque é um problema. O registro de entrada e saída de produtos é feito pelo gestor da adega de forma não estruturada, com uso de papel e caneta e sem uso de ferramentas tecnológicas que possam apoiar o gerenciamento de estoque. Esse problema afeta a eficiência do espaço físico do estoque, que por vezes pode ficar indisponível ou mal aproveitado devido à impossibilidade da análise de um histórico com indicadores que auxiliam na gestão de estoque. A qualidade do que se é vendido também é afetada, uma vez que a informação de validade de produto é verificada apenas no momento da compra com o fornecedor.

Esses problemas impactam nas vendas, no abastecimento de produtos mais pedidos, na relação dos clientes com a loja que por consequência impactam no faturamento da empresa e até mesmo em sua sustentabilidade.

Uma boa solução é a agregação de tecnologias nas quais, além de apoiar no processo de estoque, possam fornecer dados e um histórico para que o gestor da adega possa fazer as melhores decisões no setor de estoque com bases em análise de negócio.

## Sentença de Posição do Produto

A solução proposta nesse trabalho é a criação e implantação de uma aplicação chamada Adega Lovers capaz de tornar o processo de controle de estoque mais eficiente, tecnológico e rápido. É uma solução capaz de armazenar dados por intermédio da implantação e implementação de um banco de dados possibilitando a consulta, extração e análise de dados a partir de ferramentas como SQL, Power Bi e Python.

Adega Lovers é um software de gerenciamento de estoque que possibilita que o gestor da adega, bem como o funcionário responsável pelo gerenciamento de estoque entendam os ciclos de vida do e o mix de seus produtos, otimizando a operação entre demanda e oferta.

Com esse software é possível, por exemplo, prever e planejar os custos de armazenagem física, manutenção, valoração do estoque (quanto vale $$ estoque) e reduzir desperdícios e perdas. Seu gerenciamento de forma eficaz do estoque proporciona clareza no processo de compra, refletindo diretamente nas vendas da adega.

# Descrições dos Envolvidos e Usuários

O principal stakeholder (envolvido) é o proprietário da Adega Almeida - que será denominado como Gestor da Adega. Sua empresa necessita de um sistema de software que torne mais eficiente o controle de estoque.

O software Adega Lovers de gestão de estoque será utilizado pelo Setor de estoque que está organizado em áreas distribuídas da seguinte forma:

* Controle de estoque: Área responsável por controlar toda entrada de produtos entregues pelo fornecedor (serão conferidos baseados na nota fiscal de entrada), controlar saída dos produtos vendidos, executar ajuste de inventário (realizar a conferência das quantidades físicas versus quantidades registradas no sistema) e manter sempre uma quantidade de produtos reserva.

* Controle de Qualidade: Integrada com o controle de estoque, controle de qualidade é responsável por prevenir que a empresa sofra perdas por conta de prazos de validade, furtos internos por parte de funcionários, acompanhar o estado de conservação dos produtos (refrigeração), verificar existência de avarias se a mesma existir abrir uma solicitação de troca (com fornecedor) ou descarte de produtos (direcionar sempre os itens danificados e/ou impróprios para o consumo (por exemplo, com data de validade vencida) para o local de troca).

O software Adega Lovers de gestão de estoque também é utilizado como fonte de informações por outro setor:

Setor de vendas: Área responsável pela saída dos produtos resultantes de venda direta ao consumidor (a movimentação de estoque é automática após a confirmação da venda), caso consumidor exigir troca do produto abrir solicitação de troca (com setor de qualidade o estoque).

## Resumo dos Envolvidos

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Responsabilidades** |
| Alex Naldo | Proprietário da Adega Almeida | Gerenciamento de todo o negócio desde vendas até a administração de todo o fluxo de produtos. |
| Ana Paula | Professora da disciplina projeto integrador I / Coordenadora da equipe | Orientação, avaliação e direcionar os trabalhos realizados pela equipe. |
| BRENO JUAN DA FONSECA  JULIANA DE FATIMA PINTO GAMA  YASMIN APARECIDA SILVA DE SOUZA | Equipe de desenvolvimento | Reunir dados, documentar informações, desenvolver soluções, verificar a aplicabilidade da resolução, testar o desempenho das funcionalidades e implementar o sistema. |

## Resumo dos Usuários

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Proprietário da Adega | O proprietário da adega em questão também é o gestor que pode assumir diversas funções por demanda durante o processo de venda/marketing, |
| Funcionário 01- Auxiliar de estoque/qualidade. | Responsável por controlar toda entrada de produtos entregues pelo fornecedor (serão conferidos baseados na nota fiscal de entrada), controlar saída dos produtos vendidos, executar ajuste de inventário (realizar a conferência das quantidades físicas versus quantidades registradas no sistema) e manter sempre uma quantidade de produtos reserva.  Responsável por prevenir que a empresa sofra perdas por conta de prazos de validade, furtos internos por parte de funcionários, acompanhar o estado de conservação dos produtos (refrigeração), verificar existência de avarias se a mesma existir abrir uma solicitação de troca (com fornecedor) ou descarte de produtos (direcionar sempre os itens danificados e/ou impróprios para o consumo (por exemplo, com data de validade vencida) para o local de troca). |
| Funcionário 02- Auxiliar de vendas. | Responsável pelo controle saída dos produtos resultantes de venda direta ao consumidor (a movimentação de estoque é automática após a confirmação da venda), caso consumidor exigir troca do produto abrir solicitação de troca (com setor de qualidade o estoque). |
| Funcionário 03- Auxiliar de vendas. | Responsável pela saída dos produtos resultantes de venda direta ao consumidor (a movimentação de estoque é automática após a confirmação da venda). |

# Visão Geral do Produto

O software deverá gerenciar as entregas dos fornecedores, baseados na nota fiscal de entrada. Esta funcionalidade permitirá ao operador da área de estoque controlar a quantidade de itens dos produtos e data de validade. Sendo assim ele consultará por período quantos foram entregues e quantos foram vendidos para indicar a necessidade de complemento do mesmo e reposição dos itens à venda.

Após a confirmação da venda (pedido), a movimentação de estoque será automática. Esta funcionalidade permitirá verificar a quantidade de produtos que estão sendo vendidos diariamente, com a atualização em tempo real das vendas a empresa reabastece somente os produtos mais vendidos, evitando compras exageradas e percas devido a prazos de validade, afetando o controle de qualidade da empresa.

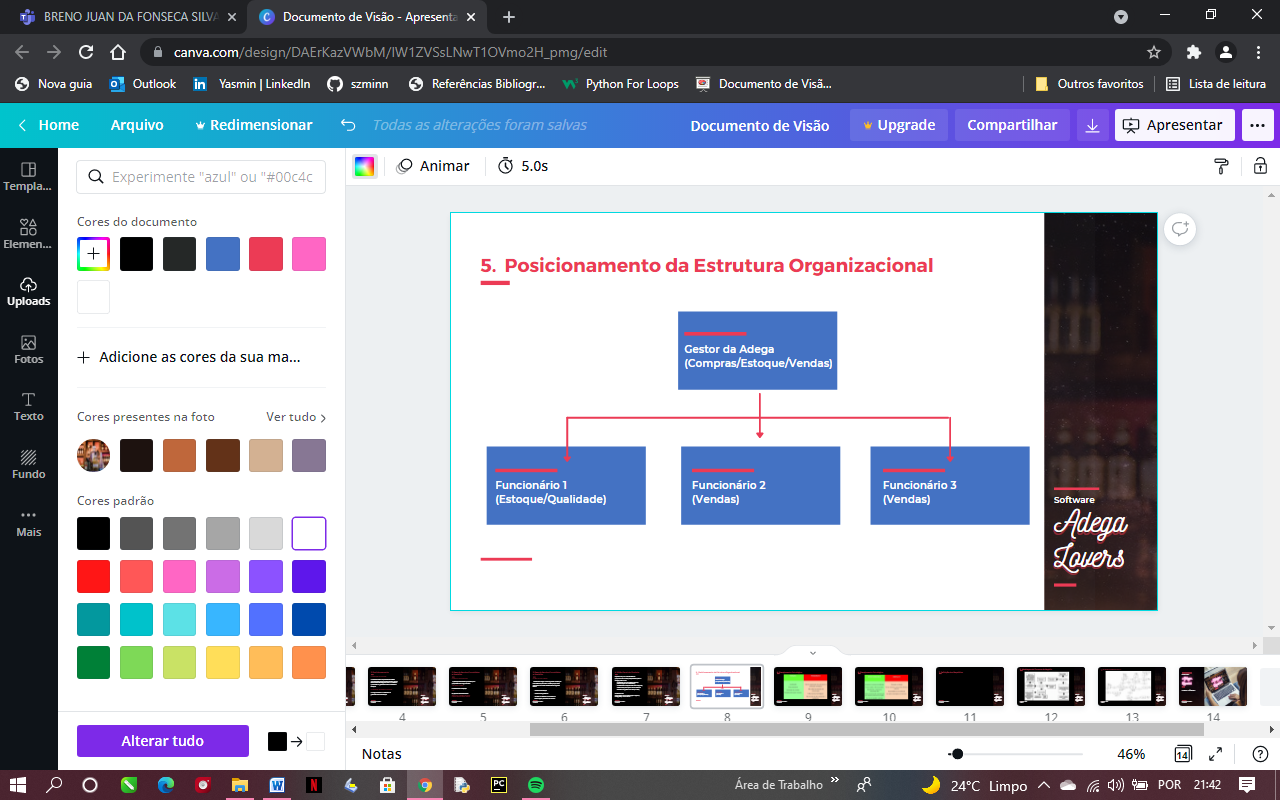
O software a ser desenvolvido auxiliará também no controle das informações geradas pela conferência das quantidades físicas versus quantidades lançadas no banco de dados (Ajuste de Inventário). Essa função visa à melhoria da gestão atual. Com a atualização imediata do banco de dados, o usuário consulta por período quantos foram estocados e quantos foram vendidos, ajudando manter sempre uma quantidade determinada dos produtos para que não fiquem indisponíveis e ocasione em perca da venda.

Espera-se, com os resultados proporcionar novas perspectivas para a Adega, tais como agilidade nos processos diários, a diminuição do retrabalho, acesso fácil e rápido das informações, relatórios para apoio na tomada de decisão do gestor em tempo real, entre outros. Usaremos a linguagem de Programação Phyton para consultar o sistema gerenciador de banco de dados Postgresql, e a ferramenta Power BI para visualização dos relatórios.

# Posicionamento da Estrutura Organizacional

A empresa cujo produto deste trabalho está destinado é do tipo de pequeno porte em que seu proprietário, além de assumir a função de gestor geral, exerce eventualmente funções operacionais, de marketing e vendas, contando com 3 familiares para suporte em determinadas funções. As tarefas, portanto, são dinâmicas e distribuídas entre proprietário da adega, o funcionário 1, 2 e 3. Essa forma de organizar a empresa, é chamada por Maximiano (2011) de “organização por pessoas” e pode ser ilustrada como no exemplo da figura 1.

*Figura 1- Organograma da empresa Adega Almeida*



# Planejamento Estratégico

A matriz SWOT é uma ferramenta de análise estratégica que consiste em avaliar o ambiente interno e externo do negócio e a partir deste cenário embasar a tomada de decisão.

Proporciona conhecimento das características do negócio, desse modo, melhora o planejamento estratégico avaliando a visão do contexto em que o negócio está inserido e do mercado em que o atua.

De acordo com as informações dessa análise, o resultado segue:

|  |  |
| --- | --- |
| FORTES   * Localização * Preço acessível * Variedade de produtos * Produto pronto para consumo | FRACOS   * Não há organização dos processos e orientação estratégica * Não utiliza ferramentas de comunicação * Falta recursos tecnológicos * Não há mão-de-obra suficiente e qualificada. |
| OPORTUNIDADE   * Aumento no consumo de bebidas * Acesso rápido e fácil ao produto * Atendimento a empresas e consumidor final * Ações de promoção e prospecção para clientes potenciais da região * Serviços de pedido e entrega | AMEAÇAS   * Tributação e inflação * Novos entrantes: concorrentes diretos e indiretos; * Sazonalidade e perecibilidade * Inovações estratégicas do mercado. |

As variáveis descritas da figura são justificadas conforme comentários a seguir.

Pontos Fortes:

Localização: localidade com muito fluxo de clientes, não há concorrentes diretos nas proximidades;

Preço acessível: valor de venda acessível em comparação ao mesmo produto, mesma condição em diferentes locais como lojas de conveniência, padarias;

Variedade de produtos: diversidade de itens dos produtos primários (bebidas) e produtos secundários (gelo, aperitivos, itens de mercearia);

Produto pronto para o consumo: oferece opção de bebidas já geladas para consumo imediato.

Pontos Fracos:

· Organização dos processos e estratégia: falta de organização nos controles administrativos de modo geral; não há controle ideal de estoque, possui anotações manuais, o controle é visual e anotações em cadernos; falta orientação estratégica qualificada nos demais departamentos, para suprir toda demanda.

· Não utiliza ferramentas de comunicação: não há divulgação, e-commerce, site ou qualquer meio de publicidade ou propaganda do negócio.

· Falta recursos tecnológicos: todo controle de documentação e dados é feito de forma manual; não há controle eletrônico ou equipamentos eletrônicos disponíveis para controle e gerenciamento dos processos administrativos do negócio.

· Não há mão-de-obra suficiente e qualificada: o quadro de funcionários não é fixo, sendo contratados esporadicamente ajudantes de serviço geral conforme demanda e necessidade para atividades como armazenamento (carga/descarga) e limpeza. Não há pessoal definido para exercer função de acordo com as qualificações necessárias nos setores do negócio (faturamento, vendas, logística etc.), tarefas realizadas principalmente pelo proprietário conforme necessidade.

Oportunidades:

· Aumento no consumo de bebidas: contexto de pandemia aumentou consumo de bebidas alcoólicas e não alcoólicas no geral, pela preferência de consumo em casa, favorecendo o modelo de comércio que o negócio opera;

· Acesso rápido e fácil ao produto: ainda considerando o contexto de pandemia, os consumidores estão optando por não frequentar locais de grande propensão de aglomeração e filas, como supermercados, por exemplo, o que aumenta a procura pelo estabelecimento.

· Ações de promoção e prospecção: negócio tem clientela robusta, mesmo não fazendo propaganda, o que mostra grande potencial de novos clientes com uso de ferramentas de propaganda e prospecção de clientes potenciais, ações de promoção e fidelização de clientes para rotina de compras.

· Serviços de pedida e entrega: com advento de recursos tecnológicos de fácil acesso, como uso de smartphones e acesso a internet, muitas empresas aderiram ao serviço de pedido e entrega online, via e-commerce e aplicativos de celulares, uma ferramenta útil que poderia ser bem aproveitada para alavancar as vendas.

Ameaças:

· Tributação e inflação: há algum no cenário político trata-se a respeito da reforma tributária e no quesito bebias alcoólicas e açucaradas, produtos ultra processados sempre são citados para que sejam elevados os impostos, com objetivo de frear o consumo; impostos elevados e exigências legais rigorosas são pertinentes ao setor, elevando o custo. Inflação e aumento em produtos de consumo e serviços secundários como frete, combustíveis impactam diretamente todo custo operacional, portanto a sustentabilidade e sobrevivência do negócio.

· Sazonalidade e perfectibilidade: produtos com apelo de consumo gelado tem maior procura nos meses de verão, apresentando então maior ciclo de venda nos meses de clima frio, necessário haver planejamento para suprir essa demanda sem ocorrência de perdas.

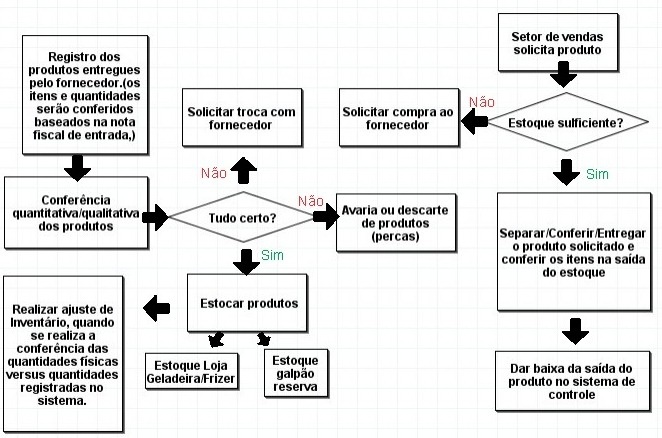
· Novos entrantes: Desenvolver estratégia para proteger sobrevivência do negócio e sempre estar atento a mudanças no cenário. Concorrentes diretos como supermercados e conveniências, mercearias e concorrentes indiretos como os serviços de delivery; trazendo melhor estrutura, conforto e opções para seus clientes.

· Novas estratégias: as empresas concorrentes diretas ou indiretas podem criar/adotar novas estratégias que atendam seu perfil de público-alvo: serviços de entrega, assinatura, promoções.

# Definição dos Requisitos

|  |  |
| --- | --- |
| **Requisitos** | **Estratégia de Implementação (SQL/ Power BI / Python)** |
| 01-Cadastrar produtos. | SQL |
| 02-Permitir entrada de produtos pela Nota Fiscal e cupom fiscal. | SQL |
| 03-Permitir troca de produtos. | SQL |
| 04-Permissão para Inventário. | SQL |
| 05-Gerar relatório de controle de estoque mínimo. | SQL |
| 06-Gerar controle pela data de validade. | SQL |
| 07-Gerar relatório com dados do fornecedor | SQL |
| 08-Conexao Postgres com código Python. | Python |
| 09- Cadastrar de produto | Python |
| 10-Remoção de produtos | Python |
| 11-Atualização de produtos | Python |
| 12-Pesquisa unitária de produtos | Python |
| 13-Pesquisa total de produtos | Python |
| 14-Data de validade dos produtos que vencem em 1 mês | Python |
| 15-Indicadores de poucos produtos no estoque (20 ou menos) | Python |
| 16-Valores dos produtos ao exibir esses dois últimos requisitos | Python |
| 17-Gerar relatório de estoque atual | POWER BI |

# Modelagem do Processo de Negócio



*Figura 2- Modelagem do processo de negócio após implementação do sistema.*

# Modelo Entidade Relacionamento

# 

*Figura 3- Modelo Entidade Relacionamento (brModelo)*

# Modelo Lógico - Esquema de relações

**categoria** (cod\_catg(PK), categoria, cod\_prod(FK), descr\_catg) *cod\_prod referencia produto*

**produto** (cod\_prod(PK), prod\_nome, prod\_descr, cod\_catg(FK), prod\_medida, qtd\_medida, estoque\_min, estoque\_max, prod\_marca)*cod\_catg referencia categoria*

**fornecedor** (cod\_forn(PK), forn\_tipo, forn\_email, forn\_fone, forn\_end, forn\_nome)

**estoque** (id\_estoque(PK), cod\_prod(FK), dt\_hora\_controle, qtd\_entrada, qtd\_saida, qtd\_total, end\_estoque, valor\_qtd\_prod)*cod\_prod referencia produto*

**Troca** (id\_troca, cod\_prod, descr\_troca, tipo\_troca qtd\_prod) *cod\_prod referencia produto*

**Descarte** (id\_desc, cod\_prod, descr\_desc, qtd\_prod)*cod\_prod referencia produto*

**Devolucao** (id\_dev, cod\_prod, descr\_devo tipo\_devo, qtd\_prod)*cod\_prod referencia produto*

**pedido** (id\_pedido(PK), num\_notafiscal\_venda, dt\_hora\_pedido, valor\_total)*(cod\_cli) referencia cliente*

**item** (id\_pedido(PK) (FK), cod\_prod(PK) (FK), qtde\_prod, valor\_venda\_uni, num\_notafiscal\_venda)id\_pedido *referencia pedido, cod\_prod referencia produto*

**fornecimento** (id\_fornecto(PK), num\_notafiscal\_compra, cod\_forn(FK), cod\_prod(FK), dt\_hora\_entrada,num\_lote, dt\_validade, qtde\_fornecto, valor\_custo\_uni, valor\_total)cod\_forn *referencia fornecedor, cod\_prod referencia produto*

# Construção Banco de Dados Relacional

SET DATESTYLE TO postgres, DMY ;

**-- Tabela Categoria**

DROP TABLE IF EXISTS categoria CASCADE;

CREATE TABLE categoria (

cod\_catg INTEGER PRIMARY KEY,

categoria VARCHAR (50) NOT NULL CHECK (categoria IN ('Bebida não alcoólica','Bebida alcoólica')),

descr\_catg VARCHAR(200));

**-- Tabela Produto**

DROP TABLE IF EXISTS produto CASCADE;

CREATE TABLE produto (

cod\_prod INT PRIMARY KEY,

prod\_nome VARCHAR(30),

prod\_descr VARCHAR(50),

cod\_catg INT, FOREIGN KEY (cod\_catg) REFERENCES categoria,

qtd\_medida NUMERIC(3,1),

prod\_medida VARCHAR(10),CHECK (prod\_medida IN ('litro', 'litros', 'ml')),

prod\_marca VARCHAR (100));

**-- Tabela Fornecedor**

DROP TABLE IF EXISTS fornecedor CASCADE;

CREATE TABLE fornecedor (

cod\_forn INT PRIMARY KEY,

forn\_tipo CHAR(2) NOT NULL CHECK (forn\_tipo IN ('PJ', 'PF')),

forn\_email VARCHAR(30),

forn\_fone NUMERIC(11),

forn\_end VARCHAR(100) ,

forn\_nome VARCHAR(100) NOT NULL);

**-- Tabela Fornecimento**

DROP TABLE IF EXISTS fornecimento CASCADE;

CREATE TABLE fornecimento (

id\_fornecto INT PRIMARY KEY,

num\_notafiscal\_compra NUMERIC(50),

cod\_forn INTEGER, FOREIGN KEY (cod\_forn) REFERENCES fornecedor,

cod\_prod INTEGER, FOREIGN KEY (cod\_prod) REFERENCES produto,

dt\_entrada DATE,

num\_lote VARCHAR (30),

dt\_validade DATE,

qtd\_prod NUMERIC(7,2),

valor\_custo\_uni NUMERIC(7,2),

valor\_total NUMERIC(7,2)

);

**-- Tabela Estoque**

DROP TABLE IF EXISTS estoque CASCADE;

CREATE TABLE estoque (

id\_estoque INT PRIMARY KEY,

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto ,

dt\_hora\_controle TIMESTAMP NOT NULL,

qtd\_entrada INTEGER NOT NULL,

qtd\_saida INTEGER NOT NULL,

qtd\_total INTEGER NOT NULL,

end\_estoque VARCHAR(100) NOT NULL,

estoque\_min integer,

estoque\_max integer

);

**-- Tabela Pedido**

DROP TABLE IF EXISTS pedido CASCADE;

CREATE TABLE pedido (

id\_pedido INT PRIMARY KEY,

num\_notafiscal\_venda CHAR(50),

dt\_hora\_pedido TIMESTAMP NOT NULL,

valor\_total NUMERIC(7,2));

**-- Tabela Item**

DROP TABLE IF EXISTS item CASCADE;

CREATE TABLE item (

id\_item integer primary key,

id\_pedido INTEGER, FOREIGN KEY (id\_pedido) REFERENCES pedido ,

cod\_prod INTEGER,FOREIGN KEY (cod\_prod) REFERENCES produto ,

quantidade NUMERIC(7,2),

valor\_venda\_uni NUMERIC(7,2),

num\_notafiscal\_venda NUMERIC(7,2));

**-- Tabela de Troca**

DROP TABLE IF EXISTS troca CASCADE;

CREATE TABLE troca (

id\_troca INT PRIMARY KEY,

qtd\_prod NUMERIC(7,2),

descr\_troca VARCHAR(100),

tipo\_troca CHAR (4) CHECK( tipo\_troca IN ('forn', 'cli')),

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto );

**-- Tabela de Descarte**

DROP TABLE IF EXISTS descarte CASCADE;

CREATE TABLE descarte (

id\_desc INT PRIMARY KEY,

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto,

descr\_desc VARCHAR(50),

qtd\_prod NUMERIC(7,2));

**-- Tabela de Devolucao**

DROP TABLE IF EXISTS devolucao CASCADE;

CREATE TABLE devolucao (

id\_dev INT PRIMARY KEY,

cod\_prod INT , FOREIGN KEY (cod\_prod) REFERENCES produto ,

tipo\_devo CHAR NOT NULL CHECK (tipo\_devo IN ('forn', 'cli')),

descr\_devo VARCHAR(50),

qtd\_prod NUMERIC(7,2)

);

**-- População tabela Categoria**

INSERT INTO categoria VALUES ( 01, 'Bebida não alcoólica', 'Água mineral,Água mineral com gás, Água de coco, Água tônica, Chá Gelado, Energético, Groselha, Isotônico, Refrigerante, Suco e Cerveja sem alcool.') ;

INSERT INTO categoria VALUES ( 02, 'Bebida alcoólica','Absinto, Cachaça, Gim, Rum, Whisky, Vodka, Cerveja, Vinho, Champanhe, Cidra, Espumante, Conhaque, Saquê, Tequila, Licor') ;

SELECT \* FROM categoria;

**Requisito 01- Cadastrar produtos. SQL**

**-- População tabela produto (categoria 1)**

ALTER TABLE produto ADD COLUMN prod\_marca VARCHAR (100);

ALTER TABLE produto ALTER COLUMN qtd\_medida TYPE NUMERIC (4,1)

INSERT INTO produto VALUES ( 100,'ÁGUA','ÁGUA MINERAL',1 ,500,'ml','BONAFONT') ;

INSERT INTO produto VALUES ( 101,'ÁGUA','ÁGUA MINERAL COM GÁS',1 ,500, 'ml', 'BONAFONT') ;

INSERT INTO produto VALUES ( 102,'ÁGUA','ÁGUA MINERAL',1 ,1, 'litro','BONAFONT') ;

INSERT INTO produto VALUES ( 103,'ÁGUA','ÁGUA MINERAL COM GÁS',1,1, 'litro', 'BONAFONT') ;

INSERT INTO produto VALUES ( 104,'ÁGUA DE COCO','ÁGUA DE COCO',1 ,200, 'ml','DU COCO') ;

INSERT INTO produto VALUES ( 105,'ÁGUA DE COCO','ÁGUA DE COCO',1 ,1, 'litro', 'DU COCO') ;

INSERT INTO produto VALUES ( 106,'ÁGUA TÔNICA','ÁGUA TÔNICA',1,350,'ml', 'ANTARCTICA') ;

INSERT INTO produto VALUES ( 107,'ÁGUA TÔNICA','ÁGUA TÔNICA',1,350 ,'ml' 'Schweppes') ;

INSERT INTO produto VALUES ( 108,'CHÁ GELADO','SABOR LIMÃO',1,450 ,'ml', 'CHÁ ICE TEA LEÃO') ;

INSERT INTO produto VALUES ( 109,'CHÁ GELADO','SABOR MATTE',1, 450, 'ml', 'CHÁ ICE TEA LEÃO') ;

INSERT INTO produto VALUES ( 110, 'CHÁ GELADO','SABOR PESSEGO',1,450 ,'ml', 'CHÁ ICE TEA LEÃO') ;

INSERT INTO produto VALUES ( 111,'ENERGÉTICO','Bebida Energética',1, 250, 'ml', 'RED BULL ENERGY DRINK');

INSERT INTO produto VALUES ( 112, 'ENERGÉTICO','Bebida Energética',1,473,'ml','Monster Energy');

INSERT INTO produto VALUES ( 113, 'ENERGÉTICO','Bebida Energética com Guaraná',1, 1, 'litro', 'Fusion Energy Drink');

INSERT INTO produto VALUES ( 114, 'ENERGÉTICO','Bebida Energética com Guaraná',1, 2, 'litros','Long One');

INSERT INTO produto VALUES ( 115, 'ENERGÉTICO','Bebida Energética com Guaraná',1, 2, 'litros', 'VIBE ENERGY');

INSERT INTO produto VALUES ( 116, 'GROSELHA','Xarope De Groselha',1, 900, 'ml','Milani');

INSERT INTO produto VALUES ( 117,'Isotônico','Isotonico Mix de Frutas',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 118, 'Isotônico','Isotonico Mix de Frutas',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 119, 'Isotônico','Isotonico Sabor Uva',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 120, 'Isotônico','Isotonico Sabor Limão',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 121, 'Isotônico','Isotonico Sabor Morango',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 122, 'Isotônico','Isotonico Sabor Futas Cítricas',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 123, 'Isotônico','Isotonico Sabor Laranja',1, 500, 'ml', 'Powerade');

INSERT INTO produto VALUES ( 124, 'Isotônico','Isotonico Sabor Maracujá',1, 500, 'ml', 'Gatorade');

INSERT INTO produto VALUES ( 125, 'Isotônico','Isotonico Sabor Tangerina',1, 500, 'ml','Gatorade');

INSERT INTO produto VALUES ( 126, 'Isotônico','Isotonico Sabor Uva',1, 500, 'ml', 'Gatorade');

INSERT INTO produto VALUES ( 127, 'Isotônico','Isotonico Sabor Limão',1, 500, 'ml','Gatorade');

INSERT INTO produto VALUES ( 128, 'Isotônico','Isotonico Sabor Morango',1, 500, 'ml', 'Gatorade');

INSERT INTO produto VALUES ( 129, 'Isotônico','Isotonico Sabor Futas Cítricas',1, 500, 'ml','Gatorade');

INSERT INTO produto VALUES ( 130, 'Isotônico','Isotonico Sabor Maracujá',1, 500, 'ml', 'Gatorade');

INSERT INTO produto VALUES ( 131, 'Isotônico','Isotonico Sabor Laranja',1, 500, 'ml', 'Gatorade');

INSERT INTO produto VALUES ( 132, 'Refrigerante Lata','Sabor Cola',1, 350, 'ml','Coca-cola');

INSERT INTO produto VALUES ( 133, 'Refrigerante Lata','Sabor Cola',1, 220, 'ml', 'Coca-cola'); ;

INSERT INTO produto VALUES ( 134, 'Refrigerante garrafa','Sabor Cola',1, 200, 'ml', 'Coca-cola');

INSERT INTO produto VALUES ( 135, 'Refrigerante garrafa','Sabor Cola',1, 1.5, 'litro', 'Coca-cola');

INSERT INTO produto VALUES ( 136, 'Refrigerante garrafa','Sabor Cola',1, 2, 'litros', 'Coca-cola');

INSERT INTO produto VALUES ( 137, 'Refrigerante garrafa','Sabor Cola',1, 600, 'ml', 'Coca-cola');

INSERT INTO produto VALUES ( 138, 'Refrigerante Lata','Sabor Laranja',1, 350, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 139, 'Refrigerante Lata','Sabor Laranja',1, 220, 'ml', 'Fanta'); ;

INSERT INTO produto VALUES ( 140, 'Refrigerante garrafa','Sabor Laranja',1, 200, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 141, 'Refrigerante garrafa','Sabor Laranja',1, 1.5, 'litro', 'Fanta');

INSERT INTO produto VALUES ( 142, 'Refrigerante garrafa','Sabor Laranja',1, 2, 'litros','Fanta');

INSERT INTO produto VALUES ( 144,'Refrigerante garrafa','Sabor Laranja',1, 600, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 145,'Refrigerante Lata','Sabor Uva',1, 350, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 146,'Refrigerante Lata','Sabor Uva',1, 220, 'ml', 'Fanta'); ;

INSERT INTO produto VALUES ( 147,'Refrigerante garrafa','Sabor Uva',1, 200, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 148,'Refrigerante garrafa','Sabor Uva',1, 1.5, 'litro', 'Fanta');

INSERT INTO produto VALUES ( 149,'Refrigerante garrafa','Sabor Uva',1, 2, 'litros', 'Fanta');

INSERT INTO produto VALUES ( 150,'Refrigerante garrafa','Sabor Uva',1, 600, 'ml', 'Fanta');

INSERT INTO produto VALUES ( 151,'Refrigerante Lata','Sabor Guaraná',1, 350, 'ml', ' Fanta ');

INSERT INTO produto VALUES ( 152,'Refrigerante Lata','Sabor Guaraná',1, 220, 'ml', 'Fanta'); ;

INSERT INTO produto VALUES ( 153,'Refrigerante garrafa','Sabor Guaraná',1,200,'ml','Fanta');

INSERT INTO produto VALUES ( 154,'Refrigerante garrafa','Sabor Guaraná',1, 1.5, 'litro', 'Fanta');

INSERT INTO produto VALUES ( 155,'Refrigerante garrafa','Sabor Guaraná',1, 2, 'litros','Fanta');

INSERT INTO produto VALUES ( 156,'Refrigerante garrafa','Sabor Guaraná',1, 600, 'ml', ' Antarctica Fanta');

INSERT INTO produto VALUES ( 157,'Refrigerante Lata','Sabor Limão',1, 350, 'ml', 'Sprite');

INSERT INTO produto VALUES ( 158,'Refrigerante Lata','Sabor Limão',1, 220, 'ml', 'Sprite'); ;

INSERT INTO produto VALUES ( 159,'Refrigerante garrafa','Sabor Limão',1, 200, 'ml', 'Sprite');

INSERT INTO produto VALUES ( 160,'Refrigerante garrafa','Sabor Limão',1, 1.5, 'litro','Sprite');

INSERT INTO produto VALUES ( 161,'Refrigerante garrafa','Sabor Limão',1, 2, 'litros', 'Sprite');

INSERT INTO produto VALUES ( 162,'Refrigerante garrafa','Sabor Limão',1, 600, 'ml', 'Sprite');

INSERT INTO produto VALUES ( 163,'Refrigerante garrafa','Sabor Limão',1, 200, 'ml', 'Sprite');

INSERT INTO produto VALUES ( 164,'Refrigerante garrafa','Sabor Limão',1, 1.5, 'litro', 'Sprite');

INSERT INTO produto VALUES ( 165,'Refrigerante garrafa','Sabor Limão',1, 2, 'litros', 'Sprite');

INSERT INTO produto VALUES ( 166,'Refrigerante garrafa','Sabor Limão',1, 600, 'ml', 'Sprite');

INSERT INTO produto VALUES ( 167,'Refrigerante Lata','Sabor Guaraná',1, 350, 'ml', 'Antarctica');

INSERT INTO produto VALUES ( 168,'Refrigerante Lata','Sabor Guaraná',1, 220, 'ml', 'Antarctica'); ;

INSERT INTO produto VALUES ( 169,'Refrigerante garrafa','Sabor Guaraná',1, 200, 'ml', 'Antarctica');

INSERT INTO produto VALUES ( 170,'Refrig.garrafa Retorn','Sabor Laranja',1, 2, 'litros', 'Fanta');

INSERT INTO produto VALUES ( 171,'Refrigerante garrafa','Sabor Limão',1, 1.5, 'litro', 'Antarctica');

INSERT INTO produto VALUES ( 172,'Refrigerante garrafa','Sabor Limão',1, 2, 'litros', 'Antarctica');

INSERT INTO produto VALUES ( 173,'Refrigerante garrafa','Sabor Limão',1, 600, 'ml', 'Antarctica');

INSERT INTO produto VALUES ( 174,'Refrigerante Lata','Sabor Tutti Frutti',1, 350, 'ml', 'Itubaína');

INSERT INTO produto VALUES ( 175,'Refrigerante garrafa', 'Sabor Tutti Frutti',1, 2, 'litro', 'Itubaína'); ;

INSERT INTO produto VALUES ( 176,'Refrigerante garrafa de vidro','Sabor Tutti Frutti',1, 600, 'ml', 'Itubaína');

INSERT INTO produto VALUES ( 177,'Refrig.garrafa Retorn','Sabor Cola',1, 2, 'litros', 'Coca-cola');

INSERT INTO produto VALUES ( 178,'Refrig. gar. de vidro Retorn','Sabor Guaraná',1, 2, 'litros', 'Antarctica');

INSERT INTO produto VALUES ( 179,'Refrig. gar. de vidro Retorn','Sabor Uva',1, 2, 'litros', 'Fanta');

INSERT INTO produto VALUES ( 180,'Refrig. gar. de vidro Retorn','Sabor Cola',1, 2, 'litros', 'Coca-cola');

INSERT INTO produto VALUES ( 181,'Refrigerante garrafa','Sabor Uva',1, 350, 'ml', 'DOLLY');

INSERT INTO produto VALUES ( 182,'Refrigerante garrafa','Sabor Uva',1, 2, 'litros','DOLLY');

INSERT INTO produto VALUES ( 183,'Refrigerante garrafa','Sabor Limão',1, 350, 'ml', 'DOLLY');

INSERT INTO produto VALUES ( 184,'Refrigerante garrafa','Sabor Limão',1, 2, 'litros', 'DOLLY');

INSERT INTO produto VALUES ( 185,'Refrigerante garrafa','Sabor Laranja',1, 350, 'ml', 'DOLLY');

INSERT INTO produto VALUES ( 186,'Refrigerante garrafa','Sabor Laranja',1, 2, 'litros', 'DOLLY');

INSERT INTO produto VALUES ( 187,'Refrigerante garrafa','Sabor Guaraná',1, 350, 'ml', 'DOLLY');

INSERT INTO produto VALUES ( 188,'Refrigerante garrafa','Sabor Guaraná',1, 2, 'litros', 'DOLLY');

INSERT INTO produto VALUES ( 189,'Refrigerante garrafa','Sabor Cola',1, 350, 'ml', 'DOLLY');

INSERT INTO produto VALUES ( 190,'Refrigerante garrafa','Sabor Cola',1, 2, 'litros', 'DOLLY');

INSERT INTO produto VALUES ( 191,'Suco de caixinha','Néctar Uva',1, 200, 'ml', 'Del Valle Kapo');

INSERT INTO produto VALUES ( 192,'Suco de caixa','Néctar Uva',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 193,'Suco de garrafa','Néctar Uva',1, 1.5, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 194,'Suco de garrafa','Néctar Uva',1, 450, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 195,'Suco de Lata','Néctar Uva',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 196,'Suco de caixinha','Néctar Limão',1, 200, 'ml', 'Del Valle Kapo');

INSERT INTO produto VALUES ( 197,'Suco de caixa','Néctar Limão',1, 1, 'litro','Del Valle');

INSERT INTO produto VALUES ( 198,'Suco de garrafa','Néctar Limão',1, 1.5, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 199,'Suco de garrafa','Néctar Limão',1, 450, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 200,'Suco de Lata','Néctar Limão',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 201,'Suco de caixinha','Néctar Morango',1, 200, 'ml', 'Del Valle Kapo');

INSERT INTO produto VALUES ( 202,'Suco de caixa','Néctar Morango',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 203,'Suco de garrafa','Néctar Morango',1, 1.5, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 204,'Suco de garrafa','Néctar Morango',1, 450, 'ml','Del Valle');

INSERT INTO produto VALUES ( 205,'Suco de Lata','Néctar Morango',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 206,'Suco de caixinha','Néctar Laranja',1, 200, 'ml', 'Del Valle Kapo');

INSERT INTO produto VALUES ( 207,'Suco de caixa','Néctar Laranja',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 208,'Suco de garrafa','Néctar Laranja',1, 1.5, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 209,'Suco de garrafa','Néctar Laranja',1, 450, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 210,'Suco de Lata','Néctar Laranja',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 211,'Suco de caixinha','Néctar Abacaxi',1, 200, 'ml', 'Del Valle Kapo');

INSERT INTO produto VALUES ( 212,'Suco de caixa','Néctar Abacaxi',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 213,'Suco de garrafa','Néctar Abacaxi',1, 1.5, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 214,'Suco de garrafa','Néctar Abacaxi',1, 450, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 215,'Suco de Lata','Néctar Abacaxi',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 216,'Suco de caixa','Néctar Pêssego',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 217,'Suco de Lata','Néctar Pêssego',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 218,'Suco de caixa','Néctar Caju',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 219,'Suco de Lata','Néctar Caju',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 220,'Suco de caixa','Néctar Manga',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 221,'Suco de Lata','Néctar Manga',1,290,'ml','Del Valle');

INSERT INTO produto VALUES ( 222,'Suco de caixa','Néctar Maracujá',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 223,'Suco de Lata','Néctar Maracujá',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 224,'Suco de caixa','Néctar Goiaba',1, 1, 'litro', 'Del Valle');

INSERT INTO produto VALUES ( 225,'Suco de Lata','Néctar Goiaba',1, 290, 'ml', 'Del Valle');

INSERT INTO produto VALUES ( 226,'Suco de garrafa','Laranja Integral',1, 900, 'ml', 'Life');

INSERT INTO produto VALUES ( 227,'Suco de garrafa','Laranja Integral',1, 300, 'ml', 'Life');

INSERT INTO produto VALUES ( 228,'Suco de garrafa','Uva Integral',1, 900, 'ml', 'Life');

INSERT INTO produto VALUES ( 229,'Suco de garrafa','Uva Integral',1, 300, 'ml', 'Life');

INSERT INTO produto VALUES ( 230,'Suco de garrafa','Concentrado Uva Integral',1, 1.5, 'litro', 'Life');

INSERT INTO produto VALUES ( 231,'Cerveja Long Neck','sem álcool 0,0%',1, 330, 'ml', 'Heineken');

INSERT INTO produto VALUES ( 232,'Cerveja Long Neck','sem álcool 0,0%',1, 355, 'ml', 'Brahma');

INSERT INTO produto VALUES ( 233,'Cerveja Long Neck','sem álcool 0,0%',1, 250, 'ml', 'Estrella');

INSERT INTO produto VALUES ( 234,'Cerveja Long Neck','sem álcool 0,0%',1, 355, 'ml', 'Itaipava');

INSERT INTO produto VALUES ( 235,'Cerveja Lata','sem álcool 0,0%',1, 350, 'ml', 'Heineken');

INSERT INTO produto VALUES ( 236,'Cerveja Lata','sem álcool 0,0%',1, 350, 'ml', 'Brahma');

INSERT INTO produto VALUES ( 237,'Cerveja Lata','sem álcool 0,0%',1, 330, 'ml', 'Estrella');

INSERT INTO produto VALUES ( 238,'Cerveja Lata','sem álcool 0,0%',1, 350, 'ml', 'Itaipava');

INSERT INTO produto VALUES ( 239,'Cerveja Lata','sem álcool 0,0%',1, 350, 'ml', 'Malta Pilsen');

INSERT INTO produto VALUES ( 240,'Cerveja Lata','sem álcool 0,0%',1, 350, 'ml', 'Colônia Pilsen');

-- CATEGORIA 2

INSERT INTO produto VALUES ( 241,'Cerveja Long Neck','Premium Pilsen Lager',2, 330, 'ml', 'Heineken');

INSERT INTO produto VALUES ( 242,'Cerveja Long Neck','Duplo Malte',2, 355, 'ml', 'Brahma');

INSERT INTO produto VALUES ( 243,'Cerveja Long Neck','Pilsen Lager',2, 250, 'ml', 'Budweiser');

INSERT INTO produto VALUES ( 244,'Cerveja Long Neck','Pilsen Puro Malte',2, 355, 'ml', 'Eisenbahn');

INSERT INTO produto VALUES ( 245,'Cerveja Lata','Premium Pilsen Lager',2, 350, 'ml', 'Heineken');

INSERT INTO produto VALUES ( 246,'Cerveja Lata','Duplo Malte',2, 350, 'ml', 'Brahma');

INSERT INTO produto VALUES ( 247,'Cerveja Lata','Pilsen Puro Malte',2, 330, 'ml', 'Eisenbahn');

INSERT INTO produto VALUES ( 248,'Cerveja Lata','Pilsen Lager',2, 350, 'ml', 'Itaipava');

INSERT INTO produto VALUES ( 249,'Cerveja Lata','Pilsen Lager',2, 350, 'ml', 'Budweiser');

INSERT INTO produto VALUES ( 250,'Cerveja Lata','Pilsen Lager',2, 350, 'ml', 'Amstel');

INSERT INTO produto VALUES ( 251,'Cerveja Lata','Puro Malte',2, 350, 'ml', 'SKOL');

INSERT INTO produto VALUES ( 252,'Cerveja garrafa','Premium Pilsen Lager',2, 600, 'ml', 'Heineken');

INSERT INTO produto VALUES ( 253,'Cerveja garrafa','Duplo Malte',2, 600, 'ml', 'Brahma');

INSERT INTO produto VALUES ( 254,'Cerveja garrafa','Cerveja antarctica original pilsen',2, 600, 'ml', 'Original');

INSERT INTO produto VALUES ( 255,'Cerveja garrafa','Pilsen Puro Malte',2, 600, 'ml', 'Eisenbahn');

INSERT INTO produto VALUES ( 256,'Cerveja garrafa','Pilsen Lager',2, 600, 'ml', 'Skol');

INSERT INTO produto VALUES ( 257,'Cerveja garrafa','Pilsen Lager',2, 600, 'ml', 'Itaipava');

INSERT INTO produto VALUES ( 258,'Cerveja garrafa','Pilsen Lager',2, 1, 'litro', 'Brahma');

INSERT INTO produto VALUES ( 259,'Cerveja garrafa','Pilsen Lager',2, 1, 'litro', 'Itaipava');

INSERT INTO produto VALUES ( 260,'Cerveja garrafa','Pilsen Lager',2, 1, 'litro', 'Skol');

**-- População tabela fornecedor**

INSERT INTO fornecedor VALUES (1,'PJ', 'cdmooca@ambev.com.br ', 08008871111, 'Av. Presidente Wilson, 274 - Mooca, São Paulo - SP, 03107-900', 'AMBEV CDD MOOCA');

INSERT INTO fornecedor VALUES (2,'PJ', 'vendas@femsa.com.br', 1136596356, 'Pc Agrícola La Paz Tristante, 35 - Industrial Anhangüera, 06276-035', 'COCA-COLA FEMSA SP');

INSERT INTO fornecedor VALUES (3,'PJ', 'Vendas@pepsico.com.br', 25359470, 'R. Panambi, 191 - Cidade Industrial Satélite de São Paulo, Guarulhos - SP, 07224-130', 'PEPSICO DO BRASIL GRU');

INSERT INTO fornecedor VALUES (4,'PJ', 'sac@distribuidorahideal.com.br', 26193222, 'Endereço: Rua Alberto Sampaio, 147, São Paulo – SP', 'DISTRIBUIDORA HIDEAL');

INSERT INTO fornecedor VALUES (5,'PJ', 'vendas2@atacadao.com.br', 43903224, 'Av. Morvan Dias de Figueiredo, 6157, São Paulo - SP, 02170-901', 'ATACADAO');

**Requisito 02 - Permitir entrada de produtos pela Nota Fiscal e cupom fiscal. SQL**

**-- População tabela fornecimento**

INSERT INTO fornecimento VALUES ( 5000 , 90000 , 4 , 101 , '15/09/2021' , 56983 , '31/12/2022' , 100 , 0.99 , 99.00 ) ;

INSERT INTO fornecimento VALUES ( 5001 , 90000 , 4 , 103 , '15/09/2021' , 790012 , '31/12/2022' , 100 , 0.99 , 99.00 ) ;

INSERT INTO fornecimento VALUES ( 5002 , 90000 , 4 , 110 , '15/09/2021' , 907652 , '31/12/2022' , 50 , 2.50 , 125.00 ) ;

INSERT INTO fornecimento VALUES ( 5003 , 90000 , 4 , 125 , '15/09/2021' , 38945 , '01/01/2024' , 20 , 0.99 , 19.80 ) ;

INSERT INTO fornecimento VALUES ( 5004 , 90000 , 4 , 122 , '15/09/2021' , 98542 , '02/01/2024' , 30 , 1.99 , 119.40 ) ;

INSERT INTO fornecimento VALUES ( 5005 , 90000 , 4 , 120 , '15/09/2021' , 2398 , '03/01/2024' , 10 , 2.99 , 89.70 ) ;

INSERT INTO fornecimento VALUES ( 5006 , 90000 , 4 , 127 , '15/09/2021' , 37494 , '31/12/2022' , 20 , 3.50 , 280.00 ) ;

INSERT INTO fornecimento VALUES ( 5007 , 90000 , 4 , 107 , '15/09/2021' , 64832 , '31/12/2022' , 100 , 1.50 , 150.00 ) ;

INSERT INTO fornecimento VALUES ( 5008 , 90000 , 4 , 106 , '15/09/2021' , 34937 , '31/12/2023' , 100 , 1.50 , 150.00 ) ;

INSERT INTO fornecimento VALUES ( 5009 , 90000 , 4 , 109 , '15/09/2021' , 58490 , '01/01/2024' , 50 , 2.50 , 125.00 ) ;

INSERT INTO fornecimento VALUES ( 5010 , 90000 , 4 , 140 , '15/09/2021' , 44312 , '02/01/2023' , 50 , 0.99 , 49.50 ) ;

INSERT INTO fornecimento VALUES ( 5011 , 90000 , 4 , 142 , '15/09/2021' , 44315 , '31/12/2022' , 50 , 2.50 , 125.00 ) ;

INSERT INTO fornecimento VALUES ( 5012 , 90001 , 2 , 132 , '22/10/2021' , 6732 , '31/12/2022' , 20 , 1.99 , 39.80 ) ;

INSERT INTO fornecimento VALUES ( 5013 , 90001 , 2 , 130 , '22/10/2021' , 67543 , '31/12/2022' , 10 , 2.50 , 25.00 ) ;

INSERT INTO fornecimento VALUES ( 5014 , 90001 , 2 , 129 , '22/10/2021' , 67351 , '31/12/2022' , 10 , 2.50 , 25.00 ) ;

INSERT INTO fornecimento VALUES ( 5015 , 90001 , 2 , 125 , '22/10/2021' , 67198 , '01/01/2023' , 20 , 2.50 , 50.00 ) ;

INSERT INTO fornecimento VALUES ( 5016 , 90001 , 2 , 212 , '22/10/2021' , 73820 , '01/01/2023' , 20 , 4.00 , 80.00 ) ;

INSERT INTO fornecimento VALUES ( 5017 , 90001 , 2 , 210 , '22/10/2021' , 73855 , '02/01/2023' , 20 , 1.99 , 39.00 ) ;

INSERT INTO fornecimento VALUES ( 5019 , 90001 , 2 , 124 , '22/10/2021' , 67198 , '01/01/2023' , 20 , 2.50 , 50.00 ) ;

INSERT INTO fornecimento VALUES ( 5020 , 90001 , 2 , 122 , '22/10/2021' , 67198 , '01/01/2023' , 20 , 2.50 , 50.00 ) ;

INSERT INTO fornecimento VALUES ( 5021 , 90001 , 2 , 158 , '22/10/2021' , 25509 , '01/01/2023' , 50 , 2.50 , 125.00 ) ;

INSERT INTO fornecimento VALUES ( 5022 , 90001 , 2 , 159 , '22/10/2021' , 2457 , '02/01/2023' , 50 , 2.50 , 125.00 ) ;

INSERT INTO fornecimento VALUES ( 5023 , 90001 , 2 , 154 , '22/10/2021' , 28974 , '31/12/2022' , 20 , 5.99 , 100.00 ) ;

INSERT INTO fornecimento VALUES ( 5024 , 90001 , 2 , 155 , '22/10/2021' , 2300 , '31/12/2022' , 50 , 4.50 , 225.00 ) ;

INSERT INTO fornecimento VALUES ( 5025 , 90001 , 2 , 160 , '22/10/2021' , 1242 , '31/12/2022' , 50 , 4.50 , 225.00 ) ;

INSERT INTO fornecimento VALUES ( 5026 , 90001 , 2 , 168 , '22/10/2021' , 3495 , '31/12/2022' , 20 , 1.50 , 30.00 ) ;

INSERT INTO fornecimento VALUES ( 5027 , 90001, 2, 205, '22/10/2021' , 73855 , '02/01/2023', 20, 1.99, 39.00) ;

INSERT INTO fornecimento VALUES ( 5028 , 90002 , 1 , 260 , '22/11/2021' , 2370 , '31/12/2021' , 48 , 5.60 , 268.80 ) ;

INSERT INTO fornecimento VALUES ( 5029 , 90002 , 1 , 246 , '22/11/2021' , 1282 , '31/12/2021' , 100 , 1.99 , 199.00 ) ;

INSERT INTO fornecimento VALUES ( 5030 , 90002 , 1 , 249 , '22/11/2021' , 3425 , '31/12/2021' , 100 , 1.99 , 199.00 ) ;

INSERT INTO fornecimento VALUES ( 5031 , 90002 , 1 , 258 , '22/11/2021' , 2070 , '31/12/2021' , 48 , 5.60 , 268.80 ) ;

INSERT INTO fornecimento VALUES ( 5032 , 90002 , 1 , 254 , '22/11/2021' , 2170 , '31/12/2021' , 48 , 5.00 , 240.00 ) ;

/\*ID PEDIDO, NUM NOTA, DATA, VALOR T\*/

**-- População tabela pedido e tabela item**

INSERT INTO pedido VALUES ( 1, 100.000, '01-05-2021', 199.9)

INSERT INTO item VALUES ( 300, 1, 124, 10, 3.50, 100.000)

INSERT INTO item VALUES ( 301, 1, 129, 5, 3.50, 100.000)

INSERT INTO item VALUES ( 302, 1, 130, 5, 3.50, 100.000)

INSERT INTO item VALUES ( 303, 1, 132, 10, 2.99, 100.000)

INSERT INTO item VALUES ( 304, 1, 158, 25, 4.00, 100.000)

INSERT INTO pedido VALUES ( 2, 101.000, '01-06-2021', 31.47)

INSERT INTO item VALUES ( 305, 2, 129, 1, 3.50, 101.000)

INSERT INTO item VALUES ( 306, 2, 130, 2, 3.50, 101.000)

INSERT INTO item VALUES ( 307, 2, 132, 3, 2.99, 101.000)

INSERT INTO item VALUES ( 308, 2, 158, 2, 4.00, 101.000)

INSERT INTO item VALUES ( 309, 2, 159, 1, 4.00, 101.000)

INSERT INTO pedido VALUES ( 3, 102.000, '01-07-2021', 19.29)

INSERT INTO item VALUES ( 310, 3, 130, 1, 3.50, 102.000)

INSERT INTO item VALUES ( 311, 3, 132, 1, 2.99, 102.000)

INSERT INTO item VALUES ( 312, 3, 158, 1, 4.00, 102.000)

INSERT INTO item VALUES ( 313, 3, 159, 1, 4.00, 102.000)

INSERT INTO item VALUES ( 314, 3, 160, 1, 6.00, 102.000)

INSERT INTO pedido VALUES ( 4, 103.000, '01-08-2021', 33.98)

INSERT INTO item VALUES ( 315, 4, 132, 2, 2.99, 103.000)

INSERT INTO item VALUES ( 316, 4, 158, 3, 4.00, 103.000)

INSERT INTO item VALUES ( 317, 4, 159, 1, 4.00, 103.000)

INSERT INTO item VALUES ( 318, 4, 160, 1, 6.00, 103.000)

INSERT INTO item VALUES ( 319, 4, 168, 2, 3.00, 103.000)

INSERT INTO pedido VALUES ( 5, 104.000, '01-09-2021', 26.99)

INSERT INTO item VALUES ( 320, 5, 158, 1, 4.00, 104.000)

INSERT INTO item VALUES ( 321, 5, 159, 2, 4.00, 104.000)

INSERT INTO item VALUES ( 322, 5, 160, 1, 6.00, 104.000)

INSERT INTO item VALUES ( 323, 5, 168, 2, 3.00, 104.000)

INSERT INTO item VALUES ( 324, 5, 210, 1, 2.99, 104.000)

INSERT INTO pedido VALUES ( 6, 105.000, '01-10-2021', 28.99)

INSERT INTO item VALUES ( 325, 6, 159, 2, 4.00, 105.000)

INSERT INTO item VALUES ( 326, 6, 160, 1, 6.00, 105.000)

INSERT INTO item VALUES ( 327, 6, 168, 2, 3.00, 105.000)

INSERT INTO item VALUES ( 328, 6, 210, 1, 2.99, 105.000)

INSERT INTO item VALUES ( 329, 6, 212, 1, 6.00, 105.000)

INSERT INTO pedido VALUES ( 7, 106.000, '01-11-2021', 33.99)

INSERT INTO item VALUES ( 330, 7, 124, 2, 3.50, 106.000)

INSERT INTO item VALUES ( 331, 7, 160, 2, 6.00, 106.000)

INSERT INTO item VALUES ( 332, 7, 168, 2, 3.00, 106.000)

INSERT INTO item VALUES ( 333, 7, 210, 1, 2.99, 106.000)

INSERT INTO item VALUES ( 334, 7, 212, 1, 6.00, 106.000)

INSERT INTO pedido VALUES ( 8, 107.000, '01-12-2020', 37.99)

INSERT INTO item VALUES ( 335, 8, 124, 3, 3.50, 107.000)

INSERT INTO item VALUES ( 336, 8, 129, 1, 3.50, 107.000)

INSERT INTO item VALUES ( 337, 8, 168, 1, 3.00, 107.000)

INSERT INTO item VALUES ( 338, 8, 210, 1, 2.99, 107.000)

INSERT INTO item VALUES ( 339, 8, 212, 3, 6.00, 107.000)

INSERT INTO pedido VALUES ( 9, 108.000, '01-01-2021', 37.49)

INSERT INTO item VALUES ( 340, 9, 124, 1, 3.50, 108.000)

INSERT INTO item VALUES ( 341, 9, 129, 1, 3.50, 108.000)

INSERT INTO item VALUES ( 342, 9, 130, 1, 3.50, 108.000)

INSERT INTO item VALUES ( 343, 9, 210, 1, 2.99, 108.000)

INSERT INTO item VALUES ( 344, 9, 212, 4, 6.00, 108.000)

INSERT INTO pedido VALUES ( 10, 109.000, '01-02-2021', 63.47)

INSERT INTO item VALUES ( 345, 10, 124, 5, 3.50, 109.000)

INSERT INTO item VALUES ( 346, 10, 129, 1, 3.50, 109.000)

INSERT INTO item VALUES ( 347, 10, 130, 1, 3.50, 109.000)

INSERT INTO item VALUES ( 348, 10, 132, 3, 2.99, 109.000)

INSERT INTO item VALUES ( 349, 10, 212, 4, 6.00, 109.000)

INSERT INTO item VALUES ( 350, 10, 159, 1, 6.00, 109.000)

INSERT INTO pedido VALUES ( 11, 110.000, '30-11-2021', 926.00)

INSERT INTO item VALUES ( 351, 11, 260, 24, 8.00, 110.000);

INSERT INTO item VALUES ( 352, 11, 246, 50, 3.50, 110.000);

INSERT INTO item VALUES ( 353, 11, 249, 50, 3.50, 110.000);

INSERT INTO item VALUES ( 354, 11, 258, 24, 8.00, 110.000);

INSERT INTO item VALUES ( 355, 11, 254, 24, 5.00, 110.000);

ALTER TABLE item ADD COLUMN num\_notafiscal\_venda CHAR(50);

**-- População tabela Descarte**

INSERT INTO descarte VALUES ( 800, 212, 'Descarte item caixa estourada', 2);

INSERT INTO descarte VALUES ( 802, 101, 'Descarte garrafa rachada', 1);

INSERT INTO descarte VALUES ( 802, 125, 'Descarte garrafa rachada', 1);

INSERT INTO descarte VALUES ( 802, 130, 'Descarte garrafa rachada', 1);

**-- População tabela Devolução**

ALTER TABLE devolucao ALTER COLUMN tipo\_devo TYPE VARCHAR (4);

INSERT INTO devolucao VALUES (1,103, 'forn','Produto não solicitado',100 );

INSERT INTO devolucao VALUES (2,106, 'forn','50 unidades de produto a mais',50)

**Requisito 03 - Permitir troca de produtos. SQL**

**-- População tabela troca**

ALTER TABLE troca ALTER COLUMN descr\_troca TYPE VARCHAR(100)

INSERT INTO troca VALUES (901,1,'Cliente solicita troca do produto por outro mais gelado', 'cli',159 )

INSERT INTO troca VALUES (902,1,'Cliente solicita troca de sabor do produto, de laranja por morango', 'cli',210 )

INSERT INTO troca VALUES (903,6,'Lata vazia', 'forn',140 )

**-- População tabela estoque**

select \* from estoque

ALTER TABLE estoque ALTER COLUMN estoque\_min TYPE integer

ALTER TABLE estoque ALTER COLUMN estoque\_max TYPE integer

ALTER TABLE estoque DROP COLUMN valor\_qtd\_prod

INSERT INTO estoque VALUES (1,101,current\_timestamp(2),100,1,99,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (2,103,current\_timestamp(2),100,100,0,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (3,110,current\_timestamp(2),50,0,50,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (4,125,current\_timestamp(2),40,1,39,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (5,122,current\_timestamp(2),80,0,80,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (6,127,current\_timestamp(2),20,0,20,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja',20,50);

INSERT INTO estoque VALUES (7,120,current\_timestamp(2),10,0,10,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja',20,50);

INSERT INTO estoque VALUES (8,107,current\_timestamp(2),100,0,100,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (9,106,current\_timestamp(2),100,50,50,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (10,109,current\_timestamp(2),50,0,50,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (11,140,current\_timestamp(2),50,6,44,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (12,142,current\_timestamp(2),50,01,49,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (13,212,current\_timestamp(2),20,15,05,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja',20,50);

INSERT INTO estoque VALUES (14,132,current\_timestamp(2),20,19,01,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja',20,50);

INSERT INTO estoque VALUES (15,130,current\_timestamp(2),10,10,0,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja',20,50);

INSERT INTO estoque VALUES (16,129,current\_timestamp(2),10,09,01,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,50);

INSERT INTO estoque VALUES (17,210,current\_timestamp(2),20,04,16,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,50);

INSERT INTO estoque VALUES (18,124,current\_timestamp(2),40,21, 19,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,50);

INSERT INTO estoque VALUES (19,158,current\_timestamp(2),50,32,18,'Rua General Molina,35 Jd.Dabril- Armazém Reserva',20,50);

INSERT INTO estoque VALUES (20,159,current\_timestamp(2),50,08,42,'Rua General Molina,35 Jd.Dabril- Armazém Reserva', 20,50);

INSERT INTO estoque VALUES (21,154,current\_timestamp(2),20,0,20,'Rua General Molina,35 Jd.Dabril- Armazém Reserva', 20,50);

INSERT INTO estoque VALUES (22,160,current\_timestamp(2),50,06,44,'Rua General Molina,35 Jd.Dabril- Armazém Reserva', 20,50);

INSERT INTO estoque VALUES (23,168,current\_timestamp(2),20,09,11,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,50);

INSERT INTO estoque VALUES (24,155,current\_timestamp(2),20,0,20,'Rua General Molina,35 Jd.Dabril- Armazém Reserva', 20,50);

INSERT INTO estoque VALUES (25,205,current\_timestamp(2),20,01,19,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,50);

INSERT INTO estoque VALUES (26,260,current\_timestamp(2),48,24,24,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,96);

INSERT INTO estoque VALUES (27,258,current\_timestamp(2),48,24,24,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,96);

INSERT INTO estoque VALUES (28,254,current\_timestamp(2),48,24,24,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 20,96);

INSERT INTO estoque VALUES (29,246,current\_timestamp(2),100,50,50,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 50,150);

INSERT INTO estoque VALUES (30,249,current\_timestamp(2),100,30,50,'Praça Thomas Coelho de Almeida,35 Jd.Dabril- Loja', 50,150);

# Implementação de Consultas SQL

**Requisito 04 - Permissão para Inventário. SQL**

SELECT c.cod\_prod as "Código Produto", p.prod\_nome as Produto, p.prod\_descr AS Descrição,

c.qtd\_total AS "Quantidade disponivel em estoque"

From estoque c JOIN produto p

ON (c.cod\_prod = p.cod\_prod)

*--Funcionário deve analisar quantidade no sistema e conferir a quantidade física em estoque*

**Requisito 05 - Gerar relatório de controle de estoque mínimo. SQL**

SELECT INITCAP('Data hora contagem: '||e.dt\_hora\_controle||', Cód. do produto: '

||e.cod\_prod||', Nome do produto: '||p.prod\_nome||', Quantidade disponivel em estoque:'

|| e.qtd\_total||' unidades.') AS "Relação de produtos com menos de 20 unidades"

FROM estoque e inner join produto p

ON (e.cod\_prod = p.cod\_prod)

WHERE e.qtd\_total <=20 ;

**Requisito 06 - Gerar controle pela data de validade. SQL**

-- Produtos com vencimento em 3 meses

SELECT c.cod\_prod, p.prod\_nome, p.prod\_marca, c.dt\_validade, c.valor\_total

From fornecimento c inner join produto p

on (c.cod\_prod=p.cod\_prod)

WHERE dt\_validade < current\_date + INTERVAL '3' MONTH

-- Produtos com vencimento em 6 meses

SELECT c.cod\_prod, p.prod\_nome, p.prod\_marca, c.dt\_validade, c.valor\_total

From fornecimento c inner join produto p

on (c.cod\_prod=p.cod\_prod)

WHERE dt\_validade < current\_date + INTERVAL '6' MONTH

-- Produtos com vencimento em 12 meses

SELECT c.cod\_prod, p.prod\_nome, p.prod\_marca, c.dt\_validade, c.valor\_total

From fornecimento c inner join produto p

on (c.cod\_prod=p.cod\_prod)

WHERE dt\_validade < current\_date + INTERVAL '1' YEAR

-- Produtos com vencimento em 24 meses

SELECT c.cod\_prod, p.prod\_nome, p.prod\_marca, c.dt\_validade, c.valor\_total

From fornecimento c inner join produto p

on (c.cod\_prod=p.cod\_prod)

WHERE dt\_validade < current\_date + INTERVAL '2' YEAR

**-- Requisito 07 - Gerar relatório com dados do fornecedor.**

SELECT ('Nome Fornecedor: '||forn\_nome||', Telefone: '||forn\_fone|| ', E-mail:'||forn\_email) AS "Dados Fornecedor"

FROM fornecedor

WHERE forn\_tipo = 'PJ';

# Extração de Dados Operacionais com Phyton

**Requisitos 08 aos 16 – Conexão Postgres com código Python.**

#SEGUE O CÓDIGO EM PYTHON:

import tkinter as tk

import PySimpleGUI as sg

from PyQt5.QtSql import QSqlDatabase, QSqlQuery

import sys

import psycopg2

def limpar1():

window['-COD1-'].update('')

window['-NOME1-'].update('')

window['-DESC-'].update('')

window['-CDC-'].update('')

window['-QDM-'].update('')

window['-MEP-'].update('')

window['-MAP-'].update('')

#window['-IDI-'].update('')

#window['-VDVI-'].update('')

#window['-QTI-'].update('')

#window['-DDV-'].update('')

def limpar2():

window['-COD2-'].update('')

window['-NOME2-'].update('')

window['-IDI-'].update('')

window['-VDVI-'].update('')

window['-QTI-'].update('')

window['-DDV-'].update('')

def atualiza1():

if len(lista1) == 0:

limpar1()

else:

print(lista1[indice1])

window['-COD1-'].update(lista1[indice1][0])

window['-NOME1-'].update(lista1[indice1][1])

window['-DESC-'].update(lista1[indice1][2])

window['-CDC-'].update(lista1[indice1][3])

window['-QDM-'].update(lista1[indice1][4])

window['-MEP-'].update(lista1[indice1][5])

window['-MAP-'].update(lista1[indice1][6])

#window['-IDI-'].update(lista[indice][7])

#window['-VDVI-'].update(lista[indice][8])

#window['-QTI-'].update(lista[indice][9])

#window['-DDV-'].update(lista[indice][10])

def atualiza2():

if len(lista2) == 0:

limpar2()

else:

print(lista2[indice2])

window['-COD2-'].update(lista2[indice2][0])

window['-NOME2-'].update(lista2[indice2][1])

window['-IDI-'].update(lista2[indice2][2])

window['-VDVI-'].update(lista2[indice2][3])

window['-QTI-'].update(lista2[indice2][4])

window['-DDV-'].update(lista2[indice2][5])

def todos():

global indice1

global lista

resposta = []

with con:

with con.cursor() as cursor:

cursor.execute("SELECT \* FROM produto;")

resposta = cursor.fetchall()

lista1.clear()

for i in range(len(resposta)):

lista1.append(list(resposta[i]))

sg.popup('Quantidade de registros: ' + str(len(resposta)))

indice1 = 0

atualiza1()

# Inicialização BD

con = psycopg2.connect(host="localhost", database="Trabson PI", user="postgres", password="brenopostgre")

'''with con:

with con.cursor() as cursor:

cursor.execute("""

SET DATESTYLE TO postgres, DMY ;

--1 Tabela Categoria

DROP TABLE IF EXISTS categoria CASCADE;

CREATE TABLE categoria (

cod\_catg INTEGER PRIMARY KEY,

categoria VARCHAR (50) NOT NULL CHECK (categoria IN ('Bebida não alcoólica','Bebida alcoólica')),

descr\_catg VARCHAR(200));

--2 Tabela Produto

DROP TABLE IF EXISTS produto CASCADE;

CREATE TABLE produto (

cod\_prod INT PRIMARY KEY,

prod\_nome VARCHAR(30),

prod\_descr VARCHAR(50),

cod\_catg INT, FOREIGN KEY (cod\_catg) REFERENCES categoria,

qtd\_medida NUMERIC(3,1),

prod\_medida VARCHAR(10),CHECK (prod\_medida IN ('litro', 'litros', 'ml')),

prod\_marca VARCHAR (100));

--3 Tabela Fornecedor

DROP TABLE IF EXISTS fornecedor CASCADE;

CREATE TABLE fornecedor (

cod\_forn INT PRIMARY KEY,

forn\_tipo CHAR(2) NOT NULL CHECK (forn\_tipo IN ('PJ', 'PF')),

forn\_email VARCHAR(30),

forn\_fone NUMERIC(11),

forn\_end VARCHAR(100) ,

forn\_nome VARCHAR(100) NOT NULL);

--4 Tabela Fornecimento

DROP TABLE IF EXISTS fornecimento CASCADE;

CREATE TABLE fornecimento (

id\_fornecto INT PRIMARY KEY,

num\_notafiscal\_compra NUMERIC(50),

cod\_forn INTEGER, FOREIGN KEY (cod\_forn) REFERENCES fornecedor,

cod\_prod INTEGER, FOREIGN KEY (cod\_prod) REFERENCES produto,

dt\_entrada DATE,

num\_lote VARCHAR (30),

dt\_validade DATE,

qtd\_prod NUMERIC(7,2),

valor\_custo\_uni NUMERIC(7,2),

valor\_total NUMERIC(7,2)

);

--5 Tabela Estoque

DROP TABLE IF EXISTS estoque CASCADE;

CREATE TABLE estoque (

id\_estoque INT PRIMARY KEY,

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto ,

dt\_hora\_controle TIMESTAMP NOT NULL,

qtd\_entrada INTEGER NOT NULL,

qtd\_saida INTEGER NOT NULL,

qtd\_total INTEGER NOT NULL,

end\_estoque VARCHAR(100) NOT NULL,

estoque\_min integer,

estoque\_max integer

);

--6 Tabela Pedido

DROP TABLE IF EXISTS pedido CASCADE;

CREATE TABLE pedido (

id\_pedido INT PRIMARY KEY,

num\_notafiscal\_venda CHAR(50),

dt\_hora\_pedido TIMESTAMP NOT NULL,

valor\_total NUMERIC(7,2));

--7 Tabela Item

DROP TABLE IF EXISTS item CASCADE;

CREATE TABLE item (

id\_item integer primary key,

id\_pedido INTEGER, FOREIGN KEY (id\_pedido) REFERENCES pedido ,

cod\_prod INTEGER,FOREIGN KEY (cod\_prod) REFERENCES produto ,

quantidade NUMERIC(7,2),

valor\_venda\_uni NUMERIC(7,2),

num\_notafiscal\_venda NUMERIC(7,2));

--8 Tabela de Troca

DROP TABLE IF EXISTS troca CASCADE;

CREATE TABLE troca (

id\_troca INT PRIMARY KEY,

qtd\_prod NUMERIC(7,2),

descr\_troca VARCHAR(100),

tipo\_troca CHAR (4) CHECK( tipo\_troca IN ('forn', 'cli')),

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto );

--9 Tabela de Descarte

DROP TABLE IF EXISTS descarte CASCADE;

CREATE TABLE descarte (

id\_desc INT PRIMARY KEY,

cod\_prod INT, FOREIGN KEY (cod\_prod) REFERENCES produto,

descr\_desc VARCHAR(50),

qtd\_prod NUMERIC(7,2));

--10 Tabela de Devolucao

DROP TABLE IF EXISTS devolucao CASCADE;

CREATE TABLE devolucao (

id\_dev INT PRIMARY KEY,

cod\_prod INT , FOREIGN KEY (cod\_prod) REFERENCES produto ,

tipo\_devo CHAR NOT NULL CHECK (tipo\_devo IN ('forn', 'cli')),

descr\_devo VARCHAR(50),

qtd\_prod NUMERIC(7,2)

);

""")'''

#lista = []

#indice = 0

lista1 = []

indice1 = 0

lista2 = []

indice2 = 0

def make\_win3():

sg.theme('LightGreen5')

layout = [ [sg.Text('Usuário:', size=(8, 1)), sg.InputText(key='-USER-', size=(15, 1))],

[sg.Text('Senha:', size=(8, 1)), sg.InputText('', size=(15, 1), key='-PASSWORD-', password\_char='\*')],

[sg.Button('Entrar', size=(10, 1), key='-ENTRAR-')] ]

return sg.Window('Login', layout, location=(550, 290), finalize=True)

def make\_win1():

sg.theme('LightGreen5')

layout = [

[sg.Text("Código Produto:", size=(19, 1)),

sg.InputText(size=([40], 1), key="-COD1-", focus=True)],

[sg.Text("Nome:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-NOME1-", focus=False)],

[sg.Text("Descrição:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-DESC-", focus=False)],

[sg.Text("Código da Categoria:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-CDC-", focus=False)],

[sg.Text("Quantidade da Medida:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-QDM-", focus=False)],

[sg.Text("Medida do Produto:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-MEP-", focus=False)],

[sg.Text("Marca do Produto:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-MAP-", focus=False)],

[sg.Button('Limpar', size=(17, 1), key="-LIMPAR1-"),

sg.Button('Inserir', size=(17, 1), key="-INSERIR-"),

sg.Button('Atualizar', size=(17, 1), key="-ATUALIZAR-")],

[sg.Button('Remover', size=(17, 1), key="-REMOVER-"),

sg.Button('Procurar', size=(17, 1), key="-PROCURAR-"),

sg.Button('Todos', size=(17, 1), key="-TODOS-")],

[sg.Button('<<', size=(17, 1), key="-<<1-"),

sg.Button('Gerenciamento', size=(17, 1), key="-GERENCIAMENTO-"),

sg.Button('>>', size=(17, 1), key="->>1-")]

]

return sg.Window('Produtos', layout, location=(425, 200), finalize=True)

def make\_win2():

sg.theme('LightGreen5')

layout = [

[sg.Text("Código Produto (P):", size=(19, 1)),

sg.InputText(size=([40], 1), key="-COD2-", focus=True)],

[sg.Text("Nome (P):", size=(19, 1)),

sg.InputText(size=(40, 1), key="-NOME2-", focus=False)],

[sg.Text("ID Item:", size=(19, 1)),

sg.InputText(size=([40], 1), key="-IDI-", focus=False)],

[sg.Text("Valor de Venda:", size=(19, 1)),

sg.InputText(size=([40], 1), key="-VDVI-", focus=False)],

[sg.Text("Quantidade Total:", size=(19, 1)),

sg.InputText(size=([40], 1), key="-QTI-", focus=False)],

[sg.Text("Data de Validade:", size=(19, 1)),

sg.InputText(size=(40, 1), key="-DDV-", focus=False)],

[sg.Button('Limpar', size=(17, 1), key="-LIMPAR2-"),

sg.Button('Indicador', size=(17, 1), key="-INDICADOR-"),

sg.Button('Validade', size=(17, 1), key="-VALIDADE-")],

[sg.Button('<<', size=(17, 1), key="-<<2-"),

sg.Button('Home', size=(17, 1), key="-HOME-"),

sg.Button('>>', size=(17, 1), key="->>2-")]

]

return sg.Window('Gerenciamento', layout, location=(425, 200), finalize=True)

#, disabled = true, readOnly = true

#, disabled="disabled"

#, readonly="readonly"

window3, window1, window2 = make\_win3(), None, None # start off with 1 window open

#make\_win3.close()

#window3 = make\_win3(), None

#app = tk.Tk()

#def callback(event):

# if labelUser == 'breno' and labelPass == 'admin':

# window1 = make\_win1()

# window3.close()

# else:

# window['-USER-'].update( '' )

# window['-PASSWORD-'].update( '' )

# tk.popup('Usuário ou senha incorretos', location=(560, 300)) #, size=(400,400)

# Run the Event Loop

while True:

#make\_win1.resize(make\_win2.sizeHint().width,

#make\_win1.size().height() + content.sizeHint().height());

#self.move(QtGui.QApplication.desktop().screen().rect().center()- self.rect().center())

# app = tk.Tk()

# labelUser = tk.Label(app,

# text = "Usuário:")

# labelUser.grid(column=0, row=0, ipadx=5, pady=5, sticky=tk.W+tk.N)

#

# labelPass = tk.Label(app,

# text = "Senha:")

# labelPass.grid(column=0, row=1, ipadx=5, pady=5, sticky=tk.W+tk.S)

#

# entryWidth = tk.Entry(app, width=20)

# entryHeight = tk.Entry(app, width=20)

#

# entryUser.grid(column=1, row=0, padx=10, pady=5, sticky=tk.N)

# entryPass.grid(column=1, row=1, padx=10, pady=5, sticky=tk.S)

#

# resultButton = tk.Button(app, text = 'Entrar')

# resultButton.grid(column=0, row=2, pady=10, sticky=tk.W)

#

# app.bind('<Return>', callback)

window, event, values = sg.read\_all\_windows()

if event == sg.WIN\_CLOSED:

break

elif event == '-ENTRAR-' and not window2:

if values['-USER-'] == 'breno' and values['-PASSWORD-'] == 'admin':

window1 = make\_win1()

window3.close()

else:

window['-USER-'].update( '' )

window['-PASSWORD-'].update( '' )

sg.popup('Usuário ou senha incorretos', location=(560, 300)) #, size=(400,400)

#sg.popup.geometry( "300x50" )

elif event == '-GERENCIAMENTO-':

window2 = make\_win2()

window1.close()

elif event == '-HOME-':

window1 = make\_win1()

window2.close()

elif event == "-LIMPAR1-":

limpar1()

elif event == "-LIMPAR2-":

limpar2()

elif event == "-INSERIR-":

with con:

with con.cursor() as cursor:

cursor.execute(

"INSERT INTO produto (cod\_prod, prod\_nome, prod\_descr, cod\_catg, qtd\_medida, prod\_medida, prod\_marca) VALUES (%s, %s, %s, %s, %s, %s, %s);",

(values['-COD1-'], values['-NOME1-'], values['-DESC-'], values['-CDC-'], values['-QDM-'],

values['-MEP-'], values['-MAP-'])

#INSERT INTO produto (cod\_prod, prod\_nome,) VALUES (%s, %s);",

#(values['-VDVI-'], values['-QTD-'], values['-DDV-'])

# INSERT INTO produto (cod\_prod, prod\_nome,) VALUES (%s, %s);",

# (values['-VDVI-'], values['-QTD-'], values['-DDV-'])

# INSERT INTO produto (cod\_prod, prod\_nome,) VALUES (%s, %s);",

# (values['-VDVI-'], values['-QTD-'], values['-DDV-'])

)

limpar1()

elif event == "-ATUALIZAR-":

with con:

with con.cursor() as cursor:

#cursor.execute("UPDATE produto SET prod\_nome = %s, prod\_descr = %s, cod\_catg = %s, qtd\_medida = %s, prod\_medida = %s, prod\_marca = %s WHERE cod\_prod = %s",

#(values['-NOME1-'], values['-DESC-'], values['-CDC-'], values['-QDM-'], values['-MEP-'], values['-MAP-']), values['-COD1-'])

#lista1[indice1] = [values['-COD1-'], values['-NOME1-'], values['-DESC-'], values['-CDC-'], values['-QDM-'], values['-MEP-'], values['-MAP-']]

cursor.execute("UPDATE produto SET prod\_nome = %s, prod\_descr = %s, cod\_catg = %s, qtd\_medida = %s, prod\_medida = %s, prod\_marca = %s WHERE cod\_prod = %s",

(values['-NOME1-'], values['-DESC-'], values['-CDC-'], values['-QDM-'], values['-MEP-'], values['-MAP-'], values['-COD1-']))

elif event == "-REMOVER-":

with con:

with con.cursor() as cursor:

cursor.execute("DELETE FROM produto WHERE cod\_prod = %s", (values['-COD1-'],))

lista1.pop(indice1)

indice1 -= 1

atualiza1()

elif event == "-PROCURAR-":

with con:

with con.cursor() as cursor:

cursor.execute("SELECT \* FROM produto WHERE prod\_nome LIKE %s;",

('%' + values['-NOME1-'] + '%',))

resposta = cursor.fetchall()

lista1.clear()

for i in range(len(resposta)):

lista1.append(list(resposta[i]))

sg.popup('Quantidade de registros: ' + str(len(resposta)))

indice1 = 0

atualiza1()

elif event == "-INDICADOR-":

with con:

with con.cursor() as cursor:

cursor.execute("SELECT po.cod\_prod, po.prod\_nome, it.id\_item, it.valor\_venda\_uni, et.qtd\_total, fm.dt\_validade FROM produto po INNER JOIN estoque et ON (po.cod\_prod = et.cod\_prod) JOIN item it ON (po.cod\_prod = it.cod\_prod) JOIN fornecimento fm ON (po.cod\_prod = fm.cod\_prod) WHERE et.qtd\_total < 20",

('%' + values['-COD2-'] + '%', '%' + values['-NOME2-'] + '%', '%' + values['-IDI-'] + '%', '%' + values['-VDVI-'] + '%', '%' + values['-QTI-'] + '%',

'%' + values['-DDV-'] + '%',)

)

resposta = cursor.fetchall()

lista2.clear()

for i in range(len(resposta)):

lista2.append(list(resposta[i]))

sg.popup('Valores: ' + str(len(resposta)))

indice2 = 0

atualiza2()

elif event == "-VALIDADE-":

with con:

with con.cursor() as cursor:

cursor.execute("SELECT po.cod\_prod, po.prod\_nome, it.id\_item, it.valor\_venda\_uni, et.qtd\_total, fm.dt\_validade FROM produto po INNER JOIN estoque et ON (po.cod\_prod = et.cod\_prod) JOIN item it ON (po.cod\_prod = it.cod\_prod) JOIN fornecimento fm ON (po.cod\_prod = fm.cod\_prod) WHERE fm.dt\_validade > fm.dt\_validade - INTERVAL '30' DAY",

('%' + values['-COD2-'] + '%', '%' + values['-NOME2-'] + '%', '%' + values['-IDI-'] + '%', '%' + values['-VDVI-'] + '%', '%' + values['-QTI-'] + '%',

'%' + values['-DDV-'] + '%',)

)

#('%' + values['-NOME-'] + '%',

#'%' + values['-COD-'] + '%'

#'%' + values['-IDI-'] + '%',

#'%' + values['-VDVI-'] + '%',

#'%' + values['-QTI-'] + '%',

#'%' + values['-DDV-'] + '%',)

resposta = cursor.fetchall()

lista2.clear()

for i in range(len(resposta)):

lista2.append(list(resposta[i]))

sg.popup('Valores: ' + str(len(resposta)))

indice2 = 0

atualiza2()

elif event == "-TODOS-":

todos()

elif event == "->>1-":

indice1 += 1

if indice1 >= len(lista1): indice1 = len(lista1) - 1

atualiza1()

elif event == "-<<1-":

indice1 -= 1

if indice1 <= 0: indice1 = 0

atualiza1()

elif event == "->>2-":

indice2 += 1

if indice2 >= len(lista2): indice2 = len(lista2) - 1

atualiza2()

elif event == "-<<2-":

indice2 -= 1

if indice2 <= 0: indice2 = 0

atualiza2()

window.close()

# Fazer as mudanças para a base persistente

con.commit()

# Fechar a comunicação com o servidor

cursor.close()

con.close()

#window.close()

#cursor.close()

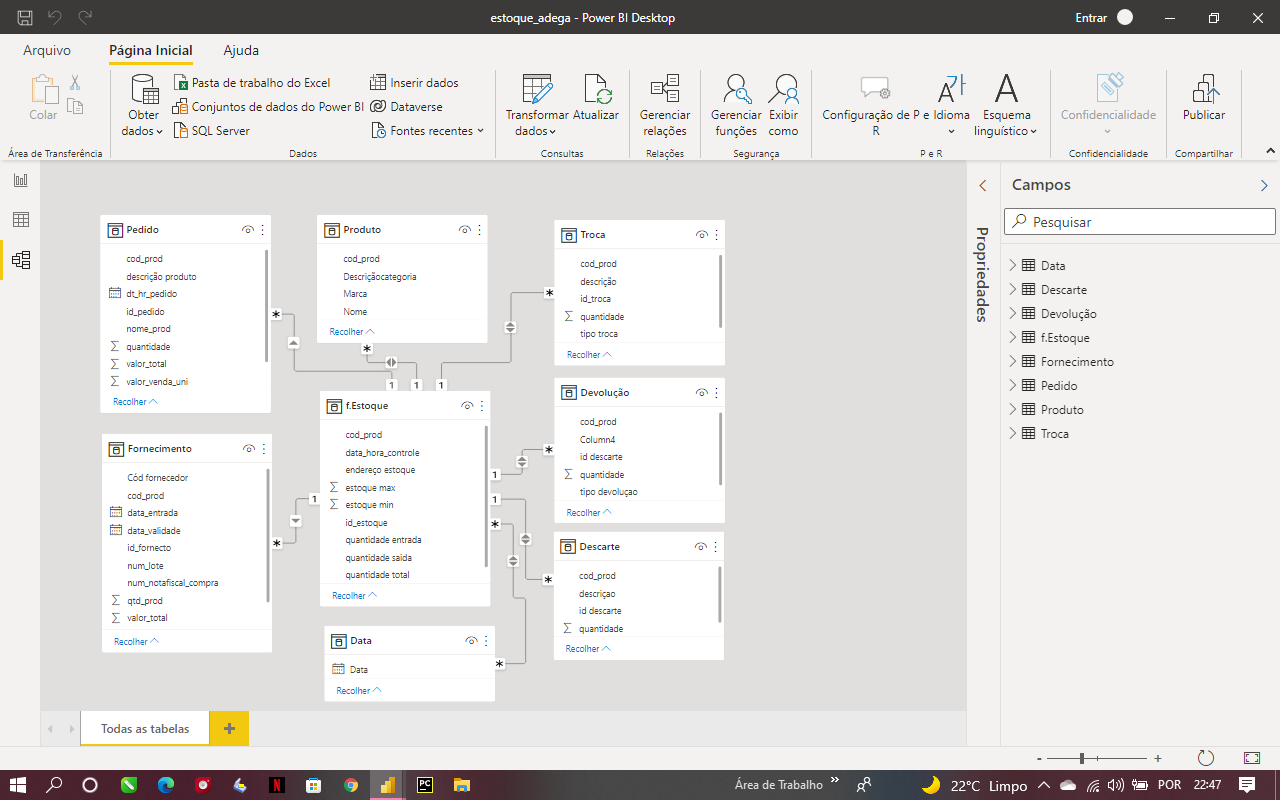
#con.close()

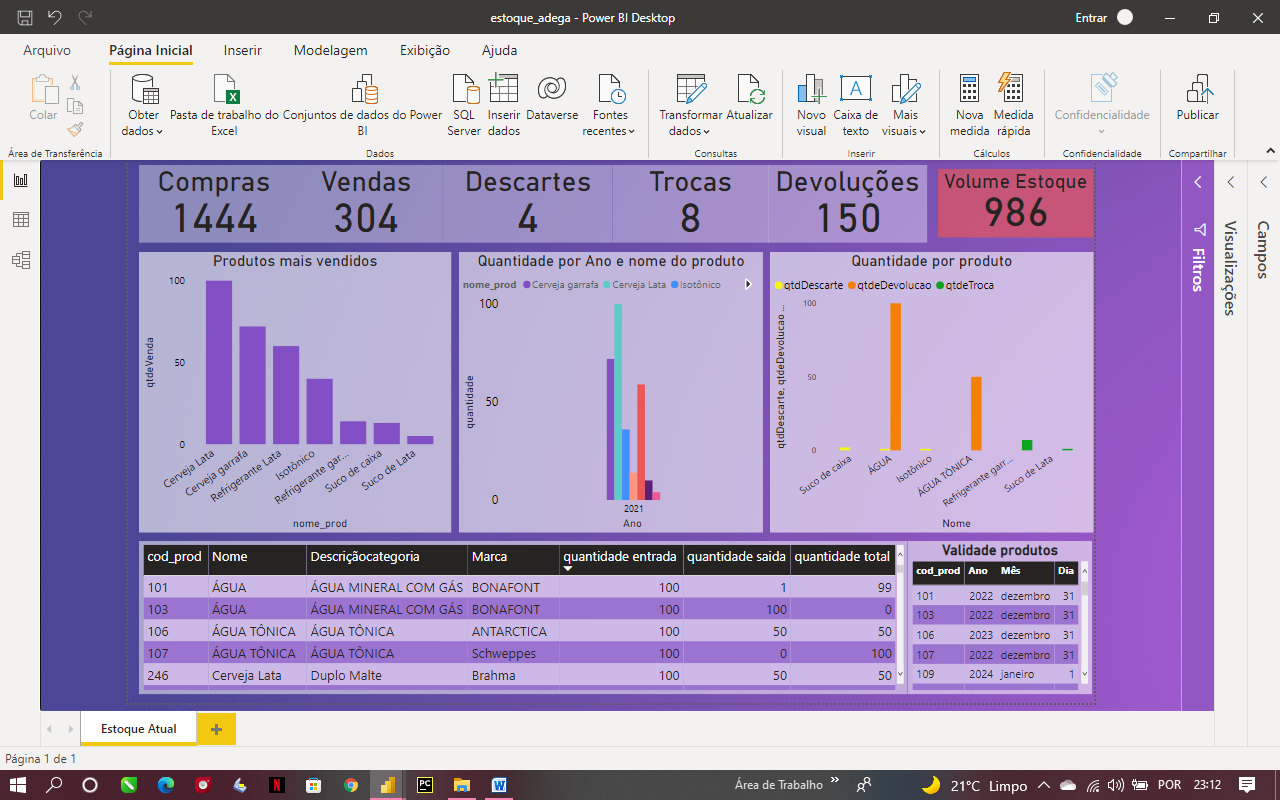
# Extração de Dados Consolidados com Power BI

**Requisito nº17 Relatório Estoque Atual**

Tabela Fato = Estoque

Tabela Dimensão = Pedido, Fornecimento, Produto, Troca, Descarte, Devolução e Data





Medidas utilizadas = Quantidade compras, quantidade Vendas, quantidade trocas, quantidade devoluções e quantidade descartes.

Cálculo utilizado = VolumeEstoque = [quantidadeCompras] – [quantidadeVendas] – [quantidadeDevoluções] – [quantidadeDescartes]

Gráficos utilizados = Cartão, gráfico de tabelas, gráfico de colunas empilhadas e gráfico de colunas agrupadas.