

UNIT 5.4 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)

Umaina Siddiqui (2023.KHI.DEG.033)

UNIT 5.4 GRADED ASSIGNMENT

Task:

Use data from today's Daily Activities tasks/5_data_pipelines/day_4_data_lake/data/output_data/employee_earnings

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).

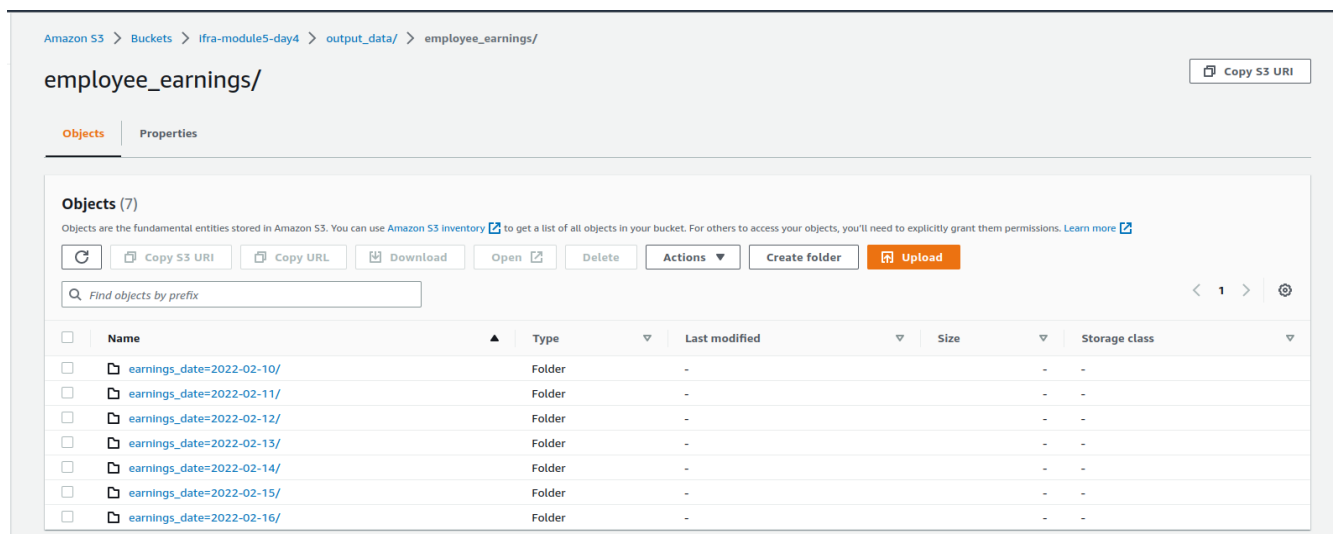
Rerun queries from Task 3 and Task 4 and see how the results change with this new data.

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

Solution:

Uploaded two new folders to S3 Bucket.

Script file is attached with the assignment which we used to create new data files for two more days.



Amazon S3 > Buckets > ifra-module5-day4 > output_data/ > employee_earnings/

employee_earnings/ [Copy S3 URI](#)

Objects Properties

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](#)

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

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	earnings_date=2022-02-10/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-11/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-12/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-13/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-14/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-15/	Folder	-	-	-
<input type="checkbox"/>	earnings_date=2022-02-16/	Folder	-	-	-

Run the Crawler again:

+

2

 Copy

Search rows

< 1 ... > ⚙

+

6

Results (46)

Copy

Download results

Q Search rows

< 1 > ⌕

#	emp_id	email	office_branch	age
1	900756	benjamin.doss@gmail.com	Scranton	38
2	215719	brent.carrillo@aol.com	New York	50
3	530134	mathew.whitfield@gmail.com	New York	36
4	597741	tonya.wilson@aol.com	New York	43
5	391837	cory.hayden@gmail.com	New York	56
6	622405	harrison.hawk@hotmail.co.uk	Scranton	60
7	595558	denisha.fitch@msn.com	Scranton	32
8	314661	charles.quintero@gmail.com	New York	65
9	654617	rogerio.woodall@gmail.com	New York	50
10	138911	claudio.heck@aol.com	Scranton	55
11	713294	sammy.dewitt@ibm.com	Scranton	35
12	312726	celine.lumpkin@gmail.com	New York	36
13	551149	michale.colson@comcast.net	Scranton	61
14	402180	allan.bernhardt@gmail.com	New York	61
15	220965	almeta.brookins@gmail.com	Scranton	38
16	537591	samuel.wendt@bellsouth.net	New York	46
17	767674	irena.dang@gmail.com	New York	51
18	505927	oswaldo.winchester@gmail.com	New York	64
19	432820	myron.marble@gmail.com	New York	42
20	317987	chastity.pineda@shaw.ca	New York	48

Query 1 : X

Query 1 : X

Query 2 : X

Query 3 : X

Query 14 : X

Query 4 : X

+

▼

1

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

2

FROM "ifrasaleem_glue_database"."ifrasaleememployee_earnings"

3

GROUP BY office_branch, earnings_date

4

ORDER BY SUM(earnings) desc;

Results (28)

Copy

Download results

Q Search rows

< 1 > ⌕

#	office_branch	min_earnings	max_earnings	avg_earnings	total_earnings	earnings_date
1	Nashua	5234	19135	11813.967741935483	366233	2022-02-16
2	Nashua	4101	18006	11617.870967741936	360154	2022-02-15
3	New York	5989	18901	12246.82142857143	342911	2022-02-16
4	New York	4605	19150	11989.5	335706	2022-02-15
5	Scranton	6444	17504	12894.04	322351	2022-02-16
6	Scranton	4521	18219	11056.88	276422	2022-02-15
7	Stanford	7846	18576	12078.9375	193263	2022-02-15
8	Nashua	2098	9728	6099.8387096774195	189095	2022-02-14
9	Nashua	2005	9786	6049.451612903225	187533	2022-02-13
10	Stanford	6063	17919	11706.5625	187305	2022-02-16
11	Nashua	2006	9603	5997.967741935484	185937	2022-02-11
12	New York	2295	9889	6631.285714285715	185676	2022-02-12
13	Nashua	2124	9978	5764.5161290322585	178700	2022-02-12
14	Nashua	2066	9801	5619.903225806452	174217	2022-02-10
15	New York	2040	9954	6109.035714285715	171053	2022-02-14
16	Scranton	2788	9916	6830.6	170765	2022-02-13
17	New York	2141	9462	5998.178571428572	167949	2022-02-11
18	New York	2376	9972	5991.321428571428	167757	2022-02-10
19	New York	2195	9734	5615.535714285715	157235	2022-02-13
20	Scranton	2465	9827	6149.72	153743	2022-02-14

Query 1 : X

Query 1 : X

Query 2 : X

Query 3 : X

Query 14 : X

Query 4 : X

+

▼

1

SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range

2

FROM (

3

SELECT office_branch as ob, AVG(earnings) AS value FROM "ifrasaleem_glue_database"."ifrasaleememployee_earnings" GROUP BY office_branch, earnings_date

4

) avg_earnings, "ifrasaleem_glue_database"."ifrasaleememployee_earnings"

5

WHERE office_branch = avg_earnings.ob

6

GROUP BY office_branch;

Query results

Query stats

Completed

Time in queue: 150 ms

Run time: 1.042 sec

Data scanned: 6.20 KB

Results (4)

Search rows

CopyDownload results

<1>

#	office_branch	earnings_range
1	Stanford	6514.5625
2	Nashua	6194.064516129031
3	New York	6631.285714285715
4	Scranton	7842.720000000001

Task 4:

S3 select has some limitations on the type of queries it supports. and complex SQL queries with window functions like LAG are not supported.

S3 Select is primarily designed for simple SQL queries that involve basic filtering and projection operations on CSV, JSON, or Parquet data stored in S3. It does not support advanced SQL features like window functions or subqueries. That's why it is giving errors in running the queries.

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](#).

```

1 SELECT *
2 FROM "frasaaleen_glue_database"."lfrasaaleenemployee_earnings";
3 -- SELECT * FROM s3object s LIMIT 5;

```

Unexpected term found EOF:UNKNOWN at line 1, column 36.

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

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](#).

```

1 SELECT DISTINCT emp_id, email, office_branch, (date_diff('year', DATE(date_of_birth), current_date)) AS age
2 FROM "frasaaleen_glue_database"."lfrasaaleenemployee_earnings"
3 WHERE office_branch IN ('New York', 'Scranton')
4 AND
5 (date_diff('year', DATE(date_of_birth), current_date)) > 30;
6 /* 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 */
7 -- SELECT * FROM s3object s LIMIT 5;

```

Unexpected term found EOF:UNKNOWN at line 1, column 36.

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

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 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
2 FROM "ifrasaleen_glue_database"."ifrasaleenemployee_earnings"
3 GROUP BY office_branch, earnings_date
4 ORDER BY SUM(earnings) desc; /* 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 */
5 -- SELECT * FROM s3object s LIMIT 5
```

Unexpected term found EOF:UNKNOWN at line 1, column 36.

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.

Download results

Status

Failed

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 SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range
2 FROM (
3 SELECT office_branch as ob, AVG(earnings) AS value FROM "ifrasaleen_glue_database"."ifrasaleenemployee_earnings" GROUP BY office_branch, earnings_date
4 ) avg_earnings, "ifrasaleen_glue_database"."ifrasaleenemployee_earnings"
5 WHERE office_branch = avg_earnings.ob
6 GROUP BY office_branch; /* 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 */
7 -- SELECT * FROM s3object s LIMIT 5
```

Unexpected term found EOF:UNKNOWN at line 1, column 36.

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.

Download results

Status

Failed

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

Query 1 : XQuery 2 : XQuery 3 : XQuery 14 : X

1- WITH earnings_data AS (
2 SELECT emp_id, earnings, earnings_date,
3 LAG(earnings) OVER (PARTITION BY emp_id ORDER BY earnings_date) AS previous_earnings
4 FROM "ifrasaleem_glue_database"."ifrasaleememployee_earnings"
5 WHERE earnings_date IN ('2022-02-13', '2022-02-14')
6)
7 SELECT emp_id, earnings_date, earnings,
8 ((earnings - previous_earnings) / CAST(previous_earnings AS double)) * 100 AS percentage_change
9 FROM earnings_data
10 WHERE earnings_date = '2022-02-14';
11

Results (100)

CopyDownload results

Search rows

< 1 ... >🔍

#	emp_id	earnings_date	earnings	percentage_change
1	149972	2022-02-14	4881	-37.750286953194745
2	174955	2022-02-14	8857	267.6629306766293
3	203380	2022-02-14	7741	-19.6574987026466
4	215719	2022-02-14	6625	-26.576526654106175
5	299088	2022-02-14	4627	-41.80606213054961
6	388558	2022-02-14	2642	-55.951983994664886
7	399707	2022-02-14	4804	-2.456852791878173
8	439483	2022-02-14	4101	19.772196261682243
9	466832	2022-02-14	8708	224.68307233407904
10	500905	2022-02-14	5554	-29.642766658221436
11	501658	2022-02-14	8127	-0.6964809384164222
12	511059	2022-02-14	7454	62.39651416122004
13	550206	2022-02-14	2870	15.400080418174507
14	595558	2022-02-14	3849	-21.175506860536554
15	627298	2022-02-14	2568	-64.16910841356216
16	637407	2022-02-14	4280	-37.81781200058114
17	754455	2022-02-14	9470	106.18332244720226
18	823898	2022-02-14	4977	-13.48861463584217
19	856379	2022-02-14	4383	11.982626469085336