< Back

Two MySQL solutions

olivia612 ★ 78 Last Edit: November 3, 2019 6:02 AM 963 VIEWS

14

Finding the second item they sold is the hardest part in this question. To solve this, we can use self-join to join the Orders table twice by using multiple conditions.

In this first solution, I used `o1.seller_id = o2.seller_id AND o1.order_date > o2.order_date` and `HAVING(COUNT order_id) =1` to give me the order they sold on the second date. And then I used two `LEFT JOIN` to link the tables together.

```
SELECT user_id as seller_id,
if(i.item_brand = u.favorite_brand, "yes", "no") as 2nd_item_fav_brand
from Users u left join
    (SELECT o1.seller_id, o1.item_id, o1.order_date
    FROM Orders o1 JOIN Orders o2
    ON o1.seller_id = o2.seller_id AND o1.order_date > o2.order_date
    GROUP BY o1.seller_id, o1.order_date
    HAVING count(o1.order_id) = 1) t
ON u.user_id = t.seller_id
LEFT JOIN Items i
ON t.item_id = i.item_id
ORDER BY u.user_id;
```

In this second solution, I used a nested sub-query to calculate the "later" order first, and use the WHERE clause to filter out the second order they sold. `CASE WHEN` is the same as the the `IF()` function in the first solution.

```
SELECT user_id AS seller_id,
(CASE WHEN i.item_brand = u.favorite_brand THEN "yes" ELSE "no" END) AS 2nd_item_fav_brand
FROM Users u LEFT JOIN
    (SELECT seller_id, item_id
    FROM orders o1
    WHERE 1 = (SELECT COUNT(order_id) FROM orders o2 WHERE o1.seller_id = o2.seller_id AND        o1.order_date>o2.o
ON u.user_id = t1.seller_id
LEFT JOIN Items i
ON t1.item_id = i.item_id
ORDER BY u.user_id;
```

subquery if case when mysql self-join

Comments: 3 Best Most Votes Newest to Oldest Oldest to Newest

Type comment here... (Markdown is supported)

Post

fmxy ★ 34 January 27, 2020 3:48 PM

how does or why does having count(order_id) = 1 work here? I wanted to do work with min(order_date)

2 Show 2 replies Reply

HYSANG ★ 3 March 27, 2020 8:15 PM

```
WITH t AS
(
    SELECT *
    FROM
    (
        SELECT seller_id, item_brand,
        ROW_NUMBER() OVER(PARTITION BY seller_id ORDER BY order_date) row
        FROM Orders o, Items i
        WHERE o.item_id = i.item_id
    ) o1
```

Read More

0 Reply