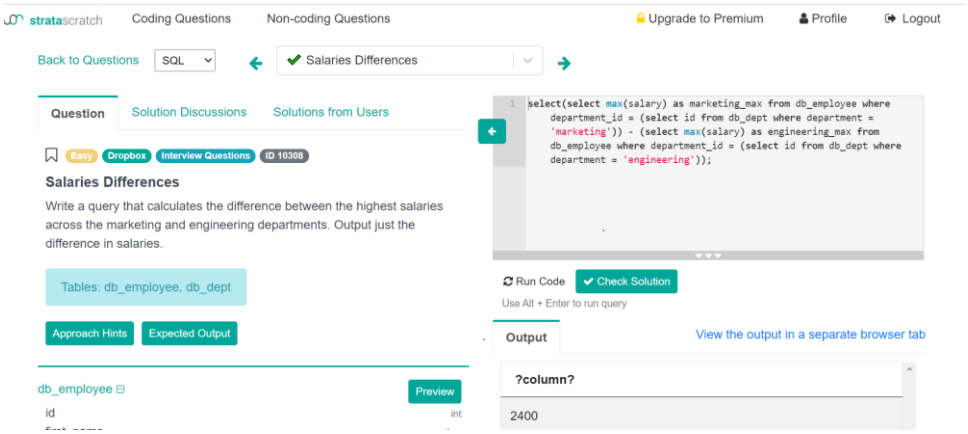
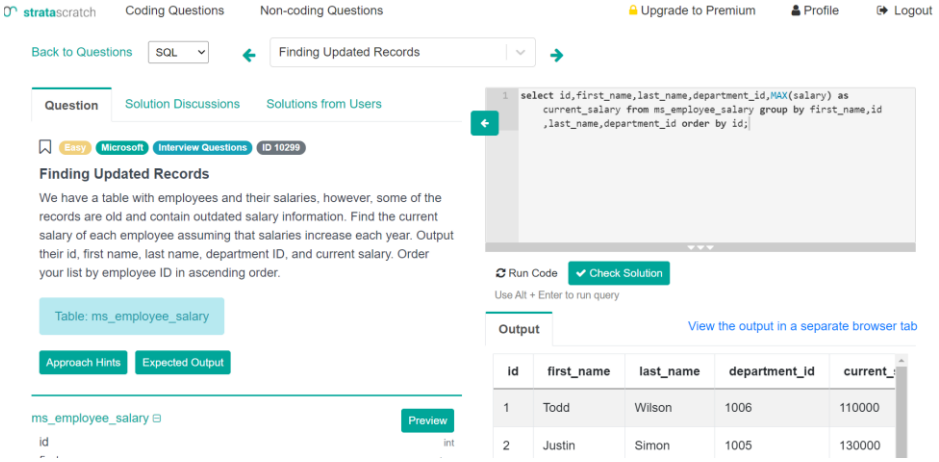
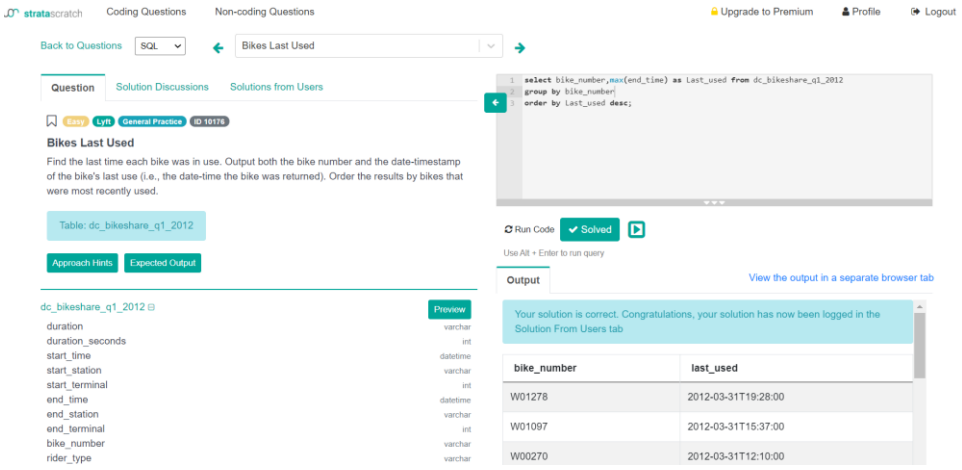
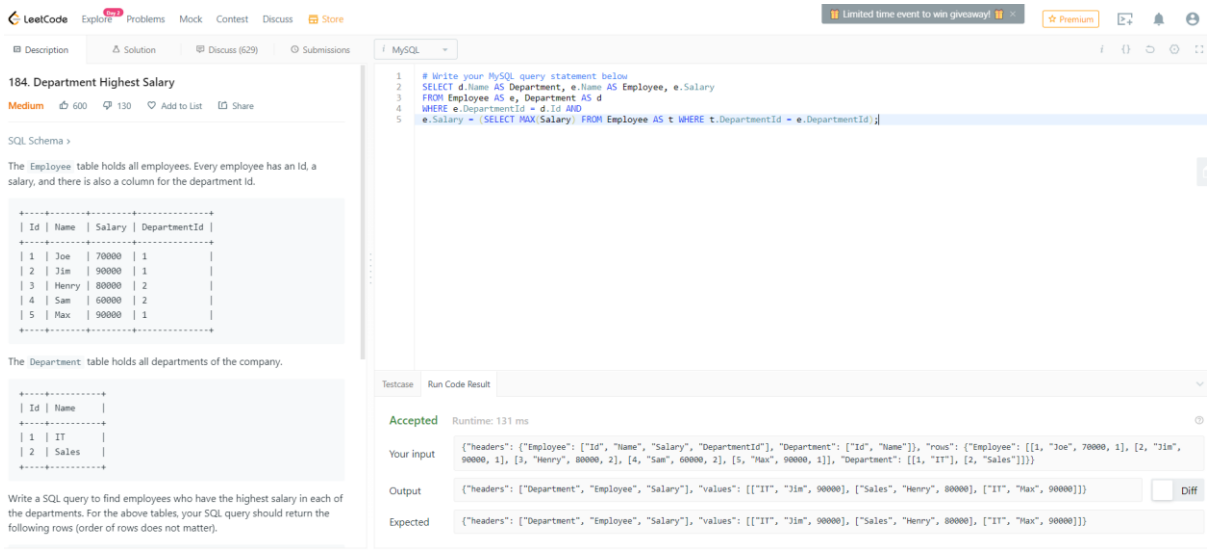
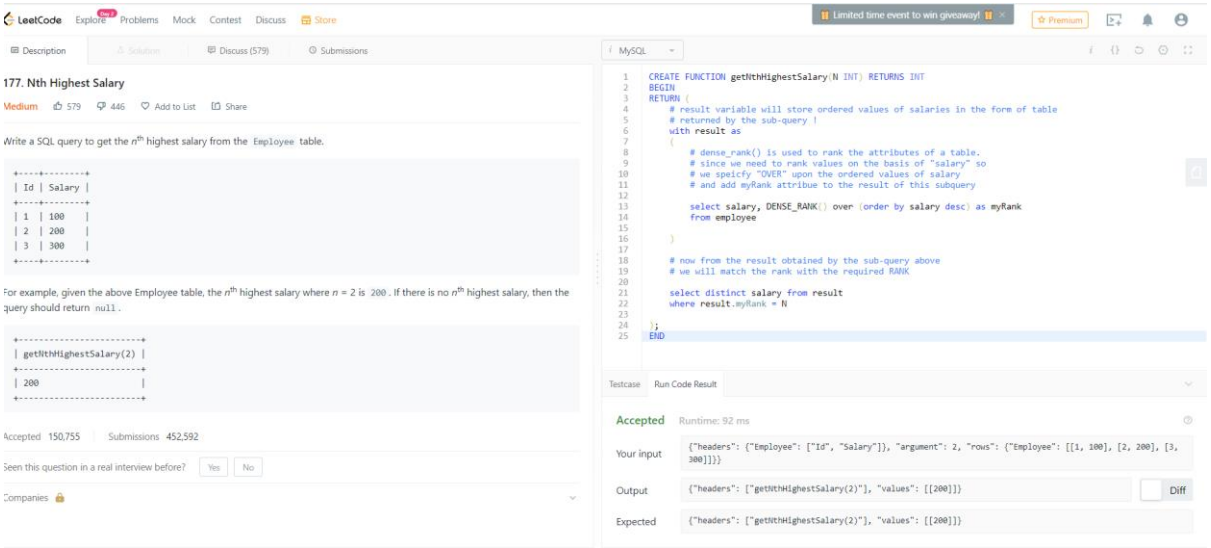
SQL Sample Codes: Below questions are from Stratascratch and Leetcode.

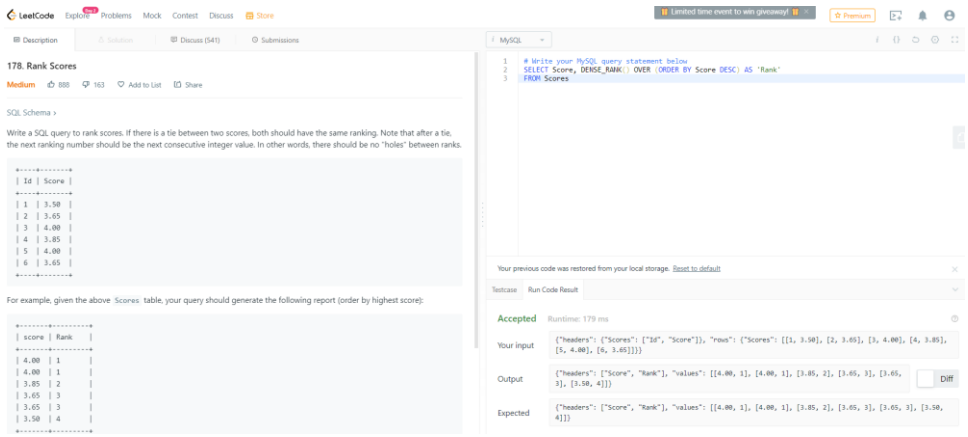



LeetCode Explore Problems Mock Contest Discuss Store
Limited time event to win giveaway!
Premium
Description Solution Discuss (629) Submissions MySQL
184. Department Highest Salary
Medium 600 130 Add to List Share
SQL Schema ›
The Employee table holds all employees. Every employee has an Id, a salary, and there is also a column for the department Id.
+----+-------+--------+--------------+
| Id | Name | Salary | DepartmentId |
+----+-------+--------+--------------+
| 1 | Joe | 70000 | 1 |
| 2 | Jim | 90000 | 1 |
| 3 | Henry | 80000 | 2 |
| 4 | Sam | 60000 | 2 |
| 5 | Max | 90000 | 1 |
+----+-------+--------+--------------+
The Department table holds all departments of the company.
+----+----------+
| Id | Name |
+----+----------+
| 1 | IT |
| 2 | Sales |
+----+----------+
Write a SQL query to find employees who have the highest salary in each of the departments. For the above tables, your SQL query should return the following rows (order of rows does not matter).
# Write your MySQL query statement below
SELECT d.Name AS Department, e.Name AS Employee, e.Salary
FROM Employee AS e, Department AS d
WHERE e.DepartmentId = d.Id AND
e.Salary = (SELECT MAX(Salary) FROM Employee AS t WHERE t.DepartmentId = e.DepartmentId);
Testcase Run Code Result
Accepted Runtime: 131 ms
Your input {"headers": {"Employee": ["Id", "Name", "Salary", "DepartmentId"], "Department": ["Id", "Name"]}, "rows": {"Employee": [[1, "Joe", 70000, 1], [2, "Jim", 90000, 1], [3, "Henry", 80000, 2], [4, "Sam", 60000, 2], [5, "Max", 90000, 1]], "Department": [[1, "IT"], [2, "Sales"]]}}
Output {"headers": ["Department", "Employee", "Salary"], "values": [["IT", "Jim", 90000], ["Sales", "Henry", 80000], ["IT", "Max", 90000]]}
Diff
Expected {"headers": ["Department", "Employee", "Salary"], "values": [["IT", "Jim", 90000], ["Sales", "Henry", 80000], ["IT", "Max", 90000]]}


LeetCode Explore Problems Mock Contest Discuss Store
Limited time event to win giveaway!
Premium
Description Discuss (579) Submissions MySQL
177. Nth Highest Salary
Medium 579 446 Add to List Share
Write a SQL query to get the nth highest salary from the Employee table.
+----+--------+
| Id | Salary |
+----+--------+
| 1 | 100 |
| 2 | 200 |
| 3 | 300 |
+----+--------+
For example, given the above Employee table, the nth highest salary where n = 2 is 200. If there is no nth highest salary, then the query should return null.
+------------------------+
| getNthHighestSalary(2) |
+------------------------+
| 200 |
+------------------------+
Accepted 150,755 Submissions 452,592
Seen this question in a real interview before? Yes No
Companies
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
RETURN (
# result variable will store ordered values of salaries in the form of table
# returned by the sub-query !
with result as
(
# dense_rank() is used to rank the attributes of a table.
# since we need to rank values on the basis of "salary" so
# we speicfy "OVER" upon the ordered values of salary
# and add myRank attribue to the result of this subquery
select salary, DENSE_RANK() over (order by salary desc) as myRank
from employee
)
# now from the result obtained by the sub-query above
# we will match the rank with the required RANK
select distinct salary from result
where result.myRank = N
);
END
Testcase Run Code Result
Accepted Runtime: 92 ms
Your input {"headers": {"Employee": ["Id", "Salary"]}, "argument": 2, "rows": {"Employee": [[1, 100], [2, 200], [3, 300]]}}
Output {"headers": ["getNthHighestSalary(2)"], "values": [[200]]}
Diff
Expected {"headers": ["getNthHighestSalary(2)"], "values": [[200]]}


LeetCode Explore Problems Mock Contest Discuss Store
Limited time event to win giveaway!
Premium
Description Discuss (541) Submissions MySQL
178. Rank Scores
Medium 888 163 Add to List Share
SQL Schema ›
Write a SQL query to rank scores. If there is a tie between two scores, both should have the same ranking. Note that after a tie, the next ranking number should be the next consecutive integer value. In other words, there should be no "holes" between ranks.
+----+-------+
| Id | Score |
+----+-------+
| 1 | 3.50 |
| 2 | 3.65 |
| 3 | 4.00 |
| 4 | 3.85 |
| 5 | 4.00 |
| 6 | 3.65 |
+----+-------+
For example, given the above Scores table, your query should generate the following report (order by highest score):
+-------+---------+
| score | Rank |
+-------+---------+
| 4.00 | 1 |
| 4.00 | 1 |
| 3.85 | 2 |
| 3.65 | 3 |
| 3.65 | 3 |
| 3.50 | 4 |
+-------+---------+
# Write your MySQL query statement below
SELECT Score, DENSE_RANK() OVER (ORDER BY Score DESC) AS 'Rank'
FROM Scores
Your previous code was restored from your local storage. Reset to default
Testcase Run Code Result
Accepted Runtime: 179 ms
Your input {"headers": {"Scores": ["Id", "Score"]}, "rows": {"Scores": [[1, 3.50], [2, 3.65], [3, 4.00], [4, 3.85], [5, 4.00], [6, 3.65]]}}
Output {"headers": ["Score", "Rank"], "values": [[4.00, 1], [4.00, 1], [3.85, 2], [3.65, 3], [3.65, 3], [3.50, 4]]}
Diff
Expected {"headers": ["Score", "Rank"], "values": [[4.00, 1], [4.00, 1], [3.85, 2], [3.65, 3], [3.65, 3], [3.50, 4]]}