**HACKER RANK – SQL CHALLENGES**

**Problem Link:**

<https://www.hackerrank.com/challenges/select-all-sql/problem>

**Solution:**

select \* from city;

**Problem Link:**

<https://www.hackerrank.com/challenges/select-by-id/problem>

**Solution:**

select \* from city where ID ='1661';

**Problem Link:**

<https://www.hackerrank.com/challenges/japanese-cities-attributes/problem>

**Solution:**

select \* from city where countrycode ='JPN';

**Problem Link:**

<https://www.hackerrank.com/challenges/japanese-cities-name/problem>

**Solution:**

select name from city where countrycode ='JPN';

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-1/problem>

**Solution:**

select city,state from station;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-3/problem>

**Solution:**

select distinct(city) from station where id % 2 =0 ;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-4/problem>

**Solution:**

select count(city) - count(distinct(city))

from station;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-5/problem>

**Solution:**

SELECT CITY, LENGTH(CITY)

FROM STATION

GROUP BY CITY

ORDER BY LENGTH(CITY), CITY ASC

LIMIT 1;

SELECT CITY, LENGTH(CITY)

FROM STATION

GROUP BY CITY

ORDER BY LENGTH(CITY) DESC, CITY ASC

LIMIT 1;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-6/problem>

**Solution**

select distinct city from station where city LIKE 'a%' or

city LIKE 'e%' or city LIKE 'i%' or city LIKE 'o%' or city LIKE 'u%';

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-7/problem>

**Solution**  
select distinct city from station where city LIKE '%a' or

city LIKE '%e' or city LIKE '%i' or city LIKE '%o' or city LIKE '%u';

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-8/problem>

**Solution**  
select distinct city from station where (SUBSTRING(city,1,1) IN ('a','e','i','o','u')) and (substring(city,length(city)) in ('a','e','u','i','o'));

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-9/problem>

**Solution**  
select distinct city from station where (left(city,1) Not in ('a','e','u','i','o'));

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-10/problem>

**Solution**  
SELECT DISTINCT CITY FROM STATION WHERE SUBSTR(CITY,Length(CITY))NOT IN ('A', 'E', 'I', 'O', 'U') ;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-11/problem>

**Solution**

SELECT DISTINCT CITY FROM STATION WHERE (Left(CITY,1)NOT IN ('A', 'E', 'I', 'O', 'U'))

or (RIGHT(CITY,1)NOT IN ('A', 'E', 'I', 'O', 'U')) ;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-12/problem>

**Solution**  
SELECT DISTINCT CITY FROM STATION WHERE SUBSTR(CITY,1,1)NOT IN ('A', 'E', 'I', 'O', 'U')

and SUBSTR(CITY,LENGTH(CITY))NOT IN ('A', 'E', 'I', 'O', 'U') ;

**Problem Link:**

<https://www.hackerrank.com/challenges/more-than-75-marks/problem>

**Solution**

select name from students where marks > 75

order by SUBSTR(NAME,CHARACTER\_LENGTH(NAME)-2,3), ID ASC

**Problem Link:**

<https://www.hackerrank.com/challenges/name-of-employees/problem>

**Solution**

select name from Employee order by name asc;

**Problem Link:**

<https://www.hackerrank.com/challenges/salary-of-employees/problem>

**Solution**

select name from Employee where salary >2000 and months <10

order by employee\_id;

**Problem Link:**

<https://www.hackerrank.com/challenges/revising-the-select-query-2/problem>

**Solution**

select name from city where countrycode ='USA' and population >120000;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-14/problem>

**Solution**

select ROUND(LAT\_N,4) from station where LAT\_N < 137.2345

order by LAT\_N DESC LIMIT 1;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-15/problem>

**Solution**

select ROUND(LONG\_W,4) from station where LAT\_N < 137.2345

order by LAT\_N desc limit 1 ;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-16/problem>

**Solution**

select ROUND(LAT\_N,4) from station where LAT\_N > 38.7780

order by LAT\_N ASC LIMIT 1;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-17/problem>

**Solution**

select ROUND(LONG\_W,4) from station where LAT\_N > 38.7780

order by LAT\_N ASC LIMIT 1;

**Problem Link:**

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

**Solution**

with points as(

select

min(LAT\_N) as a,

min(LONG\_W) as b,

max(LAT\_N) as c,

max(LONG\_W) as d

from station)

select

round(sum(abs(a-c) + abs(b-d)), 4)

from points;

**Problem Link:**

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

**Solution**

SELECT ROUND(SQRT(ABS(ROUND(POWER((MIN(LAT\_N)-MAX(LAT\_N)),2) + POWER((MIN(LONG\_W)-MAX(LONG\_W)),2),4))),4) FROM STATION;

**Problem Link:**

<https://www.hackerrank.com/challenges/revising-the-select-query/problem>

**Solution**

select \* from city where countrycode ='USA' and POPULATION > 100000;

**Problem Link:**

<https://www.hackerrank.com/challenges/revising-aggregations-the-count-function/problem>

**Solution**

select count(name) from city where population > 100000;

**Problem Link:**

<https://www.hackerrank.com/challenges/revising-aggregations-sum/problem>

**Solution**

select sum(population) from city where DISTRICT = 'CALIFORNIA';

**Problem Link:**

<https://www.hackerrank.com/challenges/revising-aggregations-the-average-function/problem>

**Solution**

select AVG(POPULATION) from CITY where DISTRICT = 'CALIFORNIA';

**Problem Link:**

<https://www.hackerrank.com/challenges/average-population/problem>

**Solution**

SELECT ROUND(AVG(POPULATION),0) FROM CITY;

**Problem Link:**

<https://www.hackerrank.com/challenges/japan-population/problem>

**Solution**

select SUM(POPULATION) from city where countrycode ='JPN';

**Problem Link:**

<https://www.hackerrank.com/challenges/population-density-difference/problem>

**Solution**

select max(POPULATION) - min(POPULATION) from city;

**Problem Link:**

<https://www.hackerrank.com/challenges/the-blunder/problem>

**Solution**

select round(avg(salary)) - round(avg(replace(salary,0,''))) from employees;

**Problem Link:**

<https://www.hackerrank.com/challenges/earnings-of-employees/problem>

**Solution**

select max(salary\*months), count(\*) from Employee

group by salary\*months

order by salary\*months desc

LIMIT 1 ;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-2/problem>

**Solution**

select ROUND(sum(LAT\_N),2), ROUND(SUM(LONG\_W),2) from station;

**Problem Link:**

<https://www.hackerrank.com/challenges/weather-observation-station-13/problem>

**Solution**

select ROUND(sum(LAT\_N),4) from station where LAT\_N > 38.7880 and LAT\_N <137.2345;

**Problem Link:**

<https://www.hackerrank.com/challenges/draw-the-triangle-1/problem>

**Solution**

SET @no\_of\_lines = 20 + 1;

SELECT REPEAT('\* ', @no\_of\_lines := @no\_of\_lines -1)

FROM INFORMATION\_SCHEMA.TABLES;

**Problem Link:**

<https://www.hackerrank.com/challenges/draw-the-triangle-2/problem>

**Solution**

set @number = 0;

select repeat('\* ', @number := @number + 1) from information\_schema.tables limit 20;

**Problem Link:**

<https://www.hackerrank.com/challenges/african-cities/problem>

**Solution**

select c.name from CITY c join Country co on c.countrycode = co.code where co.continent='Africa';

**Problem Link:**

<https://www.hackerrank.com/challenges/average-population-of-each-continent/problem>

**Solution**

SELECT CO.continent, FLOOR(AVG(C.population)) FROM City C INNER JOIN Country CO ON C.countrycode = CO.code GROUP BY CO.continent;

**Problem Link:**

<https://www.hackerrank.com/challenges/what-type-of-triangle/problem>

**Solution**

select

(case

when A + B <= c or A + C <= B or B + C <= A then 'Not A Triangle'

when A = B and A= C and B=C then 'Equilateral'

when A = B or A= C or B=C then 'Isosceles'

when A != B and A !=c and B!=c then 'Scalene'

end)

from

TRIANGLES;

**Problem Link:**

<https://www.hackerrank.com/challenges/asian-population/problem>

**Solution**

select sum(c.population) from city c

join

country co

on c.countrycode = co.code

where co.continent ='Asia';

**Problem Link:**

<https://www.hackerrank.com/challenges/the-pads/problem>

**Solution**

select CONCAT(name,'(', left(Occupation,1) ,')') from OCCUPATIONS

order by name asc ;

SELECT CONCAT('There are a total of ', COUNT(OCCUPATION), ' ', LOWER(OCCUPATION),'s.') FROM OCCUPATIONS GROUP BY OCCUPATION ORDER BY COUNT(OCCUPATION), OCCUPATION;

**Problem Link:**

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

**Solution**

CREATE VIEW OccupationsView AS

SELECT ROW\_NUMBER() OVER (PARTITION BY Occupation ORDER BY Name) AS row\_num, Occupation, Name FROM OCCUPATIONS; SELECT MAX(IF(Occupation = 'Doctor', Name, NULL)) AS Doctor, MAX(IF(Occupation = 'Professor', Name, NULL)) AS Professor, MAX(IF(Occupation = 'Singer', Name, NULL)) AS Singer, MAX(IF(Occupation = 'Actor', Name, NULL)) AS Actor FROM OccupationsView GROUP BY row\_num ORDER BY row\_num;

**Problem Link:**

<https://www.hackerrank.com/challenges/the-company/problem>

**Solution**

SELECT c.company\_code,

c.founder,

COUNT(DISTINCT(lead\_manager\_code)),

COUNT(DISTINCT(senior\_manager\_code)),

COUNT(DISTINCT(manager\_code)),

COUNT(DISTINCT(employee\_code))

FROM company c

JOIN employee e ON c.company\_code = e.company\_code

GROUP BY c.company\_code,

c.founder

ORDER BY c.company\_code

**Problem Link:**

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

**Solution:**

select ROUND(LAT\_N,4) from station order by LAT\_N ASCLIMIT 1 OFFSET 249;

**Problem Link:**

<https://www.hackerrank.com/challenges/the-report/problem>

**Solution:**

select CASE WHEN G.Grade < 8 THEN NULL

ELSE S.NAME END ,

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 , S.Name asc;

**Problem Link:**

<https://www.hackerrank.com/challenges/harry-potter-and-wands/problem>

**Solution:**

SELECT id, age, coins\_needed, Wands.power

FROM Wands,

(SELECT w.code,

age,

MIN(coins\_needed) as min\_coins,

power

FROM Wands w, Wands\_Property wp

WHERE w.code = wp.code

AND is\_evil = 0

GROUP BY w.code, age, power) AS min\_price\_table

WHERE Wands.code = min\_price\_table.code

AND Wands.coins\_needed = min\_price\_table.min\_coins

AND Wands.power = min\_price\_table.power

ORDER BY power DESC, age DESC;

**Problem Link:**

<https://www.hackerrank.com/challenges/full-score/problem>

**Solution:**

select S.hacker\_id , H.name from Hackers H

join

Submissions S

on H.hacker\_id = S.hacker\_id

join

Challenges C

on S.Challenge\_id = C.challenge\_id

join Difficulty D

on c.difficulty\_level = D.difficulty\_level

where S.Score = D.Score

group by s.hacker\_id,h.name having count(\*) > 1

order by count(\*) desc,s.hacker\_id;

**Problem Link:**

<https://www.hackerrank.com/challenges/contest-leaderboard/problem>

**Solution:**

select

hacker\_id

, name

, sum(total\_score) as total\_score

from (

select

h.hacker\_id

, h.name

, s.challenge\_id

, max(s.score) as total\_score

from hackers as h

inner join submissions as s

on h.hacker\_id = s.hacker\_id

group by h.hacker\_id,h.name,s.challenge\_id

) as t1

group by hacker\_id,name

having total\_score > 0

order by total\_score desc, hacker\_id asc

**Problem Link:**

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

**Solution:**

select S1.Name from Friends F

join Packages P1

on F.ID = P1.ID

join Packages P2

on F.Friend\_ID = P2.ID

join Students S1

on F.ID = S1.ID

join Students S2

on F.Friend\_ID = S2.ID

where P2.salary > P1.Salary

order by P2.salary

**Problem Link:**

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

**Solution:**

WITH Sub AS

(SELECT h.hacker\_id as id , h.name as name , COUNT(challenge\_id) as nbr

FROM Hackers h

JOIN Challenges c

ON h.hacker\_id = c.hacker\_id

GROUP BY h.hacker\_id, h.name),

Max\_nbr AS (SELECT MAX(nbr) as max\_n

FROM Sub),

Selected AS (SELECT nbr, COUNT(nbr) as ct

FROM Sub

GROUP BY 1)

SELECT id, name, s.nbr

FROM Sub s

JOIN Selected sl

ON s.nbr = sl.nbr

WHERE sl.nbr = (SELECT max\_n FROM Max\_nbr) OR sl.ct = 1

ORDER BY s.nbr DESC, id

**Problem Link:**

<https://www.hackerrank.com/challenges/sql-projects/problem>

**Solution:**

select min(t1.start\_date),

max(t1.end\_date)

from

(select start\_date,end\_date,end\_date-row\_number() over (order by end\_date) as rk from projects)

t1 group by t1.rk order by max(t1.end\_date)-min(t1.start\_date)

**Problem Link:**

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

**Solution:**

select f1.x , f1. y from functions f1

inner 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 f1.x <> f1.y OR COUNT(\*) > 1

ORDER BY f1.x ASC;

**Problem Link:**

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

**Solution:**

SELECT N,

CASE

WHEN P IS NULL THEN "Root"

WHEN N IN (SELECT DISTINCT P FROM BST) THEN "Inner"

ELSE "Leaf"

END

FROM BST

ORDER BY N;

**Problem Link:**

<https://www.hackerrank.com/challenges/print-prime-numbers/problem>

**Solution:**

WITH RECURSIVE t AS (

SELECT 2 AS nums

UNION

SELECT nums + 1

FROM t

WHERE nums < 1000

)

SELECT GROUP\_CONCAT(nums SEPARATOR '&')

FROM t

WHERE nums NOT IN (

SELECT DISTINCT a.nums

FROM t a

JOIN t b

ON b.nums != a.nums

AND a.nums % b.nums = 0

)

**Problem Link:**

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

**Solution:**

select a.contest\_id,

hacker\_id,

name,

sum(tot\_s),

sum(tot\_a\_s),

sum(tot\_v),

sum(tot\_u\_v)

from contests a

join colleges b on b.contest\_id = a.contest\_id

join challenges c on c.college\_id = b.college\_id

left join (select challenge\_id,

sum(total\_views) tot\_v,

sum(total\_unique\_views) tot\_u\_v

from view\_stats

group by challenge\_id

) as x on x.challenge\_id = c.challenge\_id

left join (select challenge\_id,

sum(total\_submissions) tot\_s,

sum(total\_accepted\_submissions) tot\_a\_s

from submission\_stats

group by challenge\_id

) as y on y.challenge\_id = c.challenge\_id

group by a.contest\_id,

hacker\_id,

name

having (sum(tot\_s) + sum(tot\_a\_s) + sum(tot\_v) +

sum(tot\_u\_v) !=0 )

order by a.contest\_id

**Problem Link:**

<https://www.hackerrank.com/challenges/15-days-of-learning-sql/problem>

**Solution:**

select d.submission\_date,

(

select count(\*) from

(select s.submission\_date, s.hacker\_id,

@n:=if(count(1) > 1, 1, count(1)) alt,

(select count(hacker\_id)

from

(select s.submission\_date, s.hacker\_id, count(1),

@n:=if(count(1) > 1, 1, count(1)) alt

from submissions s group by 1, 2 order by 1, 3 desc, 2) t

where submission\_date<=s.submission\_date and hacker\_id=s.hacker\_id) agg

from submissions s group by 1, 2 order by 1, 3 desc, 2) oi

where submission\_date<=d.submission\_date and agg=extract(day from d.submission\_date)

) uniq\_count,

(select s.hacker\_id from submissions s inner join hackers h on s.hacker\_id = h.hacker\_id where s.submission\_date = d.submission\_date

group by s.hacker\_id, h.name order by count(s.hacker\_id) desc, s.hacker\_id asc limit 1) id\_highest\_sub,

(select h.name from submissions s inner join hackers h on s.hacker\_id = h.hacker\_id where s.submission\_date = d.submission\_date

group by s.hacker\_id, h.name order by count(s.hacker\_id) desc, s.hacker\_id asc limit 1) name\_highest\_sub

from (select distinct s.submission\_date from submissions s order by s.submission\_date) d;