Olá, candidato!

Obrigado pelo interesse em fazer parte do time Rox!

Esse é um teste com o objetivo de conhecer um pouco mais sobre a sua forma de trabalhar com dados e a resolução de problemas que envolvem engenharia de dados.

### O problema

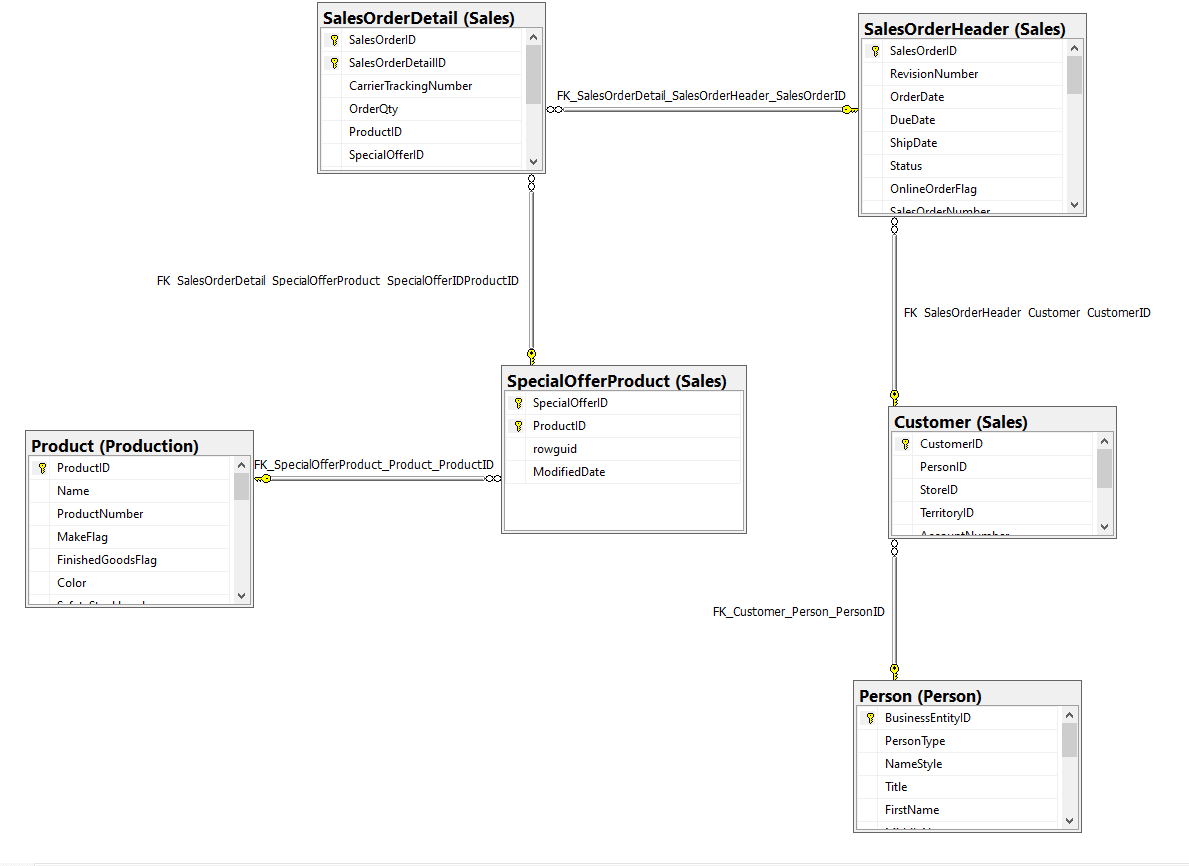
O presente problema se refere aos dados de uma empresa que produz bicicletas.

O objetivo deste desafio é compreender os seus conhecimentos e experiência analisando os seguintes aspectos:

1. Fazer a modelagem conceitual dos dados;
2. Criação da infraestrutura necessária;
3. Criação de todos os artefatos necessários para carregar os arquivos para o banco criado;
4. Desenvolvimento de SCRIPT para análise de dados;
5. (opcional) Criar um relatório em qualquer ferramenta de visualização de dados.

Os seguintes arquivos devem ser importados (ETL) para o banco de dados de sua escolha:

* Sales.SpecialOfferProduct.csv
* Production.Product.csv
* Sales.SalesOrderHeader.csv
* Sales.Customer.csv
* Person.Person.csv
* Sales.SalesOrderDetail.csv



### Resolução

Para a realização deste teste, você deve utilizar alguma das principais nuvens públicas (Azure, AWS, GCP). Pedimos também que compartilhe conosco o código fonte (preferencialmente através de um repositório git público).

A nossa expectativa é um processo de ingestão de dados utilizando os serviços da nuvem escolhida.

Além disso, inclua um arquivo README.md onde você deve compartilhar as decisões de arquitetura, implementação e instruções sobre como executar o software. Caso tenha criado um relatório em alguma ferramenta de visualização de dados, inclua um screenshot do mesmo no arquivo.

### Análise de dados

Com base na solução implantada responda aos seguintes questionamentos:

1. Escreva uma query que retorna a quantidade de linhas na tabela Sales.SalesOrderDetail pelo campo SalesOrderID, desde que tenham pelo menos três linhas de detalhes.

**Resposta:** SELECT SalesOrderID,

COUNT(\*)

FROM [importCSV].[dbo].[Sales.SalesOrderDetail]

GROUP BY SalesOrderID

HAVING COUNT (SalesOrderID) <= 3

1. Escreva uma query que ligue as tabelas Sales.SalesOrderDetail, Sales.SpecialOfferProduct e Production.Product e retorne os 3 produtos (Name) mais vendidos (pela soma de OrderQty), agrupados pelo número de dias para manufatura (DaysToManufacture).

**Resposta: SELECT**

**A.SalesOrderID,**

**B.SpecialOfferID,**

**C.Productid**

**FROM [importCSV].[dbo].[Sales.SalesOrderDetail] a**

**LEFT JOIN [importCSV].[dbo].[Sales.SpecialOfferProduct] B ON B.SpecialOfferID = A.SalesOrderID**

**LEFT JOIN [importCSV].[dbo].[Production.Product ] C ON C.Productid = b.ProductID**

1. Escreva uma query ligando as tabelas Person.Person, Sales.Customer e Sales.SalesOrderHeader de forma a obter uma lista de nomes de clientes e uma contagem de pedidos efetuados.

**Resposta:** SELECT \*

FROM [importCSV].[dbo].[Person.Person] a

LEFT JOIN [importCSV].[dbo].[Sales.Customer] b on b.PersonID = a.[ID de entidade comercial]

LEFT JOIN [importCSV].[dbo].[Sales.SalesOrderHeader] c on c.SalesOrderID = b.CustomerID

1. Escreva uma query usando as tabelas Sales.SalesOrderHeader, Sales.SalesOrderDetail e Production.Product, de forma a obter a soma total de produtos (OrderQty) por ProductID e OrderDate.

**Resposta:** SELECT c.ProductID,

c.Name,

sum(OrderQty),

b.OrderDate,

sum(OrderQty)

FROM [importCSV].[dbo].[Sales.SalesOrderDetail] a

INNER JOIN [importCSV].[dbo].[Sales.SalesOrderHeader] b ON b.SalesOrderID = a.SalesOrderID

INNER JOIN [importCSV].[dbo].[Production.Product] c ON c.ProductID = a.ProductID

GROUP BY a.ProductID, b.OrderDate

ORDER BY b.OrderDateINNER JOIN [importCSV].[dbo].Production.Products c ON c.ProductID = a.ProductID

GROUP BY a.ProductID, b.OrderDate

ORDER BY b.OrderDate

1. Escreva uma query mostrando os campos SalesOrderID, OrderDate e TotalDue da tabela Sales.SalesOrderHeader. Obtenha apenas as linhas onde a ordem tenha sido feita durante o mês de setembro/2011 e o total devido esteja acima de 1.000. Ordene pelo total devido decrescente.

**Resposta:** SELECT SalesOrderID,

OrderDate,

TotalDue

FROM [importCSV].[dbo].[Sales.SalesOrderHeader]

WHERE OrderDate BETWEEN '2011-09-01 00:00:00' and '2011-09-30 00:00:00'and TotalDue > 1000

ORDER BY TotalDue desc