Olá, Futuro Maxter!

Seja bem vindo ao teste de seleção para o cargo Engenheiro de Dados Jr no time de dados da Appmax. Esse teste tem como principal objetivo verificar suas habilidades e competências para a vaga, nele você poderá pôr em prática seus conhecimentos. É importante que você leia atentamente todas as instruções do documento antes de iniciar o desafio.

O contexto deste desafio é de prover uma pipeline de dados composta por um ETL (que além das transformações terão enriquecimento dos dados) para o time de análise de risco de um e-commerce.

Critérios de avaliação

- Coerência da solução proposta com o que foi solicitado
    - Atividades
    - Instalação e configuração do ambiente
- Uso do Airflow como orquestrador de pipeline.
- Boas práticas de programação como modularização, POO, SOLID e etc.
- Organização do projeto, estrutura, documentação e versionamento.
- Bônus:
    - Uso de Docker

Atividade 1: Ingestão e normalização dos dados

Origem dos dados: pasta em formato .zip com dois arquivos.

Os dados são compostos por dois arquivos (pedidos.csv, ips.csv) em formato csv contendo dados de pedidos e dos ips de usuário que fizeram os pedidos.

Atividades:

- Criar camadas de dados da pipeline, sendo elas bronze, silver e gold no sistema de arquivos do seu sistema operacional.
- Efetuar a extração dos dados na camada bronze.
- **Usando o pyspark** você deve efetuar as seguintes transformações na tabela ‘pedidos.csv’:

- Concatenação das colunas ‘nome’ e ‘sobrenome’ em uma única coluna chamada ‘nome_completo’.

- Remover o símbolo ‘$’ da coluna ‘valor_pedido’ e efetuar a mudança de tipo para Double.

- Alterar o tipo da coluna ‘data_pedido’ para timestamp.

- Salvar esta tabela em formato de parquet na camada Silver.

Atividade 2: Enriquecimento dos dados

Com a tabela de pedidos na camada silver, é hora de enriquecer os dados da tabela ‘ips.csv’. Esta tabela é composta de uma coluna ‘id’, que corresponde a um relacionamento com a tabela de pedidos. Nesta atividade você irá efetuar o consumo de uma API de geolocalização de IPs, tendo a sua documentação no link: https://ipwhois.io/documentation.

Atividades:

- Consumir a API com todos os IPs da coluna ‘ip’ e salvar as informações referente as keys de ‘region’, ‘city’ e ‘country’ nas colunas a serem criadas ‘regiao’, ‘cidade’ e ‘pais’ respectivamente.

- Salvar esta tabela em formato parquet na camada silver.

Note que após esta atividade a sua tabela de ips terá o seguinte cabeçalho:

|-- id: integer (nullable = true)

|-- user_agent: string (nullable = true)

|-- regiao: string (nullable = true)

|-- cidade: string (nullable = true)

|-- pais: string (nullable = true)

Atividade 3: Transformação dos dados

Com os dados normalizados na camada silver, seu desafio agora é efetuar algumas transformações que irão resultar em uma única tabela contendo as informações finais requisitadas pelos analistas de dados da empresa.

Atividades:

- **Utilizando o pyspark**, efetue a junção de ambas as tabelas usando usando a chave estrangeira ‘id_ip’ da tabela ‘pedidos’ com a chave primária ‘id’ da tabela ‘ips’.

- Você deve manter todas as colunas, exceto ip, user_agent, id_ip.

- Salve os dados em formato parquet na camada gold.

Atividade 4: Uso dos dados

Com os dados transformados e agregados na camada gold você deve criar um arquivo de jupyter notebook para consumir os dados da camada gold.

Atividades:

- Efetue o carregamento da tabela em parquet que está na camada gold em um dataframe.
- A partir deste dataset você deve:
- Agrupar e encontrar o valor total de pedidos por ‘tipo_cc’ dos cartões mastercard e visa.
- Agrupar e encontrar o valor total de pedidos por dia.

Processo de submissão

O desafio deve ser entregue via repositório privado no GitHub com o compartilhamento do repositório com o perfil lucassuchy.

Bom trabalho e boa sorte!