

# Estudo de Caso - ROX



Clarissa Souza  
[www.linkedin.com/in/clarissasouza950](https://www.linkedin.com/in/clarissasouza950)  
<https://github.com/clarissa-souza/Desafio-rox>  
[clarissasouza950@gmail.com](mailto:clarissasouza950@gmail.com)

# Requisitos

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.

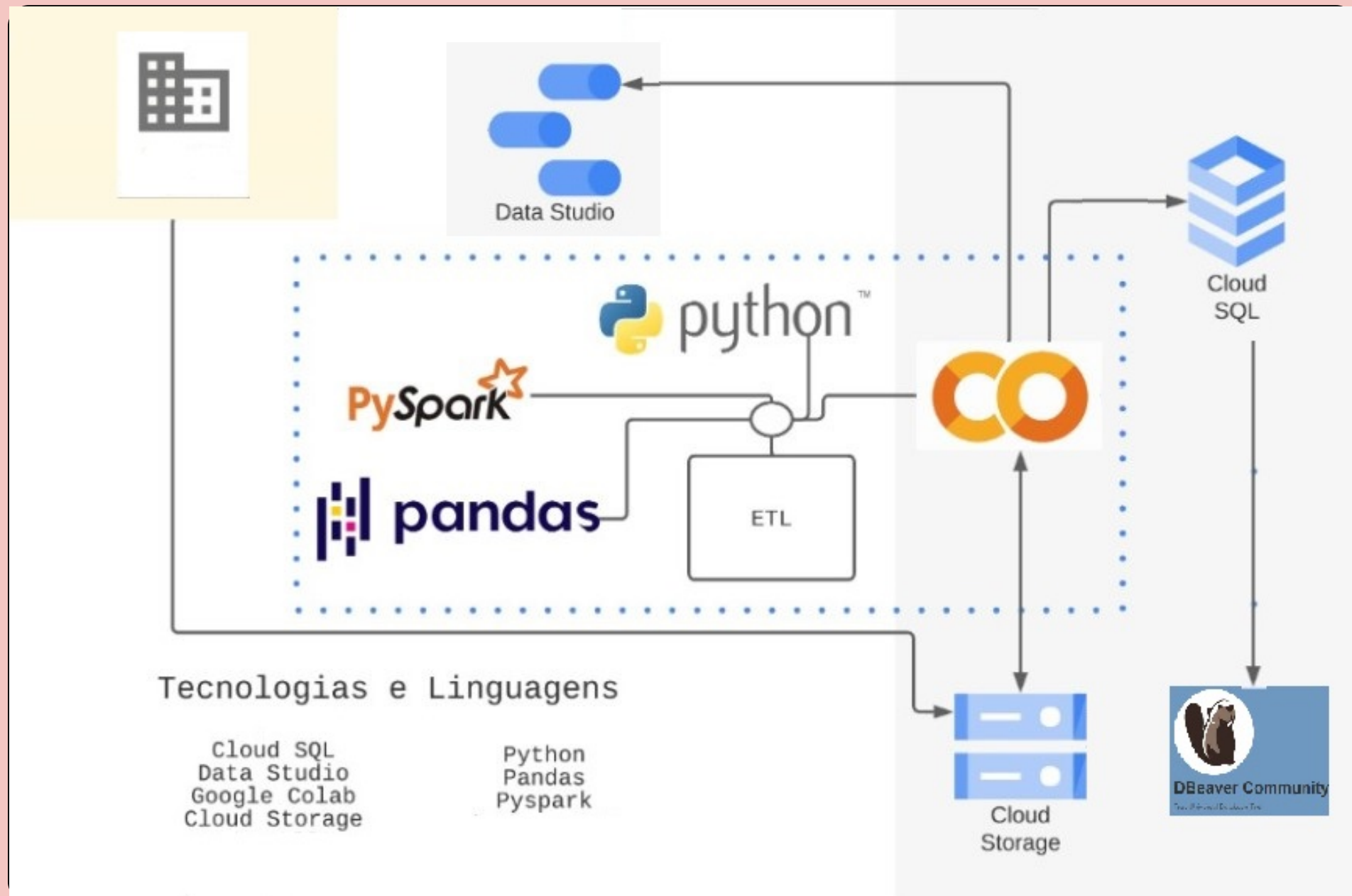
# DATASETS

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

# Cenário

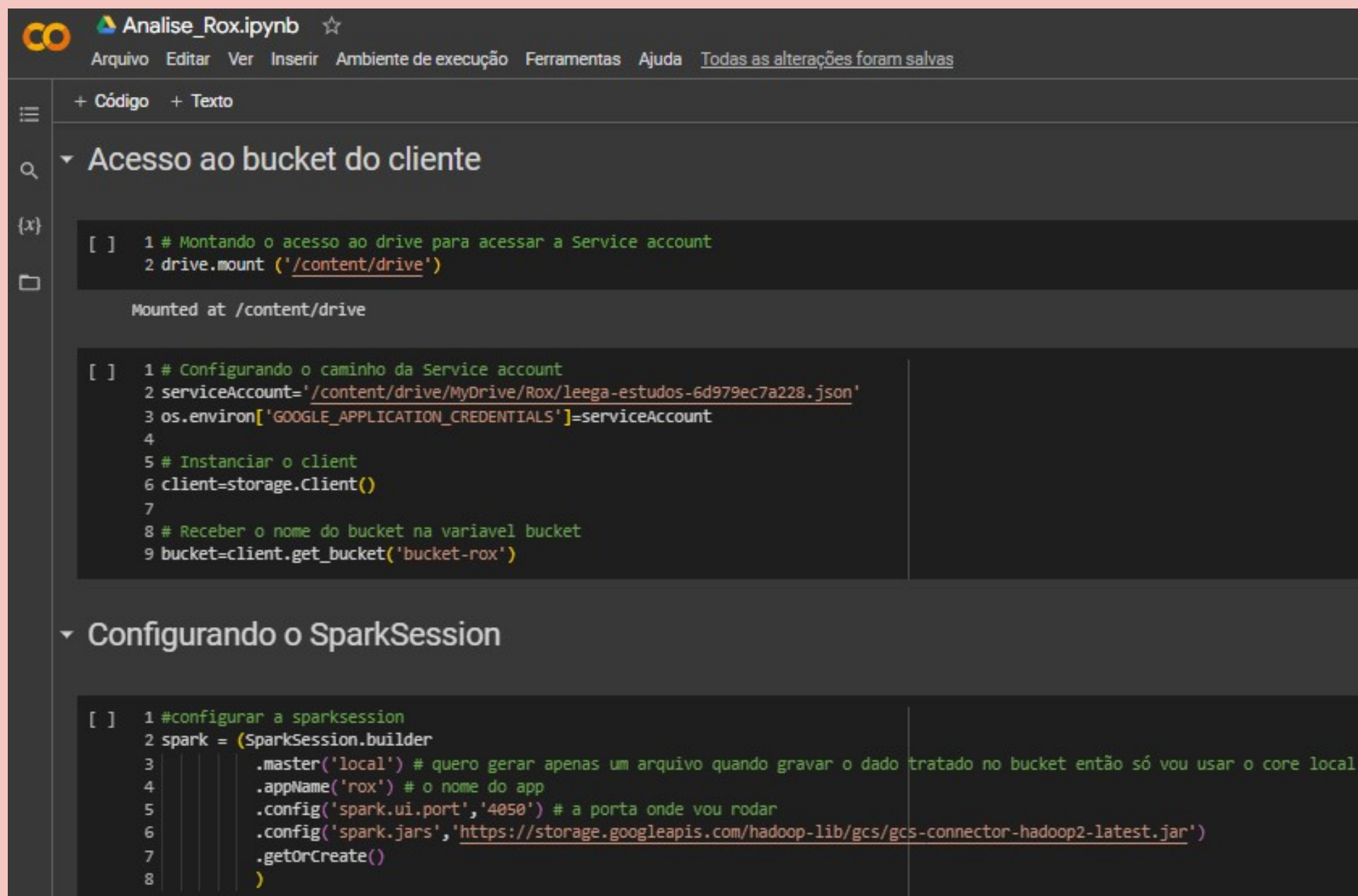
- 1 - Cliente faz o upload de seus Datasets no Datalake do GCP no diretório Dados Originais.
- 2 - Os arquivos são analisados pelo Analise\_ROX no Colab e os arquivos tratados são gravados no Datalake do cliente no diretório Dados Tratados.
- 3 - Pelo GCP foi criada a instância do banco rox. Neste momento também foi feita a configuração necessária para permitir o acesso externo.
- 4 - O Banco\_ROX no Colab, é o responsável por criar o banco dbrox, suas tabelas e consultas.
- 5 - Foi instalado o DBeaver na máquina local para acessar o banco do GCP e fazer consultas direto pelo SQL

# Workflow e tecnologias utilizadas



# Operações no Colab: Análise\_ROX

A análise consiste em identificar se existe e quais são as Primary Key e Foreign Key e o relacionamento entre as tabelas. Verificar se as colunas estão devidamente formatadas. A modelagem do banco pode ser visto neste colab.



```
Analise_Rox.ipynb ☆
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

Acesso ao bucket do cliente

[ ] 1 # Montando o acesso ao drive para acessar a Service account
    2 drive.mount('/content/drive')

Mounted at /content/drive

[ ] 1 # Configurando o caminho da Service account
    2 serviceAccount='/content/drive/MyDrive/Rox/leega-estudos-6d979ec7a228.json'
    3 os.environ['GOOGLE_APPLICATION_CREDENTIALS']=serviceAccount
    4
    5 # Instanciar o client
    6 client=storage.Client()
    7
    8 # Receber o nome do bucket na variavel bucket
    9 bucket=client.get_bucket('bucket-rox')

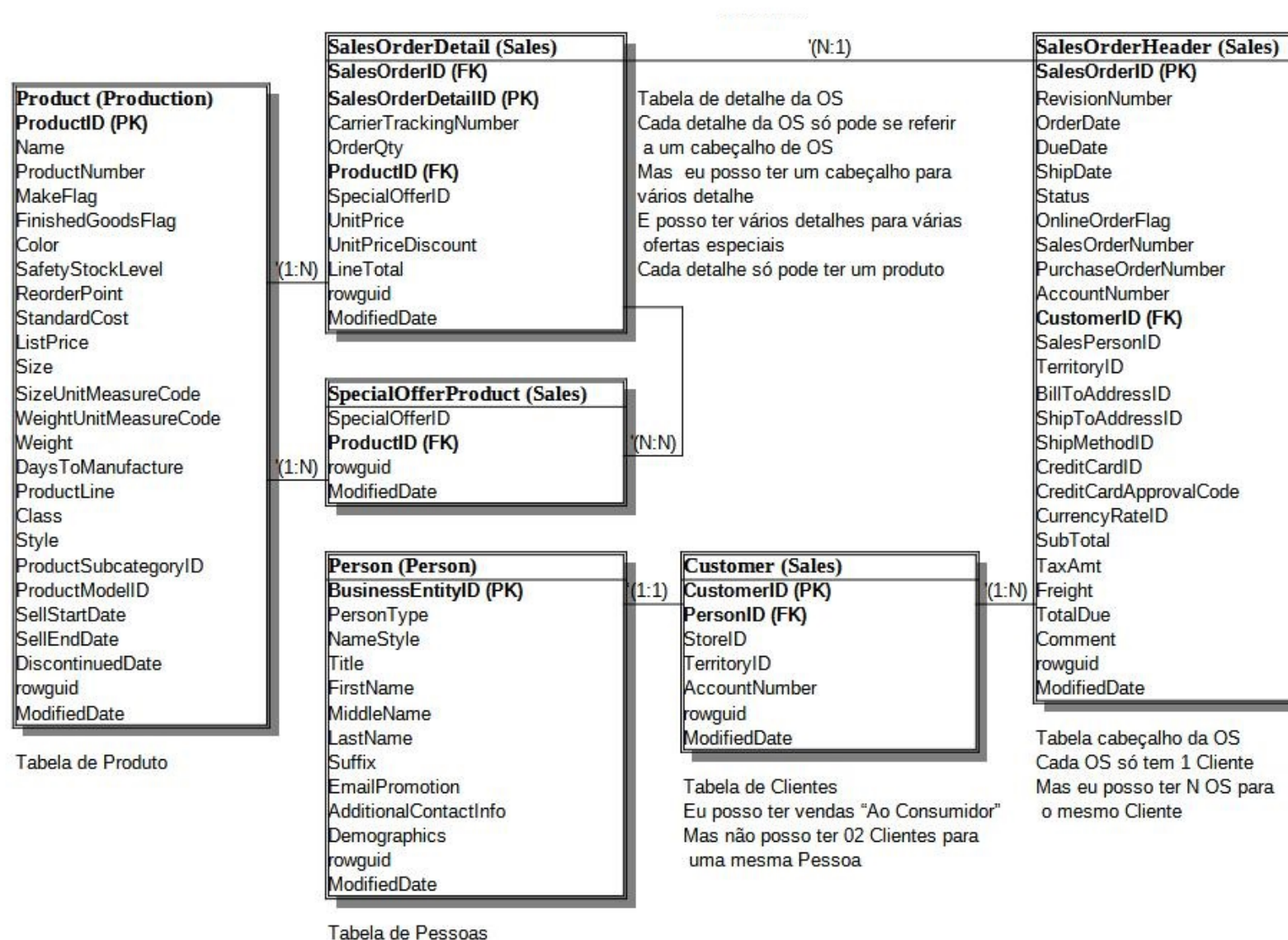
Configurando o SparkSession

[ ] 1 #configurar a sparksession
    2 spark = (SparkSession.builder
    3         .master('local') # quero gerar apenas um arquivo quando gravar o dado tratado no bucket então só vou usar o core local
    4         .appName('rox') # o nome do app
    5         .config('spark.ui.port','4050') # a porta onde vou rodar
    6         .config('spark.jars','https://storage.googleapis.com/hadoop-lib/gcs/gcs-connector-hadoop2-latest.jar')
    7         .getOrCreate()
    8         )
```



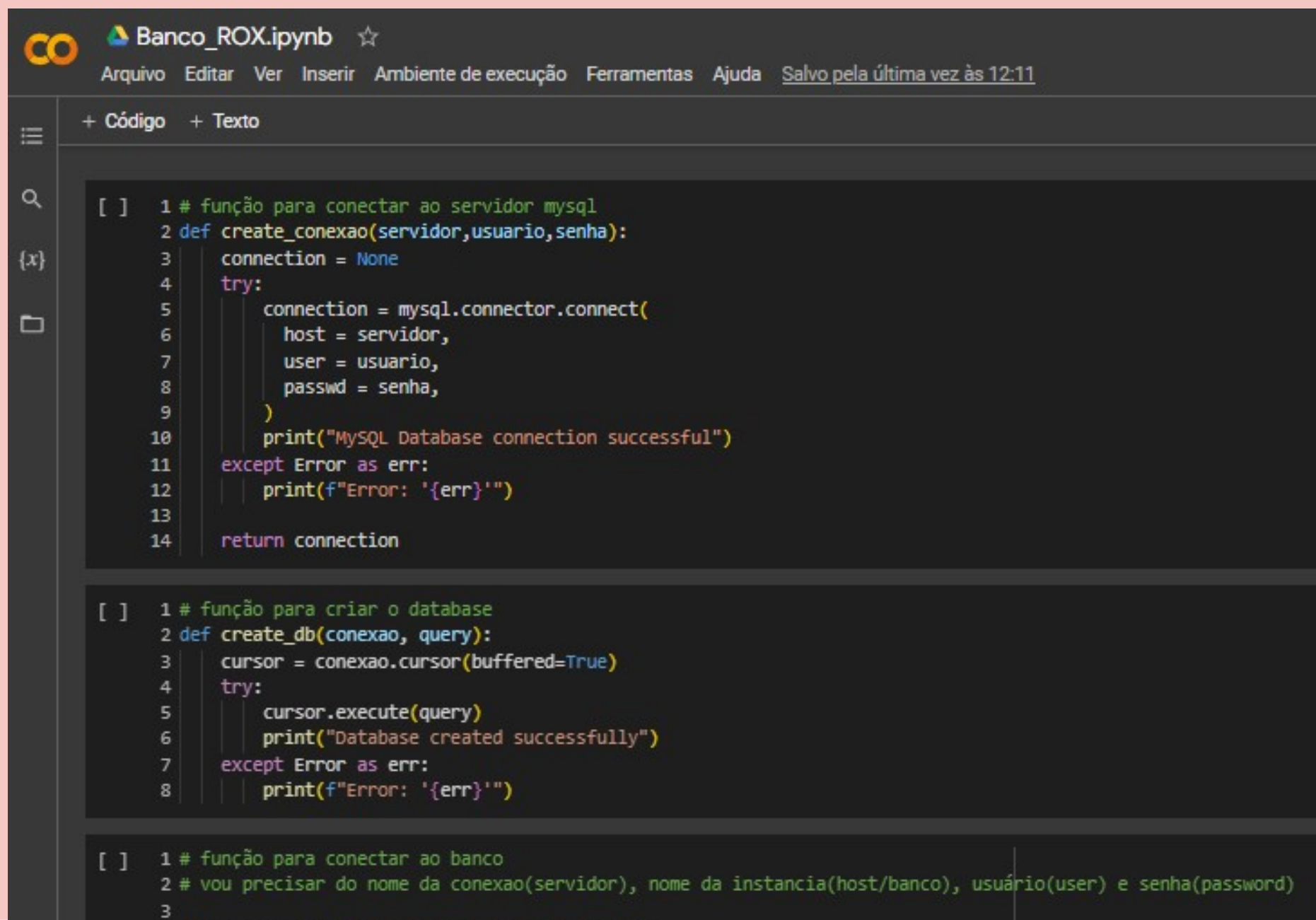
# Operações no Colab: Análise\_ROX

A análise consiste em identificar se existe e quais são as Primary Key e Foreign Key e o relacionamento entre as tabelas. Verificar se as colunas estão devidamente formatadas. A modelagem do banco pode ser visto neste colab.



# Operações no Colab: Banco\_ROX

*Neste colab você vai encontrara todas as funções necessárias para criar o banco e criar, popular, alterar e consultar as tabelas. Todas as consultas exigidas estão nesse colab*



```
CO Banco_ROX.ipynb ☆
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Salvo pela última vez às 12:11

+ Código + Texto

[ ] 1 # função para conectar ao servidor mysql
    2 def create_conexao(servidor,usuario,senha):
    3     connection = None
    4     try:
    5         connection = mysql.connector.connect(
    6             host = servidor,
    7             user = usuario,
    8             passwd = senha,
    9         )
    10         print("MySQL Database connection successful")
    11     except Error as err:
    12         print(f"Error: '{err}'")
    13
    14     return connection

[ ] 1 # função para criar o database
    2 def create_db(conexao, query):
    3     cursor = conexao.cursor(buffered=True)
    4     try:
    5         cursor.execute(query)
    6         print("Database created successfully")
    7     except Error as err:
    8         print(f"Error: '{err}'")

[ ] 1 # função para conectar ao banco
    2 # vou precisar do nome da conexao(servidor), nome da instancia(host/banco), usuário(user) e senha(password)
    3
```



# Consultas no DBeaver

DBeaver 22.3.0 - <dbrox> Script-2

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto dbrox dbrox

Database Navigator Projects

Enter a part of object name here

dbrox -

- Databases
  - dbrox
    - Tables
      - customer 4M
      - person 17M
      - production 144K
      - salesorderdetail 28M
      - salesorderheader 12M
      - specialofferproduct 112K
    - Views
    - Indexes
    - Procedures
    - Triggers
    - Events
  - sys
  - Users
  - Administer
  - System Info

Project - General

Name DataSource

- Bookmarks
- Diagrams
- Scripts

```
SELECT production.ProductID, production.Name, salesorderheader.OrderDate, sum(salesorderdetail.OrderQty)
FROM salesorderheader
inner JOIN salesorderdetail
ON salesorderdetail.SalesOrderID = salesorderheader.SalesOrderID
INNER JOIN production
ON salesorderdetail.ProductID=production.ProductID
GROUP BY production.ProductID, production.Name, salesorderheader.OrderDate
ORDER BY production.Name, salesorderheader.OrderDate
```

production(+) 1

SELECT production.ProductID, production.Name, salesorderheader.OrderDate, sum(salesorderdetail.OrderQty)

	ProductID	Name	OrderDate	sum(salesorderdetail.OrderQty)
1	879	All-Purpose Bike Stand	2013-05-30	1
2	879	All-Purpose Bike Stand	2013-06-18	1
3	879	All-Purpose Bike Stand	2013-06-20	1
4	879	All-Purpose Bike Stand	2013-06-22	1
5	879	All-Purpose Bike Stand	2013-06-30	2
6	879	All-Purpose Bike Stand	2013-07-02	2
7	879	All-Purpose Bike Stand	2013-07-03	1
8	879	All-Purpose Bike Stand	2013-07-04	1

Refresh Save Cancel Export data 200 200+

200 row(s) fetched - 1.250s (17ms fetch), on 2023-07-24 at 10:43:36

BRT en Writable Smart Insert 8 : 53 : 430 Sel: 0 | 0

10:49 24/07/2023

# Consulta no console do banco no GCP

The screenshot shows the Google Cloud Platform console interface. At the top, there are tabs for 'Caixa d', 'rox', 'Projeto', 'Banco\_', and 'Analise'. The address bar shows the URL: `console.cloud.google.com/sql/instances/rox/overview?project=leega-estudos&cloudshell=true`. Below the address bar is a search bar with the text 'Search (/) for resources, docs, products, and more' and a 'Search' button. The main content area is titled 'CLOUD SHELL' and 'Terminal (leega-estudos)'. It displays the following text:


```
mysql> use dbrox;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT SalesOrderID, OrderDate, TotalDue FROM salesorderheader where OrderDate between '2011-09-01' and '2011-09-30' and TotalDue > 1000 ORDER BY TotalDue DESC;
```



SalesOrderID	OrderDate	TotalDue
44324	2011-09-01	3953,9884
44478	2011-09-29	3953,9884
44326	2011-09-01	3953,9884
44327	2011-09-02	3953,9884
44328	2011-09-02	3953,9884
44329	2011-09-02	3953,9884
44330	2011-09-02	3953,9884
44331	2011-09-03	3953,9884
44332	2011-09-03	3953,9884
44477	2011-09-29	3953,9884
44334	2011-09-04	3953,9884
44476	2011-09-29	3953,9884
44473	2011-09-29	3953,9884
44338	2011-09-04	3953,9884
44339	2011-09-04	3953,9884
44340	2011-09-04	3953,9884
44472	2011-09-29	3953,9884
44343	2011-09-05	3953,9884
44344	2011-09-06	3953,9884
44345	2011-09-06	3953,9884
44347	2011-09-06	3953,9884
44348	2011-09-07	3953,9884
44349	2011-09-07	3953,9884
44350	2011-09-07	3953,9884
44351	2011-09-07	3953,9884
44352	2011-09-07	3953,9884
44470	2011-09-28	3953,9884
44469	2011-09-28	3953,9884



The bottom of the image shows a Windows taskbar with the Start button, a search bar labeled 'Pesquisar', and several application icons. The system tray on the right shows the time '09:45' and the date '24/07/2023'.



# Importando para o DataStudio


 Estudo de Caso - ROX


Arquivo Editar Exibir Inserir Página Organizar Recurso Ajuda


 


 


 Página 11 de 14 


 Adicionar dados


 Adicionar um gráfico

 Adicionar um controle





 Adicionar dados ao relatório



## Cloud SQL para MySQL

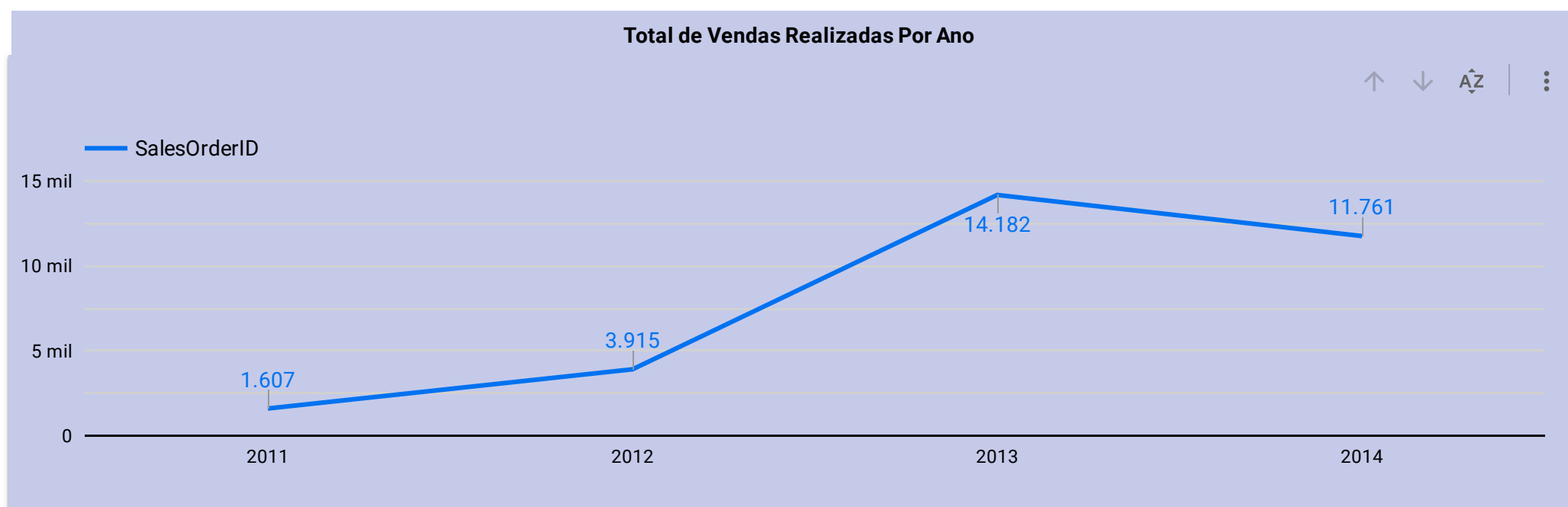
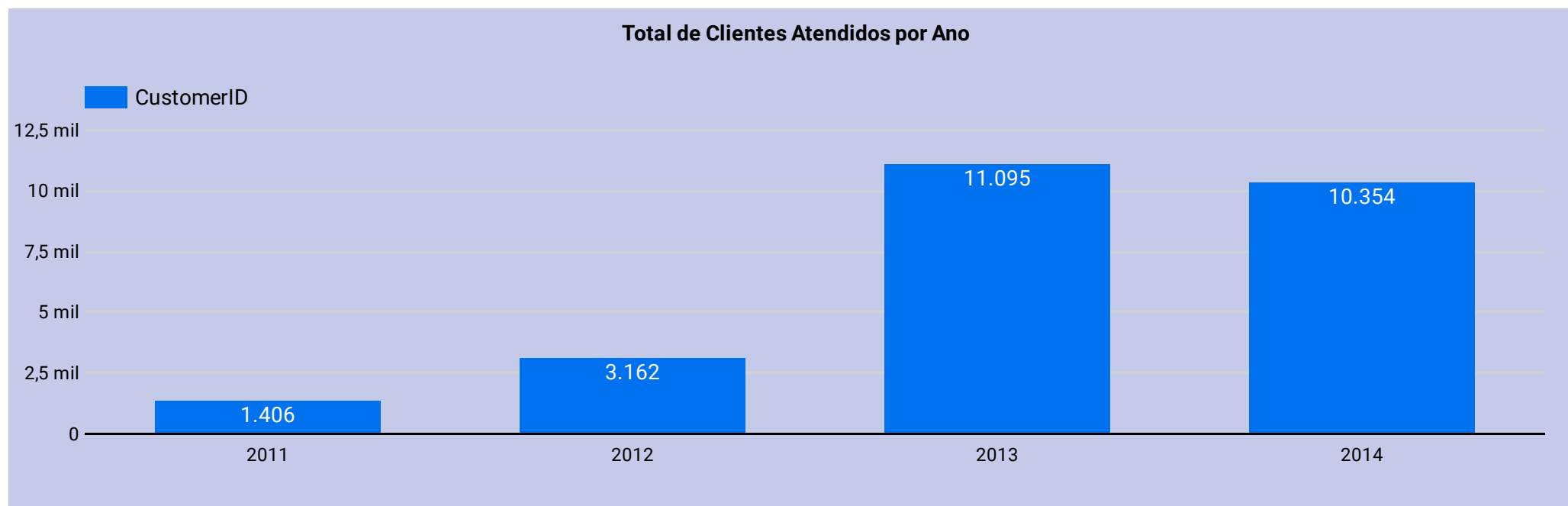
Por Google

Com o conector do Google Cloud SQL, você pode acessar dados de bancos de dados do Google Cloud SQL no Looker Studio.

[SAIBA MAIS](#) [INFORMAR UM PROBLEMA](#)

BÁSICO	Autenticação do banco de dados	TABELAS	Tabela
URL JDBC	<div>Nome da conexão da instância</div> <div><div></div></div> <div>Banco de dados</div> <div>dbrox</div> <div>Nome de usuário</div> <div>root</div> <div>Senha</div> <div>....</div> <div>AUTENTICAR</div>	CONSULTA PERSONALIZADA	<div>customer</div> <div>person</div> <div>production</div> <div>salesorderdetail</div> <div>salesorderheader</div> <div>specialofferproduct</div>

# BI Produção de Bicicletas



## *Ponto de Atenção do Estudo de Caso*

- 1 - Na tabela Person, coluna Demographics e AdditionalContactInfo com o caractere " , " onde foi necessário alterar para " ; " para a correta formatação pelo PySpark
- 2 - Na tabela SpecialOfferProduct não existe uma Primary Key. Em teoria, uma tabela não deve ficar sem uma Primary Key. Uma sugestão seria criar uma Primary Key da união das colunas ProductID e SpecialOfferID já que elas, juntas, não se repetem. Porém esta solução também precisa ser amplamente estudada para analisar o impacto sobre as demais tabelas.
- 3 - Se este é um processo diário sugiro que seja realizado através de um gerenciador de fluxo de trabalho como o Airflow automatizando o processo.
- 4 - Os arquivos do colab Analise\_ROX e Banco\_Rox podem ser encontrados na íntegra no github.

**Obrigada!**