

# 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, '(', SUBSTR(Occupation, 1, 1), ')') from Occupations order by Name;

select concat('There are a total of ', count(Occupation)," ", Lower(occupation), 's.') from Occupations Group by Occupation order by Count(Occupation), Lower(Occupation);

1. OCCUPATIONS

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

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

select min(Doctor),min(Professor),min(Singer),min(Actor)

from

(select Row\_Number() over( partition by occupation order by name) as row\_num,

case when occupation="Doctor" then name end as Doctor,

case when occupation="Professor" then name end as Professor,

case when occupation="Singer" then name end as Singer,

case when occupation="Actor" then name end as Actor

from occupations)

as emp group by row\_num

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 company.company\_code AS company\_code,

company.founder AS founder\_name,

COUNT(DISTINCT lead\_manager.lead\_manager\_code) AS total\_lead\_managers,

COUNT(DISTINCT senior\_manager.senior\_manager\_code) AS total\_senior\_managers,

COUNT(DISTINCT manager.manager\_code) AS total\_managers,

COUNT(DISTINCT employee.employee\_code) AS total\_employees

FROM company

JOIN lead\_manager AS lead\_manager

ON (company.company\_code = lead\_manager.company\_code)

JOIN senior\_manager

ON (lead\_manager.lead\_manager\_code = senior\_manager.lead\_manager\_code)

JOIN manager

ON (senior\_manager.senior\_manager\_code = manager.senior\_manager\_code)

JOIN employee

ON (manager.manager\_code = employee.manager\_code)

GROUP BY company.company\_code,

company.founder

ORDER BY 1;

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 cast(abs((a - c) + (b - d)) as decimal(10,4)) as dist from (select min(lat\_n) as a, min(long\_w) as b, max(lat\_n) as c, max(long\_w) as d from station) abc

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(POW(MAX(lAT\_N) - MIN(LAT\_N), 2) + POW(MAX(lONG\_W) - MIN(LONG\_W), 2)), 4) AS euclidean\_distance

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)

WITH median\_cte AS (

SELECT LAT\_N, ROW\_NUMBER() OVER (ORDER BY LAT\_N) AS row\_num,

COUNT(\*) OVER () AS total\_rows

FROM STATION

)

SELECT round(LAT\_N,4) AS median

FROM median\_cte

WHERE row\_num = (total\_rows + 1) / 2 OR row\_num = (total\_rows + 2) / 2;

1. The report problem

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

[WHEN GRADE <8 THEN NULL](https://www.hackerrank.com/challenges/the-report/problem)

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

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

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

[WHERE MARKS BETWEEN (MIN\_MARK) AND (MAX\_MARK)](https://www.hackerrank.com/challenges/the-report/problem)

[ORDER BY GRADE DESC,NAME ASC (https://www.hackerrank.com/challenges/the-](https://www.hackerrank.com/challenges/the-report/problem)

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

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 difficulty as d

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

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

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

where s.score=d.score

group by 1,2

having count(s.score)>1

order by count(s.score) desc, 1;

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)

select h.hacker\_id,name , cc from(select hacker\_id ,cc, rn, case when rn = 1 then hacker\_id when count(rn) over(partition by rn order by rn ) > 1 then 0 else hacker\_id end as cd from (select hacker\_id ,count(challenge\_id) as cc ,rank() over(order by count(challenge\_id) desc) as rn from challenges group by hacker\_id) a group by hacker\_id )b join hackers h on b.hacker\_id = h.hacker\_id where cd !=0 order by cc desc,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 P1.START\_DATE, P2.END\_DATE FROM (SELECT START\_DATE, ROW\_NUMBER() OVER (ORDER BY START\_DATE) AS ORDENAR FROM PROJECTS WHERE START\_DATE NOT IN (SELECT END\_DATE FROM PROJECTS) ) AS P1 JOIN (SELECT END\_DATE, ROW\_NUMBER() OVER (ORDER BY END\_DATE) AS ORDENAR2 FROM PROJECTS WHERE END\_DATE NOT IN (SELECT START\_DATE FROM PROJECTS) ) AS P2 ON P1.ORDENAR = P2.ORDENAR2 ORDER BY (P2.END\_DATE - P1.START\_DATE) 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)

WITH cte AS

(SELECT s.id,name,f.friend\_id,salary FROM students s

JOIN friends f ON s.id = f.id

JOIN packages p ON s.id = p.id)

SELECT c1.name FROM cte c1

JOIN cte c2 ON c1.friend\_id = c2.id

WHERE c1.salary < c2.salary

ORDER BY c2.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 x, y from

(select

f1.x as x, f1.y as y

from

functions f1

join functions f2

on f1.x=f2.y and f1.y=f2.x and f1.x<>f1.y and f2.x<>f2.y

where f1.x<=f1.y

group by f1.x, f1.y

union all

select x, y

from functions

where x=y

group by x, y

having count(\*)>1) final

order by x asc;

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 n, (case when p is null then 'Root' when n in (select p from BST where p is not null) then 'Inner' else 'Leaf' end)as type from BST order by n;

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

**FeedBack**

We hope you’ve enjoyed this course so far. We’re committed to help you use "AI for All" course to its full potential, so that you have a great learning experience. And that’s why we need your help in form of a feedback here.

[Please fill this feedback form https://zfrmz.in/MtRG5oWXBdesm6rmSM7N (https://zfrmz.in/MtRG5oWXBdesm6rmSM7N)](https://zfrmz.in/MtRG5oWXBdesm6rmSM7N)