Basic Select

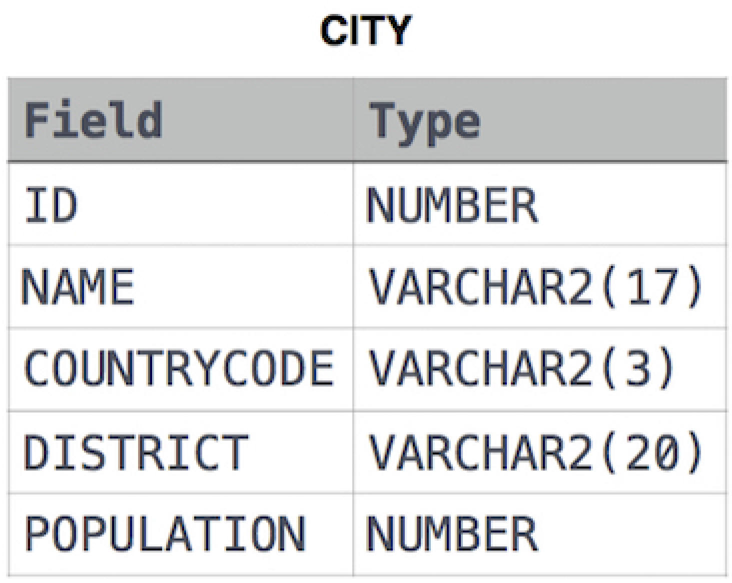
1. Population Census

Given the CITY and COUNTRY tables, query the sum of the populations of all cities where the CONTINENT is 'Asia'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

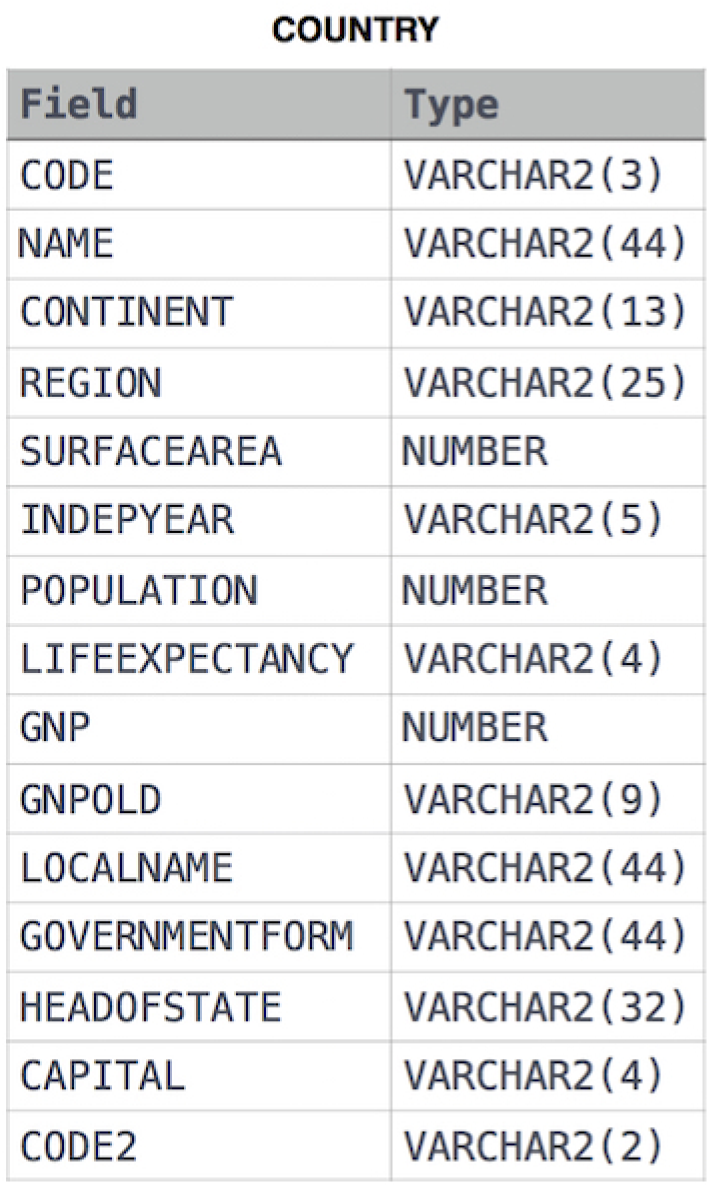
Input Format

The CITY and COUNTRY tables are described as follows:



CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER



COUNTRY

Field	Type
CODE	VARCHAR2(3)
NAME	VARCHAR2(44)
CONTINENT	VARCHAR2(13)
REGION	VARCHAR2(25)
SURFACEAREA	NUMBER
INDEPYEAR	VARCHAR2(5)
POPULATION	NUMBER
LIFEEXPECTANCY	VARCHAR2(4)
GNP	NUMBER
GNPOLD	VARCHAR2(9)
LOCALNAME	VARCHAR2(44)
GOVERNMENTFORM	VARCHAR2(44)
HEADOFSTATE	VARCHAR2(32)
CAPITAL	VARCHAR2(4)
CODE2	VARCHAR2(2)

Link of question Markdown Live Preview.

```
Query : select sum(city.population) from country left join city on country.code =
city.countrycode where country.continent = 'Asia';
```

2. African Cities

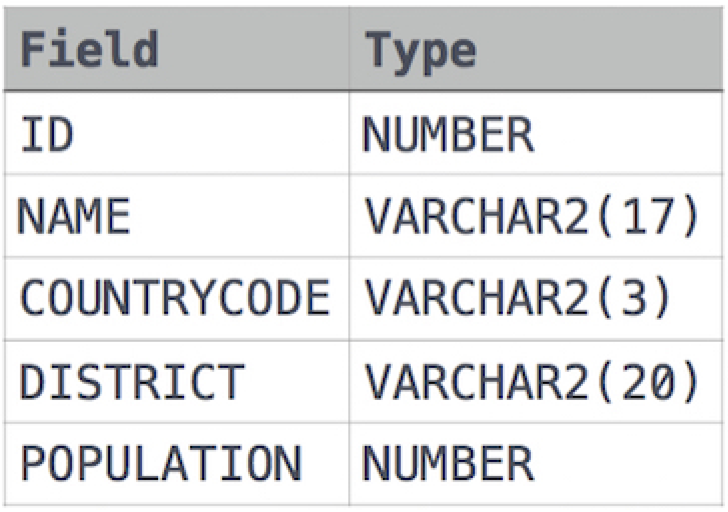
   Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is 'Africa'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

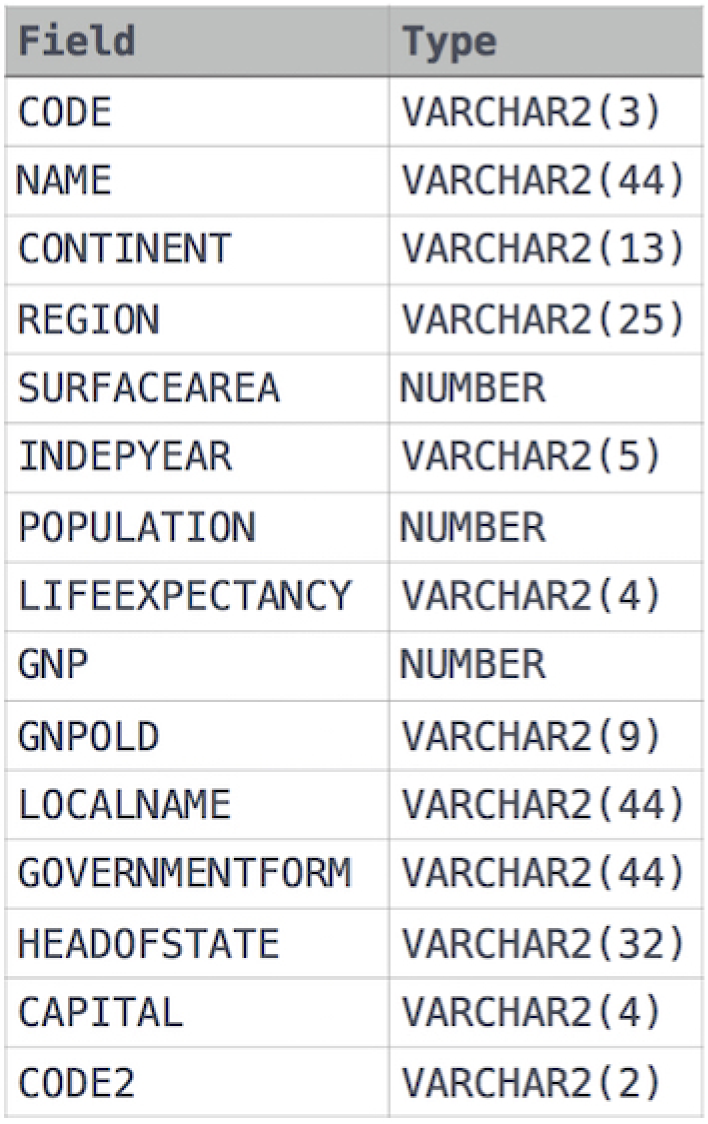
The CITY and COUNTRY tables are described as follows:

CITY



Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

COUNTRY



Field	Type
CODE	VARCHAR2(3)
NAME	VARCHAR2(44)
CONTINENT	VARCHAR2(13)
REGION	VARCHAR2(25)
SURFACEAREA	NUMBER
INDEPYEAR	VARCHAR2(5)
POPULATION	NUMBER
LIFEEXPECTANCY	VARCHAR2(4)
GNP	NUMBER
GNPOLD	VARCHAR2(9)
LOCALNAME	VARCHAR2(44)
GOVERNMENTFORM	VARCHAR2(44)
HEADOFSTATE	VARCHAR2(32)
CAPITAL	VARCHAR2(4)
CODE2	VARCHAR2(2)

Link of question Markdown Live Preview.

```
Query : select city.name from city join country on city.countrycode = country.code
where country.continent = 'Africa';
```

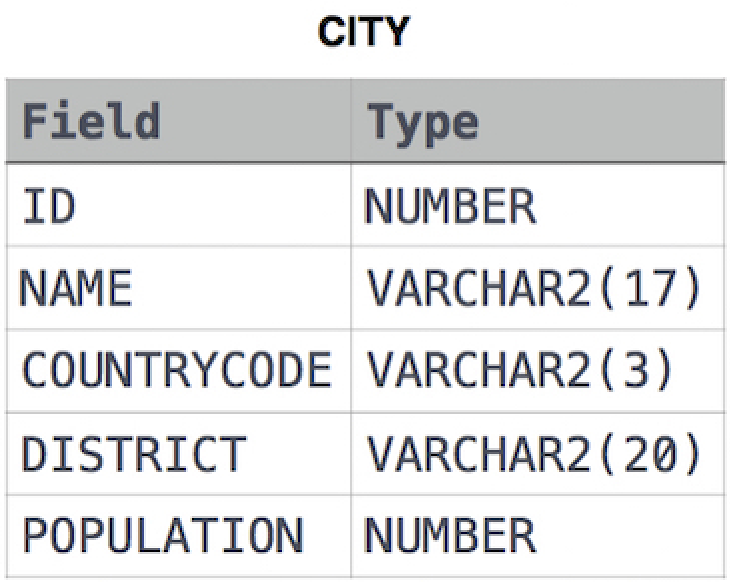
3. Averge population of each continent

   Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

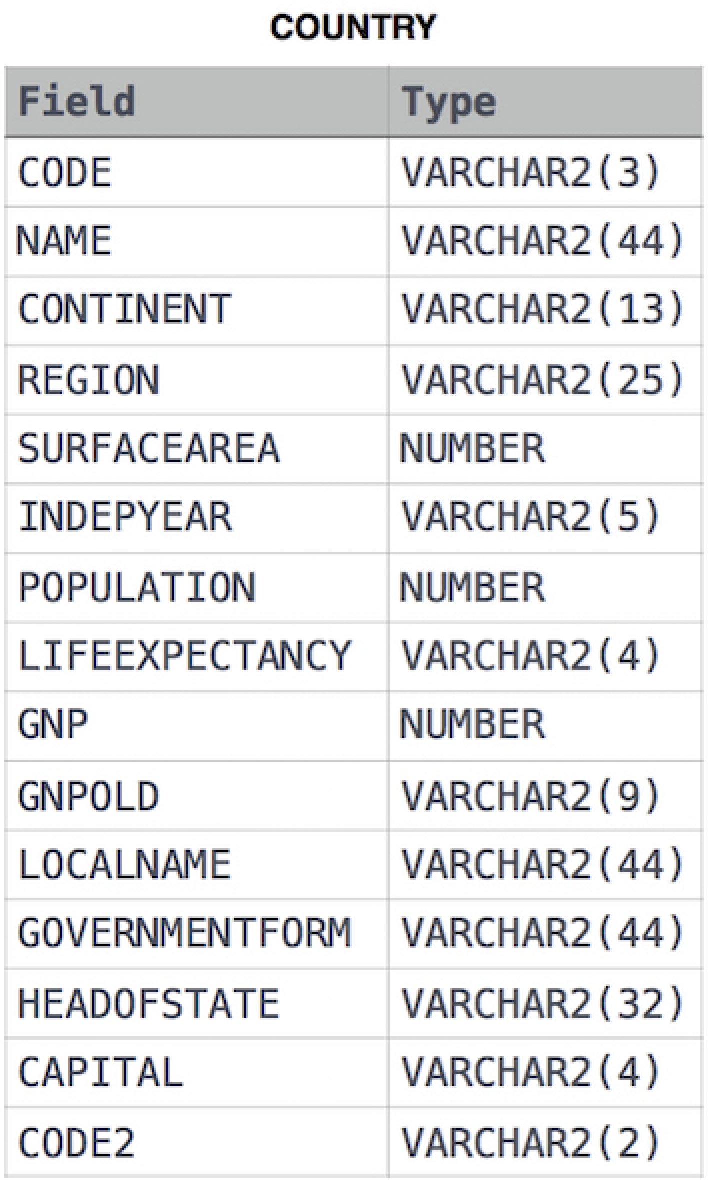
Input Format

The CITY and COUNTRY tables are described as follows:



CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER



COUNTRY

Field	Type
CODE	VARCHAR2(3)
NAME	VARCHAR2(44)
CONTINENT	VARCHAR2(13)
REGION	VARCHAR2(25)
SURFACEAREA	NUMBER
INDEPYEAR	VARCHAR2(5)
POPULATION	NUMBER
LIFEEXPECTANCY	VARCHAR2(4)
GNP	NUMBER
GNPOLD	VARCHAR2(9)
LOCALNAME	VARCHAR2(44)
GOVERNMENTFORM	VARCHAR2(44)
HEADOFSTATE	VARCHAR2(32)
CAPITAL	VARCHAR2(4)
CODE2	VARCHAR2(2)

Link of question Markdown Live Preview.

```
Query : select country.continent, floor(avg(city.population)) from country join
city on city.countrycode = country.code group by country.continent;
```

4. The Report

   You are given two tables: Students and Grades. Students contains three columns ID, Name and Marks.

Column	*Type*
ID	*Integer*
Name	*String*
Marks	*Integer*

Grades contains the following data:

Grade	*Min_Mark*	*Max_Mark*
1	*0*	*9*
2	*10*	*19*
3	*20*	*29*
4	*30*	*39*
5	*40*	*49*
6	*50*	*59*
7	*60*	*69*
8	*70*	*79*
9	*80*	*89*
10	*90*	*100*

Ketty gives Eve a task to generate a report containing three columns: Name, Grade and Mark. Ketty doesn't want the NAMES of those students who received a grade lower than 8. The report must be in descending order by grade -- i.e. higher grades are entered first. If there is more than one student with the same grade (8-10) assigned to them, order those particular students by their name alphabetically. Finally, if the grade is lower than 8, use "NULL" as their name and list them by their grades in descending order. If there is more than one student with the same grade (1-7) assigned to them, order those particular students by their marks in ascending order.

Write a query to help Eve.

Sample Input

ID	Name	Marks
1	Julia	88
2	Samantha	68
3	Maria	99
4	Scarlet	78
5	Ashley	63
6	Jane	81

Sample Output

Maria 10 99 Jane 9 81 Julia 9 88 Scarlet 8 78 NULL 7 63 NULL 7 68

Note

Print "NULL" as the name if the grade is less than 8.

Explanation

Consider the following table with the grades assigned to the students:

ID	Name	Marks	Grade
1	Julia	88	9
2	Samantha	68	7
3	Maria	99	10
4	Scarlet	78	8
5	Ashley	63	7
6	Jane	81	9

So, the following students got 8, 9 or 10 grades:

- Maria (grade 10)
- Jane (grade 9)
- Julia (grade 9)
- Scarlet (grade 8)

Link of question Markdown Live Preview.

```
SELECT CASE
         WHEN G.grade > 7 THEN S.name
         ELSE NULL
       end AS names,
       G.grade,
       S.marks
FROM   students S
       JOIN grades G
         ON S.marks BETWEEN G.min_mark AND G.max_mark
ORDER  BY G.grade DESC,
          names ASC,
          S.marks ASC;
```

5. Top Competitors

   Julia just finished conducting a coding contest, and she needs your help assembling the leaderboard! Write a query to print the respective hacker_id and name of hackers who achieved full scores for more than one challenge. Order your output in descending order by the total number of challenges in which the hacker earned a full score. If more than one hacker received full scores in same number of challenges, then sort them by ascending hacker_id.

The CITY table is described as follows:

Input Format

The following tables contain contest data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

Column	Type
hacker_id	Integer
name	String

- Difficulty: The difficult_level is the level of difficulty of the challenge, and score is the score of the challenge for the difficulty level.

Column	Type
difficulty_level	Integer
score	Integer

- Challenges: The challenge_id is the id of the challenge, the hacker_id is the id of the hacker who created the challenge, and difficulty_level is the level of difficulty of the challenge

Column	Type
challenge_id	Integer
hacker_id	Integer
difficulty_level	Integer

- Submissions: The submission_id is the id of the submission, hacker_id is the id of the hacker who made the submission, challenge_id is the id of the challenge that the submission belongs to, and score is the score of the submission

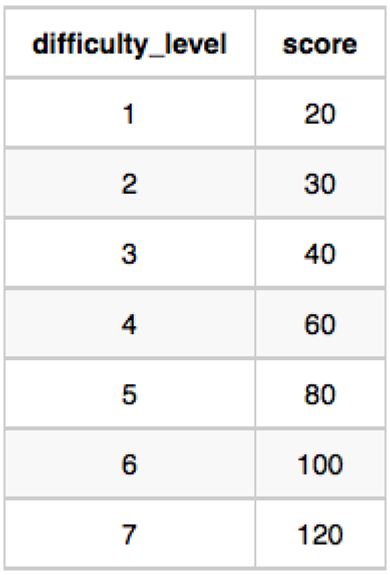
Column	Type
submission_id	Integer
hacker_id	Integer
challenge_id	Integer
score	Integer

Sample Input

- Hackers Table:

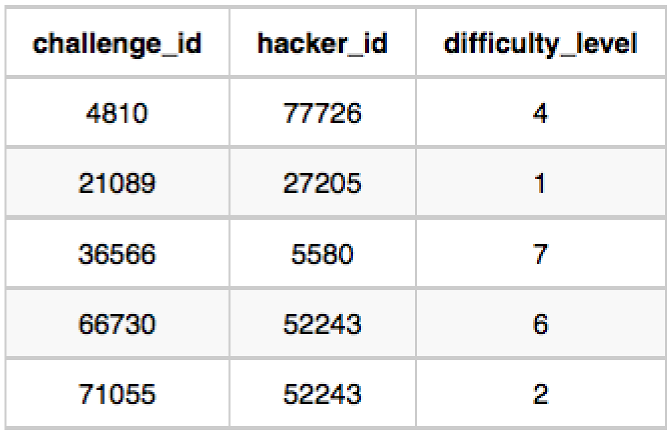
hacker_id	name
5580	Rose
8439	Angela
27205	Frank
52243	Patrick
52348	Lisa
57645	Kimberly
77726	Bonnie
83082	Michael
86870	Todd
90411	Joe

- Difficulty Table:



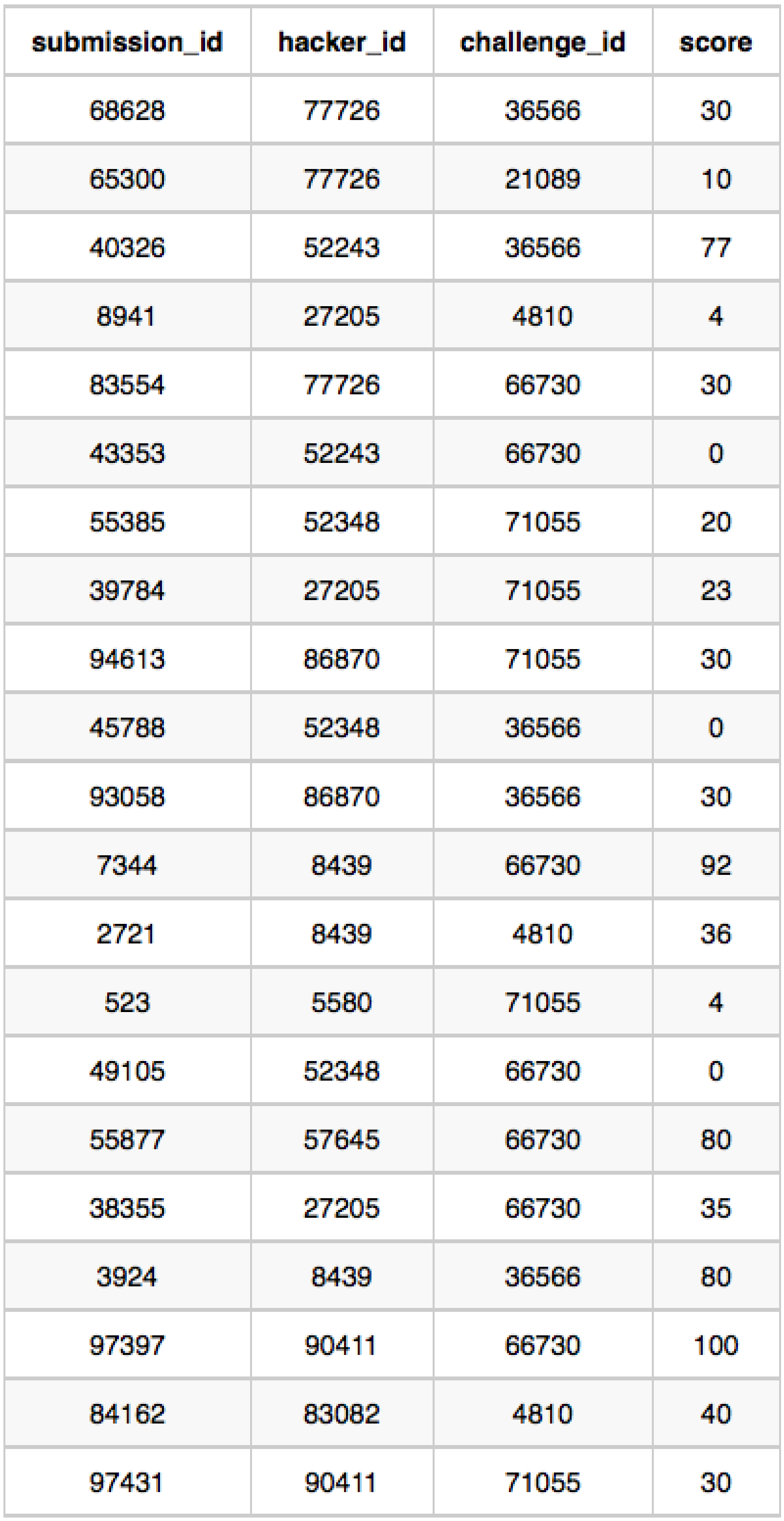
difficulty_level	score
1	20
2	30
3	40
4	60
5	80
6	100
7	120

- Challenges Table:



challenge_id	hacker_id	difficulty_level
4810	77726	4
21089	27205	1
36566	5580	7
66730	52243	6
71055	52243	2

- Submissions Table:



submission_id	hacker_id	challenge_id	score
68628	77726	36566	30
65300	77726	21089	10
40326	52243	36566	77
8941	27205	4810	4
83554	77726	66730	30
43353	52243	66730	0
55385	52348	71055	20
39784	27205	71055	23
94613	86870	71055	30
45788	52348	36566	0
93058	86870	36566	30
7344	8439	66730	92
2721	8439	4810	36
523	5580	71055	4
49105	52348	66730	0
55877	57645	66730	80
38355	27205	66730	35
3924	8439	36566	80
97397	90411	66730	100
84162	83082	4810	40
97431	90411	71055	30

Sample Output

90411 Joe Explanation

Hacker 86870 got a score of 30 for challenge 71055 with a difficulty level of 2, so 86870 earned a full score for this challenge.

Hacker 90411 got a score of 30 for challenge 71055 with a difficulty level of 2, so 90411 earned a full score for this challenge.

Hacker 90411 got a score of 100 for challenge 66730 with a difficulty level of 6, so 90411 earned a full score for this challenge.

Only hacker 90411 managed to earn a full score for more than one challenge, so we print the their hacker_id and name as space-separated values.

Link of question Markdown Live Preview.

```
SELECT H.hacker_id,
       H.name
FROM   submissions S
       JOIN challenges C
         ON S.challenge_id = C.challenge_id
       JOIN difficulty D
         ON C.difficulty_level = D.difficulty_level
       JOIN hackers H
         ON S.hacker_id = H.hacker_id
            AND S.score = D.score
GROUP  BY H.hacker_id,
          H.name
HAVING Count(S.hacker_id) > 1
ORDER  BY Count(S.hacker_id) DESC,
          S.hacker_id ASC;
```

6. Ollivander's Inventory

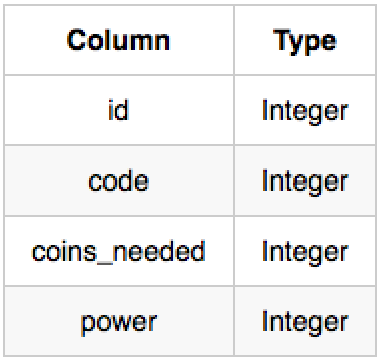
Harry Potter and his friends are at Ollivander's with Ron, finally replacing Charlie's old broken wand.

Hermione decides the best way to choose is by determining the minimum number of gold galleons needed to buy each non-evil wand of high power and age. Write a query to print the id, age, coins_needed, and power of the wands that Ron's interested in, sorted in order of descending power. If more than one wand has same power, sort the result in order of descending age.

Input Format

The following tables contain data on the wands in Ollivander's inventory:

- Wands: The id is the id of the wand, code is the code of the wand, coins_needed is the total number of gold galleons needed to buy the wand, and power denotes the quality of the wand (the higher the power, the better the wand is).



Column	Type
id	Integer
code	Integer
coins_needed	Integer
power	Integer

- Wands_Property: The code is the code of the wand, age is the age of the wand, and is_evil denotes whether the wand is good for the dark arts. If the value of is_evil is 0, it means that the wand is not evil. The mapping between code and age is one-one, meaning that if there are two pairs, (code1, age1) and (code2, age2), then code1!=code2 and .

Column	Type
code	Integer
age	Integer
is_evil	Integer

Sample Input

- Wands Table:

id	code	coins_needed	power
1	4	3688	8
2	3	9365	3
3	3	7187	10
4	3	734	8
5	1	6020	2
6	2	6773	7
7	3	9873	9
8	3	7721	7
9	1	1647	10
10	4	504	5
11	2	7587	5
12	5	9897	10
13	3	4651	8
14	2	5408	1
15	2	6018	7
16	4	7710	5
17	2	8798	7
18	2	3312	3
19	4	7651	6
20	5	5689	3

- Wands_Property Table:

code	age	is_evil
1	45	0
2	40	0
3	4	1
4	20	0
5	17	0

Sample Output

9 45 1647 10 12 17 9897 10 1 20 3688 8 15 40 6018 7 19 20 7651 6 11 40 7587 5 10 20 504 5 18 40 3312 3 20 17 5689 3 5 45 6020 2 14 40 5408 1

Explanation

The data for wands of age 45 (code 1):

id	age	coins_needed	power
5	45	6020	2
9	45	1647	10

- The minimum number of galleons needed for wand(age=45, power=2) = 6020
- The minimum number of galleons needed for wand(age=45, power=10) = 1647 The data for wands of age 40 (code 2):

id	age	coins_needed	power
14	40	5408	1
18	40	3312	3
11	40	7587	5
15	40	6018	7
17	40	8798	7
6	40	6773	7

- The minimum number of galleons needed for wand(age=40, power=1) = 5408
- The minimum number of galleons needed for wand(age=40, power=3) = 3312
- The minimum number of galleons needed for wand(age=40, power=5) = 7587
- The minimum number of galleons needed for wand(age=40, power=7) = 6018 The data for wands of age 20 (code 4):

id	age	coins_needed	power
10	20	504	5
16	20	7710	5
19	20	7651	6
1	20	3688	8

- The minimum number of galleons needed for wand(age=20, power=5) = 504
- The minimum number of galleons needed for wand(age=20, power=6) = 7651
- The minimum number of galleons needed for wand(age=20, power=8) = 3688 The data for wands of age 17 (code 5):

id	age	coins_needed	power
20	17	5689	3
12	17	9897	10

- The minimum number of galleons needed for wand(age=17, power=3) = 5689
- The minimum number of galleons needed for wand(age=17, power=10) = 9897

Link of question Markdown Live Preview.

```
Query :
SELECT a.id,
       b.age,
       a.coins_needed,
       a.power
FROM   wands a
       JOIN wands_property b
         ON a.code = b.code
WHERE  b.is_evil = 0
       AND a.coins_needed = (SELECT Min(a1.coins_needed)
                             FROM   wands a1
                                    JOIN wands_property b1
                                      ON a1.code = b1.code
                             WHERE  b.age = b1.age
                                    AND a.power = a1.power)
ORDER  BY a.power DESC,
          b.age DESC;
```

7. Chanllenges

Julia asked her students to create some coding challenges. Write a query to print the hacker_id, name, and the total number of challenges created by each student. Sort your results by the total number of challenges in descending order. If more than one student created the same number of challenges, then sort the result by hacker_id. If more than one student created the same number of challenges and the count is less than the maximum number of challenges created, then exclude those students from the result.

Input Format

The following tables contain challenge data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

Column	Type
hacker_id	Integer
name	String

- Challenges: The challenge_id is the id of the challenge, and hacker_id is the id of the student who created the challenge.

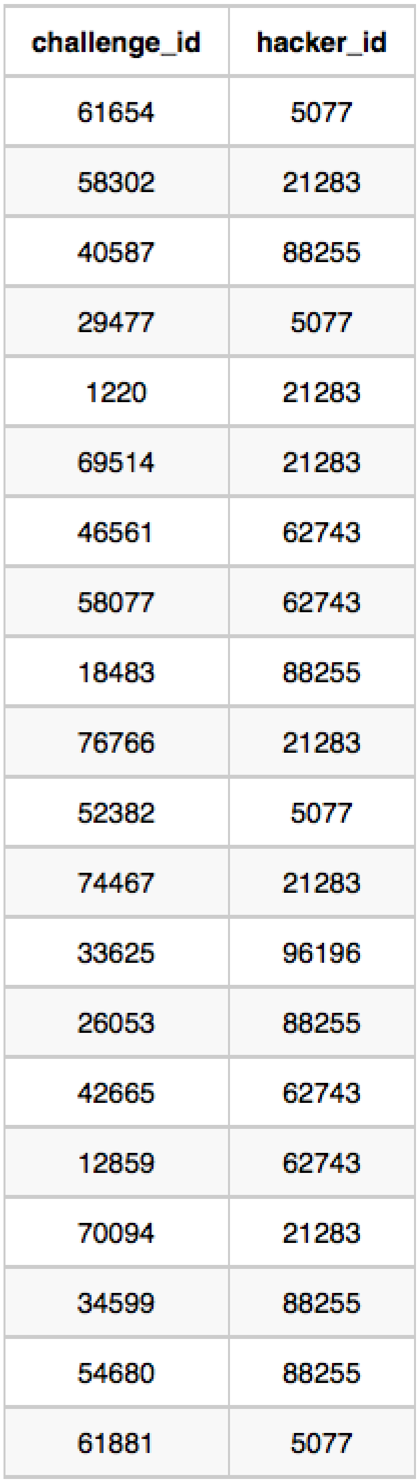
Column	Type
challenge_id	Integer
hacker_id	Integer

Sample Intput 0

- Hackers Table:

hacker_id	name
5077	Rose
21283	Angela
62743	Frank
88255	Patrick
96196	Lisa

- Challenges Table:



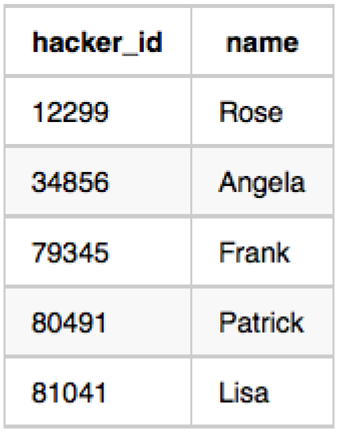
challenge_id	hacker_id
61654	5077
58302	21283
40587	88255
29477	5077
1220	21283
69514	21283
46561	62743
58077	62743
18483	88255
76766	21283
52382	5077
74467	21283
33625	96196
26053	88255
42665	62743
12859	62743
70094	21283
34599	88255
54680	88255
61881	5077

Sample Output 0

21283 Angela 6 88255 Patrick 5 96196 Lisa 1

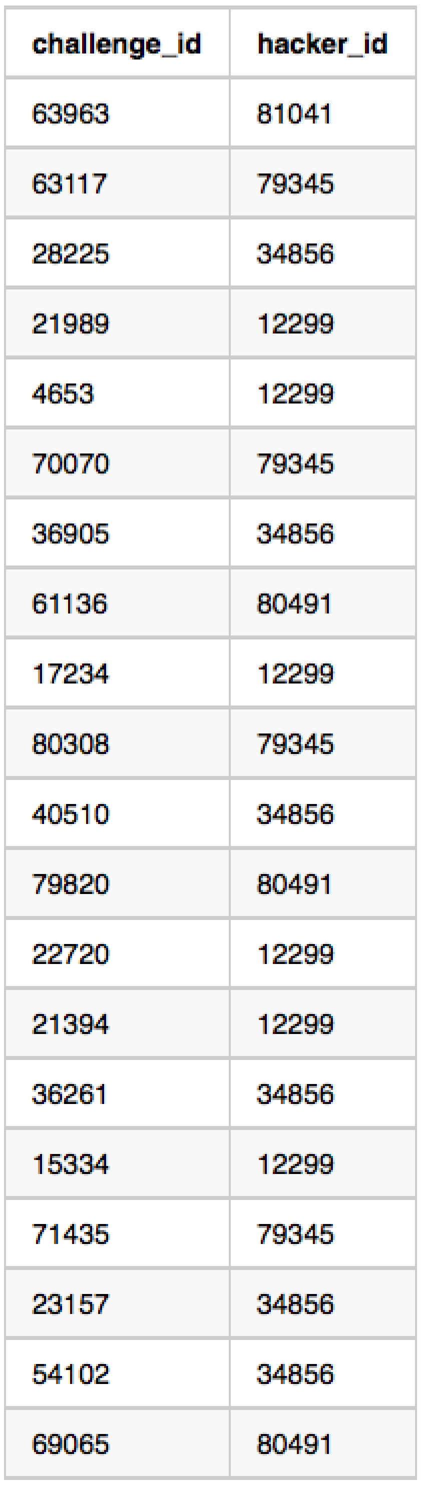
Sample Input 1

- Hackers Table:



hacker_id	name
12299	Rose
34856	Angela
79345	Frank
80491	Patrick
81041	Lisa

- Challenges Table:

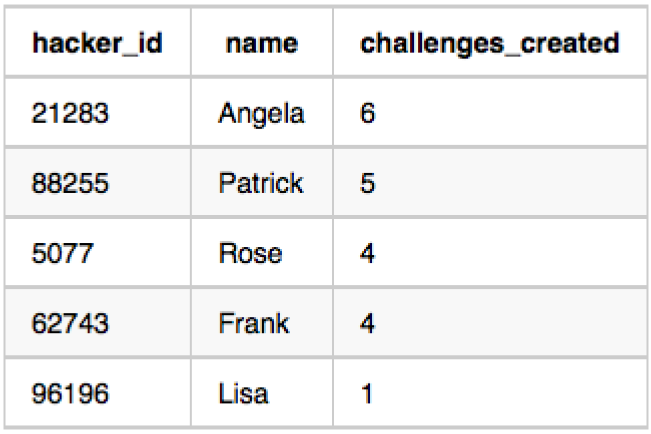


challenge_id	hacker_id
63963	81041
63117	79345
28225	34856
21989	12299
4653	12299
70070	79345
36905	34856
61136	80491
17234	12299
80308	79345
40510	34856
79820	80491
22720	12299
21394	12299
36261	34856
15334	12299
71435	79345
23157	34856
54102	34856
69065	80491

Sample Output 1

12299 Rose 6 34856 Angela 6 79345 Frank 4 80491 Patrick 3 81041 Lisa 1 Explanation

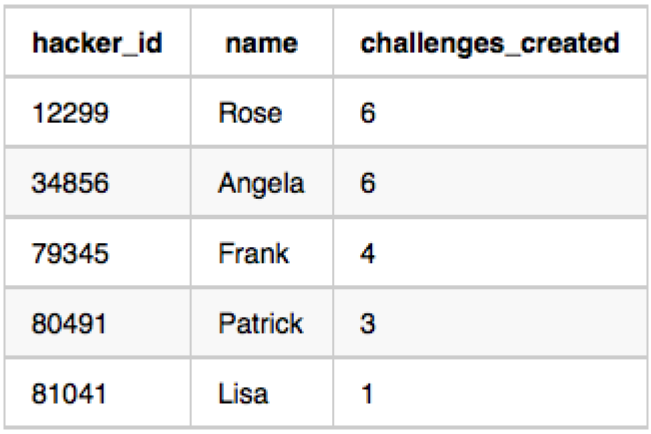
For Sample Case 0, we can get the following details:



hacker_id	name	challenges_created
21283	Angela	6
88255	Patrick	5
5077	Rose	4
62743	Frank	4
96196	Lisa	1

Students 5077 and 62743 both created 4 challenges, but the maximum number of challenges created is 6 so these students are excluded from the result.

For Sample Case 1, we can get the following details:



hacker_id	name	challenges_created
12299	Rose	6
34856	Angela	6
79345	Frank	4
80491	Patrick	3
81041	Lisa	1

Students 12299 and 34856 both created 6 challenges. Because 6 is the maximum number of challenges created, these students are included in the result.

Link of question Markdown Live Preview.

```
Query : -- Use HAVING instead of WHERE since we have to filter on groups
-- Split the total number of counts into 2 pieces
-- First piece will be the largest number
-- Second piece will be the number which doesn't repeat (Unique) or is available
once

select H.hacker_id, H.name, count(C.challenge_id) as total_count
from Hackers H join Challenges C
on H.hacker_id = C.hacker_id
group by H.hacker_id, H.name
having total_count =
(
select count(temp1.challenge_id) as max_count
    from challenges temp1
    group by temp1.hacker_id
    order by max_count desc
    limit 1
)
or total_count in
(
    select distinct other_counts from (
select H2.hacker_id, H2.name, count(C2.challenge_id) as other_counts
from Hackers H2 join Challenges C2
on H2.hacker_id = C2.hacker_id
group by H2.hacker_id, H2.name
) temp2
    group by other_counts
having count(other_counts) =1);
order by total_count desc, H.hacker_id
```

8. Content Leaderboard

You did such a great job helping Julia with her last coding contest challenge that she wants you to work on this one, too!

The total score of a hacker is the sum of their maximum scores for all of the challenges. Write a query to print the hacker_id, name, and total score of the hackers ordered by the descending score. If more than one hacker achieved the same total score, then sort the result by ascending hacker_id. Exclude all hackers with a total score of from your result.

Input Format

The following tables contain contest data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker

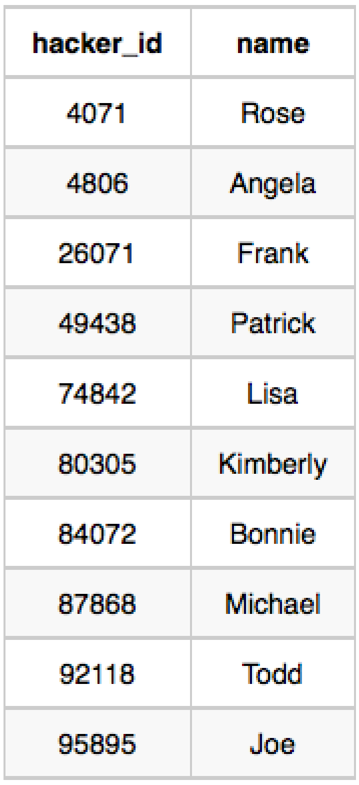
Column	Type
hacker_id	Integer
name	String

- Submissions: The submission_id is the id of the submission, hacker_id is the id of the hacker who made the submission, challenge_id is the id of the challenge for which the submission belongs to, and score is the score of the submission.

Column	Type
submission_id	Integer
hacker_id	Integer
challenge_id	Integer
score	Integer

Sample Input

- Hackers Table:



hacker_id	name
4071	Rose
4806	Angela
26071	Frank
49438	Patrick
74842	Lisa
80305	Kimberly
84072	Bonnie
87868	Michael
92118	Todd
95895	Joe

- Submissions Table:

submission_id	hacker_id	challenge_id	score
67194	74842	63132	76
64479	74842	19797	98
40742	26071	49593	20
17513	4806	49593	32
69846	80305	19797	19
41002	26071	89343	36
52826	49438	49593	9
31093	26071	19797	2
81614	84072	49593	100
44829	26071	89343	17
75147	80305	49593	48
14115	4806	49593	76
6943	4071	19797	95
12855	4806	25917	13
73343	80305	49593	42
84264	84072	63132	0
9951	4071	49593	43
45104	49438	25917	34
53795	74842	19797	5
26363	26071	19797	29
10063	4071	49593	96

Sample Output

4071 Rose 191 74842 Lisa 174 84072 Bonnie 100 4806 Angela 89 26071 Frank 85 80305 Kimberly 67 49438 Patrick 43 Explanation

Hacker 4071 submitted solutions for challenges 19797 and 49593, so the total score =95 + max(43,96) = 191.

Hacker 74842 submitted solutions for challenges 19797 and 63132, so the total score = max(98,5) + 76 = 174.

Hacker 84072 submitted solutions for challenges 49593 and 63132, so the total score =100 + 0 = 100.

The total scores for hackers 4806, 26071, 80305, and 49438 can be similarly calculated.

Link of question Markdown Live Preview.

```
Query : SELECT h.hacker_id, h.name, SUM(MAX_SCORE.t1) as total_score
FROM Hackers h inner join
(
    SELECT MAX(s.score) as t1, s.hacker_id
    FROM Submissions s
    GROUP BY s.challenge_id, s.hacker_id
    HAVING t1 > 0
) AS MAX_SCORE
ON h.hacker_id = MAX_SCORE.hacker_id
GROUP BY h.hacker_id, h.name
ORDER BY total_score DESC, hacker_id ASC;
```