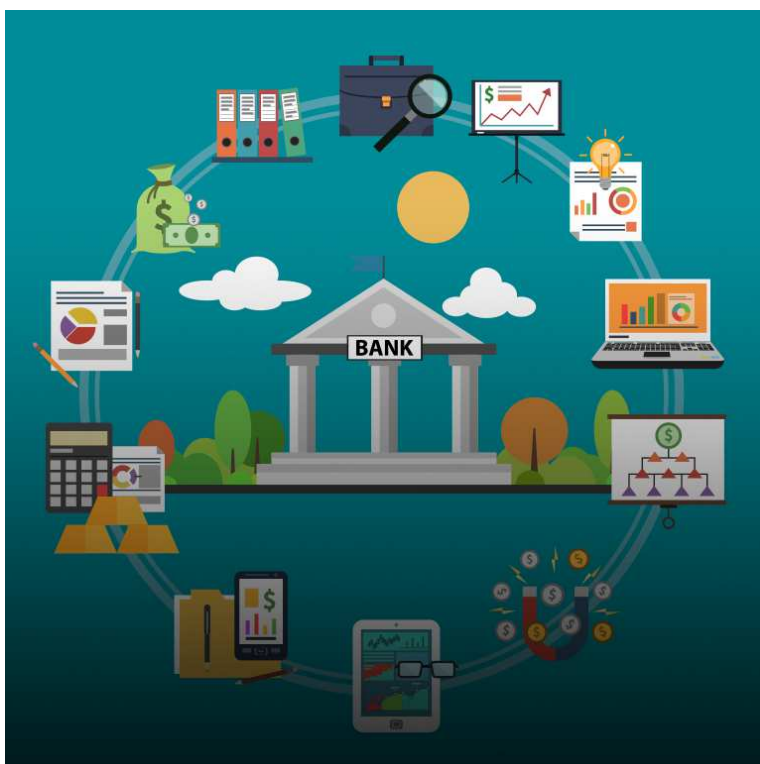


# MVP - Disciplina: Sprint: Engenharia de Dados

Robson da Silva Barbosa

Fonte : <https://www.kaggle.com/datasets/santoshd3/bank-customers>

## Bank Customers Churn



O case aqui sobre um banco de dados que traz informações sobre clientes de um banco alemão, algumas variáveis como gênero, saldo de conta corrente, salário estimado entre outras , país de origem e entre outras. Todas elas para contar a história do cliente e a partir dessas variáveis tentar explicar ou conhecer o comportamento do cliente que acaba retirando sua conta do banco, dando “churn”.

Com isso, vou trazer o trabalho de importação desse conjunto de dados , todo trabalho de ETL utilizando o AWS , onde passaremos pelo S3, Glue e por fim o ambiente do redshift que é por onde faremos nossas consultas para perguntas e curiosidades sobre esse conjunto de dados.

## Qualidade dos dados :

Feito análise, o conjunto de dados está em perfeito estado. Não havendo dados nulos , duplicados ou faltantes. Procurado Dataset com mais erros para tratamento mas não encontrei. Diante disso, segui com o Dataset atual , sem erros e com boa qualidade da apresentação dos dados.

## MetaDados :

Sobre as variáveis da tabela:

**CreditScore** – é uma medida de avaliação do cliente, juntando diversas características , ponderando seus pesos e atribuindo ao cliente uma nota . Quanto maior a nota melhor é o cliente . Está em formato número inteiro.

**Geography** – Fator regional, indica o país do cliente, está em formato string.

**Gender** – Gênero do cliente, identificando como Male ou Female , em formato de texto.

**Age** – Idade, está em formato de número inteiro

**Tenure** – Ocupação, em formato de string, traz a profissão do cliente.

**Balance** – Saldo de conta corrente, indica qual o montante de numerário constante na conta correndo do cliente, está em formato de float.

**NumOfProducts** – Número de produtos, tamanho da cesta de produtos que a pessoa possui com o banco. Está em formato de número inteiro.

**HasCrCard** – Tradução : “Tem cartão de crédito”. Indica se cliente tem cartão de crédito com o banco, sim ou não. Está como texto.

**IsActiveMember** – É membro/usuário ativo . Indica se cliente mantém relacionamento com o banco. Está em formato de texto.

**EstimatedSalary** – Salário estimado do cliente , formato de float.

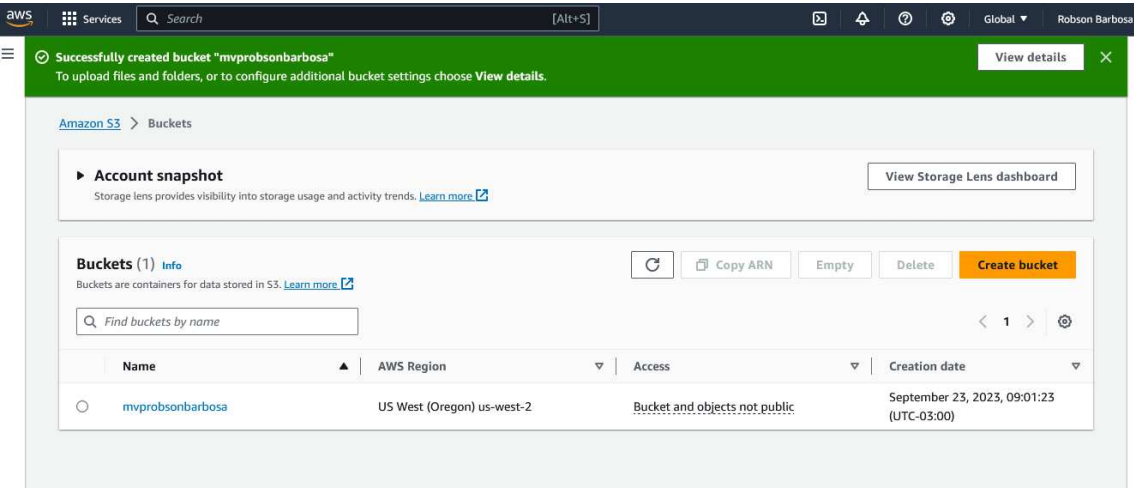
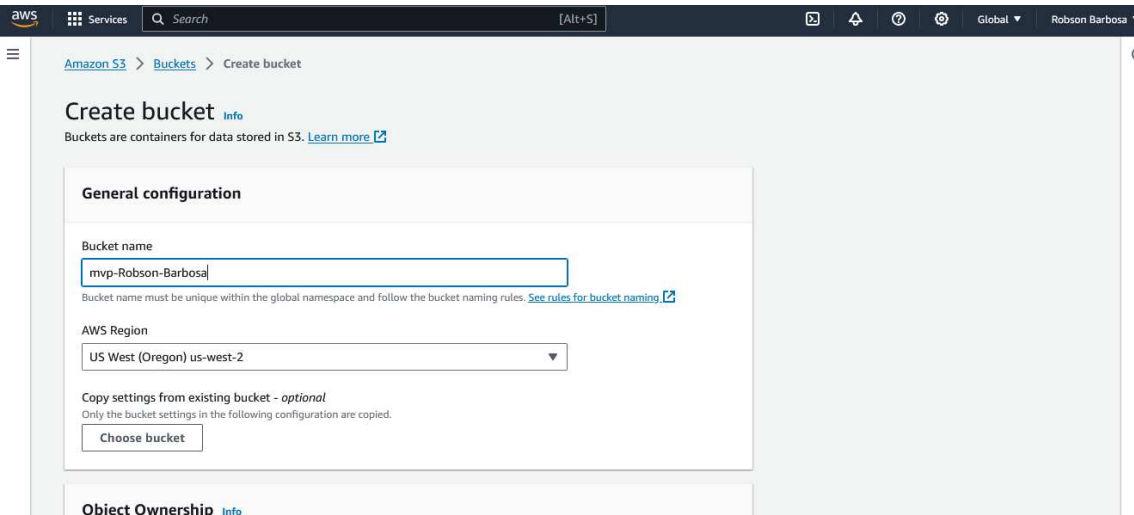
**Exited** – Evadido, simboliza se o cliente encerrou as atividades com o banco ou não, se constatado que deixou o banco o valor é 1 , se continua então o valor é 0. Formato é texto.

---

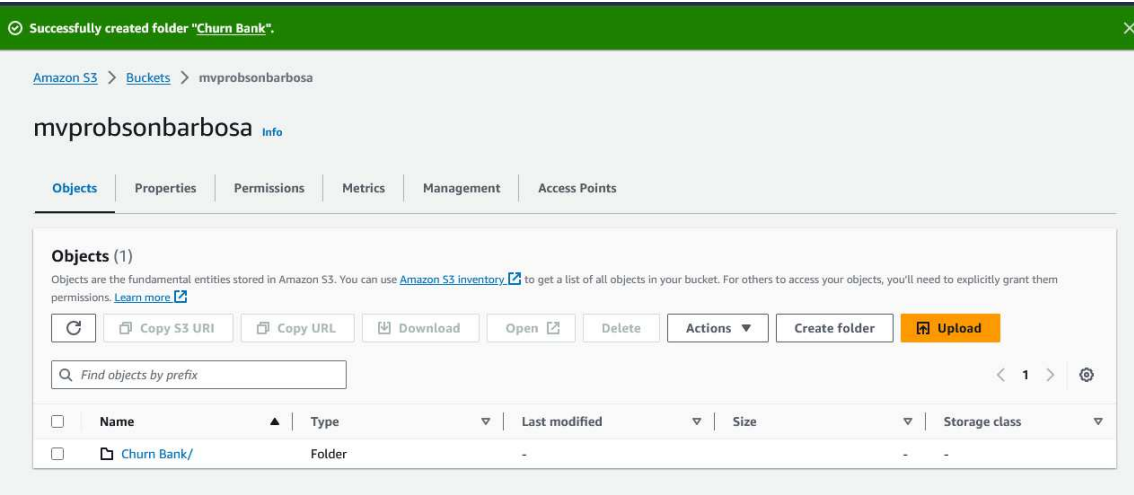
# Criação do bucket

## Definição:

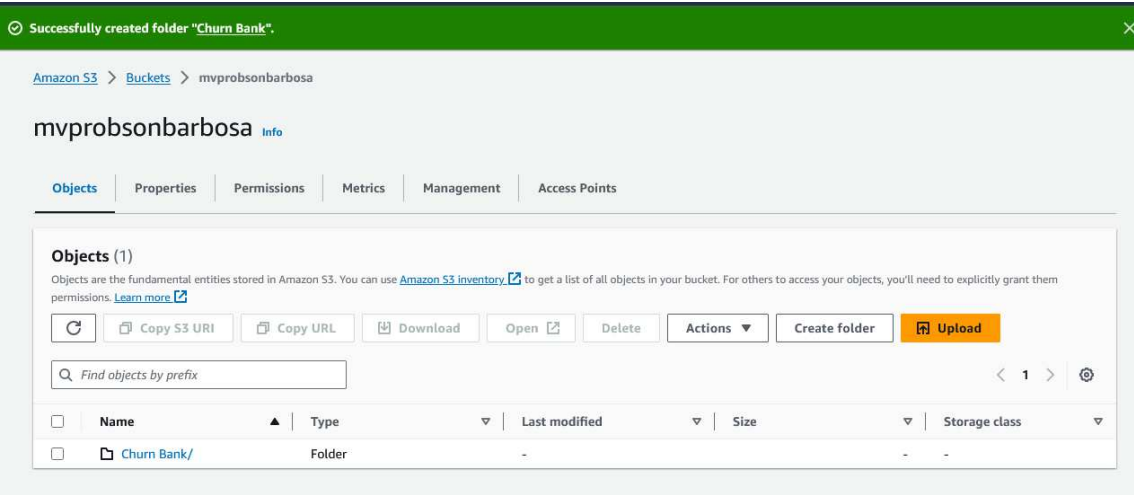
A criação de um bucket é um passo fundamental ao usar o Amazon Web Services (AWS) Cloud, especialmente quando se trata de serviços de armazenamento, como o Amazon S3 (Simple Storage Service). Um "bucket" é um contêiner de armazenamento que permite armazenar e organizar dados na nuvem da AWS.



# Criação da pasta



# Foi feito o upload do arquivo

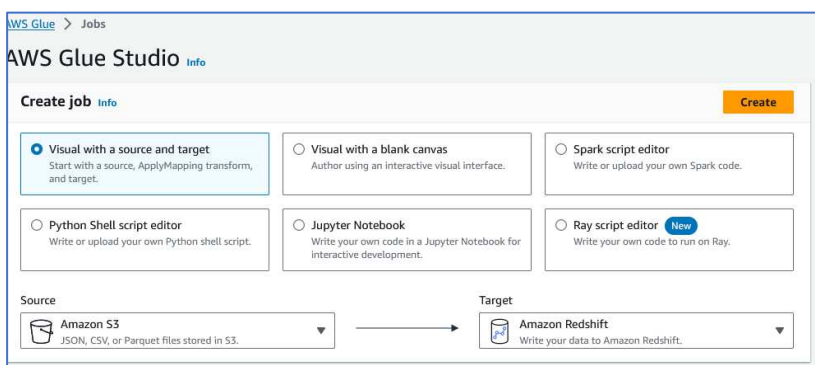
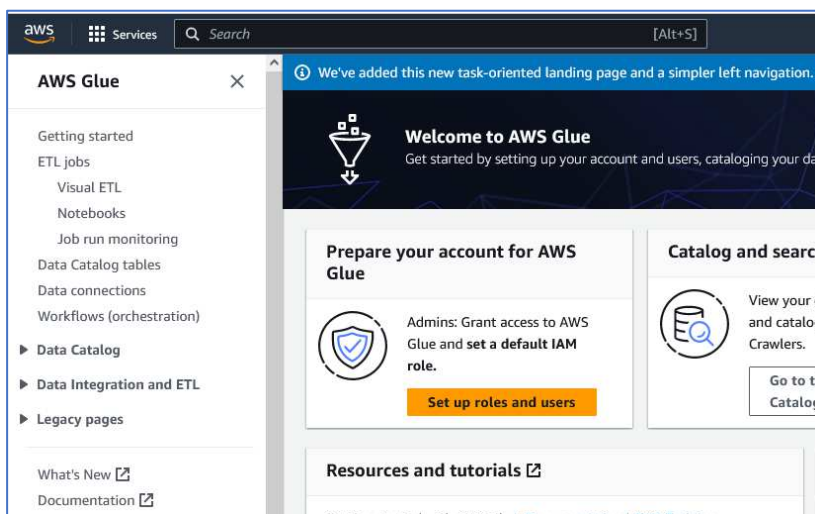


# Glue

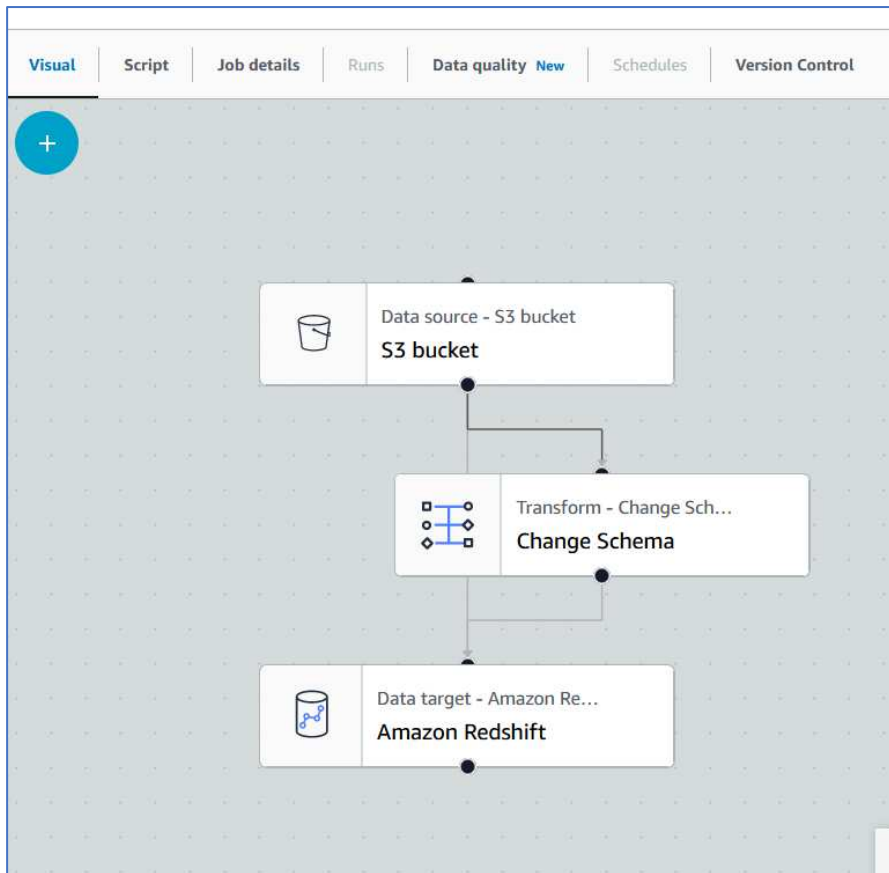
## Descrição:

O AWS Glue é um serviço totalmente gerenciado pela Amazon Web Services (AWS) que oferece recursos de ETL (Extração, Transformação e Carga) e preparação de dados na nuvem. É uma ferramenta poderosa para automatizar tarefas de integração e transformação de dados, tornando mais fácil e eficiente trabalhar com conjuntos de dados em uma variedade de fontes e formatos.

No geral, o AWS Glue é uma ferramenta poderosa para transformação e preparação de dados na nuvem da AWS. Ele ajuda as organizações a simplificar tarefas de ETL, melhorar a qualidade dos dados e acelerar o desenvolvimento de soluções de análise e processamento de dados, permitindo que você foque mais em insights e menos em gerenciamento de dados.



Aqui eu tive uma dificuldade porque não veio default o Change Schema e achei que havia errado no processo. No começo veio apenas o S3 e o Redshift, então tive que colocar manualmente o Change Schema, com isso, de início não criou um fluxo direto e único de S3 para Change e por fim o Redshift. Como pode ver na figura ficou um fluxo secundário ainda ligando o S3 diretamente no Redshift, o que não era o desejado.



Com o decorrer do tempo descobri na caixinha o contêiner do Redshift conseguimos definir o fluxo, na parte vermelha havia o o Redshift e o Change Schema, simbolizando que do S3 o fluxo seguiria para o redshift e para o change Schema, o que não queríamos. Queremos que o fluxo venha do S3 para daí então passar pelo change schema, o glue, para fazemos os tratamentos dos dados, para por fim chegar no Redshift bonitinho.

Então ao manter apenas o change schema conforme mostrado na figura , definimos o fluxo único e direto simbolizando nas marcações em azul.

Bom, aqui começamos a importação dos dados no fluxo, selecionando o S3 e trazendo os dados que são identificados automaticamente pelo AWS ao clicar em infer schema.

Visual Script Job details Runs Data quality New Schedules Version Control

Untitled job

Job has not been saved Try new UI Actions Save Run

Visual Script Job details Runs Data quality New Schedules Version Control

+

Data source - S3 bucket S3 bucket

Transform - Change Schema Change Schema

Data target - Amazon Redshift Amazon Redshift

Data source properties - S3

S3 location

Choose a file or folder in an S3 bucket.

☐ Data Catalog table

S3 URL

View Browse S3

☒ Recursive

Read files in all subdirectories.

Data format

CSV

Delimiter

Comma (,)

Escape character - optional

Enter a character to use for escaping.

Quote character

Double quote (")

☒ First line of source file contains column headers

☐ Records in source files can span multiple lines

Additional options

Neste ponto estamos no Glue, utilizando o Change Schema que é para fazer aquela história de tratamento de dados. Para motivos de processamento eu retirei algumas colunas que não trariam informação relevante como número da linha, id do cliente e nome, a não ser se tivesse um nome famoso e rico, mas mesmo assim tirei, não somos fofoqueiros. Também troquei os tipos dos dados, transformando colunas necessárias em números inteiros, decimais e mantendo outras como texto.

Visual Script Job details Runs Data quality New Schedules Version Control

Visual Script Job details Runs Data quality New Schedules Version Control

+

Data source - S3 bucket S3 bucket

Transform - Change Schema Change Schema

Data target - Amazon Redshift Amazon Redshift

Transform

Output schema

Data preview

Choose one or more parent nodes

S3 bucket X

S3 - DataSource

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
RowNumber			<input checked="" type="checkbox"/>
CustomerId			<input checked="" type="checkbox"/>
Surname			<input checked="" type="checkbox"/>
CreditScore	CreditScore	string	<input type="checkbox"/>
Geography	Geography	string	<input type="checkbox"/>
Gender	Gender	string	<input type="checkbox"/>
Age	Age	string	<input type="checkbox"/>
Tenure	Tenure	string	<input type="checkbox"/>
Balance	Balance	string	<input type="checkbox"/>
NumOfProducts	NumOfProducts	string	<input type="checkbox"/>
HasCrCard	HasCrCard	string	<input type="checkbox"/>
IsActiveMember	IsActiveMember	string	<input type="checkbox"/>
EstimatedSalary	EstimatedSalary	string	<input type="checkbox"/>
Exited	Exited	string	<input type="checkbox"/>

Aqui mostrando o que falei acima.

Change Schema (Apply mapping)			
Source key	Target key	Data type	Drop
RowNumber			<input checked="" type="checkbox"/>
CustomerId			<input checked="" type="checkbox"/>
Surname			<input checked="" type="checkbox"/>
CreditScore	CreditScore	string ▼	<input type="checkbox"/>
Geography	Geography	string ▼	<input type="checkbox"/>
Gender	Gender	string ▼	<input type="checkbox"/>
Age	Age	int ▼	<input type="checkbox"/>
Tenure	Tenure	string ▼	<input type="checkbox"/>
Balance	Balance	float ▼	<input type="checkbox"/>
NumOfProducts	NumOfProducts	int ▼	<input type="checkbox"/>
HasCrCard	HasCrCard	string ▼	<input type="checkbox"/>
IsActiveMember	IsActiveMember	string ▼	<input type="checkbox"/>
EstimatedSalary	EstimatedSalary	float ▼	<input type="checkbox"/>
Exited	Exited	string ▼	<input type="checkbox"/>



# Criação do Redshift

## Descrição:

Amazon Redshift é um serviço de armazenamento de dados e análise de data warehousing altamente escalável e totalmente gerenciado pela Amazon Web Services (AWS). Ele é projetado para processamento de consultas SQL rápido e eficiente em grandes conjuntos de dados e é amplamente utilizado por empresas para armazenar, processar e analisar grandes volumes de informações.

Em resumo, o Amazon Redshift é uma solução de data warehousing escalável e de alto desempenho que permite às empresas armazenar, processar e analisar grandes volumes de dados com facilidade. Ele é amplamente adotado para análise de negócios e geração de insights a partir de dados, tornando-se uma escolha popular para empresas de todos os tamanhos.

[Amazon Redshift Serverless](#) > Get started with Amazon Redshift Serverless

### Get started with Amazon Redshift Serverless [info](#)

To start using Amazon Redshift Serverless, set up your serverless data warehouse and create a database. You will receive \$300.00 credit towards your Redshift Serverless usage in this account.


#### Configuration

☒ **Use default settings**  
Default settings have been defined to help you get started. You can change them at any time later.

☐ **Customize settings**  
Customize your settings for your specific needs.

#### Namespace [info](#)

Namespace is a collection of database objects and users. Data properties include database name and password, permissions, and encryption and security.

 Your data is encrypted by default with an AWS owned key. To choose a different key, choose **Customize settings**.


Target namespace  
default-namespace

Database name and password

Database name  
dev

Admin user credentials  
Created based on IAM credentials

Permissions

 Associate an IAM role so that your serverless endpoint can **LOAD** and **UNLOAD** data. You can create an IAM role as the default for this configuration that has the [AmazonRedshiftAllCommandsFullAccess](#) policy attached. This policy includes permissions to run SQL commands to **COPY**, **UNLOAD**, and query data with Amazon Redshift Serverless. This policy also grants permissions to run **SELECT** statements for related services, such as Amazon S3, Amazon CloudWatch logs, Amazon SageMaker, and AWS Glue. You won't be able to run these SQL commands without an IAM role attached to your namespace.

#### Associated IAM roles (0)

Create, associate, or remove an IAM role. You can associate up to 50 IAM roles. You can also choose an IAM role and set it as the default.

Set default ▾

Manage IAM roles ▾

< 1 >

Role
------

Fiz algumas configurações personalizadas, atribui uma senha para o admin, coloquei uma capacidade menor para evitar custos adicionais . E assim, criado com sucesso o Redshift

☐

IAM roles [?](#)

▼

Status

▼

Role type

▼

<input type="checkbox"/>	<a href="#">AmazonRedshift-CommandsAccessRole-20230929T203754</a>	Not applied	--
<input type="checkbox"/>	<a href="#">AmazonRedshift-CommandsAccessRole-20230929T204052</a>	Not applied	Default

▼ Security and encryption

⚠

Your data is encrypted by default with an AWS owned key. To choose a different key, customize your encryption settings.

☐ Customize encryption settings (advanced)

Audit logging [Info](#)

Collects logging information for the database.

Export these logs:

☐ User log

☐ Connection log

☐ User activity log

Workgroup [Info](#)

Workgroup is a collection of compute resources from which an endpoint is created. Compute properties include network and security settings.

▼ Capacity

Set the base capacity used to process your data warehouse workloads. The capacity is measured in Redshift processing units (RPU). To improve query performance, increase the RPU value.

Base capacity

Base RPU capacity is set to 128 RPUs by default. To change the base RPU capacity, choose another value from the list.

8

▼

Range must be 8-512 in increments of 8.

▼ Network and security

Virtual private cloud (VPC)

This VPC defines the virtual networking environment for this database.

vpc-056d1b08d7a139431

▼

VPC security groups

This VPC security group defines which subnets and IP ranges can be used in the VPC.

Choose one or more security groups

▼

sg-0b9e1772353cdf1c

×

Subnet

The subnet in the chosen VPC that is associated with the specified database.

Choose three or more subnet IDs

▼

subnet-0a37fe3d33c824183

×

subnet-05446154904399ed0

×

subnet-0272615f082f039c6

×

subnet-0438013d7de88fe68

×

Enhanced VPC routing

Turning on this option routes network traffic between your serverless database and data repositories through a VPC instead of the internet.

☐ Turn on enhanced VPC routing

Cancel

Save configuration

# Criando conexão Glue-Redshift

Name

Enter a unique name for your connection.

conexao-glue-redshift

Connection type

Amazon Redshift

☐ Require SSL connection

The connection will fail if it's unable to connect over SSL.

Description - optional

Descriptions can be up to 2048 characters long.

Connection access

Database instances

Provisioned Amazon Relational Database Service instances.

default-workgroup2

Database name

dev

Credential type

☒ Username and password

☐ AWS Secrets Manager

Username

admin

Connectors

Info

Marketplace connectors

Subscribe to connectors from AWS partners to expand your data sources.

Go to AWS Marketplace

Custom connectors

Provide your own connector to expand your data sources.

Creating custom connectors

Create custom connector

Connectors (0)

Info

Actions

You can manage your connectors or use them to create connections.

Filter connections by property

< 1 >

Name

Type

Last modified

Connections (1)

Info

Actions

Create connection

Create job

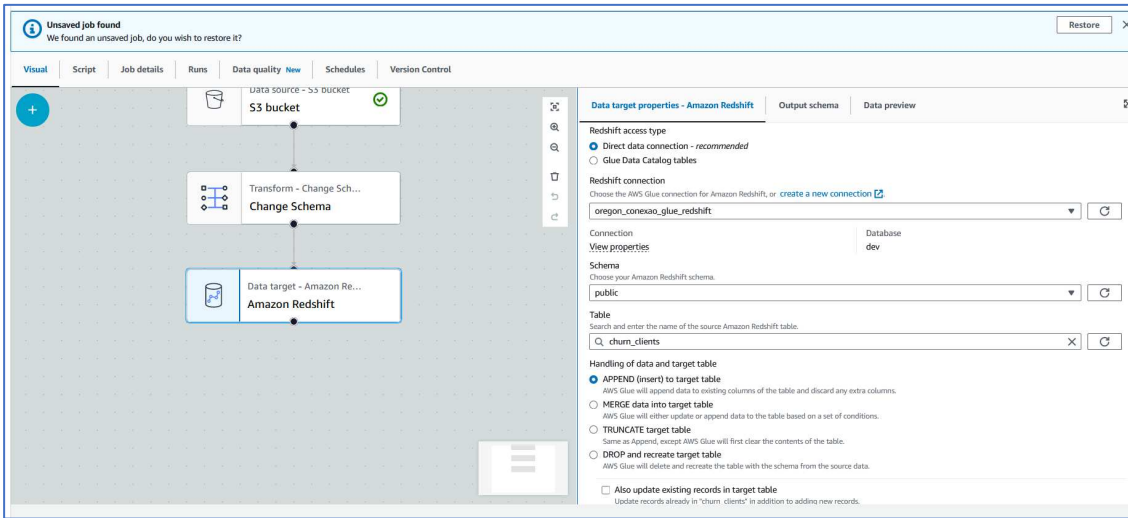
You can manage your connections or use a connection in a job.

Filter connections by property

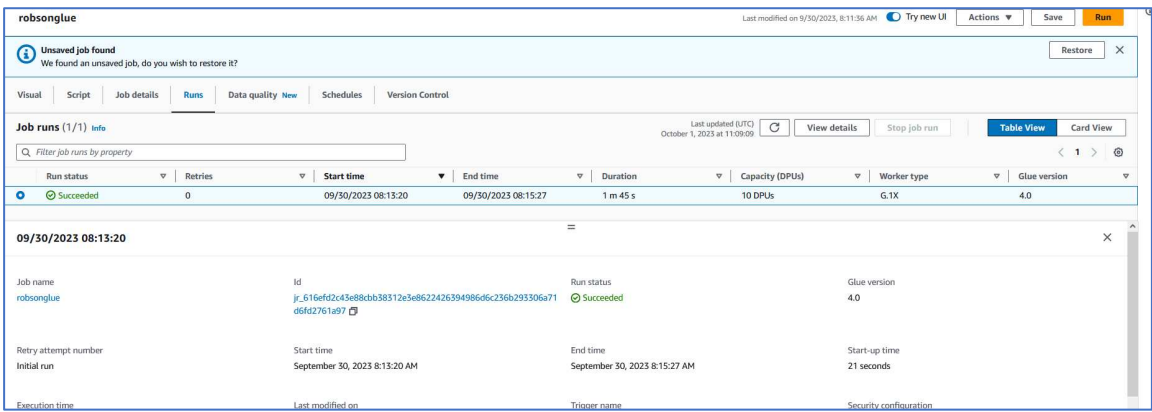
< 1 >

Name	Type	Last modified
oregon_conexao_glue_redshift	JDBC	Sep 30, 2023

Feito a conexão inclui ela no fluxo , no container do Redshift. E assim ficou o fluxo , direto, único e bem bonito.

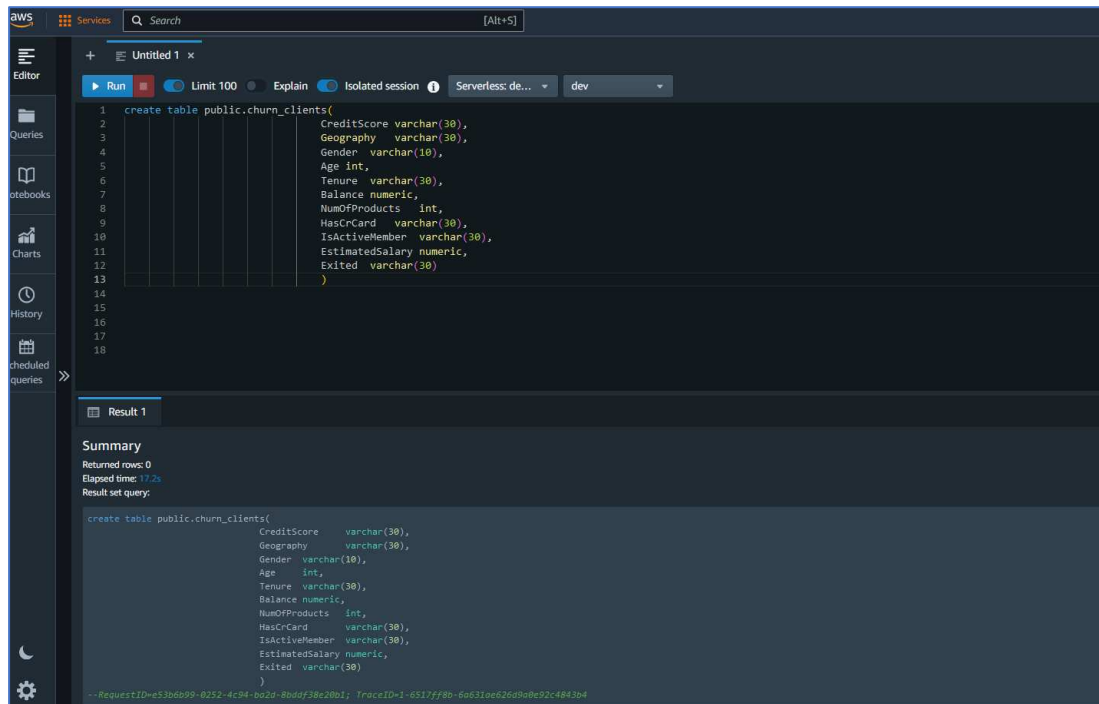


Finalizando alguns detalhes , fiz o salvamento do fluxo e rodei o mesmo.



- **Nos finalmente.**

E por fim nesta etapa de integração criei a tabela no banco de dados da query builder do Redshift.



## Mão na massa

E agora iniciamos nossa pesquisa sobre o conjunto de dados , fazendo perguntas e procurando entender um pouco mais ele.

- **O Conjunto de dados tem um 10000 clientes na amostra.**

```
32
33 SELECT count(*) as total
34 FROM churn_clients
35
36
37
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (1)
<input type="checkbox"/>	total		
<input type="checkbox"/>	10000		

- **Quantidade de homens e mulheres**

```
32
33 SELECT Gender, COUNT(*) AS quantidade
34 FROM churn_clients
35 GROUP BY Gender;
36
37
38
39
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)
<input type="checkbox"/>	gender	quantidade	
<input type="checkbox"/>	Female	4543	
<input type="checkbox"/>	Male	5457	

- **Média de idade de todo conjunto de dados**

```
32
33 SELECT AVG(age) AS media_idade
34 FROM churn_clients
35
36
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (1)
<input type="checkbox"/>	media_idade		
<input type="checkbox"/>	38		

- Nesta consulta trago a média de idade geral dos homens e mulheres :

```
26
27 SELECT Gender as sexo , AVG(age) AS media_idade
28 FROM churn_clients
29 GROUP BY Gender;
30
31
32
```

	Result 1 (100)	Result 2 (1)	Result 3 (2)
<input type="checkbox"/>	sexo		media_idade
<input type="checkbox"/>	Female		39
<input type="checkbox"/>	Male		38

- Visão global do dataset

Aqui eu respondo a quantidade de clientes do conjunto de dados que saíram do banco, o resultado foi 2037, em uma base com 10 mil clientes, isto é equivalente a 20,37%.

```
9  --SELECT COUNT(*)
10 --FROM sua_tabela
11 --WHERE coluna_B = 1
12 --AND coluna_A IN (SELECT coluna_A FROM sua_tabela ORDER BY coluna_A LIMIT 100);
13
14
15 select count(*)
16 from churn_clients
17 where exited = 1
18 --select * from churn_clients order by creditscore desc limit 100
19
20
21
```

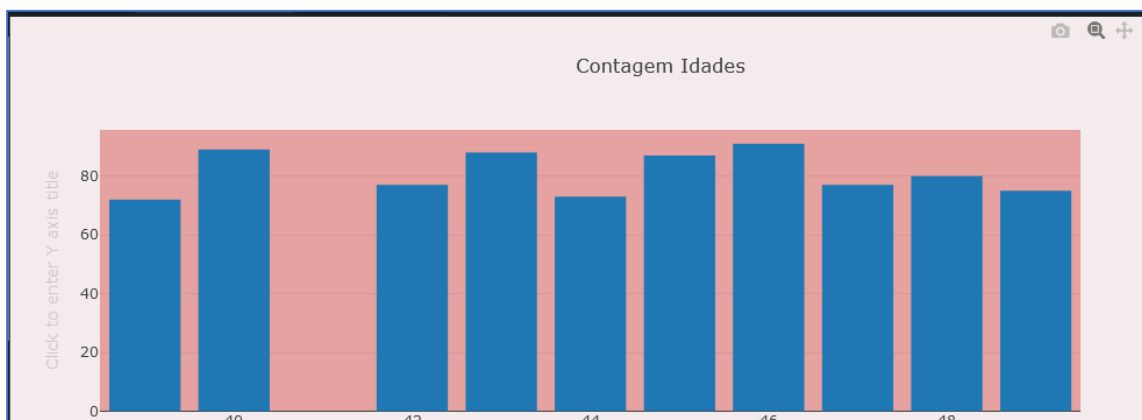
	Result 1 (1)
<input type="checkbox"/>	count
<input type="checkbox"/>	2037

- **Evasão por idade**

Sobre os clientes que evadiram segue um script que traz as 10 idades que mais tiveram clientes encerrando sua conta. Chegamos a conclusão que 46 é a idade que mais teve clientes encerrando seus negócios com o banco, totalizando 91 pessoas.

```
37
38 SELECT age as idade , COUNT(*) as contagem FROM churn_clients
39 where exited = 1
40 GROUP BY age
41 order by contagem desc
42 limit 10
43
44
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)	Result 5 (10)
<input type="checkbox"/>	idade	contagem		
<input type="checkbox"/>	46	91		
<input type="checkbox"/>	40	89		
<input type="checkbox"/>	43	88		
<input type="checkbox"/>	45	87		
<input type="checkbox"/>	48	80		
<input type="checkbox"/>	42	77		
<input type="checkbox"/>	47	77		
<input type="checkbox"/>	49	75		
<input type="checkbox"/>	44	73		





- Da mesma forma, fiz a consulta das idades que mais permaneceram :

```
37
38 SELECT  age as idade , COUNT(*) as contagem_permanecem FROM churn_clients
39 where exited = 0
40 GROUP BY age
41 order by contagem_permanecem desc
42 limit 10;
43
44
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)	Result 5 (10)	Result 6
<input type="checkbox"/>	idade	contagem_permanecem			
<input type="checkbox"/>	35	417			
<input type="checkbox"/>	37	416			
<input type="checkbox"/>	34	414			
<input type="checkbox"/>	38	414			
<input type="checkbox"/>	36	403			
<input type="checkbox"/>	33	398			
<input type="checkbox"/>	32	386			
<input type="checkbox"/>	31	371			
<input type="checkbox"/>	39	351			
<input type="checkbox"/>	40	343			

- Sobre os países agora de um modo geral

```
48
49 SELECT  geography as país , COUNT(*) as contagem FROM churn_clients
50 GROUP BY geography
51 order by contagem desc
52
53
54
55
56
57
```

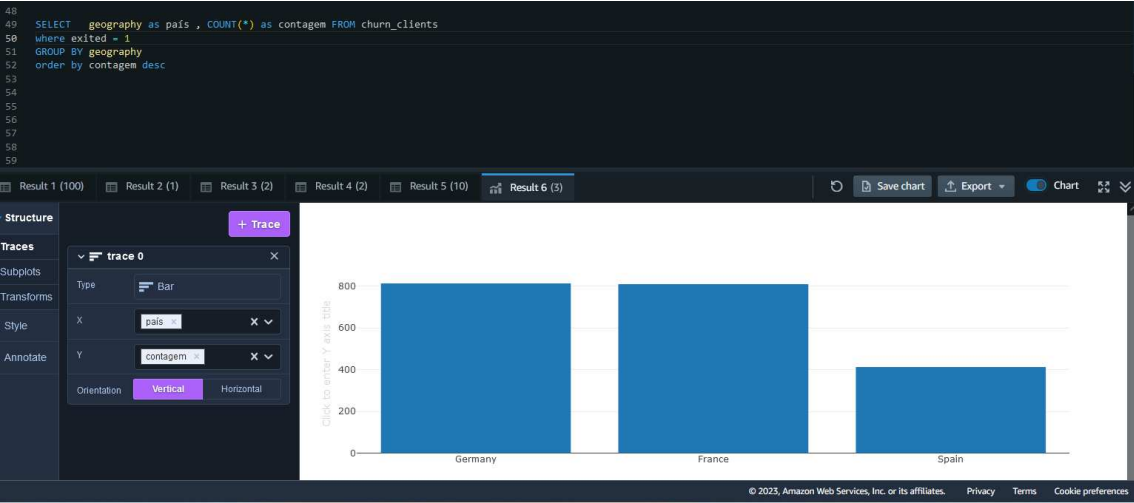
Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)	Result 5 (10)	Result 6 (3)
<input type="checkbox"/>	país	contagem			
<input type="checkbox"/>	France	5014			
<input type="checkbox"/>	Germany	2509			
<input type="checkbox"/>	Spain	2477			

- Quantidades de Churn por país:

```
47
48
49 SELECT  geography as país , COUNT(*) as contagem FROM churn_clients
50 where exited = 1
51 GROUP BY geography
52 order by contagem desc
53
54
55
56
57
```

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)	Result 5 (10)	Result 6 (3)
<input type="checkbox"/>	país	contagem			
<input type="checkbox"/>	Germany	814			
<input type="checkbox"/>	France	810			
<input type="checkbox"/>	Spain	413			

Recurso interessante é que ele já disponibiliza gráficos das consultas como abaixo :



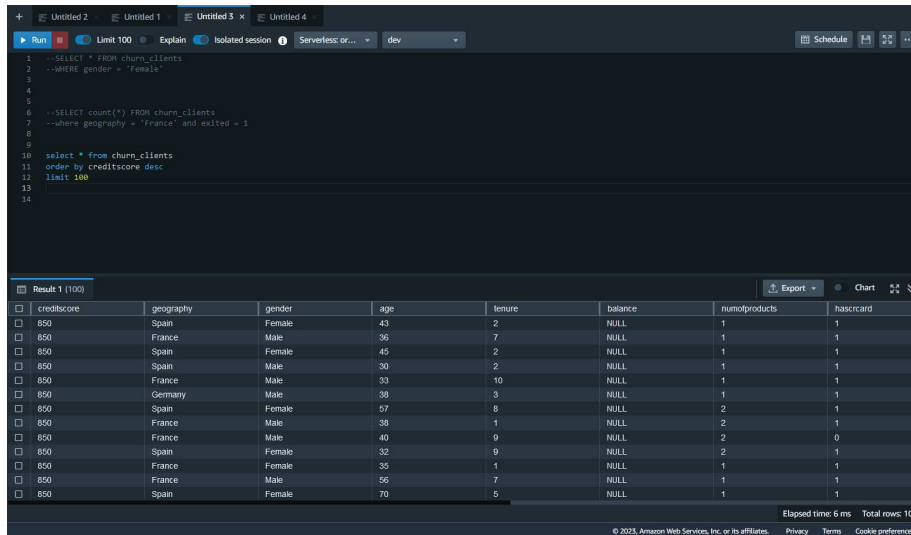
E sem Churn:

```
48
49 SELECT geography as país , COUNT(*) as contagem FROM churn_clients
50 where exited = 0
51 GROUP BY geography
52 order by contagem desc
53
54
55
56
57
```

	país	contagem
<input type="checkbox"/>	France	4204
<input type="checkbox"/>	Spain	2064
<input type="checkbox"/>	Germany	1695

- **Selecionando os 100 primeiros clientes com o creditscore mais elevado.**

O creditscore é a avaliação de um cliente, quanto mais alto o creditscore melhor é o cliente, traz um cálculo que engloba muitas variáveis para definir se aquela pessoa é ou não bom cliente.

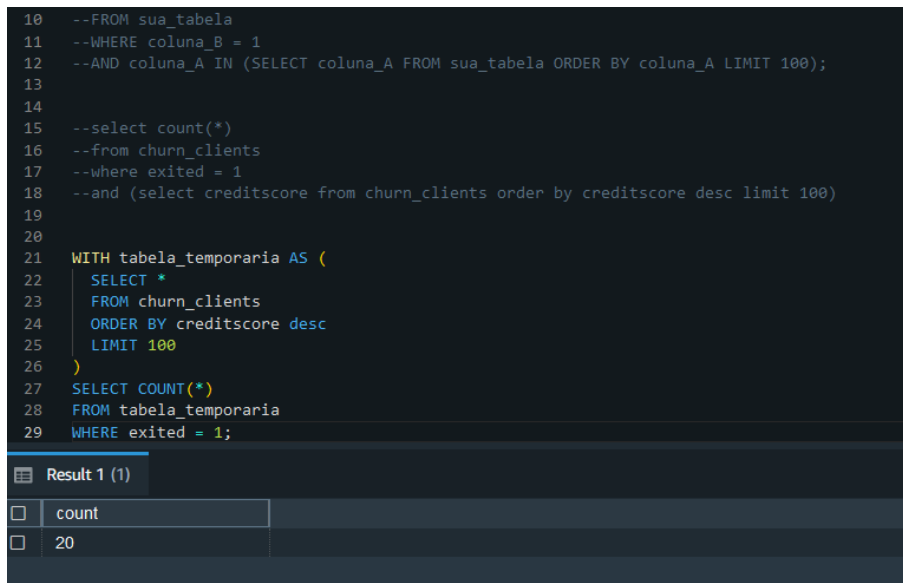


```
1 --SELECT * FROM churn_clients
2 --WHERE gender = 'female'
3
4
5
6 --SELECT count(*) FROM churn_clients
7 --where geography = 'France' and exited = 1
8
9
10 select * from churn_clients
11 order by creditscore desc
12 limit 100
13
14
```

creditscore	geography	gender	age	tenure	balance	numofproducts	hascard
850	Spain	Female	43	2	NULL	1	1
850	France	Male	36	7	NULL	1	1
850	Spain	Female	45	2	NULL	1	1
850	Spain	Male	30	2	NULL	1	1
850	France	Male	33	10	NULL	1	1
850	Germany	Male	38	3	NULL	1	1
850	Spain	Female	57	8	NULL	2	1
850	France	Male	38	1	NULL	2	1
850	France	Male	40	9	NULL	2	0
850	Spain	Female	32	9	NULL	2	1
850	France	Female	35	1	NULL	1	1
850	France	Male	56	7	NULL	1	1
850	Spain	Female	70	5	NULL	1	1

Elapsed time: 6 ms Total rows: 100

Aqui criei uma tabela temporária e dentro dela selecionei os que tinham dado churn



```
10 --FROM sua_tabela
11 --WHERE coluna_B = 1
12 --AND coluna_A IN (SELECT coluna_A FROM sua_tabela ORDER BY coluna_A LIMIT 100);
13
14
15 --select count(*)
16 --from churn_clients
17 --where exited = 1
18 --and (select creditscore from churn_clients order by creditscore desc limit 100)
19
20
21 WITH tabela_temporaria AS (
22     SELECT *
23     FROM churn_clients
24     ORDER BY creditscore desc
25     LIMIT 100
26 )
27 SELECT COUNT(*)
28 FROM tabela_temporaria
29 WHERE exited = 1;
```

count
20

Dos 100 maiores creditscore, apenas 20 deram churn , um percentual de 20% . Mantém resultado parecido com a base geral.

- Os 100 maiores salários do conjunto.

```
15 with tabela as (  
16     SELECT *  
17     FROM churn_clients  
18     ORDER BY estimatedsalary_float desc  
19     LIMIT 100  
20 )  
21 select count(*) as "Evadidos" from tabela  
22 where exited = 1;  
23  
24  
25  
26
```

Result 1 (100)   Result 2 (1)   Result 3 (2)   Result 4 (2)   Result 5 (10)   Result 6 (3)

<input type="checkbox"/>	evadidos
<input type="checkbox"/>	16

Então, em resumo, da base com os 100 maiores salários, em que 16 deram churn. Ou seja, apenas 16% saíram do banco.

- Aqui por curiosidade seleciono os 100 maiores salários e dentre eles faço uma contagem por sexo.

```
14
15 with tabela as (
16     SELECT *
17     FROM churn_clients
18     ORDER BY estimatedsalary_float desc
19     LIMIT 100
20 )
21 select count(*) as "Salário Homem" from tabela
22 where gender = 'Male';
23
24
25
26
```

Result 1 (100)   Result 2 (1)   Result 3 (2)   Result 4 (2)   Result 5 (10)   Result 6 (3)

<input type="checkbox"/>	salário homem	
<input type="checkbox"/>	52	

```
14
15 with tabela as (
16     SELECT *
17     FROM churn_clients
18     ORDER BY estimatedsalary_float desc
19     LIMIT 100
20 )
21 select count(*) as "Salário Mulher" from tabela
22 where gender = 'Female';
23
24
25
26
```

Result 1 (100)   Result 2 (1)   Result 3 (2)   Result 4 (2)

<input type="checkbox"/>	salário mulher	
<input type="checkbox"/>	48	

A gente consegue ver que a equidade aqui no mundo virtual funciona, estão com números bem próximos, 52 homens e 48 mulheres entre os 100 melhores salários. Fica a torcida para se estender pro mundo real.

- **Evasão por quantidade de produtos**

Um dado curioso é que dos 100 clientes que possuem mais produtos, em geral são 4 produtos, 93 saíram do banco. 93%.

Um fator curioso, na minha opinião pode ser que quem usa mais produtos acaba sendo um público mais exigente quanto ao serviço e busca os melhores serviços. Os tornando mais sensíveis a um serviço ruim, ou mais caro e dispostos a se mover.

```
58
59 with tabela as (
60 select * from churn_clients
61 order by NumOfProducts desc
62 limit 100)
63 select count(*) from tabela
64 where exited = 1
65
66
67
68
69
70
```

	count
<input type="checkbox"/>	93

Já quando vimos os 100 que menos tem produtos com o banco, o percentual de evasão cai drasticamente, para 29%. Será que é a parte dos clientes que utilizam pouco e tem poucas exigências para com o banco? Sabe aquele pessoal que até esquece qual banco tem conta ?!

```
58
59 with tabela as (
60 select * from churn_clients
61 order by NumOfProducts asc
62 limit 100)
63 select count(*) from tabela
64 where exited = 1
65
66
67
68
69
70
```

	count
<input type="checkbox"/>	29

- Explorando recursos do SQL :

Aqui estou fazendo um query utilizando como filtro uma subquery. Trazendo todas mulheres que tem idade maior que a idade média do público feminino.

```
68 select * from churn_clients
69 where age > (select avg(age) from churn_clients where gender = 'female')
```

	creditscore	geography	gender	age	tenure	balance	numofproducts	hascard
<input type="checkbox"/>	619	France	Female	42	2	NULL	1	1
<input type="checkbox"/>	608	Spain	Female	41	1	NULL	1	0
<input type="checkbox"/>	502	France	Female	42	8	NULL	3	1
<input type="checkbox"/>	850	Spain	Female	43	2	NULL	1	1
<input type="checkbox"/>	645	Spain	Male	44	8	NULL	2	1
<input type="checkbox"/>	822	France	Male	50	7	NULL	2	1
<input type="checkbox"/>	501	France	Male	44	4	NULL	2	0
<input type="checkbox"/>	616	Germany	Male	45	3	NULL	2	0
<input type="checkbox"/>	653	Germany	Male	58	1	NULL	1	1
<input type="checkbox"/>	587	Spain	Male	45	6	NULL	1	0
<input type="checkbox"/>	732	France	Male	41	8	NULL	2	1
<input type="checkbox"/>	669	France	Male	46	3	NULL	2	0
<input type="checkbox"/>	571	France	Male	44	9	NULL	2	0

Elapsed time: 4790 ms Total rows: 100

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Utilizando o union para selecionar, capturar na query a intersecção de dois “selects”.

```
68 select * from churn_clients
69 where age > (select avg(age) from churn_clients where gender = 'female') and gender = 'female'
71 union
72 select * from churn_clients
73 where exited = 1
```

	creditscore	geography	gender	age	tenure	balance	numofproducts	hascard
<input type="checkbox"/>	511	Spain	Female	66	4	NULL	1	1
<input type="checkbox"/>	623	France	Female	44	6	NULL	2	0
<input type="checkbox"/>	684	Germany	Female	48	10	NULL	1	1
<input type="checkbox"/>	800	France	Female	49	7	NULL	1	0
<input type="checkbox"/>	850	Spain	Female	45	2	NULL	1	1
<input type="checkbox"/>	668	France	Female	46	2	NULL	3	1
<input type="checkbox"/>	652	France	Female	80	4	NULL	2	1
<input type="checkbox"/>	667	Spain	Female	40	1	NULL	1	1
<input type="checkbox"/>	648	France	Female	50	9	NULL	1	1
<input type="checkbox"/>	611	France	Female	40	8	NULL	2	1
<input type="checkbox"/>	519	Spain	Female	57	2	NULL	2	1
<input type="checkbox"/>	601	France	Female	43	8	NULL	3	0
<input type="checkbox"/>	590	France	Female	61	1	NULL	1	1

Elapsed time: 4 ms Total rows: 1 to 100 of 3149 rows

**Observação:**

Algo que aconteceu é que ao enviar os dados do pipeline tiveram duas colunas que ficaram com seus valores nulos a balance e a estimativesalary, porém foram criados duas colunas adicionais balance\_float e estimativesalary\_float no final da tabela que tornou possível o trabalho.

Result 1 (100)	Result 2 (1)	Result 3 (2)	Result 4 (2)	Result 5 (10)	Result 6 (3)	Result 7 (100)	Result 8 (1)	Result 9 (3149)	Export	Chart	
balance	numofproducts	hasccard	isactivemember	estimatedsalary	exited	balance_float	estimatedsalary_float				
NULL	1	1	0	NULL	1	0	1643.11				
NULL	2	0	0	NULL	0	0	167162.44				
NULL	1	1	1	NULL	0	126384.42	198129.36				
NULL	1	0	0	NULL	0	108007.36	47125.11				
NULL	1	1	1	NULL	0	122311.21	19482.5				
NULL	3	1	0	NULL	1	0	89048.46				
NULL	2	1	1	NULL	0	0	188603.06				
NULL	1	1	0	NULL	0	146502.06	19162.89				
NULL	1	1	1	NULL	0	102535.57	189543.19				
NULL	2	1	0	NULL	0	100812.33	147358.27				
NULL	2	1	1	NULL	0	119035.35	29871.79				
NULL	3	0	1	NULL	1	0	110916.15				
NULL	1	1	0	NULL	1	0	64108.56				

Bom , aqui encerro meu trabalho, que trouxe algumas curiosidades desse conjunto de dados. Ao qual precisei utilizar a estrutura da AWS , criar este banco de dados e fazer as pesquisas. Sobre os códigos SQL fiz uso dos métodos de seleção, filtros, ordenação , limitadores , agrupamento junto com métodos de agregação, tabelas temporárias e entre outros. Acredito ter coberto integralmente o conteúdo que vimos de código no curso.

Deixo meus agradecimentos pelas aulas de MVP que foram fundamentais para entender esse processo de nuvem, sem a atenção especial dos professores sobre este assunto não seria possível realizar esse trabalho.

Obrigado,  
**Robson Barbosa.**



