

# Ejercicio 50 - Proyecto Cloud Analytics II

o Construcción de un ETL Avanzado que genere dos tablas adicionales a la de housing:

1) promedio de precio, tamaño, número de habitaciones por estado de amoblamiento:

## ETL JOB-01

The screenshot shows the AWS Glue console interface for a job named 'job-table-01'. The left sidebar contains navigation links for 'Getting started', 'ETL jobs', 'Data Catalog', and 'Data Integration and ETL'. The main area displays the 'Visual' tab of the workflow, showing a sequence of three nodes: 'Data source - Data Catalog AWS Glue Data Catalog', 'Transform - SQL Query SQL Query', and 'Data target - S3 bucket Amazon S3'. A message on the right indicates 'No node selected'.

## Data Source

The screenshot shows the AWS Glue console interface for a job named 'job-table-01'. The left sidebar contains navigation links for 'Getting started', 'ETL jobs', 'Data Catalog', and 'Data Integration and ETL'. The main area displays the 'Visual' tab of the workflow, showing a sequence of three nodes: 'Data source - Data Catalog AWS Glue Data Catalog', 'Transform - SQL Query SQL Query', and 'Data target - S3 bucket Amazon S3'. The 'Data source properties' panel is open, showing the 'Data source - Data Catalog' node selected. The panel includes fields for 'Name' (AWS Glue Data Catalog), 'Database' (module-50), and 'Table' (module\_49).

# Transform

The screenshot shows the AWS Glue console interface for configuring a job named 'job-table-01'. The job is set to run using a 'Transform - SQL Query' node. The data source is 'AWS Glue Data Catalog' and the data target is 'Amazon S3'. The right sidebar shows the 'Transform' tab with fields for Name, Node parents, Input sources, and SQL aliases. The SQL query is visible at the bottom.

En vez de usar la opción de “Aggregate” realicé una query en SQL:

SQL query

Enter a SQL statement to add to your job.


```
1 SELECT
2   furnishingstatus,
3   avg(price) as precio_medio,
4   avg(area) as area_media,
5   avg(bedrooms) as numerohabitaciones_medio
6 FROM myDataSource
7 group by 1
8 order by 2;
9
```

## Data Preview

Transform


Output schema


Data preview



Data preview (3) [Info](#)

Previewing 4 of 4 fields

 *Filter sample dataset*



furnishingstatus	precio_medio	area_media	numerohabitaciones_medio
unfurnished	4013831.4606741574	4707.595505617977	2.831460674157303
semi-furnished	4907524.22907489	5166.339207048458	3.0088105726872247
furnished	5495696	5688.1	3.0642857142857145

## Data Target

No se por cual motivo pero cuando utilice el formato csv la tabla final no era leida correctamente. En cambio, al hacerlo con JSON obtuve exactamente el mismo resultado del “data preview”. Por este motivo usé el formato JSON.

Data target properties - S3

Output schema

Data preview

Name

Amazon S3

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

SQL Query

SqlCode - Transform

Format

JSON

Compression Type

None

S3 Target Location

Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).

Data target properties - S3

Output schema

Data preview

☐ add new partitions

☒ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Database

Choose the database from the AWS Glue Data Catalog.

module-50

► Use runtime parameters

Table name

Enter a table name for the AWS Glue Data Catalog.

table01-furnishingstatus

Partition keys - optional

Add partition keys.

Add a partition key

# Amazon athena

aws

Services

Search

[Alt+S]

S3RDSVPCAWS Glue

Amazon Athena > Query editor

Editor

Recent queries

Saved queries

Settings

Workgroup primary

Data

Refresh

Back

Data source

AwsDataCatalog

Database

module-50

Tables and views

Create

Filter tables and views

Tables (3)

1

module\_49

table01-furnishingstatus

table02-parkingandstories

1

SELECT \* FROM "module-50"."table01-furnishingstatus" limit 30;

SQL

Ln 1, Col 62

Run again

Explain

Cancel

Clear

Create

Reuse query results

up to 60 minutes ago

aws

Services

Search

[Alt+S]

S3RDSVPCAWS Glue

table01-furnishingstatus

table02-parkingandstories

Views (0)

1

Run again

Explain

Cancel

Clear

Create

Reuse query results

up to 60 minutes ago

Query results

Query stats

Completed

Time in queue: 127 ms

Run time: 303 ms

Data scanned: 0.77 KB

Results (9)

Copy

Download results

Search rows

1

Settings

#	furnishingstatus	precio_medio	area_medio	numerohabitaciones_medio
2	furnished	5495696.0	5688.1	3.0642857142857145
1	semi-furnished	4907524.22907489	5166.339207048458	3.0088105726872247
3	unfurnished	4013831.4606741574	4707.595505617977	2.831460674157303
4				
5				
6				
7				

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

## 2) precio promedio de casa por stories y parking

### ETL JOB-02

The screenshot shows the AWS Glue console interface for a job named 'job-table-02'. The left sidebar contains navigation links for 'Getting started', 'ETL jobs', 'Data Catalog', and 'Data Integration and ETL'. The main area displays the workflow graph in 'Visual' mode. The workflow consists of three nodes connected sequentially: 'Data source - Data Catalog AWS Glue Data Catalog', 'Transform - SQL Query SQL Query', and 'Data target - S3 bucket Amazon S3'. The 'Visual' tab is selected, and the 'No node selected' message is displayed on the right. The top bar shows the job name, last modified date, and buttons for 'Actions', 'Save', and 'Run'.

### Data Source

The screenshot shows the AWS Glue console interface for the same job 'job-table-02'. The 'Data source properties - Data Catalog' tab is selected on the right, displaying the configuration for the 'Data source - Data Catalog AWS Glue Data Catalog' node. The configuration includes the Name, Database (module-50), and Table (module\_49). The workflow graph is visible in the background, showing the same three nodes as in the previous screenshot.

# Transform

The screenshot shows the AWS Glue console interface for configuring a job named 'job-table-02'. The job is configured with a 'Data source - Data Catalog AWS Glue Data Catalog', a 'Transform - SQL Query' node, and a 'Data target - S3 bucket Amazon S3'. The 'Transform' tab is selected, showing the SQL query configuration.

**Transform**

Name: SQL Query

Node parents: Choose one or more parent node

Input sources: AWS Glue Data Catalog

SQL aliases: myDataSource

SQL query

En vez de usar la opción de “Aggregate” realicé una query en SQL:

SQL query

Enter a SQL statement to add to your job.

```
1 SELECT
2   stories,
3   parking,
4   avg(price) as precio_medio
5 FROM myDataSource
6 group by 1,2
7 order by 3;
```

## Data Preview

Transform	Output schema	Data preview
Data preview (15) <a href="#">Info</a>		
Filter sample dataset		
stories	parking	precio_medio
1	0	3695366.1417322834
2	0	4079343.5114503815
1	1	4428355.555555556
1	3	4565166.666666667
1	2	5117571.428571428
2	1	5239104.838709678
3	0	5343722.222222222
3	1	5798916.666666667

## Data Target

No se por cual motivo pero cuando utilice el formato csv la tabla final no era leida correctamente. En cambio, al hacerlo con JSON obtuve exactamente el mismo resultado del “data preview”. Por este motivo usé el formato JSON.

Data target properties - S3

Output schema

Data preview

Name

Amazon S3

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

SQL Query

SqlCode - Transform

Format

JSON

Compression Type

None

S3 Target Location

Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).

Data target properties - S3

Output schema

Data preview

Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

☒ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Database

Choose the database from the AWS Glue Data Catalog.

module-50

► Use runtime parameters

Table name

Enter a table name for the AWS Glue Data Catalog.

table02-parkingandstories

Partition keys - optional

Add partition keys.

Add a partition key

# Amazon athena

aws

Services

Search

[Alt+S]

S3RDSVPCAWS Glue

Amazon Athena > Query editor

EditorRecent queriesSaved queriesSettings

Workgroupprimary

Data

Data source

AwsDataCatalog

Database

module-50

Tables and views

Create

Filter tables and views

Tables (3)

module\_49

table01-furnishingstatus

table02-parkingandstories

Query 2

Query 4

Query 5

1

SELECT \* FROM "module-50"."table02-parkingandstories" limit 20;

SQL

Ln 1, Col 63

Run again

Explain

Cancel

Clear

Create

Reuse query results

up to 60 minutes ago

aws

Services

Search

[Alt+S]

S3RDSVPCAWS Glue

table01-furnishingstatus

table02-parkingandstories

Views (0)

Run again

Explain

Cancel

Clear

Create

Reuse query results

up to 60 minutes ago

Query results

Query stats

Completed

Time in queue: 160 ms

Run time: 346 ms

Data scanned: 0.83 KB

Results (15)

Copy

Download results

Search rows

#	stories	parking	precio_medio
13	4	3	8096666.666666667
12	4	2	7952585.454545454
4	4	1	7315000.0
3	3	2	7109666.666666667
1	4	0	6334500.0
11	2	3	6241666.666666667
10	2	2	6093000.0

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences



# Soportes adicionales

Services

Search

[Alt+S]

S3

RDS

VPC

AWS Glue

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

Connections

Crawlers

Classifiers

Catalog settings

Data Integration and ETL

ETL jobs

Visual ETL

You can now create Apache Iceberg tables in the AWS Glue Data Catalog. To learn more, visit the documentation.

Create table

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (3)

Last updated (UTC)  
September 18, 2023 at 24:46:40

Refresh

Delete

Add tables using crawler

Add table

Filter tables

< 1 >

Settings

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	module_49	module-50	s3://module-49/	CSV	-	Table data	View data quality
<input type="checkbox"/>	table01-furnishingstatu	module-50	s3://final-result-1/	JSON	-	Table data	View data quality
<input type="checkbox"/>	table02-parkingandstor	module-50	s3://final-result-02/	JSON	-	Table data	View data quality

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Services

Search

[Alt+S]

S3

RDS

VPC

AWS Glue

AWS Glue Studio

Info

Create job

Info

Create

☒ Visual with a source and target

Start with a source, ApplyMapping transform, and target.

☐ Visual with a blank canvas

Author using an interactive visual interface.

☐ Spark script editor

Write or upload your own Spark code.

☐ Python Shell script editor

Write or upload your own Python shell script.

☐ Jupyter Notebook

Write your own code in a Jupyter Notebook for interactive development.

☐ Ray script editor

New

Write your own code to run on Ray.

Source

Target

Amazon S3

JSON, CSV, or Parquet files stored in S3.

→

Amazon S3

S3 bucket by specifying a bucket path as the data target.

Your jobs (2)

Info

Refresh

Actions

Run job

Filter jobs

< 1 >

Settings

<input type="checkbox"/>	Job name	Type	Last modified	AWS Glue version
<input type="checkbox"/>	job-table-02	Glue ETL	17/9/2023, 19:14:35	4.0
<input type="checkbox"/>	job-table-01	Glue ETL	17/9/2023, 19:09:39	4.0

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences