AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Data Catalog
Databases
Tables
Stream schema registries
Schemas

job_aws_glue_lab_4

Last modified on 21/05/2023, 00:21:46 | Try new UI | Actions | Save | Run

Successfully updated job
Successfully updated job job_aws_glue_lab_4. To run the job choose the **Run** Job button.

Script | Job details | Runs | Data quality New | Schedules | Version Control

Script Info

```python
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
from pyspark.sql import functions as F

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['job_aws_glue_lab_4','S3_INPUT_PATH','S3_TARGET_PATH'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['job_aws_glue_lab_4'], args)

# Ler o arquivo CSV
df = glueContext.create_dynamic_frame.from_options(
    "s3",
    {
        "paths": [args['S3_INPUT_PATH']],
    },
    "csv",
    {
        "withHeader": True,
        "separator": ","
    },
)

# Imprimir o schema do dataframe
df.printSchema()

# Alterar a caixa dos valores da coluna nome para MAIÚSCULO
df = df.toDF()
df = df.withColumn('nome', F.upper(F.col('nome')))
df = DynamicFrame.fromDF(df, glueContext, "df")

# Imprimir a contagem de linhas presentes no dataframe
print("Número de linhas: ", df.count())

# Imprimir a contagem de nomes, agrupando os dados do dataframe pelas colunas ano e sexo.
df.groupBy("ano", "sexo").agg(F.sum("total").alias("contagem")).show()

# Ordene os dados de modo que o ano mais recente apareça como primeiro registro do dataframe.
df = df.orderBy(df.ano.desc())

# Apresentar qual foi o nome feminino com mais registros e em que ano ocorreu.
nome_feminino_mais_registrado = df.filter(df.sexo == 'F').groupBy('nome', 'ano').sum('total').orderBy('sum(total)',
    ascending=False).first()
print(f"Nome feminino mais registrado: {nome_feminino_mais_registrado['nome']} no ano
    {nome_feminino_mais_registrado['ano']}")

# Apresentar qual foi o nome masculino com mais registros e em que ano ocorreu.
nome_masculino_mais_registrado = df.filter(df.sexo == 'M').groupBy('nome', 'ano').sum('total').orderBy('sum(total)',
    ascending=False).first()
print(f"Nome masculino mais registrado: {nome_masculino_mais_registrado['nome']} no ano
    {nome_masculino_mais_registrado['ano']}")

# Apresentar o total de registros (masculinos e femininos) para cada ano presente no dataframe.
df.groupBy('ano').sum('total').show()

# Considere apenas as primeiras 10 linhas, ordenadas pelo ano, de forma crescente.
df = df.orderBy(df.ano.asc())
df = DynamicFrame.fromDF(df.limit(10), glueContext, "df")

# Escrever o conteúdo do dataframe com os valores de nome em maiúsculo no S3.
glueContext.write_dynamic_frame.from_options(
    frame = df,
    connection_type = "s3",
    connection_options = {
        "path": args['S3_TARGET_PATH'] + "/frequencia_registro_nomes_eua",
        "partitionKeys": ["sexo", "ano"]
    },
    format = "json"
```

```
)
job.commit()

```

Python Ln 67, Col 22 Errors: 0 Warnings: 0

Connections
Crawlers
Classifiers