



Second Day Confirmation (TikTok)

QUESTION

Assume you're given tables with information about TikTok user sign-ups and confirmations through email and text. New users on TikTok sign up using their email addresses, and upon sign-up, each user receives a text message confirmation to activate their account.

Write a query to display the user IDs of those who did not confirm their sign-up on the first day, but confirmed on the second day.

Definition:

- `action_date` refers to the date when users activated their accounts and confirmed their sign-up through text messages.

`emails` Table:

Column Name	Type
email_id	integer
user_id	integer

`emails` Example Input:

email_id	user_id	signup_date
125	7771	06/14/2022 00:00:00
433	1052	07/09/2022

signup_date	datetime			00:00:00
-------------	----------	--	--	----------

texts Table:

Column Name	Type
text_id	integer
email_id	integer
signup_action	string ('Confirmed', 'Not confirmed')
action_date	datetime

texts Example Input:

text_id	email_id	signup_action	action_date
6878	125	Confirmed	06/14/2022 00:00:00
6997	433	Not Confirmed	07/09/2022 00:00:00
7000	433	Confirmed	07/10/2022 00:00:00

Step 1: Identify the problem of the case

When registering on TikTok, new users provide their email address and are then sent a text message confirmation to activate their accounts. For individuals who didn't confirm their registration on the first day but did so on the second, create a query to display their user IDs. The results are presented in the following format:

Column Name	Type
user_id	integer

Step 2 : Analyze and solve problems

First, we use the `JOIN` statement to combine two given data tables

```
SELECT *
FROM emails
JOIN texts
  ON emails.email_id = texts.email_id
```

Next we find the accounts that have registered activity over 2 days with `WHERE` and the `EXTRACT()` function to get the date from the time fields. Also filter to see on Monday which accounts have confirmed email.

```
SELECT *
FROM emails
JOIN texts
  ON emails.email_id = texts.email_id
WHERE EXTRACT(day FROM (action_date - signup_date)) = 1
ORDER BY emails.email_id
```

finally out simply returns the value of `user_id` .

```
SELECT user_id
FROM emails
JOIN texts
  ON emails.email_id = texts.email_id
WHERE EXTRACT(day FROM (action_date - signup_date)) = 1
ORDER BY emails.email_id
```

user_id
1052
1235