

**NAME: MAVIA ALAM KHAN (2303.KHI.DEG.017)**  
**PAIRING WITH : Mohammad Hussam (2303.KHI.DEG.020)**  
**ASSIGNMENT NO : 5.4**

Here we run the task 3 queries

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings

Workgroup: primary

Data

Data source: AwsDataCatalog

Database: mavia-glue-database

Tables and views

Filter tables and views

Tables (8)

- alam\_employee\_earnings
- mavia\_earnings\_1\_csv

Query 1: X Query 2: X Query 3: X

```
1 -- SELECT * FROM "mavia-glue-database"."alam_employee_earnings" limit 10;
2
3 SELECT DISTINCT emp_id, email, office_branch, (date_diff('year', DATE(date_of_birth), current_date)) AS age
4 FROM "mavia-glue-database"."alam_employee_earnings"
5 WHERE office_branch IN ('New York', 'Scranton')
6 AND
7 (date_diff('year', DATE(date_of_birth), current_date)) > 30;
```

SQL Ln 7, Col 61

Download results

Query results | Query stats

Completed Time in queue: 109 ms Run time: 924 ms Data scanned: 26.13 KB

Results (46)

Copy Download results

Search rows

#	emp_id	email	office_branch	age
1	900756	benjamin.doss@gmail.com	Scranton	38
2	654617	rogerio.woodall@gmail.com	New York	50
3	138911	claudio.heck@aol.com	Scranton	55
4	713294	sammy.dewitt@ibm.com	Scranton	35
5	215719	brent.carrillo@aol.com	New York	50
6	312726	celine.lumpkin@gmail.com	New York	36
7	530134	mathew.whitfield@gmail.com	New York	36

Tables and views

Filter tables and views

Tables (8)

- alam\_employee\_earnings
- mavia\_earnings\_1\_csv
- mavia\_earnings\_2\_csv
- mavia\_employee\_earnings
- mavia\_locations
- mavia\_locations\_csv
- mavia\_output\_data
- maviakhanearnings

Views (0)

Query 1: X Query 2: X Query 3: X

```
13 SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range
14 FROM (
15 SELECT office_branch as ob, AVG(earnings) AS value FROM "mavia-glue-database"."mavia_employee_earnings"
16 GROUP BY office_branch, earnings_date
17 ) avg_earnings, "mavia-glue-database"."alam_employee_earnings"
18 WHERE office_branch = avg_earnings.ob
19 GROUP BY office_branch;
```

SQL Ln 18, Col 38

Run again Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Query results | Query stats

Completed Time in queue: 151 ms Run time: 1.348 sec Data scanned: 4.69 KB

Results (4)

Copy Download results

Search rows

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

8  
9 SELECT office\_branch, MIN(earnings) as min\_earnings, MAX(earnings) as max\_earnings, AVG(earnings) as avg\_earnings, SUM(earnings) as total\_earnings  
10 , earnings\_date  
11 FROM "mavia-glue-database"."alam\_employee\_earnings"  
12 GROUP BY office\_branch, earnings\_date  
13 ORDER BY SUM(earnings) desc;

SQL Ln 12, Col 29

Run again Explain Cancel Clear Create

Query results Query stats

Completed Time in queue: 346 ms Run time: 791 ms Data scanned: 5.24 KB

Results (28)

Search rows

#	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	2017	9614	6008.967741935484	186278	2022-02-16
4	Nashua	2006	9603	5997.967741935484	185937	2022-02-11
5	New York	2295	9889	6631.285714285715	185676	2022-02-12
6	Nashua	2124	9978	5764.5161290322585	178700	2022-02-12
7	Nashua	2076	9811	5629.903225806452	174527	2022-02-15

We calculates the % change in earnings for every employee from a given day compared to the previous day.

a

source  
sDataCatalog  
base  
via-glue-database

les and views  
Create Filter tables and views

Tables (8)  
alam\_employee\_earnings Partitioned

1 - WITH earnings\_change AS (  
2 SELECT  
3 emp\_id,  
4 earnings,  
5 earnings\_date,  
6 LAG(earnings) OVER (PARTITION BY emp\_id ORDER BY earnings\_date) AS previous\_earnings  
7 FROM  
8 "mavia-glue-database"."alam\_employee\_earnings"  
9 )  
10 SELECT  
11 emp\_id,  
12 earnings\_date,  
13 earnings,  
14 previous\_earnings,  
15 CASE

SQL Ln 24, Col 25

Run again Explain Cancel Clear Create

Query results Query stats

Completed Time in queue: 142 ms Run time: 747 ms Data scanned: 9.16 KB

Results (100)

Search rows

#	emp_id	earnings_date	earnings	previous_earnings	percentage_change
1	138911	2022-02-16	2210	3826	-42.23732357555804
2	143711	2022-02-16	4431	2831	56.51713175556341
3	147133	2022-02-16	3422	5088	-32.7437106918239
4	149972	2022-02-16	7918	5353	47.91705585652905
5	155097	2022-02-16	2703	5957	-54.62481114655028

Data

Data source  
AwsDataCatalog  
Database  
mavia-glue-database

Tables and views  
Create Filter tables and views

Tables (8)  
alam\_employee\_earnings Partitioned  
mavia\_earnings\_1\_csv  
mavia\_earnings\_2\_csv  
mavia\_employee\_earnings Partitioned  
mavia\_locations  
mavia\_locations\_csv  
mavia\_output\_data Partitioned  
maviakhanearnings Partitioned

Views (0)

11 emp\_id,  
12 earnings\_date,  
13 earnings,  
14 previous\_earnings,  
15 CASE  
16 WHEN previous\_earnings IS NOT NULL THEN (earnings - previous\_earnings) / cast(previous\_earnings as double) \* 100  
17 ELSE NULL  
18 END AS percentage\_change  
19 FROM  
20 earnings\_change  
21 WHERE  
22 earnings\_date = '2022-02-16' -- Replace with your desired date  
23 ORDER BY  
24 emp\_id, earnings\_date;

SQL Ln 24, Col 25

Run again Explain Cancel Clear Create

Query results Query stats

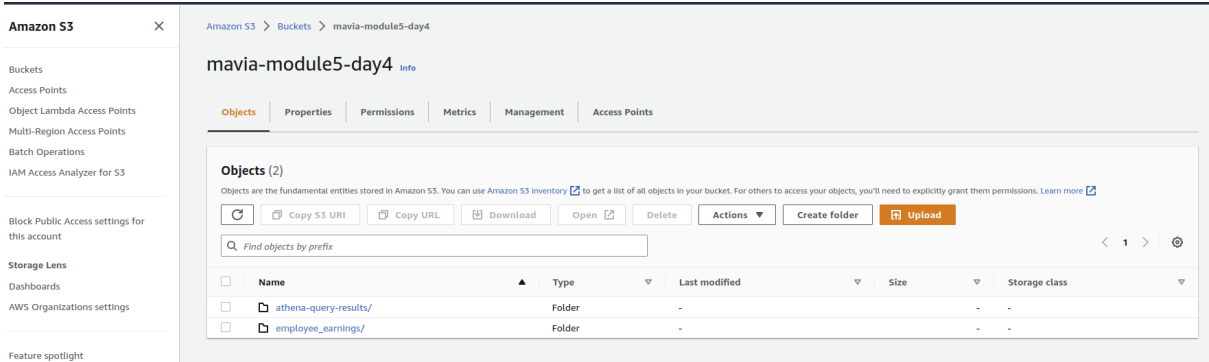
Completed Time in queue: 142 ms Run time: 747 ms Data scanned: 9.16 KB

Results (100)

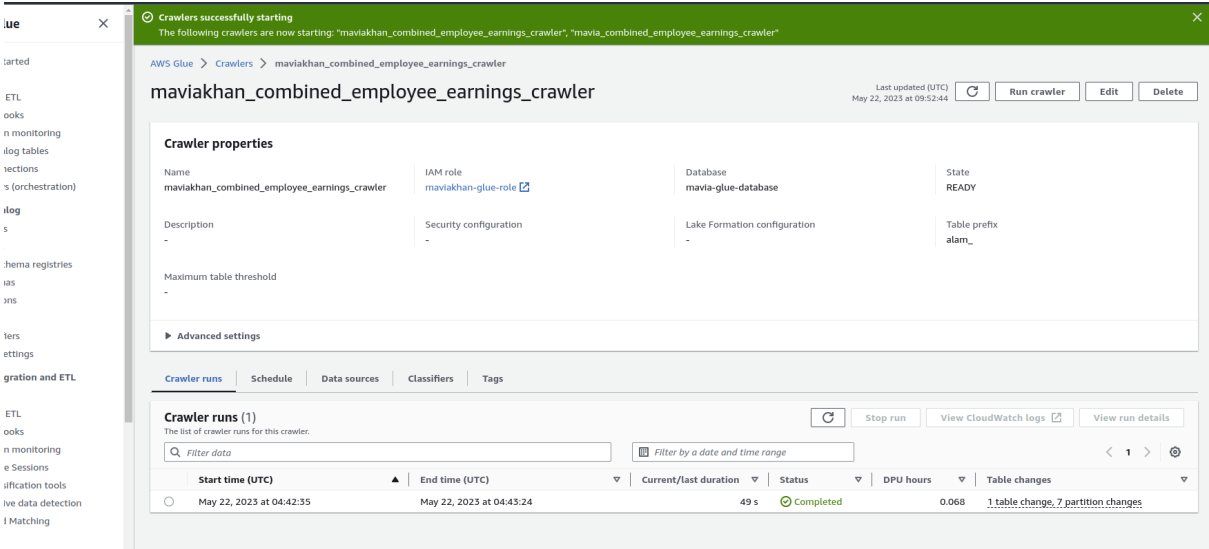
Search rows

#	emp_id	earnings_date	earnings	previous_earnings	percentage_change
1	138911	2022-02-16	2210	3826	-42.23732357555804
2	143711	2022-02-16	4431	2831	56.51713175556341
3	147133	2022-02-16	3422	5088	-32.7437106918239
4	149972	2022-02-16	7918	5353	47.91705585652905
5	155097	2022-02-16	2703	5957	-54.62481114655028

Here we create s3 bucket and create 2 folder one is employee earning and second is athena-query-result. In employee earning we stored a dataset



We create a crawler



## Using pandas we read a dataset

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[54]: import pandas as pd
import os

[55]: df=pd.read_parquet("output_data/employee_earnings/earnings_date=2022-02-10/employee_earnings.parquet")

[56]: df.head()
```

	emp_id	first_name	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	of
0	526540	Angelique	K	Goodwin	angelique.goodwin@gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{. @	
1	859327	Jeni	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!^O	
2	887387	Donald	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dtfarris	rX.F{j&j&m&&X	
3	779497	Steven	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2;sx<Gjy	
4	896517	Jenell	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	jialmanza	Ou7RX{yT	

```
[57]: df["earnings"]=df["earnings"]+10
```

Here we change the value of earning column

The screenshot shows the same Jupyter Notebook after running the code to update the earnings column. The code and the resulting DataFrame are as follows:

```
[57]: df["earnings"]=df["earnings"]+10

[58]: df
```

	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	office_branch	earnings
	K	Goodwin	angelique.goodwin@gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{. @	Nashua	6237
	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!^O	Stanford	4447
	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dtfarris	rX.F{j&j&m&&X	Stanford	6238
	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2;sx<Gjy	Nashua	3137
	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	jialmanza	Ou7RX{yT	New York	3940
...	...	...	...	...	...	...	...	...	...	...	...
	M	Gould	clemente.gould@hotmail.com	1961-12-31	1992-10-02	271-17-5467	228-485-0919	cmgould	m1%+0ojh7VlvJ	Stanford	4062
	K	Roden	chang.roden@yahoo.com	1988-09-07	2010-08-06	074-02-9202	316-256-7851	ckroden	5jRn]G:~58f\$>+S	Nashua	2896
						552-					

## We create a directory and stored a new parquet file

Last Modified

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

6 hours ago

ngs\_da...

6 hours ago

```
[59]: directory = 'output_data/employee_earnings/earnings_date=2022-02-10'
      os.makedirs(directory, exist_ok=True)

[60]: df.to_parquet('output_data/employee_earnings/earnings_date=2022-02-15/new_dataset.parquet', index=False)

[49]: df_1=pd.read_parquet("output_data/employee_earnings/earnings_date=2022-02-11/employee_earnings.parquet")

[50]: df_1.head()

[50]:
```

	emp_id	first_name	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	of
0	526540	Angelique	K	Goodwin	angelique.goodwin@gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{.@	
1	859327	Jeni	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!^O	
2	887387	Donald	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dtfarris	rX.F{&}&m&&X	
3	779497	Steven	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2;sx<Gjy	
4	896517	Jenell	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	jialmanza	Ou7RX{yT	

```
[51]: df_1["earnings"]=df_1["earnings"]+11

[52]: df_1
```

Last Modified

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

4 days ago

ngs\_da...

6 hours ago

ngs\_da...

6 hours ago

```
[52]: df_1

[52]:
```

	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	office_branch	earnings
	K	Goodwin	angelique.goodwin@gmail.com	1964-05-15	2001-03-24	471-57-0359	212-884-7146	akgoodwin	z{d>ez%{.@	Nashua	6107
	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!^O	Stanford	4294
	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dtfarris	rX.F{&}&m&&X	Stanford	3449
	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2;sx<Gjy	Nashua	6236
	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	jialmanza	Ou7RX{yT	New York	5159
	...	...	...	...	...	...	...	...	...	...	...
	M	Gould	clemente.gould@hotmail.com	1961-12-31	1992-10-02	271-17-5467	228-485-0919	cmgould	m1%+0ojh7VlvJ	Stanford	5277
	K	Roden	chang.rodan@yahoo.com	1988-09-07	2010-08-06	074-02-9202	316-256-7851	ckroden	5[Rn]G:~58f\$>+S	Nashua	2226
	R	Nickel	marvin.nickel@ibm.com	1986-11-25	2012-10-06	552-99-5545	270-750-7760	mrnickel	8^E[g_~X]	Scranton	6364
	Y	Tribble	eldora.tribble@earthlink.net	1995-05-29	2016-10-17	763-12-2082	236-584-1916	eytribble	z>ms?.\$8-u	Nashua	8916

```
[53]: df_1.to_parquet('output_data/employee_earnings/earnings_date=2022-02-16/new1_dataset.parquet', index=False)
```

We run the task 4 queries but all the queries are failed  
May be appropriate filters, reducing the data volume, or using partitioning techniques,  
depending on the query engine .

Access Analyzer for IAM

Public Access settings for account

ge Lens

boards

Organizations settings

ire spotlight

Marketplace for S3

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](#).

Add SQL from templates

Run SQL query

```
1 /* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
2 SELECT DISTINCT emp_id, email, office_branch, (date_diff('year', DATE(date_of_birth), current_date)) AS age
3 FROM "aws-glue-databases-huan.employee_earings"
4 WHERE office_branch IN ('New York', 'Scranton')
5 AND
6 (date_diff('year', DATE(date_of_birth), current_date)) > 30;
```

Unexpected keyword found, KEYWORD:UNKNOWN at line 1, column 20.

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

Download results