

**The next big everything**

# Tasks comments

By Denys Murynka

# Task 1

Після запуску скрипта отримуємо результат на скриншоті зліва (це лише частина даних)

```
15 DATE_SUB(CURRENT_DATE(), INTERVAL day DAY) AS action_
16 FROM
17 UNNEST(GENERATE_ARRAY(0, 9)) AS day
18 ),
```

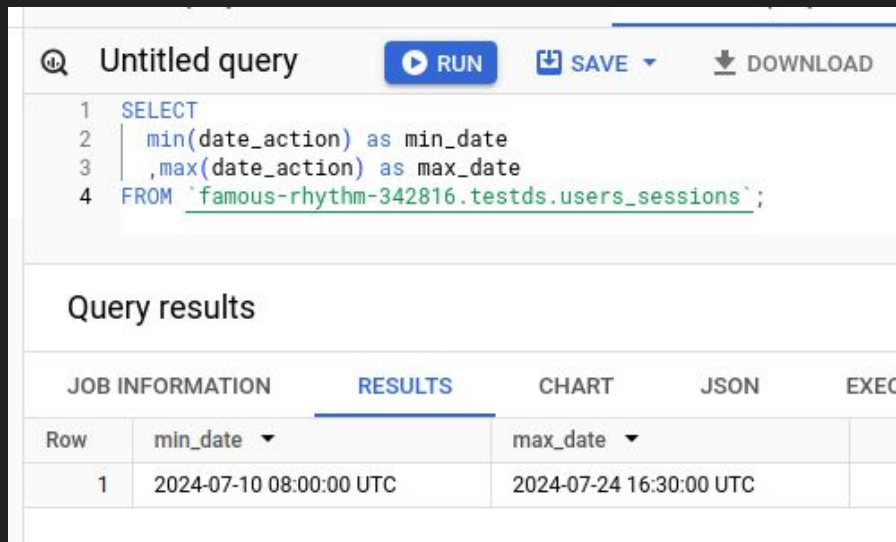
Query results

JOB INFORMATION	RESULTS	CHART	JSON	EXEC
Row	action_date	id_user	hours_online	
1	2024-07-20	101	0.0	
2	2024-07-20	102	0.0	
3	2024-07-20	103	0.0	
4	2024-07-20	104	0.0	
5	2024-07-21	101	0.0	
6	2024-07-21	102	1.5	
7	2024-07-21	103	0.0	
8	2024-07-21	104	0.0	
9	2024-07-22	101	0.0	
10	2024-07-22	102	0.0	
11	2024-07-22	103	0.75	
12	2024-07-22	104	0.0	
13	2024-07-23	101	1.0	
14	2024-07-23	102	0.0	
15	2024-07-23	103	0.0	
16	2024-07-23	104	0.0	
17	2024-07-24	101	0.0	
18	2024-07-24	102	0.5	
19	2024-07-24	103	0.0	
20	2024-07-24	104	0.0	
21	2024-07-25	101	0.0	
22	2024-07-25	102	0.0	
23	2024-07-25	103	0.0	
24	2024-07-25	104	0.0	
25	2024-07-26	101	0.0	
26	2024-07-26	102	0.0	
27	2024-07-26	103	0.0	
28	2024-07-26	104	0.0	

Load more

Попередньо мною було внесено записи для різних днів для чотирьох користувачів. З виконання бачимо, що для кожного дня (з останніх 10 днів) є обрахунок для кожного з чотирьох користувачів. Це завдання було виконане 24-го числа, тому і “останній запис з даними” є на цю дату”

На попередньому слайді “найдавніша” дата була 20-07, та якщо подивитись на скриншот знизу побачим, що в бд це 10-07.



Untitled query

```
1 SELECT
2   min(date_action) as min_date
3   ,max(date_action) as max_date
4 FROM `famous-rhythm-342816.testds.users_sessions`;
```

Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXEC
Row	min_date	max_date			
1	2024-07-10 08:00:00 UTC	2024-07-24 16:30:00 UTC			

Також бачимо, що максимальна дата це 24-07, це додатково доказує, що справді після 24 числа немає записів в бд

```

#####
1
2 -- Generate the last 10 days, including the current day
WITH date_range AS (
3     SELECT
4         DATE_SUB(CURRENT_DATE(), INTERVAL day DAY) AS action_date
5     FROM
6         UNNEST(GENERATE_ARRAY(0, 9)) AS day
7 )
8
9 -- We extract all sessions for the last 10 days, converting the effective date
10 session_times AS (
11     SELECT
12         id_user, |
13         action,
14         DATE(TIMESTAMP_TRUNC(date_action, DAY)) AS action_date,
15         date_action
16     FROM 'DB_NAME.DATA_SET.TABLE_NAME'
17     WHERE date_action >= TIMESTAMP_SUB(CURRENT_TIMESTAMP(), INTERVAL 9 DAY)
18 ),
19
20 -- Extract all open sessions
21 open_sessions AS (
22     SELECT
23         id_user,
24         action_date,
25         date_action AS open_time
26     FROM session_times
27     WHERE action = 'open'
28 ),
29
30 -- We pull out all closed sessions
31 close_sessions AS (
32     SELECT
33         id_user,

```

Коментарі по коду  
наведені як  
коментарі в коді :)  
Там пояснюється,  
за що  
відповідальний який  
блок коду

# Task 2

Q \*Untitled query X

Q \*Untitled query X

dim\_user X

+

☰

🏠

🔍

📄

✍️

💡

🔗

dim\_user

QUERY

SHARE

COPY

SNAPSHOT

DELETE

EXPORT

REFRESH

SCHEMA

DETAILS

PREVIEW

TABLE EXPLORER

PREVIEW

INSIGHTS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALITY

Row	user_sk	user_bk	name	country	city	_valid_from	_valid_to
1	d8baed66-c9e1-42f4-9ca5-832...	788d58fb	Melisa	Canada	Toronto	1000-01-01 00:00:00 UTC	2024-07-25 17:12:27.346446 U...
2	86724574-e50f-4536-b9ba-918...	8e7e4f9a	Ovaness	France	Leon	2023-05-06 00:00:00 UTC	2024-07-25 17:12:27.346446 U...
3	a9709d5a-2c84-4ec4-a58b-b65...	8e7e4f9a	Ovaness	France	Paris	1000-01-01 00:00:00 UTC	2023-05-05 23:59:59 UTC
4	84b40c7b-f369-46ec-83a2-180...	6a94d22b	Kim	Poland	Warsaw	1000-01-01 00:00:00 UTC	9999-12-31 23:59:59 UTC
5	7b852ba5-00fa-4799-83d0-ee6...	23bef18a	Leopold	Ukraine	Ternopil	1000-01-01 00:00:00 UTC	9999-12-31 23:59:59 UTC
6	59146044-1937-4e1d-947c-08...	8e7e4f9a	Ovaness	France	Nice	2024-07-25 17:12:27.346446 U...	9999-12-31 23:59:59 UTC
7	dc20bd3d-ccba-4d69-9393-7e0...	788d58fb	Melisa	USA	New York	2024-07-25 17:12:27.346446 U...	9999-12-31 23:59:59 UTC
8	e7562bef-c266-4557-bf24-890...	5da53bcd	Vasyl	USA	Los-Angeles	2024-07-25 17:12:31.859271 U...	9999-12-31 23:59:59 UTC

NEW record

- 1) Першим ділом було оновлено (перезаписано) за SCD1 імена; значення відображені в рамках в стовпці **name**;
- 2)
  - а) Після цього за SCD2 було оновлено дати valid\_to для користувачів, для яких в нас є нові записи;
  - б) Тепер для цих ж користувачів, було внесено нові записи з відповідними значеннями для valid\_from та valid\_to;
- 3) Після цього було внесено записи по новим користувачам;

Як тест, було внесено ще один запис:

```
INSERT INTO `DB_NAME.DATA_SET.stg_user`(user_bk, name, country, city)
VALUES ('8e7e4f9a', 'Ovaness', 'France', 'Paris');
```

```
97 where user_bk = '8e7e4f9a'
98 order by _valid_from asc;
99
```

Press Alt+F1 for Accessibility Options

## Query results

 SAVE RESULTS ▾

 EXPLORE DATA ▾



JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	user_sk ▾	user_bk ▾	name ▾	country ▾	city ▾	_valid_from ▾	_valid_to ▾
1	a9709d5a-2c84-4ec4-a58b-b65...	8e7e4f9a	Ovaness	France	Paris	1000-01-01 00:00:00 UTC	2023-05-05 23:59:59 UTC
2	86724574-e50f-4536-b9ba-918...	8e7e4f9a	Ovaness	France	Leon	2023-05-06 00:00:00 UTC	2024-07-25 17:12:27.346446 U...
3	59146044-1937-4e1d-947c-08...	8e7e4f9a	Ovaness	France	Nice	2024-07-25 17:12:27.346446 U...	2024-07-25 18:25:57.016202 U...
4	3412b52c-b073-454a-85a4-b6...	8e7e4f9a	Ovaness	France	Paris	2024-07-25 18:25:57.016202 U...	9999-12-31 23:59:59 UTC

Prev  
New



# Task 3

# Як запустити контейнер:

# Open a terminal (Command Prompt or PowerShell for Windows, Terminal for macOS or Linux)

# Ensure Git is installed

# Visit <https://git-scm.com> to download and install console Git if not already installed

# Clone the repository

```
git clone https://github.com/Abblix/Oidc.Server.git
```

# Ensure Docker is installed

# Visit <https://www.docker.com/get-started/> to download and install console Docker if not already installed

# Run the command below to check if the Docker is installed

```
docker -v
```

# Navigate to the project directory

```
cd Task3/docker-airflow currency etl
```

# Run the command below to get up and running needed services

```
docker-compose up
```

# Open the link below to find the DAG on UI

```
http://0.0.0.0:8080/dags/exchange_rate_dag
```


# Login to Airflow

```
Login and passw: airflow
```

\* Ці ж коментарі, можна знайти в README файлі

> Task2

> Task3

 README

1. Було вирішено зберігати `Currency`, `Rate` та `Date` як дані файлу. Таким чином, влюбий момент часу, ми можемо приконектитись до “потрібного” файлу і отримати курс за потрібний день для конкретної валюти.

\*звісно, можна конектитись до фолдера і юніюніти всі файли

2. Також було вирішено зберігати файли в такій структурі:  
Рік-місяць



```
data/2024/July
├── exchange_rates_2024-07-01.parquet
```

3. При потребі джоб можна розширити, щоб отримувати погодинні/більше детальні значення та для цього free trial не підходить, щось по типу time-series за підпискою тільки
4. Якщо в джобі `start_date` замінити на `2024-06-01` то допишуться дані за червень в коректний фолдер
5. Для тесту можна видалити дані з папки `data`, після запуску джобу усі дані знову запишуться