



MVP: PIPELINE DE DADOS NA PLATAFORMA AWS

Relatório final da Sprint III - Engenharia de dados

Sumário

Este relatório tem como objetivo mostrar o passo a passo durante a construção do pipeline de dados na plataforma AWS, incluindo a busca, coleta, modelagem, carga e análise de dados.

Annanda M. Silveira
nanda_masi@homail.com

Confidential C

Conteúdos

Objetivo.....	2
Busca pelos dados	2
Coleta	2
Carga	3
Etapa 1: <i>Data source – S3 bucket</i>	3
Etapa 2: <i>Transform – Change Schema</i>	5
Etapa 3: <i>Data target – Amazon Redshift</i>	7
Catálogo de dados.....	10
Análise	11
Qualidade dos dados	11
Solução do problema	12

Objetivo

Este trabalho tem como intuito analisar, a partir da base de dados “Global Cargo Ships Dataset”, a relação dos navios com suas respectivas medidas de volume interno, capacidade total de carga e tamanho. Com a análise das características mencionadas, tem-se como objetivo responder as seguintes questões:

1. Qual é a capacidade de carga do navio mais novo e do mais antigo?
2. Qual é a medida do volume interno do navio mais novo e mais antigo?
3. É possível deduzir que o volume interno esteja diretamente relacionado com a capacidade total de carga?
4. Quais foram os três navios mais produzidos?
5. Qual navio foi produzido em todos os anos?
6. Qual navio foi produzido em apenas um ano?
7. Por quais motivos os navios das questões anteriores foram produzidos todos os anos e apenas por um ano?

Busca pelos dados

Os dados escolhidos foram extraídos a partir da base pública Kaggle. Os dados podem ser encontrados no site a partir do endereço a seguir:

`<https://www.kaggle.com/datasets/ibrahimonmars/global-cargo-ships-dataset?select=Ship_Uncleaned.csv>`

Coleta

Primeiramente os dados foram baixados na máquina local e, posteriormente, inseridos manualmente em um bucket do S3, conforme a Figura 1.

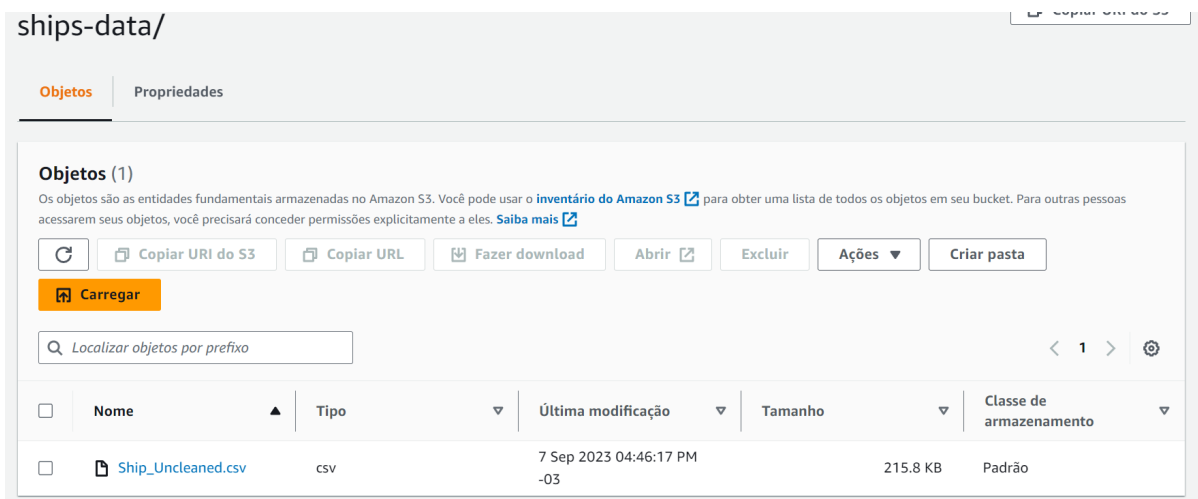


Figura 1: bucket S3 com csv de dados

Carga

O ETL foi realizado usando o serviço AWS Glue. Através da sua interface visual foram criadas as seguintes etapas de extração, transformação e carga.

Etapa 1: *Data source – S3 bucket*

Primeiramente foi criado um job com nome “ships_job” a partir do Amazon S3 com objetivo no Amazon Redshift, conforme a Figura 2, que apresenta visual gráfico conforme Figura 3.

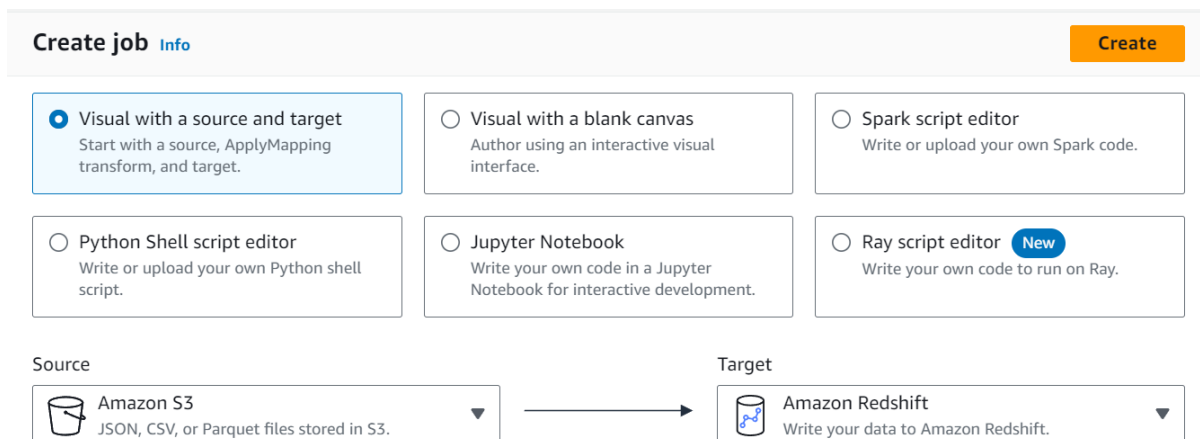


Figura 2: Criação do job

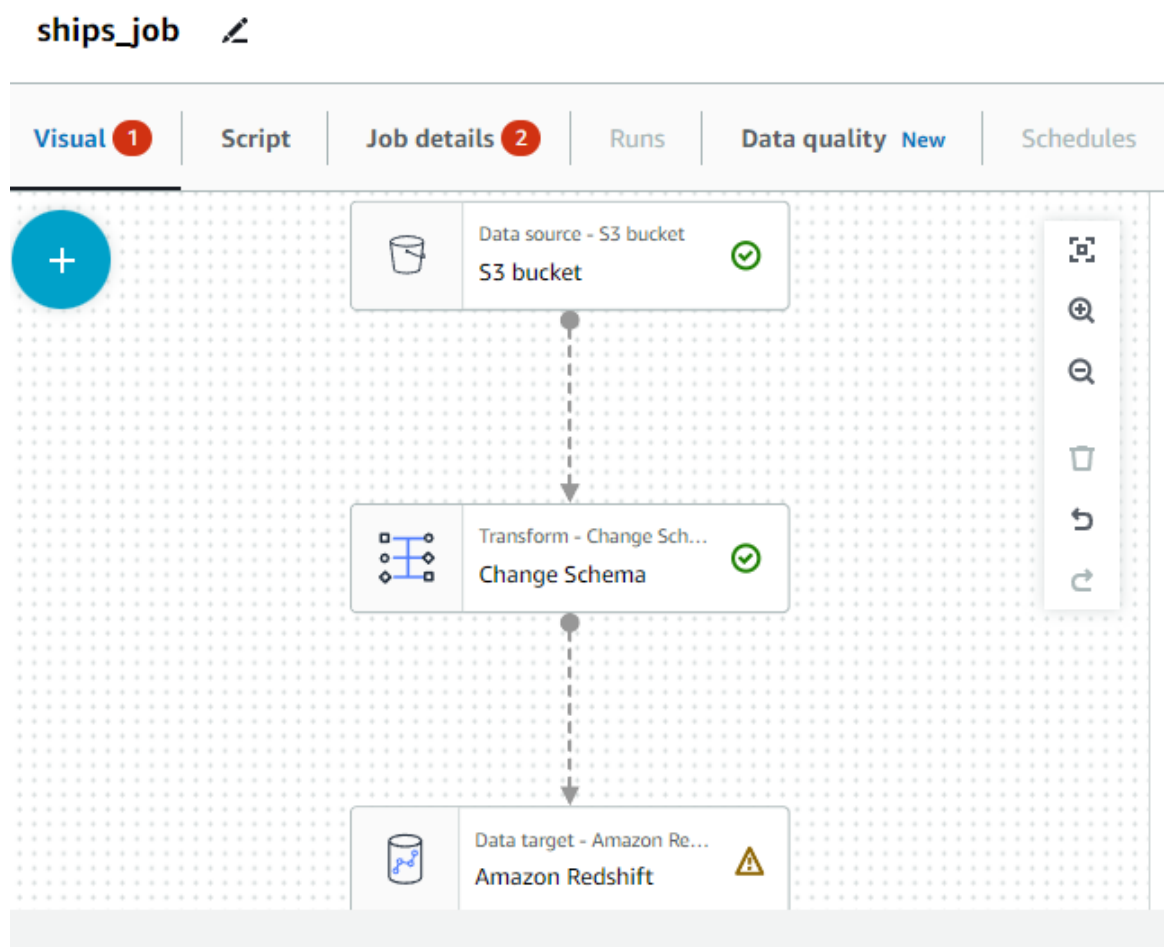


Figura 3: Visual gráfico do job “ships_job”

Então, foram realizadas as configurações para extrair (Extract) os dados da fonte, no caso a pasta “ship_uncleaned” do bucket “mvp3_data_pipeline”, conforme Figura 4.

S3 URL


☒ Recursive
Read files in all subdirectories.

Data format

Delimiter

Figura 4: Extração dos dados da fonte

O esquema foi auto detectado: arquivo .csv, separado por vírgulas, contendo as chaves “Company_Name”, “ship_name”, “built_year”, “gt”, “dwt” e “size”, sendo todas do tipo *string*, conforme Figura 5.

ships_job 

Visual 1	Script	Job details 2	Runs	Data quality New	Schedules
Data source properties - S3		Output schema	Data preview		
Key	Data type	Partition			
Company_Name	string	-			
ship_name	string	-			
built_year	string	-			
gt	string	-			
dwt	string	-			
size	string	-			

Figura 5: Esquema do job

Etapa 2: Transform – Change Schema

Foi realizado a etapa de transformação (Transform) dos dados, convertendo os tipos dos campos “built_year” para date e “gt”, “dwt” e “size” para int. O campo “gt” foi renomeado para gross_tonnage e o campo “dwt” foi renomeado para deadweight_tonnage. Nenhuma chave foi excluída, pois o esquema já é enxuto e encontra apenas as informações relevantes. A transformação dos dados pode ser visualizada na Figura 6, enquanto o esquema transformado na Figura 7.

Transform

Output schema

Data preview

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
Company_Name	company_name	string ▼	<input type="checkbox"/>
ship_name	ship_name	string ▼	<input type="checkbox"/>
built_year	built_year	date ▼	<input type="checkbox"/>
gt	gross_tonnage	int ▼	<input type="checkbox"/>
dwt	deadweight_tonnage	int ▼	<input type="checkbox"/>
size	size	int ▼	<input type="checkbox"/>

Figura 6: Transformação dos dados

Key	Data type
company_name	string
ship_name	string
built_year	date
gross_tonnage	int
deadweight_tonnage	int
size	int

Figura 7: Esquema de dados transformados

Etapa 3: *Data target* – Amazon Redshift

Um ambiente no Amazon Redshift foi criado, com, principalmente, configurações padrões para carregamento (*Load*) dos dados transformados no banco de dados. Porém, algumas configurações foram personalizadas, como a de capacidade básica de RPU para 8 e VPC com 4 sub-redes (região de Oregon). A Figura 8 ilustra a configuração finalizada.

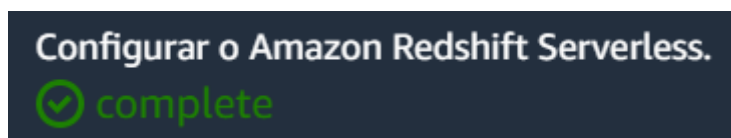
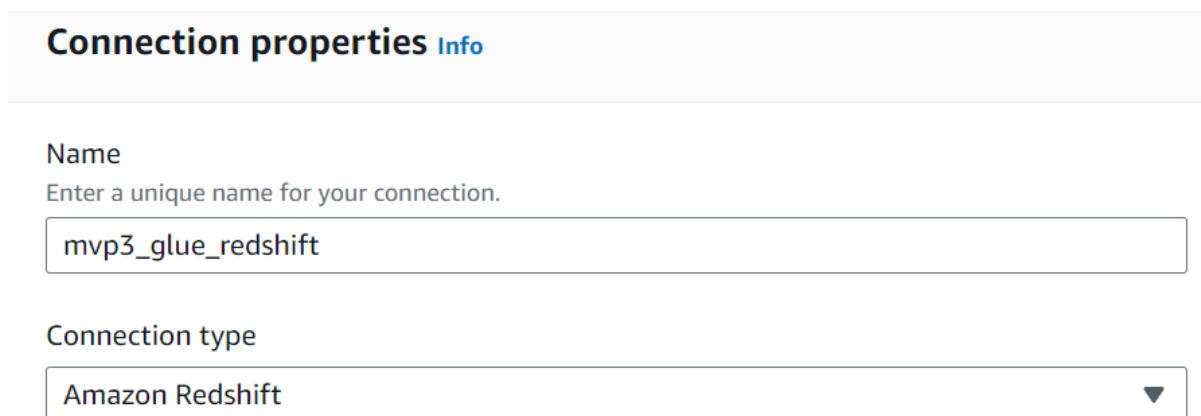


Figura 8: Configuração Amazon Redshift finalizada

Após, foi necessário criar uma conexão, como ilustra a Figura 9, 10 e 11.

The screenshot shows the "Connection properties" section of the Amazon Redshift console. It has a title bar with "Connection properties" and an "Info" link. Below the title bar, there are two fields: "Name" with a placeholder "Enter a unique name for your connection." and a text input containing "mvp3_glue_redshift"; and "Connection type" with a dropdown menu showing "Amazon Redshift".

Connection properties [Info](#)

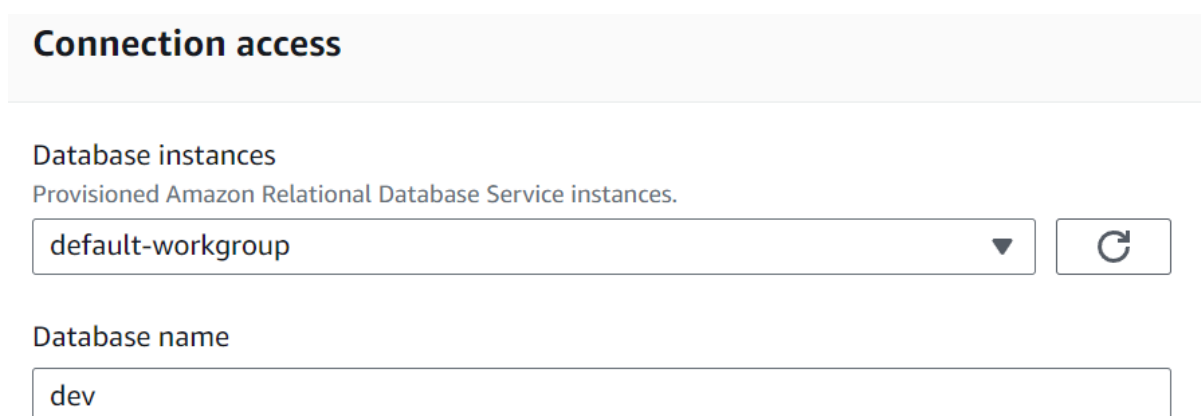
Name
Enter a unique name for your connection.

mvp3_glue_redshift

Connection type


Amazon Redshift ▼

Figura 9: Criando uma conexão – propriedades

The screenshot shows the "Connection access" section of the Amazon Redshift console. It has a title bar with "Connection access". Below the title bar, there are two fields: "Database instances" with a placeholder "Provisioned Amazon Relational Database Service instances.", a dropdown menu showing "default-workgroup", and a refresh button; and "Database name" with a text input containing "dev".

Connection access

Database instances
Provisioned Amazon Relational Database Service instances.

default-workgroup ▼ 

Database name

dev

Figura 10: Criando uma conexão – acesso

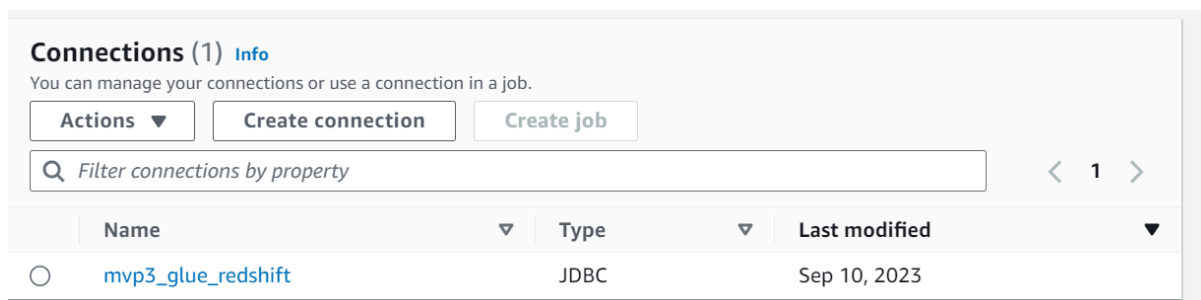


Figura 11: Conexão criada

Então, foi feito um teste de conexão com a função `mvp3_glue`, recém-criada com acesso de administrador.

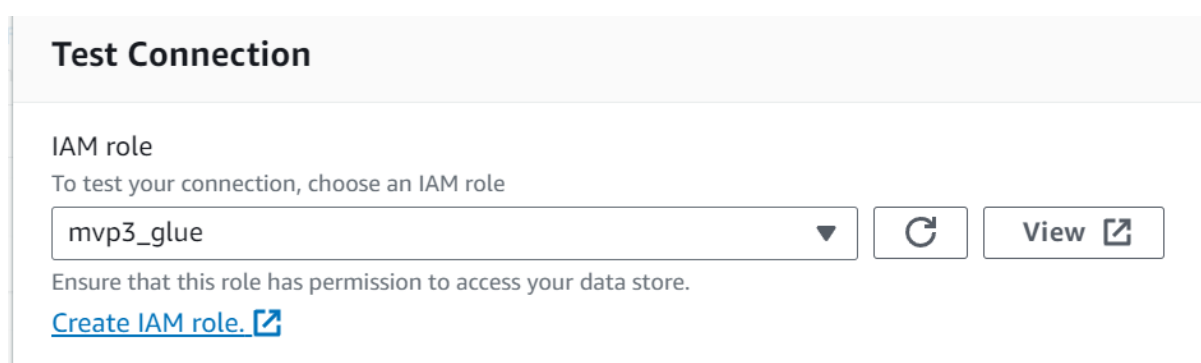


Figura 12: Teste de conexão

O primeiro teste teve erro, pois não havia endpoint entre o Glue. Para solucionar o problema foi criado o endpoint, como mostra a Figura 13.

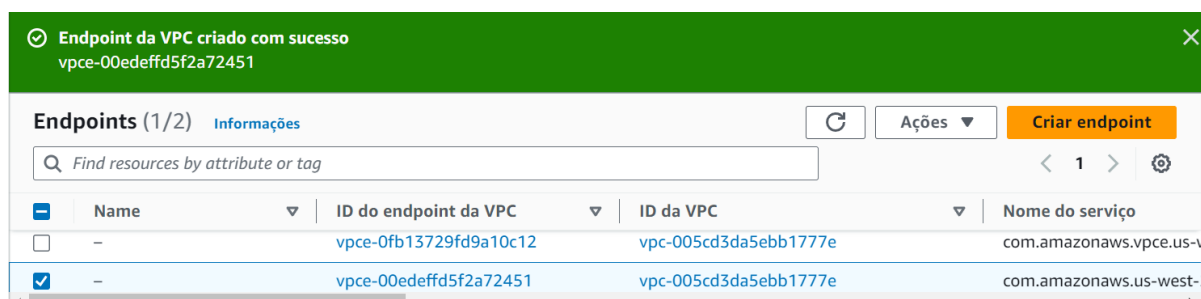


Figura 13: Endpoint criado

O segundo teste de conexão teve sucesso, como mostra a Figura 14.

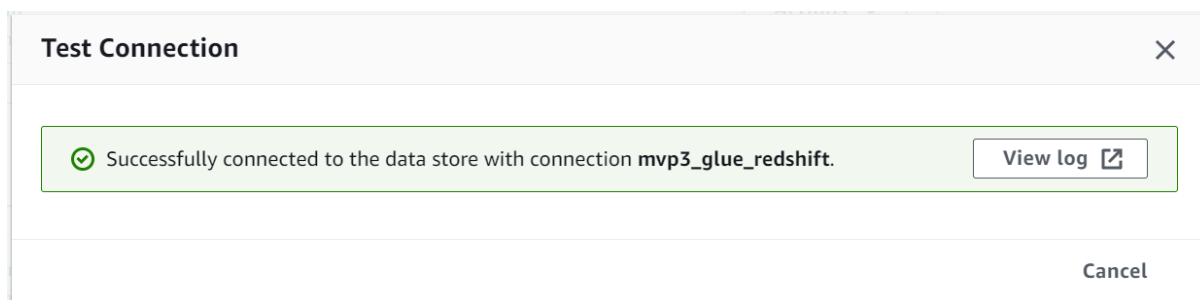


Figura 14: Segundo teste de conexão

Após a conexão ter funcionado, foi criada uma tabela com as colunas já anteriormente mencionadas (Figura 15), que posteriormente foi selecionada no job, finalizando suas configurações (Figura 16).

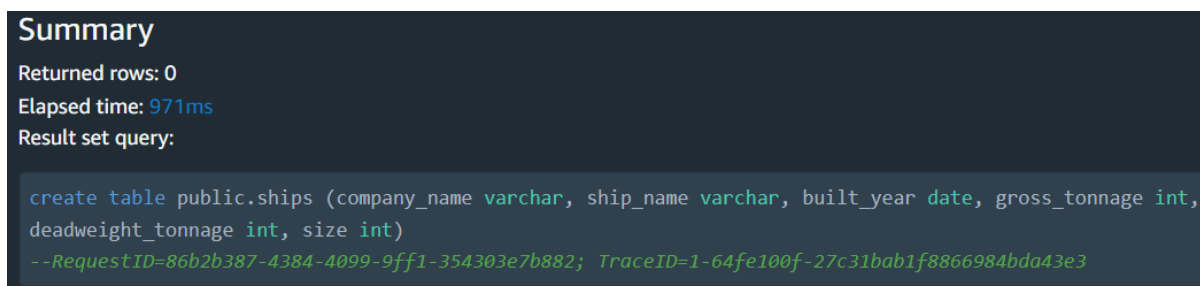


Figura 15: Criação da tabela

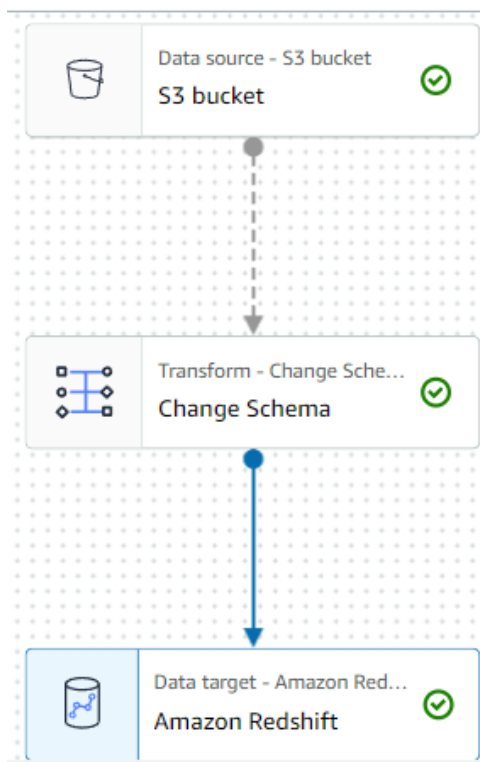
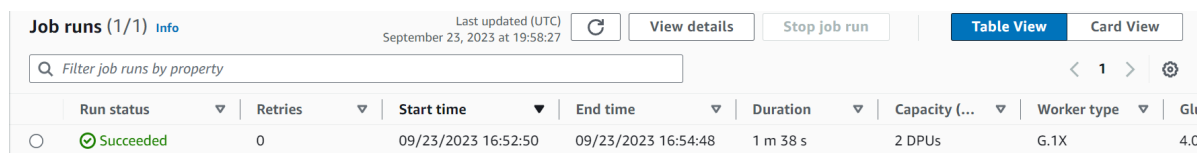


Figura 16: Configurações finalizadas

Então, com o job pronto, ele foi salvo e executado, conforme Figura 17.



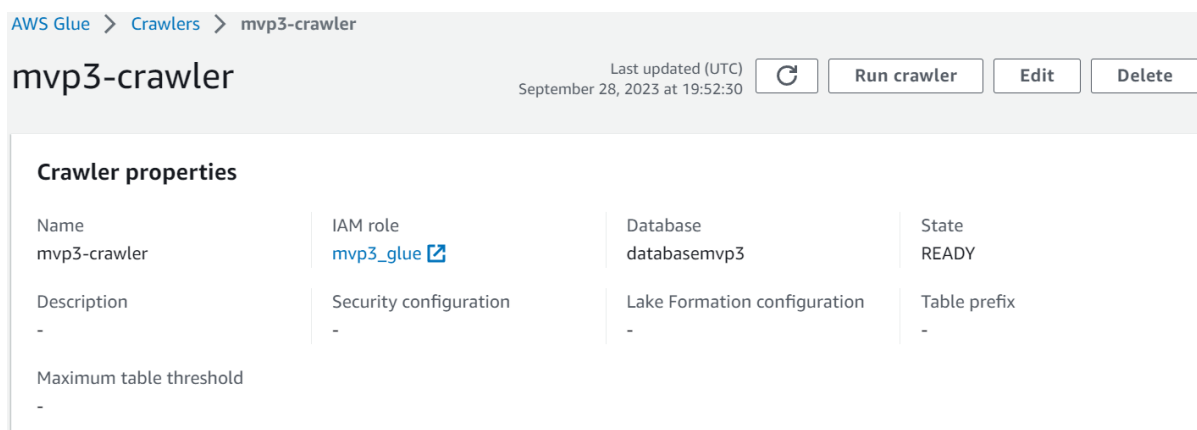
The screenshot shows the 'Job runs' section for a crawler named 'mvp3-crawler'. It displays a single job run that has 'Succeeded'. The table includes columns for Run status, Retries, Start time, End time, Duration, Capacity, Worker type, and Glue version.

Run status	Retries	Start time	End time	Duration	Capacity (...)	Worker type	Glue
✓ Succeeded	0	09/23/2023 16:52:50	09/23/2023 16:54:48	1 m 38 s	2 DPU's	G.1X	4.0

Figura 17: Job salvo e executado.

Catálogo de dados

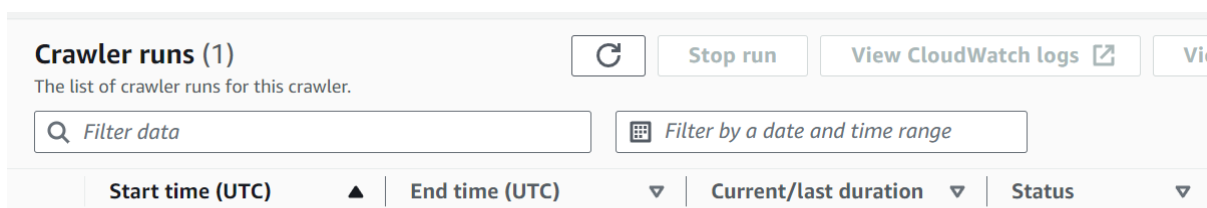
Para a catalogação dos dados, foi utilizado o *Data Catalog*, dentro do AWS Glue. Primeiramente, foi necessário a criação de um *crawler* com *Database* nomeado “*databasemvp3*” (Figura 18), para então ser feita a sua execução (Figura 19).



The screenshot shows the configuration page for the 'mvp3-crawler'. It includes fields for Name, IAM role, Database, State, Description, Security configuration, Lake Formation configuration, Table prefix, and Maximum table threshold.

Crawler properties			
Name	IAM role	Database	State
mvp3-crawler	mvp3_glue	databasemvp3	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			

Figura 18: Criação do *crawler*



The screenshot shows the 'Crawler runs' section for the 'mvp3-crawler'. It displays a single crawler run that has 'Completed'. The table includes columns for Start time (UTC), End time (UTC), Current/last duration, and Status.

Start time (UTC)	End time (UTC)	Current/last duration	Status
September 28, 2023 at 19:52	September 28, 2023 at 19:55	02 min 26 s	✓ Completed

Figura 19: Execução do *crawler*

A partir do *Database*, a tabela de dados dos navios foi acessada e comentários foram adicionados, a fim de explicar o significado de cada coluna e fornecer informações úteis para sua utilização, conforme Figura 20.

Schema (6)					Edit schema as JSON	Edit schema
View and manage the table schema.						
<input type="text" value="Filter schemas"/>					< 1 >	
#	Column name	Data type	Partitio...	Comment		
1	company_name	string	-	Nome da empresa de navios		
2	ship_name	string	-	Nome do navio		
3	built_year	date	-	Ano em que o navio foi produzido.		
4	gross_tonnage	bigint	-	Medida do volume interno total do navio, incluindo espa...		
5	deadweight_tonnage	bigint	-	Peso total que o navio pode carregar com segurança, inc...		
6	size	int	-	Dimensões do navio.		

Figura 20: Comentários da tabela no *Data Catalog*

Análise

Nesta seção serão abordados os temas de qualidade de dados e resolução dos problemas objetivos.

Qualidade dos dados

A qualidade dos dados foi analisada a fim de saber se existe algum problema entre os dados que possa afetar na solução das questões propostas do objetivo deste trabalho. A primeira análise feita foi de valores NULL.

Para todas as colunas da tabela “ships”, não houve nenhum valor NULL, exceto a coluna “size”, com 4000 valores NULL, conforme a Figura 21.

```

SELECT COUNT(*) FROM ships WHERE company_name IS NULL;
SELECT COUNT(*) FROM ships WHERE ship_name IS NULL;
SELECT COUNT(*) FROM ships WHERE built_year IS NULL;
SELECT COUNT(*) FROM ships WHERE gross_tonnage IS NULL;
SELECT COUNT(*) FROM ships WHERE deadweight_tonnage IS NULL;
SELECT COUNT(*) FROM ships WHERE size IS NULL;

```

Result 3 (1)	Result 4 (1)	Result 5 (1)	Result 6 (1)
count			
4000			

Figura 21: Query para qualidade das colunas

Apesar da falta de valores válidos nesta última coluna, isso não afetará na solução das questões. Portanto, nenhum tratamento destes dados será necessário.

Além disso, para se certificar que as datas contêm valores coerentes e no formato adequado, foi feita mais uma *query*, indicando o formato desejado. A contagem igual a zero significa

que não foi encontrado nenhum valor que não atenda ao formato da data mencionada, conforme Figura 22.

```
SELECT COUNT(*)
FROM ships
WHERE
    TO_DATE(built_year, 'YYYY-MM-DD') IS NULL;
```

Result 1 (1)

count
0

Figura 22: Query para qualidade das datas

Portanto, a qualidade dos dados foi demonstrada como boa para a solução dos problemas propostos, e nenhuma alteração ou tratamento será necessário.

Solução do problema

Nesta seção, serão demonstradas as *queries* realizadas para que se obtenham as soluções para as questões elaboradas no início do trabalho.

- 1) Qual é a capacidade de carga do navio mais novo e do mais antigo?

Resposta: a capacidade do navio mais novo é de 300000 toneladas, e do mais antigo 240000 toneladas. Isso foi indicado a partir da query realizada conforme Figura 23.

```
1  SELECT
2      MIN(ship_name) AS ship_name_oldest,
3      MAX(ship_name) AS ship_name_newest,
4      MIN(deadweight_tonnage) AS capacity_oldest_ship,
5      MAX(deadweight_tonnage) AS capacity_newest_ship
6  FROM
7      ships
8  WHERE
9      built_year = (SELECT MIN(built_year) FROM ships)
10     OR
11     built_year = (SELECT MAX(built_year) FROM ships);
```

Result 1 (1)

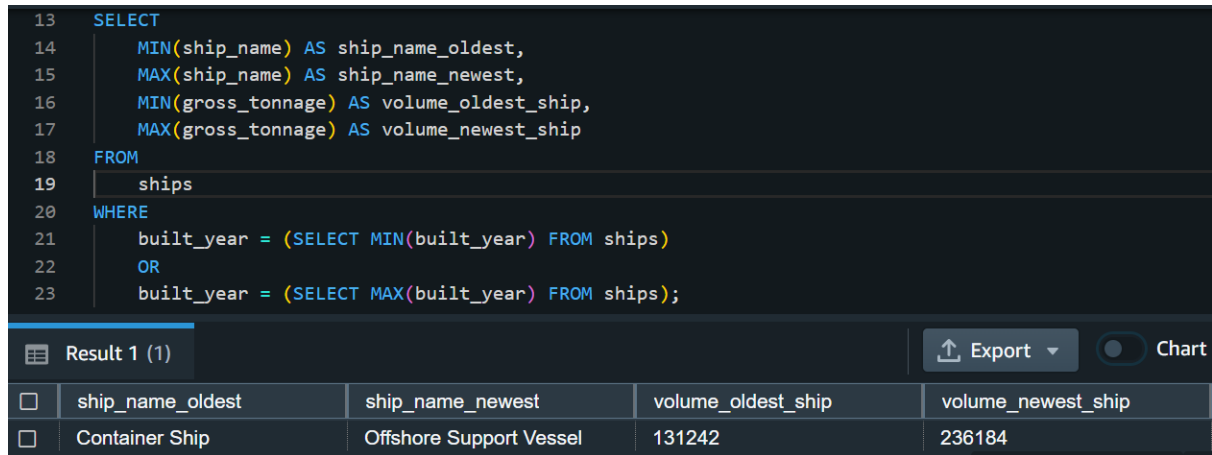
	ship_name_oldest	ship_name_newest	capacity_oldest_ship	capacity_newest_ship
<input type="checkbox"/>	Container Ship	Offshore Support Vessel	240000	300000

Export Chart

Figura 23: Query do navio mais novo e antigo

2) Qual é a medida do volume interno do navio mais novo e mais antigo?

Resposta: a medida do volume interno do navio mais novo é 236184 (Offshore Support Vessel), e do mais antigo é 131242 (Container Ship). Esta é uma medida usada para relações de volume dos navios e é adimensional. A Figura 24 ilustra a *query* realizada nesta questão.




```
13 SELECT
14     MIN(ship_name) AS ship_name_oldest,
15     MAX(ship_name) AS ship_name_newest,
16     MIN(gross_tonnage) AS volume_oldest_ship,
17     MAX(gross_tonnage) AS volume_newest_ship
18 FROM
19     ships
20 WHERE
21     built_year = (SELECT MIN(built_year) FROM ships)
22     OR
23     built_year = (SELECT MAX(built_year) FROM ships);
```

ship_name_oldest	ship_name_newest	volume_oldest_ship	volume_newest_ship
Container Ship	Offshore Support Vessel	131242	236184

Figura 24: *Query* de volume interno dos navios

3) É possível deduzir que o volume interno esteja diretamente relacionado com a capacidade total de carga?

Resposta: primeiramente, será calculado a correlação entre as duas variáveis, conforme a Figura 25.



```
25 SELECT
26     (
27         SUM(gross_tonnage::FLOAT * deadweight_tonnage::FLOAT) - COUNT(*)::FLOAT * AVG(gross_tonnage::FLOAT) * AVG(deadweight_tonnage::FLOAT)
28     ) /
29     (
30         Sqrt(
31             (SUM(POWER(gross_tonnage::FLOAT, 2)) - COUNT(*)::FLOAT * POWER(AVG(gross_tonnage::FLOAT), 2))
32             *
33             (SUM(POWER(deadweight_tonnage::FLOAT, 2)) - COUNT(*)::FLOAT * POWER(AVG(deadweight_tonnage::FLOAT), 2))
34         )
35     ) AS correlation
36 FROM
```

correlation
0.49138374876123164

Figura 25: Correlação entre volume interno e capacidade total de carga

A correlação não se mostrou muito alta. Além disso, por serem variáveis que representam conceitos diferentes, a correlação pode não ser a medida de comparação mais adequada, visto que estatisticamente pode não haver uma relação entre elas. Portanto, medidas de comparação serão adicionadas na análise, conforme Figura 26.



Figura 26: medidas de comparação entre variáveis

A seguir será analisado graficamente os valores mínimos, médios e máximos entre as duas variáveis (Figura 27).

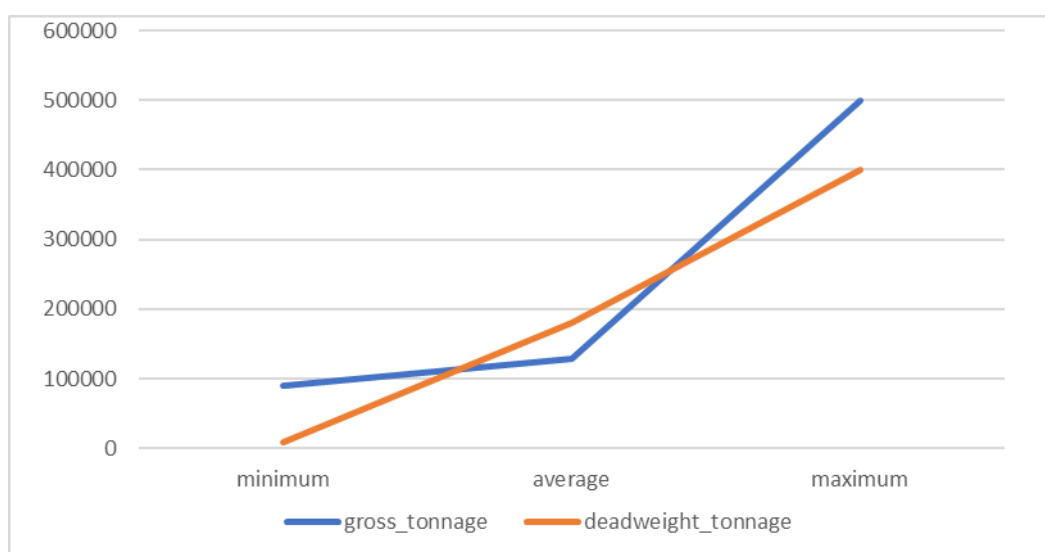


Figura 27: Gráfico de valores entre variáveis

De maneira geral, observa-se que quando a capacidade do navio aumenta (deadweight_tonnage), a tendência é que o volume interno (gross_tonnage) também aumente. Porém, a capacidade aumenta linearmente, enquanto há uma “quebra” para o volume interno, indicando que navios com volumes internos similares podem possuir capacidades de carga diferentes.

Além disso, no intervalo em que a capacidade é maior que o volume interno, é possível que seja uma boa relação de custo-benefício para os navios. Outro fator que confirmaria essa hipótese é que os navios que estão na média provavelmente são os mais utilizados (e mais escolhidos na hora de compra).

4) Quais foram os três navios mais produzidos?

Resposta: conforme Figura 28, os navios mais produzidos foram o Bulk Carrier, com 1360 unidades, o Container Ship, com 1020 unidades e o Crude Oil Tanker, com 720 unidades.

```

49 SELECT
50     ship_name,
51     COUNT(*) AS total_produzido
52 FROM
53     ships
54 GROUP BY
55     ship_name
56 ORDER BY
57     total_produzido DESC
58 LIMIT
59     3;

```

Result 1 (3)		
<input type="checkbox"/>	ship_name	total_produzido
<input type="checkbox"/>	Bulk Carrier	1360
<input type="checkbox"/>	Container Ship	1020
<input type="checkbox"/>	Crude Oil Tanker	720

Figura 28: Navios mais produzidos

5) Qual navio foi produzido em todos os anos?

Resposta: conforme a Figura 29, não houve nenhum navio que foi produzido todos os anos.

```

61 SELECT ship_name
62 FROM (
63     SELECT ship_name, COUNT(DISTINCT built_year) AS unique_years
64     FROM ships
65     GROUP BY ship_name
66 ) subquery
67 WHERE unique_years = (SELECT COUNT(DISTINCT built_year) FROM ships);
68

```

Result 1	
<input type="checkbox"/>	ship_name
No Rows To Show	

Figura 29: Navios produzidos todos os anos

Resposta: nenhum navio foi produzido em todos os anos disponíveis na tabela.

6) Qual navio foi produzido em apenas um ano?

Resposta: conforme a Figura 30, nenhum navio foi produzido em apenas um ano.

```
69  SELECT ship_name
70  FROM (
71      SELECT ship_name, COUNT(DISTINCT built_year) AS unique_years
72      FROM ships
73      GROUP BY ship_name
74  ) subquery
75  WHERE unique_years = 1;
76
77
```

Result 1

ship_name

No Rows To Show

Figura 30: Navios produzidos em apenas um ano

7) Por quais motivos os navios das questões anteriores foram produzidos todos os anos e apenas por um ano?

Resposta: como não há nenhum navio que tenha sido produzido todos os anos ou em apenas um ano, não é possível chegar a uma conclusão. Além disso, as informações contidas na tabela “ships” não evidenciam motivos para as produções dos navios em determinados anos.