Documentação do Projeto de Web Scraping e Carga de Dados Python + SQL

SISTEMA DE VISITAS – SOVIS

Este script realiza um processo completo de extração de dados (Web Scraping), tratamento, cruzamento e carga de dados em um Data Warehouse PostgreSQL com estrutura dimensional. A finalidade é automatizar a coleta de informações de visitas e vendas do sistema de gestão, possibilitando análise via BI.

==========
CONFIGURAÇÃO
==========

Define parâmetros de conexão com o banco PostgreSQL e o caminho local para o arquivo Excel com informações de vendedores.

==========
WEB SCRAPING COM SELENIUM
==========

Esta etapa utiliza o Selenium para automação do navegador e extração dos dados do sistema.

- `iniciar_driver()`: Inicializa o navegador Chrome com as configurações necessárias.
- `login_sovis()`: Acessa o site, insere login e senha automaticamente (com base em XPath).
- `acessar_aba_cliente()`: Navega até a aba de clientes para acesso à tabela de dados.
- `extrair_dados_pagina(driver)`: Extrai as colunas de cada cliente visível na página atual.
- `clicar_proxima_pagina(driver)`: Verifica se há nova página e avança, se possível.

O loop principal itera por até 503 páginas coletando os campos:

- código
- id_sovis
- razão social
- DSV
- última visita
- situação

Os dados são tratados com valores padrões para registros como 'Nunca Vendido' e 'Nunca Visitado'.

==========

INTEGRAÇÃO PYTHON + SQL

==========

O script conecta-se ao banco PostgreSQL por meio do SQLAlchemy e executa instruções SQL diretamente com o psycopg2 para controle total do processo de carga. As queries SQL são usadas para truncamento das tabelas, criação de constraints e inserção dos dados com chaves estrangeiras referenciando as dimensões.

==========

CRUZAMENTO DOS DADOS

==========

Após a coleta dos dados, o script cruza os códigos de cliente com uma planilha Excel para identificar o vendedor responsável por cada cliente.

- `codigo_ajustado`: remove os últimos 4 dígitos do código original para permitir o cruzamento correto.

Dados ausentes são preenchidos com 'Sem Informação'.

==========

CARGA NO DATA WAREHOUSE

==========

- Conexão ao PostgreSQL via SQLAlchemy.
- Limpeza total das tabelas de dimensões e fato.

- Recriação das constraints entre fato e dimensões.
- Carga única nas dimensões, sem duplicatas.
- Carga linha a linha na tabela fato utilizando `psycopg2`.

==========

VALIDAÇÕES E SEGURANÇA

==========

- Tratamento de exceções durante extração e carga.
- Apenas registros com todas as dimensões válidas são inseridos.
- Datas e campos padronizados evitam erros futuros.

==========

FINALIDADE

==========

O pipeline permite a criação de relatórios e painéis que auxiliam na gestão comercial e acompanhamento de visitas e vendas.

- Identificação de clientes inativos
- Análise de visitas e vendas por vendedor
- Visualização temporal das interações
- Suporte à tomada de decisão via Power BI

==========

IMPLEMENTAÇÕES FUTURAS

==========

- Log de execução e erros
- Parâmetro para número de páginas
- Agendamento via Task Scheduler ou Airflow
- Tratamento de sessão expirada
- Dashboard e relatórios gerenciais

==========

RESPONSÁVEL
==========

Desenvolvido por: Lucca Carnaúba Peixoto Rosário

Stack utilizada: Python + Selenium + Pandas + SQLAlchemy + psycopg2 + PostgreSQL