Teste técnico de engenharia de dados

Este teste está dividido em duas etapas, sendo uma delas mais voltada para uso de SQL e outra com foco em Python, ambas linguagens muito utilizadas no dia a dia de um engenheiro de dados do Bullla. As seções a seguir possuem todas as instruções necessárias para que você consiga desenvolver e entregar o teste da maneira que se espera!

Controle de versão e hospedagem de código

Todo o código fonte, documentação e resultados produzidos no seu teste deverão ser armazenados em um repositório **PRIVADO** do **GitHub**. Esse repositório será posteriormente compartilhado com o time técnico do Bullla para avaliação.

É extremamente recomendável que você crie o repositório GIT desde o início do projeto, realizando **commits** de acordo com o seu progresso no desafio. Os commits também farão parte da sua avaliação.

Documentação e ambiente

Toda a documentação do projeto deve ser inserida dentro do arquivo **README** na raiz do repositório. A documentação deve conter instruções de como instalar e reproduzir o código que foi desenvolvido.

Além disso, ela também pode conter comentários a respeito da solução desenvolvida, desenho de arquitetura, insights e o que mais achar relevante de ser compartilhado. Os commits e a documentação podem ser em português, mas é recomendável utilizar inglês no código fonte, visto que é o padrão adotado no Bullla.

Por fim, o engenheiro do Bullla responsável por testar sua aplicação deve ser capaz de instalar e executar todo o código desenvolvido da maneira mais fácil possível. Por conta disso, a gestão de dependências para tornar o código reprodutível em outra máquina é uma habilidade adicional que será levada em conta.

Fase 1

A primeira fase do projeto consiste no uso de SQL. Para isso, você deverá utilizar o Jupyter notebook que foi fornecido junto ao material do teste. Neste notebook, você irá encontrar um

snippet básico de código com as instruções de como transformar o seu Dataframe em um banco de dados SQLite para conseguir realizar as Querys.

Dito isso, seu desafio inicial gira em torno da habilidade de manipular diferentes fontes de dados e criar querys para extrair informações valiosas que serão utilizadas pelas demais áreas do Bullla. Para isso, iremos utilizar dados fictícios de clientes, contas, cartões e transações. Todas as bases de dados já se encontram no material de apoio do teste.

Primeira Query

O Bullla gostaria de realizar uma campanha de marketing por e-mail, a fim de promover o uso mais intenso do cartão de crédito por parte dos clientes que utilizaram o cartão recentemente.

Para isso, é preciso extrair o nome, CPF e e-mail de todos os clientes que tenham ao menos R$ 400,00 reais de compras aprovadas nos últimos dois meses. Além disso, esses clientes precisam estar com a conta ativa e com o cartão desbloqueado, a menos que o código de bloqueio do cartão seja igual a “M”.

Segunda Query

Pensando em promover o uso do cartão para as compras da *blackfriday*, o Bullla fez uma parceria com alguns lojistas e gostaria de propor um aumento temporário no limite do cartão de crédito para alguns clientes específicos. Para isso, é necessário classificar os clientes em um Ranking, a fim de posteriormente aplicar o modelo de cálculo de crédito.

Dessa forma, você precisa extrair o CPF, número da conta, número do cartão e o Ranking de todos os clientes COMPRADORES e NÃO COMPRADORES nas lojas parceiras.

Os clientes considerados COMPRADORES são aqueles que tenham realizado uma média de ao menos R$ 300,00 em compras aprovadas nos últimos 6 meses, considerando as lojas de ID (6, 18, 24, 25). Além disso, esses clientes precisam estar com pelo menos 70% do limite do cartão consumido, para justificar o aumento temporário.

De igual modo, não possuímos interesse em clientes com conta inativa ou com o cartão bloqueado (a menos que o código de bloqueio do cartão seja igual a “M”). Todos os clientes que se enquadrarem como COMPRADORES, devem receber o Ranking = “A”.

Os NÃO COMPRADORES, por sua vez, são todos os demais clientes da base que possuem conta ativa e cartão desbloqueado (a menos que o código de bloqueio do cartão seja igual a “M”). Porém, o Ranking dos clientes NÃO COMPRADORES deve ser atribuído de acordo com as seguintes regras:

- Clientes entre 70% e 80% do limite do cartão consumido e com um saldo na conta maior que R$10,000.00 = Ranking “B”.
- Clientes entre 80% e 90% do limite do cartão consumido e com um saldo na conta maior que R$15,000.00 = Ranking “C”.
- Clientes entre 90% e 95% do limite do cartão consumido e com um saldo na conta maior que R$20,000.00 = Ranking “D”.

Todos os demais clientes que não se enquadram em nenhuma categoria devem ficar com o Ranking vazio, podendo ser removidos da base de dados.

Resultados esperados da Fase 1

Ao final da fase 1, espera-se que você tenha sido capaz de carregar os dados presentes nas diferentes bases de dados fornecidas, bem como manipulá-los utilizando Querys SQL. Caso você não consiga realizar a query completa, não se preocupe, faça tudo aquilo que for capaz da melhor forma possível.

Os conjuntos de dados fornecidos não são tão grandes. Mesmo assim, caso você tenha limitações de memória na sua máquina pessoal, você pode reduzir a quantidade de linhas no momento da leitura dos dados.

Por fim, você precisa descobrir uma forma de salvar os dados resultantes da primeira e da segunda query em arquivos CSV, visto que esses arquivos serão utilizados na segunda fase do projeto. Cada query deve ter seu próprio arquivo com os respectivos dados. Por exemplo: *query1_data.csv*, *query2_data.csv*.

Fase 2

A segunda fase do projeto consiste na criação de uma pipeline ETL para tratar os dados que foram extraídos anteriormente. Sua pipeline deve possuir uma etapa de pré-processamento, uma de adição de informações, e o salvamento dos resultados.

O **Apache Beam** é o modelo de dados padrão que utilizamos no Bullla para criar pipelines desse tipo e, por conta disso, é a ferramenta preferível. Ainda assim, você tem liberdade para escolher outras bibliotecas e/ou frameworks para criar sua pipeline, caso prefira.

A etapa de pré-processamento deve ser responsável por tratar os dados e prepará-los para serem disponibilizados na versão final. Para isso, o seu pré-processamento de dados deve ser capaz de:

- Remover quaisquer clientes duplicados
- Remover acentuação e espaços em branco das strings

- Padronizar todos os caracteres em letra maiúscula

Você é livre para utilizar sua criatividade e inventar outras funções para pré-processar os dados e garantir que eles estarão adequados para consumo.

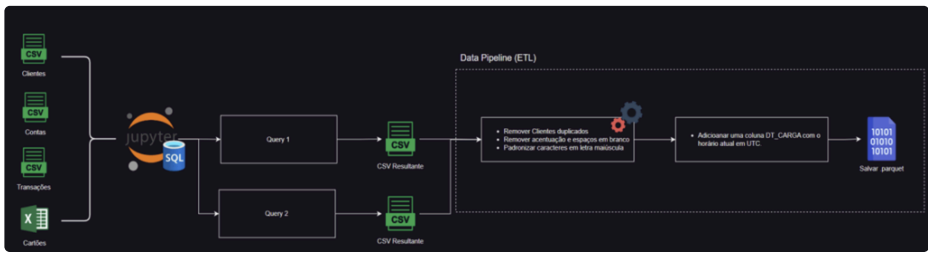
Na fase de adição, sua função deve criar uma “coluna” nos dados, chamada “DT_CARGA”. A coluna DT_CARGA, como o nome sugere, deve possuir a data e horário atual, considerando o fuso horário UTC.

Por fim, você deve salvar os dados processados em arquivos colunares no formato *parquet*.

Resultados esperados da Fase 2

Ao final da segunda fase, espera-se que você tenha conseguido criar uma pipeline em Python que seja capaz de carregar, tratar, modificar e salvar os dados resultantes. Embora seja possível concluir essa etapa utilizando diferentes abordagens, é essencial que todas as boas práticas de engenharia de software e código limpo sejam seguidas, pois isso será parte fundamental da avaliação.

Ao final da pipeline, você deve possuir dois arquivos com os dados tratados. Por exemplo: *query1_data.parquet*, *query2_data.parquet*. A arquitetura completa do projeto deve ser algo parecido com a Figura abaixo:



Arquitetura resultante.