



1

Page With No Likes (Facebook)

Question:

In this scenario, imagine that you have access to two tables containing data about Facebook Pages and Page likes. The first table is called "pages" and contains information about each Facebook page, including a unique ID number and the name of the page. The second table is called "page_likes" and contains information about users who have liked a particular Facebook page, including the ID of the user, the ID of the page they liked, and the date on which they liked the page.

Given this data, you are tasked with writing a query that will return the IDs of the Facebook pages that have not received any likes. The output of your query should be sorted in ascending order. To help you better understand the data and task at hand, example inputs for both tables have been provided above.

Now, let's move on to step one of solving this SQL case question.

pages Table:

Column Name	Type
page_id	integer
page_name	varchar

pages Example Input:

page_id	page_name
20001	SQL Solutions
20045	Brain Exercises
20701	Tips for Data Analysts

page_likes Table:

Column Name	Type
user_id	integer
page_id	integer
liked_date	datetime

Example Input:

user_id	page_id	liked_date
111	20001	04/08/2022 00:00:00
121	20045	03/12/2022 00:00:00
156	20001	07/25/2022 00:00:00

Step 1: Identify the problem of the case

In the context of a SQL case question, the first step is to identify the problem that needs to be solved. In this particular case, we are given access to two tables containing information about Facebook pages and page likes. The "pages" table contains a unique ID number and the name of the page, while the "page_likes" table contains information about users who have liked a particular Facebook page, including the ID of the user, the ID of the page they liked, and the date on which they liked the page.

The task at hand is to write a query that will return the IDs of the Facebook pages that have not received any likes, with the output sorted in ascending order. To better understand the data and the task, example inputs for both tables have been provided.

Therefore, the first step in solving this SQL case question is to make assumptions about the data and identify the specific problem that needs to be addressed. We can start by examining the structure and contents of the two tables, and then use this information to formulate an appropriate query.

We need an overview of the data table:

For the page table, we use the following statement to access the entire table data:

```
SELECT * FROM pages;
```

We get the following results:

page_id	page_name
20001	SQL Solutions

20045	Brain Exercises
20701	Tips for Data Analysts
31111	Postgres Crash Course
32728	Break the thread

Đối với bảng pages_like:

```
SELECT * FROM page_likes;
```

Here is a table of the data provided in the selection:

user_id	page_id	liked_date
111	20001	04/08/2022 00:00:00
121	20045	03/12/2022 00:00:00
156	20001	07/25/2022 00:00:00
255	20045	07/19/2022 00:00:00
125	20001	07/19/2022 00:00:00
144	31111	06/21/2022 00:00:00
125	31111	07/04/2022 00:00:00

Finally, the result we get is the ID of the page that has not been liked yet. and the result looks like this:

Column Name	Type
page_id	integer

Step 2 : Make Assumptions about the SQL Case Question

To know if a page has been liked or not, we rely on the page_likes table. If there is no page data in the page_likes table, the page will not be liked.

We need to merge two tables together with the join statement with the key being page_id. Here, we use full outer join to return results with full data of two tables.

```
SELECT * FROM pages
FULL OUTER JOIN page_likes
ON pages.page_id = page_likes.page_id
```

We get the following results:

page_id	page_name	user_id	page_id	liked_date
20001	SQL Solutions	111	20001	04/08/2022 00:00:00
20045	Brain Exercises	121	20045	03/12/2022 00:00:00
20001	SQL Solutions	156	20001	07/25/2022 00:00:00
20045	Brain Exercises	255	20045	07/19/2022 00:00:00
20001	SQL Solutions	125	20001	07/19/2022 00:00:00
31111	Postgres Crash Course	144	31111	06/21/2022 00:00:00
31111	Postgres Crash Course	125	31111	07/04/2022 00:00:00
32728	Break the thread	NULL	NULL	NULL
20701	Tips for Data Analysts	NULL	NULL	NULL

So from the returned results table, we see that there are two pages (Break the thread and Tips for Data Analysts) that have no data in the page_likes table (displaying NULL values), so these two pages have no likes.

Using the WHERE statement to filter out pages with no likes is equivalent to having liked_date with a NULL value.

```
SELECT * FROM pages
FULL OUTER JOIN page_likes
    ON pages.page_id = page_likes.page_id
WHERE page_likes.liked_date IS NULL
```

We get the following results:

page_id	page_name	user_id	page_id	liked_date
32728	Break the thread	NULL	NULL	NULL
20701	Tips for Data Analysts	NULL	NULL	NULL

Nearly done? Note that the result only needs page_id, so we only need to get page_id, and sort them in ascending id order.

Step 3: SQL Coding + Analysis

Through the above steps, we have the following command:

```
SELECT pages.page_id FROM pages
FULL OUTER JOIN page_likes
    ON pages.page_id = page_likes.page_id
WHERE page_likes.liked_date IS NULL
ORDER BY pages.page_id
```

We get the following results:

page_id
20701
32728