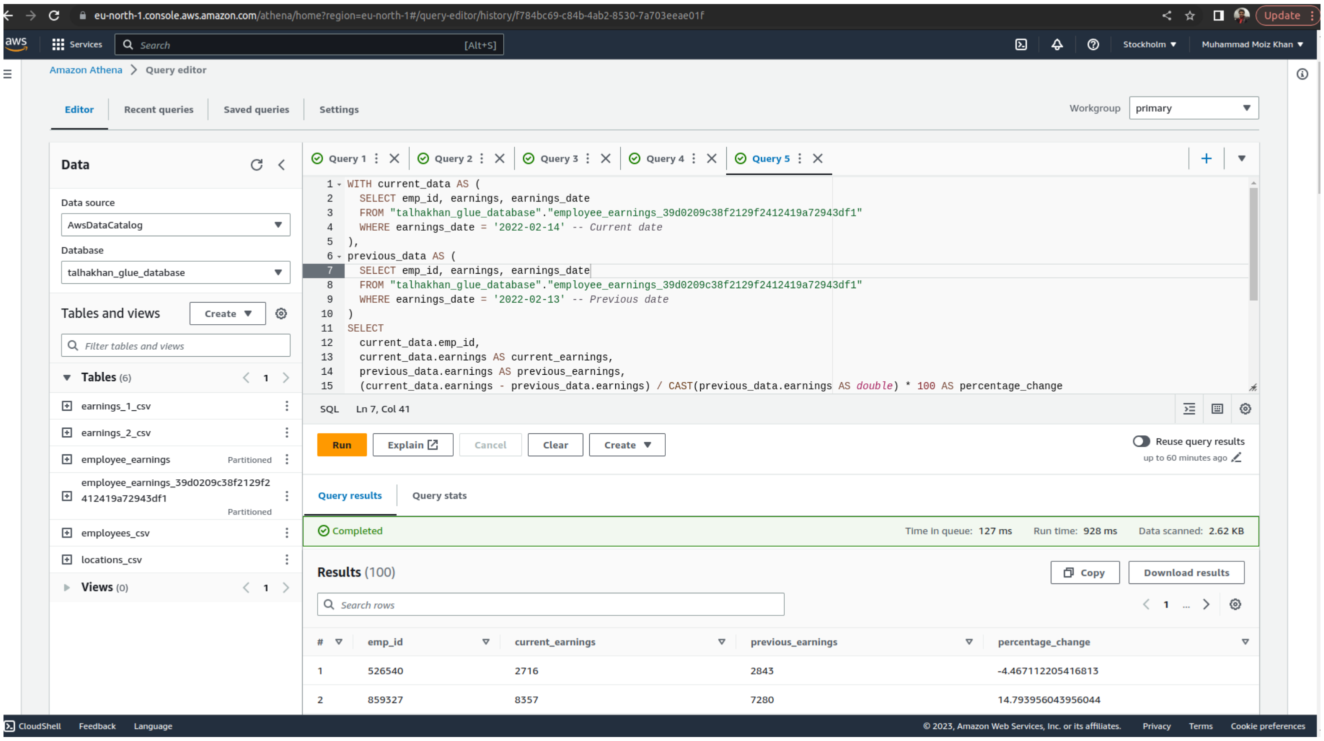
Talha Khan (2303.009.KHI.DEG)

Muhammad Moiz Khan (2303.022.KHI.DEG)

ASSIGNMNET 5.4

Create a new query in Athena that calculates the % change in earnings for every employee from a given day compared to the previous day.



Workgroup: primary

Data source: AwsDataCatalog

Database: talhakhan_glue_database

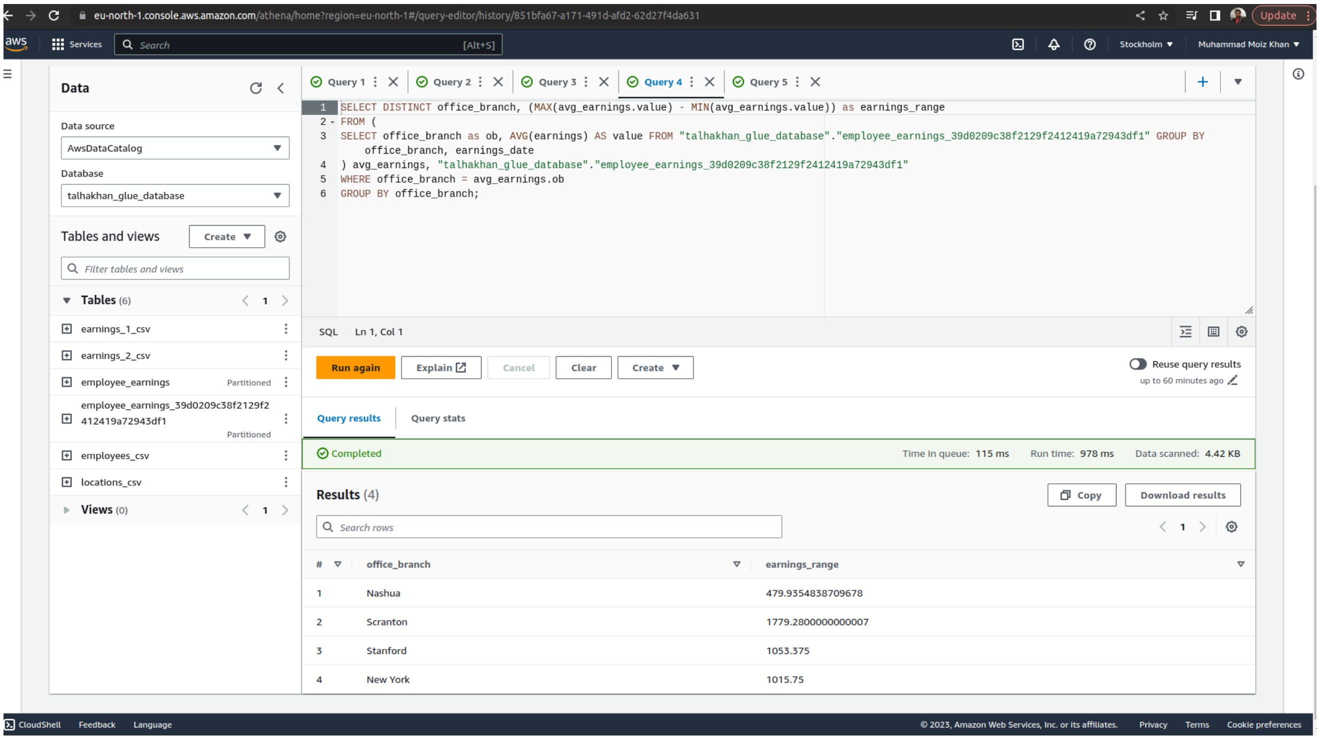
```sql
WITH current_data AS (
  SELECT emp_id, earnings, earnings_date
  FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
  WHERE earnings_date = '2022-02-14' -- Current date
),
previous_data AS (
  SELECT emp_id, earnings, earnings_date
  FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
  WHERE earnings_date = '2022-02-13' -- Previous date
)
SELECT
  current_data.emp_id,
  current_data.earnings AS current_earnings,
  previous_data.earnings AS previous_earnings,
  (current_data.earnings - previous_data.earnings) / CAST(previous_data.earnings AS double) * 100 AS percentage_change
```

SQL Ln 7, Col 41

Completed Time in queue: 127 ms Run time: 928 ms Data scanned: 2.62 KB

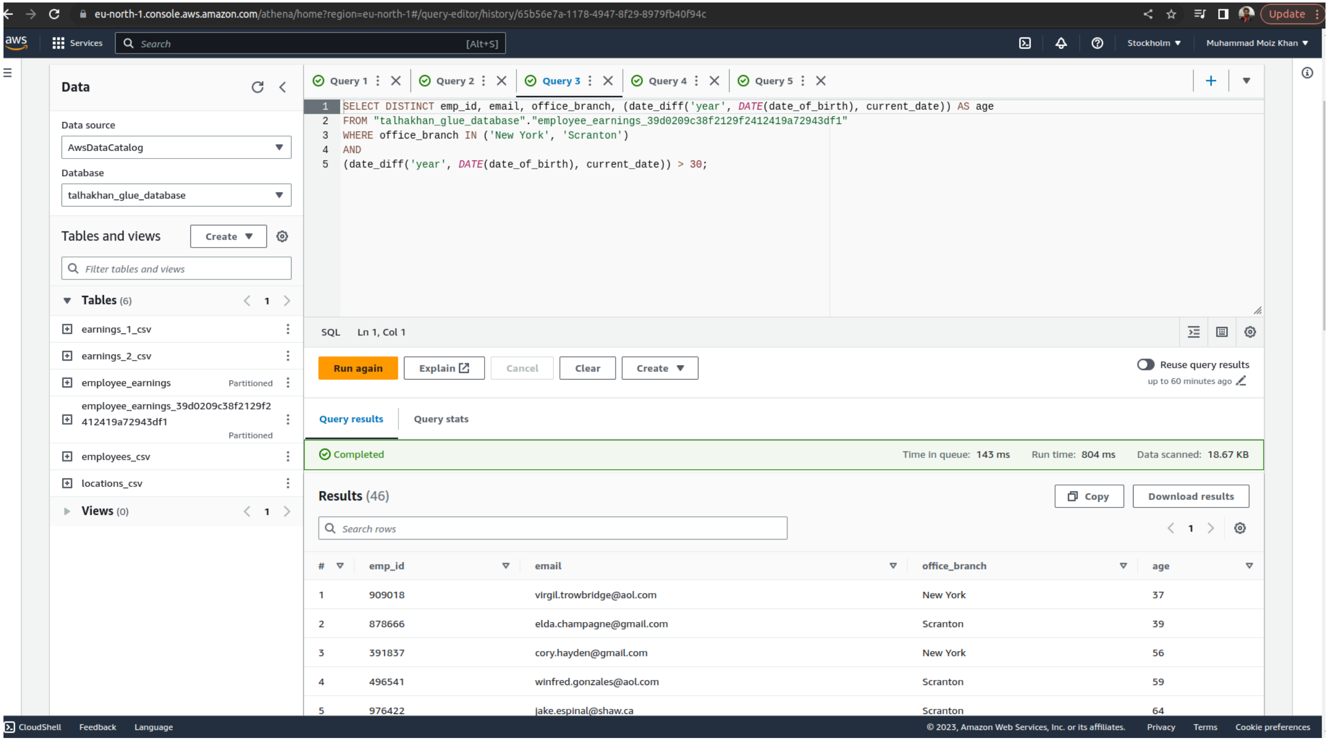
Results (100)

#	emp_id	current_earnings	previous_earnings	percentage_change
1	526540	2716	2843	-4.467112205416813
2	859327	8357	7280	14.793956043956044



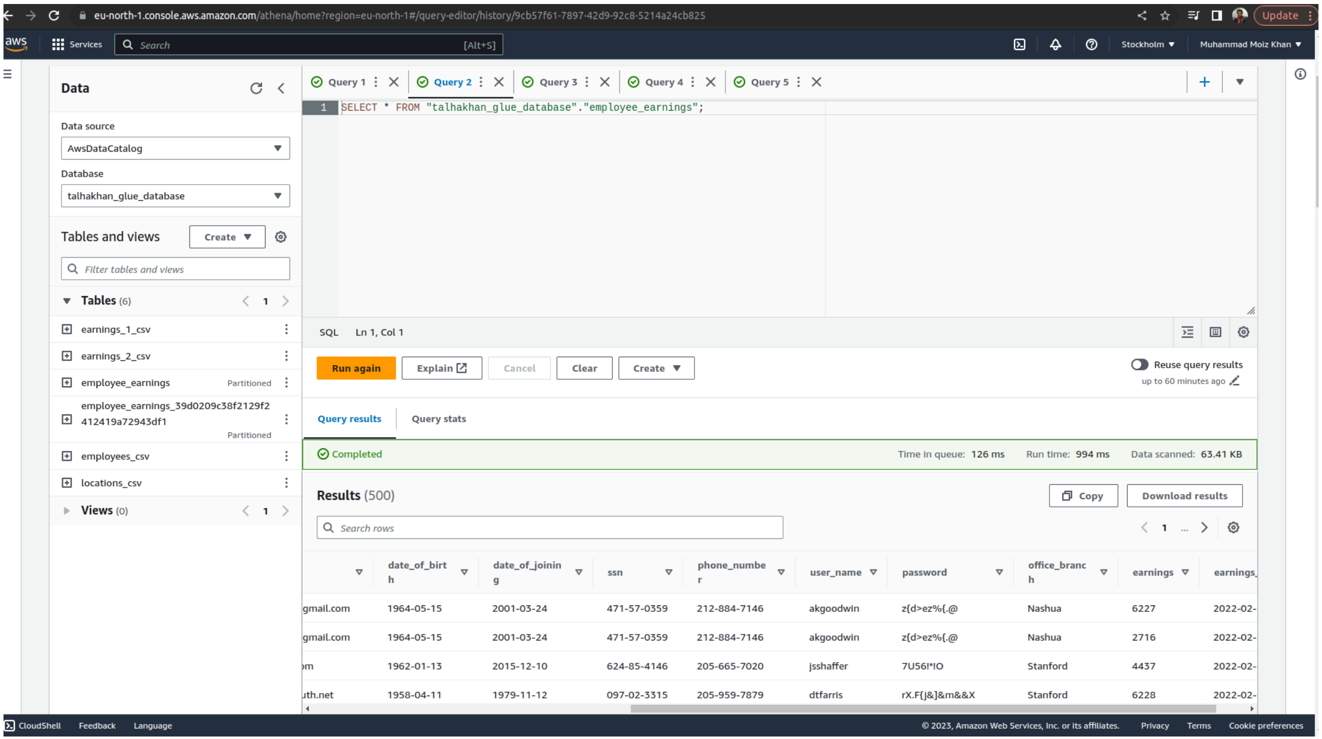
```sql
SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range
FROM (
SELECT office_branch as ob, AVG(earnings) AS value FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1" GROUP BY
    office_branch, earnings_date
) avg_earnings, "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
WHERE office_branch = avg_earnings.ob
GROUP BY office_branch;
```

#	office_branch	earnings_range
1	Nashua	479.9354838709678
2	Scranton	1779.2800000000007
3	Stanford	1053.375
4	New York	1015.75



```sql
SELECT DISTINCT emp_id, email, office_branch, (date_diff('year', DATE(date_of_birth), current_date)) AS age
FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
WHERE office_branch IN ('New York', 'Scranton')
AND
(date_diff('year', DATE(date_of_birth), current_date)) > 30;
```

#	emp_id	email	office_branch	age
1	909018	virgil.trowbridge@aol.com	New York	37
2	878666	elda.champagne@gmail.com	Scranton	39
3	391837	cory.hayden@gmail.com	New York	56
4	496541	winfred.gonzales@aol.com	Scranton	59
5	976422	jake.espinal@shaw.ca	Scranton	64



eu-north-1.console.aws.amazon.com/athena/home?region=eu-north-1#/query-editor/history/9cb57f61-7897-42d9-92c8-5214a24cb825

Data source: AwsDataCatalog

Database: talhakhan_glue_database

Tables (6)

- earnings_1_csv
- earnings_2_csv
- employee_earnings (Partitioned)
- employee_earnings_39d0209c38f2129f2412419a72943df1 (Partitioned)
- employees_csv
- locations_csv

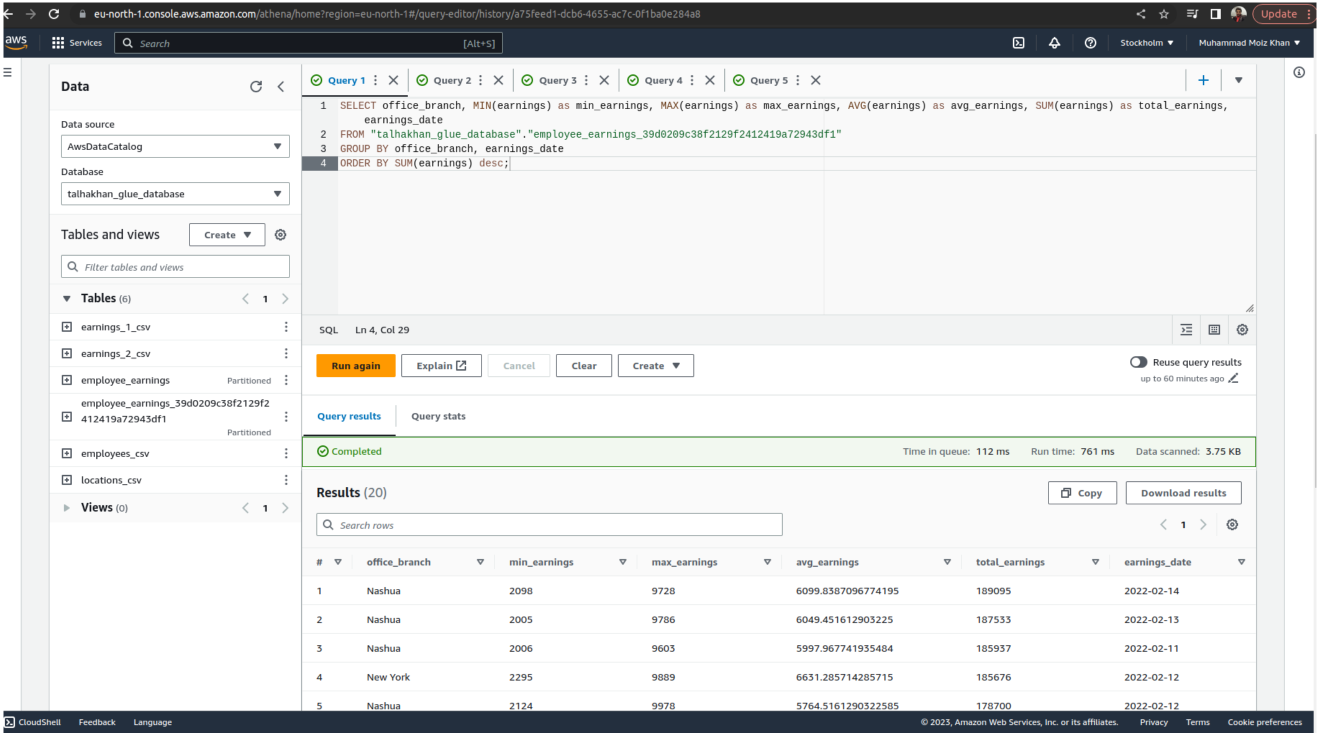
Views (0)

```
SELECT * FROM "talhakhan_glue_database"."employee_earnings";
```

Completed — Time in queue: 126 ms — Run time: 994 ms — Data scanned: 63.41 KB

Results (500)

	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	office_branch	earnings	earnings_
gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{.@	Nashua	6227	2022-02-
gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{.@	Nashua	2716	2022-02-
om	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!*IO	Stanford	4437	2022-02-
uth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dtfarris	rX.F[j&]&m&&X	Stanford	6228	2022-02-



eu-north-1.console.aws.amazon.com/athena/home?region=eu-north-1#/query-editor/history/a75feed1-dcb6-4655-ac7c-0f1ba0e284a8

Data source: AwsDataCatalog

Database: talhakhan_glue_database

Tables (6)

- earnings_1_csv
- earnings_2_csv
- employee_earnings (Partitioned)
- employee_earnings_39d0209c38f2129f2412419a72943df1 (Partitioned)
- employees_csv
- locations_csv

Views (0)

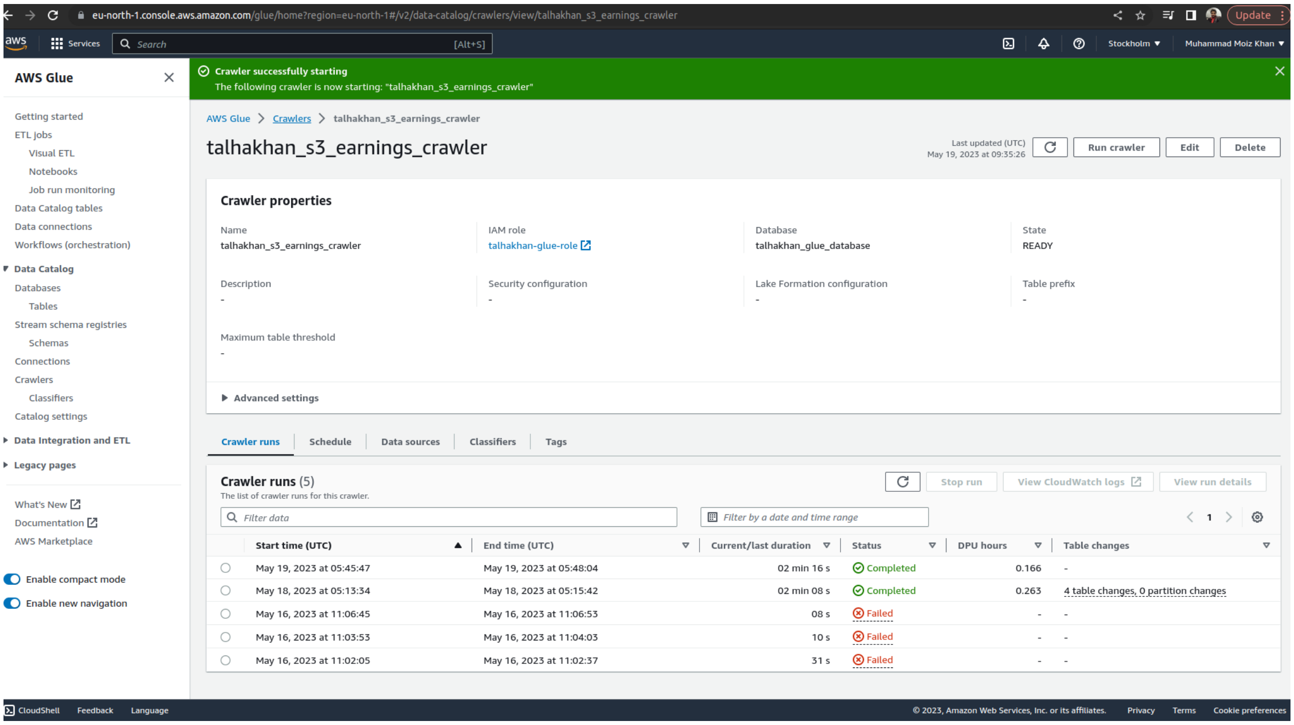
```
SELECT office_branch, MIN(earnings) as min_earnings, MAX(earnings) as max_earnings, AVG(earnings) as avg_earnings, SUM(earnings) as total_earnings,
    earnings_date
FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
GROUP BY office_branch, earnings_date
ORDER BY SUM(earnings) desc;
```

Completed — Time in queue: 112 ms — Run time: 761 ms — Data scanned: 3.75 KB

Results (20)

#	office_branch	min_earnings	max_earnings	avg_earnings	total_earnings	earnings_date
1	Nashua	2098	9728	6099.8387096774195	189095	2022-02-14
2	Nashua	2005	9786	6049.451612903225	187533	2022-02-13
3	Nashua	2006	9603	5997.967741935484	185937	2022-02-11
4	New York	2295	9889	6631.285714285715	185676	2022-02-12
5	Nashua	2124	9978	5764.5161290322585	178700	2022-02-12

S3 Earnings Crawler



Crawler successfully starting
The following crawler is now starting: "talhakhan_s3_earnings_crawler"

talhakhan_s3_earnings_crawler

Last updated (UTC)
May 19, 2023 at 09:35:26

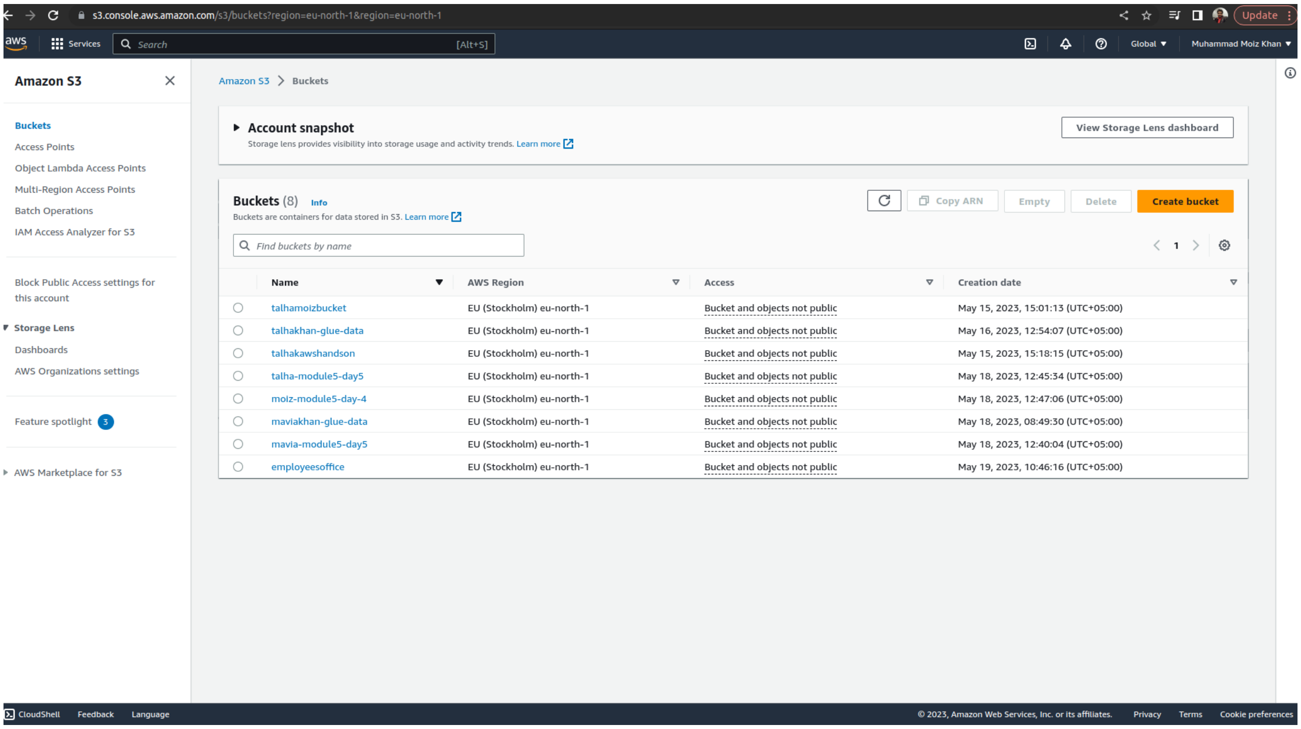
Crawler properties

Name talhakhan_s3_earnings_crawler	IAM role talhakhan-glue-role	Database talhakhan_glue_database	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			

Crawler runs (5)
The list of crawler runs for this crawler.

	Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
○	May 19, 2023 at 05:45:47	May 19, 2023 at 05:48:04	02 min 16 s	Completed	0.166	-
○	May 18, 2023 at 05:13:34	May 18, 2023 at 05:15:42	02 min 08 s	Completed	0.263	4 table changes, 0 partition changes
○	May 16, 2023 at 11:06:45	May 16, 2023 at 11:06:53	08 s	Failed	-	-
○	May 16, 2023 at 11:03:53	May 16, 2023 at 11:04:03	10 s	Failed	-	-
○	May 16, 2023 at 11:02:05	May 16, 2023 at 11:02:37	31 s	Failed	-	-

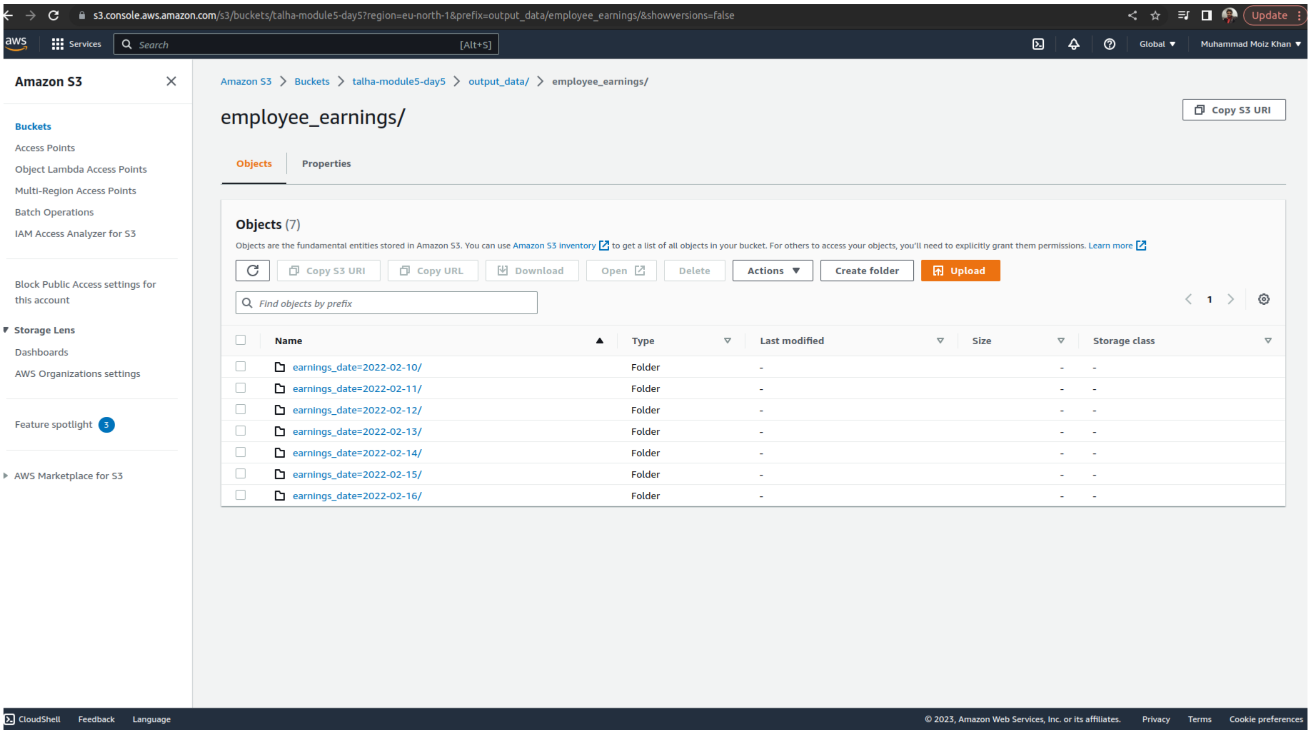
S3 Buckets


s3.console.aws.amazon.com/s3/buckets?region=eu-north-1&region=eu-north-1
Update
Services
Search
[Alt+S]
Global ▼
Muhammad Moiz Khan ▼
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (8) Info
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	talhamoizbucket	EU (Stockholm) eu-north-1	Bucket and objects not public	May 15, 2023, 15:01:13 (UTC+05:00)
○	talhakhan-glue-data	EU (Stockholm) eu-north-1	Bucket and objects not public	May 16, 2023, 12:54:07 (UTC+05:00)
○	talhakawshandson	EU (Stockholm) eu-north-1	Bucket and objects not public	May 15, 2023, 15:18:15 (UTC+05:00)
○	talha-module5-day5	EU (Stockholm) eu-north-1	Bucket and objects not public	May 18, 2023, 12:45:34 (UTC+05:00)
○	moiz-module5-day-4	EU (Stockholm) eu-north-1	Bucket and objects not public	May 18, 2023, 12:47:06 (UTC+05:00)
○	maviakhan-glue-data	EU (Stockholm) eu-north-1	Bucket and objects not public	May 18, 2023, 08:49:30 (UTC+05:00)
○	mavia-module5-day5	EU (Stockholm) eu-north-1	Bucket and objects not public	May 18, 2023, 12:40:04 (UTC+05:00)
○	employeesoffice	EU (Stockholm) eu-north-1	Bucket and objects not public	May 19, 2023, 10:46:16 (UTC+05:00)



s3.console.aws.amazon.com/s3/buckets/talha-module5-day5?region=eu-north-1&prefix=output_data/employee_earnings/&showversions=false
Update
aws
Services
Search
[Alt+S]
Global ▼
Muhammad Moiz Khan ▼
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets > talha-module5-day5 > output_data/ > employee_earnings/
Copy S3 URI
Objects
Properties
Copy S3 URI
Copy URL
Download
Open
Delete
Actions ▼
Create folder
Upload
Find objects by prefix
1
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

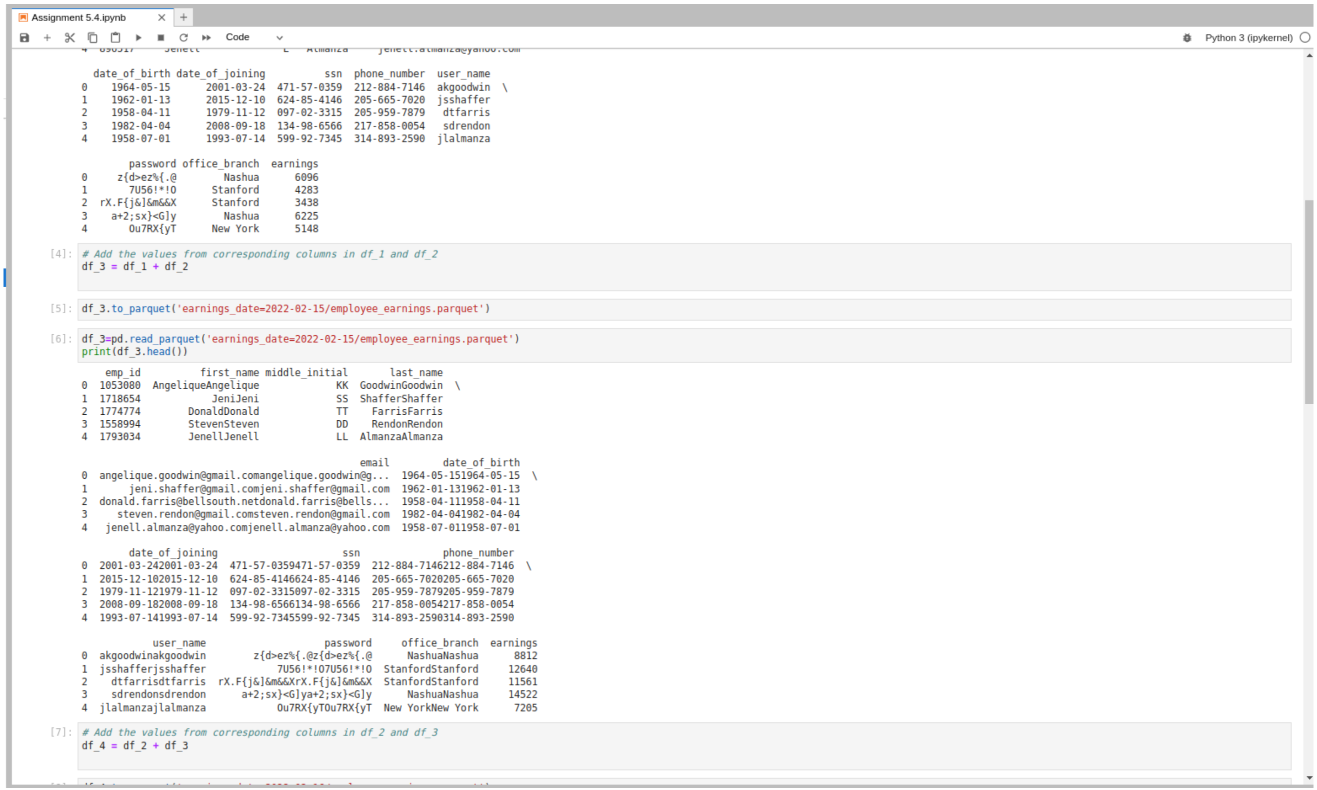
employee_earnings/

Objects (7)

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

□	Name	Type	Last modified	Size	Storage class
□	earnings_date=2022-02-10/	Folder	-	-	-
□	earnings_date=2022-02-11/	Folder	-	-	-
□	earnings_date=2022-02-12/	Folder	-	-	-
□	earnings_date=2022-02-13/	Folder	-	-	-
□	earnings_date=2022-02-14/	Folder	-	-	-
□	earnings_date=2022-02-15/	Folder	-	-	-
□	earnings_date=2022-02-16/	Folder	-	-	-

Using the data manipulation tool of your choice (eg. Python) simulate the earnings predictions for 2 more days. Load it to the Data Lake that you've created today (Task 1-2).



```
  date_of_birth date_of_joining          ssn  phone_number  user_name
0    1964-05-15      2001-03-24  471-57-0359  212-884-7146  akgoodwin  \
1    1962-01-13      2015-12-10  624-85-4146  205-665-7020  jsshaffer
2    1958-04-11      1979-11-12  097-02-3315  205-959-7879   dtfarris
3    1982-04-04      2008-09-18  134-98-6566  217-858-0054   sdrendon
4    1958-07-01      1993-07-14  599-92-7345  314-893-2590  jlalmanza

        password office_branch  earnings
0     z{d>ez%{.@        Nashua      6096
1       7U56!*!O      Stanford      4283
2  rX.F{j&]&m&&X      Stanford      3438
3    a+2;sx}<G]y        Nashua      6225
4       Ou7RX{yT      New York      5148
```

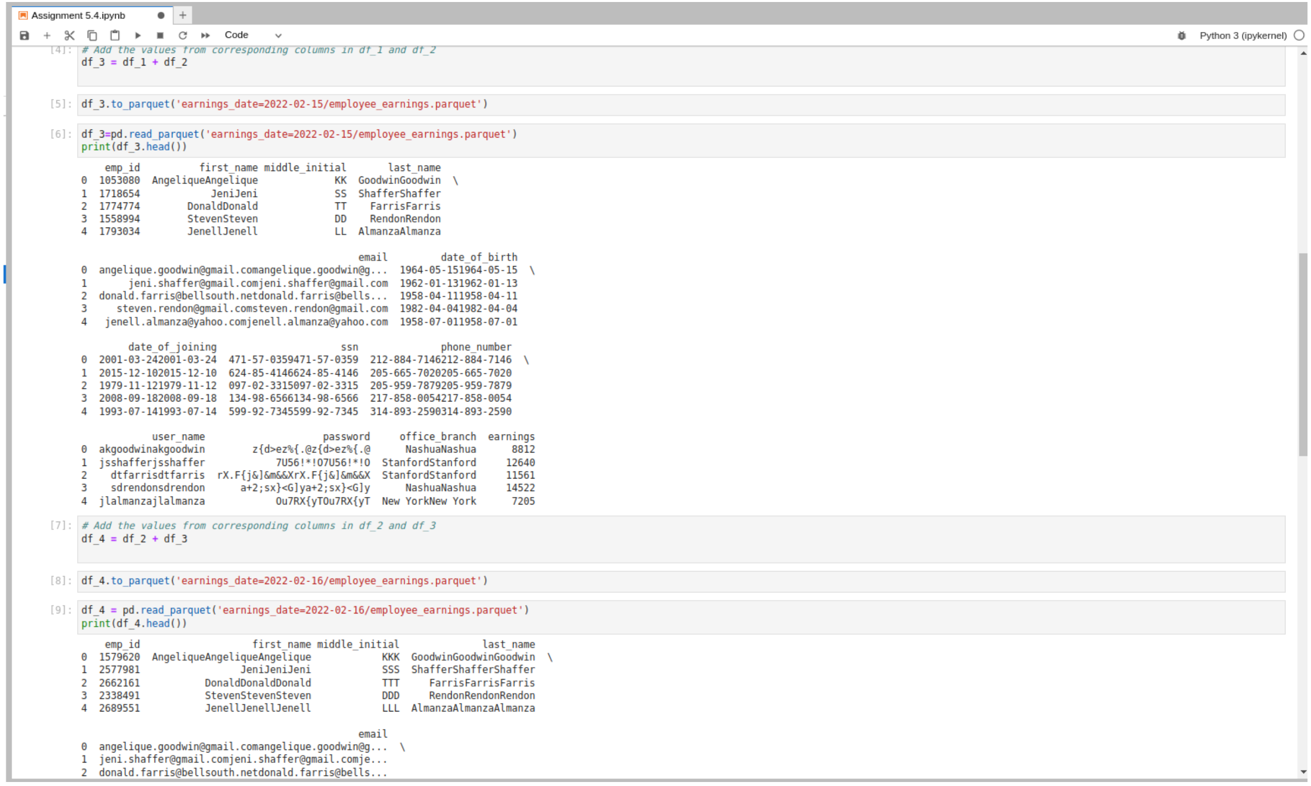
```
[4]: # Add the values from corresponding columns in df_1 and df_2
     df_3 = df_1 + df_2
```

```
[5]: df_3.to_parquet('earnings_date=2022-02-15/employee_earnings.parquet')
```

```
[6]: df_3=pd.read_parquet('earnings_date=2022-02-15/employee_earnings.parquet')
     print(df_3.head())
```

```
    emp_id          first_name middle_initial       last_name
0  1053080  AngeliqueAngelique             KK  GoodwinGoodwin  \
1  1718654            JeniJeni             SS  ShafferShaffer
2  1774774        DonaldDonald             TT    FarrisFarris
3  1558994        StevenSteven             DD    RendonRendon
4  1793034        JenellJenell             LL  AlmanzaAlmanza

                                             email         date_of_birth
0  angelique.goodwin@gmail.comangelique.goodwin@g...  1964-05-151964-05-15  \
1      jeni.shaffer@gmail.comjeni.shaffer@gmail.com  1962-01-131962-01-13
2  donald.farris@bellsouth.netdonald.farris@bells...  1958-04-111958-04-11
3    steven.rendon@gmail.comsteven.rendon@gmail.com  1982-04-041982-04-04
4   jenell.almanza@yahoo.comjenell.almanza@yahoo.com  1958-07-011958-07-01

       date_of_joining                     ssn            phone_number
0  2001-03-242001-03-24  471-57-0359471-57-0359  212-884-7146212-884-7146  \
1  2015-12-102015-12-10  624-85-4146624-85-4146  205-665-7020205-665-7020
2  1979-11-121979-11-12  097-02-3315097-02-3315  205-959-7879205-959-7879
3  2008-09-182008-09-18  134-98-6566134-98-6566  217-858-0054217-858-0054
4  1993-07-141993-07-14  599-92-7345599-92-7345  314-893-2590314-893-2590

           user_name                  password     office_branch  earnings
0  akgoodwinakgoodwin      z{d>ez%{.@z{d>ez%{.@      NashuaNashua      8812
1  jsshafferjsshaffer          7U56!*!O7U56!*!O  StanfordStanford     12640
2    dtfarrisdtfarris  rX.F{j&]&m&&XrX.F{j&]&m&&X  StanfordStanford     11561
3    sdrendonsdrendon    a+2;sx}<G]ya+2;sx}<G]y      NashuaNashua     14522
4  jlalmanzajlalmanza          Ou7RX{yTOu7RX{yT  New YorkNew York      7205
```

```
[7]: # Add the values from corresponding columns in df_2 and df_3
     df_4 = df_2 + df_3
```



```
# Add the values from corresponding columns in df_1 and df_2
df_3 = df_1 + df_2
```

```
[5]: df_3.to_parquet('earnings_date=2022-02-15/employee_earnings.parquet')
```

```
[6]: df_3=pd.read_parquet('earnings_date=2022-02-15/employee_earnings.parquet')
     print(df_3.head())
```

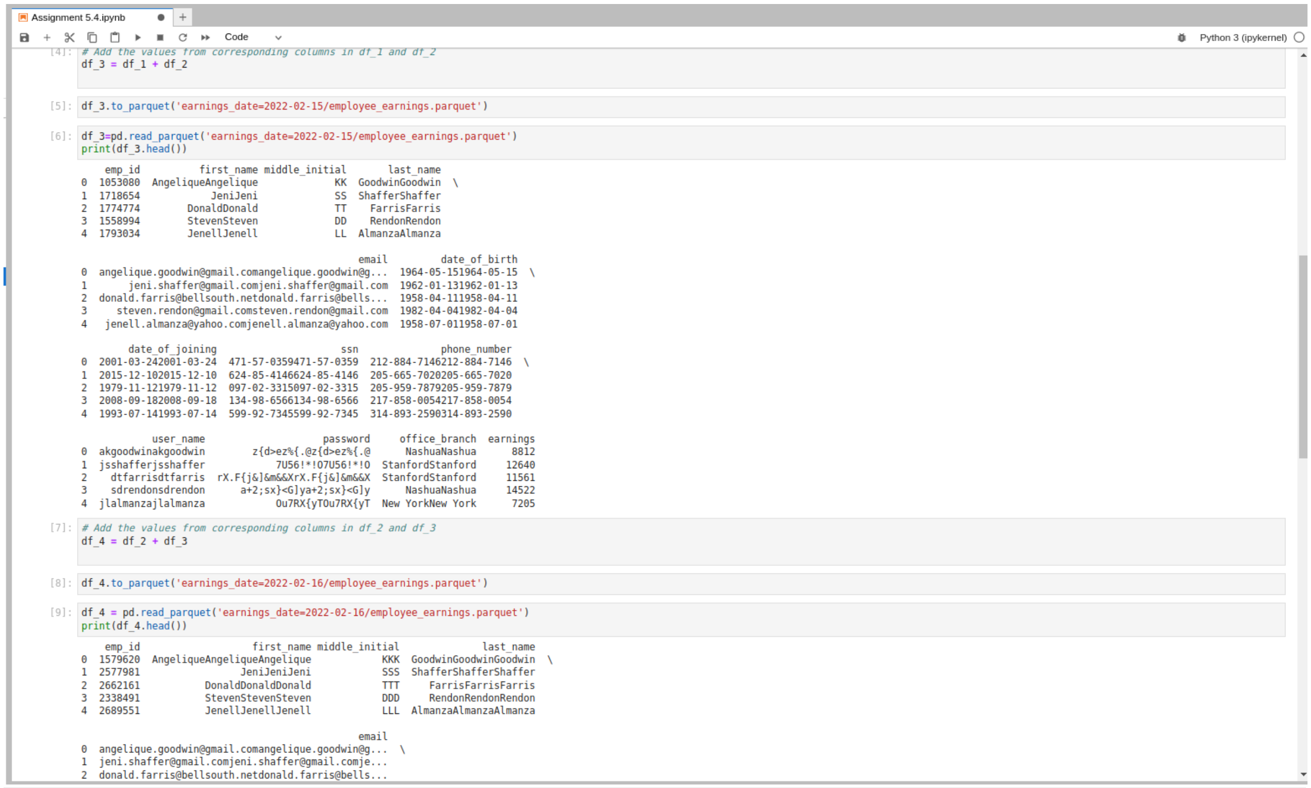
```
    emp_id          first_name middle_initial       last_name
0  1053080  AngeliqueAngelique             KK  GoodwinGoodwin  \
1  1718654            JeniJeni             SS  ShafferShaffer
2  1774774        DonaldDonald             TT    FarrisFarris
3  1558994        StevenSteven             DD    RendonRendon
4  1793034        JenellJenell             LL  AlmanzaAlmanza

                                             email         date_of_birth
0  angelique.goodwin@gmail.comangelique.goodwin@g...  1964-05-151964-05-15  \
1      jeni.shaffer@gmail.comjeni.shaffer@gmail.com  1962-01-131962-01-13
2  donald.farris@bellsouth.netdonald.farris@bells...  1958-04-111958-04-11
3    steven.rendon@gmail.comsteven.rendon@gmail.com  1982-04-041982-04-04
4   jenell.almanza@yahoo.comjenell.almanza@yahoo.com  1958-07-011958-07-01

       date_of_joining                     ssn            phone_number
0  2001-03-242001-03-24  471-57-0359471-57-0359  212-884-7146212-884-7146  \
1  2015-12-102015-12-10  624-85-4146624-85-4146  205-665-7020205-665-7020
2  1979-11-121979-11-12  097-02-3315097-02-3315  205-959-7879205-959-7879
3  2008-09-182008-09-18  134-98-6566134-98-6566  217-858-0054217-858-0054
4  1993-07-141993-07-14  599-92-7345599-92-7345  314-893-2590314-893-2590

           user_name                  password     office_branch  earnings
0  akgoodwinakgoodwin      z{d>ez%{.@z{d>ez%{.@      NashuaNashua      8812
1  jsshafferjsshaffer          7U56!*!O7U56!*!O  StanfordStanford     12640
2    dtfarrisdtfarris  rX.F{j&]&m&&XrX.F{j&]&m&&X  StanfordStanford     11561
3    sdrendonsdrendon    a+2;sx}<G]ya+2;sx}<G]y      NashuaNashua     14522
4  jlalmanzajlalmanza          Ou7RX{yTOu7RX{yT  New YorkNew York      7205
```

```
[7]: # Add the values from corresponding columns in df_2 and df_3
     df_4 = df_2 + df_3
```

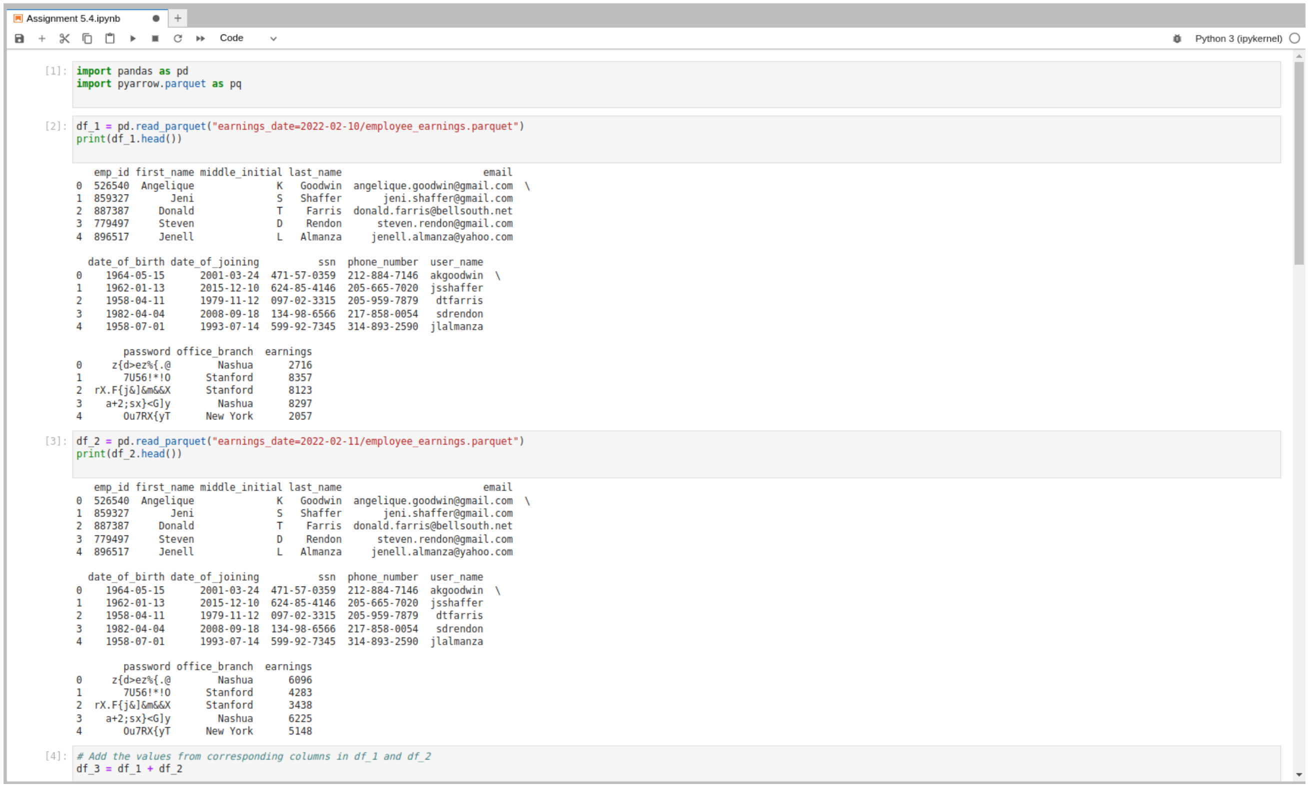
```
[8]: df_4.to_parquet('earnings_date=2022-02-16/employee_earnings.parquet')
```

```
[9]: df_4 = pd.read_parquet('earnings_date=2022-02-16/employee_earnings.parquet')
     print(df_4.head())
```

```
    emp_id                first_name middle_initial              last_name
0  1579620  AngeliqueAngeliqueAngelique            KKK  GoodwinGoodwinGoodwin  \
1  2577981              JeniJeniJeni            SSS  ShafferShafferShaffer
2  2662161        DonaldDonaldDonald            TTT     FarrisFarrisFarris
3  2338491        StevenStevenSteven            DDD     RendonRendonRendon
4  2689551        JenellJenellJenell            LLL  AlmanzaAlmanzaAlmanza

                                              email
0  angelique.goodwin@gmail.comangelique.goodwin@g...  \
1  jeni.shaffer@gmail.comjeni.shaffer@gmail.comje...
2  donald.farris@bellsouth.netdonald.farris@bells...
```

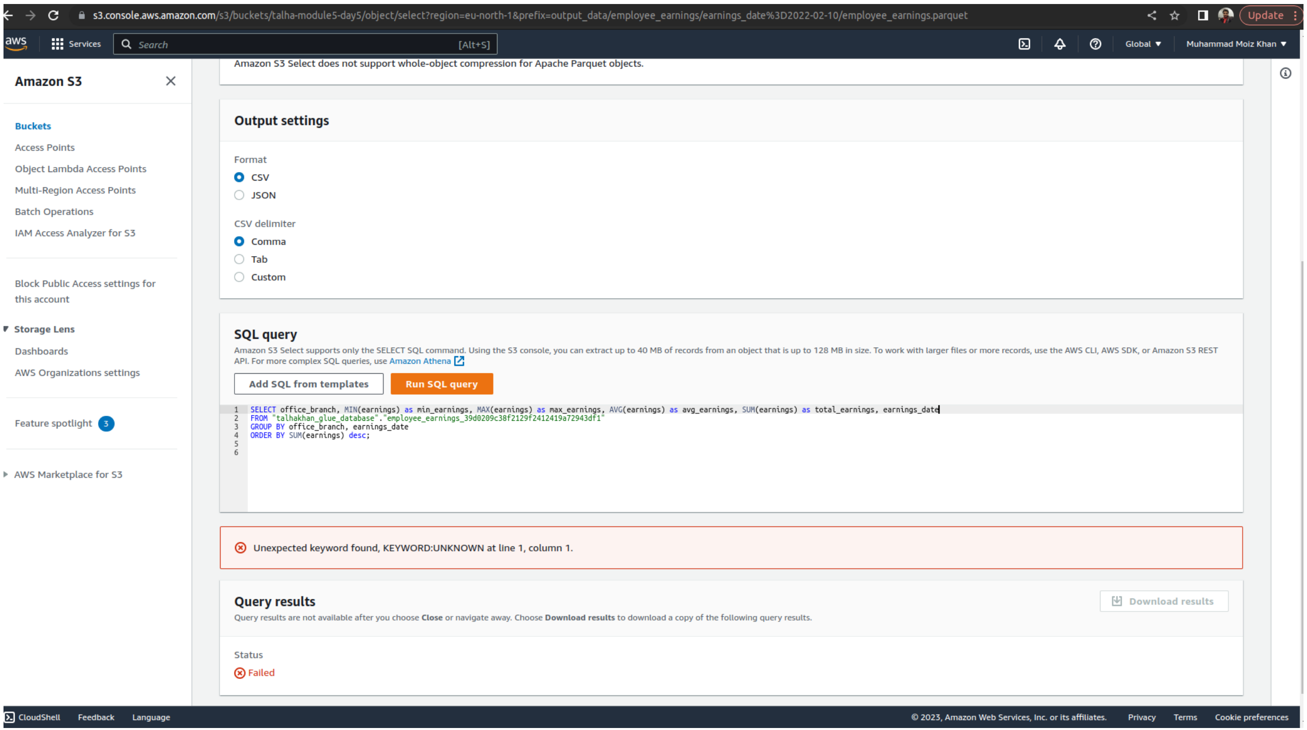


```
[4]: # Add the values from corresponding columns in df_1 and df_2
     df_3 = df_1 + df_2

[5]: df_3.to_parquet('earnings_date=2022-02-15/employee_earnings.parquet')

[6]: df_3=pd.read_parquet('earnings_date=2022-02-15/employee_earnings.parquet')
     print(df_3.head())

    emp_id          first_name middle_initial       last_name  
0  1053080  AngeliqueAngelique             KK  GoodwinGoodwin  \
1  1718654            JeniJeni             SS  ShafferShaffer   
2  1774774        DonaldDonald             TT    FarrisFarris   
3  1558994        StevenSteven             DD    RendonRendon   
4  1793034        JenellJenell             LL  AlmanzaAlmanza   

                                               email         date_of_birth  
0  angelique.goodwin@gmail.comangelique.goodwin@g...  1964-05-151964-05-15  \
1       jeni.shaffer@gmail.comjeni.shaffer@gmail.com  1962-01-131962-01-13   
2  donald.farris@bellsouth.netdonald.farris@bells...  1958-04-111958-04-11   
3     steven.rendon@gmail.comsteven.rendon@gmail.com  1982-04-041982-04-04   
4   jenell.almanza@yahoo.comjenell.almanza@yahoo.com  1958-07-011958-07-01   

         date_of_joining                     ssn          phone_number  
0  2001-03-242001-03-24  471-57-0359471-57-0359  212-884-7146212-884-7146  \
1  2015-12-102015-12-10  624-85-4146624-85-4146  205-665-7020205-665-7020   
2  1979-11-121979-11-12  097-02-3315097-02-3315  205-959-7879205-959-7879   
3  2008-09-182008-09-18  134-98-6566134-98-6566  217-858-0054217-858-0054   
4  1993-07-141993-07-14  599-92-7345599-92-7345  314-893-2590314-893-2590   

            user_name                password     office_branch  earnings  
0  akgoodwinakgoodwin      z{d>ez%{.@z{d>ez%{.@      NashuaNashua      8812  
1  jsshafferjsshaffer          7U56!*!O7U56!*!O  StanfordStanford     12640  
2    dtfarrisdtfarris  rX.F{j&]&m&&XrX.F{j&]&m&&X  StanfordStanford     11561  
3    sdrendonsdrendon      a+2;sx}<G]ya+2;sx}<G]y      NashuaNashua     14522  
4  jlalmanzajlalmanza          Ou7RX{yTOu7RX{yT  New YorkNew York      7205  

[7]: # Add the values from corresponding columns in df_2 and df_3
     df_4 = df_2 + df_3

[8]: df_4.to_parquet('earnings_date=2022-02-16/employee_earnings.parquet')

[9]: df_4 = pd.read_parquet('earnings_date=2022-02-16/employee_earnings.parquet')
     print(df_4.head())

    emp_id                     first_name middle_initial              last_name  
0  1579620  AngeliqueAngeliqueAngelique            KKK  GoodwinGoodwinGoodwin  \
1  2577981                 JeniJeniJeni            SSS  ShafferShafferShaffer   
2  2662161           DonaldDonaldDonald            TTT     FarrisFarrisFarris   
3  2338491           StevenStevenSteven            DDD     RendonRendonRendon   
4  2689551           JenellJenellJenell            LLL  AlmanzaAlmanzaAlmanza   

                                               email  
0  angelique.goodwin@gmail.comangelique.goodwin@g...  \
1  jeni.shaffer@gmail.comjeni.shaffer@gmail.comje...   
2  donald.farris@bellsouth.netdonald.farris@bells...   
```



```
[1]: import pandas as pd
     import pyarrow.parquet as pq

[2]: df_1 = pd.read_parquet("earnings_date=2022-02-10/employee_earnings.parquet")
     print(df_1.head())

   emp_id first_name middle_initial last_name                        email  
0  526540  Angelique              K   Goodwin  angelique.goodwin@gmail.com  \
1  859327       Jeni              S   Shaffer       jeni.shaffer@gmail.com   
2  887387     Donald              T    Farris  donald.farris@bellsouth.net   
3  779497     Steven              D    Rendon      steven.rendon@gmail.com   
4  896517     Jenell              L   Almanza     jenell.almanza@yahoo.com   

  date_of_birth date_of_joining          ssn  phone_number  user_name  
0    1964-05-15      2001-03-24  471-57-0359  212-884-7146  akgoodwin  \
1    1962-01-13      2015-12-10  624-85-4146  205-665-7020  jsshaffer   
2    1958-04-11      1979-11-12  097-02-3315  205-959-7879   dtfarris   
3    1982-04-04      2008-09-18  134-98-6566  217-858-0054   sdrendon   
4    1958-07-01      1993-07-14  599-92-7345  314-893-2590  jlalmanza   

        password office_branch  earnings  
0     z{d>ez%{.@        Nashua      2716  
1       7U56!*!O      Stanford      8357  
2  rX.F{j&]&m&&X      Stanford      8123  
3    a+2;sx}<G]y        Nashua      8297  
4       Ou7RX{yT      New York      2057  

[3]: df_2 = pd.read_parquet("earnings_date=2022-02-11/employee_earnings.parquet")
     print(df_2.head())

   emp_id first_name middle_initial last_name                        email  
0  526540  Angelique              K   Goodwin  angelique.goodwin@gmail.com  \
1  859327       Jeni              S   Shaffer       jeni.shaffer@gmail.com   
2  887387     Donald              T    Farris  donald.farris@bellsouth.net   
3  779497     Steven              D    Rendon      steven.rendon@gmail.com   
4  896517     Jenell              L   Almanza     jenell.almanza@yahoo.com   

  date_of_birth date_of_joining          ssn  phone_number  user_name  
0    1964-05-15      2001-03-24  471-57-0359  212-884-7146  akgoodwin  \
1    1962-01-13      2015-12-10  624-85-4146  205-665-7020  jsshaffer   
2    1958-04-11      1979-11-12  097-02-3315  205-959-7879   dtfarris   
3    1982-04-04      2008-09-18  134-98-6566  217-858-0054   sdrendon   
4    1958-07-01      1993-07-14  599-92-7345  314-893-2590  jlalmanza   

        password office_branch  earnings  
0     z{d>ez%{.@        Nashua      6096  
1       7U56!*!O      Stanford      4283  
2  rX.F{j&]&m&&X      Stanford      3438  
3    a+2;sx}<G]y        Nashua      6225  
4       Ou7RX{yT      New York      5148  

[4]: # Add the values from corresponding columns in df_1 and df_2
     df_3 = df_1 + df_2
```

Rerun queries from Task 3 and Task 4 and see how the results change with this new data.


s3.console.aws.amazon.com/s3/buckets/talha-module5-day5/object/select?region=eu-north-1&prefix=output_data/employee_earnings/earnings_date%3D2022-02-10/employee_earnings.parquet
Update
aws
Services
Search
[Alt+S]
Global
Muhammad Moiz Khan
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3

Amazon S3 Select does not support whole-object compression for Apache Parquet objects.

Output settings

Format

● CSV

○ JSON

CSV delimiter

● Comma

○ Tab

○ Custom

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```sql
SELECT office_branch, MIN(earnings) as min_earnings, MAX(earnings) as max_earnings, AVG(earnings) as avg_earnings, SUM(earnings) as total_earnings, earnings_date
FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
GROUP BY office_branch, earnings_date
ORDER BY SUM(earnings) desc;


```

Ⓧ Unexpected keyword found, KEYWORD:UNKNOWN at line 1, column 1.

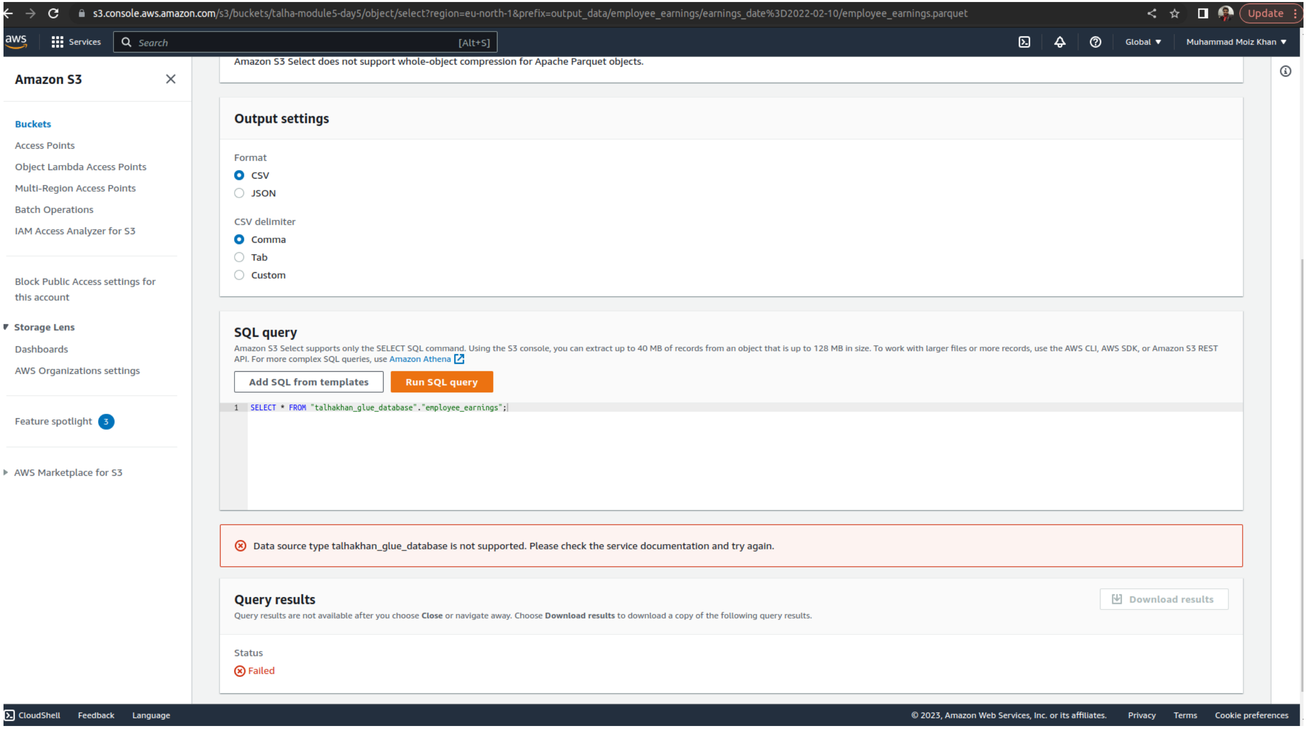
Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Ⓧ Failed

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


s3.console.aws.amazon.com/s3/buckets/talha-module5-day5/object/select?region=eu-north-1&prefix=output_data/employee_earnings/earnings_date%3D2022-02-10/employee_earnings.parquet
Update
aws
Services
Search
[Alt+S]
Global
Muhammad Moiz Khan
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3

Amazon S3 Select does not support whole-object compression for Apache Parquet objects.

Output settings

Format

● CSV

○ JSON

CSV delimiter

● Comma

○ Tab

○ Custom

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```sql
SELECT * FROM "talhakhan_glue_database"."employee_earnings";
```

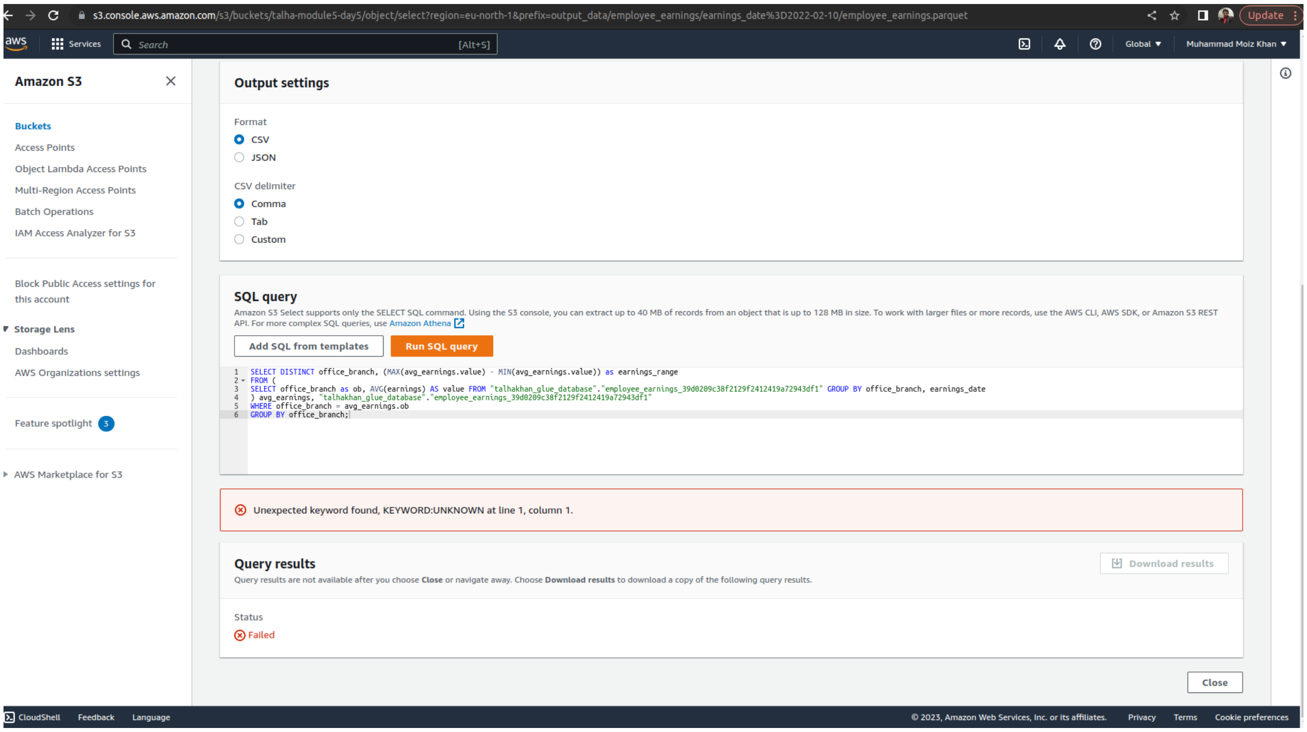
Ⓧ Data source type talhakhan_glue_database is not supported. Please check the service documentation and try again.

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Ⓧ Failed


s3.console.aws.amazon.com/s3/buckets/talha-module5-day5/object/select?region=eu-north-1&prefix=output_data/employee_earnings/earnings_date%3D2022-02-10/employee_earnings.parquet
Update
aws
Services
Search
[Alt+S]
Global
Muhammad Moiz Khan
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
3
AWS Marketplace for S3
Add SQL from templates
Run SQL query
Download results
Close
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Output settings

Format

● CSV
○ JSON

CSV delimiter

● Comma
○ Tab
○ Custom

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range
FROM (
SELECT office_branch as ob, AVG(earnings) AS value FROM "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1" GROUP BY office_branch, earnings_date
) avg_earnings, "talhakhan_glue_database"."employee_earnings_39d0209c38f2129f2412419a72943df1"
WHERE office_branch = avg_earnings.ob
GROUP BY office_branch;
```

Unexpected keyword found, KEYWORD:UNKNOWN at line 1, column 1.

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Failed