Bom dia! Excelente iniciativa. Aprofundar-se em SQL através de um projeto prático e estruturado é, sem dúvida, a melhor maneira de consolidar o conhecimento e se preparar para os desafios do mercado de trabalho.

Preparei um case completo para você, o "Desafio de Dados da Varejo Crescer". Ele foi desenhado para cobrir todas as etapas que você mencionou, desde a engenharia de dados até a análise final, com desafios realistas pelo caminho.

Vamos começar!

Passo 1: O Desafio de Negócio - "Varejo Crescer"

Contexto:

A "Varejo Crescer" é uma empresa de médio porte que vende produtos eletrônicos e de informática. Ela começou com lojas físicas e recentemente expandiu para o e-commerce. Atualmente, os dados de vendas estão isolados em dois sistemas diferentes: um para as lojas físicas e outro para a plataforma online.

A diretoria está com dificuldade para ter uma visão unificada do negócio. Eles não conseguem responder perguntas simples como:

- Qual nosso faturamento total consolidado?
- Qual produto vende mais, independentemente do canal?
- O perfil do nosso cliente online é diferente do cliente da loja física?
- Existe sazonalidade nas nossas vendas?

Sua Missão:

Você foi contratado(a) como Analista de Dados para construir o primeiro ambiente analítico (Data Mart) da empresa. Seu objetivo é unificar os dados, modelá-los de forma otimizada para análises e, finalmente, gerar insights que respondam às perguntas da diretoria.

Passo 2: Os Dados Brutos (Mock Data)

Aqui estão os dados como seriam extraídos dos sistemas de origem. Eles estão em formato de texto (CSV) para você simular um processo de ingestão. Você deve criar arquivos .csv com esses conteúdos.

Tabela 1: vendas_online.csv

(Dados do sistema de e-commerce)

Snippet de código

id_transacao,id_cliente,id_produto,quantidade,valor_unitario,data_venda,cupom_desconto
ONL-20240115-001,C085,P004,1,3500.00,15/01/2024,NATAL10
ONL-20240116-002,C123,P002,1,2200.00,2024-01-16,NULL
ONL-20240210-003,C085,P005,2,150.00,2024-02-10,PROMOFEB
ONL-20240212-004,C250,P001,1,4500.00,12/02/2024,NULL
ONL-20240305-005,C301,P007,1,89.90,2024-03-05,NULL
ONL-20240305-005,C301,P007,1,89.90,2024-03-05,NULL

```
ONL-20240420-006,C123,P003,1,5500.00,20/04/2024,FRETEGRATIS
ONL-20240501-007,C412,P010,1,1200.00,2024-05-01,NULL
ONL-20240502-008,C085,P004,1,3600.00,2024-05-02,MAI15
```

Desafios nesta base:

- 1. Formato de data inconsistente: dd/mm/yyyy e yyyy-mm-dd.
- 2. Valores NULOS: A coluna cupom_desconto tem valores NULL.
- 3. Transação duplicada: A transação ONL-20240305-005 está repetida.

Tabela 2: vendas_loja_fisica.csv

(Dados do sistema de PDV das lojas)

Snippet de código

```
id_venda,id_loja,id_vendedor,id_cliente,id_produto,qtd,preco_final,data_venda
LF001,LJ01,V01,C085,P001,1,4500.00,2024-01-20
LF002,LJ02,V04,C301,P006,1,300.00,2024-01-25
LF003,LJ01,V02,C555,P002,1,2250.50,2024-02-18
LF004,LJ03,V05,C123,P008,2,79.99,2024-03-10
LF005,LJ02,,C412,P009,1,45.00,2024-03-11
LF006,LJ01,V01,C085,P003,1,-5500.00,2024-04-05
LF007,LJ03,V06,C301,P007,3,89.90,2024-05-12
```

Desafios nesta base:

- 1. Nomes de colunas diferentes: quantidade VS qtd , valor_unitario VS preco_final .
- 2. Dados Faltantes: A venda LF005 não tem id vendedor.
- 3. Dados Inválidos: A venda LF006 tem um preco_final negativo, o que é impossível.

Tabela 3: cadastro clientes.csv

Snippet de código

```
id_cliente,nome,email,data_cadastro,cidade,estado
C085,Ana Silva,ana.silva@email.com,2023-05-12,São Paulo,SP
C123,Bruno Costa,bruno.c@email.com,2023-07-20,Rio de Janeiro,RJ
C250,Carlos Dias,c.dias@email.com,2023-11-01,sao paulo,São Paulo
C301,Daniela Souza,dani.souza@email.com,2024-01-05,Belo Horizonte,MG
C412,Eduarda Lima,edu.lima@email.com,2024-02-28,Porto Alegre,RS
C555,Fernanda Alves,fer.alves@email.com,2024-03-15,Curitiba,PR
```

Desafios nesta base:

1. Inconsistência de formatação: "São Paulo" e "sao paulo".

2. Estado com nome completo e sigla: "São Paulo" e "SP".

Tabela 4: catalogo_produtos.csv

Snippet de código

id_produto,nome_produto,categoria,fornecedor,custo_unitario
P001,Notebook Pro X1,Notebook,TechCorp,3800.00
P002,Notebook Gamer Z,notebook,GameMax,1800.00
P003,Monitor UltraWide 4K,Monitor,Display soluções,4200.00
P004,Smartphone Alpha 10,Celular,MobileFast,2800.00
P005,Teclado Mecânico RGB,Acessório,Clicky,120.00
P006,Mouse Gamer Pro,Acessório,Clicky,250.00
P007,Cabo HDMI 2.1,Cabo,ConectaTudo,60.00
P008,Webcam HD,Acessório,Vision,70.00
P009,Pen Drive 64GB,Armazenamento,DataSafe,35.00
P010,SSD Externo 1TB, armazenamento,DataSafe,950.00

Desafios nesta base:

1. **Inconsistência de formatação em categoria:** "Notebook" e "notebook", "Armazenamento" e "armazenamento".

Tabela 5: lista_lojas.csv

Snippet de código

id_loja,nome_loja,cidade,estado
LJ01,Shopping Center Norte,São Paulo,SP
LJ02,Barra Shopping,Rio de Janeiro,RJ
LJ03,BH Shopping,Belo Horizonte,MG

Passo 3: Modelagem de Dados (Star Schema)

Antes de criar as tabelas, desenhe o modelo Star Schema. O objetivo é ter uma tabela central de fatos (com as métricas) e tabelas de dimensão ao redor (com os contextos).

- Tabela Fato (Fact Table): Fato Vendas
 - Métricas: quantidade_vendida , valor_total_venda , custo_total_venda , lucro_venda .
 - Chaves Estrangeiras (Foreign Keys): sk_cliente , sk_produto , sk_loja , sk_data , sk canal .
- Tabelas de Dimensão (Dimension Tables):
 - Dim_Cliente: sk_cliente (chave substituta/surrogate key), id_cliente (chave de negócio), nome, email, cidade, estado.

- Dim_Produto: sk_produto, id_produto, nome_produto, categoria, fornecedor.
- Dim_Loja: sk_loja, id_loja, nome_loja, cidade, estado. (**Dica:** Inclua um registro para "Online" aqui).
- Dim_Data: sk_data, data_completa, ano, mes, dia, dia_da_semana, trimestre, nome mes.
- Dim_Canal: sk_canal, nome_canal ("Online", "Loja Física").

Passo 4: Implementação (Medallion Architecture)

No seu banco de dados PostgreSQL, crie 3 schemas: bronze, silver e gold.

1. Camada Bronze (Raw Data)

Crie tabelas no schema bronze que espelhem exatamente a estrutura dos arquivos CSV. Use o tipo de dado TEXT para todas as colunas para garantir que a ingestão não falhe por tipos de dados incorretos.

- bronze.vendas_online
- bronze.vendas_loja_fisica
- bronze.clientes
- bronze.produtos
- bronze.lojas

Sua Tarefa:

Use o comando copy do PostgreSQL para carregar os dados dos seus arquivos .csv para estas tabelas.

SQL

COPY bronze.vendas_online FROM '/caminho/para/seu/arquivo/vendas_online.csv' DELIMITER ', -- Repita para todas as outras tabelas

2. Camada Silver (Cleaned & Transformed Data)

Aqui você resolve todos os desafios dos dados. Crie tabelas no schema silver com tipos de dados corretos e dados já limpos e padronizados.

- silver.vendas_unificadas : Uma única tabela com todas as vendas.
- silver.clientes: Tabela de clientes com nomes de cidades e estados padronizados.
- silver.produtos: Tabela de produtos com categorias padronizadas.
- silver.lojas: Tabela de lojas.

Sua Tarefa (Exemplos de Lógica SQL):

Para silver.clientes:

```
INSERT INTO silver.clientes
SELECT
   id_cliente,
   nome,
   email,
   data_cadastro::DATE,
   INITCAP(TRIM(LOWER(cidade))) AS cidade, -- Padroniza "sao paulo" -> "Sao Paulo"
   CASE
       WHEN estado = 'São Paulo' THEN 'SP'
       ELSE UPPER(estado)
   END AS estado
FROM bronze.clientes:
```

- Para silver.vendas_unificadas:
 - Use UNION ALL para juntar as duas tabelas de vendas.
 - Use **COALESCE** ou **CASE** WHEN para tratar as datas inconsistentes.
 - Use CAST para converter os tipos de dados (ex: valor_unitario::NUMERIC(10,2)).
 - Remova duplicatas usando DISTINCT ou ROW_NUMBER().
 - Filtre dados inválidos (ex: WHERE preco_final > 0).
 - Crie uma coluna canal_venda ('Online' ou 'Loja Física').

3. Camada Gold (Business Ready / Star Schema)

Crie as tabelas do seu modelo Star Schema no schema gold .

Sua Tarefa:

- Crie as tabelas Dim_Cliente , Dim_Produto , Dim_Loja , Dim_Data , Dim_Canal e Fato_Vendas . Use SERIAL PRIMARY KEY para as surrogate keys (sk_).
- **Popular a** Dim_Data: Crie uma função ou use generate_series para popular a dimensão de tempo com todas as datas do seu intervalo de vendas.

SQL

```
INSERT INTO gold.dim_data (data_completa, ano, mes, dia, dia_da_semana, trimestre, nor
SELECT
    datum AS data_completa,
    EXTRACT(YEAR FROM datum) AS ano,
    EXTRACT(MONTH FROM datum) AS mes,
    EXTRACT(DAY FROM datum) AS dia,
    EXTRACT(ISODOW FROM datum) AS dia_da_semana, -- 1 para Segunda, 7 para Domingo
    EXTRACT(QUARTER FROM datum) AS trimestre,
```

```
TO_CHAR(datum, 'Month') AS nome_mes FROM GENERATE_SERIES('2024-01-01'::DATE, '2024-12-31'::DATE, '1 day'::INTERVAL) datum
```

- **Popular as outras Dimensões:** Use INSERT INTO ... SELECT DISTINCT ... a partir das tabelas silver.
- Popular a Fato_Vendas: Esta é a etapa mais complexa. Você precisará fazer JOIN da sua tabela silver.vendas_unificadas com todas as tabelas de dimensão da camada gold para buscar as sk_ correspondentes e calcular as métricas.

Passo 5: Análises e Perguntas de Negócio (SQL em Ação)

Com o modelo pronto na camada gold , responda às seguintes perguntas usando consultas SQL:

- - Qual o faturamento (valor total) e lucro total por mês?
 - Existe um dia da semana que vende mais?
 - Como as vendas se comportaram no primeiro trimestre de 2024?

Exemplo de consulta para começar:

SQL

```
SELECT
    d.ano,
    d.mes,
    d.nome_mes,
    SUM(f.valor_total_venda) AS faturamento_mensal
FROM gold.fato_vendas f
JOIN gold.dim_data d ON f.sk_data = d.sk_data
GROUP BY d.ano, d.mes, d.nome_mes
ORDER BY d.ano, d.mes;
```

2. Análise de Produto:

- Quais são os 5 produtos mais vendidos em quantidade?
- Quais são os 5 produtos que geraram mais lucro?
- Qual a categoria de produto mais rentável?

3. Análise de Cliente e Canal:

- Qual o faturamento por canal de venda (Online vs. Loja Física)?
- Qual o ticket médio (valor total da venda / quantidade de vendas) por canal?
- Quais são os 10 clientes que mais compraram (em valor)?
- De qual estado vêm a maioria das nossas vendas?

Passo 6: Visualização (Produto Final no Power BI)

Agora, vamos transformar os dados em um produto final visual.

- 1. Conecte o Power BI ao seu banco de dados PostgreSQL.
- 2. Importe as tabelas do schema gold (as Fatos e Dimensões).
- 3. **Crie as relações no Power Bl** entre a Fato_Vendas e as Dimensões, usando as colunas sk_ .
- 4. Crie um Dashboard chamado "Análise de Vendas Varejo Crescer" contendo:
 - Cartões (KPIs): Faturamento Total, Lucro Total, Quantidade de Vendas, Número de Clientes Únicos.
 - **Gráfico de Linha:** Faturamento Mensal (eixo X: Mês/Ano, eixo Y: Soma do valor total venda). Isso mostrará sua análise de série temporal.
 - **Gráfico de Barras Empilhadas:** Top 5 Produtos por Faturamento, com as barras divididas por Canal (Online/Loja Física).
 - Gráfico de Pizza (ou Treemap): Vendas por Categoria de Produto.
 - Mapa: Vendas por Estado do Cliente.
 - **Tabela:** Detalhes dos Top 10 Clientes por Lucro.
 - **Filtros (Slicers):** Adicione filtros por Ano, Mês e Canal para que a diretoria possa explorar os dados dinamicamente.

Este case te dará uma experiência completa, passando por desafios de qualidade de dados, modelagem, transformação (ETL/ELT com SQL) e visualização. Ao final, você terá um portfólio sólido para demonstrar suas habilidades.

Mãos à obra e bom projeto! Se tiver qualquer dúvida em alguma das etapas, pode perguntar.