

1. SQL + Python + ETL

Você tem a tabela raw “raw_data.csv”*, ela carrega dados de navegação do usuário. Infelizmente a tabela é muito pesada para utilizarmos em rotinas diárias. A tabela “transactions.csv” traz informações sobre qual compra foi realmente paga. Precisamos criar um fluxo de dados eficiente que nos retorne os dados de todos os clientes que realizaram uma compra paga.

Esses servirão para alimentar uma plataforma de mídia e precisamos enviá-los em forma de um payload.

- a. Você deve desenhar e explicar o fluxo de dados eficiente cruzando as duas tabelas e explicando como funciona essa rotina
- b. Crie um código que vai tratar a “raw_data” e vai transformar numa tabela somente com as compras realizadas “web_orders”, ela deve ser usada nas próximas questões
- c. Crie uma tabela agregada com os pedidos que tenha *data do evento*, *client_id*, *session_id*, *transaction_id*, *status do pedido* e *origem (source)*.
- d. Usando um ambiente python, utilize os dados gerados acima e monte um payload com os dados dos usuários que realizaram uma compra e a pagaram. As variáveis que devem ser enviadas são **client_id**, **session_id**, **value**, **product_category**.

**as tabelas foram geradas com dados aleatórios e podem não ter correlação com uma estratégia real*

2. Data Visualization

Crie uma visão com os dados que manipulou acima e adicione a tabela ‘media_spend’. Nela você encontra os investimentos por dia por mídia. Nessa visão será necessário colocar:

- Evolução diária de Resultados (compras, compras pagas, receita[revenue] e investimento)
- Visão de compras foram realizadas comparando com as compras realmente pagas (taxa de pagamento)
- Uma vista com o acumulado de resultados do período por source e adicionando CPA (investimento/compras pagas) e ROAS (receita/investimento)
- Uma vista de vendas por categoria de produto

Coloque o material executado em um documento e deixe-o preparado para a apresentação do case