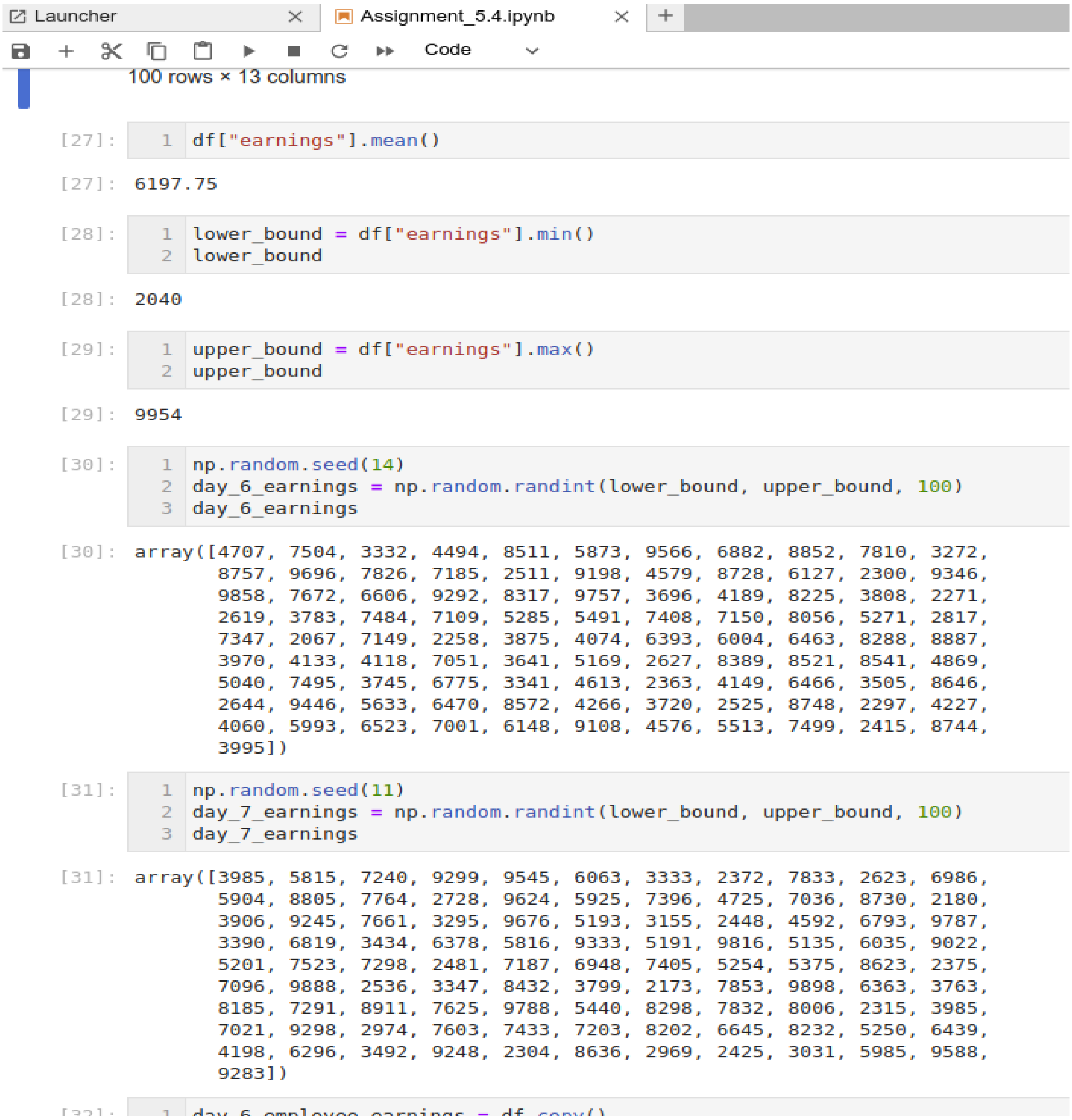
Graded Assignment 5.4

Name: Saad Sameer Khan
Employee#: 2303.KHI.DEG.034
Collaborated with: Mohammad Hamza Asim (2303.KHI.DEG.014)

Creating data for 2 more days (day 6 & day 7)

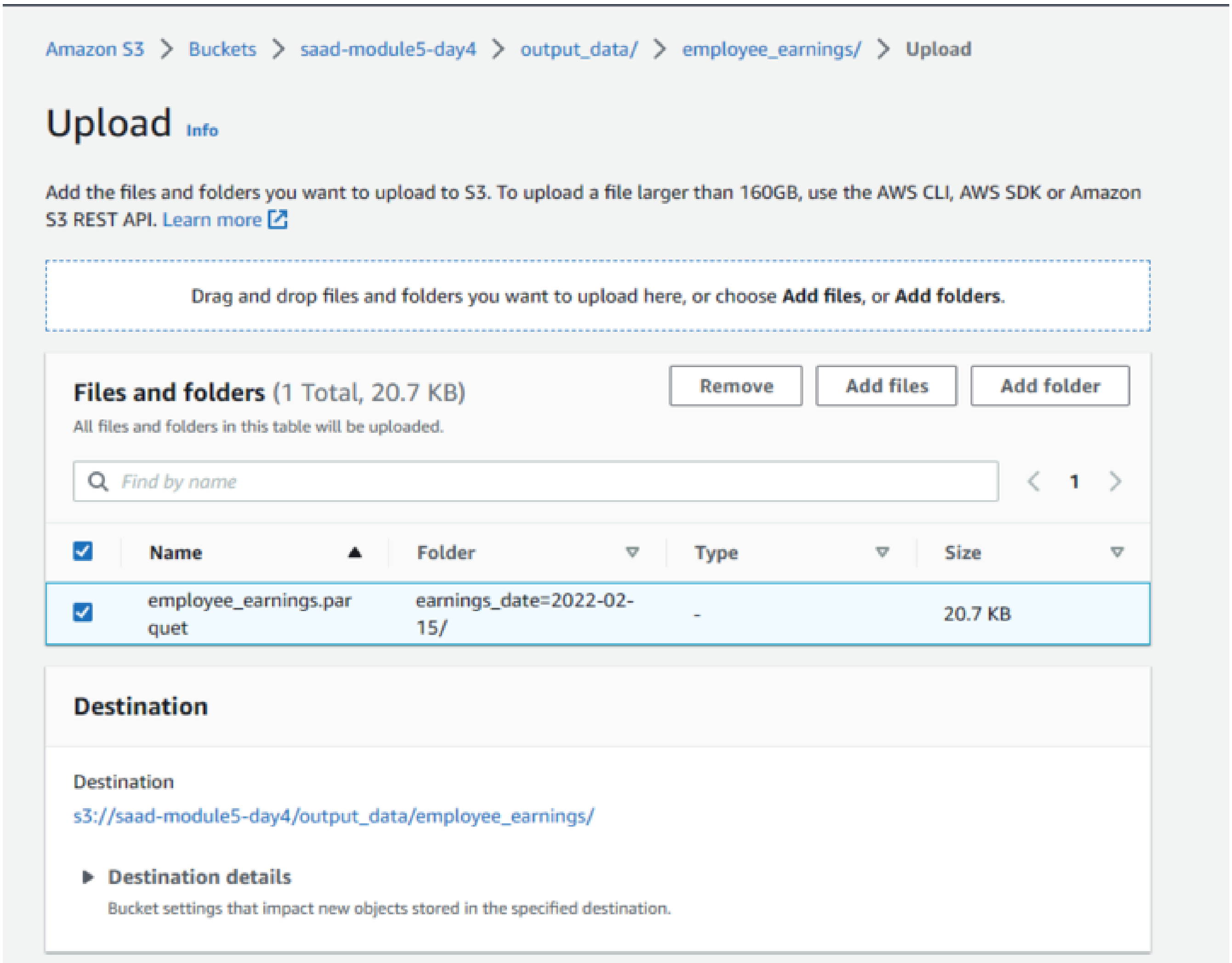
We used numpy's random.randint method to generate random earnings data within the range of the maximum and minimum value of the 'earnings' column from one of the day's data. *ipynb file has been uploaded to git*



```
100 rows × 13 columns

[27]: df["earnings"].mean()

[27]: 6197.75

[28]: lower_bound = df["earnings"].min()
      lower_bound

[28]: 2040

[29]: upper_bound = df["earnings"].max()
      upper_bound

[29]: 9954

[30]: np.random.seed(14)
      day_6_earnings = np.random.randint(lower_bound, upper_bound, 100)
      day_6_earnings

[30]: array([4707, 7504, 3332, 4494, 8511, 5873, 9566, 6882, 8852, 7810, 3272,
             8757, 9696, 7826, 7185, 2511, 9198, 4579, 8728, 6127, 2300, 9346,
             9858, 7672, 6606, 9292, 8317, 9757, 3696, 4189, 8225, 3808, 2271,
             2619, 3783, 7484, 7109, 5285, 5491, 7408, 7150, 8056, 5271, 2817,
             7347, 2067, 7149, 2258, 3875, 4074, 6393, 6004, 6463, 8288, 8887,
             3970, 4133, 4118, 7051, 3641, 5169, 2627, 8389, 8521, 8541, 4869,
             5040, 7495, 3745, 6775, 3341, 4613, 2363, 4149, 6466, 3505, 8646,
             2644, 9446, 5633, 6470, 8572, 4266, 3720, 2525, 8748, 2297, 4227,
             4060, 5993, 6523, 7001, 6148, 9108, 4576, 5513, 7499, 2415, 8744,
             3995])

[31]: np.random.seed(11)
      day_7_earnings = np.random.randint(lower_bound, upper_bound, 100)
      day_7_earnings

[31]: array([3985, 5815, 7240, 9299, 9545, 6063, 3333, 2372, 7833, 2623, 6986,
             5904, 8805, 7764, 2728, 9624, 5925, 7396, 4725, 7036, 8730, 2180,
             3906, 9245, 7661, 3295, 9676, 5193, 3155, 2448, 4592, 6793, 9787,
             3390, 6819, 3434, 6378, 5816, 9333, 5191, 9816, 5135, 6035, 9022,
             5201, 7523, 7298, 2481, 7187, 6948, 7405, 5254, 5375, 8623, 2375,
             7096, 9888, 2536, 3347, 8432, 3799, 2173, 7853, 9898, 6363, 3763,
             8185, 7291, 8911, 7625, 9788, 5440, 8298, 7832, 8006, 2315, 3985,
             7021, 9298, 2974, 7603, 7433, 7203, 8202, 6645, 8232, 5250, 6439,
             4198, 6296, 3492, 9248, 2304, 8636, 2969, 2425, 3031, 5985, 9588,
             9283])

[32]: day_6_employee_earnings = df.copy()
```

Uploading new folders to S3 bucket

Next, we uploaded two new folders containing the data created in the previous step, in the output_data folder in the S3 bucket.



Amazon S3 > Buckets > saad-module5-day4 > output_data/ > employee_earnings/ > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (1 Total, 20.7 KB)

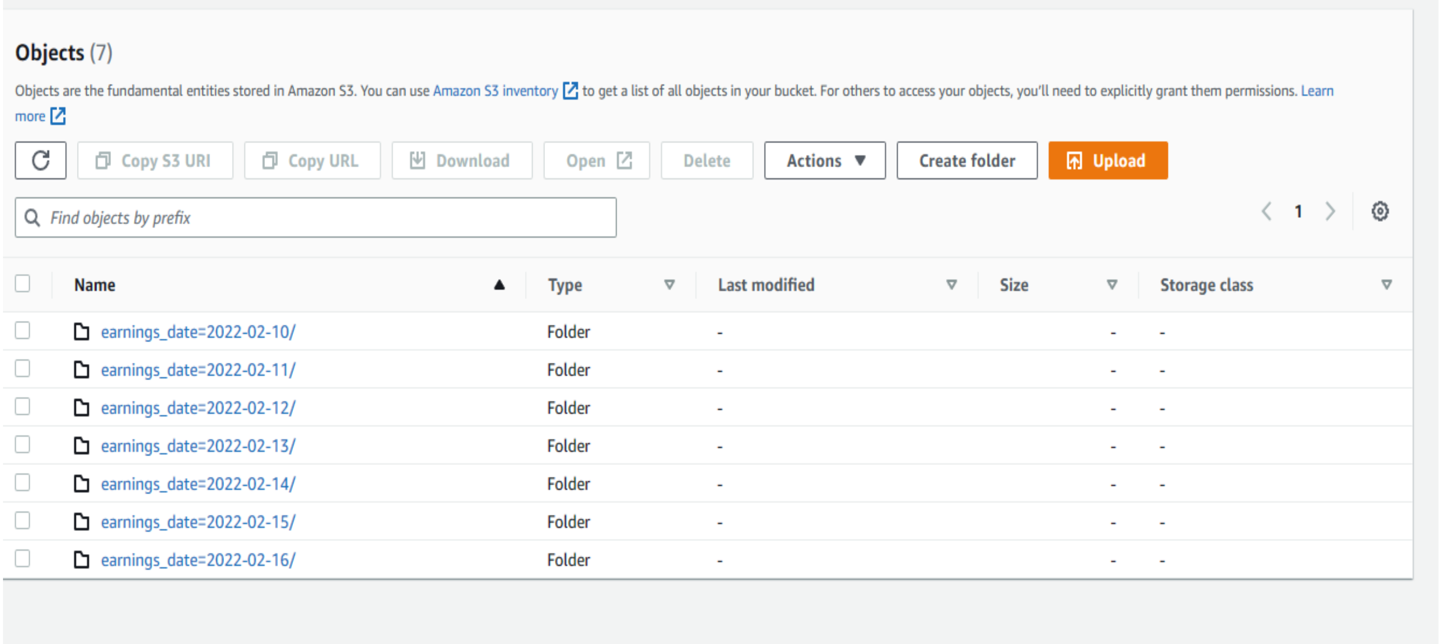
All files and folders in this table will be uploaded.

☑	Name	Folder	Type	Size
☑	employee_earnings.parquet	earnings_date=2022-02-15/	-	20.7 KB

Destination

Destination
s3://saad-module5-day4/output_data/employee_earnings/

▶ **Destination details**
Bucket settings that impact new objects stored in the specified destination.



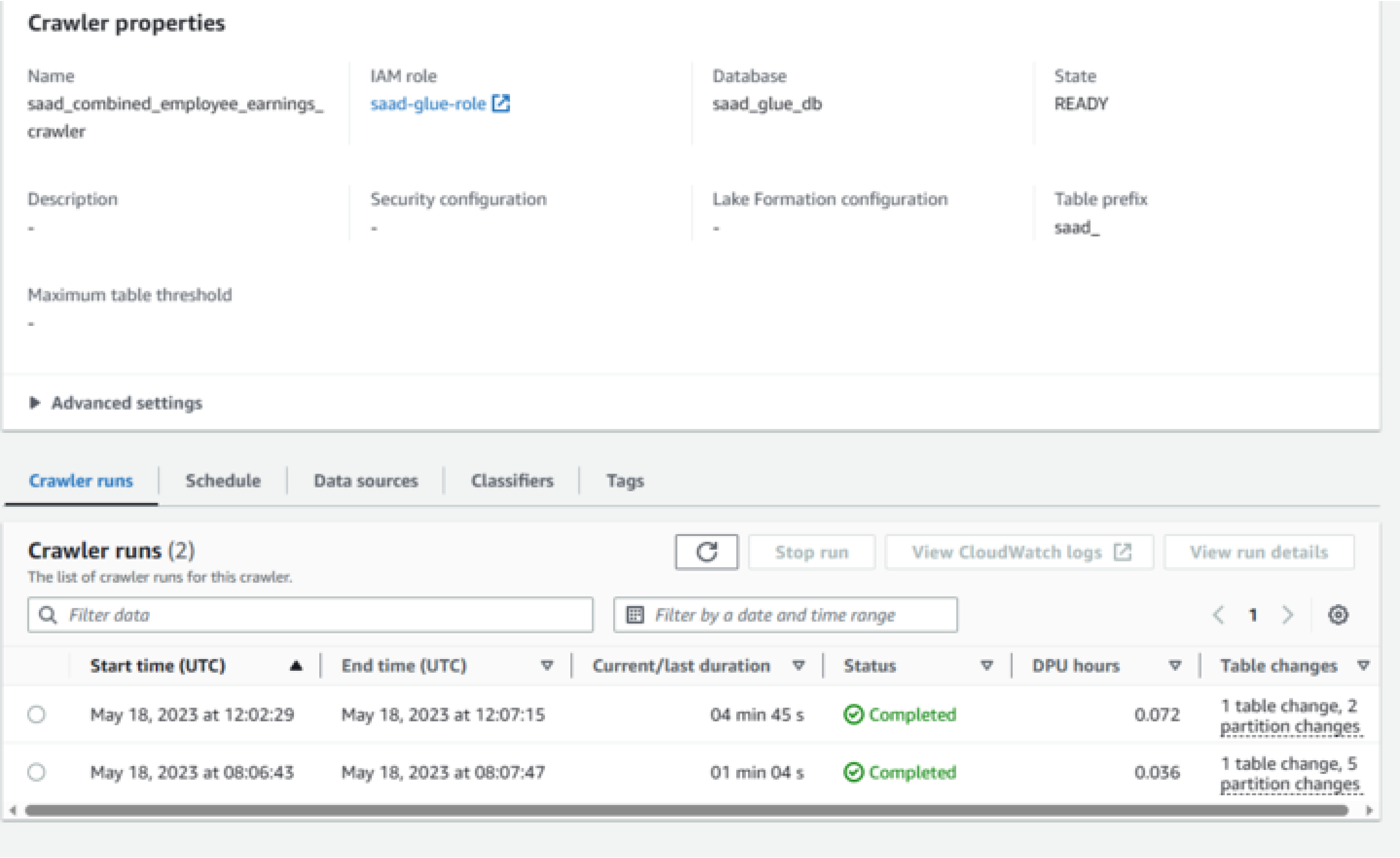
Objects (7)

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	-	-	-

Running crawler

After the new folders were uploaded, we ran the crawler so that it could fetch the new data from the bucket, and update its data catalog, so that we could use the new data in Athena.



Crawler properties

Name
saad_combined_employee_earnings_crawler

IAM role
saad-glue-role

Database
saad_glue_db

State
READY

Description
-

Security configuration
-

Lake Formation configuration
-

Table prefix
saad_

Maximum table threshold
-

Crawler runs (2)
The list of crawler runs for this crawler.

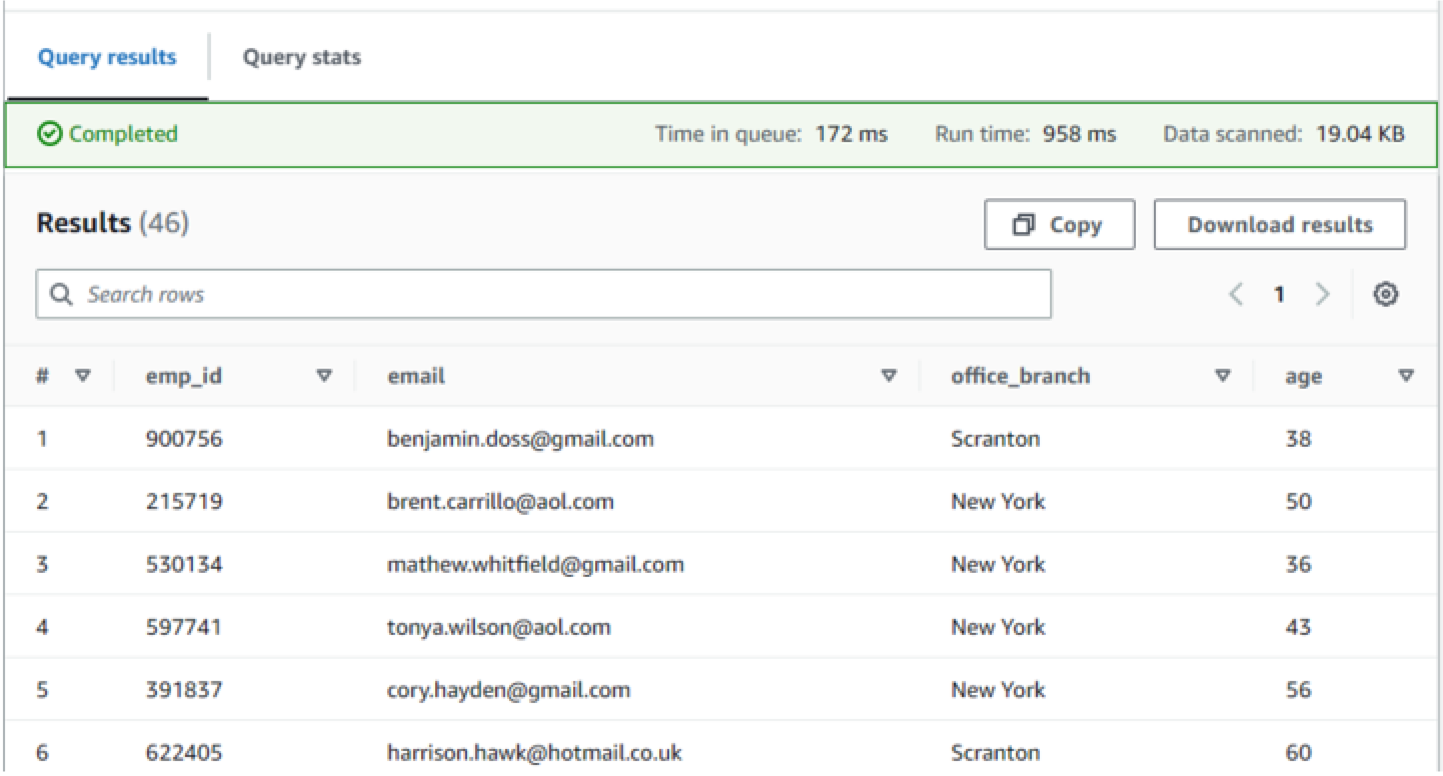
	Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
○	May 18, 2023 at 12:02:29	May 18, 2023 at 12:07:15	04 min 45 s	✓ Completed	0.072	1 table change, 2 partition changes
○	May 18, 2023 at 08:06:43	May 18, 2023 at 08:07:47	01 min 04 s	✓ Completed	0.036	1 table change, 5 partition changes

Re-running queries on updated data

Now we re-ran the queries that were previously run, and observe what changes took place in the results.

Query 1

Before:

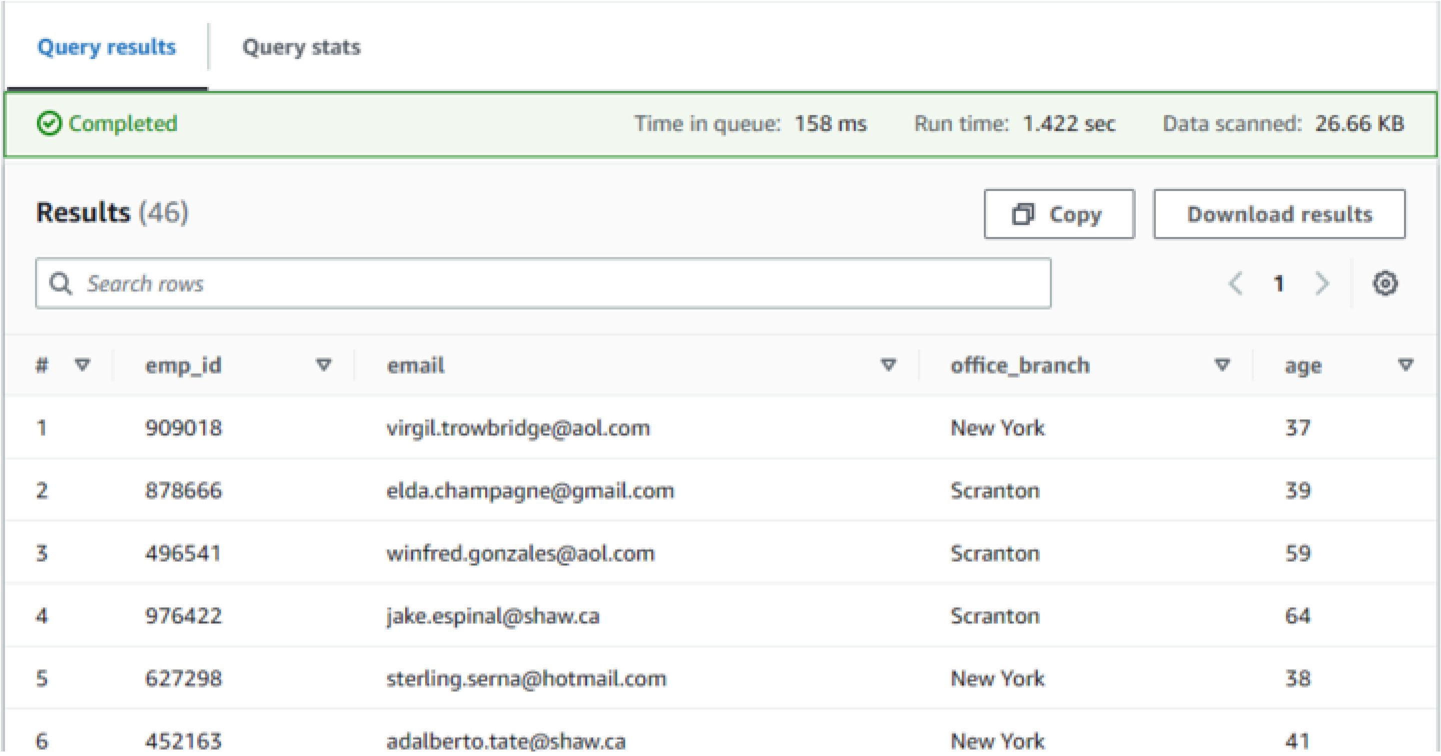


✓ Completed Time in queue: 172 ms Run time: 958 ms Data scanned: 19.04 KB

Results (46)

#	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

After



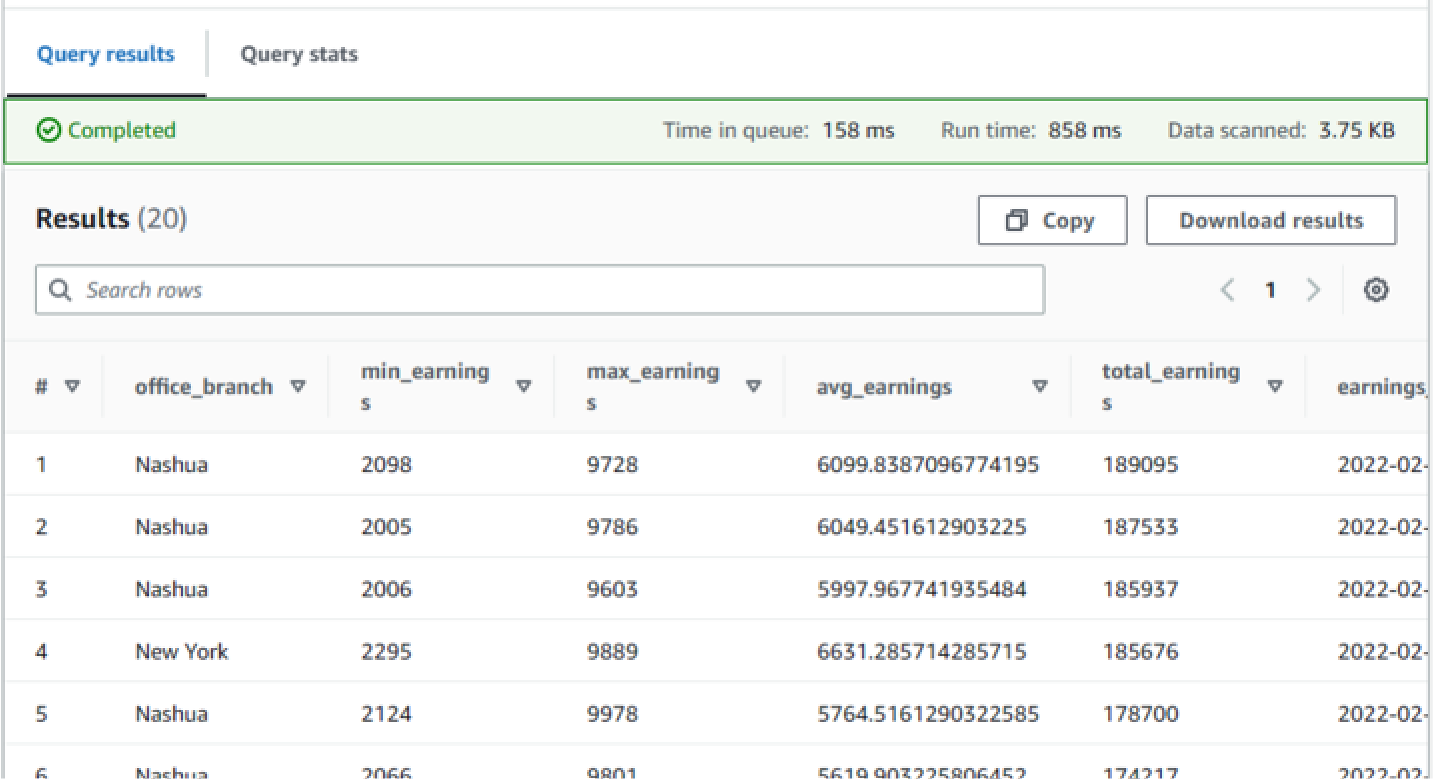
✓ Completed Time in queue: 158 ms Run time: 1.422 sec Data scanned: 26.66 KB

Results (46)

#	emp_id	email	office_branch	age
1	909018	virgil.trowbridge@aol.com	New York	37
2	878666	elda.champagne@gmail.com	Scranton	39
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

Query 2

Before:

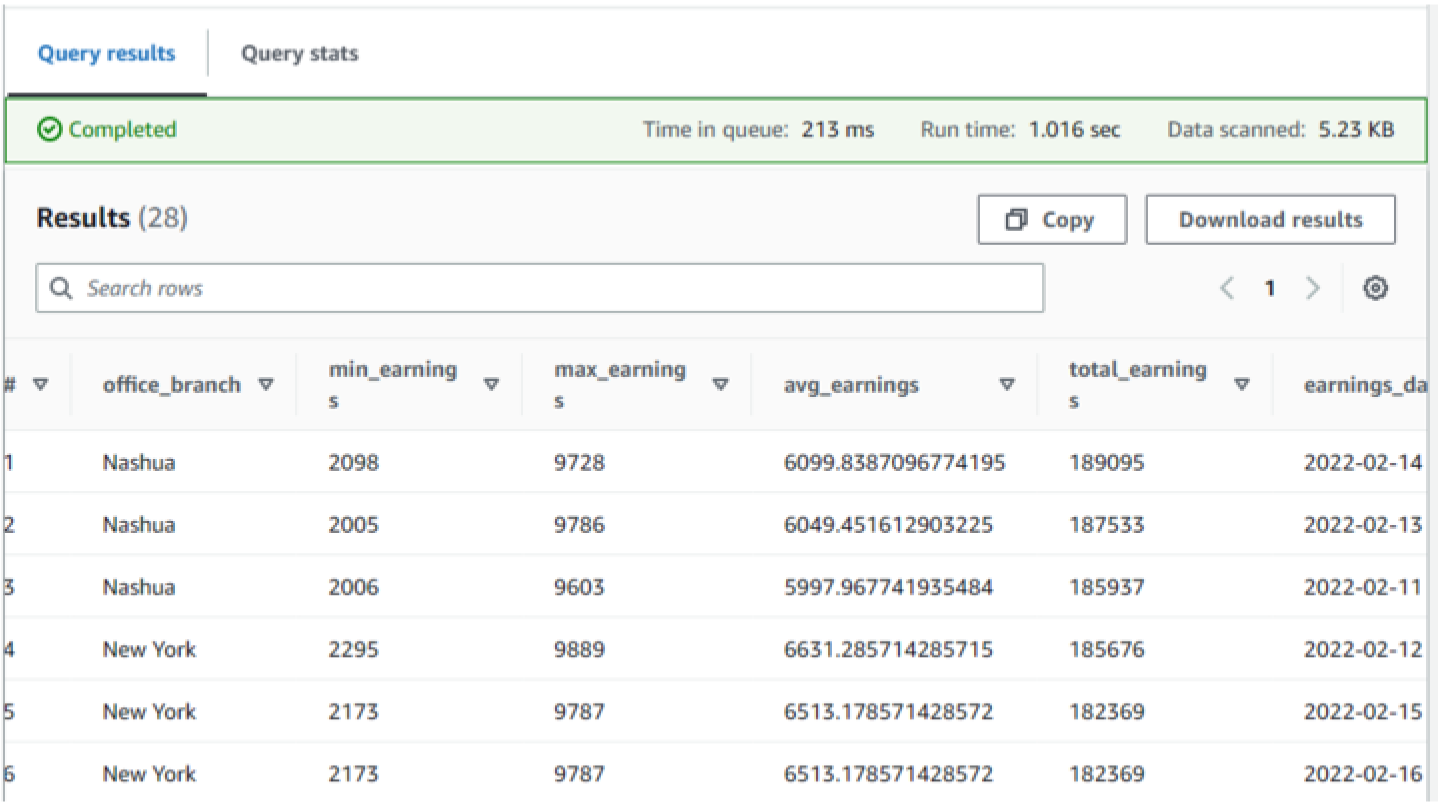


Completed Time in queue: 158 ms Run time: 858 ms Data scanned: 3.75 KB

Results (20)

#	office_branch	min_earnings	max_earnings	avg_earnings	total_earnings	earnings_
1	Nashua	2098	9728	6099.8387096774195	189095	2022-02-
2	Nashua	2005	9786	6049.451612903225	187533	2022-02-
3	Nashua	2006	9603	5997.967741935484	185937	2022-02-
4	New York	2295	9889	6631.285714285715	185676	2022-02-
5	Nashua	2124	9978	5764.5161290322585	178700	2022-02-
6	Nashua	2066	9801	5619.903225806452	174217	2022-02-

After:



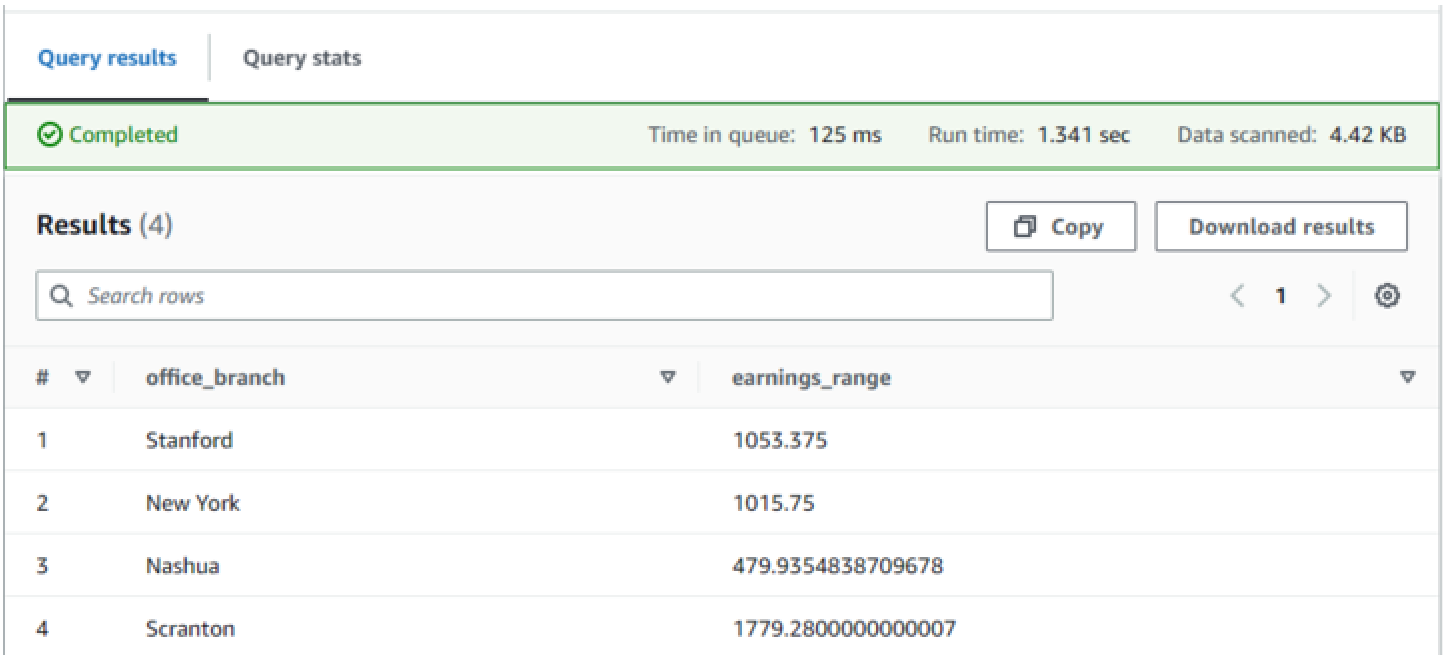
Completed Time in queue: 213 ms Run time: 1.016 sec Data scanned: 5.23 KB

Results (28)

#	office_branch	min_earnings	max_earnings	avg_earnings	total_earnings	earnings_da
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	New York	2173	9787	6513.178571428572	182369	2022-02-15
6	New York	2173	9787	6513.178571428572	182369	2022-02-16

Query 3

Before:

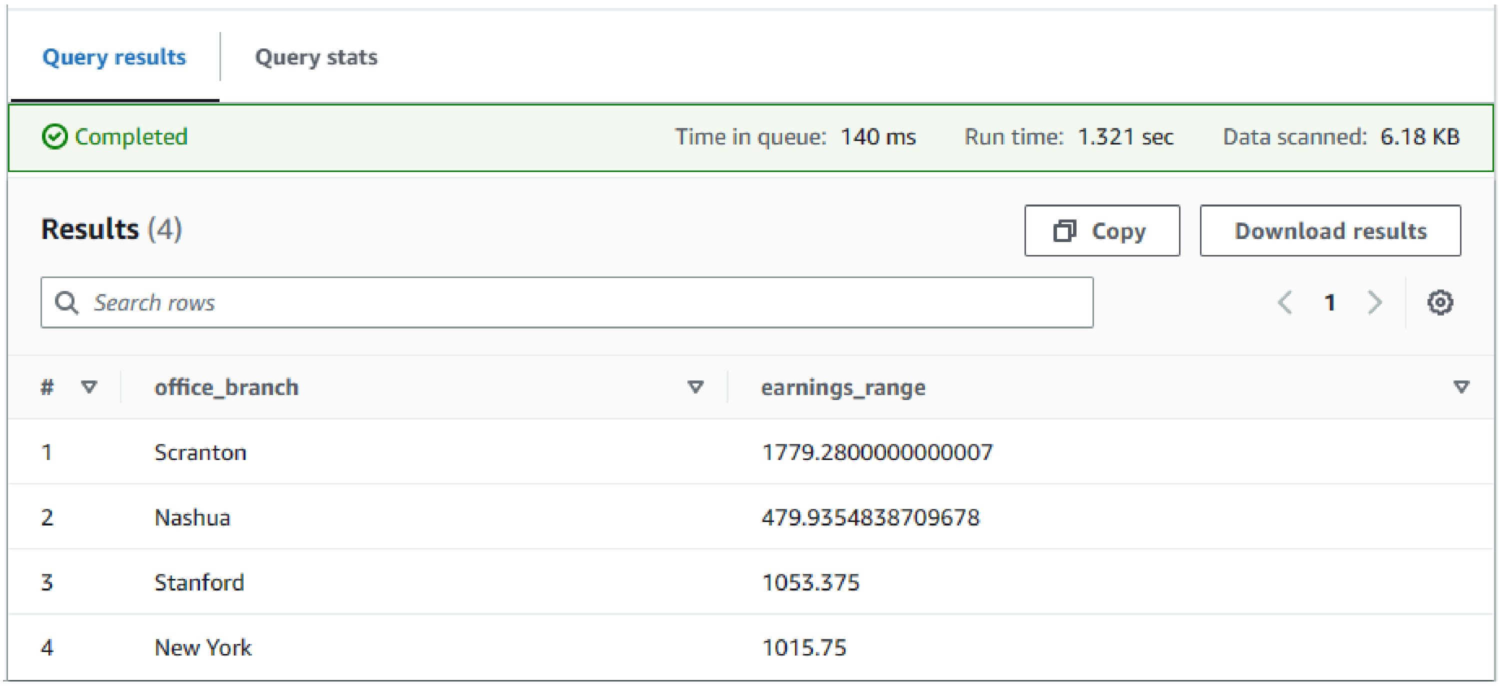


Completed Time in queue: 125 ms Run time: 1.341 sec Data scanned: 4.42 KB

Results (4)

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

After:



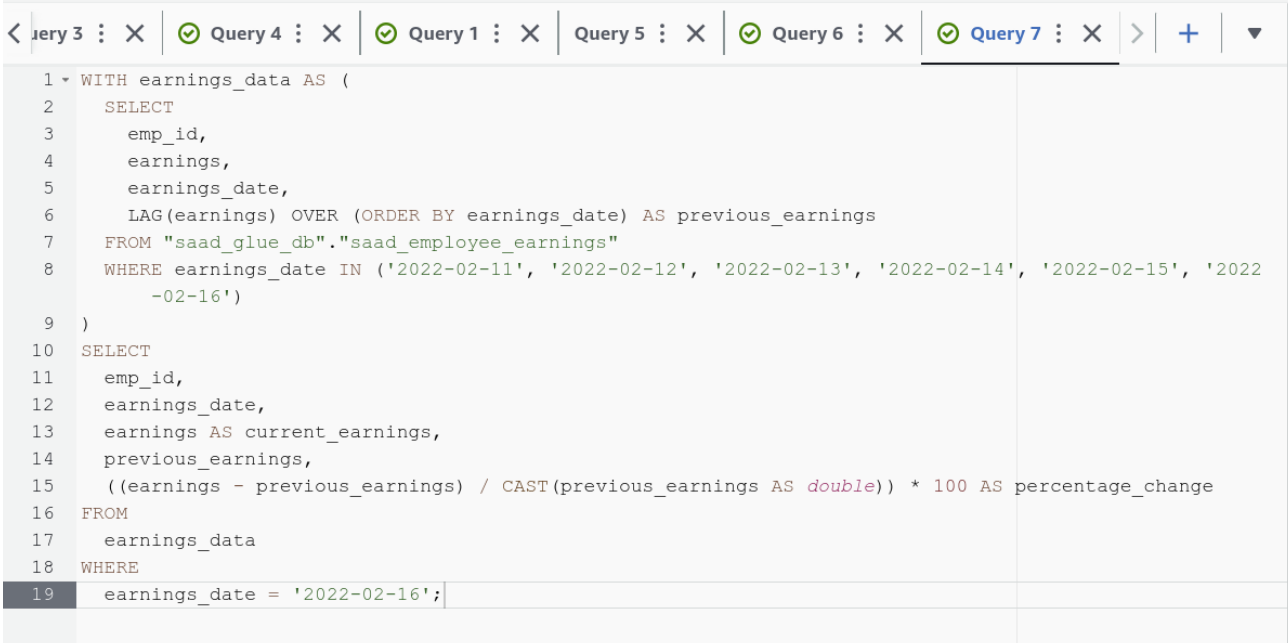
Completed Time in queue: 140 ms Run time: 1.321 sec Data scanned: 6.18 KB

Results (4)

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

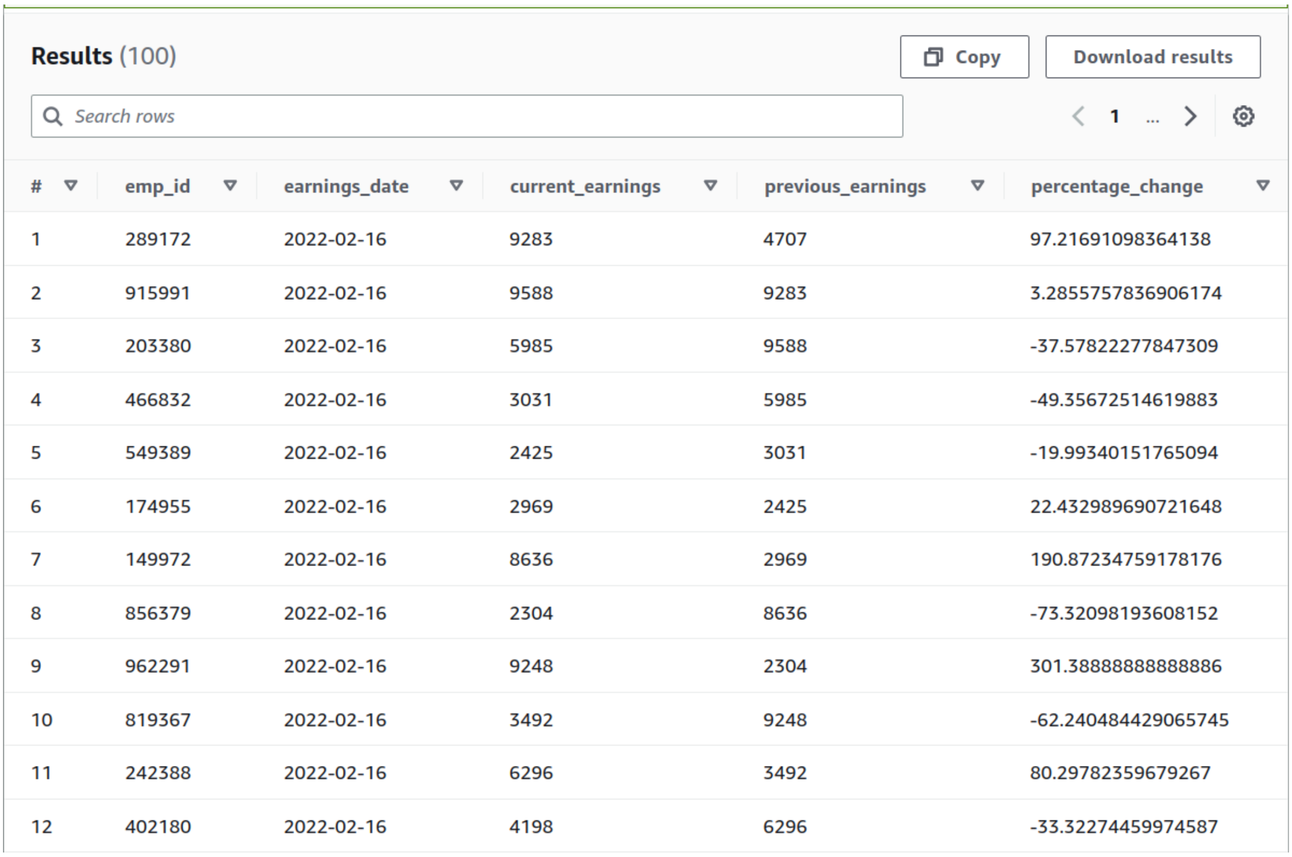
Calculating percentage change in earnings

Now running a new query in Athena that takes in an input day (day 16) and calculates the % change in earnings for every employee from compared to the previous day.



```
WITH earnings_data AS (
  SELECT
    emp_id,
    earnings,
    earnings_date,
    LAG(earnings) OVER (ORDER BY earnings_date) AS previous_earnings
  FROM "saad_glue_db"."saad_employee_earnings"
  WHERE earnings_date IN ('2022-02-11', '2022-02-12', '2022-02-13', '2022-02-14', '2022-02-15', '2022-02-16')
)
SELECT
  emp_id,
  earnings_date,
  earnings AS current_earnings,
  previous_earnings,
  ((earnings - previous_earnings) / CAST(previous_earnings AS double)) * 100 AS percentage_change
FROM
  earnings_data
WHERE
  earnings_date = '2022-02-16';
```

Here are the results:



#	emp_id	earnings_date	current_earnings	previous_earnings	percentage_change
1	289172	2022-02-16	9283	4707	97.21691098364138
2	915991	2022-02-16	9588	9283	3.2855757836906174
3	203380	2022-02-16	5985	9588	-37.57822277847309
4	466832	2022-02-16	3031	5985	-49.35672514619883
5	549389	2022-02-16	2425	3031	-19.99340151765094
6	174955	2022-02-16	2969	2425	22.432989690721648
7	149972	2022-02-16	8636	2969	190.87234759178176
8	856379	2022-02-16	2304	8636	-73.32098193608152
9	962291	2022-02-16	9248	2304	301.38888888888886
10	819367	2022-02-16	3492	9248	-62.240484429065745
11	242388	2022-02-16	6296	3492	80.29782359679267
12	402180	2022-02-16	4198	6296	-33.32274459974587