

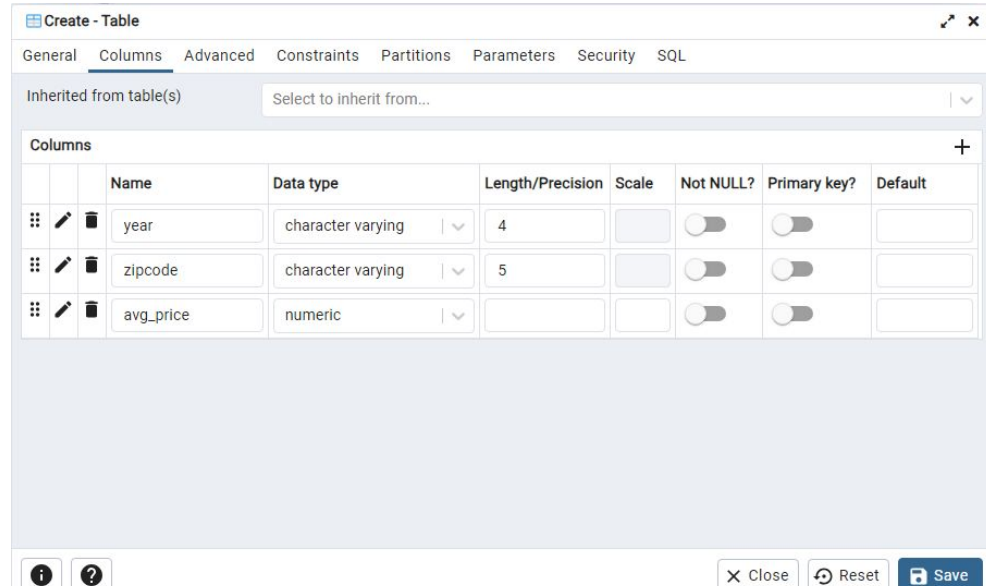
Módulo 46 - Cloud Computing II

Carlos Fernando Del Valle Reyes

Creación de la tabla destino en BD mediante PgAdmin

El nombre de la tabla es avg_price_per_zip

Los campos de Año y zipcode se generan como charvar puesto que el año no es la fecha completa y el zipcode una característica más que una medida



The screenshot shows the 'Create - Table' dialog box in PgAdmin, with the 'Columns' tab selected. The dialog has tabs for General, Columns, Advanced, Constraints, Partitions, Parameters, Security, and SQL. The 'Inherited from table(s)' field is empty, and the 'Select to inherit from...' dropdown is also empty. The 'Columns' section contains a table with three columns: 'year', 'zipcode', and 'avg_price'. Each column has a 'Data type' dropdown, a 'Length/Precision' field, a 'Scale' field, a 'Not NULL?' toggle, a 'Primary key?' toggle, and a 'Default' field. The 'year' and 'zipcode' columns are set to 'character varying' with lengths of 4 and 5 respectively. The 'avg_price' column is set to 'numeric'. All 'Not NULL?' and 'Primary key?' toggles are turned off.

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
⋮ ✎ 🗑	year	character varying v	4		<input type="checkbox"/>	<input type="checkbox"/>	
⋮ ✎ 🗑	zipcode	character varying v	5		<input type="checkbox"/>	<input type="checkbox"/>	
⋮ ✎ 🗑	avg_price	numeric v			<input type="checkbox"/>	<input type="checkbox"/>	

At the bottom of the dialog, there are buttons for 'Close', 'Reset', and 'Save'.

Creación del crawler para la tabla creada en PgAdmin

Add data source ✕

Data source
Choose the source of data to be crawled.

JDBC ▼

Connection
Select a connection to access the data sources below.

final_rds_postgres13 ▼ ↻

Clear selection

Add new connection 🔗

Include path

postgres/public/avg_price_per_zip

You can substitute the percent (%) character for a schema or table. For databases that support schemas, enter MyDatabase/MySchema/% to match all tables in MySchema within MyDatabase. Oracle Database and MySQL don't support schema in the path; instead, enter MyDatabase/%. For Oracle database without SSL, MyDatabase can be either the system identifier (SID) or the service name (SERVICE_NAME). For Oracle database with SSL, MyDatabase must be the service name (SERVICE_NAME).

Additional metadata - optional

Select additional metadata properties for the crawler to crawl.

☐ Exclude files matching pattern

Cancel

Add a JDBC data source

Creación de tabla destino en Glue

[AWS Glue](#) > [Databases](#) > Add database

Create a database
Create a database in the AWS Glue Data Catalog.

Database details

Name

db-crawler-avgpriceperzip

Database name is required, in lowercase characters, and no longer than 255 characters.

Location - optional
Set the URI location for use by clients of the Data Catalog.

Description - optional

Enter text

Descriptions can be up to 2048 characters long.

Cancel

Create database

Creación del crawler para la tabla creada en PgAdmin

Detalles del crawler

[AWS Glue](#) > [Crawlers](#) > Add crawler

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

Review and create

Step 1: Set crawler properties Edit

Set crawler properties

Name	Description	Tags
crawler-rds-avgprice	-	-

Step 2: Choose data sources and classifiers Edit

Data sources (1) [Info](#)
The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
JDBC	postgres/public/avg_price_per_zip	-

Step 3: Configure security settings Edit

Configure security settings

IAM role	Security configuration	Lake Formation configuration
AWSGlueServiceRole-climasp	-	-

Step 4: Set output and scheduling Edit

Set output and scheduling

Database	Table prefix - optional	Schedule
db-crawler-avgpriceperzip	-	On demand

Cancel Previous Create crawler

Creación del crawler para la tabla creada en PgAdmin

Corrida exitosa del crawler

[AWS Glue](#) > [Crawlers](#) > crawler-rds-avgprice

crawler-rds-avgprice

Last updated (UTC)
February 23, 2024 at 03:04:05

Run crawler

Edit

Delete

Crawler properties

Name	IAM role	Database	State
crawler-rds-avgprice	AWSGlueServiceRole-climasp	db-crawler-avgpriceperzip	READY
Description	Security configuration	Table prefix	
-	-	-	

▶ Advanced settings

[Crawler runs](#) | [Schedule](#) | [Data sources](#) | [Classifiers](#) | [Tags](#)

Crawler runs (1)

The list of crawler runs for this crawler.

Stop run

[View CloudWatch logs](#)

[View run details](#)

Filter data

Filter by a date and time range

< 1 >

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
February 23, 2024 at 03:04:10	February 23, 2024 at 03:08:47	04 min 37 s	Completed	-	-

Creación del Job de Glue

Fuente de datos - Glue Catalog que ya se había cargado con el crawler de S3 de la lección en video dado que el archivo `kc_house_data.csv` ya se había cargado al bucket.

The screenshot displays the AWS Glue console interface for configuring a job named 'job-avgprice_per_year_zipcode'. The top navigation bar includes tabs for Visual, Script, Job details, Runs, Data quality - updated, Schedules, and Version Control. The main workspace shows a 'Data source - Data Catalog' widget with 'AWS Glue Data Catalog' selected. The right sidebar, titled 'Data source properties - Data Catalog', contains fields for Name (AWS Glue Data Catalog), Database (crawldb-climasp), and Table (kc_house_data.csv). The bottom section, 'Data preview', shows the 'IAM role' dropdown set to 'AWSGlueServiceRole-climasp'. A 'Start session' button is visible at the bottom right. A notification at the top right states 'Job has not been saved' and 'Try new UI'. A bottom right alert indicates 'Unsaved job found' and offers a 'Restore' option.

Creación del Job de Glue

Bloque de transformación de datos del archivo csv mediante código SQL de group by. También se cambian los nombres de las columnas para que encaje con la tabla destino

The screenshot displays the AWS Glue console interface for a job named `job-avgprice_per_year_zipcode`. The job is in a 'Successfully updated' state. The main workspace shows a workflow diagram with two nodes: 'Data source - Data Catalog' (AWS Glue Data Catalog) and 'Transform - SQL Query' (SQL Query - group b...). The 'Transform' node is selected, and its configuration is shown on the right. The 'Name' field is 'SQL Query - group by year and zipcode'. The 'Node parents' dropdown is set to 'Choose one or more parent node'. The 'Input sources' section shows 'AWS Glue Data Catalog' as the source and 'myDataSource' as the SQL alias. The 'SQL query' field contains the following SQL statement:

```
1 select yr_built as year,
2       zipcode,
3       avg(price) as avg_price
4 from myDataSource
5 group by yr_built, zipcode;
```

The 'Data preview' tab is active, showing a table with 200 rows. The columns are 'year', 'zipcode', and 'avg_price'. The data is filtered to show the first 5 rows:

year	zipcode	avg_price
1955	98178	305216.92307692306
1987	98074	603685.1
2001	98053	709019.4444444445
1995	98003	286250
2003	98038	361705.45454545453

Creación del Job de Glue

Configuración del destino del Job como la tabla de la base de datos de Postgresql que se configuró en el crawler

The screenshot displays the AWS Glue console interface for a job named "job-avgprice_per_year_zipcode". At the top, a green banner indicates the job was "Successfully updated". Below this, the job's details are shown, including a status of "Data quality - updated" and a "Last modified" timestamp of "2/24/2024, 5:59:29 PM". The job is configured with three nodes in a sequential flow:

- Data source - Data Catalog:** AWS Glue Data Catalog.
- Transform - SQL Query:** SQL Query - group b...
- Data target - PostgreSQL:** PostgreSQL.

The right-hand panel shows the "Data target properties - PostgreSQL" configuration. The "Name" is set to "PostgreSQL". The "Node parents" section shows a dropdown menu with "SQL Query - group by year and zipcode" selected. The "Database" is set to "db-crawler-avgpriceperzip", and the "Table" is set to "postgres_public_avg_price_per_zip". Both the database and table names are highlighted in yellow. The "Use runtime parameters" section is expanded, showing "Use runtime parameters" as an option.

Demostración del Job

Query de la tabla vacía antes de correr el Job



The screenshot displays a database query interface. At the top, there are tabs for 'Query' and 'Query History', and a 'Scratch Pad' tab on the right. The 'Query' tab is active, showing a single query:

```
1 select * from avg_price_per_zip;
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with three columns: 'year', 'zipcode', and 'avg_price'. The table is empty, indicating that the query returned no results.

year	zipcode	avg_price
------	---------	-----------

Demostración el Job

Corrida exitosa del Job



job-avgprice_per_year_zipcode

Visual | Script | Job details | **Runs** | Data quality - *updated* | Schedules | Version Control

Job runs (1/1) [Info](#)

Filter job runs by property

	Run status	Retries	Start time (UTC)	End time (UTC)
	Succeeded	0	2024/02/25 00:03:43	2024/02/25 00:06:07

Query despues del Job

postgres/postgres@EBAC Postgres 13

Query Query History

```
1 select * from avg_price_per_zip;
```

Data Output Messages Notifications

	year character varying (4)	zipcode character varying (5)	avg_price numeric
1	1942	98115	516269.230769231
2	1929	98117	591546.620689655
3	2000	98074	984625
4	1957	98040	891800
5	1965	98003	307593.333333333
6	1905	98122	845160
7	1984	98074	542526.315789474
8	2010	98059	462055.555555556
9	1983	98031	313058.333333333