

Samantha interviews many candidates from different colleges using coding challenges and contests.

Write a query to print the *contest_id*, *hacker_id*, *name*, and the sums

of *total_submissions*, *total_accepted_submissions*, *total_views*, and *total_unique_views* for each contest sorted by *contest_id*. Exclude the contest from the result if all four sums are .

Note: A specific contest can be used to screen candidates at more than one college, but each college only holds screening contest.

Input Format

The following tables hold interview data:

- *Contests*: The *contest_id* is the id of the contest, *hacker_id* is the id of the hacker who

Column	Type
contest_id	Integer
hacker_id	Integer
name	String

created the contest, and *name* is the name of the hacker.

- *Colleges*: The *college_id* is the id of the college, and *contest_id* is the id of the contest that

Column	Type
college_id	Integer
contest_id	Integer

Samantha used to screen the candidates.

- *Challenges*: The *challenge_id* is the id of the challenge that belongs to one of the contests whose *contest_id* Samantha forgot, and *college_id* is the id of the college where the

Column	Type
challenge_id	Integer
college_id	Integer

challenge was given to candidates.

- *View Stats*: The *challenge_id* is the id of the challenge, *total_views* is the number of times the challenge was viewed by candidates, and *total_unique_views* is the number of times

Column	Type
challenge_id	Integer
total_views	Integer
total_unique_views	Integer

the challenge was viewed by unique candidates.

- *Submission Stats*: The *challenge_id* is the id of the challenge, *total_submissions* is the number of submissions for the challenge, and *total_accepted_submission* is the number of

Column	Type
challenge_id	Integer
total_submissions	Integer
total_accepted_submissions	Integer

submissions that achieved full scores.

Sample Input

contest_id	hacker_id	name
66406	17973	Rose
66556	79153	Angela
94828	80275	Frank

Contests Table:

Colleges Table:

college_id	contest_id
11219	66406
32473	66556
56685	94828

Challenges Table:

challenge_id	college_id
18765	11219
47127	11219
60292	32473
72974	56685

challenge_id	total_views	total_unique_views
47127	26	19
47127	15	14
18765	43	10
18765	72	13
75516	35	17
60292	11	10
72974	41	15
75516	75	11

View_Stats Table:

Submission_Stats Table:

challenge_id	total_submissions	total_accepted_submissions
75516	34	12
47127	27	10
47127	56	18
75516	74	12
75516	83	8
72974	68	24
72974	82	14
47127	28	11

Sample Output

66406 17973 Rose 111 39 156 56
66556 79153 Angela 0 0 11 10
94828 80275 Frank 150 38 41 15

Explanation

The contest 66406 is used in the college 11219. In this college 11219, challenges 18765 and 47127 are asked, so from the view and submission stats:

- Sum of total submissions = $27 + 56 + 28 = 111$
- Sum of total accepted submissions = $10 + 18 + 11 = 39$
- Sum of total views = $43 + 72 + 26 + 15 = 156$
- Sum of total unique views = $10 + 13 + 19 + 14 = 56$

Similarly, we can find the sums for contests 66556 and 94828.

WITH contest_w_challenge_id AS

(

SELECT c.contest_id, ch.challenge_id, co.name, co.hacker_id

FROM challenges AS ch

INNER JOIN colleges AS c

ON c.college_id = ch.college_id

INNER JOIN contests AS co

ON c.contest_id = co.contest_id

),

view_stats_no_dup AS

(

SELECT challenge_id, SUM(total_views) AS view_total, SUM(total_unique_views) AS uview_total

FROM view_stats

GROUP BY challenge_id

),

submission_stats_no_dup AS

(

SELECT challenge_id, SUM(total_submissions) AS sub_total, SUM(total_accepted_submissions) AS sub_atotal

FROM submission_stats

GROUP BY challenge_id

),

challenge_stats AS

```

(
    SELECT v.challenge_id AS v_challenge_id, s.challenge_id AS s_challenge_id, v.view_total,
    v.uview_total, s.sub_total, s.sub_atotal

    FROM view_stats_no_dup AS v

    FULL JOIN submission_stats_no_dup AS s

    ON v.challenge_id = s.challenge_id
),
challenge_stats_wo_null AS
(
SELECT
    CASE
        WHEN v_challenge_id IS NULL THEN s_challenge_id
        WHEN s_challenge_id IS NULL THEN v_challenge_id
        ELSE v_challenge_id
        END AS challenge_id_not_null,
    CASE
        WHEN view_total IS NULL THEN 0
        ELSE view_total
        END AS total_views_not_null,
    CASE
        WHEN uview_total IS NULL THEN 0
        ELSE uview_total
        END AS total_unique_views_not_null,
    CASE
        WHEN sub_total IS NULL THEN 0
        ELSE sub_total
        END AS total_submissions_not_null,
    CASE
        WHEN sub_atotal IS NULL THEN 0
        ELSE sub_atotal
        END AS total_accepted_submissions_not_null

```

```

FROM challenge_stats
),
challenge_stat_summary AS
(
    SELECT
        challenge_id_not_null,
        SUM(total_views_not_null) AS challenge_total_view,
        SUM(total_unique_views_not_null) AS challenge_total_uview,
        SUM(total_submissions_not_null) AS challenge_total_sub,
        SUM(total_accepted_submissions_not_null) AS challenge_total_acc_sub
    FROM challenge_stats_wo_null
    GROUP BY challenge_id_not_null
)
SELECT
    c.contest_id,
    c.hacker_id,
    c.name,
    SUM(challenge_total_sub),
    SUM(challenge_total_acc_sub),
    SUM(challenge_total_view),
    SUM(challenge_total_uview)
FROM challenge_stat_summary AS ch
LEFT JOIN contest_w_challenge_id AS c
ON ch.challenge_id_not_null = c.challenge_id
GROUP BY
    c.contest_id,
    c.hacker_id,
    c.name
ORDER BY contest_id

```