

## Members:

1. Syed Mohammad Anjil Hussain Rizvi (2303.KHI.DEG.031)
2. Huzefa Anver (2303.KHI.DEG.002)

## Simulate earning predictions for two more days:

Manipulations to the data are made in Untitled.ipynb file. It can be found inside the same folder as this pdf file.

Data has been manipulated using pandas. The manipulation we are talking about is simply multiplying the earnings column with 4.

## Parquet file loaded:

taking a parquet file and manipulating its earnings column to produce a new parquet file

```
[1]: import pandas as pd
import numpy as np

[2]: df_1 = pd.read_parquet('/home/syedmohamadanijilhussainrizvi/Desktop/data_engineering_bootcamp_2303/tasks/5_data_pipelines/day_4_data_lake/data/output_data/employee_earnin
```

[3]: df\_1

	emp_id	first_name	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	office_branch	earnings
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%{ @	Nashua	6227
1	859327	Jeni	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56!MO	Stanford	4437
2	887387	Donald	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dfarris	rX.Fj[a]4m&AX	Stanford	6228
3	779497	Steven	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2,sv}<Gjy	Nashua	3127
4	896517	Jenell	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	j.almanza	Ou7RXjyT	New York	3930
...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	549389	Clemente	M	Gould	clemente.gould@hotmail.com	1961-12-31	1992-10-02	271-17-5467	228-485-0919	cmgould	m1%+0qjh7VwJ	Stanford	4052
96	466832	Chang	K	Roden	chang.roden@yahoo.com	1988-09-07	2010-08-06	074-02-9202	316-256-7851	ckroden	5jRtnjG:~5eIS->+S	Nashua	2886
97	203380	Marvin	R	Nickel	marvin.nickel@ibm.com	1986-11-25	2012-10-06	552-99-5545	270-750-7760	mrnickel	8*E[ig_-X]	Scranton	8586
98	915991	Eldora	Y	Tribble	eldora.tribble@earthlink.net	1995-05-29	2016-10-17	763-12-2082	236-584-1916	eytribble	z>ms7,\$8-u	Nashua	6493
99	289172	Azzie	L	Layman	azzie.layman@hotmail.co.uk	1961-09-06	2004-03-26	637-29-1007	503-456-5899	allayman	k<%%7%TML_].1ZY	New York	5261

100 rows x 13 columns

## Accessing the earnings column:

```
[4]: earnings = df_1.loc[:, 'earnings']
earnings
```

0	6227
1	4437
2	6228
3	3127
4	3930
...	...
95	4052
96	2886
97	8586
98	6493
99	5261

Name: earnings, Length: 100, dtype: int64

## Manipulating the earnings column

```
[12]: new_earnings = earnings * 4
```

```
[13]: df_1.loc[:, 'earnings'] = new_earnings
df_1
```

```
[13]:
```

	emp_id	first_name	middle_initial	last_name	email	date_of_birth	date_of_joining	ssn	phone_number	user_name	password	office_branch	earnings
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%{.@	Nashua	99632
1	859327	Jeni	S	Shaffer	jeni.shaffer@gmail.com	1962-01-13	2015-12-10	624-85-4146	205-665-7020	jsshaffer	7U56*MO	Stanford	70992
2	887387	Donald	T	Farris	donald.farris@bellsouth.net	1958-04-11	1979-11-12	097-02-3315	205-959-7879	dfarris	rX.F[a]4m&AX	Stanford	99648
3	779497	Steven	D	Rendon	steven.rendon@gmail.com	1982-04-04	2008-09-18	134-98-6566	217-858-0054	sdrendon	a+2;sxj<Gjy	Nashua	50032
4	896517	Jenell	L	Almanza	jenell.almanza@yahoo.com	1958-07-01	1993-07-14	599-92-7345	314-893-2590	j.almanza	Ou7RXjyT	New York	62880
...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	549389	Clemente	M	Gould	clemente.gould@hotmail.com	1961-12-31	1992-10-02	271-17-5467	228-485-0919	cmgould	m1%+0qj7Vhvj	Stanford	64832
96	466832	Chang	K	Roden	chang.roden@yahoo.com	1988-09-07	2010-08-06	074-02-9202	316-256-7851	ckroden	5jRtnjG:5e\$>+1S	Nashua	46176
97	203380	Marvin	R	Nickel	marvin.nickel@ibm.com	1986-11-25	2012-10-06	552-99-5545	270-750-7760	mmnickel	8*Ejg_Xj	Scranton	137376
98	915991	Eldora	Y	Tribble	eldora.tribble@earthlink.net	1995-05-29	2016-10-17	763-12-2082	236-584-1916	eytribble	z>ms?;58-u	Nashua	103888
99	289172	Azzie	L	Layman	azzie.layman@hotmail.co.uk	1961-09-06	2004-03-26	637-29-1007	503-456-5899	allayman	k<%?%TML_l1ZY	New York	84176

100 rows x 13 columns

## Creating a new parquet file:

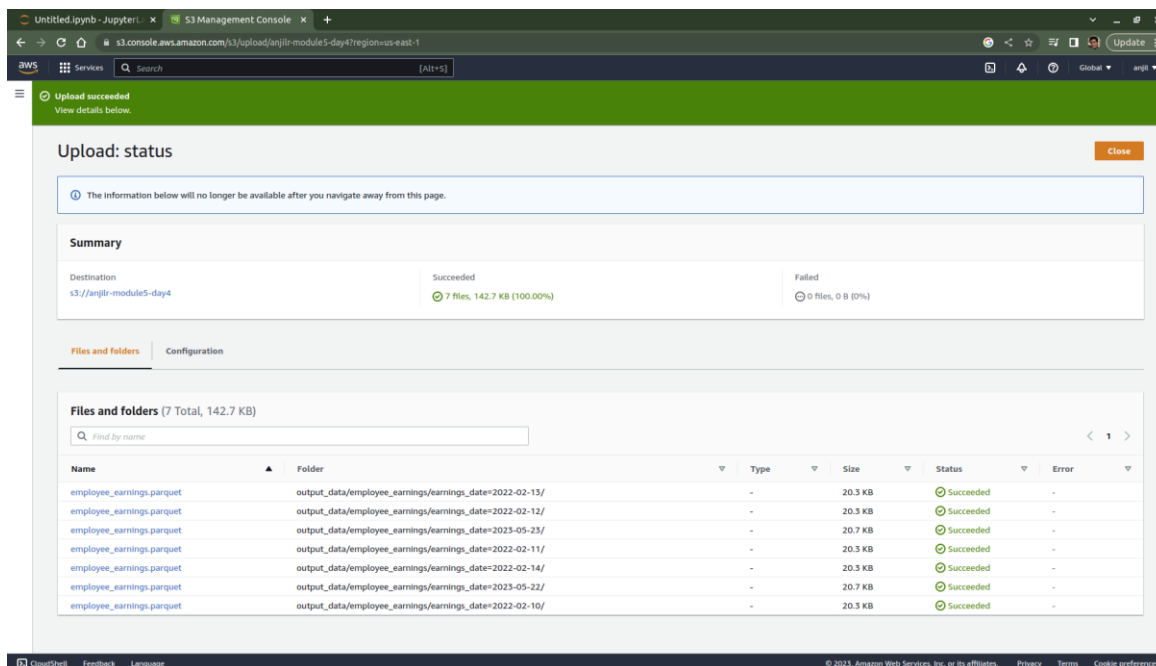
```
[16]: path = "/home/syednohammadanjilhussainrizvi/Desktop/data_engineering_bootcamp_Z383/tasks/5_data_pipelines/day_4_data_lake/data/output_data/employee_earnings/earnings_data"
```

```
[17]: df_1.to_parquet(path)
```

All of these steps were repeated on another existing parquet file to create one more parquet file for another day.

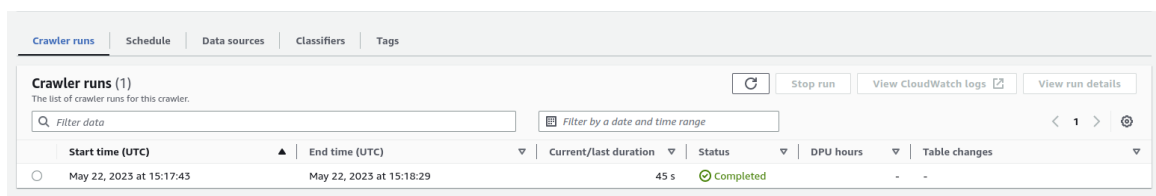
## Data lake preparation:

1. Contents loaded into an s3 bucket:

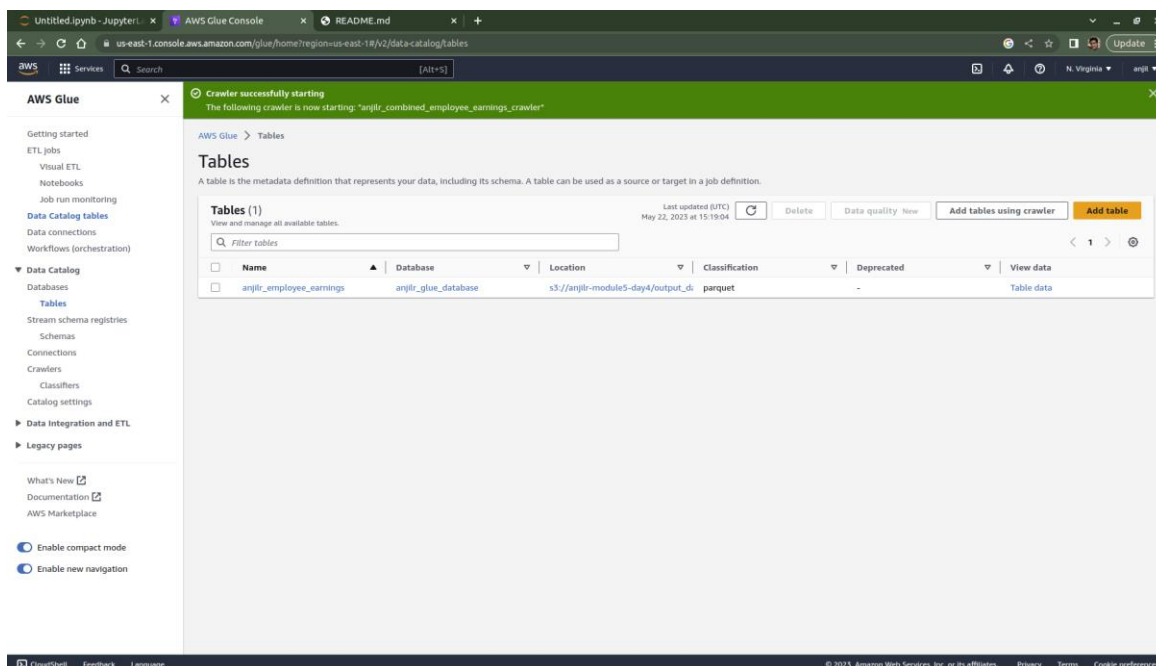


(folders with year 2023 in the title are the new folders containing new parquet files)

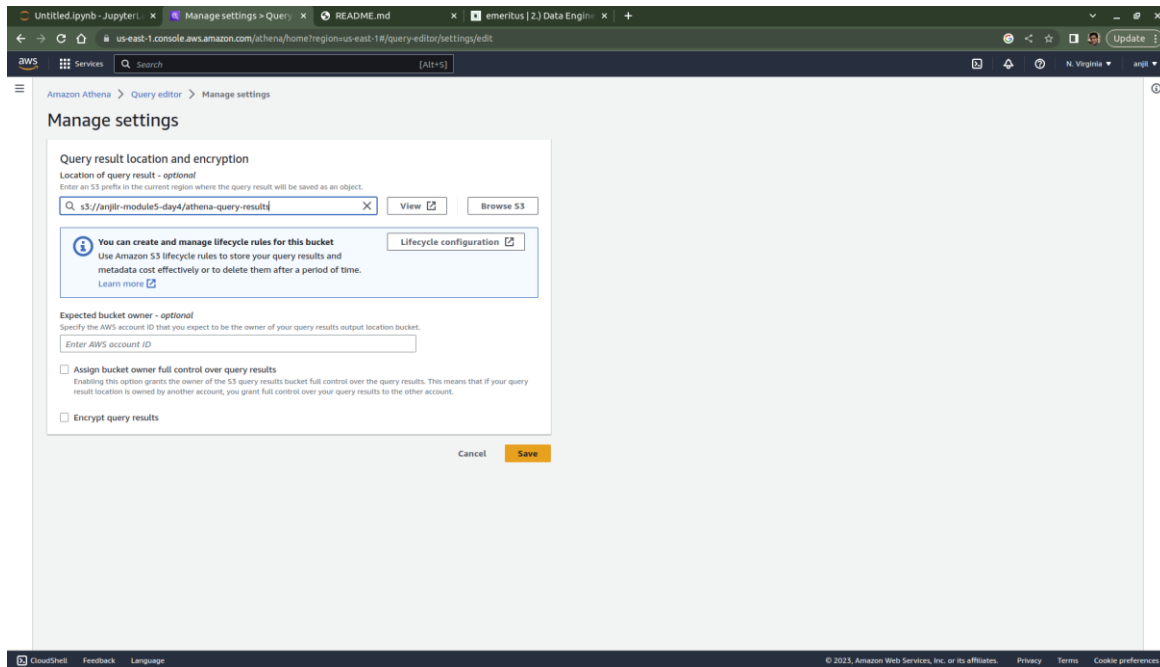
## 2. Crawler created and spun up:



## 3. We can see that it has produced our table that we are going to query from:



#### 4. Query result location set inside s3 bucket:



## Querying using Athena:

All employees from offices 'New York' and 'Scranton' with age > 30:

The screenshot shows the AWS Athena console interface. At the top, there's a search bar and navigation icons. Below, the 'Query 3' tab is active, displaying a SQL query:

```
1 SELECT DISTINCT emp_id, email, office_branch, (date_diff('year', DATE(date_of_birth), current_date)) AS age
2 FROM "anjilr_glue_database"."anjilr_employee_earnings"
3 WHERE office_branch IN ('New York', 'Scranton')
4 AND
5 (date_diff('year', DATE(date_of_birth), current_date)) > 30;
```

Below the query editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Run again' button is highlighted in orange. To the right, there's a toggle for 'Reuse query results' and a note '\*Athena engine version 3 only'.

The 'Query results' tab is selected, showing a status bar with 'Completed', 'Time in queue: 155 ms', 'Run time: 1.187 sec', and 'Data scanned: 26.66 KB'. Below this, there's a search bar for results and a table of 46 results.

#	emp_id	email	office_branch	age
1	909018	virgil.trowbridge@aol.com	New York	37
2	878666	elda.champagne@gmail.com	Scranton	40
3	496541	winfred.gonzales@aol.com	Scranton	59
4	976422	jake.espinal@shaw.ca	Scranton	64
5	627298	sterling.serna@hotmail.com	New York	38
6	452163	adalberto.tate@shaw.ca	New York	41
7	147133	tommie.weller@cox.net	Scranton	63
8	403207	michal.maurer@yahoo.com	Scranton	46
9	900756	benjamin.doss@gmail.com	Scranton	38

At the bottom of the console, there's a footer with 'cloudshell', 'Feedback', 'Language', and copyright information: '© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

(no changes anticipated between this run and queries run before manipulation of data)

Min, max, average and total earnings for each office and each day - sorted by total earnings, highest to lowest:

The screenshot displays the AWS Athena console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and a language dropdown set to 'en\_US'. Below this, a tabbed interface shows 'Query 2', 'Query 3', and 'Query 4'. The active tab is 'Query 4', which contains the following 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,
2 earnings_date
3 FROM "anjilr_glue_database"."anjilr_employee_earnings"
4 GROUP BY office_branch, earnings_date
5 ORDER BY SUM(earnings) desc;
```

Below the query editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. A status bar indicates 'SQL Ln 4, Col 29'. To the right, there are icons for 'Run query results' and 'Athena engine version 3 only'.

The 'Query results' tab is selected, showing a green bar with 'Completed' status. Performance metrics are displayed: 'Time in queue: 197 ms', 'Run time: 1.133 sec', and 'Data scanned: 5.30 KB'. Below this, a search bar for 'Search rows' is present. The results are shown in a table with 8 rows and 7 columns:

#	office_branch	min_earnings	max_earnings	avg_earnings	total_earnings	earnings_date
1	Nashua	33056	156816	89918.45161290323	2787472	2023-05-22
2	New York	38016	159552	95861.14285714286	2684112	2023-05-22
3	Scranton	32528	158208	96088.96	2402224	2023-05-22
4	Stanford	58272	151472	93460.0	1495360	2023-05-22
5	Nashua	2098	9728	6099.8387096774195	189095	2022-02-14
6	Nashua	2005	9786	6049.451612903225	187533	2022-02-13
7	Nashua	2006	9603	5997.967741935484	185937	2023-05-23
8	Nashua	2006	9603	5997.967741935484	185937	2022-02-11

At the bottom of the console, there's a footer with 'cloudshell', 'Feedback', 'Language', and copyright information: '© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

(in this one however we can expect different query results between this run and the same query run before data manipulation; this query is related to earnings which is why we can expect a difference. In fact, we can already see the data points from new parquet files being reflected in the first few rows of the query)

Difference between worst and best day earnings for every office branch:

The screenshot shows the Amazon Athena Query Editor interface. The SQL query is as follows:

```
1 SELECT DISTINCT office_branch, (MAX(avg_earnings.value) - MIN(avg_earnings.value)) as earnings_range
2 -
3 FROM (
4   SELECT office_branch as ob, AVG(earnings) AS value FROM "anjilr_glue_database"."anjilr_employee_earnings" GROUP BY office_branch, earnings_date
5   WHERE office_branch = avg_earnings.ob
6   GROUP BY office_branch;
```

The query has been executed successfully, and the results are displayed in a table with 4 rows:

#	office_branch	earnings_range
1	Stanford	87895.625
2	Scranton	91037.64000000001
3	New York	90245.60714285714
4	Nashua	84298.54838709677

## Assignment Query:

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

The screenshot shows the Amazon Athena Query Editor interface with a new query. The SQL query is as follows:

```
1 WITH earnings_data AS (
2   SELECT
3     emp_id,
4     earnings_date,
5     earnings,
6     LAG(earnings) OVER (PARTITION BY emp_id ORDER BY earnings_date) AS previous_earnings
7   FROM
8     "anjilr_glue_database"."anjilr_employee_earnings"
9   WHERE
10    earnings_date BETWEEN '2022-02-13' AND '2022-02-14'
11 )
12 SELECT
13   emp_id,
14   earnings_date,
15   earnings,
16   previous_earnings,
17   ((earnings - CAST(previous_earnings as double)) / previous_earnings) * 100 AS percentage_change
18 FROM
19   earnings_data
20 WHERE earnings_date = '2022-02-14'
21 ORDER BY
22   emp_id, earnings_date
```

Completed							Time in queue: 156 ms	Run time: 852 ms	Data scanned: 2.62 KB
Results (100)							<div>Copy</div>		<div>Download results</div>
<div>Search rows</div>							<div>&lt; 1 ... &gt; </div>		
#	emp_id	earnings_date	earnings	previous_earnings	percentage_change				
1	138911	2022-02-14	6709	6112	9.767670157068062				
2	143711	2022-02-14	8447	9462	-10.72711900232509				
3	147133	2022-02-14	6348	6502	-2.368501999384805				
4	149972	2022-02-14	4881	7841	-37.750286953194745				
5	155097	2022-02-14	3945	8825	-55.297450424929174				
6	160938	2022-02-14	3469	9033	-61.596368869699994				
7	163409	2022-02-14	5323	7281	-26.891910451861005				
8	170637	2022-02-14	8950	8601	4.057667713056621				
9	174955	2022-02-14	8857	2409	267.6629306766293				
10	184257	2022-02-14	5190	8394	-38.17012151536812				
11	203380	2022-02-14	7741	9635	-19.6574987026466				
12	215719	2022-02-14	6625	9023	-26.576526654106175				
13	220965	2022-02-14	9378	6721	39.532807617914				
14	233136	2022-02-14	6499	8704	-25.333180147058826				
15	235295	2022-02-14	5760	7327	-21.38665210863928				
16	242388	2022-02-14	3467	8227	-57.85827154491309				
17	289172	2022-02-14	5868	4817	21.818559269254724				
18	299088	2022-02-14	4627	7951	-41.80606213054961				
19	312726	2022-02-14	6055	3109	94.75715664200708				
20	314661	2022-02-14	8483	4480	89.35267857142857				