

# INTERMEDIATE SQL QUERIES

Hey! You did a great job in your previous SQL assignment. Let's solve few more advance sql queries in this section.

Instruction:

1. Visit the given question link
2. Login with your Hackerank credential if available, if not then create one.
3. Understand the demand of the question
4. Write your SQL query in the solution box provided there.
5. Run the code and check if it matches with the Sample output provided there.
6. If your SQL query solves the problem then paste it in this notebook below each correponding question cell.
7. THE PADS

[https://www.hackerrank.com/challenges/the-pads/problem (https://www.hackerrank.com/challenges/the- pads/problem)](https://www.hackerrank.com/challenges/the-pads/problem)

select concat(name,'(',substring(OCCUPATION,1,1),')') as names from OCCUPATIONS

order by names;

select concat('There are a total of ',count(\*),' ',lower(OCCUPATION),'s.')

from OCCUPATIONS

group by OCCUPATION

order by count(\*);

1. OCCUPATIONS

<https://www.hackerrank.com/challenges/occupations/problem>

[(https://www.hackerrank.com/challenges/occupations/problem)](https://www.hackerrank.com/challenges/occupations/problem)

SELECT

MAX(CASE WHEN Occupation = 'Doctor' THEN Name END) AS Doctor,

MAX(CASE WHEN Occupation = 'Professor' THEN Name END) AS Professor,

MAX(CASE WHEN Occupation = 'Singer' THEN Name END) AS Singer,

MAX(CASE WHEN Occupation = 'Actor' THEN Name END) AS Actor

FROM (

SELECT

Name, Occupation,

ROW\_NUMBER() OVER (PARTITION BY Occupation ORDER BY Name) AS RowNum

FROM OCCUPATIONS

) AS PivotTable

GROUP BY RowNum

ORDER BY RowNum;

1. THE COMPANY PROBLEM

[https://www.hackerrank.com/challenges/the-company/problem (https://www.hackerrank.com/challenges/the-](https://www.hackerrank.com/challenges/the-company/problem)

[company/problem)](https://www.hackerrank.com/challenges/the-company/problem)

select c.company\_code, c.founder,

count(distinct lm.lead\_manager\_code),

count(distinct sm.senior\_manager\_code),

count(distinct m.manager\_code),

count(distinct e.employee\_code)

from company as c

join lead\_manager as lm on c.company\_code = lm.company\_code

join senior\_manager as sm on lm.lead\_manager\_code = sm.lead\_manager\_code

join manager as m on sm.senior\_manager\_code = m.senior\_manager\_code

join employee as e on m.manager\_code=e.manager\_code

group by c.company\_code,c.founder

order by c.company\_code;

1. Weather observation problem 18

<https://www.hackerrank.com/challenges/weather-observation-station-18/problem>

[(https://www.hackerrank.com/challenges/weather-observation-station-18/problem)](https://www.hackerrank.com/challenges/weather-observation-station-18/problem)

select round(abs(min(lat\_n)-max(lat\_n))+(abs(min(long\_w)-max(long\_w))),4) from station;

1. Weather observation problem 19

<https://www.hackerrank.com/challenges/weather-observation-station-19/problem>

[(https://www.hackerrank.com/challenges/weather-observation-station-19/problem)](https://www.hackerrank.com/challenges/weather-observation-station-19/problem)

select round(sqrt(power((max(lat\_n)-min(lat\_n)),2)+power((max(long\_w )-min(long\_w)),2)),4) from station;

1. Weather observation problem 20

<https://www.hackerrank.com/challenges/weather-observation-station-20/problem>

[(https://www.hackerrank.com/challenges/weather-observation-station-20/problem)](https://www.hackerrank.com/challenges/weather-observation-station-20/problem)

set @index := -1;

select

avg(m.lat\_n) as median

from

(select @index:=@index + 1 as i,

station.lat\_n as lat\_n

from station

order by station.lat\_n) as m

where

m.i in (floor(@index / 2), ceil(@index / 2));

1. The report problem

[https://www.hackerrank.com/challenges/the-report/problem (https://www.hackerrank.com/challenges/the-](https://www.hackerrank.com/challenges/the-report/problem)

[report/problem)](https://www.hackerrank.com/challenges/the-report/problem)

select case

when grade < 8 then "NULL"

else Name

end as Name, Grade, Marks

from Students join Grades

on marks between min\_mark and Max\_mark

order by Grade desc, Name, Marks

1. The competitors

[https://www.hackerrank.com/challenges/full-score/problem (https://www.hackerrank.com/challenges/full-](https://www.hackerrank.com/challenges/full-score/problem)

[score/problem)](https://www.hackerrank.com/challenges/full-score/problem)

select h.hacker\_id, h.name

from

hackers as h join submissions as s on h.hacker\_id = s.hacker\_id

join challenges as c on c.challenge\_id = s.challenge\_id

join difficulty as d on d.difficulty\_level = c.difficulty\_level

where d.score=s.score

group by h.hacker\_id,h.name

having count(c.challenge\_id)>1

order by count(c.challenge\_id) desc, h.hacker\_id;

1. The challenge problem

<https://www.hackerrank.com/challenges/challenges/problem>

[(https://www.hackerrank.com/challenges/challenges/problem)](https://www.hackerrank.com/challenges/challenges/problem)

with cte as (select hacker\_id, count(challenge\_id) as number\_create

from Challenges

group by hacker\_id)

select distinct h.hacker\_id, h.name, c.number\_create

from Hackers as h

left join cte as c

on h.hacker\_id = c.hacker\_id

where c.number\_create = (select max(number\_create) from cte) or

c.number\_create in (select number\_create

from cte

group by number\_create

having count(\*) = 1)

order by c.number\_create desc,

h.hacker\_id

1. The project problem

[https://www.hackerrank.com/challenges/sql-projects/problem (https://www.hackerrank.com/challenges/sql- projects/problem)](https://www.hackerrank.com/challenges/sql-projects/problem)

select min(Start\_Date), max(End\_Date)

from (

select Start\_Date, End\_Date,

adddate(Start\_Date, interval

- (row\_number() over (order by Start\_Date) - 1) day) as project\_group

from Projects) as temp

group by project\_group

order by count(\*) asc

1. The placement problem

<https://www.hackerrank.com/challenges/placements/problem>

[(https://www.hackerrank.com/challenges/placements/problem)](https://www.hackerrank.com/challenges/placements/problem)

select s.name from students as s

join friends as f on s.id=f.id

join packages as p1 on f.id=p1.id

join packages p2 on f.friend\_id=p2.id

where p2.salary>p1.salary

order by p2.salary

1. The symmetric pair problem

<https://www.hackerrank.com/challenges/symmetric-pairs/problem>

[(https://www.hackerrank.com/challenges/symmetric-pairs/problem)](https://www.hackerrank.com/challenges/symmetric-pairs/problem)

select f1.X,f1.Y from functions f1

join functions f2 on f1.X=f2.Y and f1.Y=f2.X

where f1.X <= f1.Y

group by f1.X,f1.Y

having count(\*)>1 or f1.X <> f1.Y

order by f1.X,f1.Y

1. The binary tree problem

<https://www.hackerrank.com/challenges/binary-search-tree-1/problem>

[(https://www.hackerrank.com/challenges/binary-search-tree-1/problem)](https://www.hackerrank.com/challenges/binary-search-tree-1/problem)

select b1.n,

case

when b1.p IS NULL then 'Root'

when b1.n in (select b2.p from BST b2) then 'Inner'

else 'Leaf'

end as Node\_type

from BST b1

order by b1.n

# grin like a Cheshire cat :) Congratulations! you have completed another SQL challenge.