Experiment No. 9

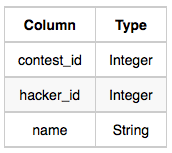
Samantha interviews many candidates from different colleges using coding challenges and contests. Write a query to print the contest\_id, hacker\_id, name, and the sums of total\_submissions, total\_accepted\_submissions, total\_views, and total\_unique\_views for each contest sorted by contest\_id. 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.

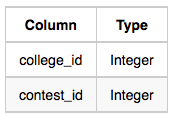
**Input Format**

The following tables hold interview data:

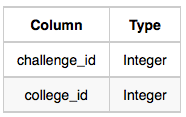
* Contests: The contest\_id is the id of the contest, hacker\_id is the id of the hacker who created the contest, and name is the name of the hacker.



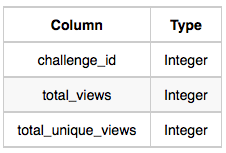
* Colleges: The college\_id is the id of the college, and contest\_id is the id of the contest that Samantha used to screen the candidates.



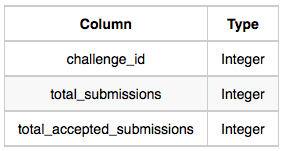
* Challenges: The challenge\_id is the id of the challenge that belongs to one of the contests whose contest\_id Samantha forgot, and college\_id is the id of the college where the challenge was given to candidates.



* 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 the challenge was viewed by unique candidates.

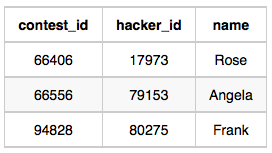


* Submission\_Stats: The challenge\_id is the id of the challenge, total\_submissions is the number of submissions for the challenge, and total\_accepted\_submission is the number of submissions that achieved full scores.

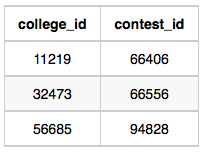


**Sample Input**

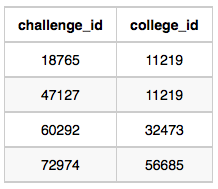
Contests Table:



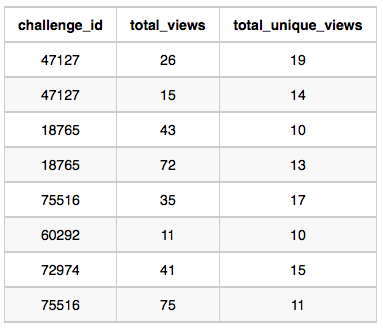
Colleges Table:



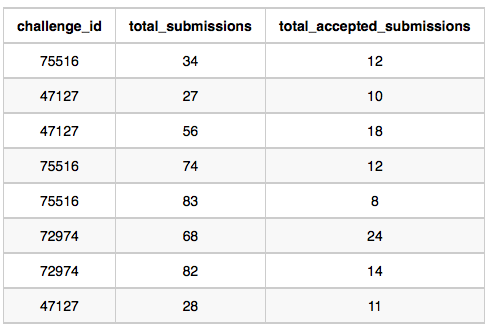
Challenges Table:



View\_Stats Table:



Submission\_Stats Table:



**Sample Output**

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

*To-Do:*

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 0 .

*Logic:*

create: select contest\_id, hacker\_id, name, and the SUM(total\_submissions), SUM(total\_accepted\_submissions) FROM Submission\_Stats table

and create another table: select challenge\_id, SUM(total\_views), SUM(total\_unique\_views) FROM View\_Stats table

and then join these two tables with Contests, Colleges, and Challenges tables, filter the sum of all fours > 0 and order by contest\_id

*Solution:*

SELECT con.contest\_id, con.hacker\_id, con.name, SUM(sg.total\_submissions), SUM(sg.total\_accepted\_submissions),  
SUM(vg.total\_views), SUM(vg.total\_unique\_views)  
FROM Contests AS con   
JOIN Colleges AS col  
ON con.contest\_id = col.contest\_id  
JOIN Challenges AS cha   
ON cha.college\_id = col.college\_id  
LEFT JOIN  
(SELECT ss.challenge\_id, SUM(ss.total\_submissions) AS total\_submissions, SUM(ss.total\_accepted\_submissions) AS total\_accepted\_submissions FROM   
Submission\_Stats AS ss GROUP BY ss.challenge\_id) AS sg  
ON cha.challenge\_id = sg.challenge\_id  
LEFT JOIN  
(SELECT vs.challenge\_id, SUM(vs.total\_views) AS total\_views, SUM(total\_unique\_views) AS total\_unique\_views FROM View\_Stats AS vs GROUP BY vs.challenge\_id) AS vg  
ON cha.challenge\_id = vg.challenge\_id  
GROUP BY con.contest\_id, con.hacker\_id, con.name  
HAVING SUM(sg.total\_submissions)+  
 SUM(sg.total\_accepted\_submissions)+  
 SUM(vg.total\_views)+  
 SUM(vg.total\_unique\_views) > 0  
ORDER BY con.contest\_id;