TASK and ASSIGNMENT 5.2

on

AWS ETL

Submitted by:

Haseebullah Shaikh (2303.KHI.DEG.015)

and

Faiza Gulzar Ahmed (2303.khi.deg.001)

Dated: 16th May 2023

Solution:

Objective:

To explain in each step:

- Why we need to take this step?
- What is the service's purpose?

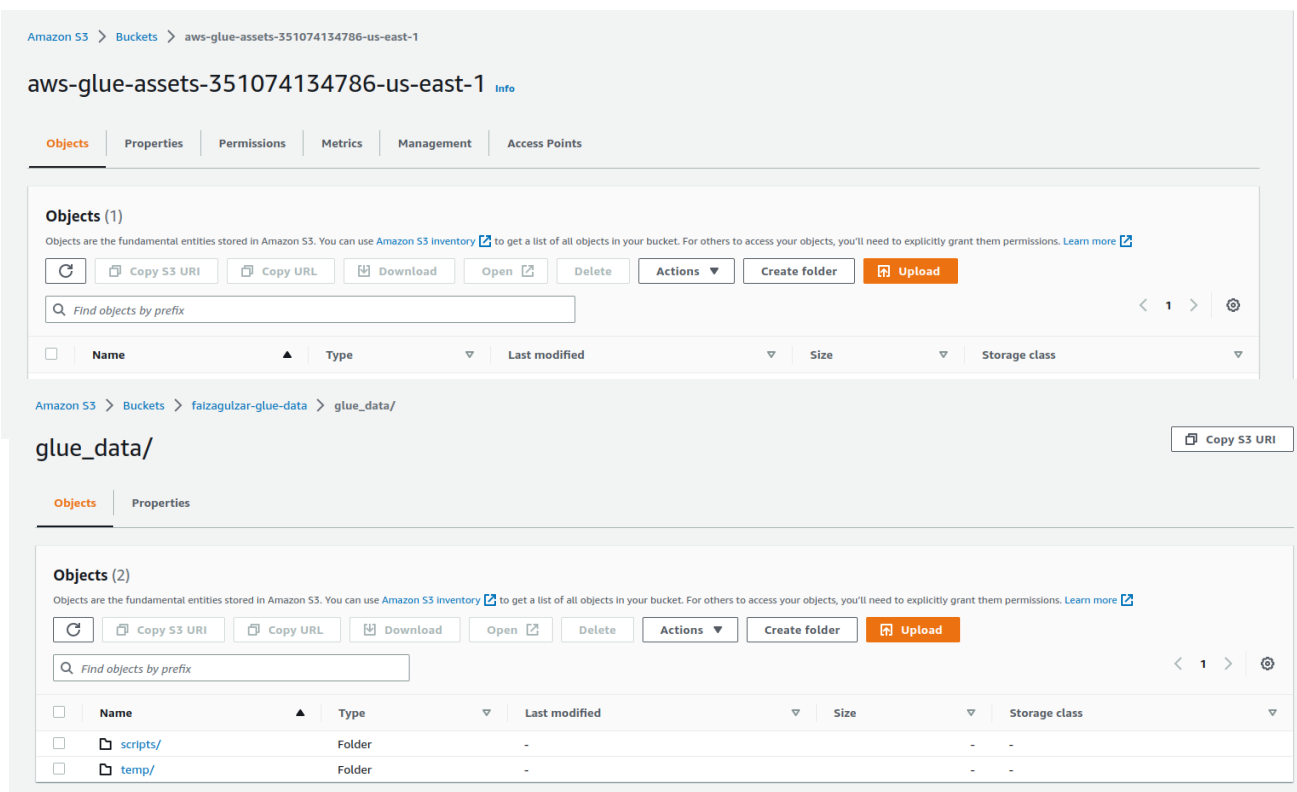
Task 1: Data access preparation.

We have to prepare to data sources on AWS, S3 bucket and RDS database.

As we are performing ETL process using AWS, the very first step of ETL process is to extract the data from multiple sources. So, we need to have storage on AWS so we can load data to move towards the next step, here the use of **S3** and **RDS** services comes.

S3 is one of services of AWS which can be used to store the data, it provides overall mechanism to load the data, make directories, and make it interact with another services. We are using it to store the data so we can, we can access it in ETL jobs with the help Glue crawlers, do some transformation and return the output tables in S3 bucket.

Below are the necessary required directories that we have created for input and output. Bucket contain all the files related to task and assignment.



Amazon S3 > Buckets > faizagulzar-glue-data > glue_data/ > scripts/

scripts/

Copy S3 URI

Objects | Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
assignment_job.py	py	May 18, 2023, 10:34:54 (UTC+05:00)	3.5 KB	Standard
faiza_gulzar.py	py	May 18, 2023, 08:15:33 (UTC+05:00)	2.9 KB	Standard

Amazon S3 > Buckets > faizagulzar-glue-data > input_data/

input_data/

Copy S3 URI

Objects | Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
earnings/	Folder	-	-	-
location/	Folder	-	-	-

Amazon S3 > Buckets > faizagulzar-glue-data > input_data/ > earnings/

earnings/

Copy S3 URI

Objects | Properties

Objects (2)

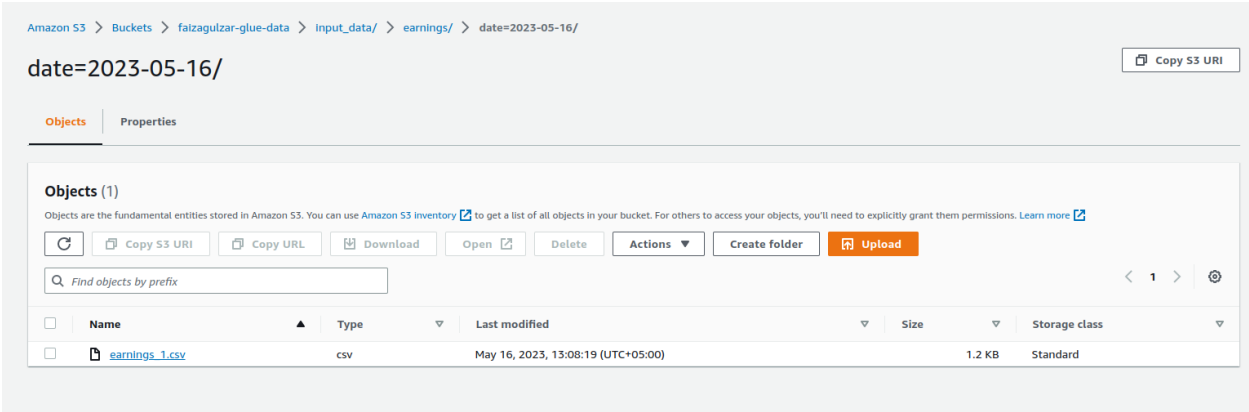
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

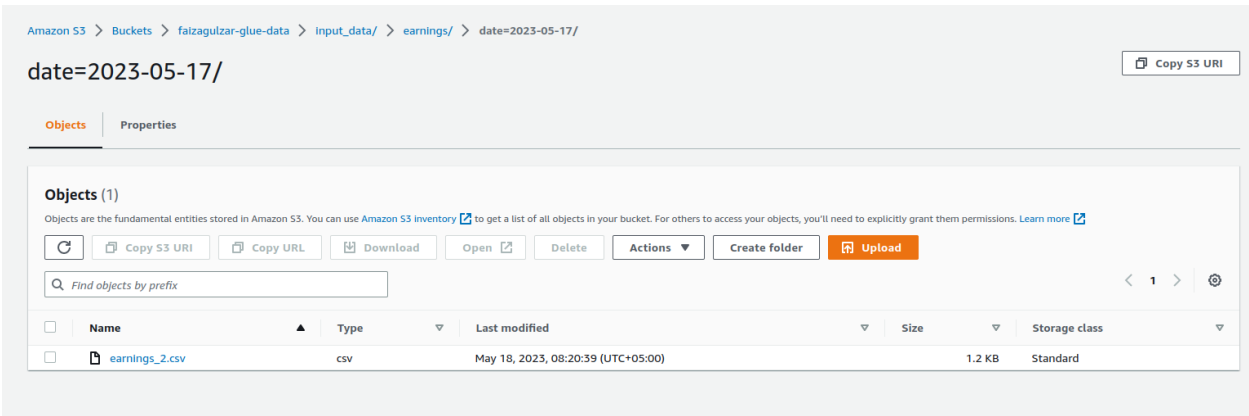
Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

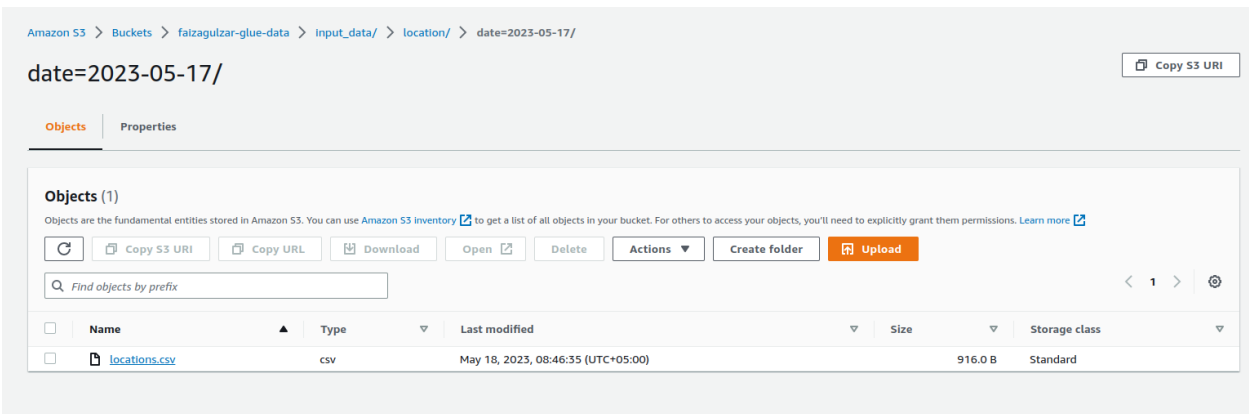
Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
date=2023-05-16/	Folder	-	-	-
date=2023-05-17/	Folder	-	-	-





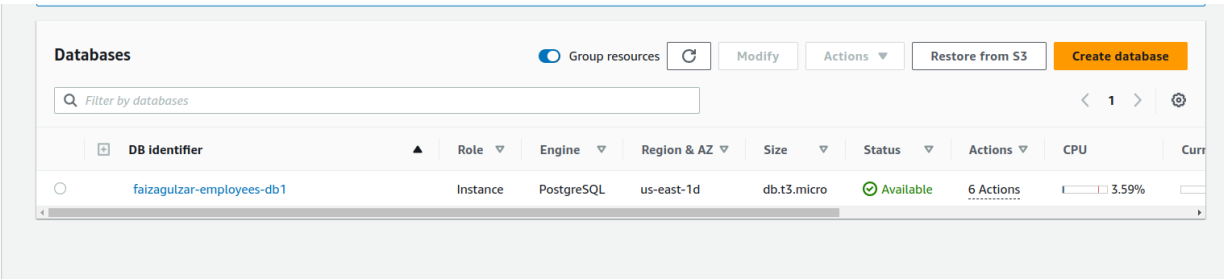


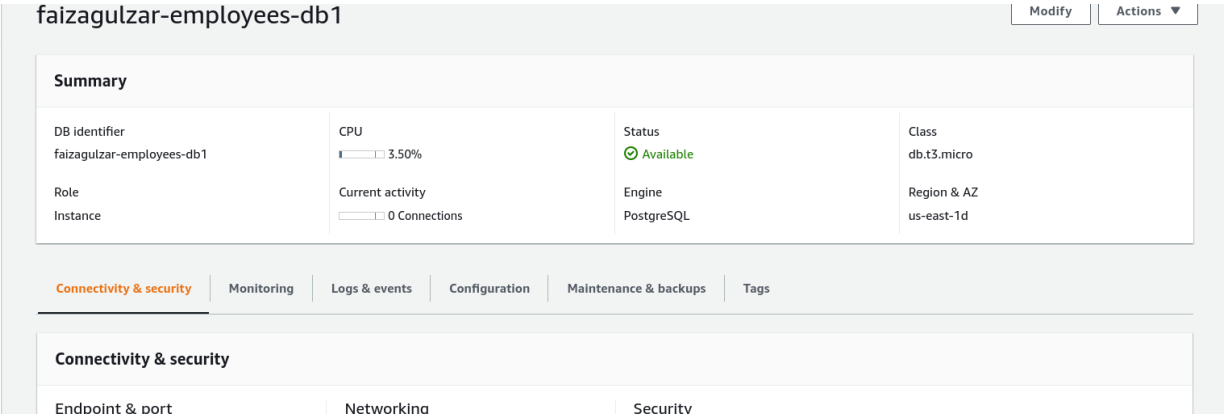
Now we have data stored in S3 bucket, we are creating database using **RDS** service of AWS to make another data source.

RDS service help to create databases, provide overall mechanism to create and manage the database. It supports various type of relational databases such as PostgreSQL, MySQL oracle etc.

We are using it as another data source as the main objective of ETL process to extract the data from multiple sources. We are using various python modules to access RDS and store or populate the data into it.

We have created the RDS database by following given guides.



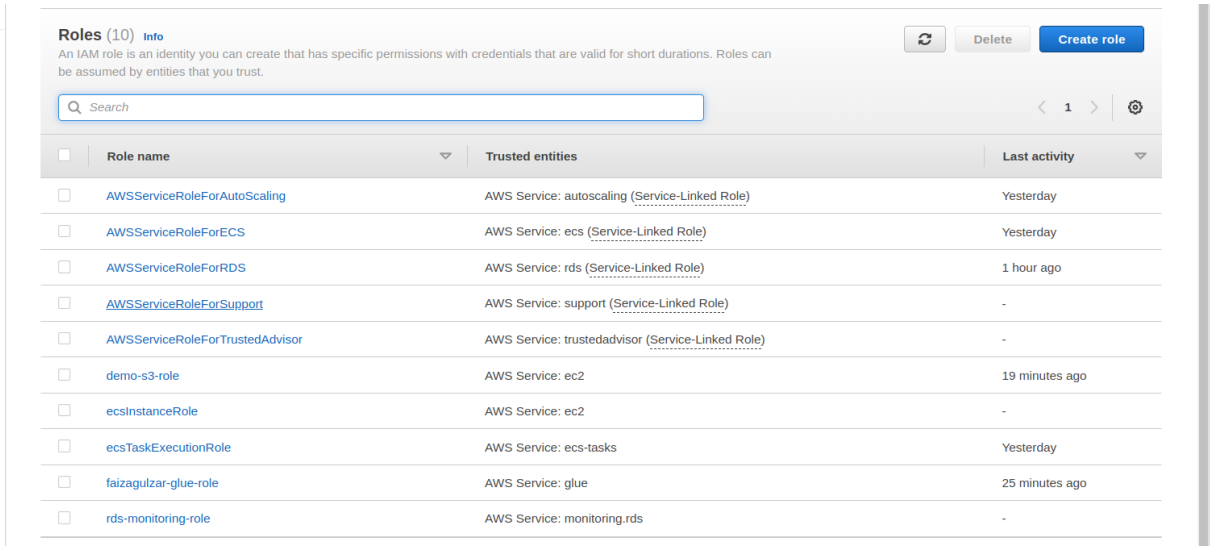


IAM roles.

IAM service is for managing users and roles for using AWS services, it helps to create users and roles to assign them permissions according to their requirement and job.

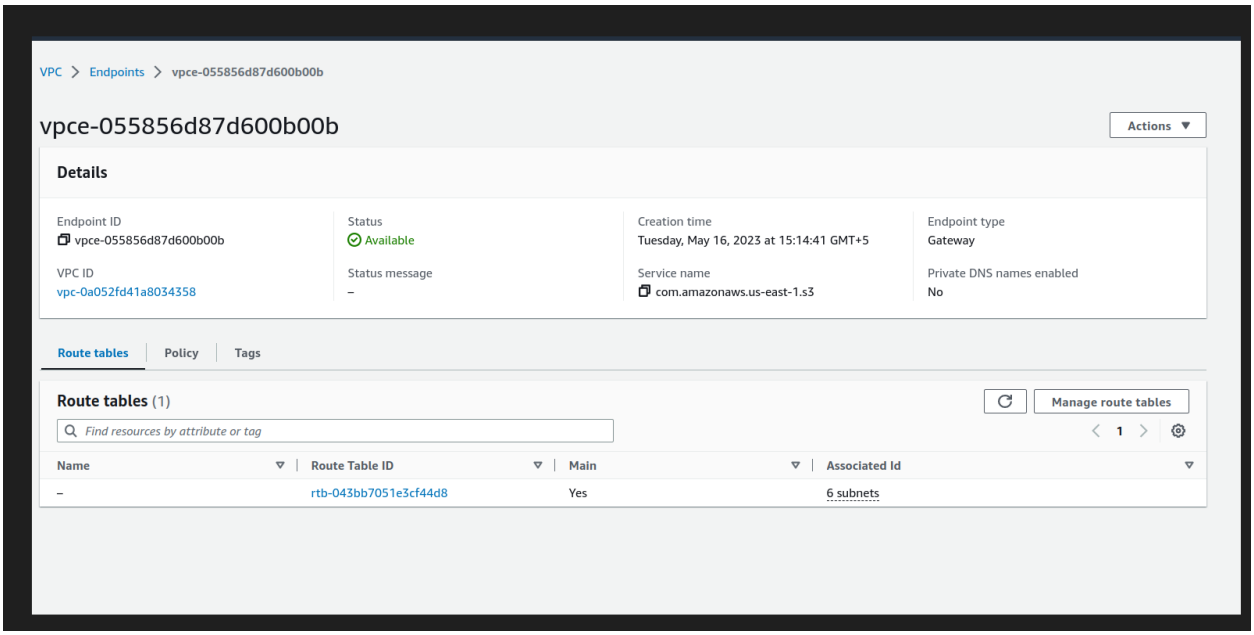
Here we are using it to create role for glue crawlers for allowing it to have full access of RDS and S3 service of database for retrieving and storing the data.

The role is created named **faizagulzar_glue_role**



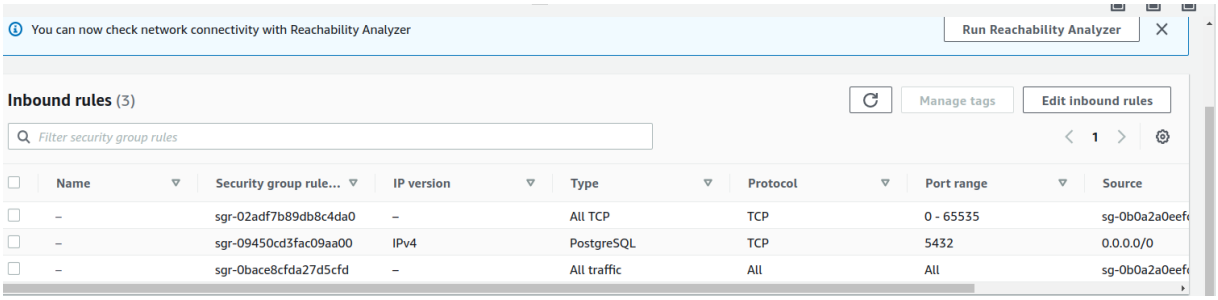
VPC endpoints

VPC service is commonly used to make the secure and private network between or within the AWS services, here we are using it to securely access data in glue jobs and crawlers from s3 service.



Security group rules

Security group on AWS are similar to firewall which monitor and control incoming and outgoing traffic between or within the AWS services. Here we are using them to define rules for the access RDS from glue crawlers and python file which is created in our local directory. The reason access RDS is to store and retrieve the data from database.



Populate the database

Here we are making sure RDS is accessible to our local machine by using python file and also for inserting data through defining required configuration in our database.

```python
def main():
    try:
        db_connector = DatabaseConnector(
            drivername="postgresql",
            database="employees_db",
            # username you set during the creation of the database
            username="faizagulzar",
            # endpoint url, to be found in the `Connectivity & security` section in RDS
            host="faizagulzar-employees-db1.cslrt2k7hrrx.us-east-1.rds.amazonaws.com",
            # password you set during the creation of the database
            password="mod4day2empl",
        )
    except OperationalError:
        raise ConnectionError("Could not establish DB connection.")
```

Used the given commands run given py file and insert the data in to RDS

```shell
sudo apt install python3-dev
sudo apt install libpq-dev
pip install -r requirements.txt
python3 populate_db.py
```

Connection verified and data is successfully inserted in our RDS database.

```
faizakiyani@all-MS-7D35:~/Documents/day_2_aws_etl$ python3 populate_db.py
(526540, 'Angelique', 'K', 'Goodwin', 'angelique.goodwin@gmail.com', '1964-05-15', '2001-03-24', '471-57-0359', '212-884-7146', 'akgoodwin', 'z
{d>ez%{.@')
(859327, 'Jeni', 'S', 'Shaffer', 'jeni.shaffer@gmail.com', '1962-01-13', '2015-12-10', '624-85-4146', '205-665-7020', 'jsshaffer', '7U56!*!O')
(887387, 'Donald', 'T', 'Farris', 'donald.farris@bellsouth.net', '1958-04-11', '1979-11-12', '097-02-3315', '205-959-7879', 'dtfarris', 'rX.F{j
&]&m&&X')
(779497, 'Steven', 'D', 'Rendon', 'steven.rendon@gmail.com', '1982-04-04', '2008-09-18', '134-98-6566', '217-858-0054', 'sdrendon', 'a+2;sx}<G]
y')
(896517, 'Jenell', 'L', 'Almanza', 'jenell.almanza@yahoo.com', '1958-07-01', '1993-07-14', '599-92-7345', '314-893-2590', 'jlalmanza', 'Ou7RX{y
T')
(220965, 'Almeta', 'Y', 'Brookins', 'almeta.brookins@gmail.com', '1985-05-08', '2017-04-25', '109-98-3095', '229-238-0915', 'aybrookins', 'HQHK
E_+9hv')
```

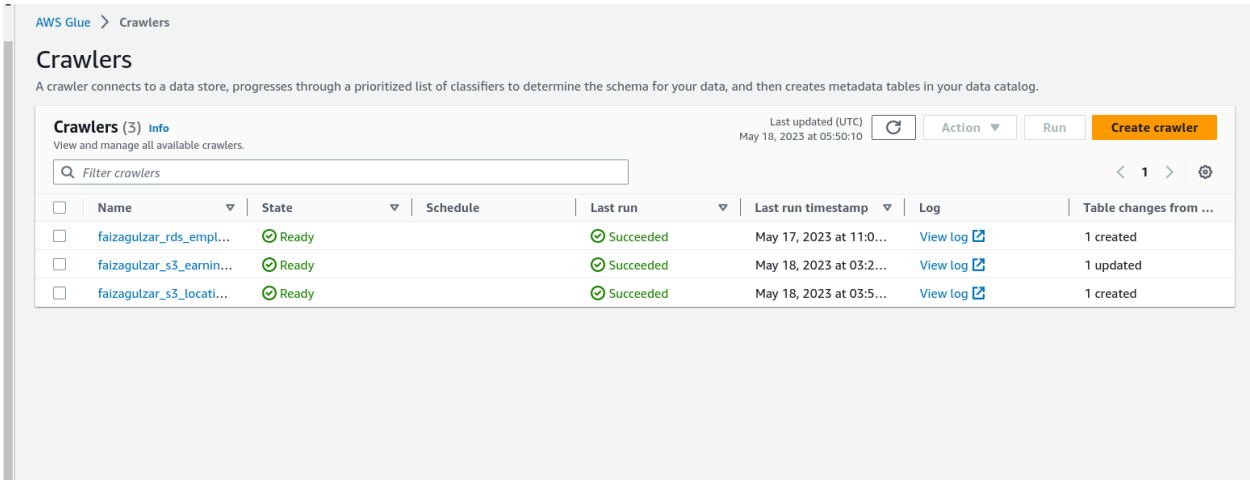
Task 02: ETL in Glue, prepare the glue crawlers and ETL jobs.

AWS Glue service is famous for performing ETL (Extract transform Load) process. It allows us to extract the data from various sources using it's crawlers called as glue crawlers and transform the loaded data for analytics, it provides various functionalities to transform your data such as SQL query, joins, and many more. After transforming it loads your data in your defined sources, like here we are returning the output in S3 bucket in parquet format. It also supports various widely used formats.

Initially, we create a data catalog which helps us to understand and the manages our data sources, it stores the metadata which helps to load the unique data in the crawlers.

We are creating three crawlers here one for S3 bucket, RDS and last for assignment which is also from the S3 bucket. These crawlers extract the data from given sources once we run them, after every run they extract the unique data only not the duplicate.

The crawlers are ready to use and successfully ran which means they extracted the data for transformation.



While crating crawler, we define source of data for S3 bucket we defined the path, while for RDS database we created connection, which allows communication between RDS and crawler for extracting the data.

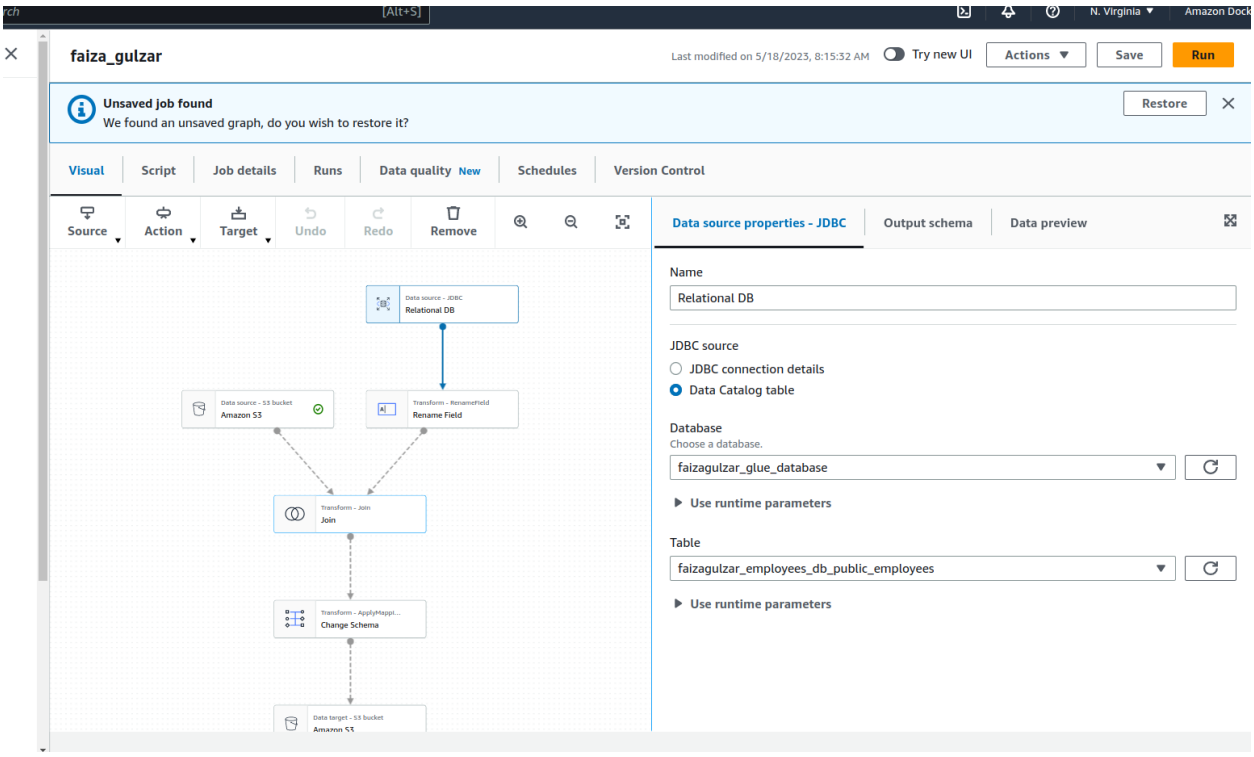
Data sources job

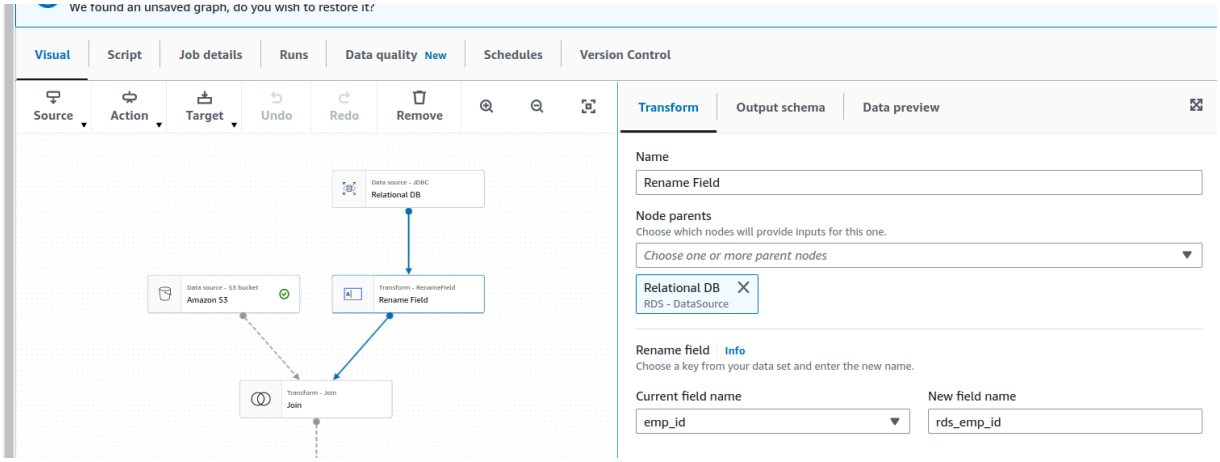
Once our crawlers are running we have created two jobs one for task and another for assignment.

In the job section we transform and load the data in to target sources.

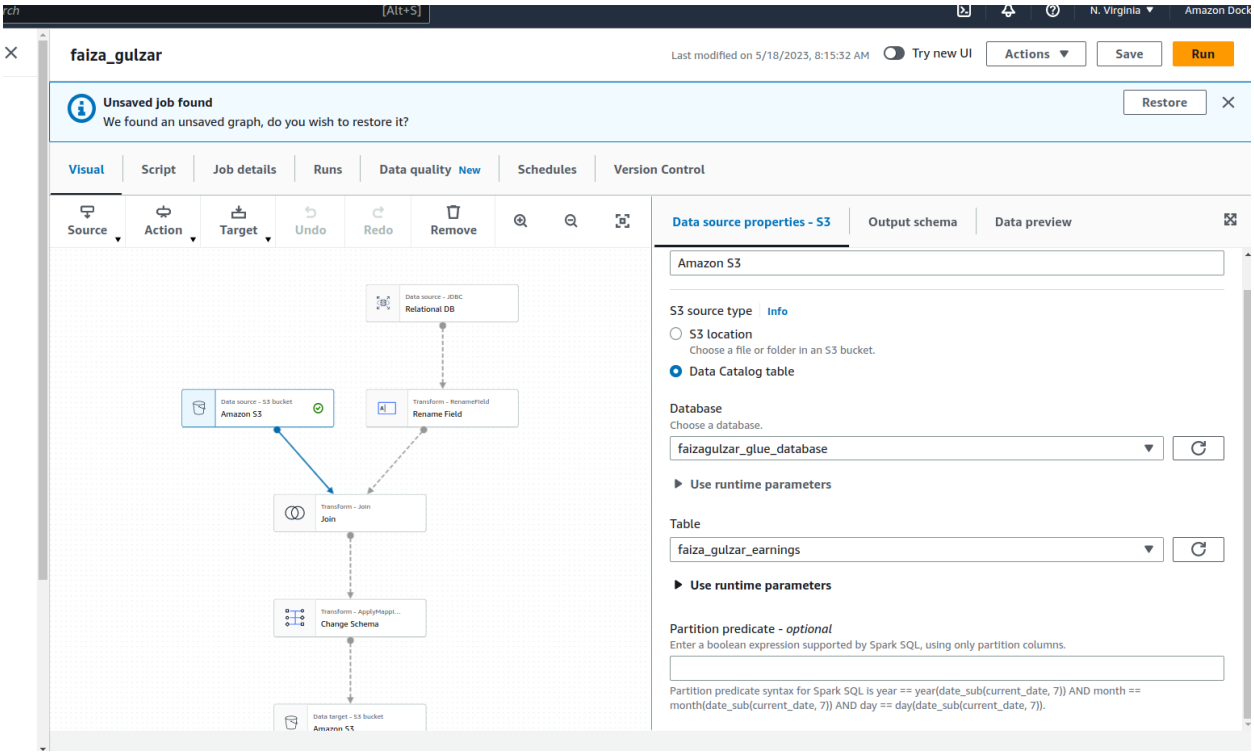
Task job

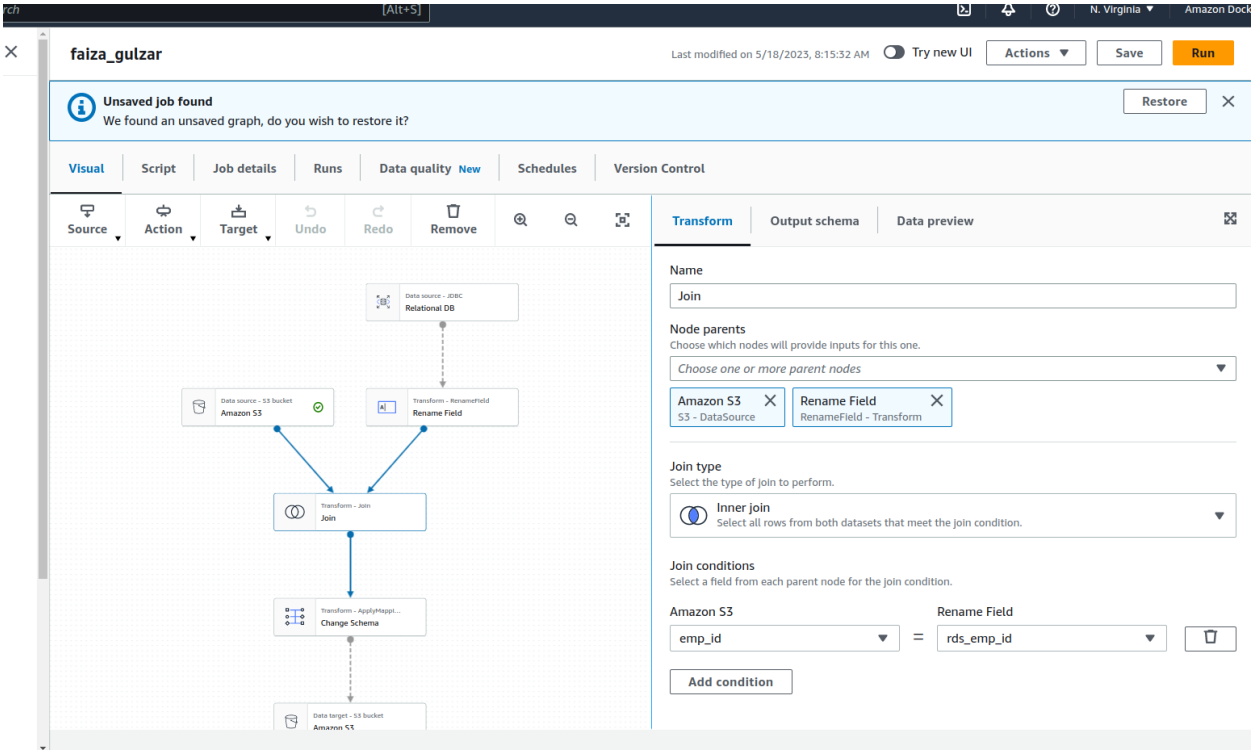
We add defined Relational DB from data source, and add rename field action for renaming emp_id.

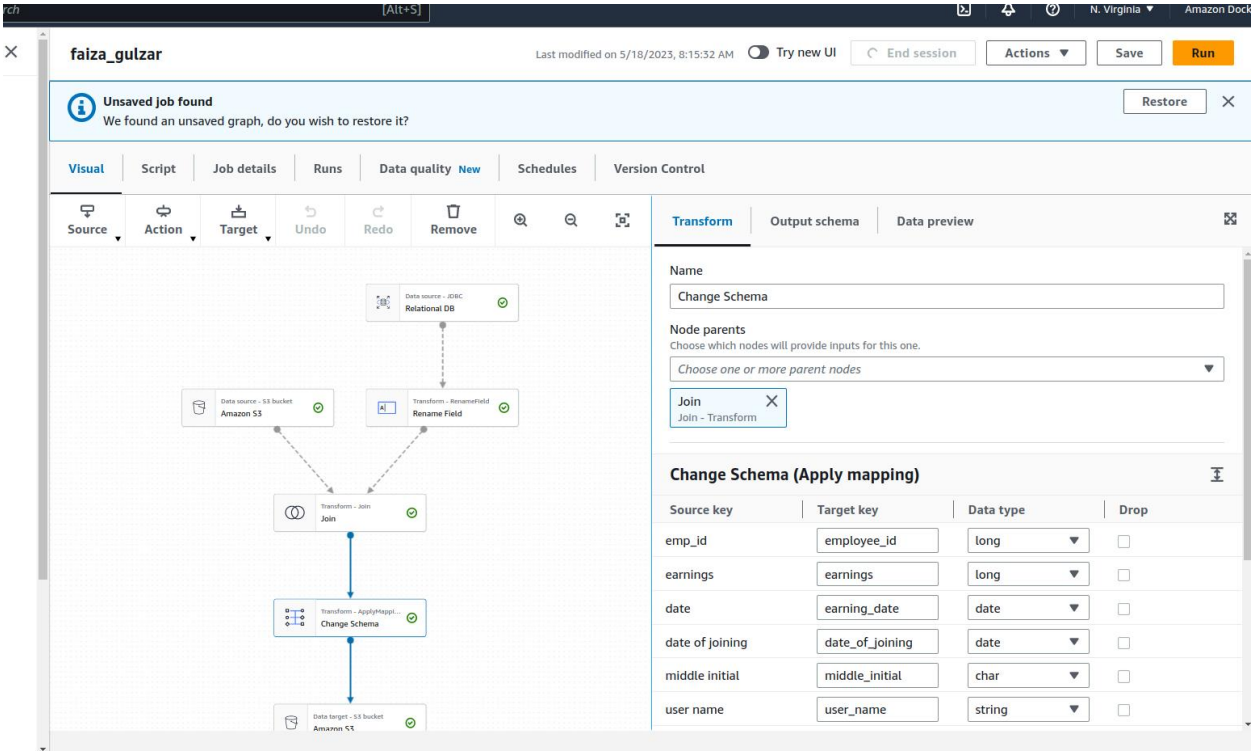


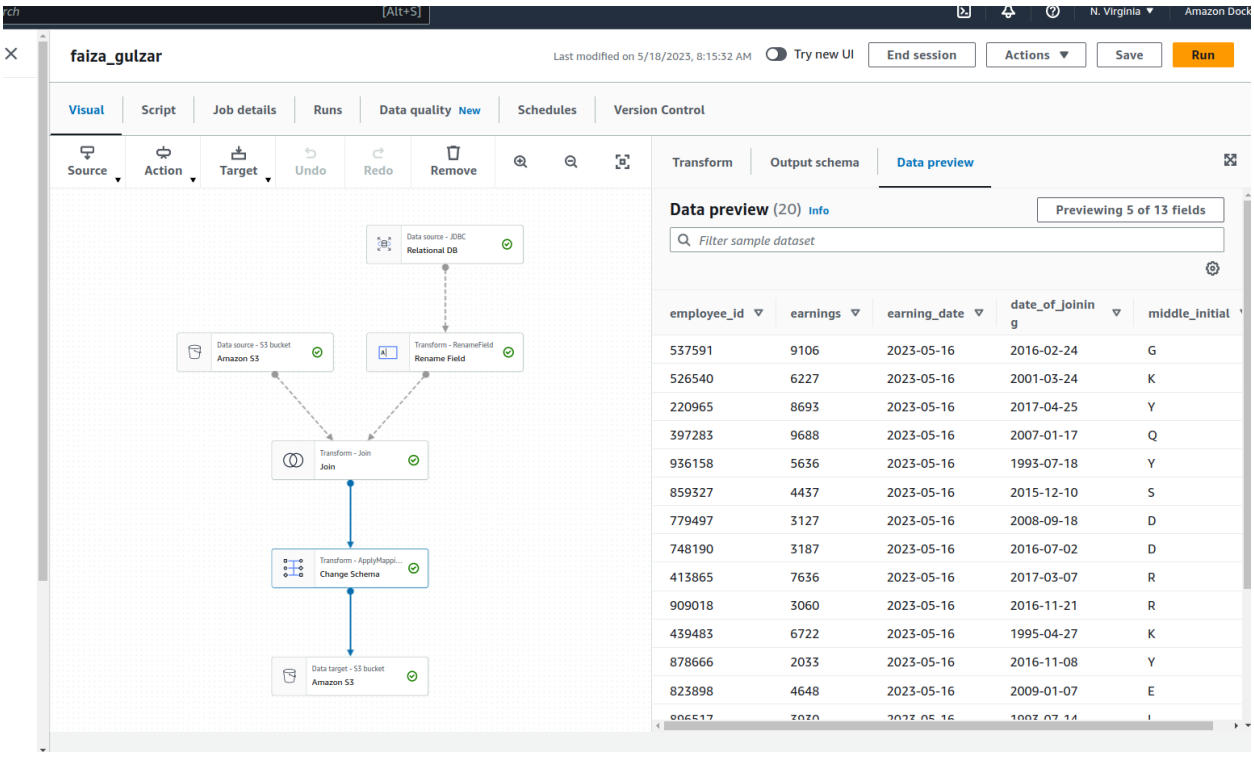


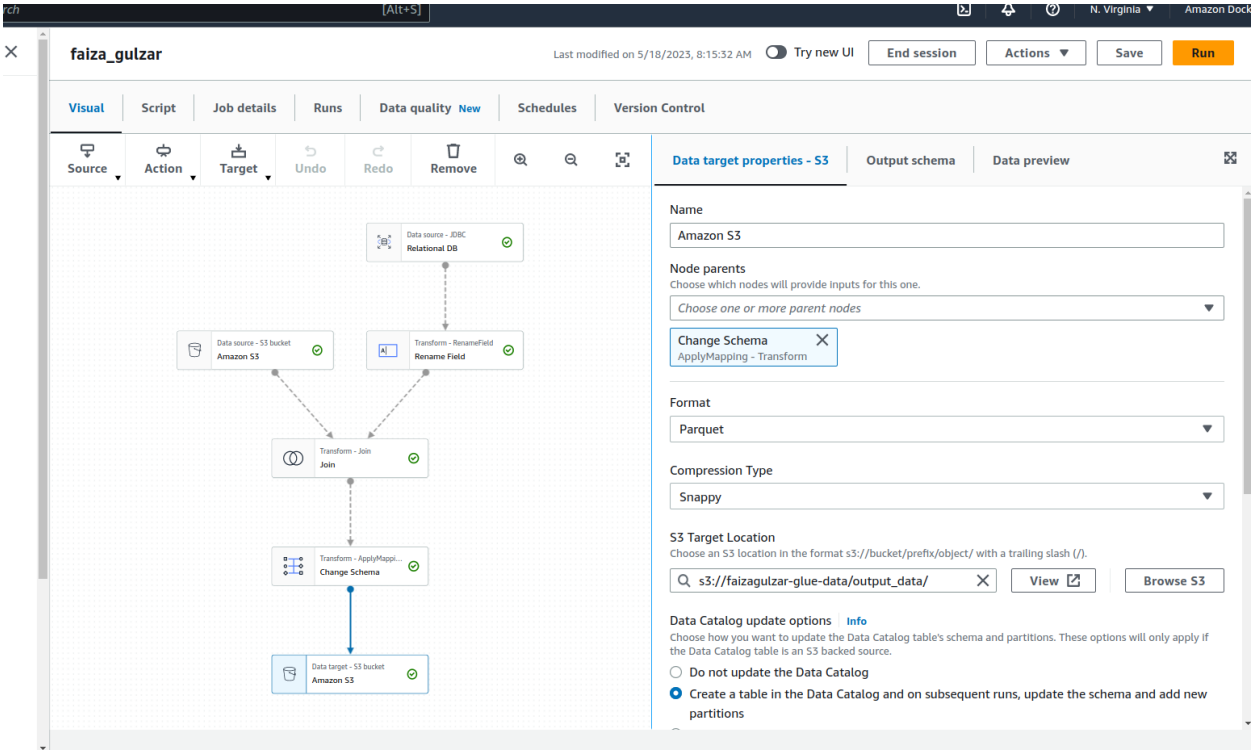
Then add another data source S3 bucket and joined it with RDS source with join section, also implement the given small tasks for rename, changing data types and others and pass the results in change schema.



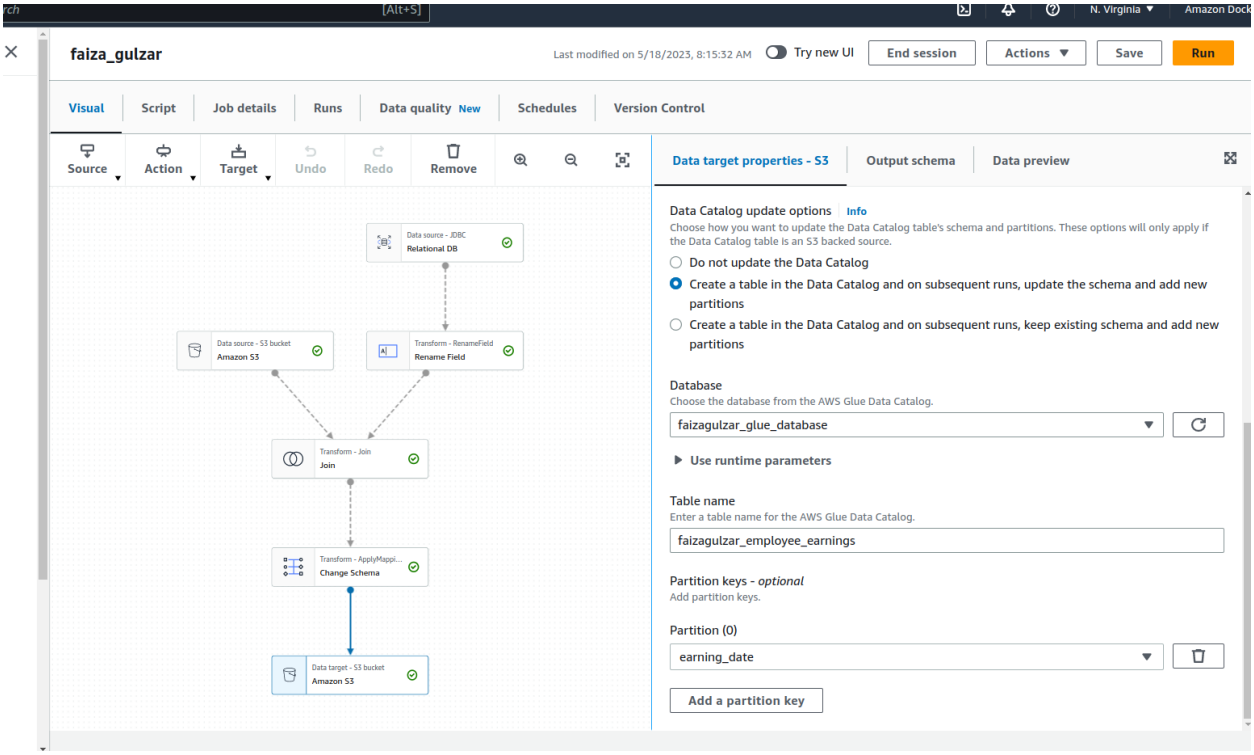



faiza_gulzar
Last modified on 5/18/2023, 8:15:32 AM
Try new UI
End session
Actions
Save
Run
Unsaved job found
We found an unsaved graph, do you wish to restore it?
Restore
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Transform
Output schema
Data preview
Relational DB
Amazon S3
Rename Field
Join
Change Schema
Name
Change Schema
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent nodes
Join
Join - Transform
Change Schema (Apply mapping)
Source key
Target key
Data type
Drop
emp_id
employee_id
long
earnings
earnings
long
date
earning_date
date
date of joining
date_of_joining
date
middle initial
middle_initial
char
user name
user_name
string

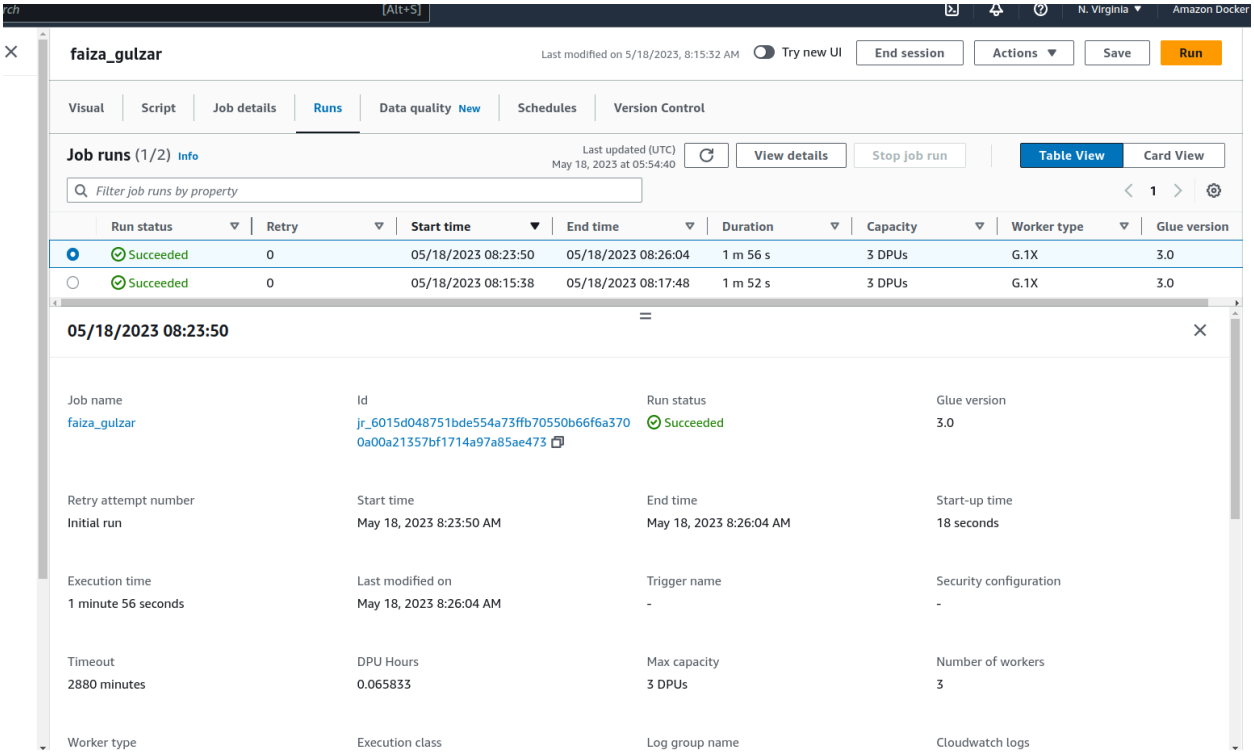

faiza_gulzar
Last modified on 5/18/2023, 8:15:32 AM
Try new UI
End session
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Transform
Output schema
Data preview
Relational DB
Amazon S3
Rename Field
Join
Change Schema
Data preview (20) Info
Previewing 5 of 13 fields
Filter sample dataset
employee_id
earnings
earning_date
date_of_joining
middle_initial
537591 9106 2023-05-16 2016-02-24 G
526540 6227 2023-05-16 2001-03-24 K
220965 8693 2023-05-16 2017-04-25 Y
397283 9688 2023-05-16 2007-01-17 Q
936158 5636 2023-05-16 1993-07-18 Y
859327 4437 2023-05-16 2015-12-10 S
779497 3127 2023-05-16 2008-09-18 D
748190 3187 2023-05-16 2016-07-02 D
413865 7636 2023-05-16 2017-03-07 R
909018 3060 2023-05-16 2016-11-21 R
439483 6722 2023-05-16 1995-04-27 K
878666 2033 2023-05-16 2016-11-08 Y
823898 4648 2023-05-16 2009-01-07 E



Now we add target source for loading the transformed data which is directory of created S3 bucket set the output nodes and their required configuration for format.



The provided given job details and successfully ran the job.



Assignment job

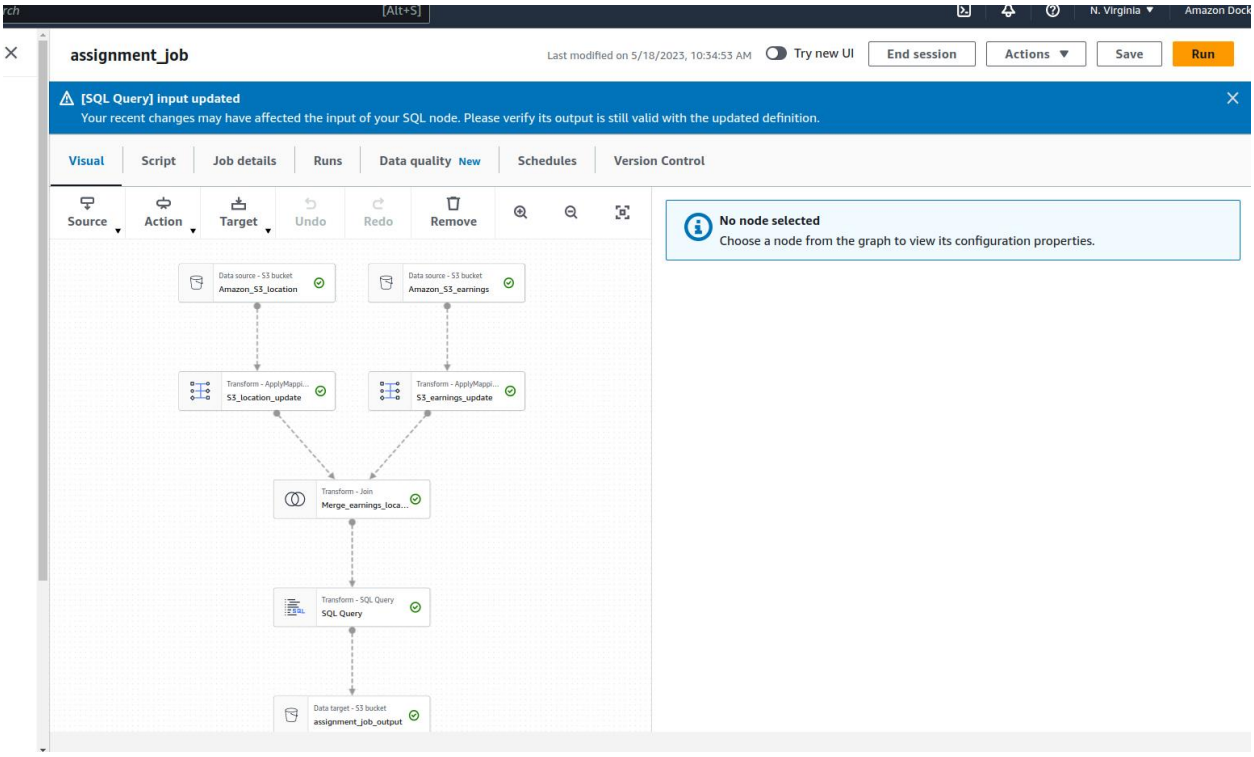
Initially we created directories for location input and output in s3 bucket, then crawler which is already well defined in above tasks and can be seen in screenshots.

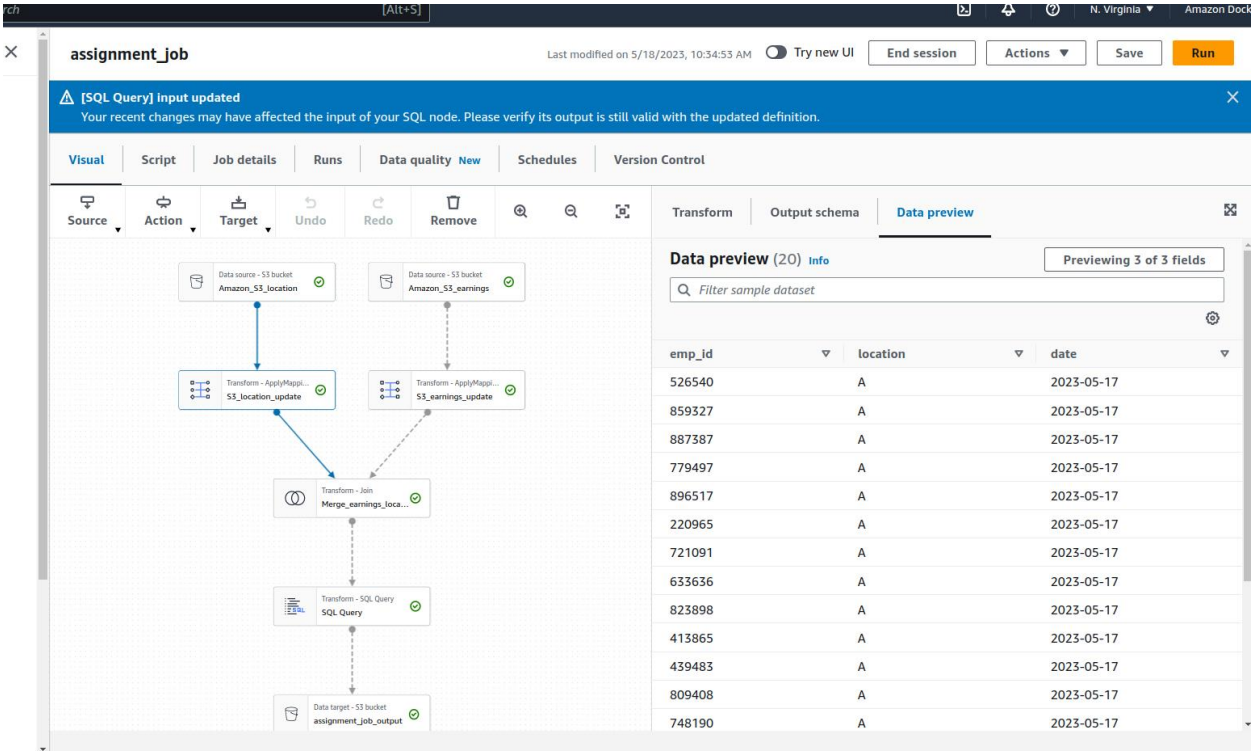
To creating separate job for assignment.

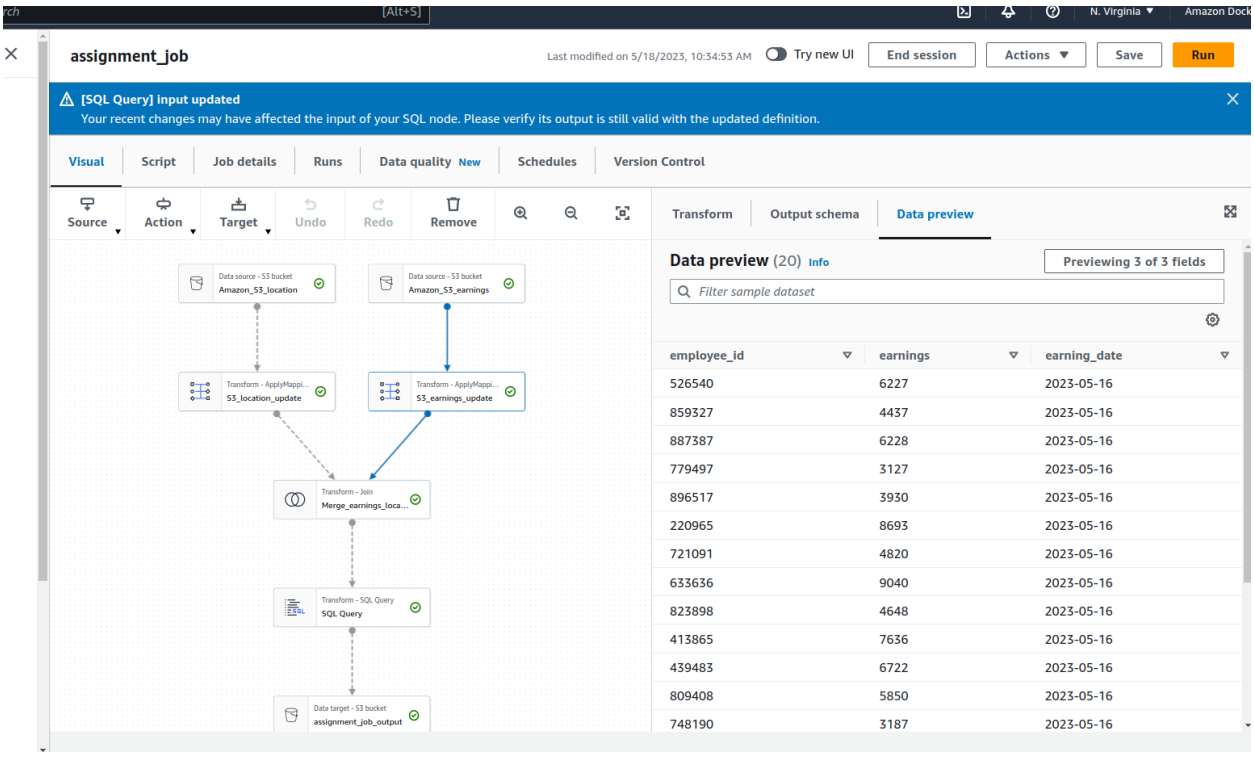
Selected two data sources from s3 bucket earnings files and location files. We did few updates in the tables such as renaming and giving appropriate data type. Then we merge them based on employee id.

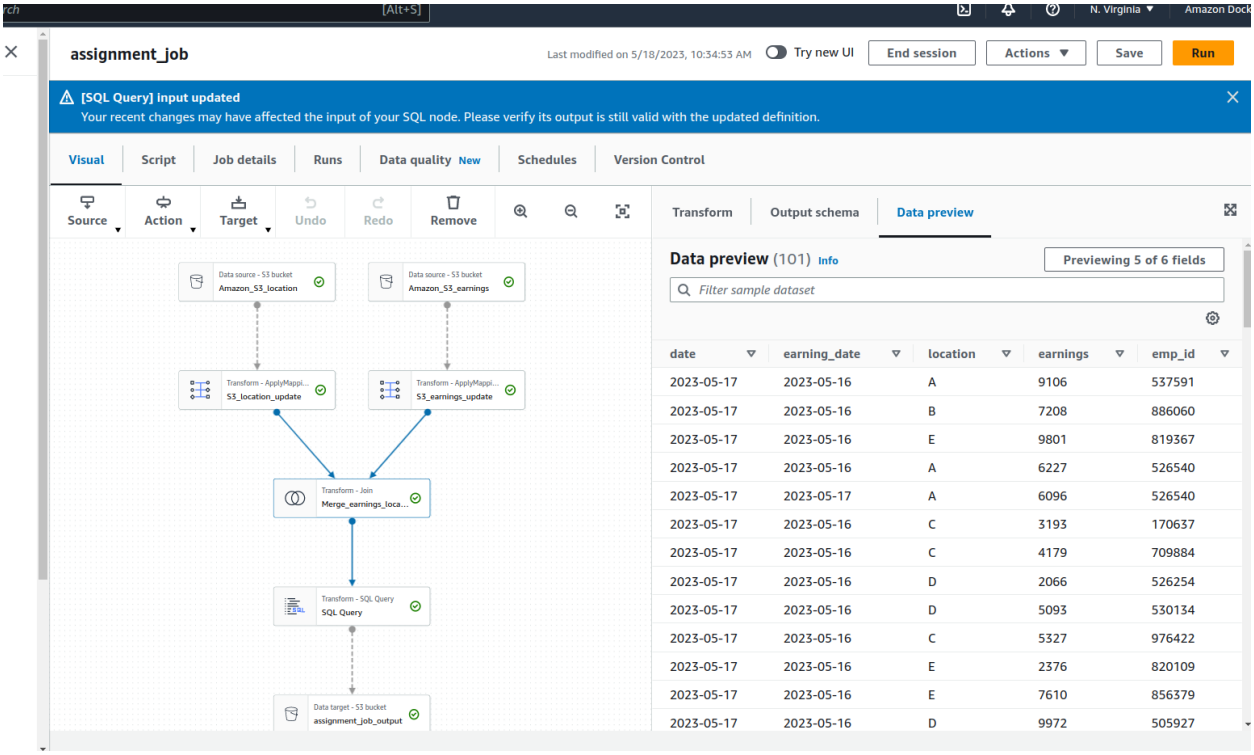
Added SQL Query section for performing given SQL query task.

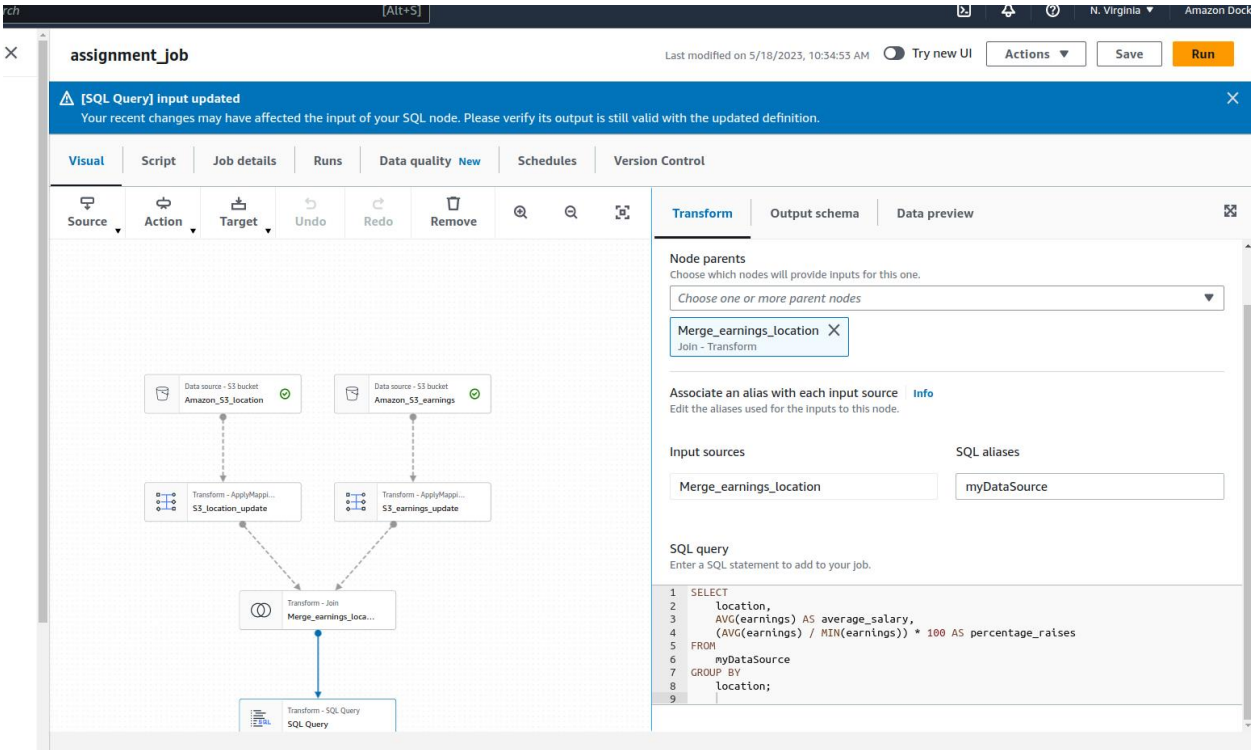
Finally added the data target to load the transformed output data.

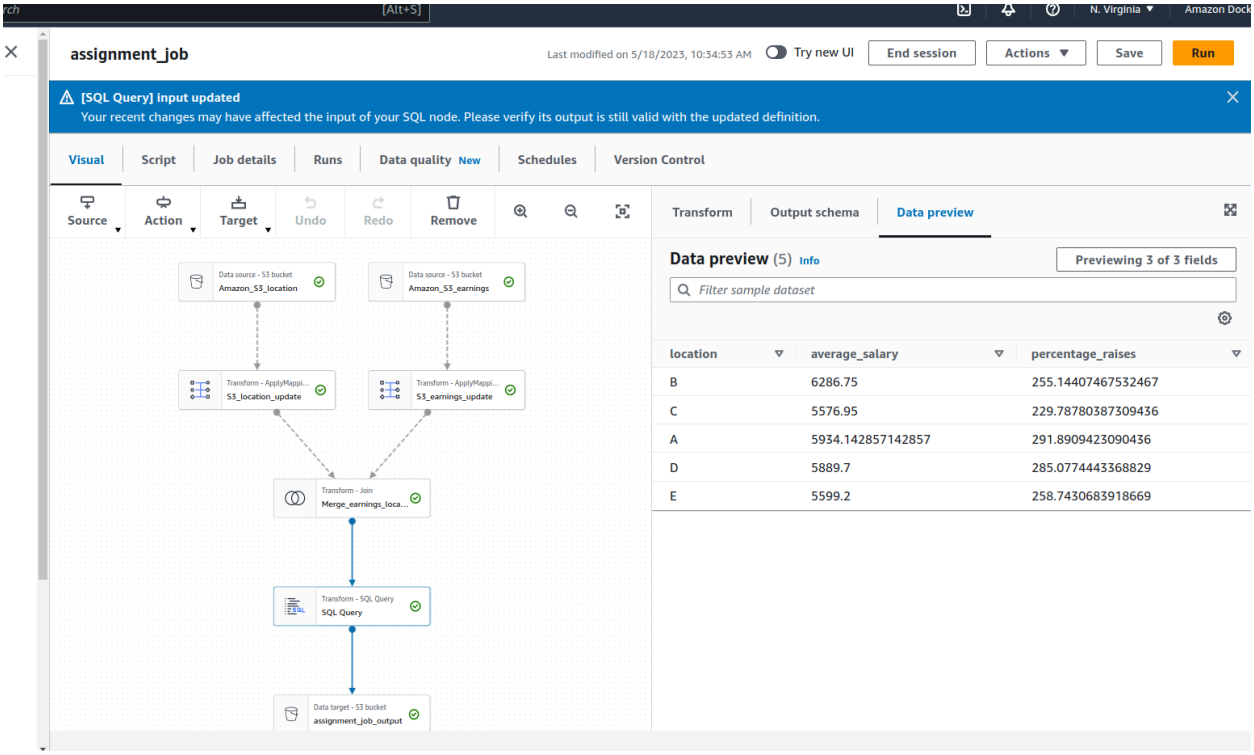


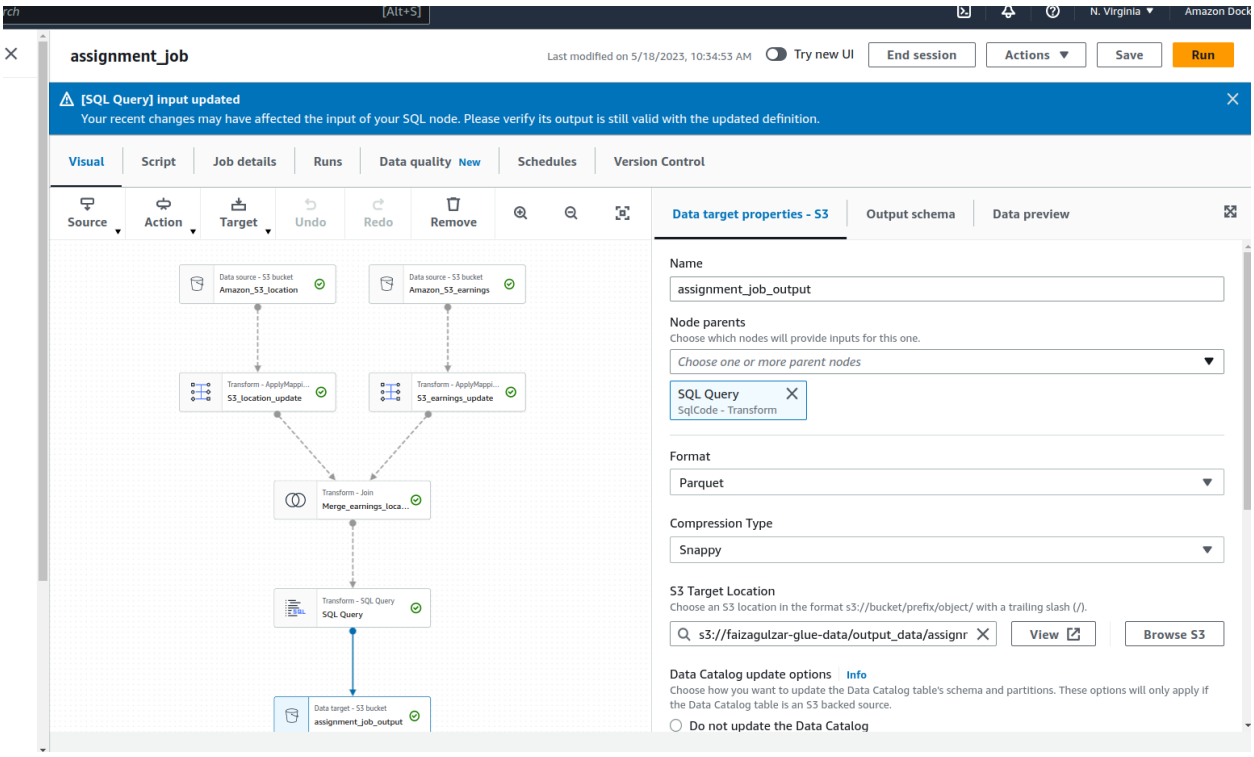

assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
End session
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Data source - S3 bucket
Amazon_S3_location
Data source - S3 bucket
Amazon_S3_earnings
Transform - ApplyMappi...
S3_location_update
Transform - ApplyMappi...
S3_earnings_update
Transform - Join
Merge_earnings_loca...
Transform - SQL Query
SQL Query
Data target - S3 bucket
assignment_job_output
Transform
Output schema
Data preview
Data preview (20) Info
Previewing 3 of 3 fields
Filter sample dataset
emp_id
location
date
526540 A 2023-05-17
859327 A 2023-05-17
887387 A 2023-05-17
779497 A 2023-05-17
896517 A 2023-05-17
220965 A 2023-05-17
721091 A 2023-05-17
633636 A 2023-05-17
823898 A 2023-05-17
413865 A 2023-05-17
439483 A 2023-05-17
809408 A 2023-05-17
748190 A 2023-05-17

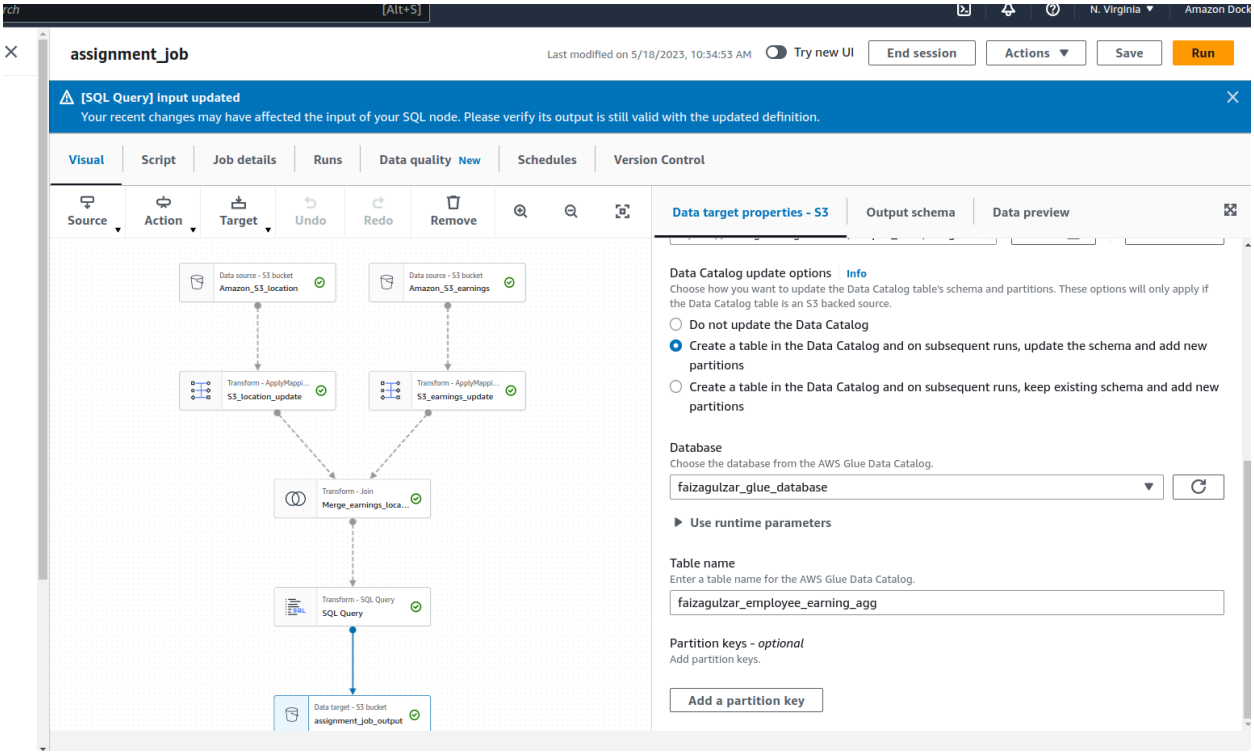

assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
End session
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Data source - S3 bucket
Amazon_S3_location
Data source - S3 bucket
Amazon_S3_earnings
Transform - ApplyMappi...
S3_location_update
Transform - ApplyMappi...
S3_earnings_update
Transform - Join
Merge_earnings_loca...
Transform - SQL Query
SQL Query
Data target - S3 bucket
assignment_job_output
Transform
Output schema
Data preview
Data preview (20) Info
Previewing 3 of 3 fields
Filter sample dataset
employee_id
earnings
earning_date
526540 6227 2023-05-16
859327 4437 2023-05-16
887387 6228 2023-05-16
779497 3127 2023-05-16
896517 3930 2023-05-16
220965 8693 2023-05-16
721091 4820 2023-05-16
633636 9040 2023-05-16
823898 4648 2023-05-16
413865 7636 2023-05-16
439483 6722 2023-05-16
809408 5850 2023-05-16
748190 3187 2023-05-16

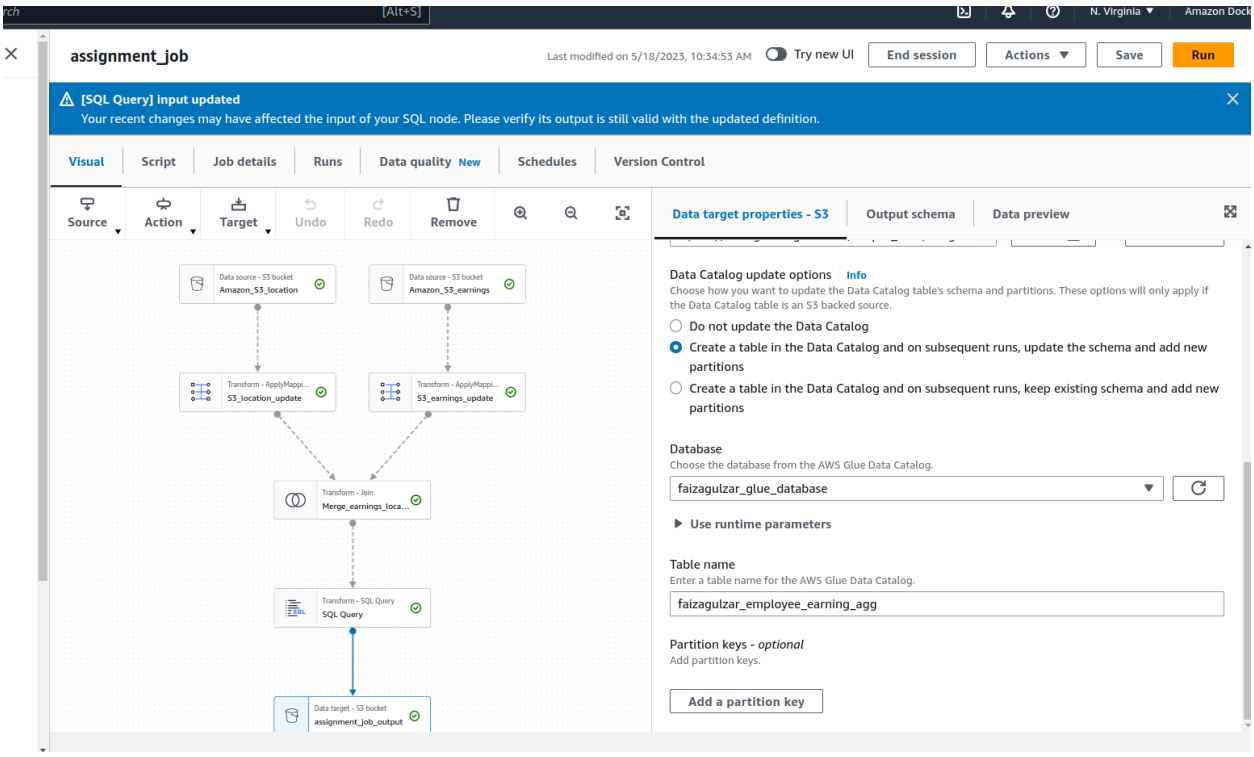

assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
End session
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Transform
Output schema
Data preview
Data source - S3 bucket
Amazon_S3_location
Data source - S3 bucket
Amazon_S3_earnings
Transform - ApplyMappi...
S3_location_update
Transform - ApplyMappi...
S3_earnings_update
Transform - Join
Merge_earnings_loca...
Transform - SQL Query
SQL Query
Data target - S3 bucket
assignment_job_output
Data preview (101) Info
Previewing 5 of 6 fields
Filter sample dataset
date
earning_date
location
earnings
emp_id
2023-05-17 2023-05-16 A 9106 537591
2023-05-17 2023-05-16 B 7208 886060
2023-05-17 2023-05-16 E 9801 819367
2023-05-17 2023-05-16 A 6227 526540
2023-05-17 2023-05-17 A 6096 526540
2023-05-17 2023-05-16 C 3193 170637
2023-05-17 2023-05-16 C 4179 709884
2023-05-17 2023-05-16 D 2066 526254
2023-05-17 2023-05-16 D 5093 530134
2023-05-17 2023-05-16 C 5327 976422
2023-05-17 2023-05-16 E 2376 820109
2023-05-17 2023-05-16 E 7610 856379
2023-05-17 2023-05-16 D 9972 505927


assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Transform
Output schema
Data preview
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent nodes
Merge_earnings_location
Join - Transform
Associate an alias with each input source Info
Edit the aliases used for the inputs to this node.
Input sources
SQL aliases
Merge_earnings_location
myDataSource
SQL query
Enter a SQL statement to add to your job.
SELECT
location,
AVG(earnings) AS average_salary,
(AVG(earnings) / MIN(earnings)) * 100 AS percentage_raises
FROM
myDataSource
GROUP BY
location;


assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
End session
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Data source - S3 bucket
Amazon_S3_location
Data source - S3 bucket
Amazon_S3_earnings
Transform - ApplyMappi...
S3_location_update
Transform - ApplyMappi...
S3_earnings_update
Transform - Join
Merge_earnings_loca...
Transform - SQL Query
SQL Query
Data target - S3 bucket
assignment_job_output
Transform
Output schema
Data preview
Data preview (5) Info
Previewing 3 of 3 fields
Filter sample dataset
location
average_salary
percentage_raises
B 6286.75 255.14407467532467
C 5576.95 229.78780387309436
A 5934.142857142857 291.8909423090436
D 5889.7 285.0774443368829
E 5599.2 258.7430683918669


assignment_job
Last modified on 5/18/2023, 10:34:53 AM
Try new UI
End session
Actions
Save
Run
[SQL Query] input updated
Your recent changes may have affected the input of your SQL node. Please verify its output is still valid with the updated definition.
Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Source
Action
Target
Undo
Redo
Remove
Data target properties - S3
Output schema
Data preview
Name
assignment_job_output
Node parents
Choose which nodes will provide inputs for this one.
Choose one or more parent nodes
SQL Query
SqlCode - Transform
Format
Parquet
Compression Type
Snappy
S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
s3://faizagulzar-glue-data/output_data/assignr
View
Browse S3
Data Catalog update options Info
Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.
Do not update the Data Catalog





Confirming output of both task and assignment output directories in bucket

Amazon S3 > Buckets > faizagulzar-glue-data > output_data/

output_data/

Copy S3 URI

Objects | Properties

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
assignment_job/	Folder	-	-	-
earning_date=2023-05-16/	Folder	-	-	-
earning_date=2023-05-17/	Folder	-	-	-

Amazon S3 > Buckets > faizagulzar-glue-data > output_data/ > assignment_job/

assignment_job/

Copy S3 URI

Objects | Properties

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
run-1684388185965-part-block-0-r-00001-snappy.parquet	parquet	May 18, 2023, 10:36:31 (UTC+05:00)	606.0 B	Standard
run-1684388185965-part-block-0-r-00002-snappy.parquet	parquet	May 18, 2023, 10:36:31 (UTC+05:00)	597.0 B	Standard
run-1684388185965-part-block-0-r-00003-snappy.parquet	parquet	May 18, 2023, 10:36:31 (UTC+05:00)	597.0 B	Standard
run-1684388185965-part-block-0-r-00006-snappy.parquet	parquet	May 18, 2023, 10:36:31 (UTC+05:00)	597.0 B	Standard

Amazon S3 > Buckets > faizagulzar-glue-data > output_data/ > earning_date=2023-05-16/

earning_date=2023-05-16/

Copy S3 URI

Objects | Properties

Objects (16)

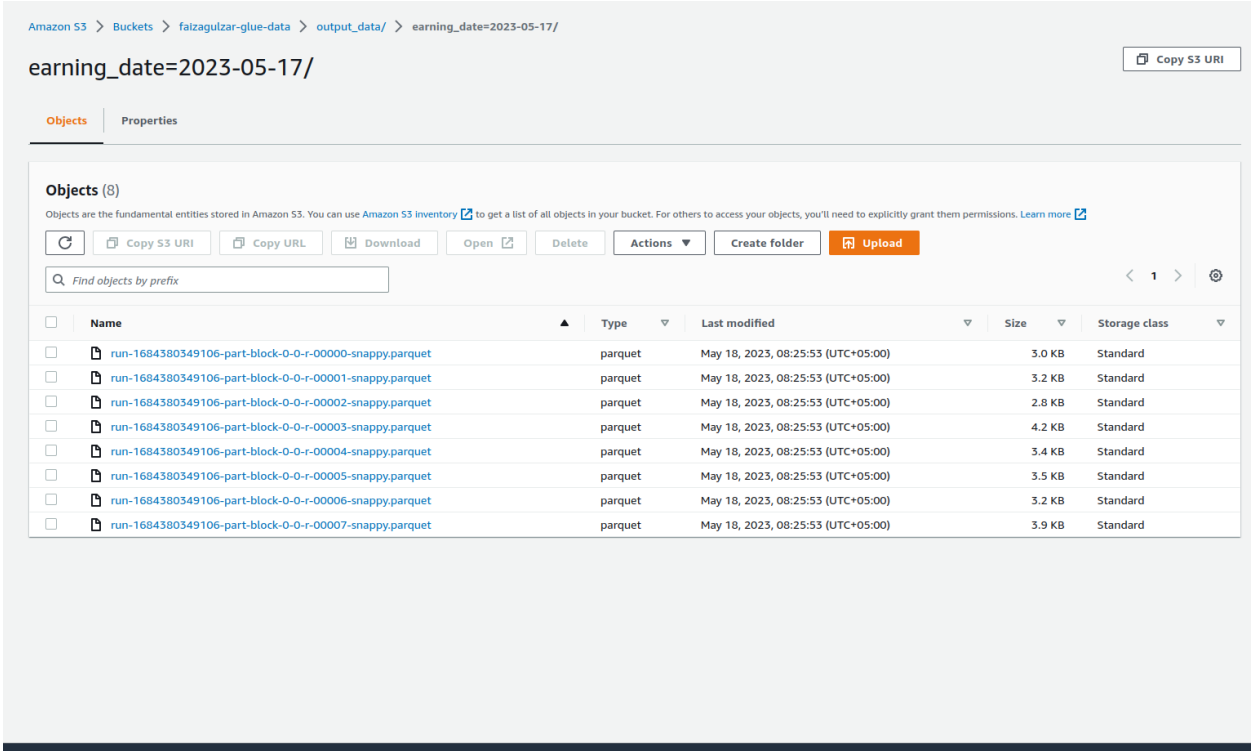
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
run-1684379852970-part-block-0-0-r-00000-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	3.0 KB	Standard
run-1684379852970-part-block-0-0-r-00001-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	3.2 KB	Standard
run-1684379852970-part-block-0-0-r-00002-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	2.8 KB	Standard
run-1684379852970-part-block-0-0-r-00003-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	4.2 KB	Standard
run-1684379852970-part-block-0-0-r-00004-snappy.parquet	parquet	May 18, 2023, 08:17:37 (UTC+05:00)	3.5 KB	Standard
run-1684379852970-part-block-0-0-r-00005-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	3.5 KB	Standard
run-1684379852970-part-block-0-0-r-00006-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	3.2 KB	Standard
run-1684379852970-part-block-0-0-r-00007-snappy.parquet	parquet	May 18, 2023, 08:17:36 (UTC+05:00)	3.9 KB	Standard
run-1684380349106-part-block-0-0-r-00000-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.0 KB	Standard
run-1684380349106-part-block-0-0-r-00001-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.2 KB	Standard
run-1684380349106-part-block-0-0-r-00002-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	2.8 KB	Standard
run-1684380349106-part-block-0-0-r-00003-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	4.2 KB	Standard
run-1684380349106-part-block-0-0-r-00004-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.5 KB	Standard
run-1684380349106-part-block-0-0-r-00005-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.5 KB	Standard
run-1684380349106-part-block-0-0-r-00006-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.2 KB	Standard
run-1684380349106-part-block-0-0-r-00007-snappy.parquet	parquet	May 18, 2023, 08:25:53 (UTC+05:00)	3.9 KB	Standard


Amazon S3 > Buckets > faizagulzar-glue-data > output_data/ > earning_date=2023-05-17/
earning_date=2023-05-17/
Copy S3 URI
Objects
Properties
Objects (8)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
run-1684380349106-part-block-0-0-r-00000-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.0 KB Standard
run-1684380349106-part-block-0-0-r-00001-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.2 KB Standard
run-1684380349106-part-block-0-0-r-00002-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 2.8 KB Standard
run-1684380349106-part-block-0-0-r-00003-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 4.2 KB Standard
run-1684380349106-part-block-0-0-r-00004-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.4 KB Standard
run-1684380349106-part-block-0-0-r-00005-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.5 KB Standard
run-1684380349106-part-block-0-0-r-00006-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.2 KB Standard
run-1684380349106-part-block-0-0-r-00007-snappy.parquet parquet May 18, 2023, 08:25:53 (UTC+05:00) 3.9 KB Standard

The End ☺