Query

sql
CopyEdit

```
SELECT contest_id,
       ROUND(COUNT(DISTINCT user_id) * 100 / (SELECT COUNT(user_id) FROM users), 2) AS percentage
FROM Register
GROUP BY contest_id
ORDER BY percentage DESC, contest_id ASC;
```

Objective

Calculate the **percentage of users** registered for each contest compared to the total number of users, and **rank the contests** accordingly.

Step-by-Step Explanation

1. **FROM Register**

- This is the base table which contains user registrations for contests.
- Each row maps a user_id to a contest_id.

2. **COUNT(DISTINCT user_id)**

- This counts the **unique users** registered for each contest.
- DISTINCT ensures that if a user is accidentally registered more than once for the same contest, they are counted only once.

New Concept: **COUNT(DISTINCT column)**

- This function is used to **eliminate duplicates** before counting.
- It's often used in **analytics or participation-based queries**.

---

3. *** 100 / (SELECT COUNT(user_id) FROM users)**

- This calculates the percentage of total users registered for a specific contest.
- The subquery: (SELECT COUNT(user_id) FROM users) counts **total users** in the platform.

New Concept: Scalar Subquery

- A scalar subquery returns a **single value**.
- Here, it returns the total number of users once and is used in the percentage calculation.

---

4. **ROUND(..., 2)**

- Rounds the resulting percentage to **2 decimal places**, a common format in reporting.

---

5. **GROUP BY contest_id**

- We want to calculate the percentage **for each contest**, so grouping is required.

---

6. **ORDER BY percentage DESC, contest_id ASC**

- Results are sorted:
    - By percentage in **descending order** (higher participation first).
    - Then by contest_id in **ascending order** to break ties.

Key SQL Concepts Used

Concept	Description
COUNT(DISTINCT user_id)	Counts only unique users per contest.
Scalar Subquery	A single-value subquery used in a calculation.
ROUND(x, 2)	Rounds a number to 2 decimal places.
GROUP BY	Required for aggregating by contest_id.
ORDER BY	Sorts by multiple columns, with different sort directions.

Sample Output

contest_id	percentage
3	75.00
1	50.00
2	50.00

Summary for Revision

- **Goal**: For each contest, calculate how many users participated, expressed as a percentage of total users.
- **Key Formulas**:
  - Percentage = (Count of unique users in contest × 100) / Total users
- **Why ROUND**: To limit decimal digits in the output.
- **Why ORDER BY**: To list most popular contests first.

Useful in Real Life

This type of query is commonly used in **reporting systems** to show participation, conversion rates, or engagement percentages across events, promotions, or contests.