

Problem 2026: Low-Quality Problems

Problem Information

Difficulty: Easy

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Table:

Problems

+-----+-----+ | Column Name | Type | +-----+-----+ | problem_id | int | | likes | int | | dislikes | int | +-----+-----+ In SQL, problem_id is the primary key column for this table. Each row of this table indicates the number of likes and dislikes for a LeetCode problem.

Find the IDs of the

low-quality

problems. A LeetCode problem is

low-quality

if the like percentage of the problem (number of likes divided by the total number of votes) is

strictly less than

60%

.

Return the result table ordered by

problem_id

in ascending order.

The result format is in the following example.

Example 1:

Input:

Problems table: +-----+-----+-----+ | problem_id | likes | dislikes |
+-----+-----+-----+ | 6 | 1290 | 425 | | 11 | 2677 | 8659 | | 1 | 4446 | 2760 | | 7 | 8569 |
6086 | | 13 | 2050 | 4164 | | 10 | 9002 | 7446 | +-----+-----+-----+

Output:

+-----+ | problem_id | +-----+ | 7 | | 10 | | 11 | | 13 | +-----+

Explanation:

The like percentages are as follows: - Problem 1: $(4446 / (4446 + 2760)) * 100 = 61.69858\%$ - Problem 6: $(1290 / (1290 + 425)) * 100 = 75.21866\%$ - Problem 7: $(8569 / (8569 + 6086)) * 100 = 58.47151\%$ - Problem 10: $(9002 / (9002 + 7446)) * 100 = 54.73006\%$ - Problem 11: $(2677 / (2677 + 8659)) * 100 = 23.61503\%$ - Problem 13: $(2050 / (2050 + 4164)) * 100 = 32.99002\%$ Problems 7, 10, 11, and 13 are low-quality problems because their like percentages are less than 60%.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd

def low_quality_problems(problems: pd.DataFrame) -> pd.DataFrame:
```

Solutions

MySQL Solution:

```
# Write your MySQL query statement below
```

MS SQL Server Solution:

```
/* Write your T-SQL query statement below */
```

PostgreSQL Solution:

```
-- Write your PostgreSQL query statement below
```

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

def low_quality_problems(problems: pd.DataFrame) -> pd.DataFrame:
```