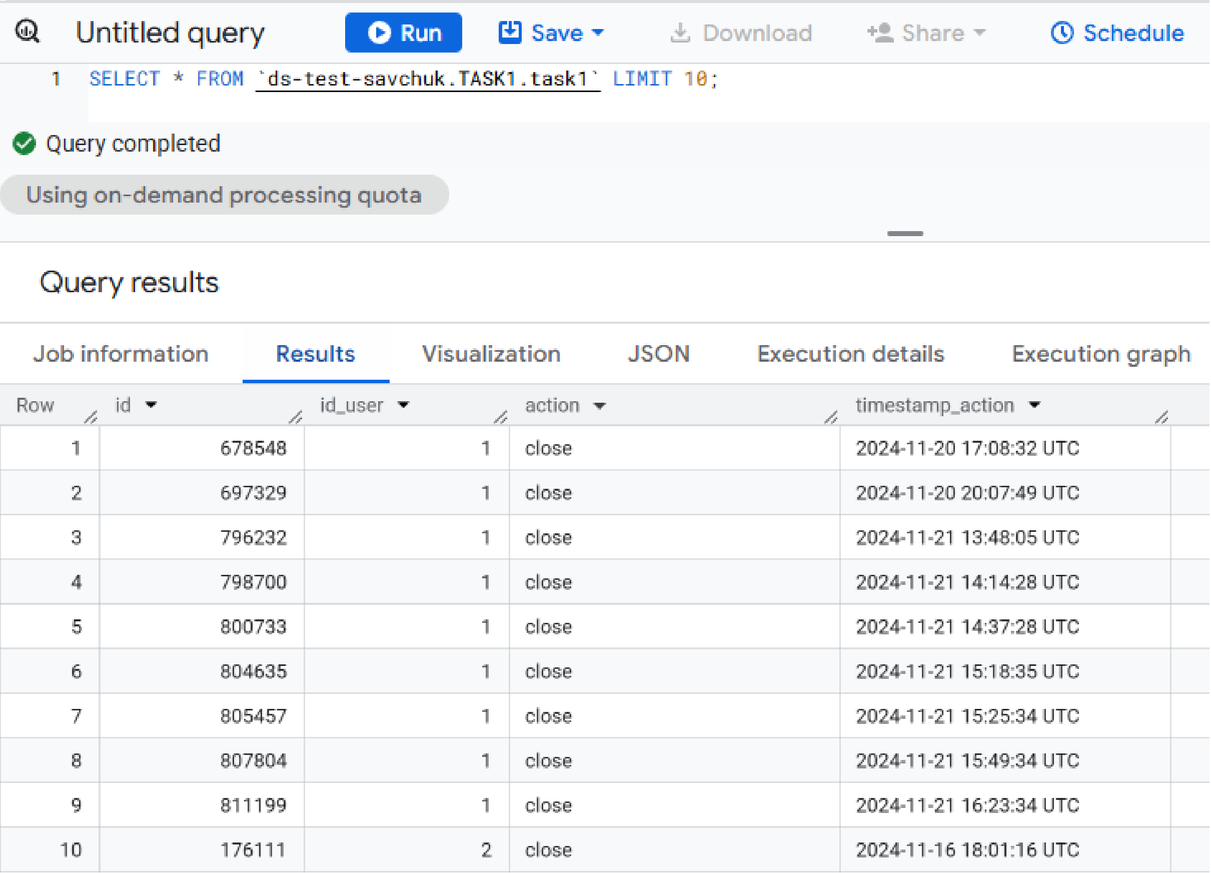
The task is to **calculate how much time (in hours)** users spend on the website over the **last 10 days**.

We have the following data:

1. **id** (int) – unique record identifier.
2. **id_user** (int) – unique user identifier.
3. **action** (string) – a field indicating the opening or closing of a session (two possible values: **'open'** and **'close'**).
4. **timestamp_action** (timestamp) – the date and time of the session opening/closing (depending on the value in the 'action' field) for online presence.



```
SELECT * FROM `ds-test-savchuk.TASK1.task1` LIMIT 10;
```

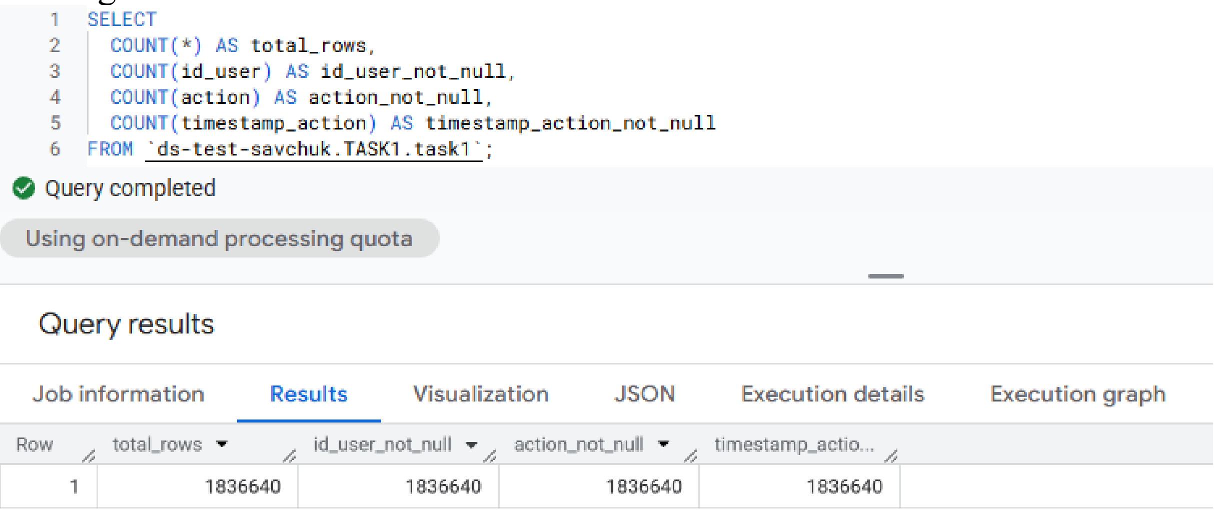
Query completed

Using on-demand processing quota

Query results

Row	id	id_user	action	timestamp_action
1	678548	1	close	2024-11-20 17:08:32 UTC
2	697329	1	close	2024-11-20 20:07:49 UTC
3	796232	1	close	2024-11-21 13:48:05 UTC
4	798700	1	close	2024-11-21 14:14:28 UTC
5	800733	1	close	2024-11-21 14:37:28 UTC
6	804635	1	close	2024-11-21 15:18:35 UTC
7	805457	1	close	2024-11-21 15:25:34 UTC
8	807804	1	close	2024-11-21 15:49:34 UTC
9	811199	1	close	2024-11-21 16:23:34 UTC
10	176111	2	close	2024-11-16 18:01:16 UTC

By the way, a positive aspect is that the data is complete — there are no empty or missing values in this dataset.



```
SELECT
  COUNT(*) AS total_rows,
  COUNT(id_user) AS id_user_not_null,
  COUNT(action) AS action_not_null,
  COUNT(timestamp_action) AS timestamp_action_not_null
FROM `ds-test-savchuk.TASK1.task1`;
```

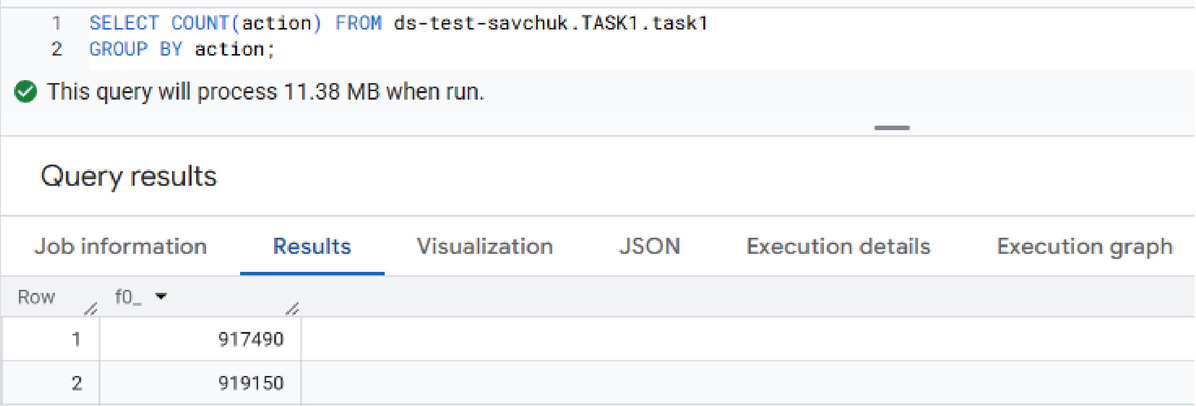
Query completed

Using on-demand processing quota

Query results

Row	total_rows	id_user_not_null	action_not_null	timestamp_actio...
1	1836640	1836640	1836640	1836640

The main challenge is that there are users who opened sessions but did not close them. This can be explained by the fact that the website remained open, although in reality they were no longer actively using it. There are 1,660 such sessions in the dataset, which is rather an exception, considering that the total number of rows is 1,836,640. This is less than 1%.

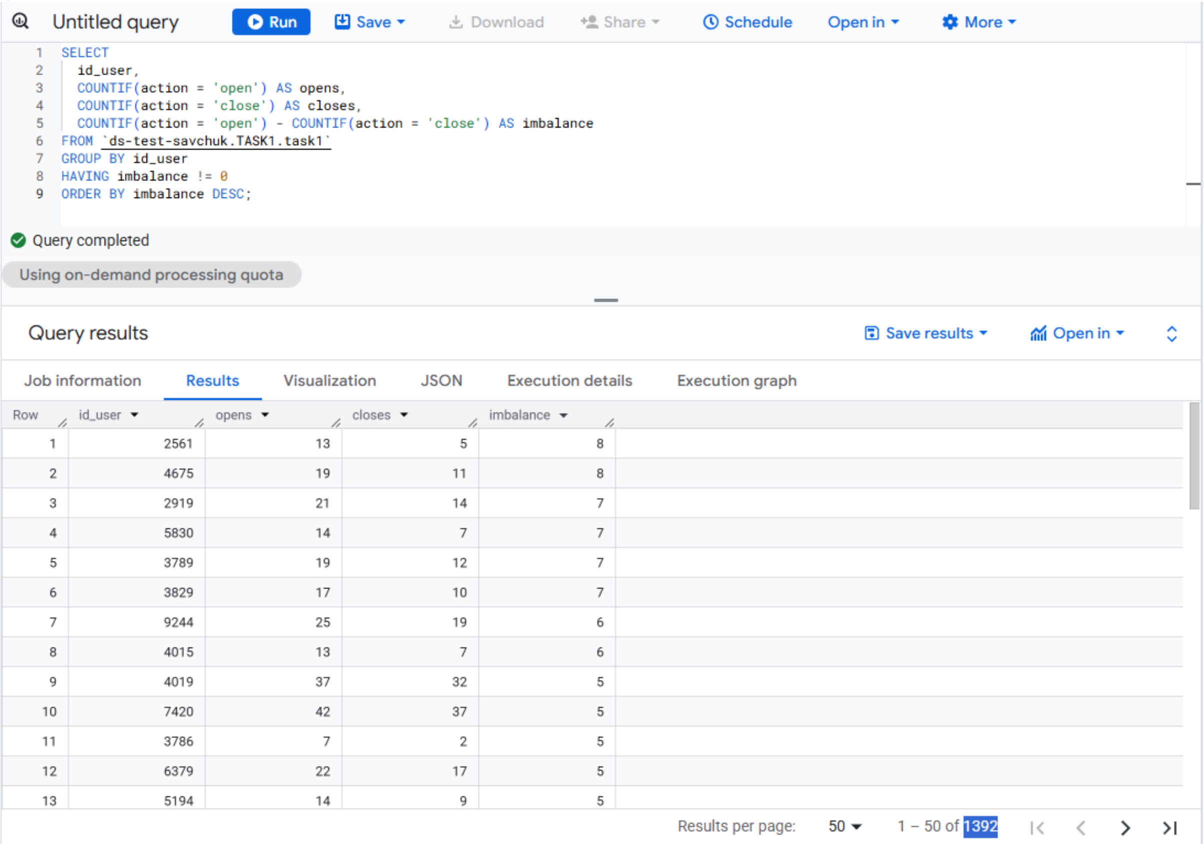


```
SELECT COUNT(action) FROM ds-test-savchuk.TASK1.task1
GROUP BY action;
```

This query will process 11.38 MB when run.

Query results

Row	f0_
1	917490
2	919150



```
SELECT
  id_user,
  COUNTIF(action = 'open') AS opens,
  COUNTIF(action = 'close') AS closes,
  COUNTIF(action = 'open') - COUNTIF(action = 'close') AS imbalance
FROM `ds-test-savchuk.TASK1.task1`
GROUP BY id_user
HAVING imbalance != 0
ORDER BY imbalance DESC;
```

Query completed

Using on-demand processing quota

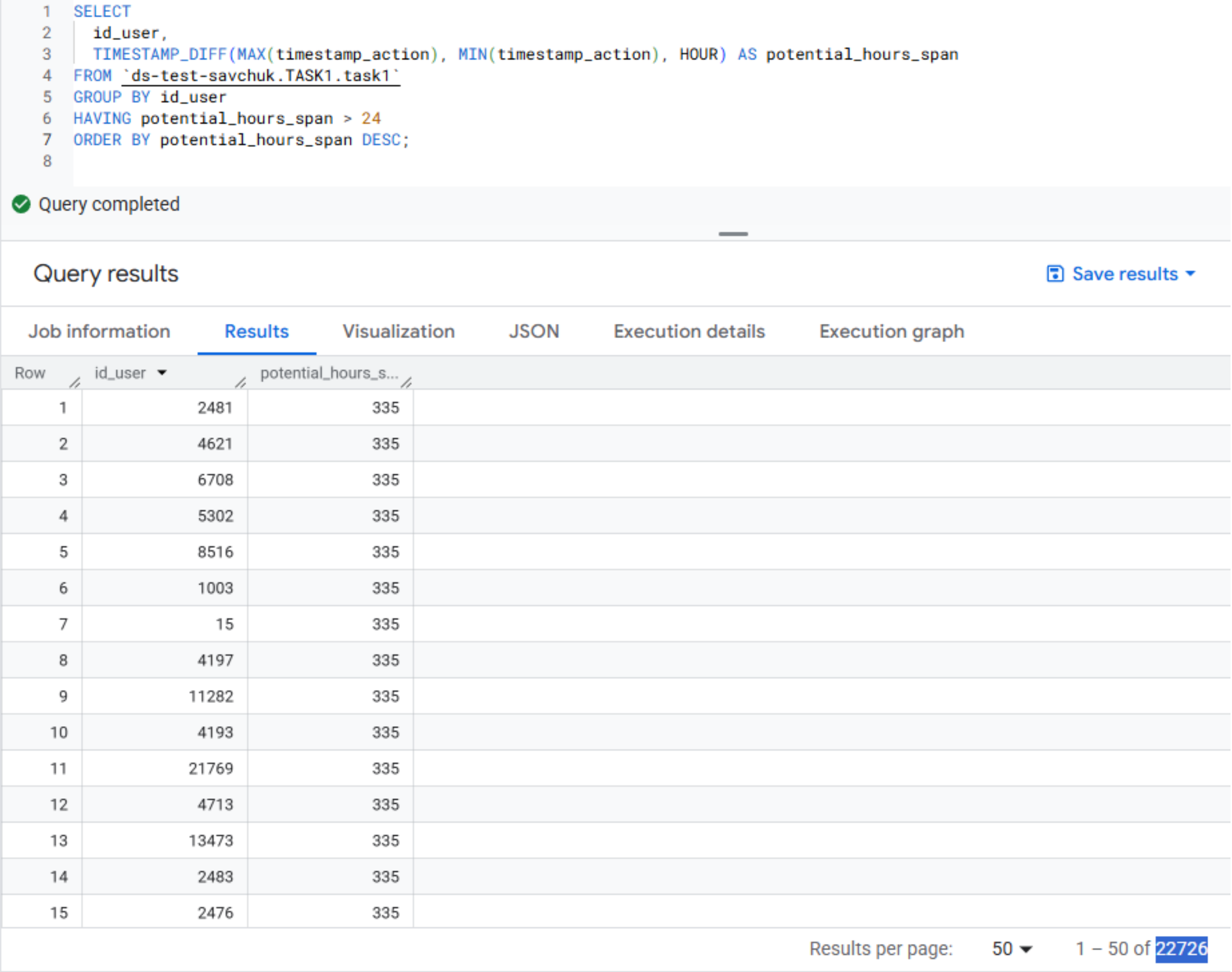
Query results

Row	id_user	opens	closes	imbalance
1	2561	13	5	8
2	4675	19	11	8
3	2919	21	14	7
4	5830	14	7	7
5	3789	19	12	7
6	3829	17	10	7
7	9244	25	19	6
8	4015	13	7	6
9	4019	37	32	5
10	7420	42	37	5
11	3786	7	2	5
12	6379	22	17	5
13	5194	14	9	5

Results per page: 50 ▾ 1 – 50 of 1392

Such events can significantly affect the final result, as their duration will be an outlier and the calculated statistical metrics will be incorrect. Depending on the goal of the task and the further manipulations planned, these data points should either be completely excluded from the dataset (less than 1% will not affect the overall result, and we understand that this is an outlier rather than normal behavior), or marked so that during further calculations we can choose whether to include them or not. It is worth noting that an additional column in a large dataset will take up storage space and slow down query execution. If resource optimization is required, this option is not advisable. Another possible approach is to replace these outliers with the mean or median value calculated separately for each user.

We will also calculate the number of cases where a user stays on the website for more than 24 hours. This does not make sense either from the perspective of the task (to calculate the number of hours per day) or from a logical point of view, since a person cannot be productive for more than a day. In fact, the filtering threshold could even be reduced, for example, to 12 hours.

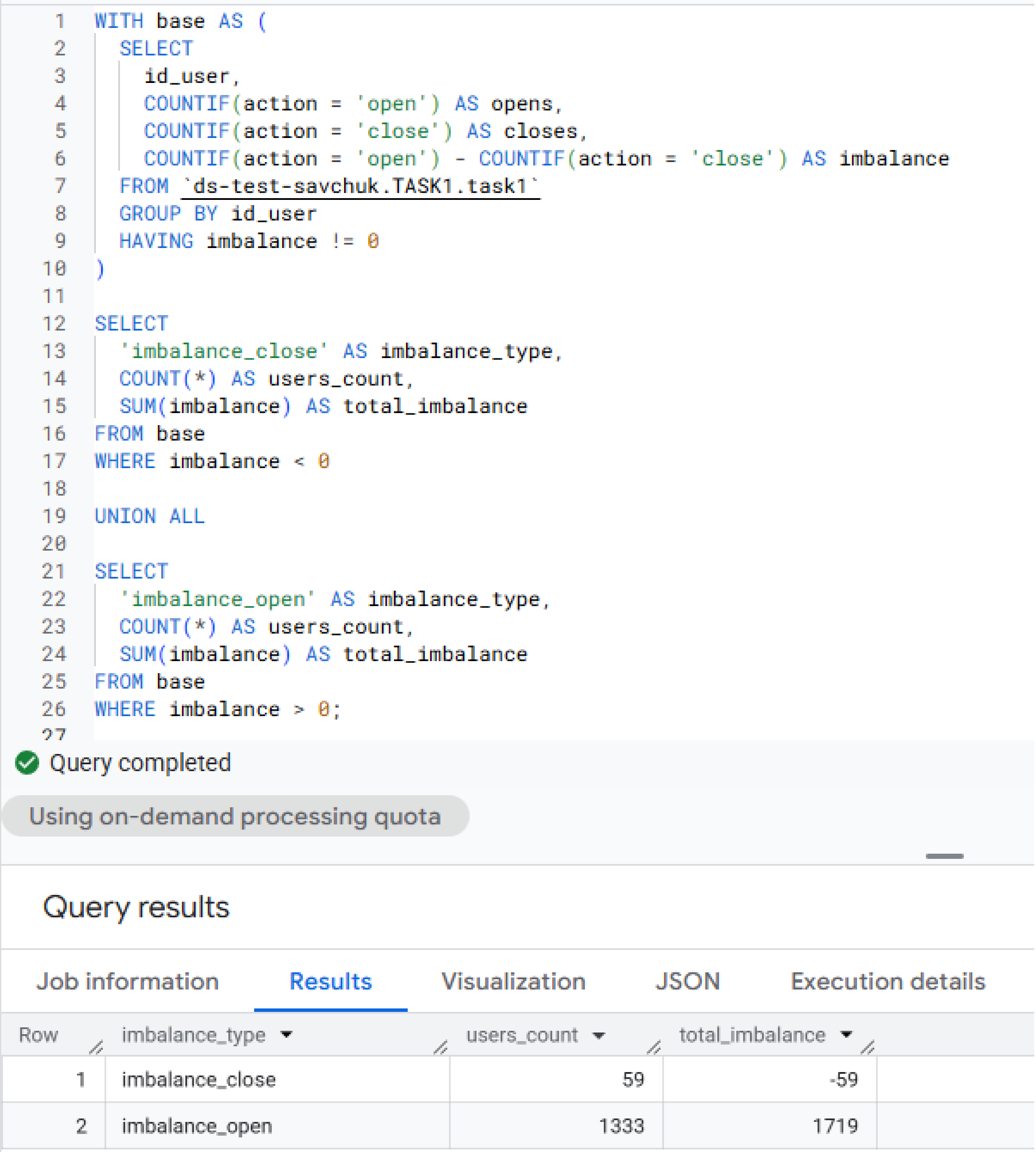


```
SELECT
  id_user,
  TIMESTAMP_DIFF(MAX(timestamp_action), MIN(timestamp_action), HOUR) AS potential_hours_span
FROM `ds-test-savchuk.TASK1.task1`
GROUP BY id_user
HAVING potential_hours_span > 24
ORDER BY potential_hours_span DESC;
```

Row	id_user	potential_hours_s...
1	2481	335
2	4621	335
3	6708	335
4	5302	335
5	8516	335
6	1003	335
7	15	335
8	4197	335
9	11282	335
10	4193	335
11	21769	335
12	4713	335
13	13473	335
14	2483	335
15	2476	335

Results per page: 50 1 – 50 of 22726

An interesting observation is that there are cases where the number of close events exceeds the number of open events, which is logically impossible. This is either caused by duplicates or by a technical issue.



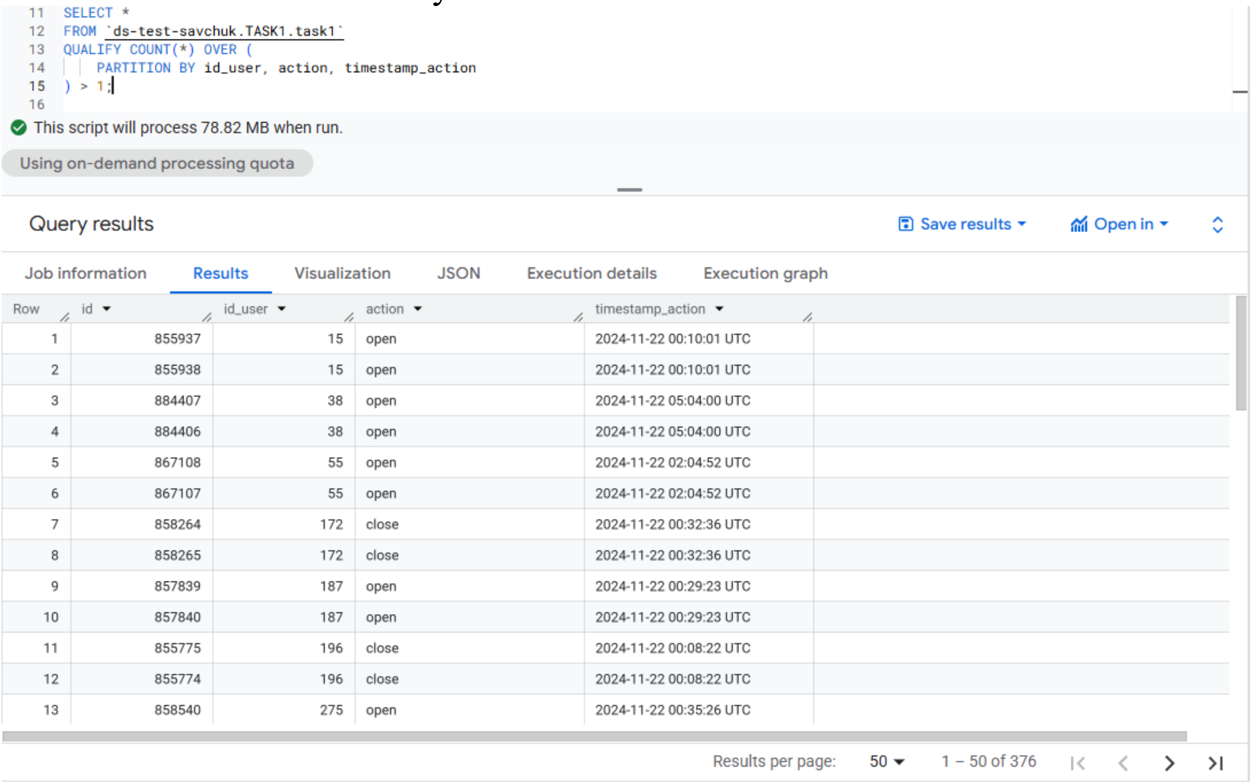
```
WITH base AS (
  SELECT
    id_user,
    COUNTIF(action = 'open') AS opens,
    COUNTIF(action = 'close') AS closes,
    COUNTIF(action = 'open') - COUNTIF(action = 'close') AS imbalance
  FROM `ds-test-savchuk.TASK1.task1`
  GROUP BY id_user
  HAVING imbalance != 0
)

SELECT
  'imbalance_close' AS imbalance_type,
  COUNT(*) AS users_count,
  SUM(imbalance) AS total_imbalance
FROM base
WHERE imbalance < 0

UNION ALL

SELECT
  'imbalance_open' AS imbalance_type,
  COUNT(*) AS users_count,
  SUM(imbalance) AS total_imbalance
FROM base
WHERE imbalance > 0;
```

Query completed

Using on-demand processing quota

Row	imbalance_type	users_count	total_imbalance
1	imbalance_close	59	-59
2	imbalance_open	1333	1719

A user could not perform two identical actions at the same time. Even if this did happen, it is more likely a random occurrence rather than a real need, and such data does not make sense for analysis.

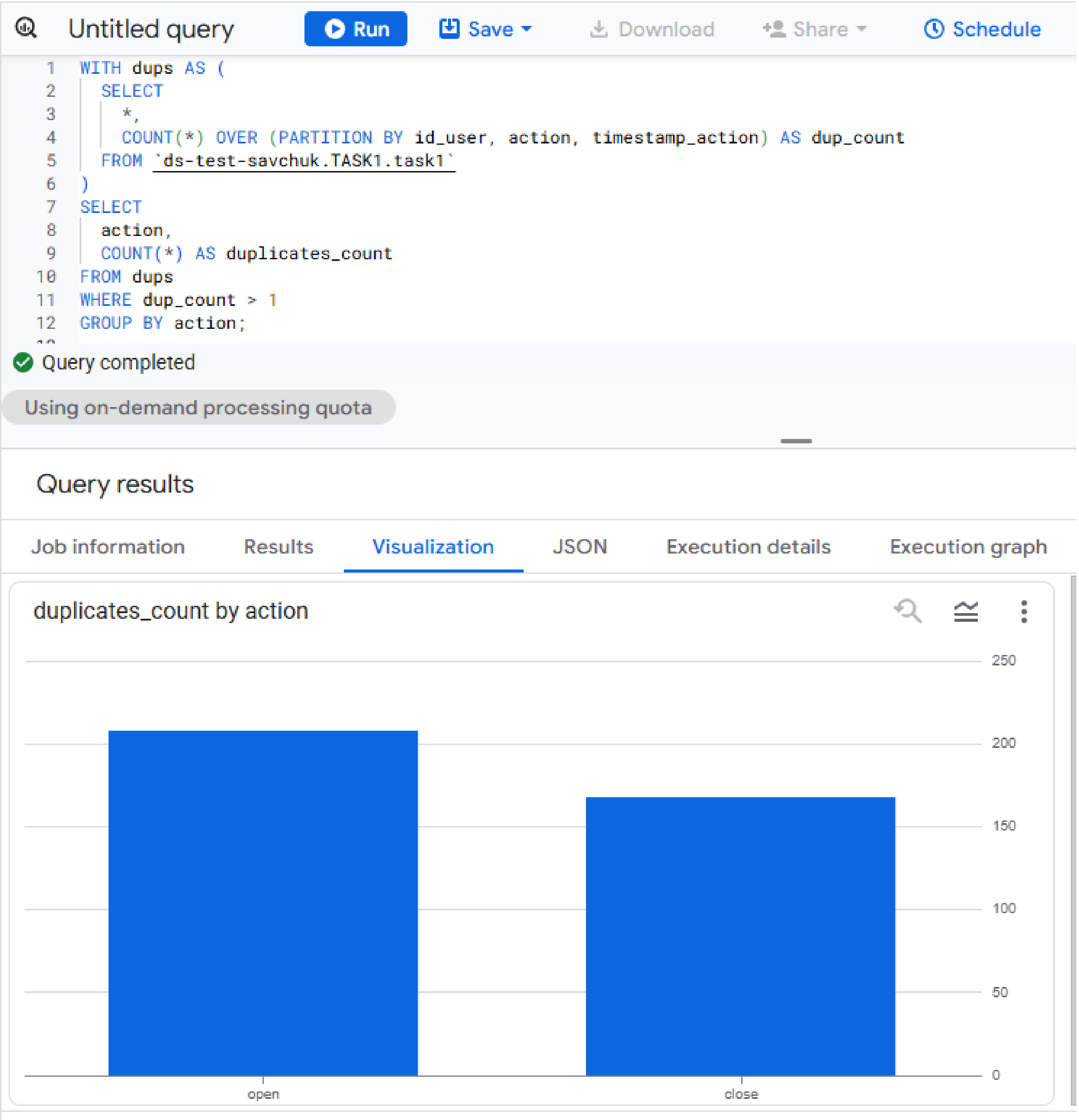


```sql
SELECT *
FROM `ds-test-savchuk.TASK1.task1`
QUALIFY COUNT(*) OVER (
    PARTITION BY id_user, action, timestamp_action
) > 1;
```

This script will process 78.82 MB when run.

Row	id	id_user	action	timestamp_action
1	855937	15	open	2024-11-22 00:10:01 UTC
2	855938	15	open	2024-11-22 00:10:01 UTC
3	884407	38	open	2024-11-22 05:04:00 UTC
4	884406	38	open	2024-11-22 05:04:00 UTC
5	867108	55	open	2024-11-22 02:04:52 UTC
6	867107	55	open	2024-11-22 02:04:52 UTC
7	858264	172	close	2024-11-22 00:32:36 UTC
8	858265	172	close	2024-11-22 00:32:36 UTC
9	857839	187	open	2024-11-22 00:29:23 UTC
10	857840	187	open	2024-11-22 00:29:23 UTC
11	855775	196	close	2024-11-22 00:08:22 UTC
12	855774	196	close	2024-11-22 00:08:22 UTC
13	858540	275	open	2024-11-22 00:35:26 UTC

Results per page: 50 1 – 50 of 376



```sql
WITH dups AS (
  SELECT
    *,
    COUNT(*) OVER (PARTITION BY id_user, action, timestamp_action) AS dup_count
  FROM `ds-test-savchuk.TASK1.task1`
)
SELECT
  action,
  COUNT(*) AS duplicates_count
FROM dups
WHERE dup_count > 1
GROUP BY action;
```

Query completed

Using on-demand processing quota

Query results

To summarize the findings of the preliminary data analysis:

- the data contains duplicates that should be removed;
- there are events that are logically impossible (closing a session without opening it);
- we assume that there are users who forgot to end a session, or that a technical issue occurred. In any case, such data should not be included in further analysis — it should either be removed, replaced, or flagged.