DESCRIPTION

Table: Users

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| user_id | int |

| user_name | varchar |

+-------------+---------+

user_id is the primary key (column with unique values) for this table.

Each row of this table contains the name and the id of a user.

Table: Register

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| contest_id | int |

| user_id | int |

+-------------+---------+

(contest_id, user_id) is the primary key (combination of columns with unique values) for this table.

Each row of this table contains the id of a user and the contest they registered into.

Write a solution to find the percentage of the users registered in each contest rounded to **two decimals**.

Return the result table ordered by percentage in **descending order**. In case of a tie, order it by contest_id in **ascending order**.

The result format is in the following example.

Example 1:

Input:

Users table:

+---------+-----------+

| user_id | user_name |

+---------+-----------+

| 6 | Alice |

| 2 | Bob |

| 7 | Alex |

+---------+-----------+

Register table:

+------------+---------+

| contest_id | user_id |

+------------+---------+

| 215 | 6 |

| 209 | 2 |

| 208 | 2 |

| 210 | 6 |

| 208 | 6 |

| 209 | 7 |

| 209 | 6 |

| 215 | 7 |

| 208 | 7 |

| 210 | 2 |

| 207 | 2 |

| 210 | 7 |

+------------+---------+

Output:

+------------+------------+

| contest_id | percentage |

+------------+------------+

| 208 | 100.0 |

| 209 | 100.0 |

| 210 | 100.0 |

| 215 | 66.67 |

| 207 | 33.33 |

+------------+------------+

Explanation:

All the users registered in contests 208, 209, and 210. The percentage is 100% and we sort them in the answer table by contest_id in ascending order.

Alice and Alex registered in contest 215 and the percentage is ((2/3) * 100) = 66.67%

Bob registered in contest 207 and the percentage is ((1/3) * 100) = 33.33%

SOLUTION

MySQL:

- Find number of users from Users table using COUNT() in a subquery
- Find percentage (number of users (Register table) * 100 / number of user from subquery), and ROUND the result to 2 decimals using ROUND()
- GROUP BY contest_id and ORDER BY percentage in descending order, and contest_id in ascending order

```
SELECT contest_id, ROUND(COUNT(user_id)*100/(SELECT COUNT(*) FROM Users), 2) percentage
FROM Register
GROUP BY contest_id
ORDER BY 2 DESC, 1;
```

PostgreSQL:

- Add a CTE (WITH) to find number of users from Users table using COUNT()
- Find percentage (number of users (Register table) * 100 / total users from CTE), and ROUND the result to 2 decimals using ROUND()
- Join tables using JOIN
- GROUP BY contest_id and ORDER BY percentage in descending order, and contest_id in ascending order

```sql
WITH t1 AS(
    SELECT COUNT(u.user_id) total_users
    FROM Users u)
SELECT r.contest_id, ROUND((ROUND(COUNT(user_id), 2) / t1.total_users) * 100, 2)
percentage
FROM Register r, t1
GROUP BY 1, t1.total_users
ORDER BY 2 DESC, 1;
```