Olá, candidato!

Alessandro de Mattos Silva

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

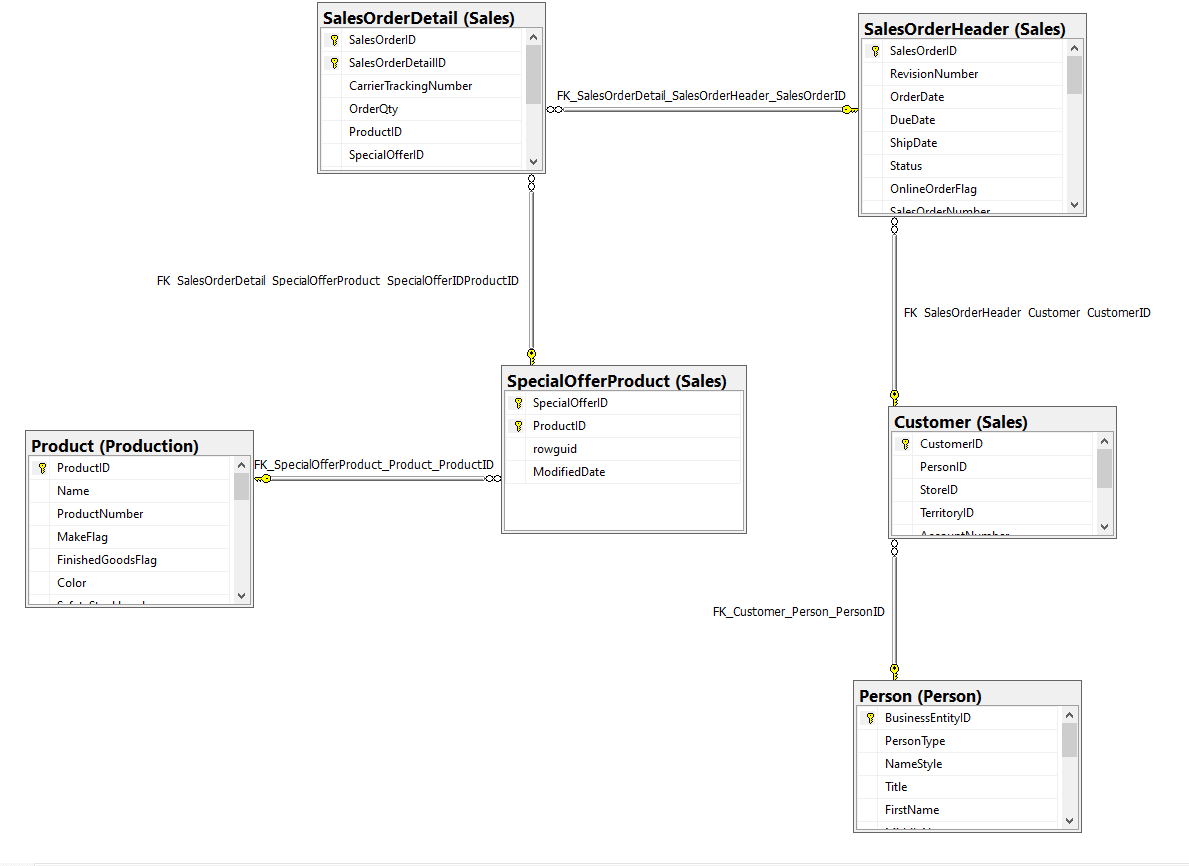
O presente problema se refere aos dados de uma empresa que produz bicicletas.

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.

SELECT

  COUNT(SalesOrderID)

FROM

  `rox-bike.BIKE.SalesOrderDetail`

WHERE

  OrderQty > 2

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).

SELECT P.Name, SUM(O.OrderQty) As Vendas

FROM `rox-bike.BIKE.Product` AS P

INNER JOIN `rox-bike.BIKE.SpecialOfferProduct` AS S

    ON P.ProductID=S.ProductID

INNER JOIN `rox-bike.BIKE.SalesOrderDetail` AS O

    ON S.ProductID=O.ProductID

GROUP BY P.DaysToManufacture, P.Name, O.OrderQty

ORDER BY  O.OrderQty DESC

LIMIT 3

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.

SELECT  P.FirstName, P.LastName, COUNT(H.CustomerID) AS Pedidos

FROM `rox-bike.BIKE.Person` AS P

INNER JOIN `rox-bike.BIKE.Customer` AS C

    ON P.BusinessEntityID=C.PersonID

INNER JOIN `rox-bike.BIKE.SalesOrderHeader` AS H

    ON C.CustomerID=H.CustomerID

GROUP BY H.CustomerID, P.FirstName, P.LastName

ORDER BY Pedidos DESC

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.

SELECT  P.ProductID, H.OrderDate, SUM(D.OrderQty) AS SOMAPRODUTOS

FROM `rox-bike.BIKE.SalesOrderDetail` AS D

INNER JOIN `rox-bike.BIKE.SalesOrderHeader` AS H

    ON D.SalesOrderID=H.SalesOrderID

INNER JOIN `rox-bike.BIKE.Product` AS P

    ON D.ProductID=P.ProductID

GROUP BY P.ProductID, H.OrderDate, D.OrderQty

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.

SELECT  H.SalesOrderID, H.OrderDate, H.TotalDue

FROM `rox-bike.BIKE.SalesOrderHeader` AS H

WHERE H.TotalDue > 1000 AND H.OrderDate BETWEEN "2011-09-01" AND "2011-09-30"

ORDER BY H.TotalDue DESC