DAILY SQL

01.08.24

<https://leetcode.com/problems/rising-temperature/description/?envType=study-plan-v2&envId=top-sql-50>

Write a solution to find all dates' Id with higher temperatures compared to its previous dates (yesterday).

select w1.id

from Weather w1, Weather w2

where datediff (w1.recordDate,w2.recordDate) = 1 and w1.temperature > w2.temperature;

02.08.24

<https://leetcode.com/problems/average-time-of-process-per-machine/description/?envType=study-plan-v2&envId=top-sql-50>

# Write your MySQL query statement below

select a1.machine\_id, round(avg(a2.timestamp-a1.timestamp), 3) as processing\_time

from Activity a1

join Activity a2

on a1.machine\_id=a2.machine\_id and a1.process\_id=a2.process\_id

and a1.activity\_type='start' and a2.activity\_type='end'

group by a1.machine\_id

05.08.24

<https://leetcode.com/problems/employee-bonus/?envType=study-plan-v2&envId=top-sql-50>

select Employee.name, Bonus.bonus from Employee

left join Bonus on Employee.empId  = Bonus.empId

where Bonus.bonus is Null or Bonus.bonus < 1000

06.08.24

<https://leetcode.com/problems/students-and-examinations/?envType=study-plan-v2&envId=top-sql-50>

SELECT s.student\_id, s.student\_name, sub.subject\_name, COUNT(e.student\_id) AS attended\_exams

FROM Students s

CROSS JOIN Subjects sub

LEFT JOIN Examinations e ON s.student\_id = e.student\_id AND sub.subject\_name = e.subject\_name

GROUP BY s.student\_id, s.student\_name, sub.subject\_name

ORDER BY s.student\_id, sub.subject\_name;