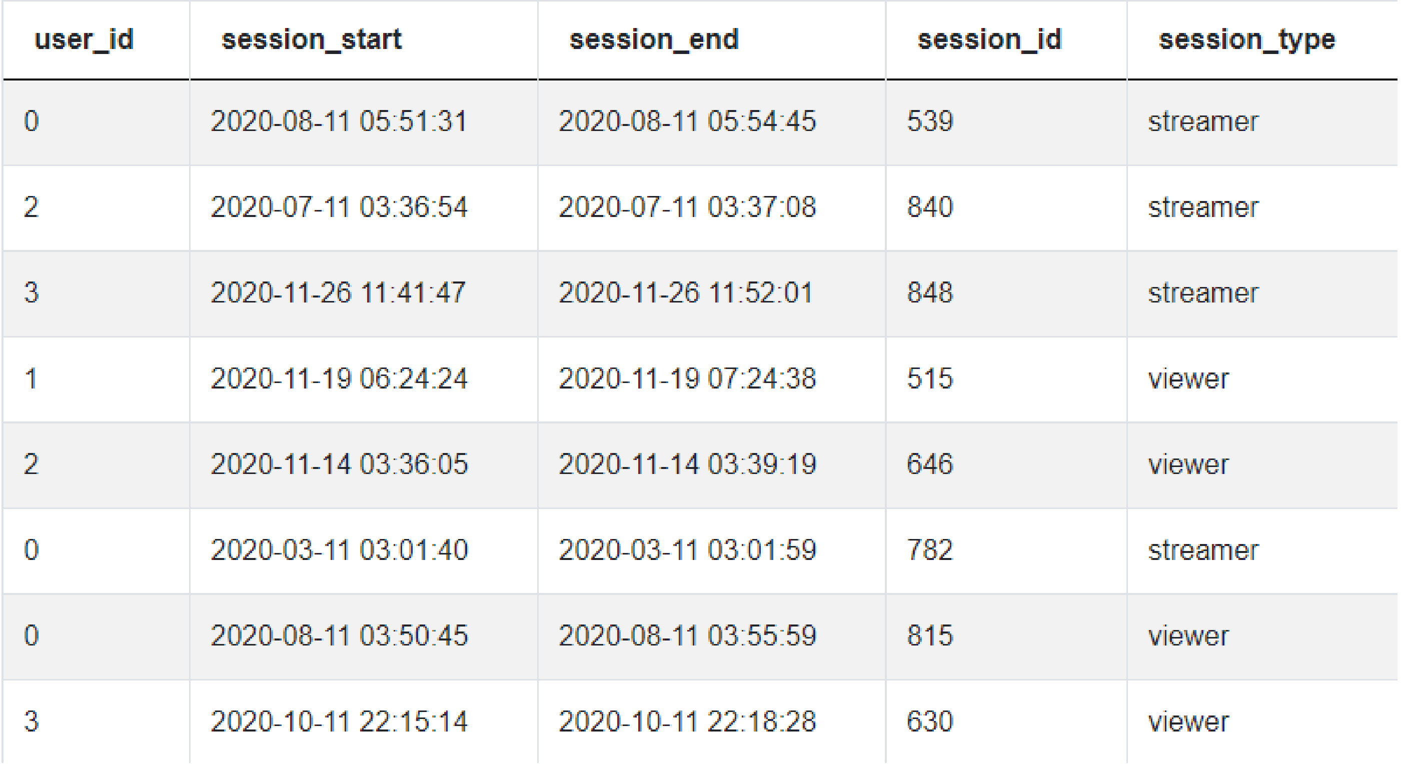
Coding Test by Habrie (Optional)

Problem #1– RANK( ) Function

As you can see, the table **twitch_sessions** includes – among the others – an **user_id** field that is not unique, but is instead recorded every time a specific user starts a new session. The **session_type** field will tell you if that specific user_id was a viewer or a streamer at the time the session started



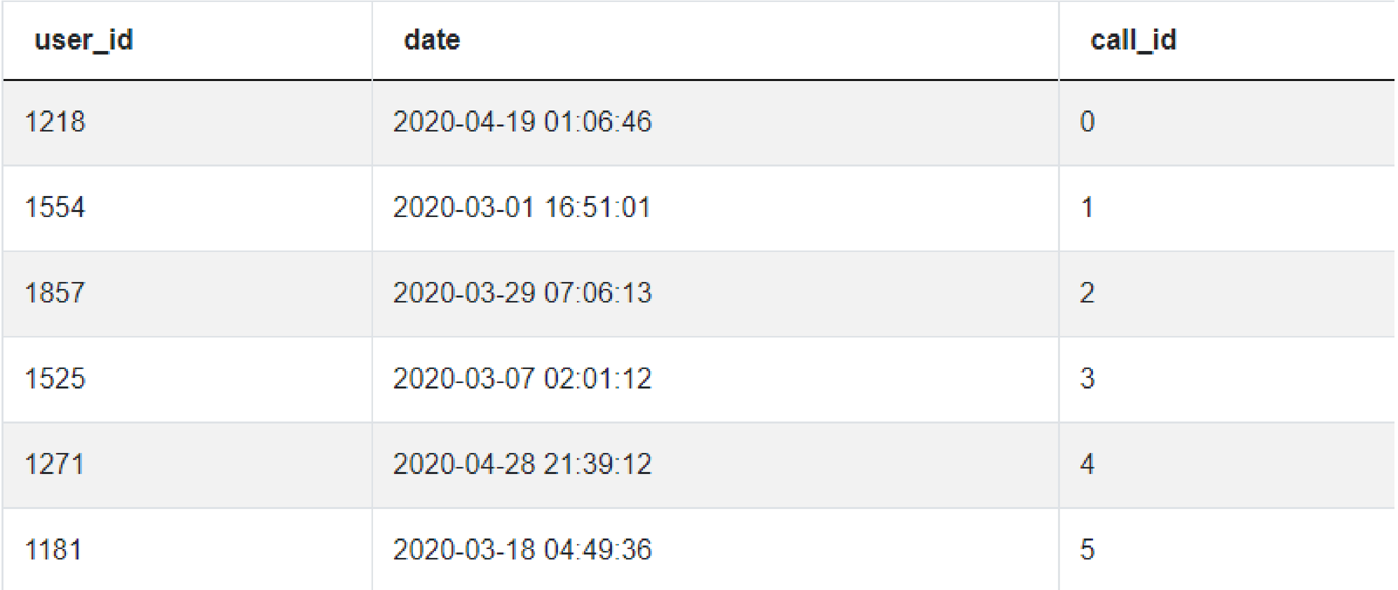
user_id	session_start	session_end	session_id	session_type
0	2020-08-11 05:51:31	2020-08-11 05:54:45	539	streamer
2	2020-07-11 03:36:54	2020-07-11 03:37:08	840	streamer
3	2020-11-26 11:41:47	2020-11-26 11:52:01	848	streamer
1	2020-11-19 06:24:24	2020-11-19 07:24:38	515	viewer
2	2020-11-14 03:36:05	2020-11-14 03:39:19	646	viewer
0	2020-03-11 03:01:40	2020-03-11 03:01:59	782	streamer
0	2020-08-11 03:50:45	2020-08-11 03:55:59	815	viewer
3	2020-10-11 22:15:14	2020-10-11 22:18:28	630	viewer

Find the so and so number of sessions completed by users that had their very first session as viewers. You should just return their user ID and number of sessions.

Coding Test by Habrie (Optional)

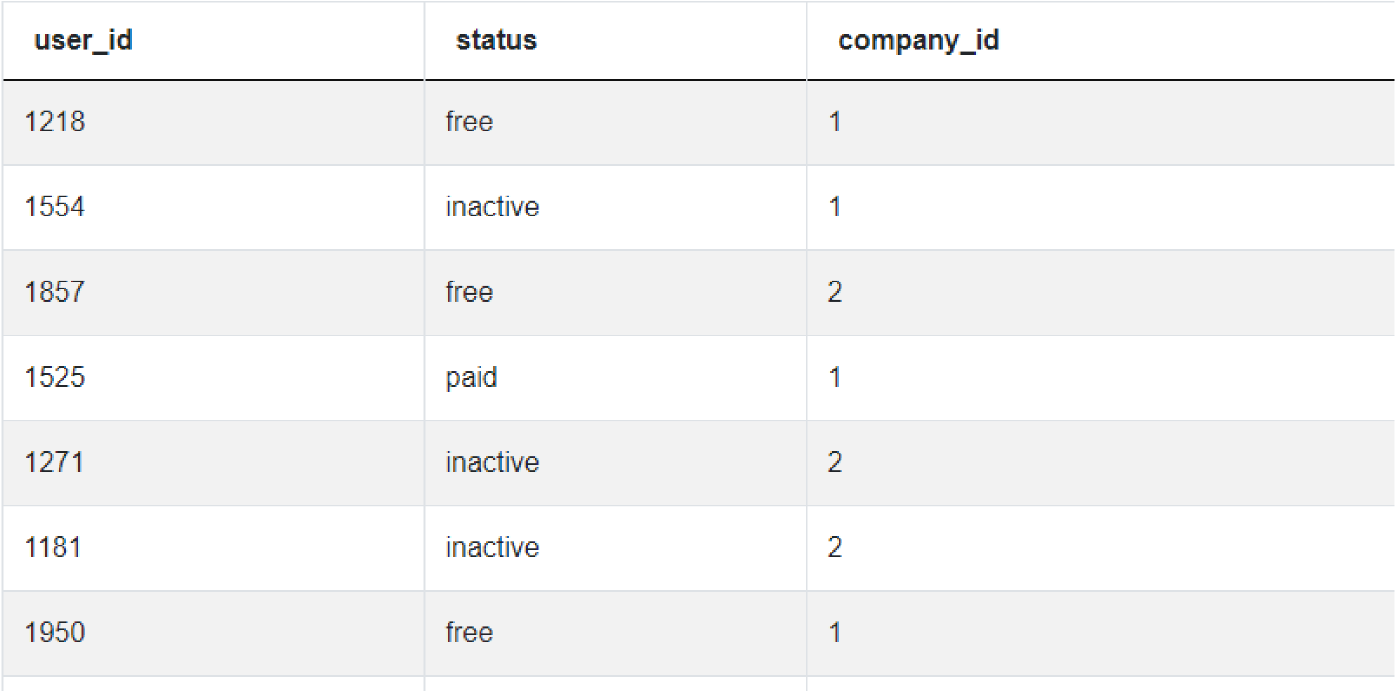
Problem #2– DENSE_RANK( ) Function

In this problem you are going to need two tables. The first is rc_calls that includes details on the calls received by our customer service department, that supports both our parent company (company 1) and its child company (company 2). The second table rc_users includes details on the users and the specific company they contacted. We wish to learn more about users skewing our contact rate.



user_id	date	call_id
1218	2020-04-19 01:06:46	0
1554	2020-03-01 16:51:01	1
1857	2020-03-29 07:06:13	2
1525	2020-03-07 02:01:12	3
1271	2020-04-28 21:39:12	4
1181	2020-03-18 04:49:36	5

First few rows from rc_calls table.



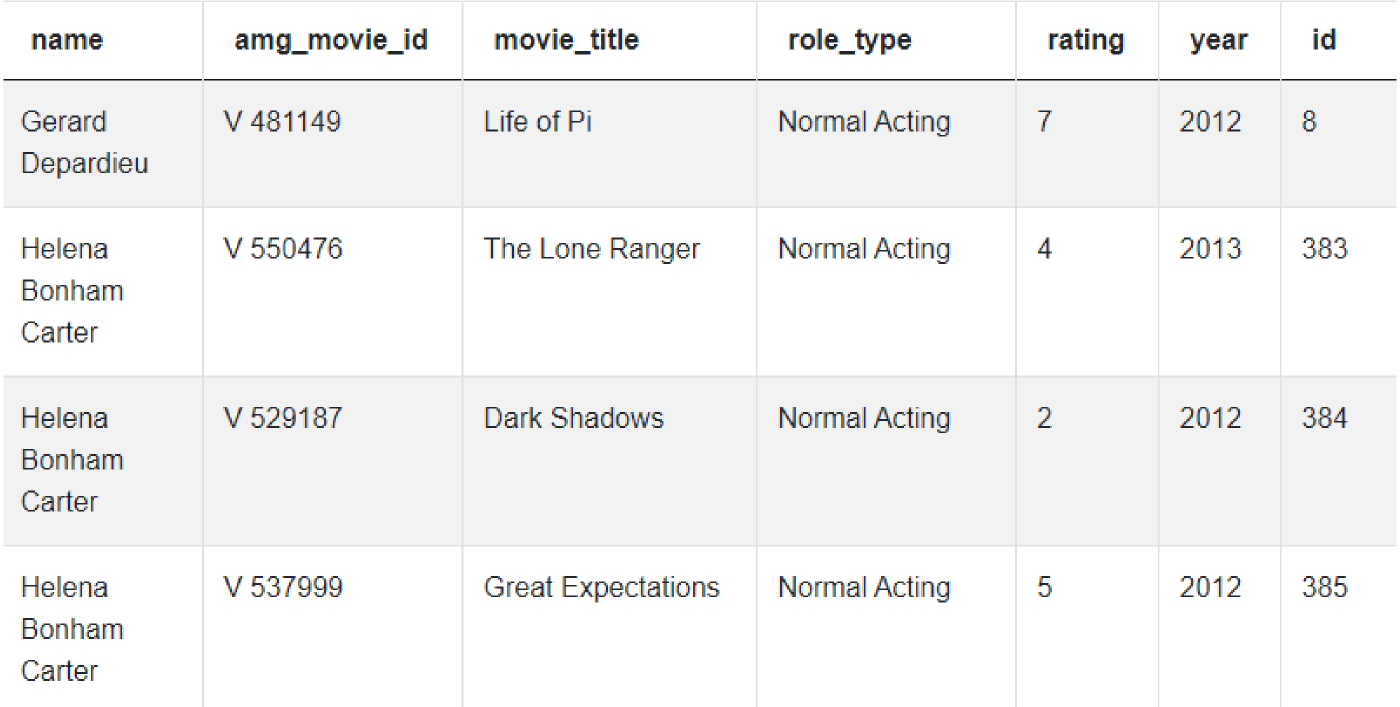
user_id	status	company_id
1218	free	1
1554	inactive	1
1857	free	2
1525	paid	1
1271	inactive	2
1181	inactive	2
1950	free	1

To achieve this, for each company, I ask you to return the top 2 users that called customer service the most. You should output the company_id, user_id, and the user's rank. If there are multiple users with the same rank, keep all of them.

Coding Test by Habrie (Optional)

Problem #3 – ROW_NUMBER() Function

In this last exercise you are going to work with the nominee_filmography table. This is a list of actors (name) and movies in which they have been involved and a rating for their acting. While providing a solution, bear in mind that names might appear multiple times, whereas the movie id field is unique.



name	amg_movie_id	movie_title	role_type	rating	year	id
Gerard Depardieu	V 481149	Life of Pi	Normal Acting	7	2012	8
Helena Bonham Carter	V 550476	The Lone Ranger	Normal Acting	4	2013	383
Helena Bonham Carter	V 529187	Dark Shadows	Normal Acting	2	2012	384
Helena Bonham Carter	V 537999	Great Expectations	Normal Acting	5	2012	385

You should calculate the rating from the second latest film and the average lifetime rating for each one of the actors and movies they acted in. Output a list of actors, their rating from the second latest film, the average lifetime rating and the difference between the two ratings (second last - average).