

Introdução

Processos de ETL (Extract, Transform and Load) estão presentes em todos os projetos de dados. O cenário costuma ser o mesmo: fontes de dados diversas com datasets de interesse que precisam ser ingeridos, transformados e armazenados em um ou mais destinos, com formatos diferentes da origem.

Neste laboratório você será guiado na construção de um processo de ETL simplificado utilizando o serviço AWS Glue.

1 - Preparando os dados de origem

Faremos uso do arquivo `nomes.csv`, um dataset que contém os nomes mais comuns de registro de nascimento dos cartórios americanos entre os anos de 1880 e 2014. Trata-se de um arquivo CSV, com a estrutura descrita na amostra a seguir.

```
nome,sexo,total,ano  
Jennifer,F,54336,1983
```

Para nosso laboratório, o arquivo deverá estar em um bucket do S3. Vamos considerar que o path do arquivo seja `s3://{BUCKET}/lab-glue/input/nomes.csv`. Lembre-se que o valor `{BUCKET}` deve ser substituído por um dos disponíveis em sua conta.

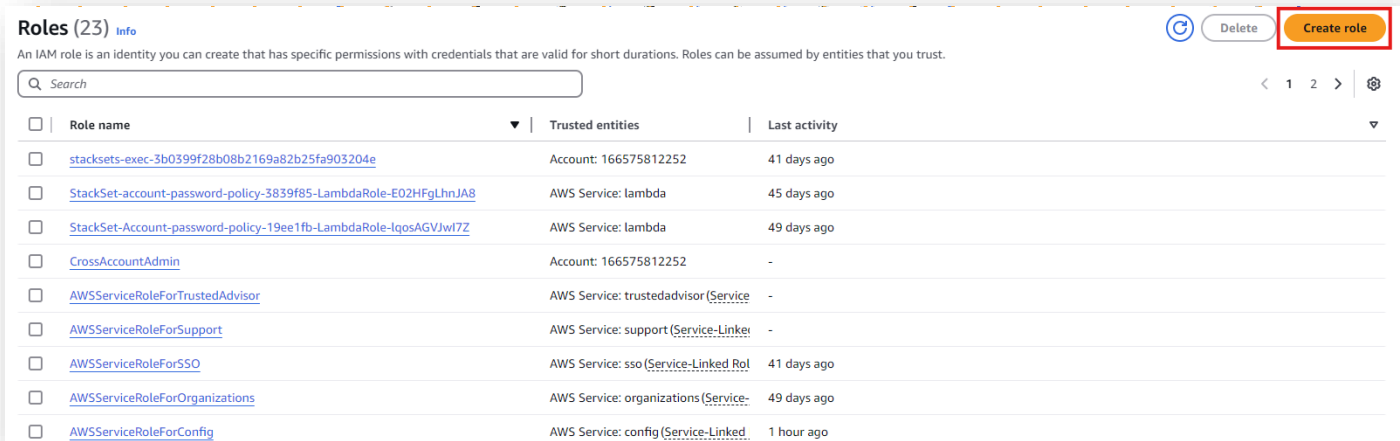
2 - Criando a IAM Role para os jobs do AWS Glue

Você deve estar lembrado que Roles são credenciais temporárias assumidas por serviços e aplicações para realizar operações em favor do usuário.

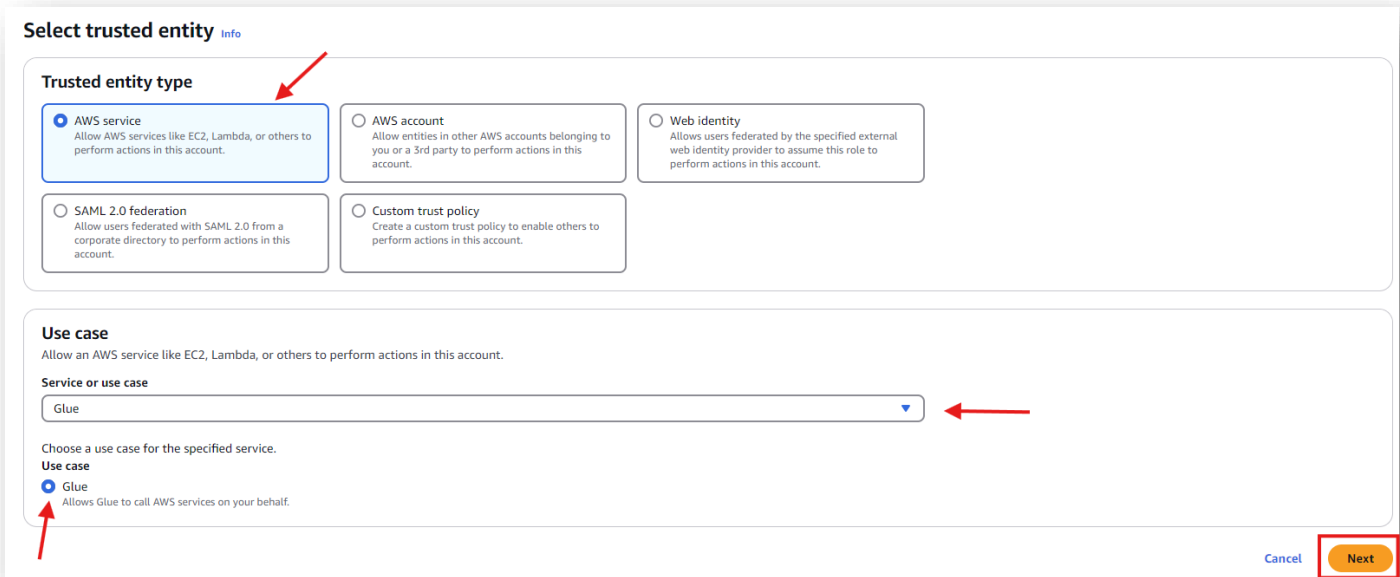
Logo, criaremos uma nova role chamada `AWSGlueServiceRole-Lab4`, associada a políticas geridas pela AWS (`AmazonS3FullAccess`, `AWSLakeFormationDataAdmin`, `AWSGlueConsoleFullAccess` e `CloudWatchFullAccess`). Tais políticas irão permitir acesso ao serviço do Glue ao **S3**, bem como outras ações, como executar códigos via *Notebooks*. Observe que estamos utilizando *políticas* permissivas, o que vai de encontro ao princípio de privilégio mínimo que deve-se seguir em projetos reais. O objetivo aqui é simplificar o processo, apenas.

Vamos aos passos:

- No console, acesse a página do serviço **Identity and Access Management (IAM)** e clique no menu **Roles** à esquerda. Na sequência, clique no botão **Create Role**.



- Na primeira etapa, **Select trusted entity**, escolha *AWS Service* e para **Use Case**, informe *Glue*. Clique em **Next**.



- Na etapa **Add permissions**, pesquise por *AmazonS3FullAccess*, selecione a mesma da lista. Repita o processo para adicionar as demais políticas necessárias: *AWSLakeFormationDataAdmin*, *AWSGlueConsoleFullAccess* e *CloudWatchFullAccess*. Em seguida, clique em **Next**.

Add permissions

Permissions policies (1/996)

Choose one or more policies to attach to your new role.

✕

Filter by Type
All types 1 match

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the ...

▶ Set permissions boundary - optional

CancelPreviousNext

- Na última etapa, informe em Role name o valor `AWSGlueServiceRole-Lab4` e, para finalizar, clique em Create Role.

Step 2: Add permissions

Permissions policy summary

Policy name ⓘ	Type	Attached as
AmazonS3FullAccess	AWS managed	Permissions policy
AWSGlueConsoleFullAccess	AWS managed	Permissions policy
AWSLakeFormationDataAdmin	AWS managed	Permissions policy
CloudWatchFullAccess	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional ⓘ

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

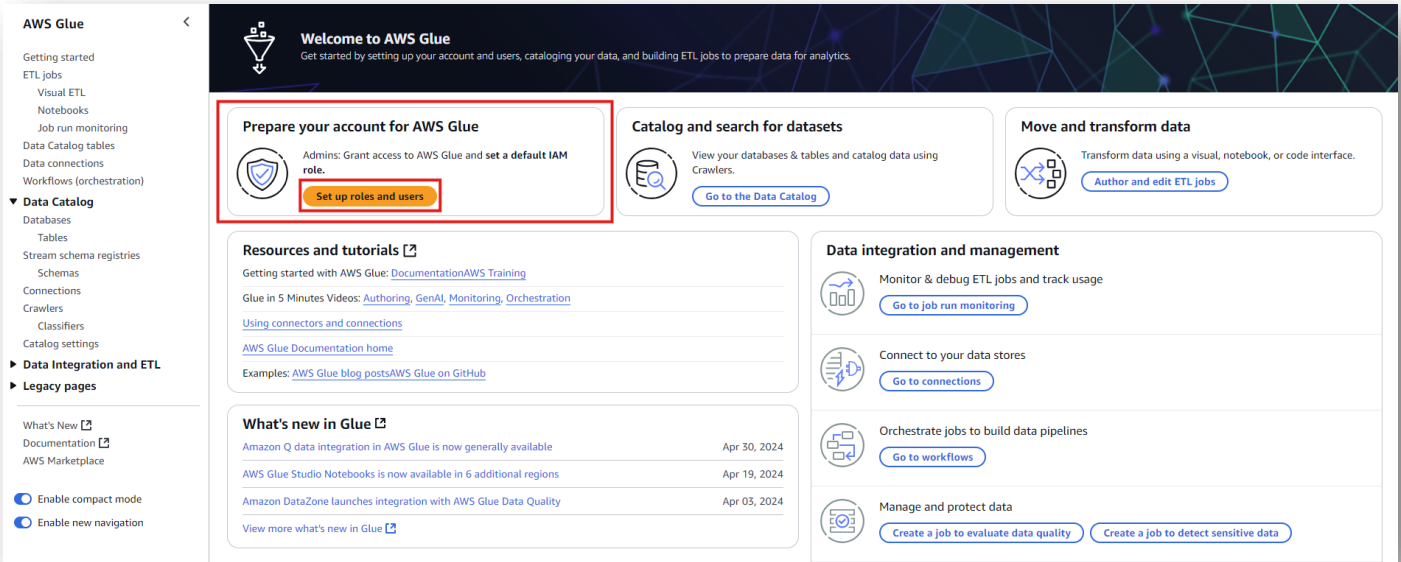
Cancel

Previous

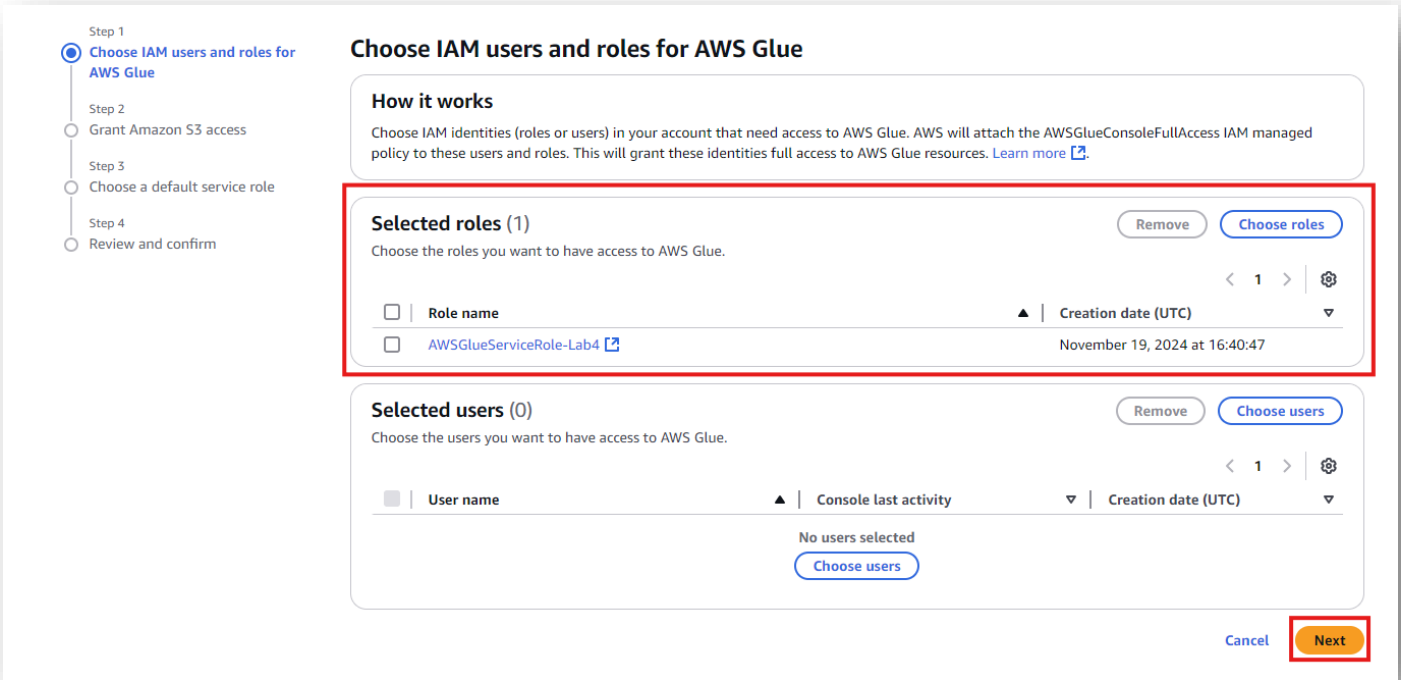
Create role

3 - Configurando sua conta para utilizar o AWS Glue

Acesse a página inicial do serviço AWS Glue. Para que possamos utilizar o serviço com as permissões necessárias, devemos seguir o passo-a-passo disponível a partir da opção **“Set up roles and users”** no card **“Prepare your account for AWS Glue”**.



No primeiro passo devemos indicar quais roles terão acesso ao serviço **AWS Glue**. Procure pela role **“AWSGlueServiceRole-Lab4”** em Choose roles e o adicione à lista, depois pressione **Next**.



No passo seguinte, informe acesso total ao S3 para leitura e escrita.

Step 1
● Choose IAM users and roles for AWS Glue

Step 2
● **Grant Amazon S3 access**

Step 3
○ Choose a default service role

Step 4
○ Review and confirm

Grant Amazon S3 access

How it works
Grant S3 access to the users and roles that you selected in Step 1. Glue will attach permissions to those identities based on the type of access you choose here. [Learn more](#)

Choose S3 locations
Targeted users and roles
0 users and 1 roles

Choose access to Amazon S3

- ☐ No additional access
Do not change permissions.
- ☐ Add access to specific Amazon S3 locations
Choose specific S3 paths that you want to grant access to.
- ☒ **Grant full access to Amazon S3**
Grant access to all S3 resources in your AWS account.

Data access permissions
Set the type of data access for the Glue users and roles.

Data access permissions

- ☐ Read only (*recommended*)
- ☒ **Read and write**

Cancel Previous **Next**

Por fim, marque a opção **“Update the standard AWS Glue service role and set it as the default (recommended)”** e finalize o processo, clicando em **Next** e depois **Apply changes**.

Step 1
● Choose IAM users and roles for AWS Glue

Step 2
● Grant Amazon S3 access

Step 3
● **Choose a default service role**

Step 4
○ Review and confirm

Choose a default service role

How it works
AWS Glue uses an IAM service role to run jobs, access data, and run Data Quality tasks. We recommend that you start with the standard AWSGlueServiceRole as the default. [Learn more](#)

Choose a default AWS Glue service role

IAM role for AWS Glue

- ☒ **Update the standard AWS Glue service role and set it as the default (*recommended*)**
AWS will update the role with the IAM policies needed to run AWS Glue jobs, then set it as the default.
- ☐ Set an existing IAM role as the default
Select an IAM role that you've configured to use as an AWS Glue service role. Glue will set this role as the default, but won't add any permissions to it. [Learn more](#)

① The following IAM role will be created and automatically configured for you:

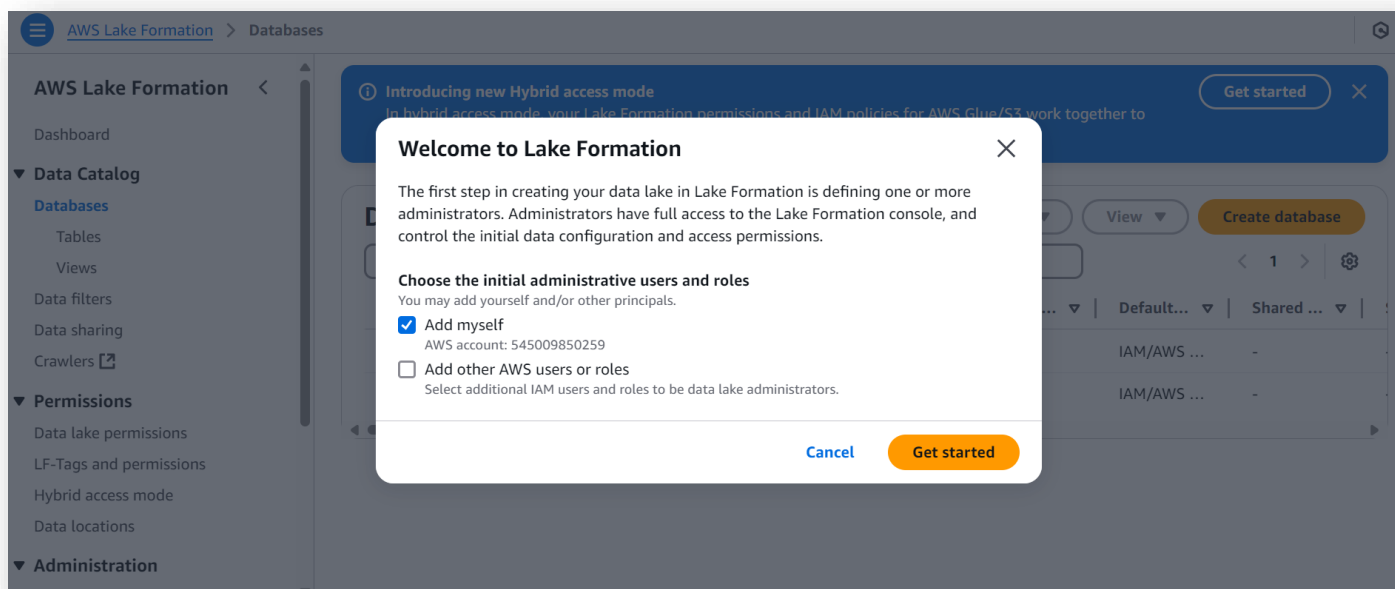
- AWSGlueServiceRole

Cancel Previous **Next**

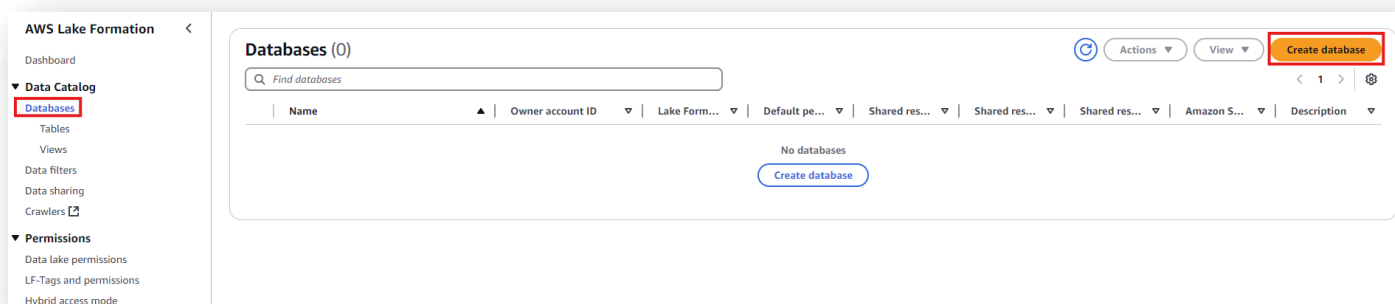
4 - Configurando as permissões no AWS Lake Formation

AWS Lake Formation é um serviço que facilita a criação e gerenciamento de data lakes. Nos iremos utilizá-lo para criar o banco de dados no qual nosso crawler irá adicionar automaticamente uma tabela a partir dos dados armazenados no S3.

No primeiro acesso aparecerá uma tela, selecione **Add myself** e **Get started**.



Após acessar o serviço **AWS Lake Formation** no console, clique na opção **Databases**, no menu à esquerda. Na sequência, clique no botão **Create Database**.



O nome do novo banco deverá ser *glue-lab*. Observe que estamos criando um banco de dados no catálogo do Glue e não um banco de dados das características dos SGBD Relacionais. Clique em **Create Database**.

Create database

Database details

Create a database in the Data Catalog.



Database

Create a database in my account.



Resource link

Create a resource link to a shared database.

Name

Enter a unique name for the database. The name cannot be changed after the database is created. This field is required.

glue-lab

Location - optional

Choose an Amazon S3 path for this database, which eliminates the need to grant data location permissions on catalog table paths that are this location's children

Q e.g.: s3://bucket/prefix/



Browse

Description - optional

Enter a description

Descriptions can be up to 2048 characters long.

Default permissions for newly created tables

This setting maintains existing Data Catalog behavior. You can still set individual permissions, which will take effect when you revoke the Super permission from IAMAllowedPrincipals. See [Changing Default Settings for Your Data Lake](#).

☐ Use only IAM access control for new tables in this database

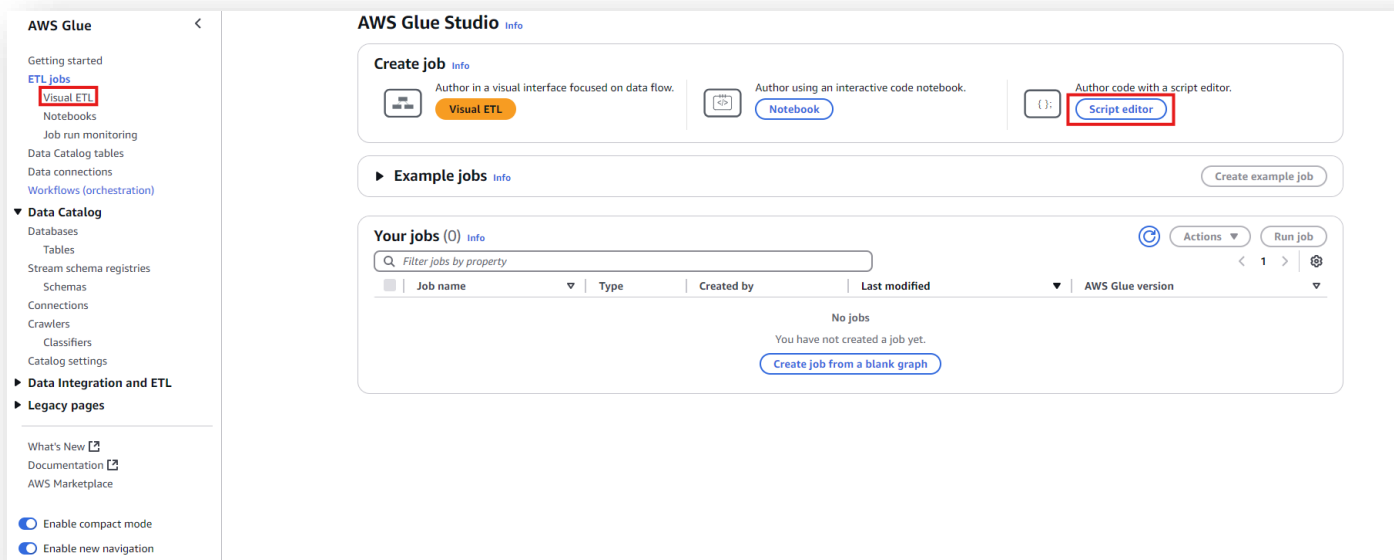
Cancel

Create database

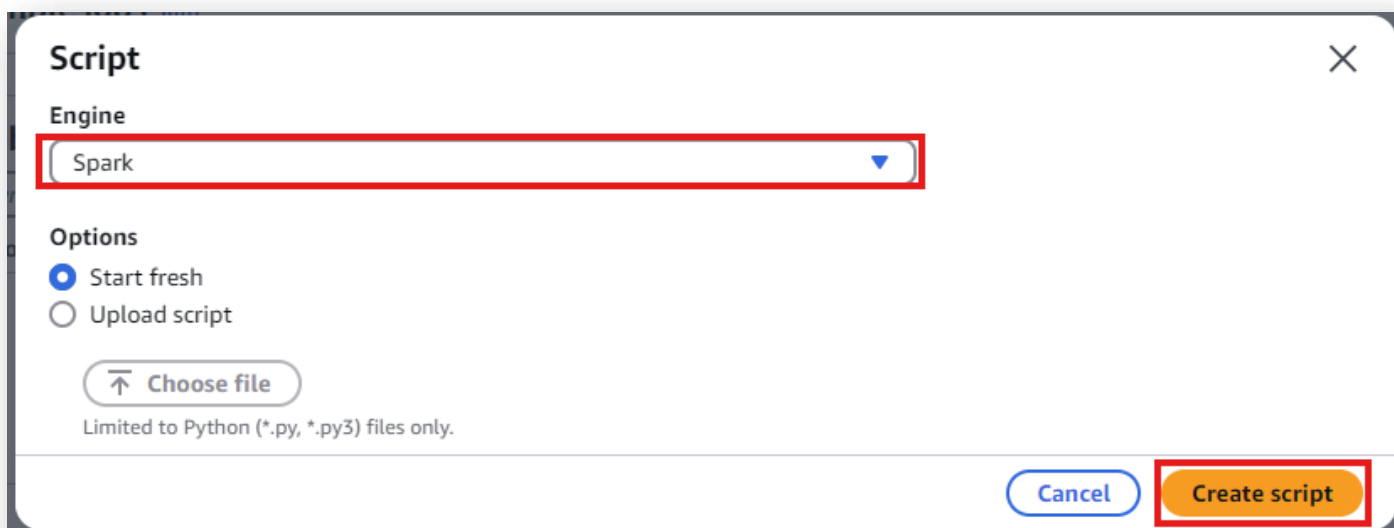
5 - Criando novo job no AWS Glue

Para realizar o processamento do arquivo *nomes.csv* iremos criar um *job* através do serviço **AWS Glue**.

Após acessar a página inicial na console, busque pela opção *Visual ETL* em *ETL jobs* no menu à esquerda. Depois clique em **Script editor**.



Abrirá uma tela, selecione **Spark** em *Engine* e clique em **Create script**.



Vá em **Job details**:

The screenshot shows the 'Job details' tab in the AWS Glue console. The 'Basic properties' section is active, displaying the following configuration:

- Name:** job_aws_glue_lab_4
- Description - optional:** (Empty text box)
- IAM Role:** AWSGlueServiceRole-Lab4
- Type:** Spark
- Glue version:** Glue 3.0 - Supports spark 3.1, Scala 2, Python 3
- Language:** Python 3
- Worker type:** G 1X (4vCPU and 16GB RAM)

Essas serão as propriedades:

- Name: Corresponde ao nome do job. Informe *job_aws_glue_lab_4*;
- IAM Role: Informe *AWSGlueServiceRole-Lab4*;
- Type: Mantenha *Spark*.
- Glue version: *Glue 3.0*
- Language: Python 3
- Worker Type: Escolha G 1x, ou seja, opção com menos vCPUs e RAM.
- Opção *Automatically scale the number of workers* deve estar **desmarcada**.
- Requested number of workers: 2.
- Number of retries: 0.
- Job timeout (minutes): 5.

(demais opções permanecem iguais)

Em Advanced properties, informe:

- Script filename: Defina o nome do seu script.
- Spark UI: Desmarque a opção.

(demais opções permanecem iguais)

Agora clique em **Save**.

Você deve ter percebido que na aba Script há um código base para você iniciar o desenvolvimento. O código é semelhante a este:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
job.commit()
```

Perceba que o código já oferece um objeto de sessão do Spark (`spark = glueContext.spark_session`) que você pode utilizar para realizar as atividades propostas na sequência.

Todo código que você construir, deverá estar entre os comandos `job.init(args['JOB_NAME'], args)` e `job.commit()` .

Vamos imaginar que o objetivo seja ler um arquivo CSV do S3, filtrar os dados pelo ano de 1934 e armazenar o resultado para PARQUET, em outro local do S3. O código a seguir realiza justamente tal tarefa.

Para abordar parâmetros, vamos considerar a existência de 2 neste job:

- **S3_INPUT_PATH:** Indica nosso caminho da origem no S3 (incluir o nome do arquivo).
- **S3_TARGET_PATH:** Indica nosso caminho de destino no S3.

Os parâmetros são utilizados para tornar o código flexível, genérico. Este é sempre um ponto importante a considerar. Você pode criá-los na aba *Job Details*, opção *Advanced Options*, *Job parameters*. Eles devem iniciar com --.

job_aws_glue_lab_4

Script

Job details

Runs

Data quality

Schedules

Version Control

Dependent JARs path

Referenced files path

Job parameters

Info

Key

Value - optional

Q

--S3_INPUT_PATH

X

Q

s3://[REDACTED]/nomes.csv

X

Remove

Q

--S3_TARGET_PATH

X

Q

s3://[REDACTED]

X

Remove

Add new parameter

You can add 48 more parameters.

Veja o código de exemplo:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME', 'S3_INPUT_PATH', 'S3_TARGET_PATH'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

source_file = args['S3_INPUT_PATH']
target_path = args['S3_TARGET_PATH']

df = glueContext.create_dynamic_frame.from_options(
    "s3",
    {
        "paths": [
            source_file
        ]
    },
    "csv",
    {"withHeader": True, "separator": "|"},
)

only_1934 = df.filter(lambda row: row['anoLancamento']=='1934')

glueContext.write_dynamic_frame.from_options(
    frame = only_1934 ,
    connection_type = "s3",
    connection_options = {"path": target_path},
    format = "parquet")

job.commit()
```

No exemplo estamos utilizando dynamic frames, uma abstração sobre um dataframe Spark oferecida pelo Glue. Naturalmente nós podemos alternar entre dynamic frames e dataframes conforme demonstramos no exemplo que segue.

#Obtendo um dataframe Spark a partir de um dataframe dinâmico do Glue

```
spark_df = my_dynamic_df.toDF()
```

#Obtendo um dataframe dinâmico do Glue a partir de um dataframe Spark

```
dynamic_df = DynamicFrame(spark_df, glueContext)
```

Para mais informações sobre o desenvolvimento de jobs ETL com Glue, você pode acessar o endereço [Program AWS Glue ETL scripts in PySpark](#).

Para executar o Job, clique em **Run**.

5.1 - Eliminando execuções de jobs

Após executar jobs, devemos nos certificar que não haja sessões em execução desnecessárias. Para tal, acesse a opção *Job run monitoring* no menu à esquerda do Console. Caso haja execuções em andamento e você não precisa mais delas, solicite a finalização delas. O processo é simples e compreende escolher a execução e ir até o botão **Actions**, opção **Stop run**.

The screenshot shows the AWS Glue Monitoring console. On the left, the 'Job run monitoring' option is highlighted in the navigation menu. The main area displays a 'Monitoring' dashboard with a 'Job runs summary' section showing statistics: Total runs (10), Running (1), Canceled (1), Successful runs (2), Failed runs (6), Run success rate (25%), and DPU hours (1). Below this, a table lists 10 job runs. The 'Actions' menu is open for the selected job 'job glue teste', showing options like 'Stop run', 'View job', and 'Rewind job bookmark'.

Job name	Run status	Type	Start time (Local)	End time (Local)	Duration	Attempts	Worker type	DPU hours
job_aws_glue_lab_4	Succeeded	Glue ETL	11/21/2024 15:26:54	11/21/2024 15:28:30			G.1X	0.05
job glue teste	Succeeded	Glue ETL	11/19/2024 16:51:36	11/19/2024 16:53:09	1 minute	2	G.1X	0.05
job_aws_glue_lab_4	Succeeded	Glue ETL	11/19/2024 15:20:18	11/19/2024 15:22:32	2 minutes	2	G.1X	0.07

5.2 Sua vez!

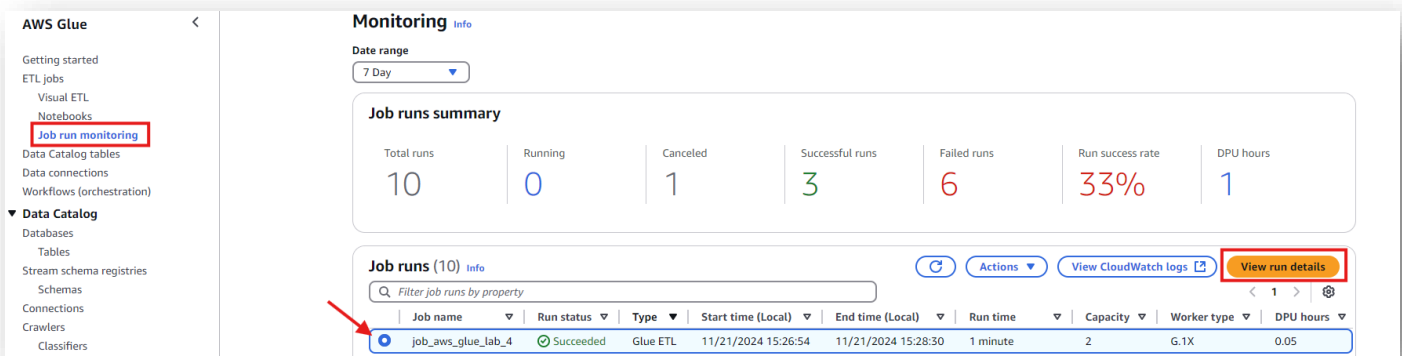
Agora vamos construir um *job Glue* nos moldes dos exemplos anteriores. Seguem os passos para você desenvolver:

- Ler o arquivo nomes.csv no S3 (lembre-se de realizar upload do arquivo antes).
- Imprimir o schema do dataframe gerado no passo anterior.
- Escrever o código necessário para alterar a caixa dos valores da coluna nome para MAIÚSCULO.
- Imprimir a contagem de linhas presentes no dataframe.
- Imprimir a contagem de nomes, agrupando os dados do dataframe pelas colunas ano e sexo. Ordene os dados de modo que o ano mais recente apareça como primeiro registro do dataframe.
- Apresentar qual foi o nome feminino com mais registros e em que ano ocorreu.
- Apresentar qual foi o nome masculino com mais registros e em que ano ocorreu.

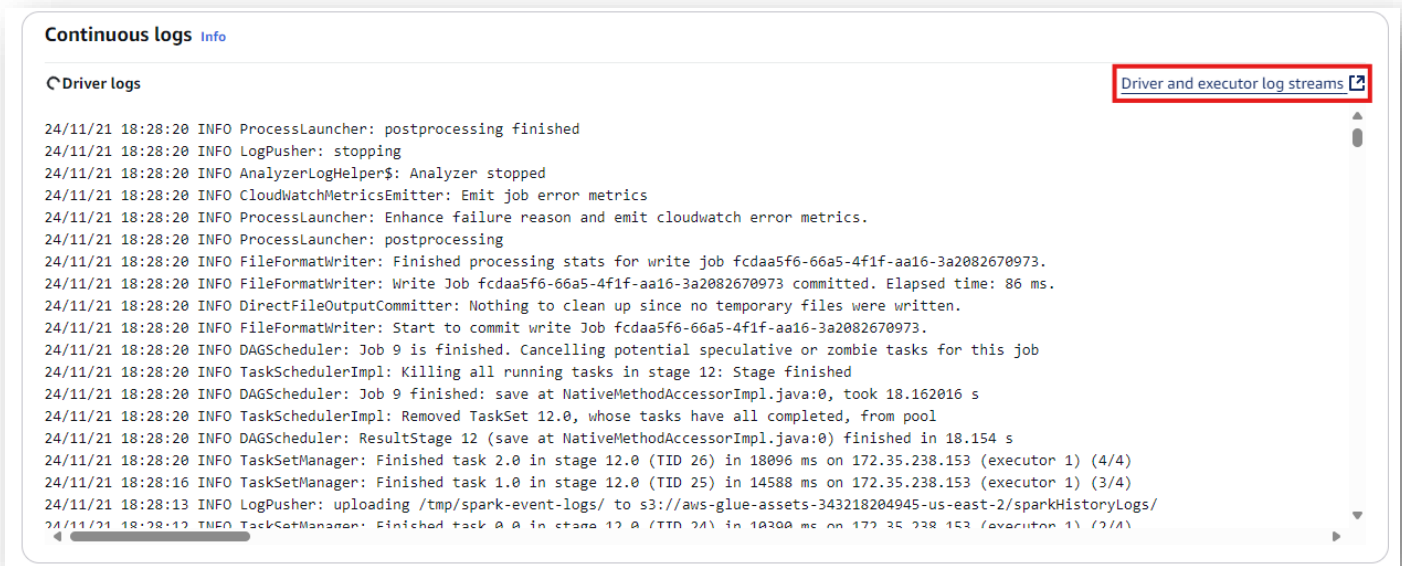
- Apresentar o total de registros (masculinos e femininos) para cada ano presente no dataframe. Considere apenas as primeiras 10 linhas, ordenadas pelo ano, de forma crescente.
- Escrever o conteúdo do dataframe com os valores de nome em maiúsculo no S3.
 - Atenção aos requisitos:
 - A gravação deve ocorrer no subdiretório *frequencia_registro_nomes_eua* do path *s3:///lab-glue/*
 - O formato deve ser JSON
 - O particionamento deverá ser realizado pelas colunas sexo e ano (nesta ordem)

Algumas dicas:

- Os logs de execução estarão disponíveis através do menu **Job run monitoring**, opção **View run details**.



- Dentre as opções apresentadas, busque pela seção **CloudWatch continuous logs**.



- Você encontrará informações complementares sobre desenvolvimento de Scripts ETL com Glue na [documentação oficial do produto](#).

6 - Criando crawler

Crawlers são mecanismos que podemos utilizar para monitorar nosso armazenamento de dados de modo a criar/atualizar metadados no catálogo do Glue de forma automática.

Na sequência iremos desenvolver um crawler para automaticamente criar uma tabela chamada **frequencia_registro_nomes_eua** a partir dos dados escritos no S3 (verifique a última atividade do notebook).

Vamos aos passos para criação de nosso crawler:

- No console, acesse o serviço **AWS Glue**. Na página do serviço, escolha a opção **Crawlers** no menu à esquerda. Na sequência, clique no botão **Create**.
- No primeiro passo de criação do **Crawler**, informe **FrequenciaRegistroNomesCrawler** no campo **Name**. Clique em **Next**.

The screenshot shows the 'Set crawler properties' step in the AWS Glue console. On the left, a vertical navigation pane lists five steps: 'Set crawler properties' (selected), 'Choose data sources and classifiers', 'Configure security settings', 'Set output and scheduling', and 'Review and create'. The main content area is titled 'Set crawler properties' and contains a 'Crawler details' section. Under 'Name', the text 'FrequenciaRegistroNomesCrawler' is entered in a text box. Below this, the 'Description - optional' section has a text box with the placeholder 'Enter a description'. At the bottom right of the form, there are 'Cancel' and 'Next' buttons, with the 'Next' button highlighted by a red rectangle.

- Em **Choose data sources and classifiers**, devemos informar o caminho do S3 a ser monitorado. Para **Is your data already mapped to Glue tables?**, informe **Not yet**. E, na sequência, clique em **Add a data source**.

The screenshot shows the 'Choose data sources and classifiers' step in the AWS Glue console. The left navigation pane shows 'Choose data sources and classifiers' as the selected step. The main content area is titled 'Choose data sources and classifiers' and includes a 'Data source configuration' section. It asks 'Is your data already mapped to Glue tables?' with two radio button options: 'Not yet' (selected) and 'Yes'. Below this, the 'Data sources (0)' section shows a table with columns 'Type', 'Data source', and 'Parameters'. The table is currently empty, with a message 'You don't have any data sources.' and an 'Add a data source' button. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons, with the 'Add a data source' button and the 'Next' button highlighted by red rectangles.

- Na tela aberta, em **Data source**, certifique que esteja S3. Em **Location of S3 data**, informe **In this account**. Finalmente, no campo **S3 path**, informe o caminho `s3:///lab-glue/frequencia_registro_nomes_eua/`, lembrando de substituir pelo utilizado anteriormente.

Add data source [X]

Data source
Choose the source of data to be crawled.

S3

Network connection - optional
Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

[Empty dropdown menu] [Refresh icon]

[Clear selection] [Add new connection]

Location of S3 data

☒ In this account
☐ In a different account

S3 path
Browse for or enter an existing S3 path.

[Search icon] [Redacted path] /frequencia_registro_nomes_eua/ [X] [View] [Browse S3]

All folders and files contained in the S3 path are crawled. For example, type `s3://MyBucket/MyFolder/` to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs
This field is a global field that affects all S3 data sources.

☒ Crawl all sub-folders
Crawl all folders again with every subsequent crawl.

☐ Crawl new sub-folders only
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

☐ Crawl based on events
Rely on Amazon S3 events to control what folders to crawl.

☐ Sample only a subset of files

☐ Exclude files matching pattern

[Cancel] [Add an S3 data source]

- Na etapa **Configure security settings** informe a role `AWSGlueServiceRole-Lab4` no campo **Existing IAM role**. Avance clicando em **Next**.

Step 1
● Set crawler properties

Step 2
● Choose data sources and classifiers

Step 3
● **Configure security settings**

Step 4
○ Set output and scheduling

Step 5
○ Review and create

Configure security settings

IAM role [info](#)

Existing IAM role

AWSGlueServiceRole-Lab4 [View](#)

[Create new IAM role](#) [Update chosen IAM role](#)

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

Lake Formation configuration - optional

Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more](#)

☐ Use Lake Formation credentials for crawling S3 data source

Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

► **Security configuration - optional**

Enable at-rest encryption with a security configuration.

[Cancel](#) [Previous](#) [Next](#)

- Em **Set output and scheduling**, no campo **Target database**, informe **glue-lab**. Em **Crawler schedule**, no campo **Frequency**, defina **On Demand**. Clique em **Next**.

Step 1
● Set crawler properties

Step 2
● Choose data sources and classifiers

Step 3
● Configure security settings

Step 4
● **Set output and scheduling**

Step 5
○ Review and create

Set output and scheduling

Output configuration [info](#)

Target database

glue-lab [Clear selection](#) [Add database](#)

Table name prefix - optional

Type a prefix added to table names

Maximum table threshold - optional

This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.

Type a number greater than 0

► **Advanced options**

Crawler schedule

You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. [Learn more](#)

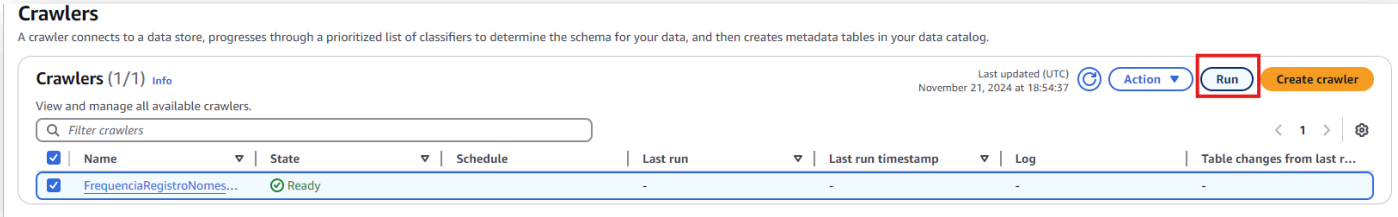
Frequency

On demand

[Cancel](#) [Previous](#) [Next](#)

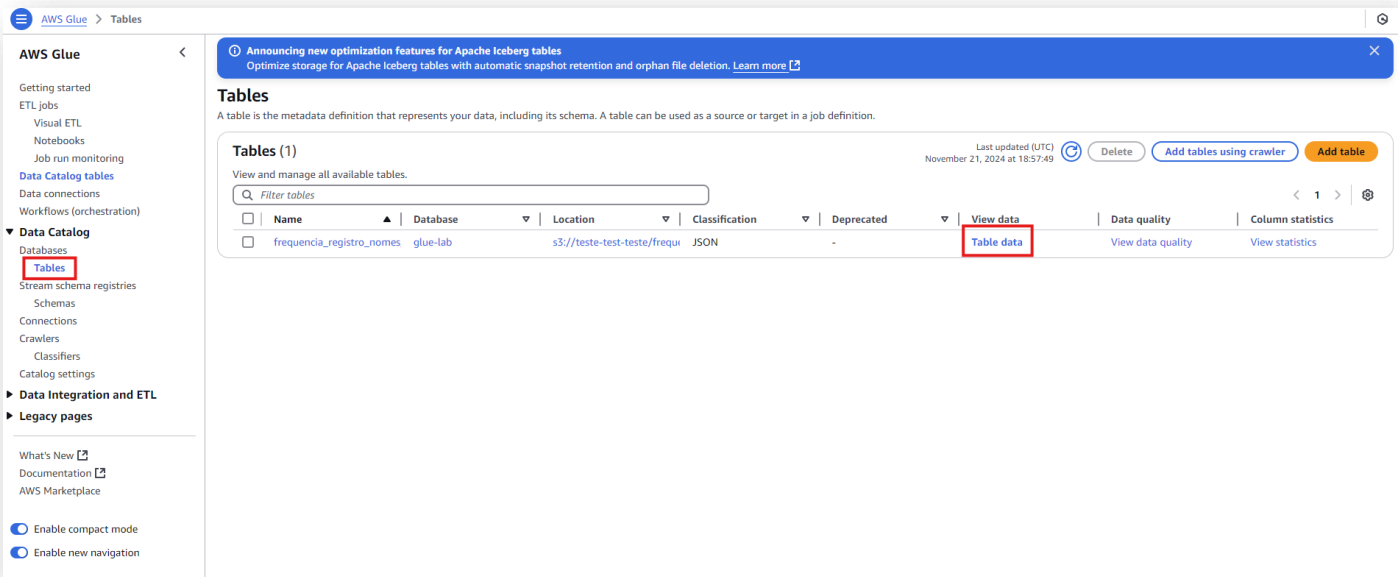
- Avance e finalize o processo de criação clicando em **Create Crawler**. Crawler criado, agora vamos executá-lo.

Na tela inicial (Crawlers), selecione **FrequenciaRegistroNomesCrawler** e clique em **Run**. A execução pode levar alguns segundos e você pode acompanhar o resultado na própria tela em que está.

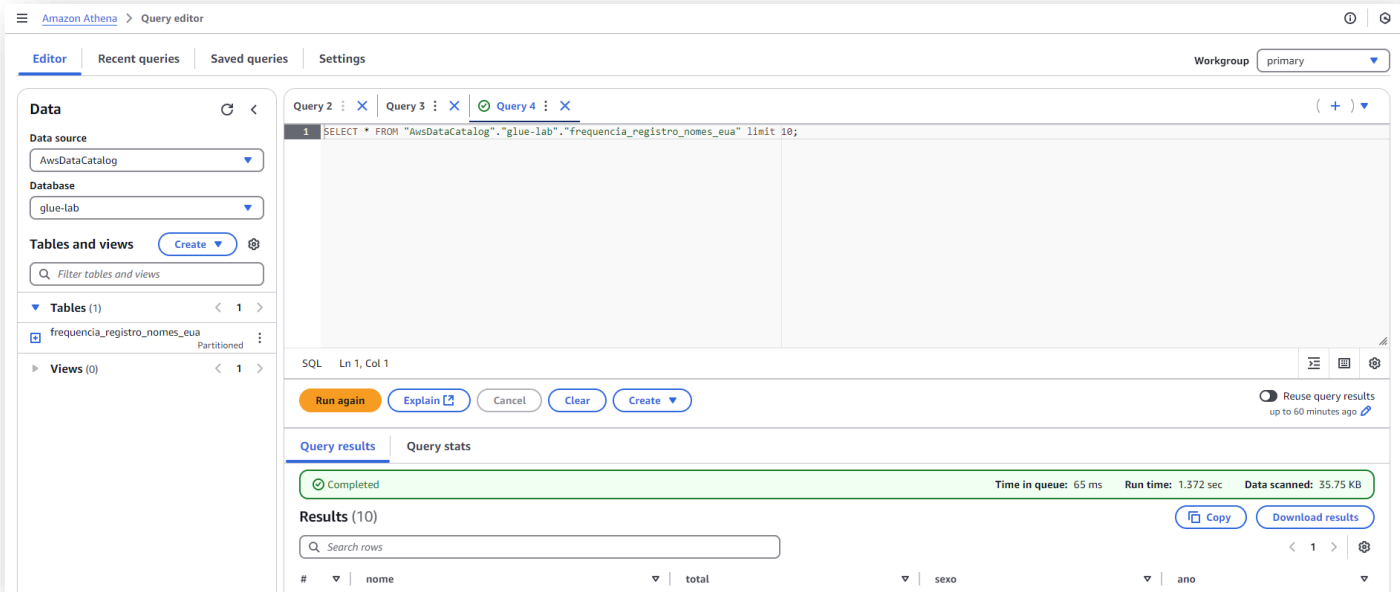


Se a execução for bem-sucedida, uma nova tabela, de nome frequencia_registro_nomes_eua será criada na base *glue-lab*. Você pode vê-la por meio do Glue Catalog e no Athena.


Um jeito de testar é no Glue, clique em **Tables** na esquerda, e depois **Table data** e depois **Proceed**.



Abrirá o Athena com a o comando SQL já criado para fazer um Select. Certifique-se de estar como a imagem abaixo:



No primeiro acesso, precisará informar onde serão salvas as queries feitas:

 Before you run your first query, you need to set up a query result location in Amazon S3.

Edit settings