



Data Engineering Cohort 1

Module 5

Assignment 5.2

Glue Jobs in AWS

Member #1

Name: Muhammad Humza

Roll Number: 2211-018-KHI-DEG

Member #2

Name: Muhammad Ovais

Roll Number: 2211-023-KHI-DEG

Task

Using the earnings CSV as a base, prepare a new data file with employees' office locations. Make sure there are 5-6 distinct locations that are shared between employees.

Create a Glue job that aggregates the data based on the office location to calculate average salaries and raise percentages for these locations.

Solution

Step 1: Creating a bucket for Glue Job

Here we've created a bucket in AWS-S3 with our name a

Account snapshot [View Storage Lens dashboard](#)

Buckets (3) [Info](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

	Name	AWS Region	Access	Creation date
<input type="radio"/>	aws-glue-assets-753805937683-us-east-1	US East (N. Virginia) us-east-1	Bucket and objects not public	January 4, 2023, 13:57:54 (UTC+05:00)
<input type="radio"/>	ovais-module5-day4	US East (N. Virginia) us-east-1	Bucket and objects not public	January 5, 2023, 12:29:19 (UTC+05:00)
<input type="radio"/>	ovaissaleem-glue-data	US East (N. Virginia) us-east-1	Bucket and objects not public	January 4, 2023, 11:16:06 (UTC+05:00)

Step 2: Creating directories in bucket

After creating the bucket, we added three directories in it according to the Readme file.

Amazon S3 > Buckets > ovaissaleem-glue-data

ovaissaleem-glue-data [Info](#)

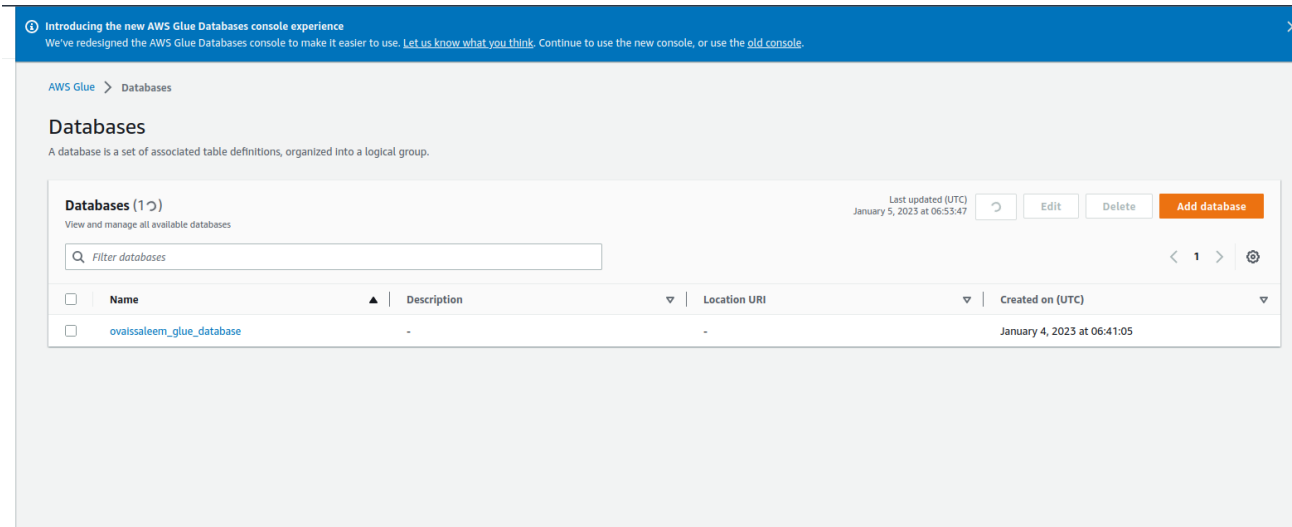
[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (3) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	glue_data/	Folder	-	-	-
<input type="checkbox"/>	input_data/	Folder	-	-	-
<input type="checkbox"/>	output_data/	Folder	-	-	-

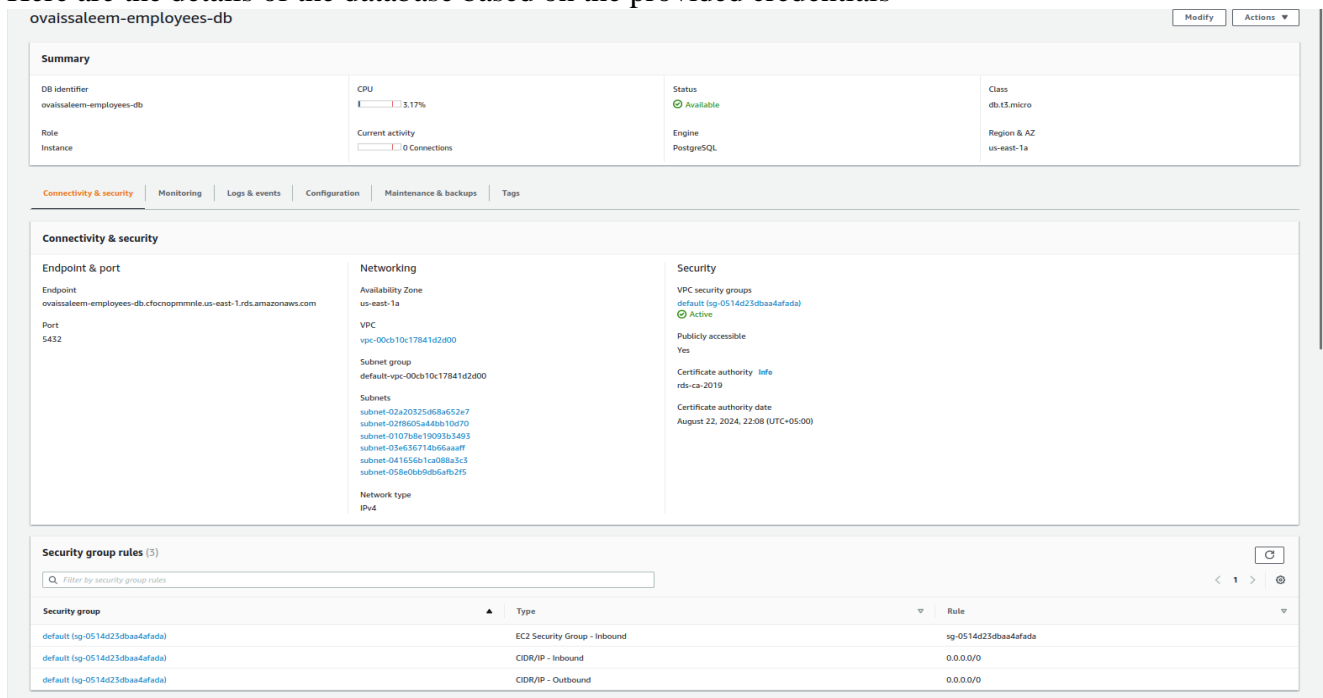
Step 3.1: Creating database in RDS

Here we create a database in RDS.



Step 3.2: Database Credentials in RDS

Here are the details of the database based on the provided credentials



Step 4: Creating Security Groups

Here we create the security group with rules for our database

EC2 > Security Groups > sg-0514d23dbaa4afada - default

sg-0514d23dbaa4afada - default Actions ▾

Details

Security group name default	Security group ID sg-0514d23dbaa4afada	Description default VPC security group	VPC ID vpc-00cb10c17841d2d00 ↗
Owner 753805937603	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

📘 You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer ✕

Inbound rules (2) 🔄 Manage tags Edit inbound rules

<input type="checkbox"/>	Name ▾	Security group rule... ▾	IP version ▾	Type ▾	Protocol ▾	Port range ▾	Source ▾	Description
<input type="checkbox"/>	-	sgr-0c6c3caa318f67f9b	IPv4	PostgreSQL	TCP	5432	0.0.0.0/0	Rule for ex
<input type="checkbox"/>	-	sgr-012fae233008d2900	-	All TCP	TCP	0 - 65535	sg-0514d23dbaa4afad...	Rule for Gli

Step 5: Creating Endpoint

Here we created the endpoint for our database

Endpoints (1) Info 🔄 Actions ▾ Create endpoint

< **1** > 🔍

<input type="checkbox"/>	Name ▾	VPC endpoint ID ▾	VPC ID ▾	Service name ▾	Endpoint type ▾	Status
<input type="checkbox"/>	-	vpce-0bd61c355deeb60b1	vpc-00cb10c17841d2d00	com.amazonaws.us-east-1.s3	Gateway	🟢 Available

Step 6: Creating Role

Here we create a role from IAMRole in AWS that will be used in crawler.

Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. [Let us know what you think.](#)

IAM > Roles > ovaissaleem-glue-role

ovaissaleem-glue-role

Allows Glue to call AWS services on your behalf.

Delete

Edit

Summary

Creation date
January 04, 2023, 11:30 (UTC+05:00)

Last activity
17 hours ago

ARN
arn:aws:iam::753805937683:role/ovaissaleem-glue-role

Maximum session duration
1 hour

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (2) [Info](#)

You can attach up to 10 managed policies.

☐

Policy name

Type

Description

☐

AmazonS3FullAccess

AWS managed

Provides full access to all buckets via the AWS Management Console.

☐

AWSGlueServiceRole

AWS managed

Policy for AWS Glue service role which allows access to related services including EC2, S3, and Cloudwatch Logs

Step 7.1: Creating the crawler

Here we created the crawler with the provided credentials

AWS Glue > Crawlers > View crawler details

ovaissaleem_rds_employees_crawler

Run crawler

Edit

Delete

Page last updated: January 5, 2023 at 11:39:38 (UTC)

Crawler properties

Name
ovaissaleem_rds_employees_crawler

Description
-

IAM role
[ovaissaleem-glue-role](#)

Security configuration
-

Database
ovaissaleem_glue_database

Table prefix
ovaissaleem_

State
READY

Advanced settings

Crawler runs

Schedule

Data sources

Classifiers

Tags

Crawler runs (3)

Stop run

View CloudWatch logs

View run details

The list of crawler runs for this crawler.

< 1 >

	Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
<input type="radio"/>	January 4, 2023 at 12:27:03	January 4, 2023 at 12:29:27	02 min 24 s	Completed	0.244	-
<input type="radio"/>	January 4, 2023 at 10:42:59	January 4, 2023 at 10:46:45	03 min 46 s	Completed	0.207	-
<input type="radio"/>	January 4, 2023 at 06:48:56	January 4, 2023 at 06:51:44	02 min 47 s	Completed	0.148	1 table change, 0 partition changes

Step 7.2: Running the crawler

Here we can see that the crawler runs successfully for the glue job

Introducing the new AWS Glue Crawlers console experience

We've redesigned the AWS Glue Crawlers console to make it easier to use. [Let us know what you think](#). Continue to use the new console, or use the [old console](#). Including new features: S3 event crawler, Crawler history and Cross-account crawlers (preview).

AWS Glue > Crawlers

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (2) Info

View and manage all available crawlers.

Last updated (UTC)
January 5, 2023 at 06:53:36

Refresh

Action

Run

Create crawler

Filter crawlers

< 1 > ⚙

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table cha..
<input type="checkbox"/>	ovaissaleem_rds_employees_crawler	Ready		Succeeded	January 4, 2023 at 12:2...	View log	-
<input type="checkbox"/>	ovaissaleem_s3_earnings_crawler	Ready		Succeeded	January 4, 2023 at 12:2...	View log	1 created

Step 8: Checking the tables

Here we check that the tables are fetched properly

Introducing the new AWS Glue Tables console experience

We've redesigned the AWS Glue Tables console to make it easier to use. [Let us know what you think](#). Continue to use the new console, or use the [old console](#).

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 (7)

View and manage all available tables.

Last updated (UTC)
January 5, 2023 at 06:53:49

Refresh

Delete

Data quality

Add tables using crawler

Add table

Filter tables

< 1 > ⚙

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data
<input type="checkbox"/>	ovaissaleem_average_salaries	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/output_d...	parquet	-	Table data
<input type="checkbox"/>	ovaissaleem_date_2023_01_04	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/input_dat	csv	-	Table data
<input type="checkbox"/>	ovaissaleem_date_2023_01_05	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/input_dat	csv	-	Table data
<input type="checkbox"/>	ovaissaleem_earnings	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/input_dat	csv	1672829032948	Table data
<input type="checkbox"/>	ovaissaleem_employees_db_public_en	ovaissaleem_glue_database	employees_db.public.employees	postgresql	-	-
<input type="checkbox"/>	ovaissaleem_employees_earnings	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/output_d...	parquet	-	Table data
<input type="checkbox"/>	ovaissaleem_location	ovaissaleem_glue_database	s3://ovaissaleem-glue-data/input_dat	csv	-	Table data

Step 9: Checking the table earning 1

Here we check the values of table earning 1 in query editor

The screenshot displays the AWS Glue console interface. On the left, a workflow diagram shows a job named 'earnings' with two parallel paths. The first path starts with a 'Data source - Data Catalog' node pointing to 'earning_1', followed by a 'Transform - Map' node, an 'Aggregate' node, and a 'Transform - Map' node. The second path starts with a 'Data source - Data Catalog' node pointing to 'earning_2', followed by a 'Transform - Map' node, an 'Aggregate' node, and a 'Transform - Map' node. The 'Transform - Map' nodes are connected to a central 'Transform - Map' node. On the right, the 'Data source properties - Data Catalog' tab is selected. The 'Database' dropdown is set to 'ovaissaleem_glue_database'. The 'Table' dropdown is set to 'ovaissaleem_date_2023_01_04'. The 'Use runtime parameters' section is expanded.

Step 10: Checking the table earning 2

Here we check the values of table earning 2 in query editor

The screenshot displays the AWS Glue console interface. On the left, a workflow diagram shows a job named 'earnings' with two parallel paths. The first path starts with a 'Data source - Data Catalog' node pointing to 'earning_1', followed by a 'Transform - Map' node, an 'Aggregate' node, and a 'Transform - Map' node. The second path starts with a 'Data source - Data Catalog' node pointing to 'earning_2', followed by a 'Transform - Map' node, an 'Aggregate' node, and a 'Transform - Map' node. The 'Transform - Map' nodes are connected to a central 'Transform - Map' node. On the right, the 'Data source properties - Data Catalog' tab is selected. The 'Database' dropdown is set to 'ovaissaleem_glue_database'. The 'Table' dropdown is set to 'ovaissaleem_date_2023_01_05'. The 'Use runtime parameters' section is expanded.

Step 11: Applying Joins

Here multiple joins were applied.

The screenshot displays the AWS Glue console interface. On the left, a workflow diagram shows a sequence of nodes: 'Data source - Data Catalog Relational DB', 'Transform - ApplyMapping', 'Transform - Join', 'Transform - Aggregate', and 'Transform - ApplyMapping'. The 'Transform - Join' node is highlighted. On the right, the 'Node properties' panel for the 'Join' node is shown. The 'Join type' is set to 'Inner join'. The 'Join conditions' section shows a condition: 'right_emp_id = emp_id'. The 'Renamed keys for Join' section shows 'earning_1' and 'emp_id'.

Step 12: Applying Joins and Aggregation

Here multiple joins and aggregation for average were applied.

The screenshot displays the AWS Glue console interface. On the left, a workflow diagram shows a sequence of nodes: 'Data source - Data Catalog Relational DB', 'Transform - ApplyMapping', 'Transform - Join', 'Transform - Aggregate', and 'Transform - ApplyMapping'. The 'Transform - Aggregate' node is highlighted. On the right, the 'Node properties' panel for the 'Aggregate' node is shown. The 'Fields to group by - optional' section shows 'right_location'. The 'Field to aggregate' section shows '2nd_earnings'. The 'Aggregation function' is set to 'avg'.

Step 13: Aggregation for earning 1

Here aggregation for earning 1 was applied

Job details

Runs

Data quality

Schedules

Version Control

Target

Undo

Redo

Remove

🔍

🔍

🔍

Node properties

Transform

Output schema

Data preview

🔍

Data preview (5) [Info](#)

Previewing 2 of 2 fields

🔍

right_location	avg(earnings)
B	6286.75
C	5576.95
A	5926.05
D	5889.7
E	5599.2

Step 14: Aggregation for earning 2

Here aggregation for earning 2 was applied

Job details
Runs
Data quality
Schedules
Version Control

Target

Undo Redo Remove

Node properties Transform Output schema **Data preview**

Data preview (5) [Info](#)

Filter sample dataset

right_location	avg(2nd_earnings)
B	5887
C	5813.65
A	6509.9
D	5380.45
E	5407.6

Step 15: Combined Aggregation

Here combined aggregation of both earnings is shown.

Job detailsRunsData qualitySchedulesVersion Control

TargetUndoRedoRemove

```
graph TD; A[Data source - Data Catalog] --> B[Transform - Aggregating  
Renamed keys for join]; B --> C[Join]; C --> D[Transform - Aggregating  
Aggregate]; C --> E[Transform - Aggregating  
Aggregate]; D --> F[Transform - Aggregating  
Renamed keys for join]; E --> G[Transform - Aggregating  
Renamed keys for join]; F --> H[Table  
ovaissaleem_date_2023_01_04]; G --> H;
```

Node propertiesData source properties - Data CatalogOutput schemaData preview

Database

Choose a database.

ovaissaleem_glue_database

Use runtime parameters

Table

ovaissaleem_date_2023_01_04

Use runtime parameters

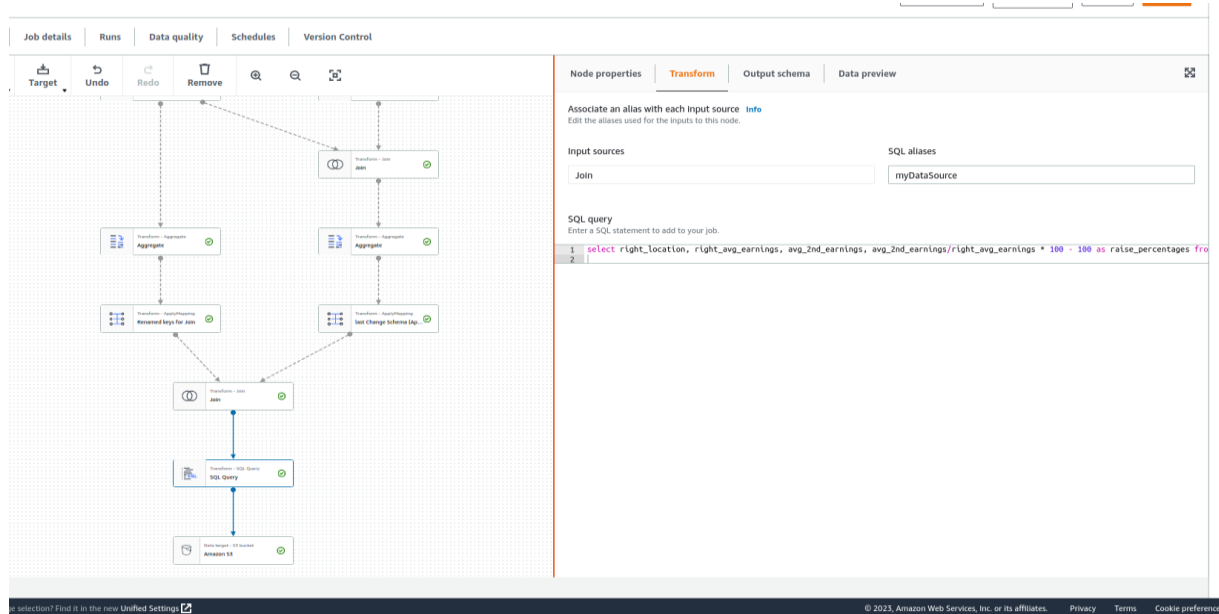
selection? Find it in the new Unified Settings

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

Step 16: Calculating the earning raise

Here we calculated the raise in earnings

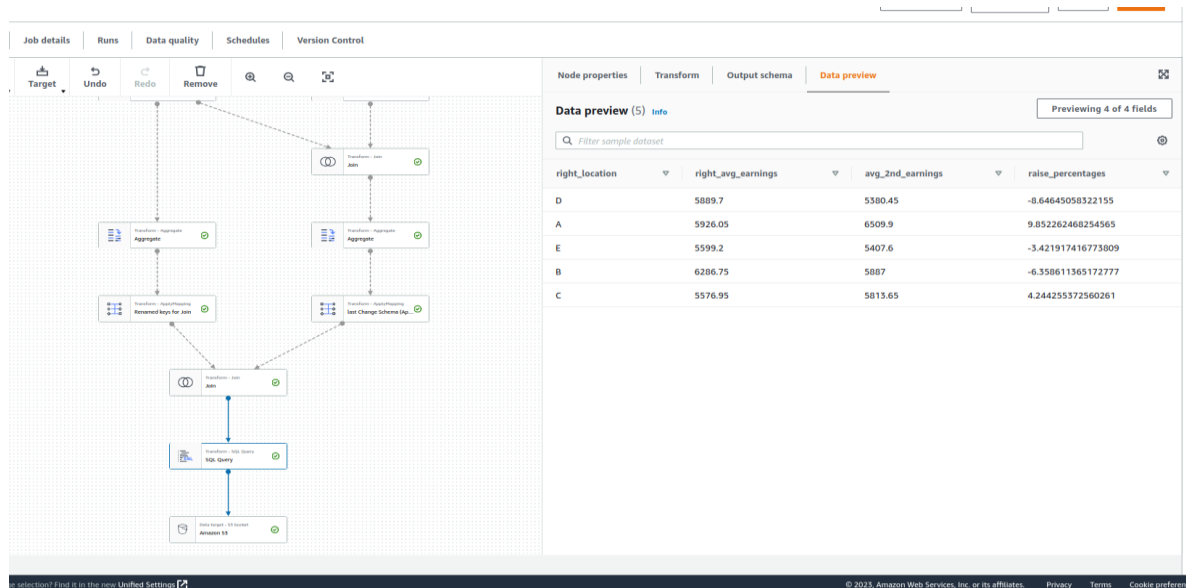
Query: select right_location, right_avg_earnings, avg_2nd_earnings, avg_2nd_earnings/right_avg_earnings * 100 - 100 as raise_percentages from myDataSource



The screenshot shows the AWS Glue console interface. The 'Transform' tab is selected, displaying the SQL query: `select right_location, right_avg_earnings, avg_2nd_earnings, avg_2nd_earnings/right_avg_earnings * 100 - 100 as raise_percentages from myDataSource`. The 'Data preview' tab is also visible, showing a table with 5 rows and 4 columns: right_location, right_avg_earnings, avg_2nd_earnings, and raise_percentages. The job is configured with a 'Transform' node and a 'Data preview' node.

Step 17: The raise between earnings

Here we can see the raise in earnings between earning 1 and earning 2



The screenshot shows the AWS Glue console interface. The 'Data preview' tab is selected, displaying a table with 5 rows and 4 columns: right_location, right_avg_earnings, avg_2nd_earnings, and raise_percentages. The data is as follows:

right_location	right_avg_earnings	avg_2nd_earnings	raise_percentages
D	5889.7	5380.45	-8.64645058322155
A	5926.05	6509.9	9.852262468254565
E	5599.2	5407.6	-3.421917416773809
B	6286.75	5887	-6.358611365172777
C	5576.95	5813.65	4.244255372560261

Step 18: The raise between earnings

Here we can see the raise in earnings between earning 1 and earning 2 that increased

Query: select right_location, right_avg_earnings, avg_2nd_earnings, avg_2nd_earnings/right_avg_earnings * 100 - 100 as raise_percentages from myDataSource where (avg_2nd_earnings/right_avg_earnings * 100 - 100) > 0

The screenshot displays a data pipeline tool interface. On the left, a workflow diagram shows a sequence of nodes: a 'Transform - Aggregate' node, a 'Transform - Map' node, a 'Transform - SQL Query' node, and a 'Data Export - S3 bucket' node. The 'Transform - SQL Query' node is highlighted. On the right, the 'Data preview' tab is active, showing a table with 4 columns: 'right_location', 'right_avg_earnings', 'avg_2nd_earnings', and 'raise_percentages'. The table contains two rows of data, labeled 'A' and 'C'.

right_location	right_avg_earnings	avg_2nd_earnings	raise_percentages
A	5928.05	6509.9	9.852262468254565
C	5576.95	5813.65	4.244255372560261

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences