DIFFICULT

Find the top business categories based on the total number of reviews. Output the category along with the total number of reviews. Order by total reviews in descending order.

yelp_business(business_id:varchar, name:varchar, neighborhood:varchar, address:varchar, city:varchar, state:varchar, postal_code:varchar, latitude:float, longitude:float, stars:float, review_count:int, is_open:int, categories:varchar)

NOTE: categories column is a multi-valued column with values separated by semi-colon.

HINT: UNNEST function converts a multi-valued column to different rows.

ANSWER

```
WITH cats AS
(SELECT UNNEST(STRING_TO_ARRAY(categories, ';')) AS category, review_count FROM yelp_business)

SELECT category, SUM(review_count) AS review_cnt
FROM cats GROUP BY category ORDER BY review_cnt DESC
```

What is the overall friend acceptance rate by date? Your output should have the rate of acceptances by the date the request was sent. Order by the earliest date to latest.

Assume that each friend request starts by a user sending (i.e., user_id_sender) a friend request to another user (i.e., user_id_receiver) that's logged in the table with action = 'sent'. If the request is accepted, the table logs action = 'accepted'. If the request is not accepted, no record of action = 'accepted' is logged.

fb_friend_requests(user_id_sender:varchar, user_id_receiver:varchar, date:datetime, action:varchar)

ANSWER

```
WITH
sent_cte AS (SELECT date, user_id_sender, user_id_receiver FROM fb_friend_requests WHERE action='sent' ),
accepted_cte AS (SELECT date, user_id_sender, user_id_receiver FROM fb_friend_requests WHERE action='accepted' )

SELECT a.date, COUNT(b.user_id_receiver)/CAST(COUNT(a.user_id_sender) AS decimal) AS percentage_acceptance
FROM sent_cte a LEFT JOIN accepted_cte b
ON a.user_id_sender=b.user_id_sender AND a.user_id_receiver=b.user_id_receiver
GROUP BY a.date
```

Calculate each user's average session time. A session is defined as the time difference between a page_load and page_exit. For simplicity, assume a user has only 1 session per day and if there are multiple of the same events on that day, consider only the latest page_load and earliest page_exit. Output the user_id and their average session time.

facebook_web_log(user_id:int, timestamp:datetime, action:varchar)

ANSWER

```
WITH all_user_sessions AS (SELECT t1.user_id,
t1.timestamp::DATE AS date, MIN(t2.timestamp::TIMESTAMP) - MAX(t1.timestamp::TIMESTAMP) AS session_duration
FROM facebook_web_log t1 JOIN facebook_web_log t2 ON t1.user_id = t2.user_id
WHERE t1.action = 'page_load' AND t2.action = 'page_exit'  AND t2.timestamp > t1.timestamp
GROUP BY 1, 2)

SELECT user_id, avg(session_duration) FROM all_user_sessions GROUP BY user_id
```

EXAMPLE: Average time spent per user per day on an app (consecutive login/logout events of a user to be ignored)

```
-- create
CREATE TABLE EVENTS (event_type VARCHAR(20), ts TIMESTAMP, user_id INTEGER );

-- insert
INSERT INTO EVENTS VALUES ('login', '2019-11-20 00:14:46', 978699);
INSERT INTO EVENTS VALUES ('logout', '2019-11-20 00:14:46', 992210);
INSERT INTO EVENTS VALUES ('login', '2019-11-20 00:14:46', 823323);
INSERT INTO EVENTS VALUES ('like', '2019-11-20 00:14:47', 978699);
INSERT INTO EVENTS VALUES ('logout', '2019-11-20 00:14:48', 978699);
INSERT INTO EVENTS VALUES ('logout', '2019-11-20 00:14:47', 823323);
INSERT INTO EVENTS VALUES ('login', '2019-11-20 00:14:50', 978699);
INSERT INTO EVENTS VALUES ('logout', '2019-11-20 00:14:57', 978699);

-- fetch
CREATE TABLE event_tbl WITH new_events AS (
SELECT event_type, ts,user_id, DATE(ts) AS dt FROM events
WHERE  event_type="login" OR event_type="logout"
ORDER BY user_id, dt)

SELECT event_type, ts, user_id, dt,
LAG(event_type, 1, 0) OVER (PARTITION BY user_id, dt ORDER BY user_id,dt) AS  prev_event,
LAG(ts, 1, 0) OVER (PARTITION BY user_id, dt ORDER BY user_id,dt) AS prev_ts
FROM new_events;

SELECT user_id, AVG(duration)
FROM (
SELECT user_id, dt, TIMESTAMPDIFF(SECOND, prev_ts, ts) AS duration
FROM  event_tbl
WHERE event_type != prev_event) AS tdiff
GROUP BY user_id;
```