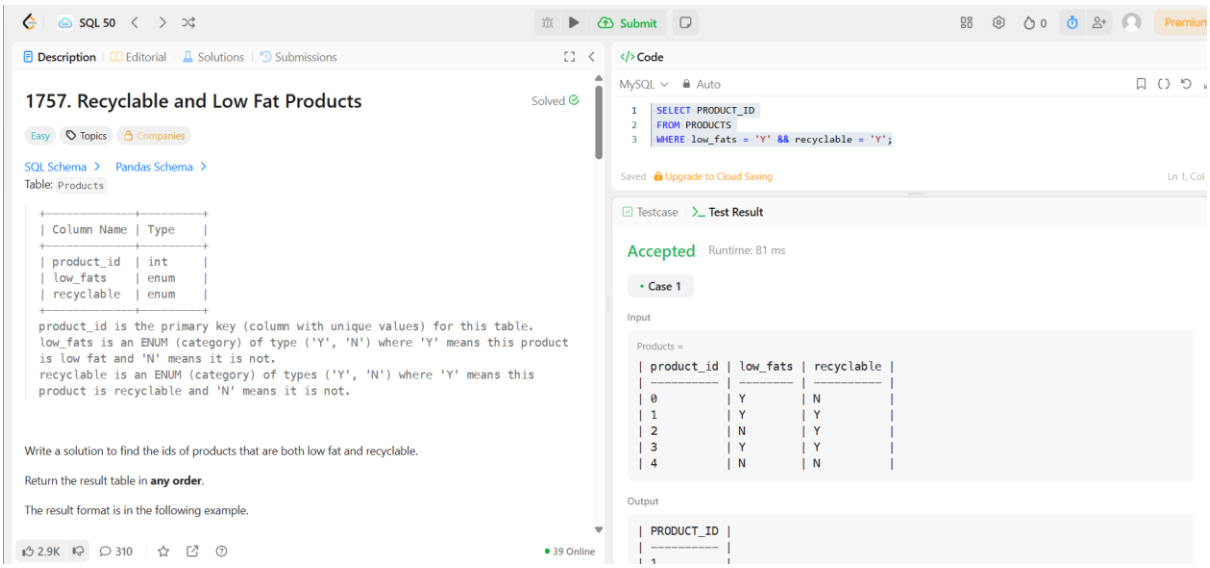
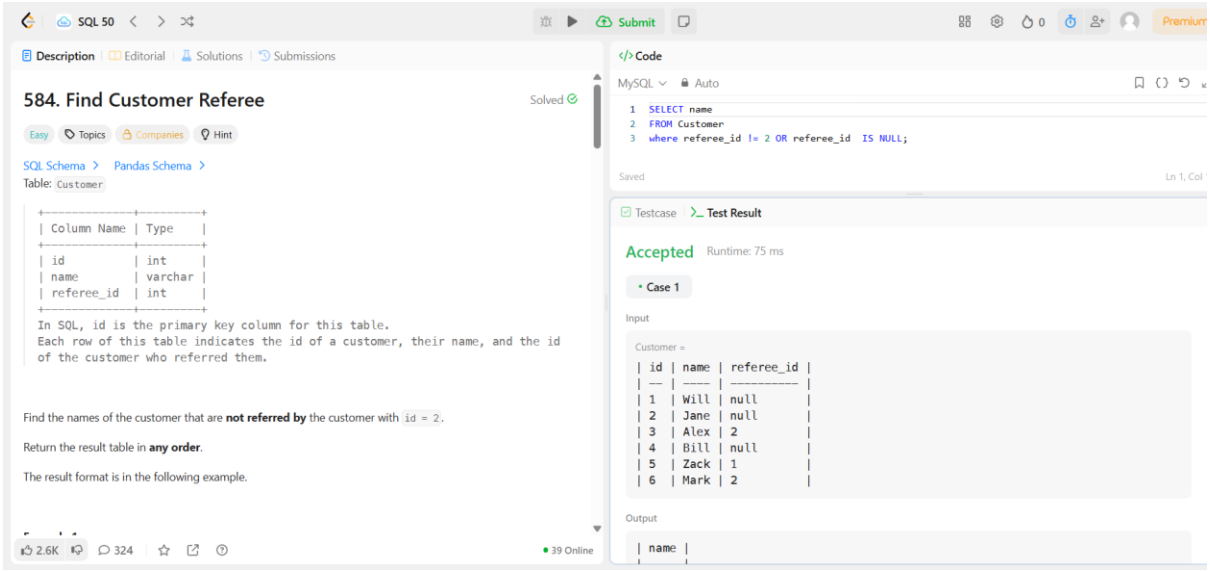
LEETCODE SQL (50) QUESTION & SOLUTION

➢ SELECT-

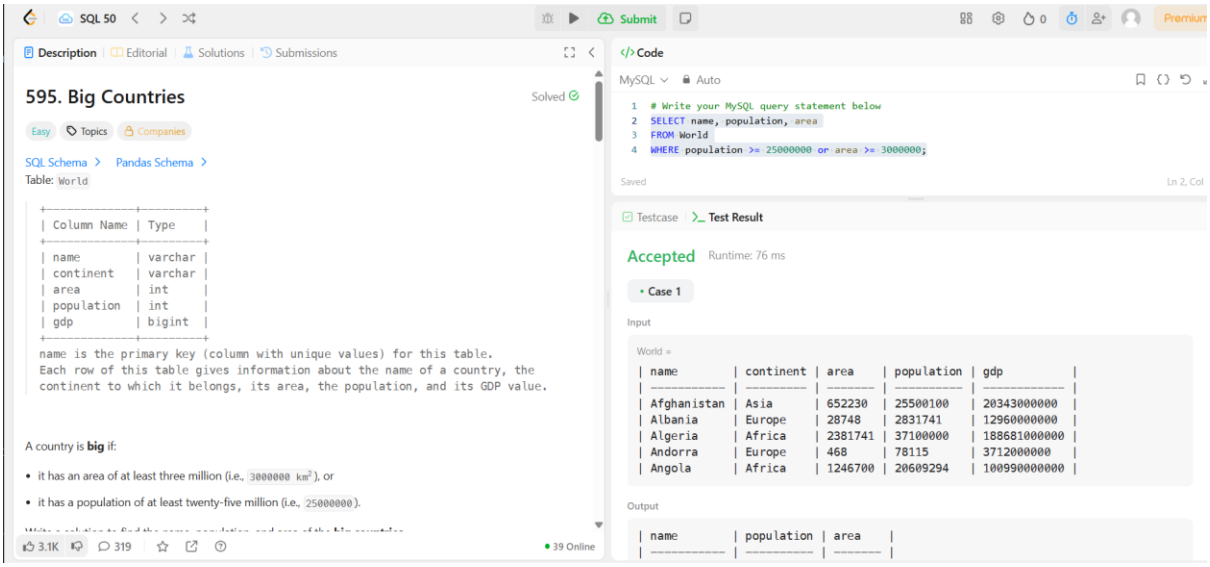
1) 1757. Recyclable and Low Fat Products



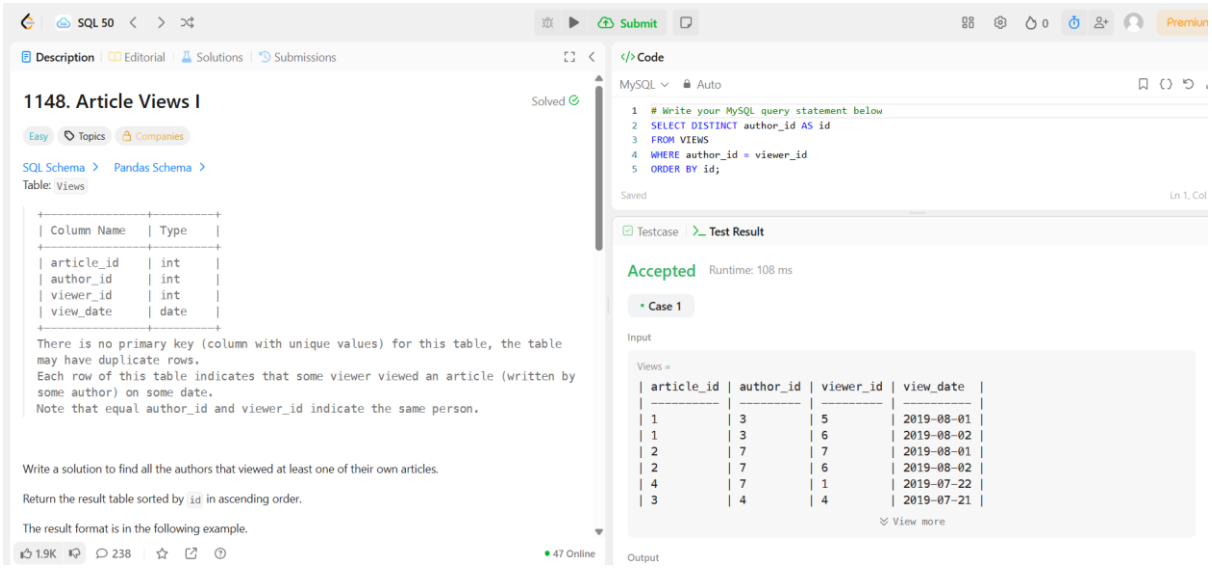
2) 584. Find Customer Referee



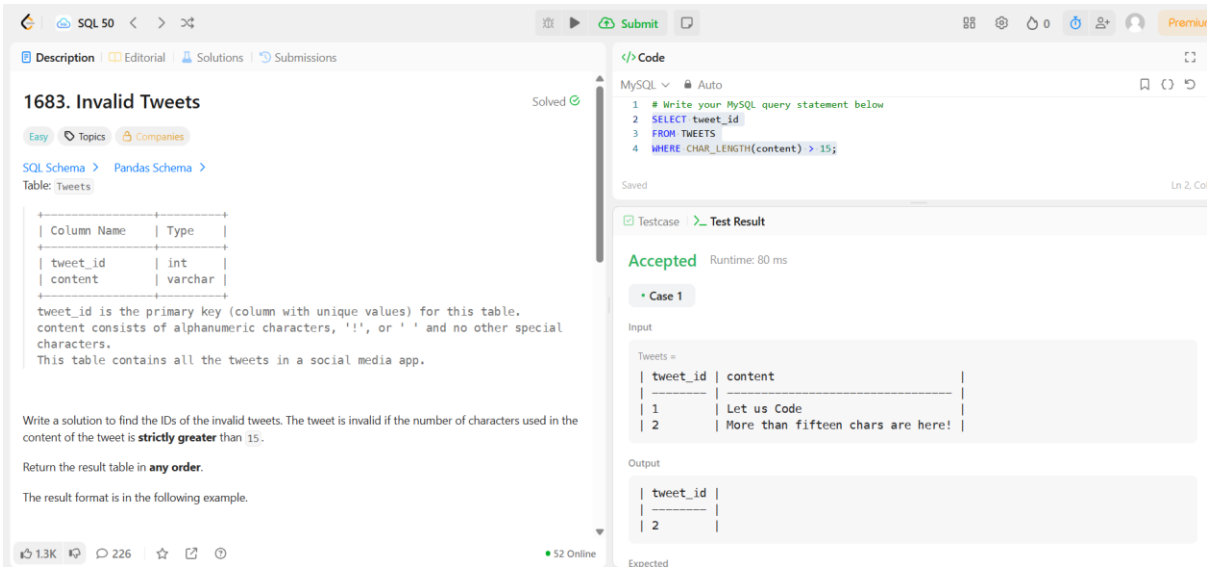
3) 595. Big Countries



4) 1148. Article Views I

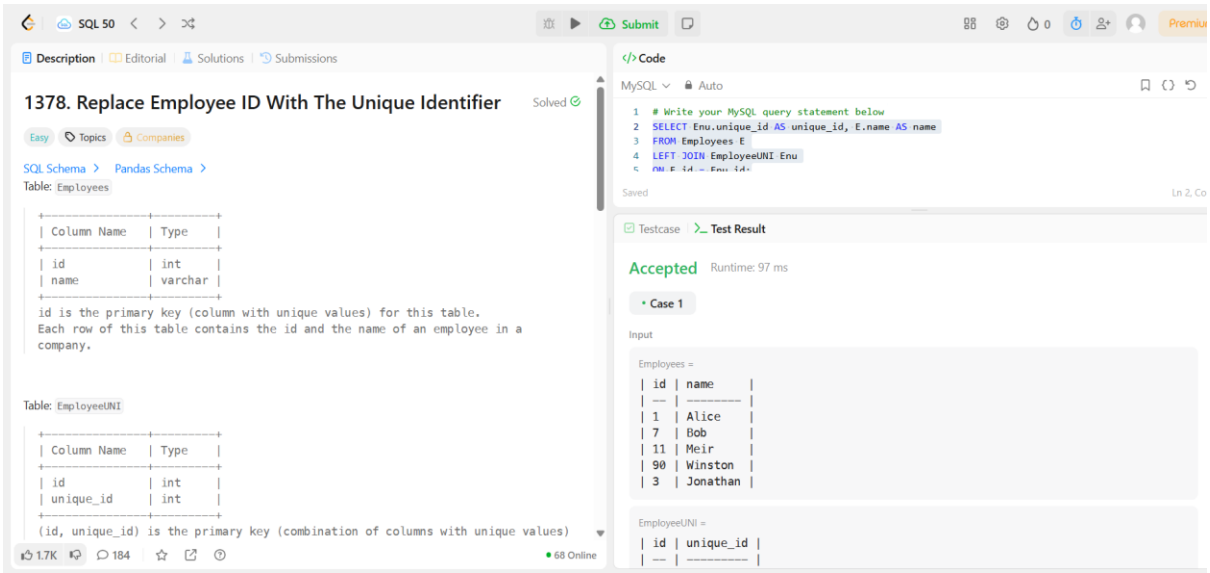


5) 1683. Invalid Tweets

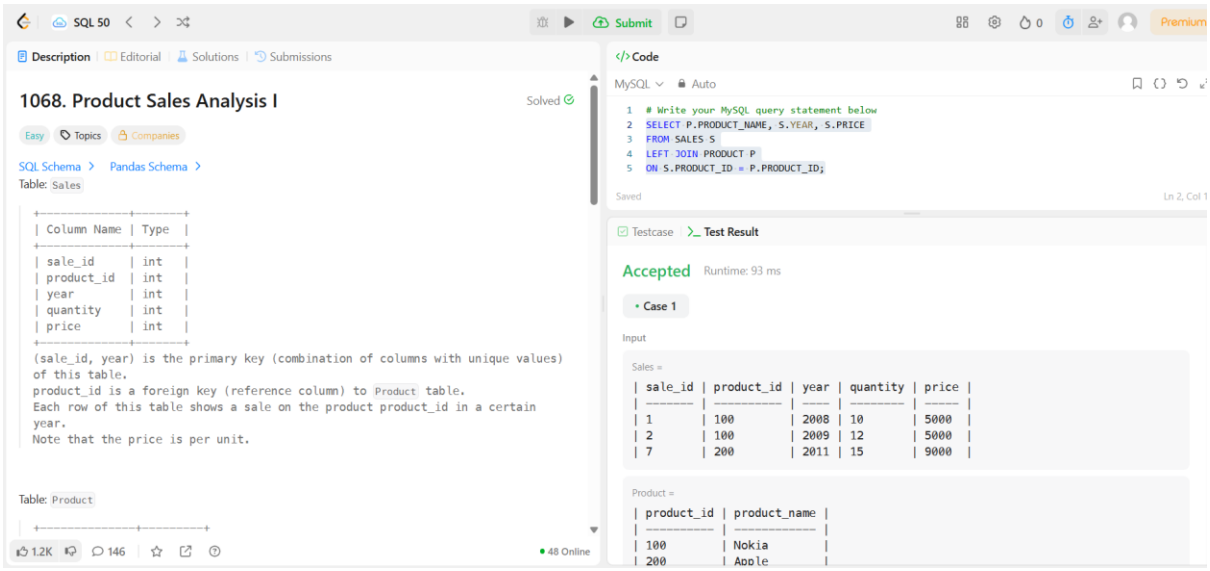


➢ Basic Join

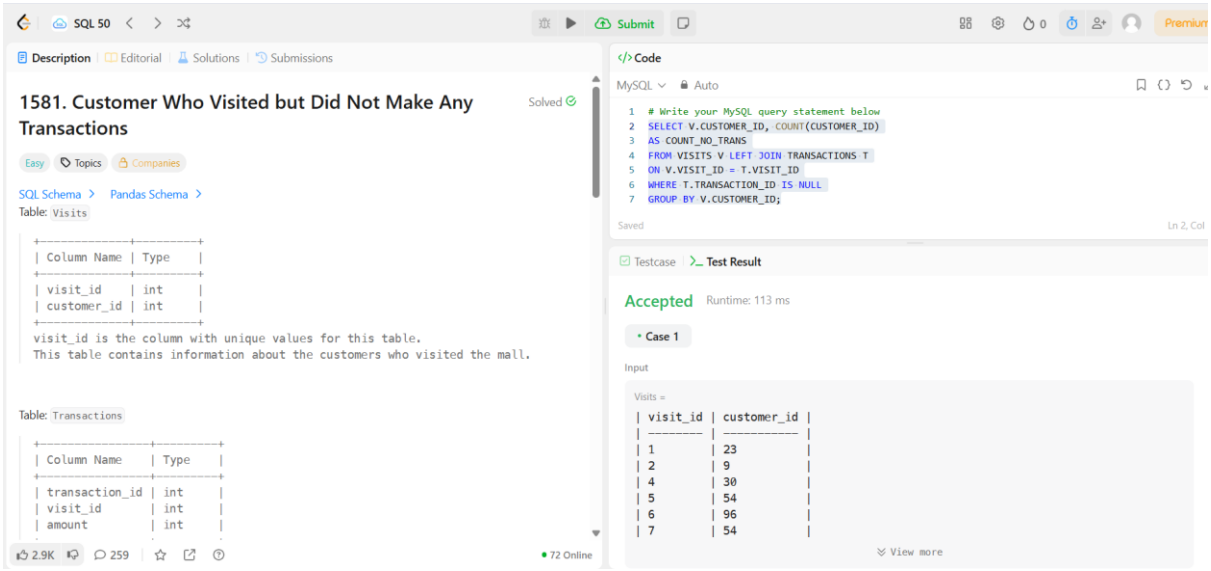
6) 1378. Replace Employee ID With The Unique Identifier



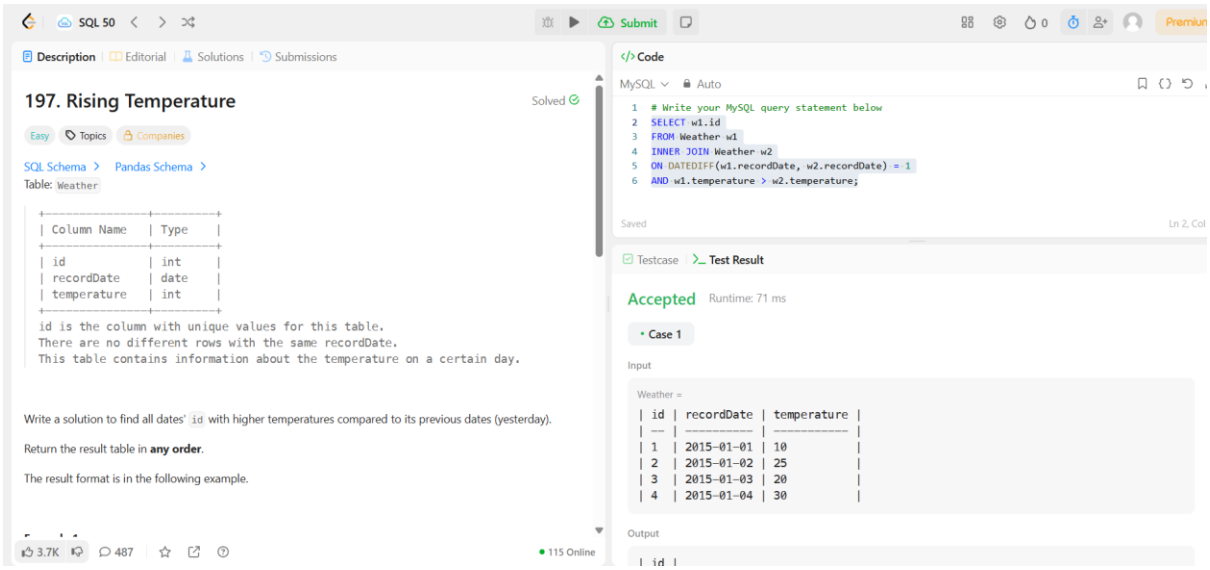
7) 1068. Product Sales Analysis I



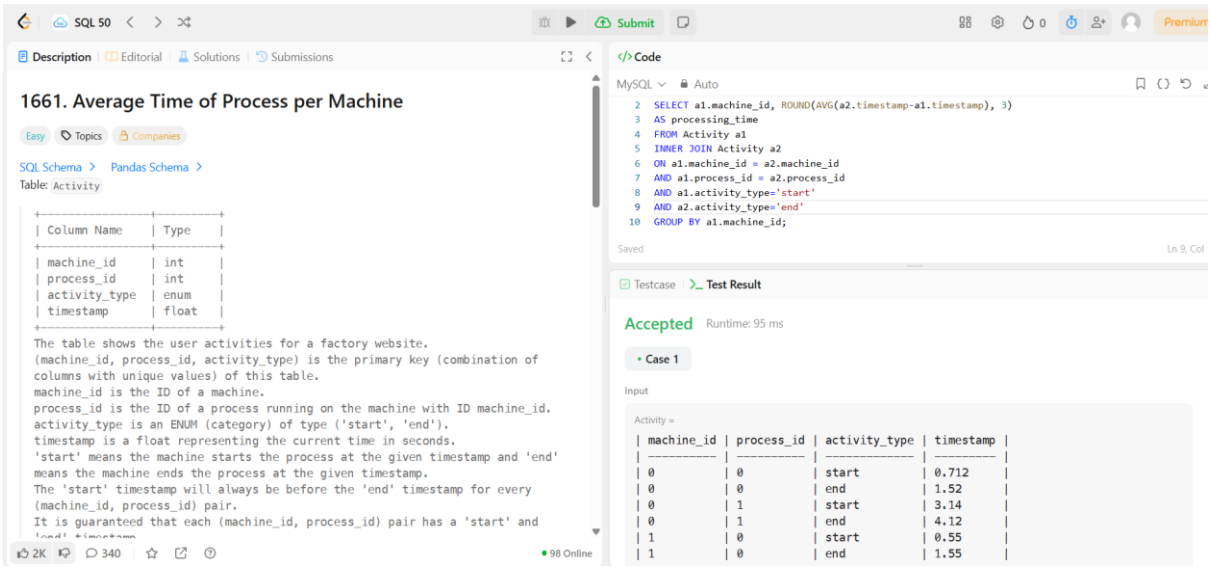
8) 1581. Customer Who Visited but Did Not Make Any Transactions



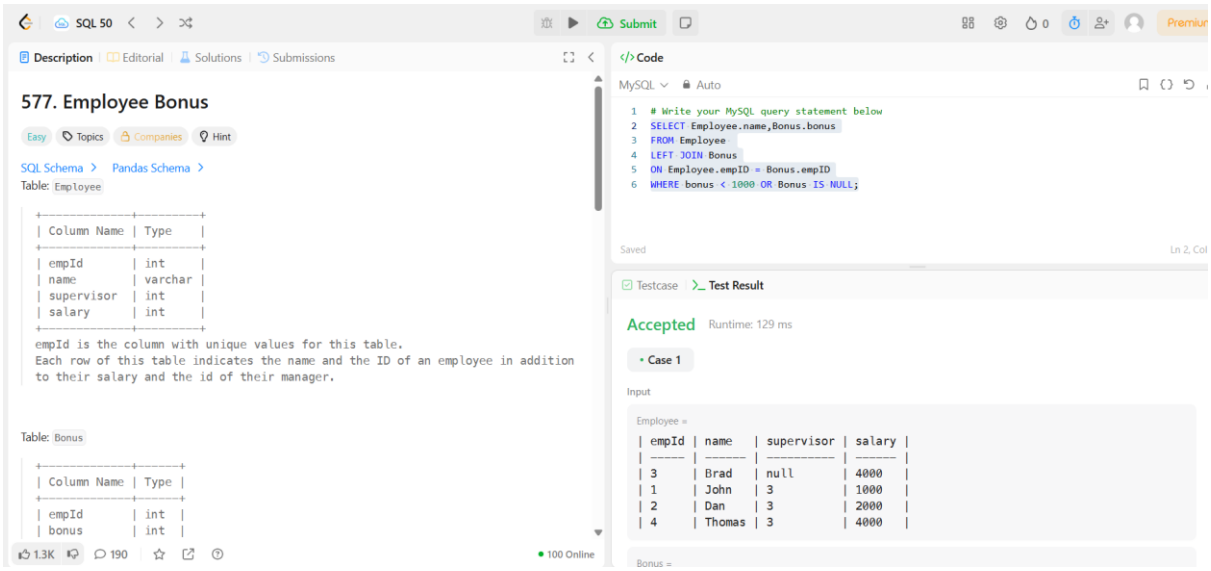
9) 197. Rising Temperature



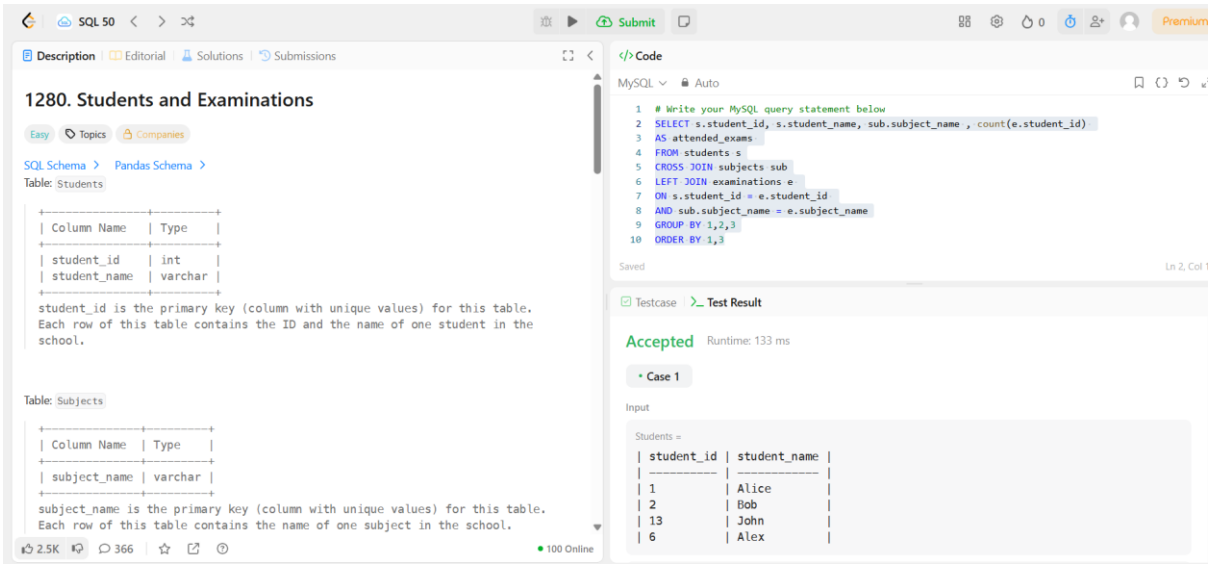
10) [1661. Average Time of Process per Machine]



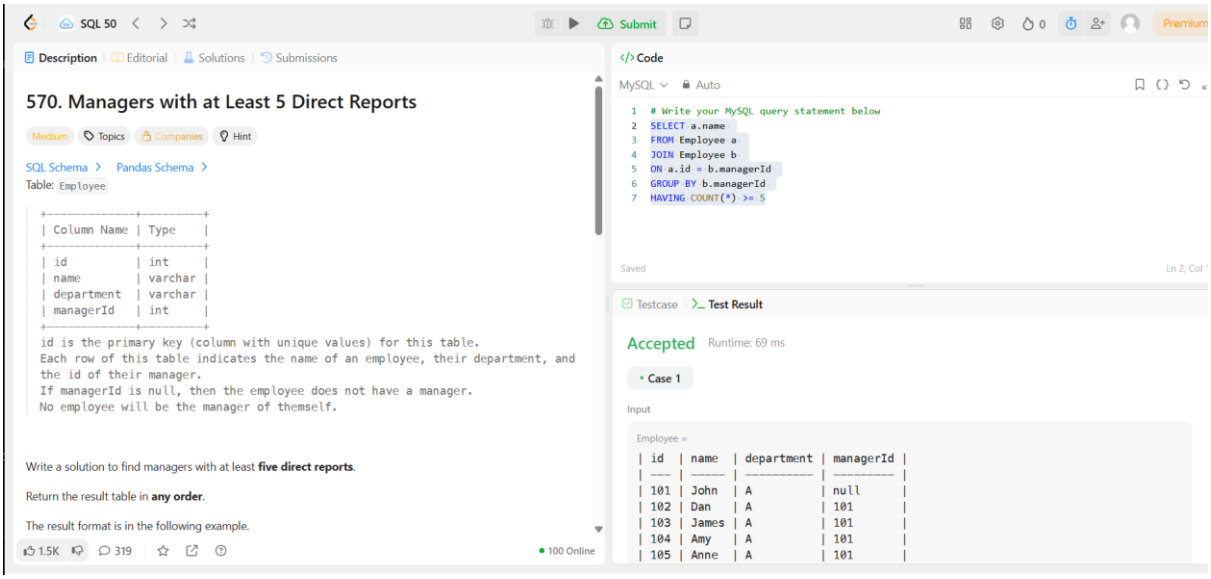
11) [577. Employee Bonus]



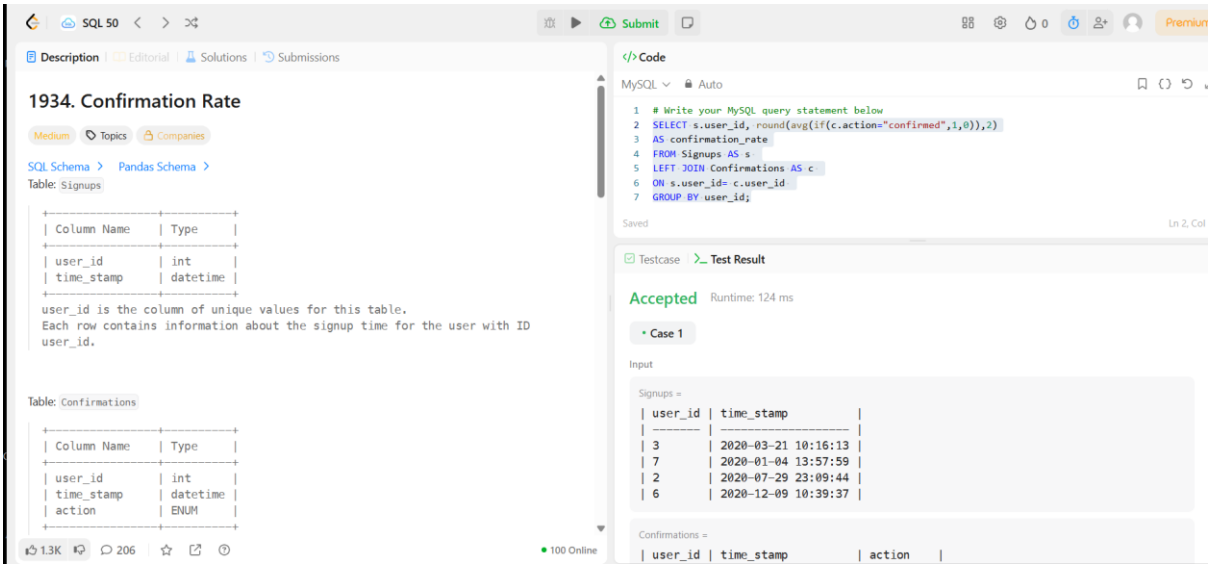
12) [1280. Students and Examinations]



13) [570. Managers with at Least 5 Direct Reports]

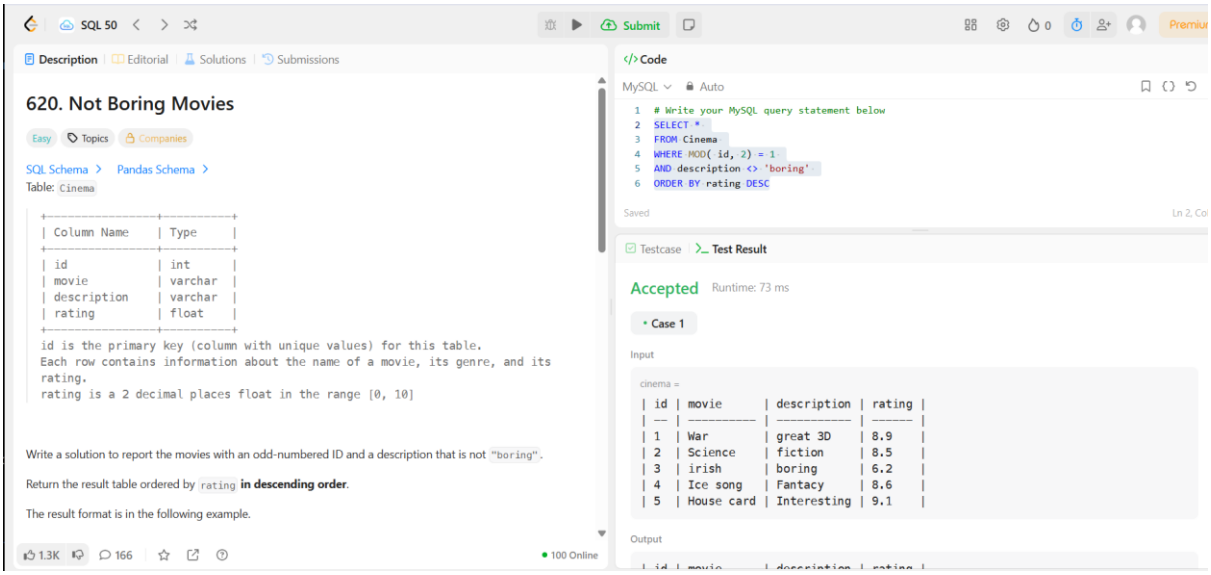


14) [1934. Confirmation Rate]

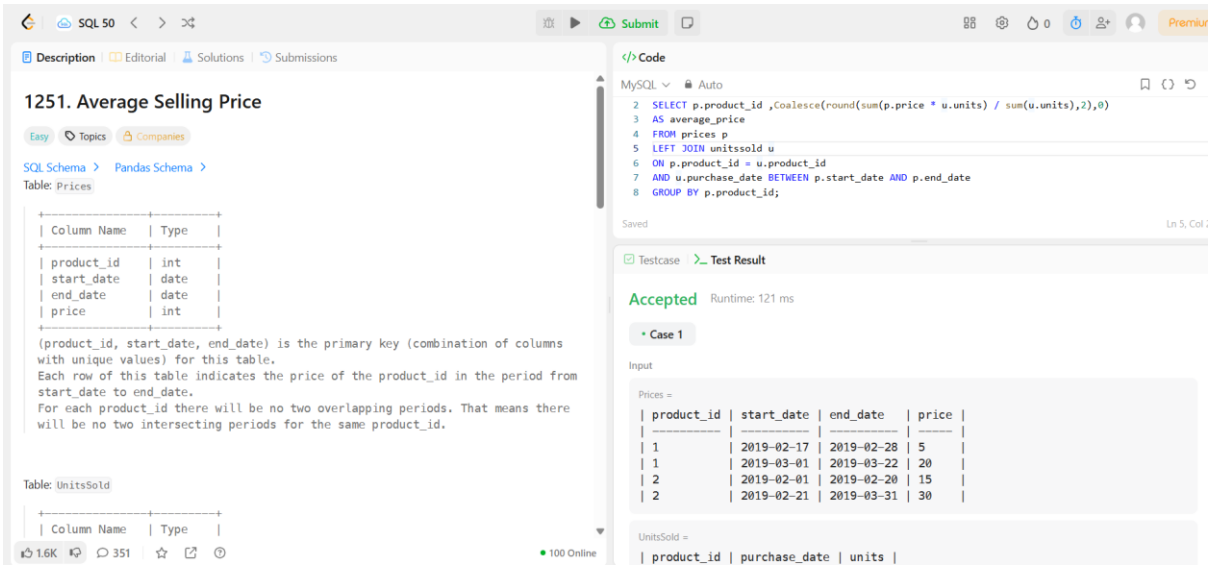


➢ Basic Aggregate Functions

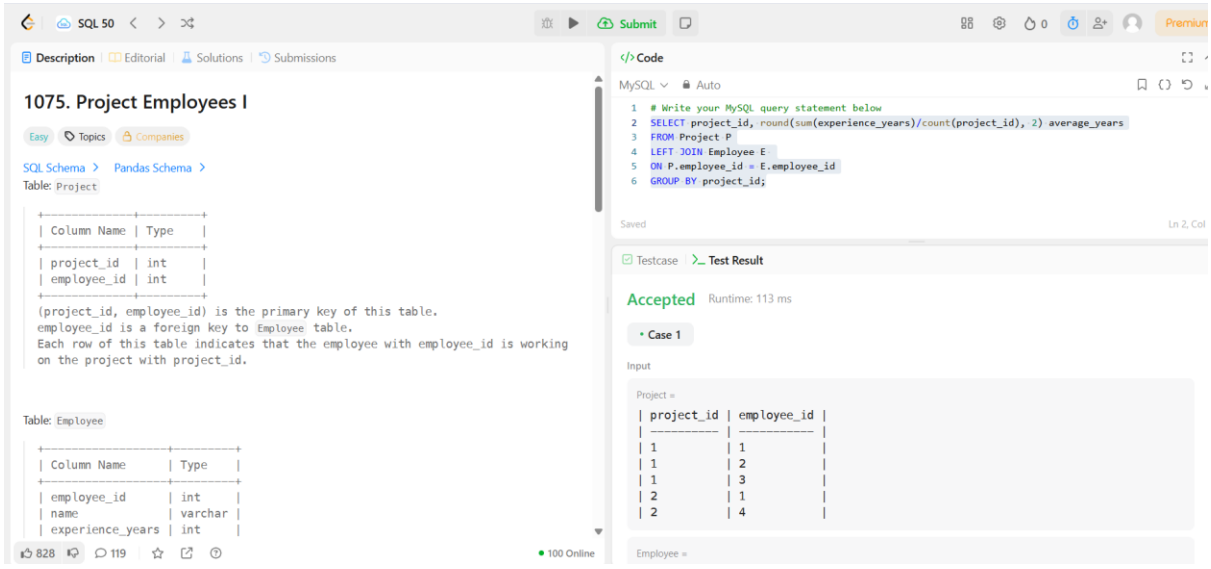
15) 620. Not Boring Movies



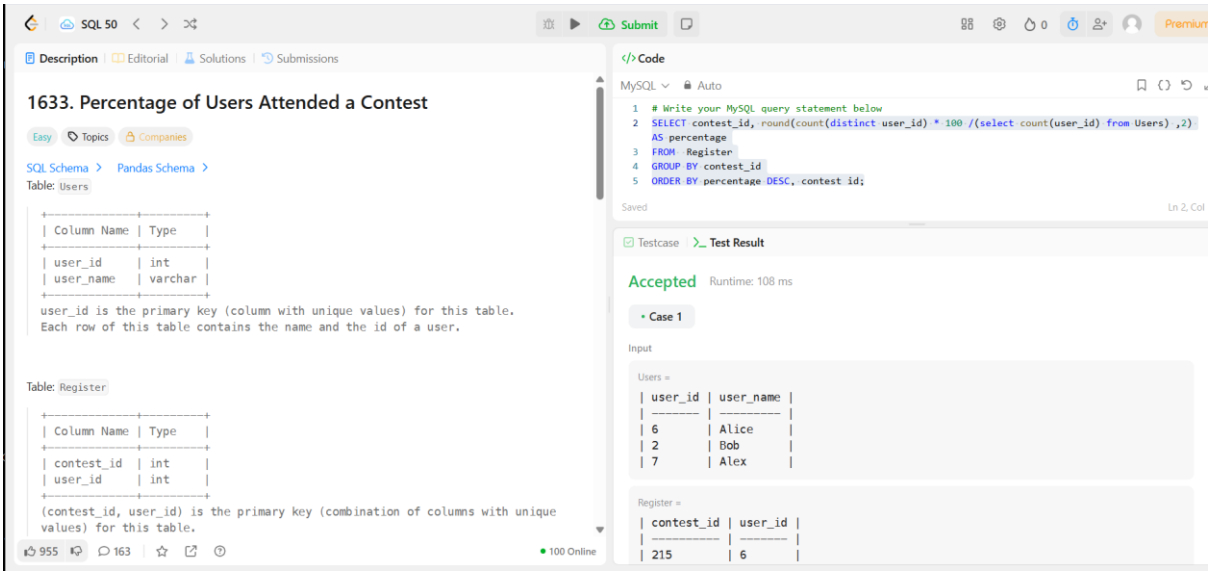
16) 1251. Average Selling Price



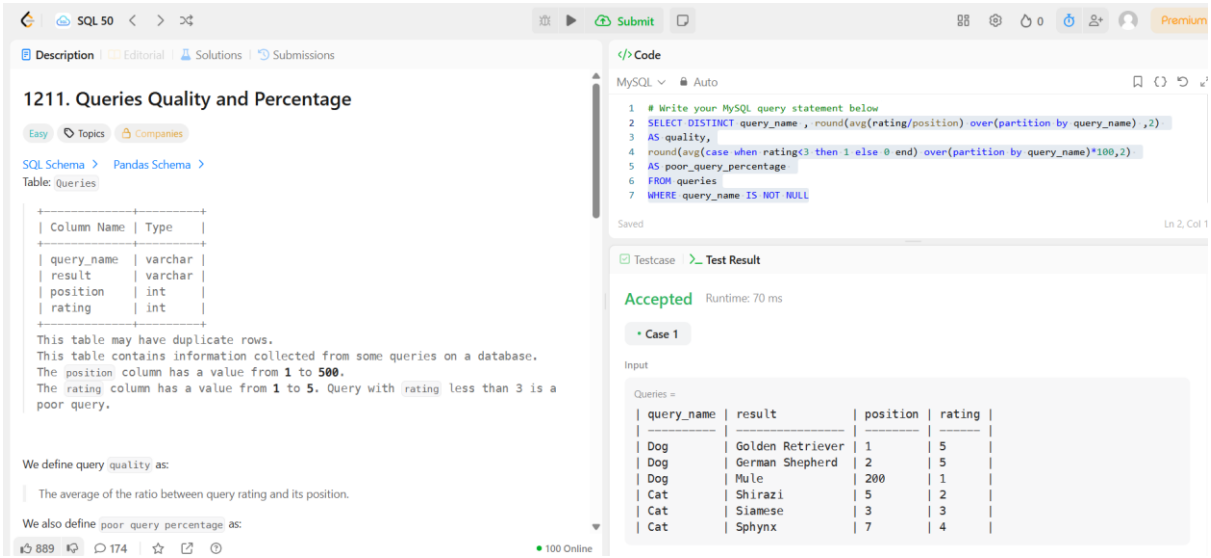
17) 1075. Project Employees I



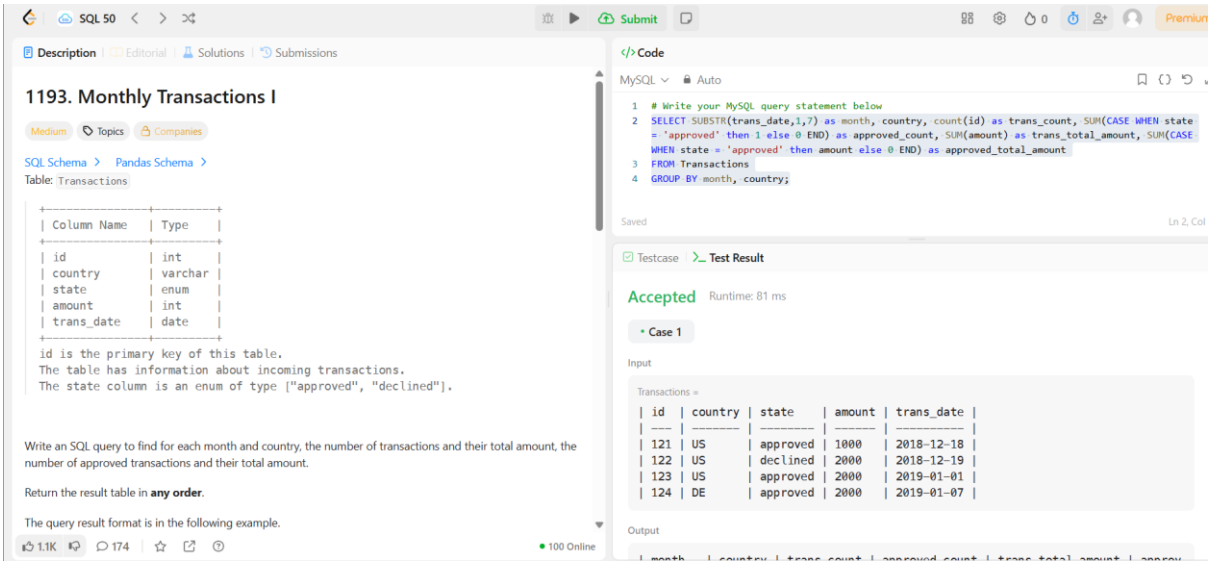
18) 1633. Percentage of Users Attended a Contest



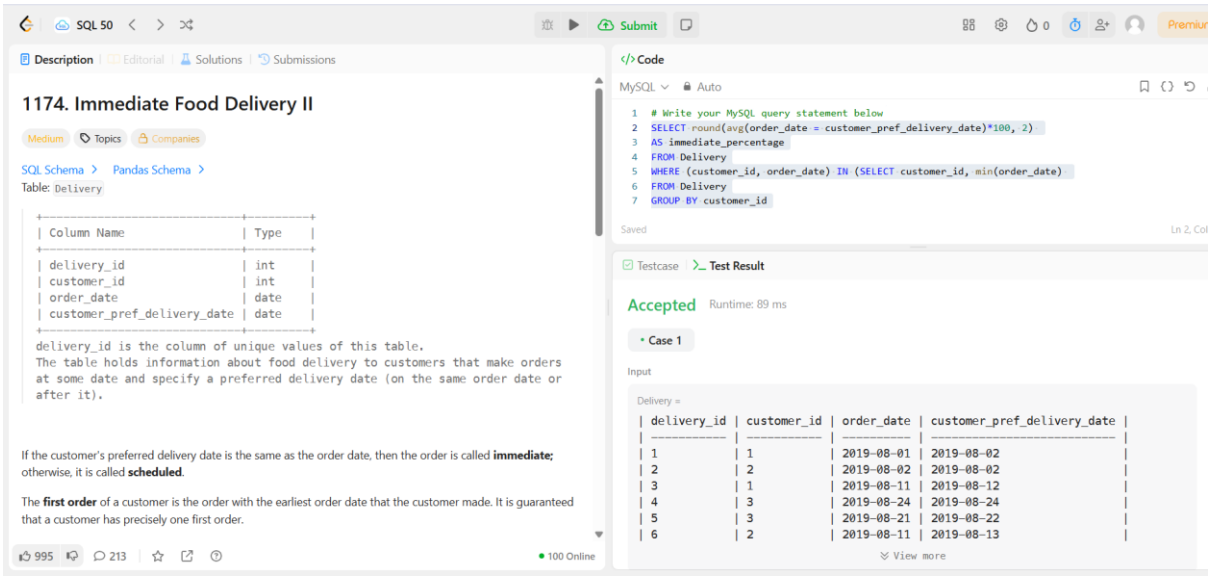
19) [1211. Queries Quality and Percentage]



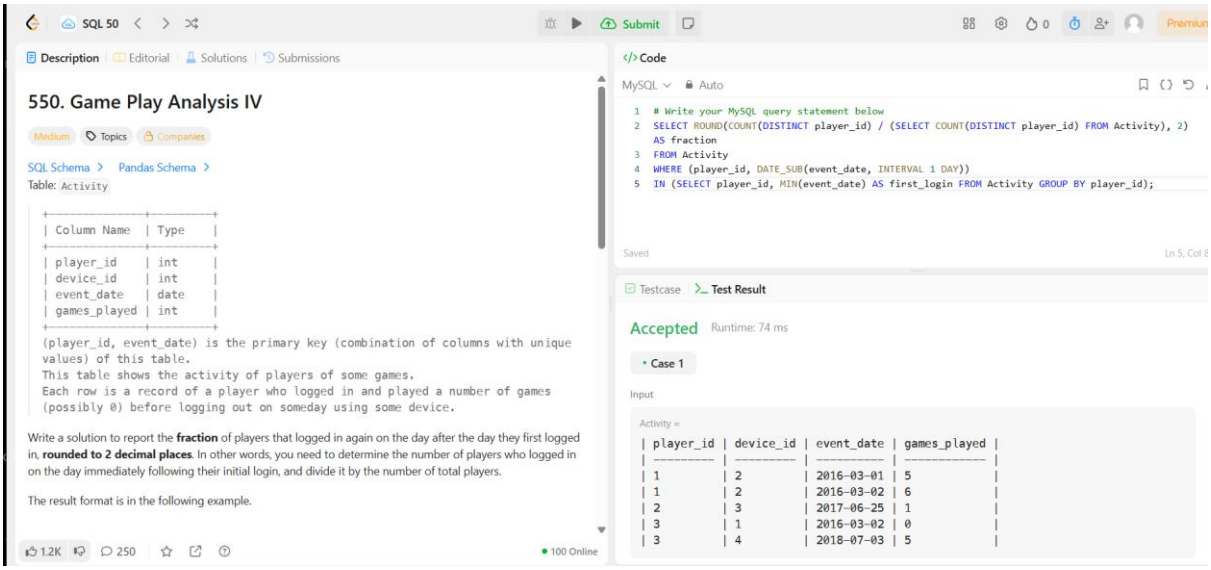
20) [1193. Monthly Transactions I]



21) [1174. Immediate Food Delivery II]

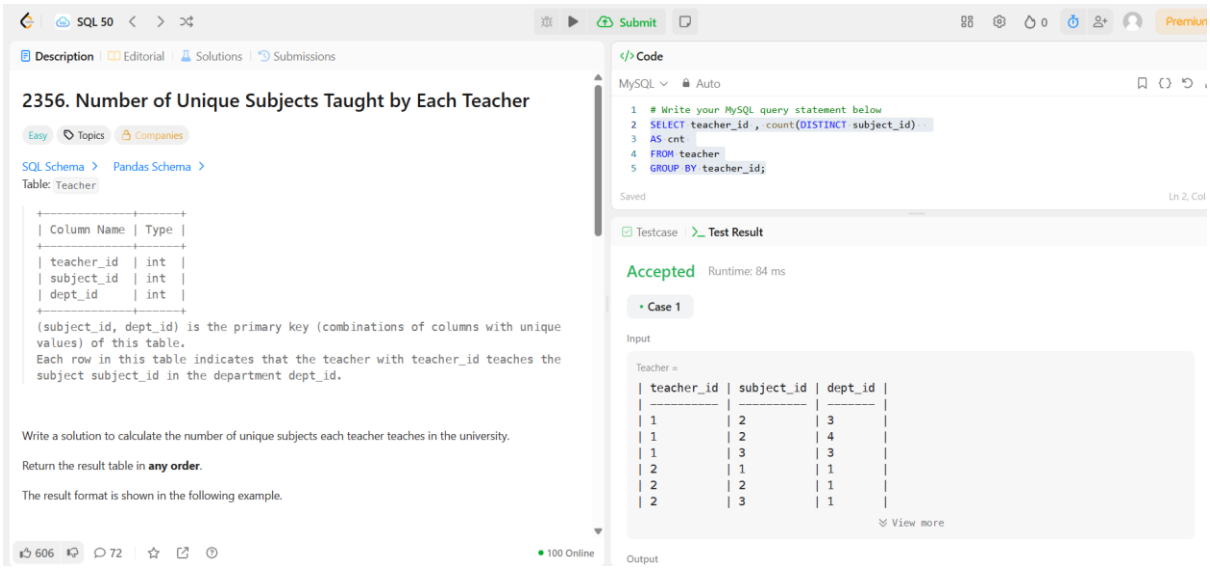


22) [550. Game Play Analysis IV]

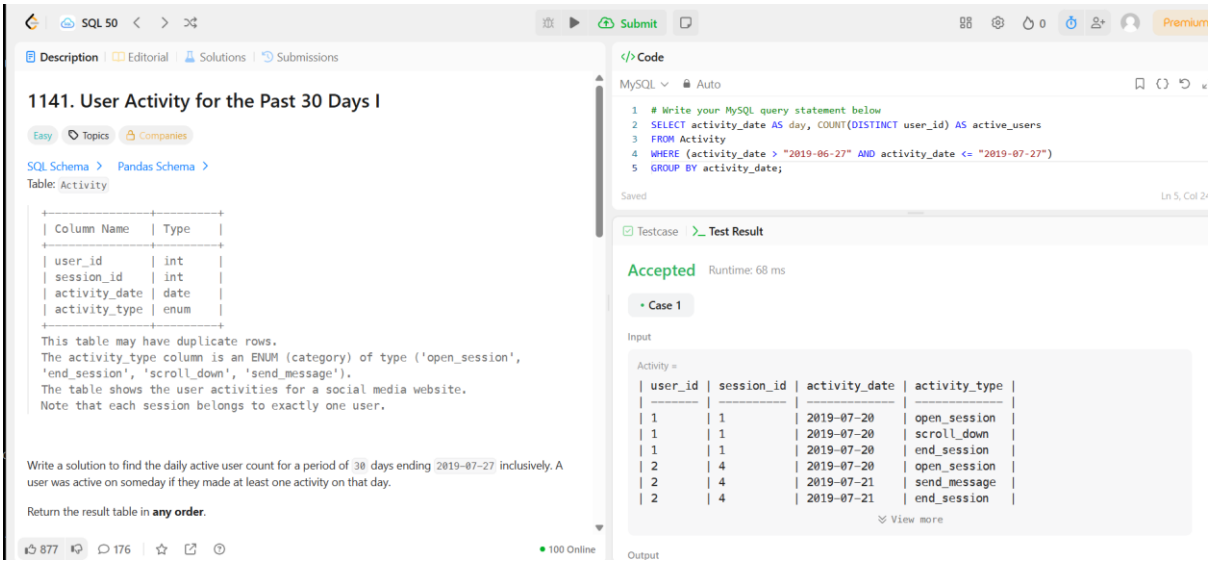


➢ Sorting and Grouping

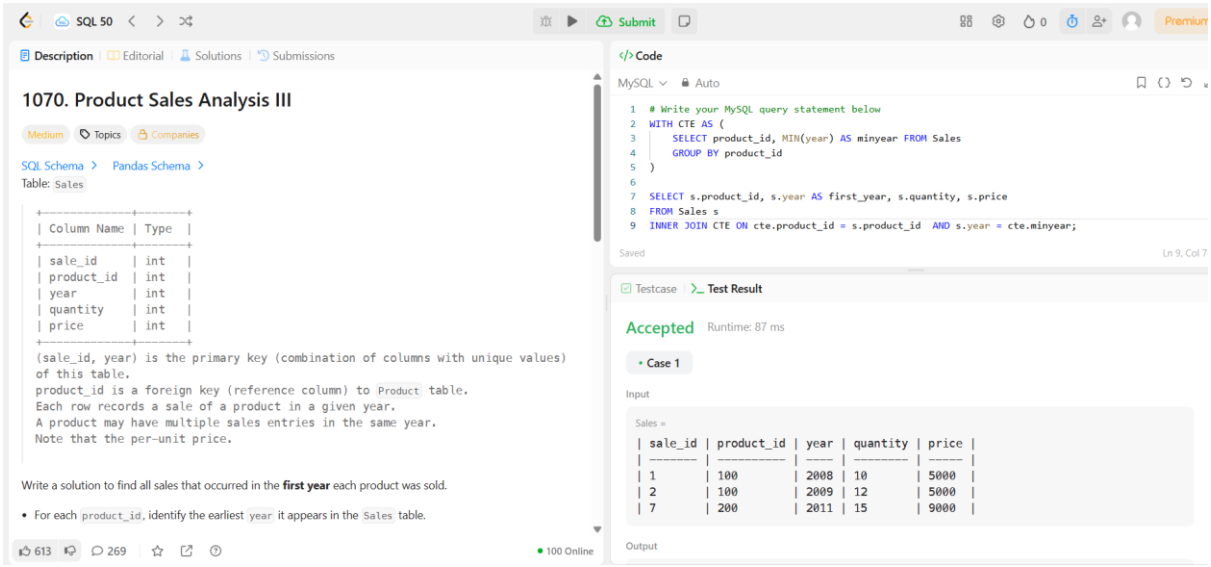
23) 2356. Number of Unique Subjects Taught by Each Teacher



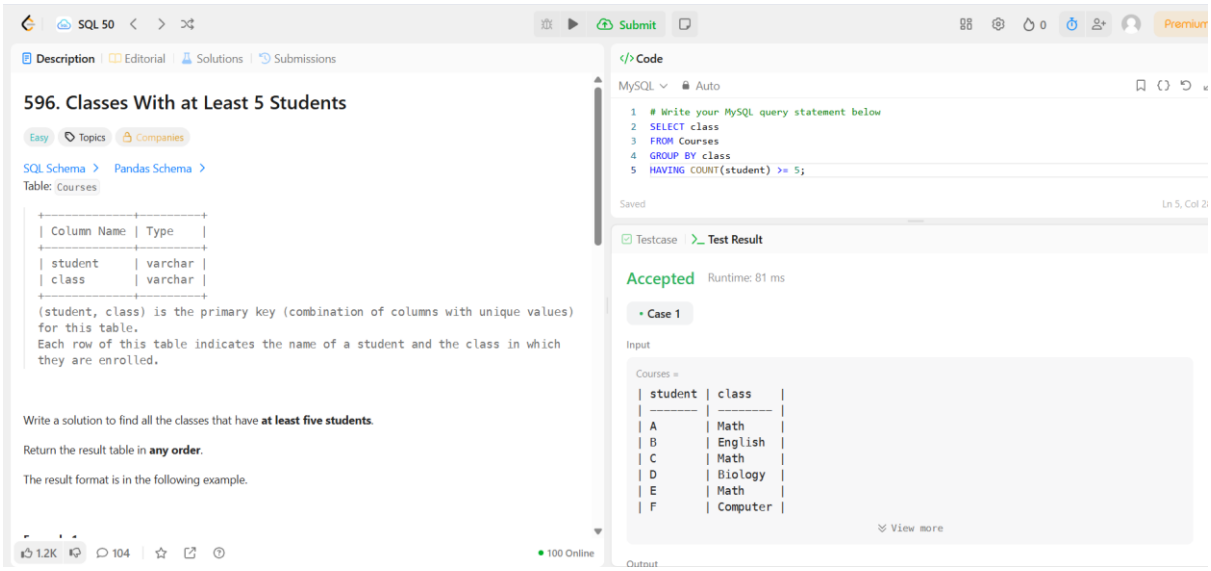
24) 1141. User Activity for the Past 30 Days I



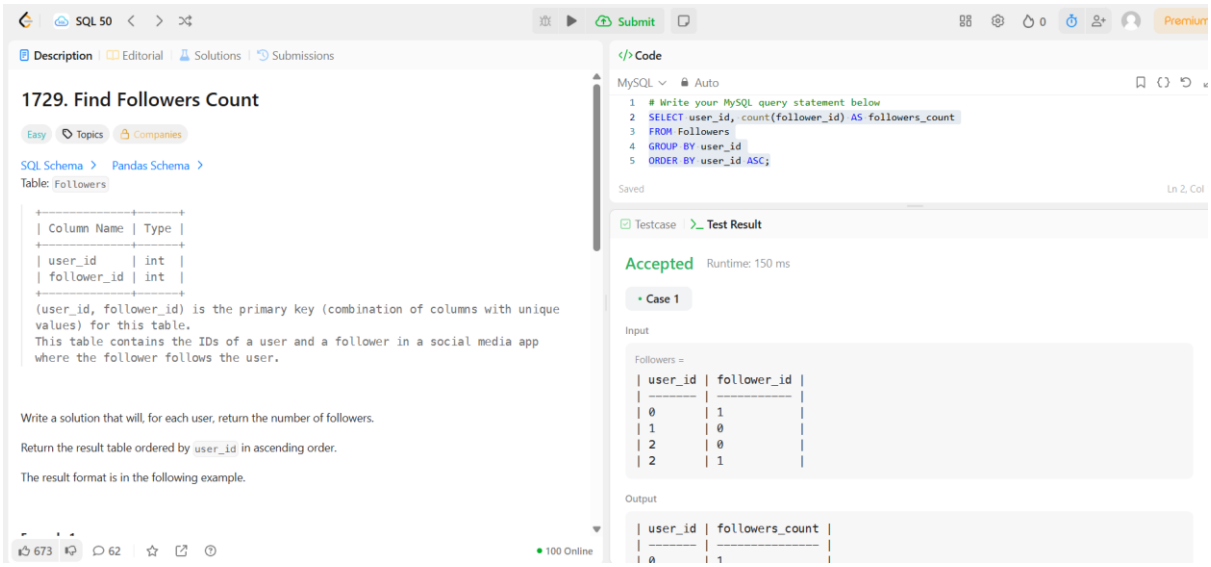
25) [1070. Product Sales Analysis III]



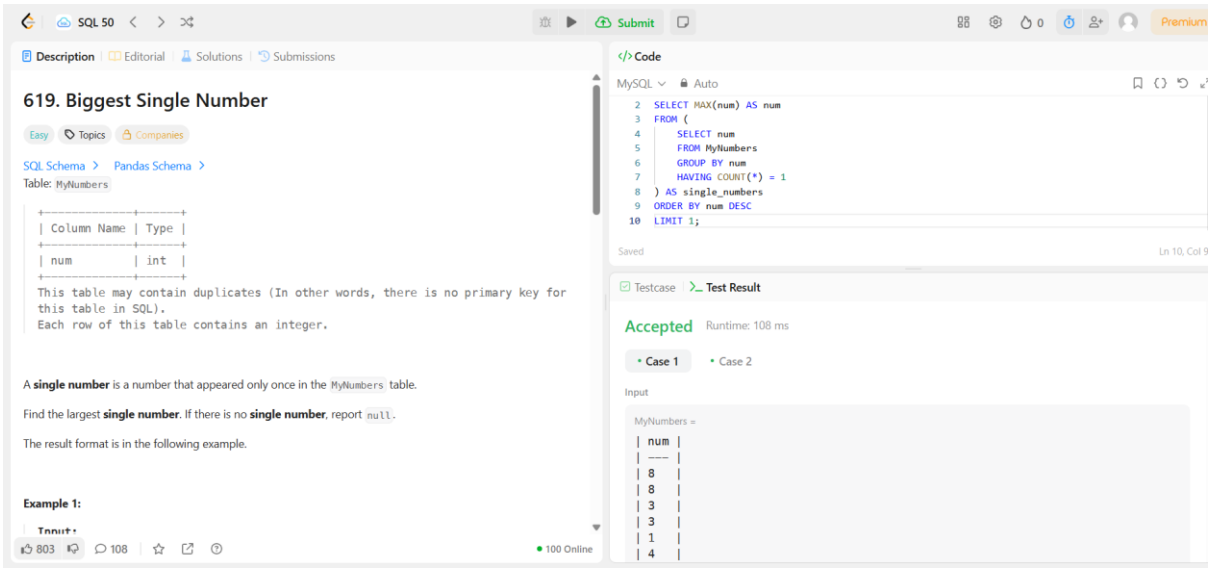
26) [596. Classes With at Least 5 Students]



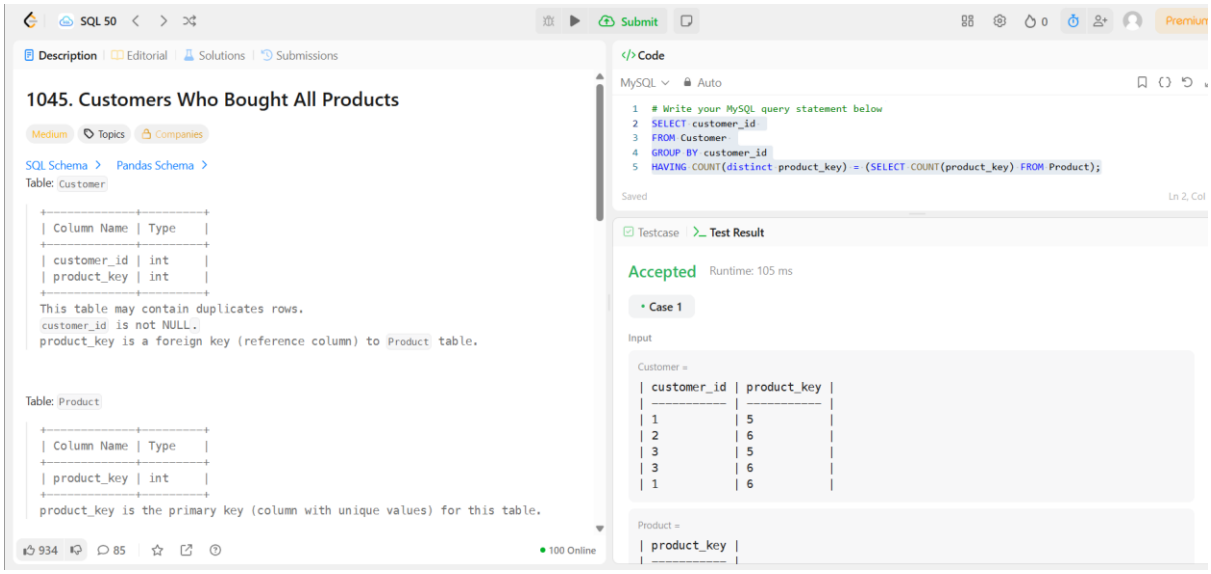
27) [1729. Find Followers Count]



28) [619. Biggest Single Number]

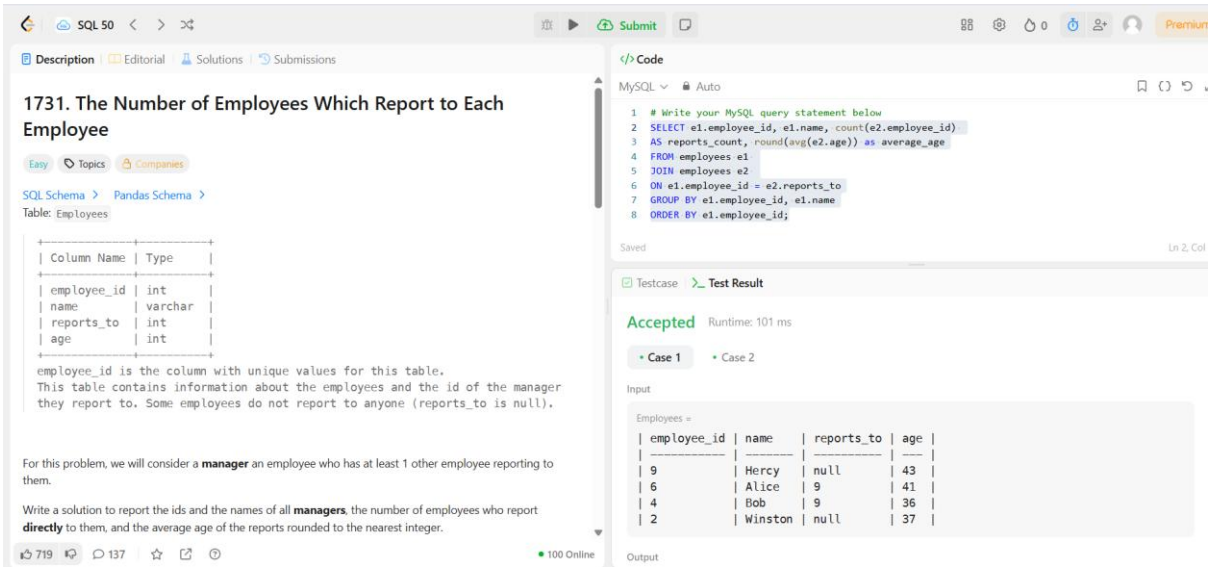


29) [1045. Customers Who Bought All Products]

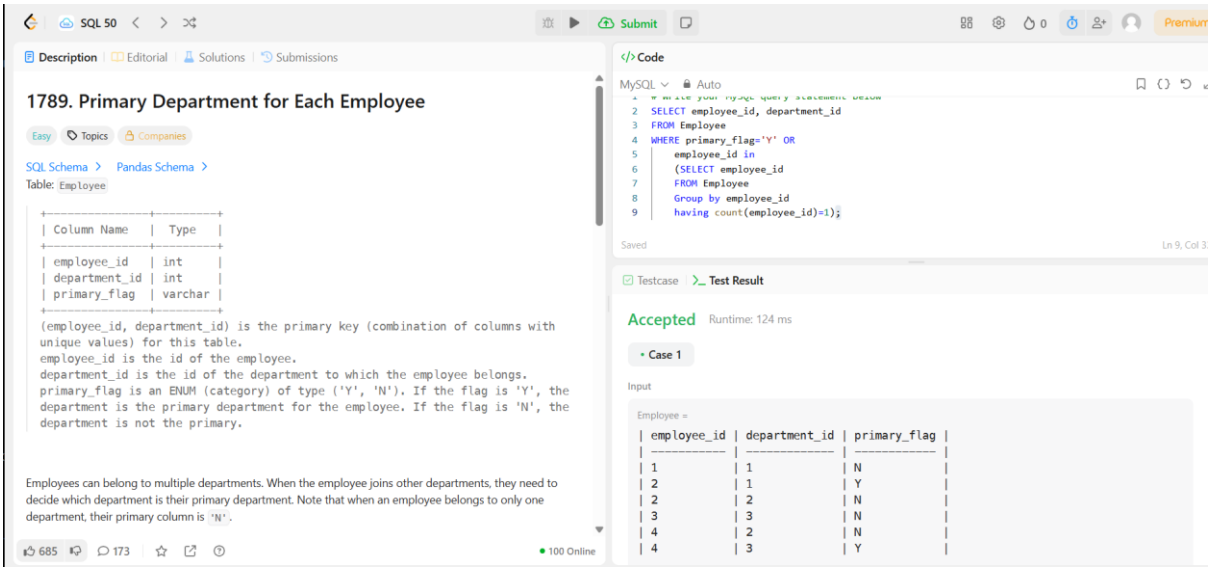


➢ Advanced Select and Joins

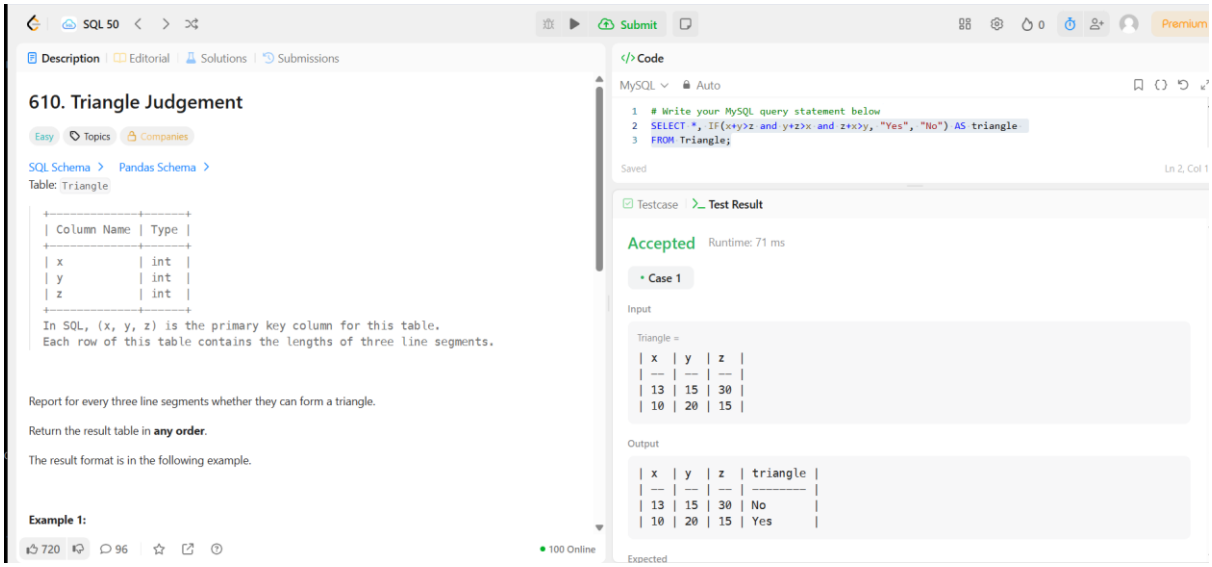
30) [1731. The Number of Employees Which Report to Each Employee]



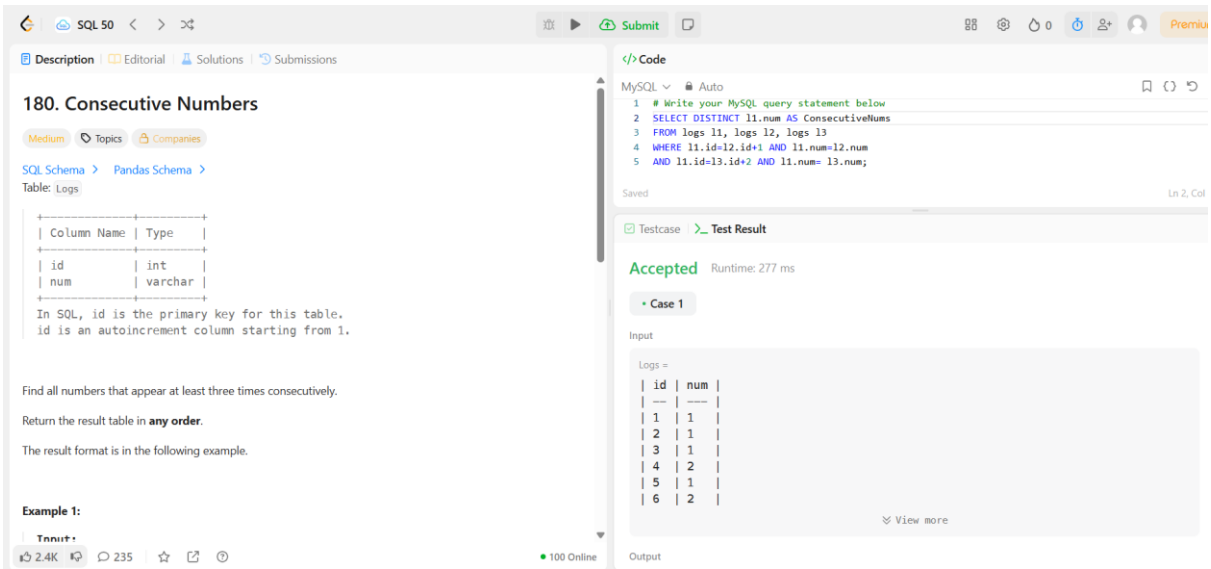
31) [1789. Primary Department for Each Employee]



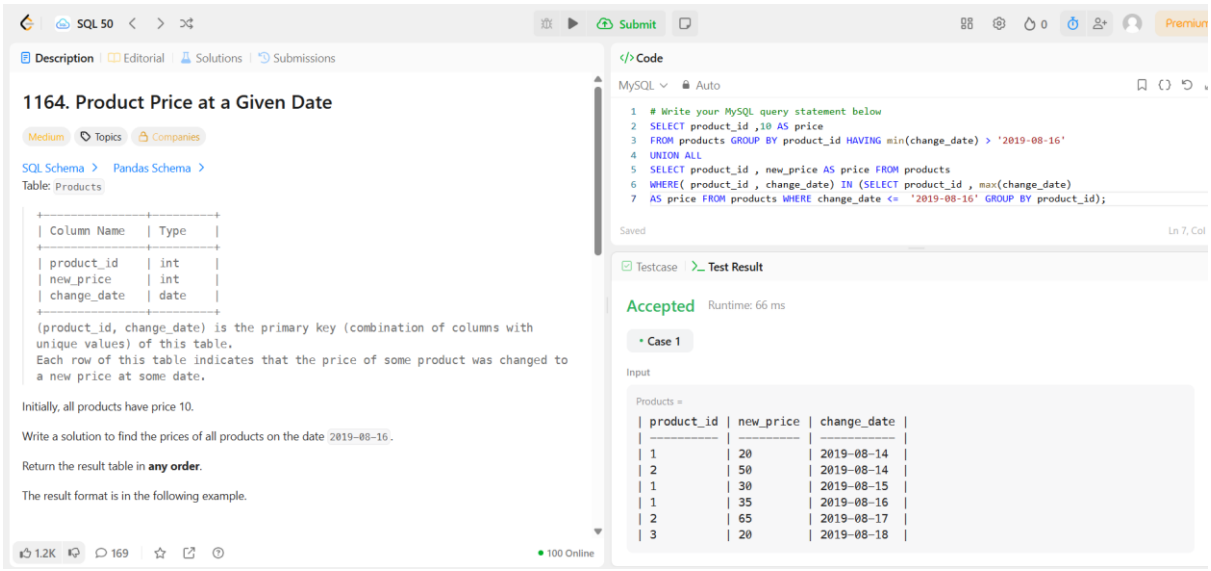
32) [610. Triangle Judgement]



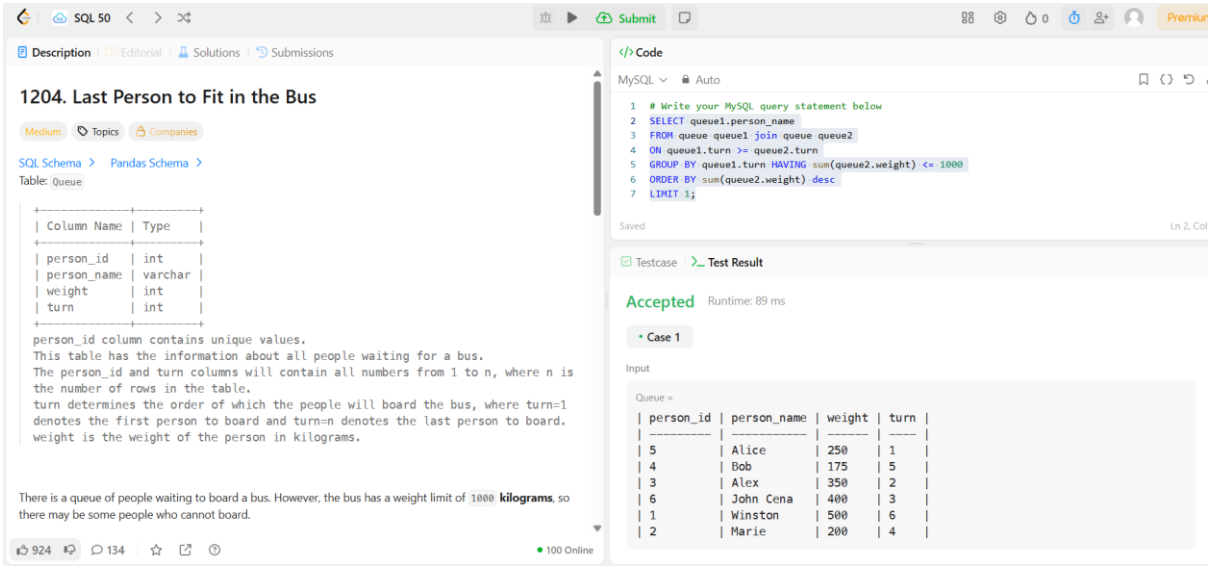
33) [180. Consecutive Numbers]



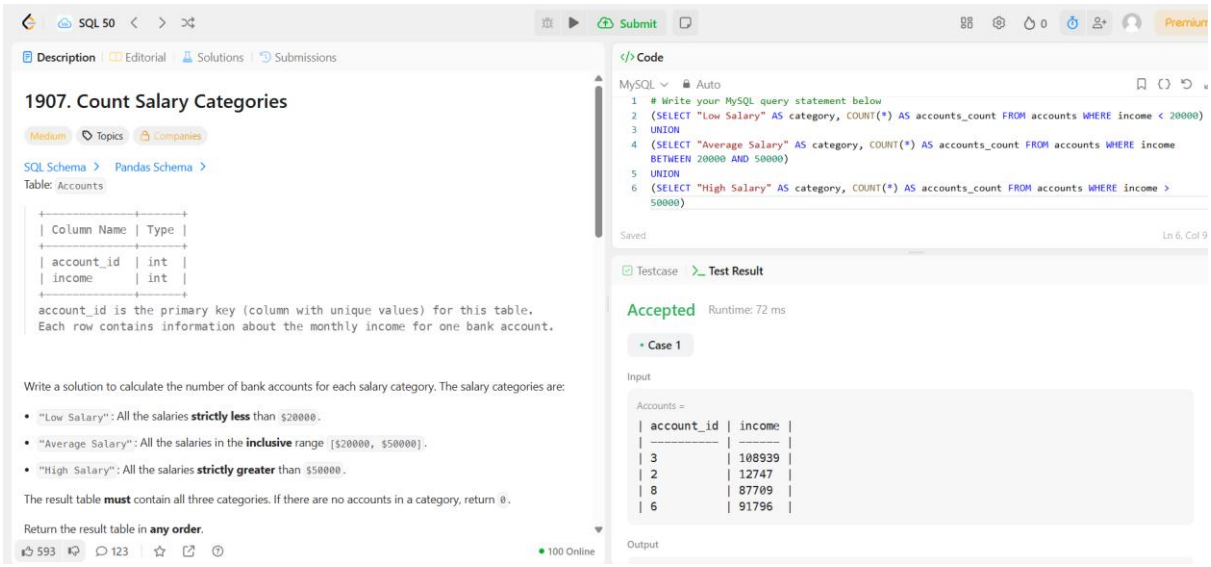
34) [1164. Product Price at a Given Date]



35) 1204. Last Person to Fit in the Bus

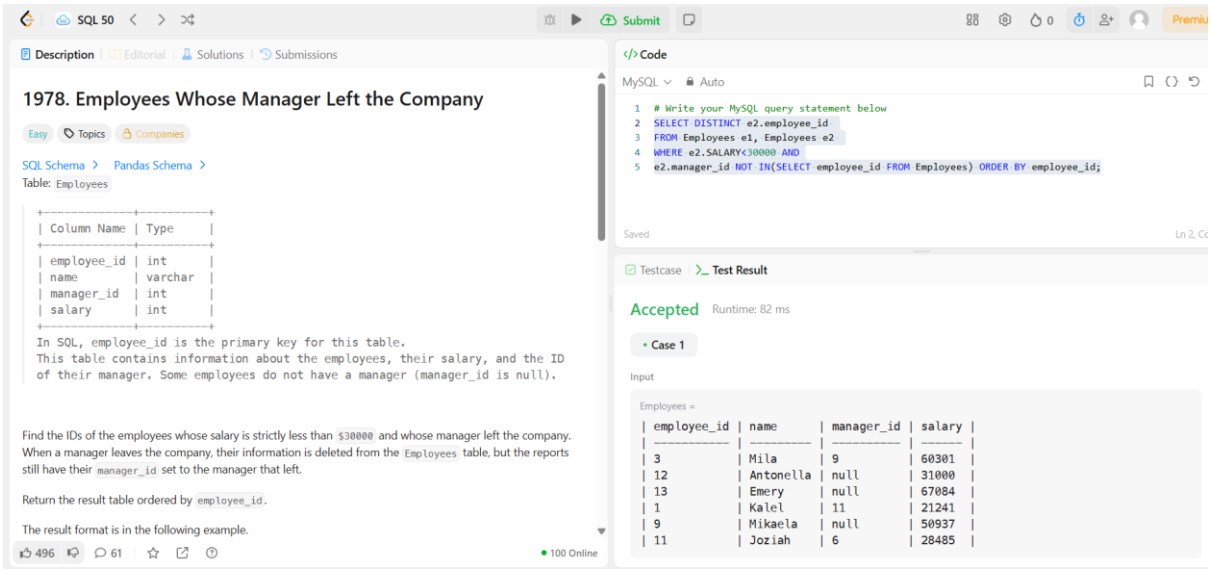


36) 1907. Count Salary Categories

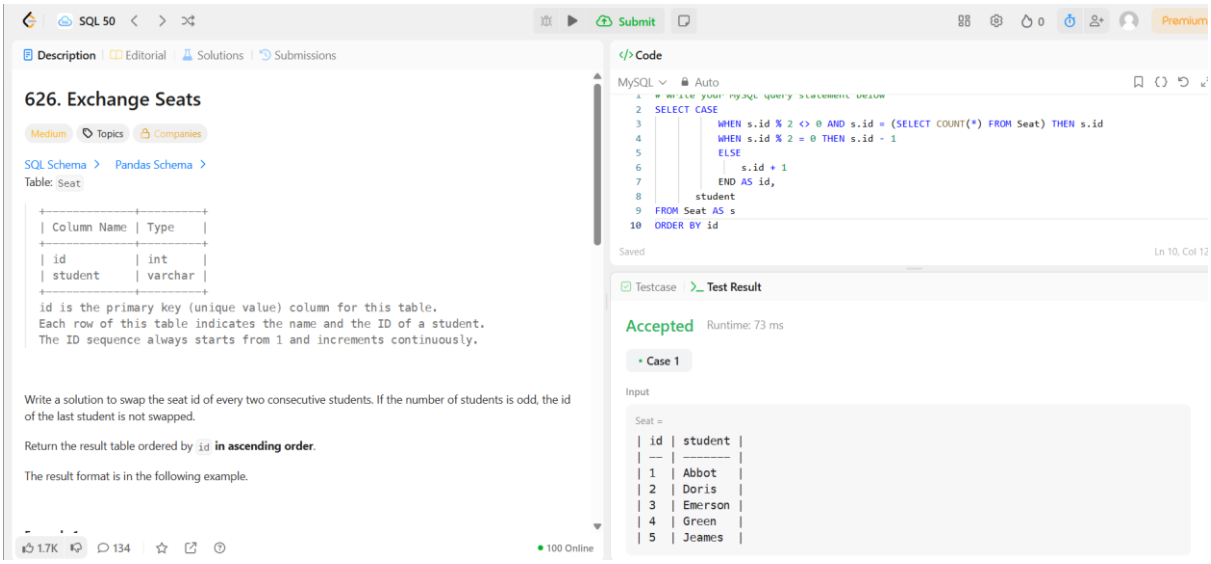


➢ Subqueries

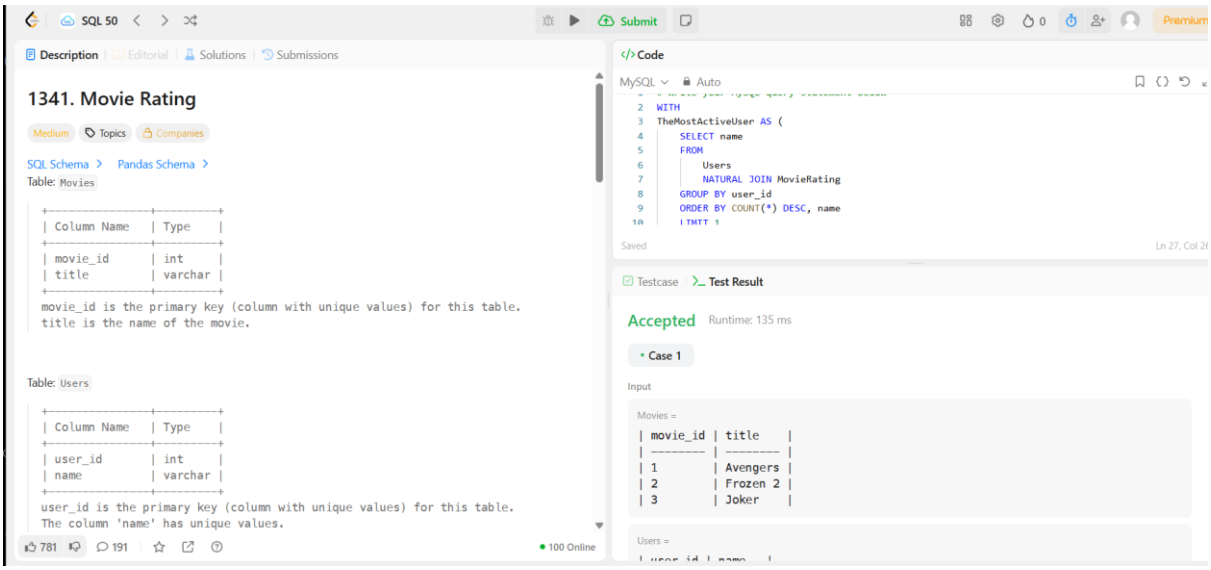
37) 1978. Employees Whose Manager Left the Company



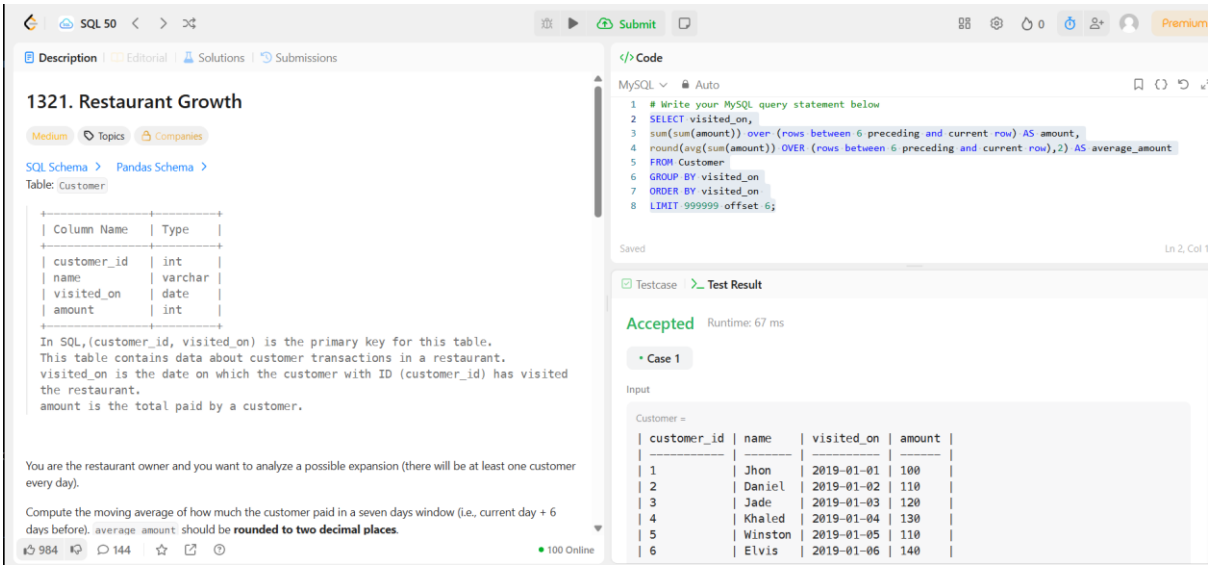
38) 626. Exchange Seats



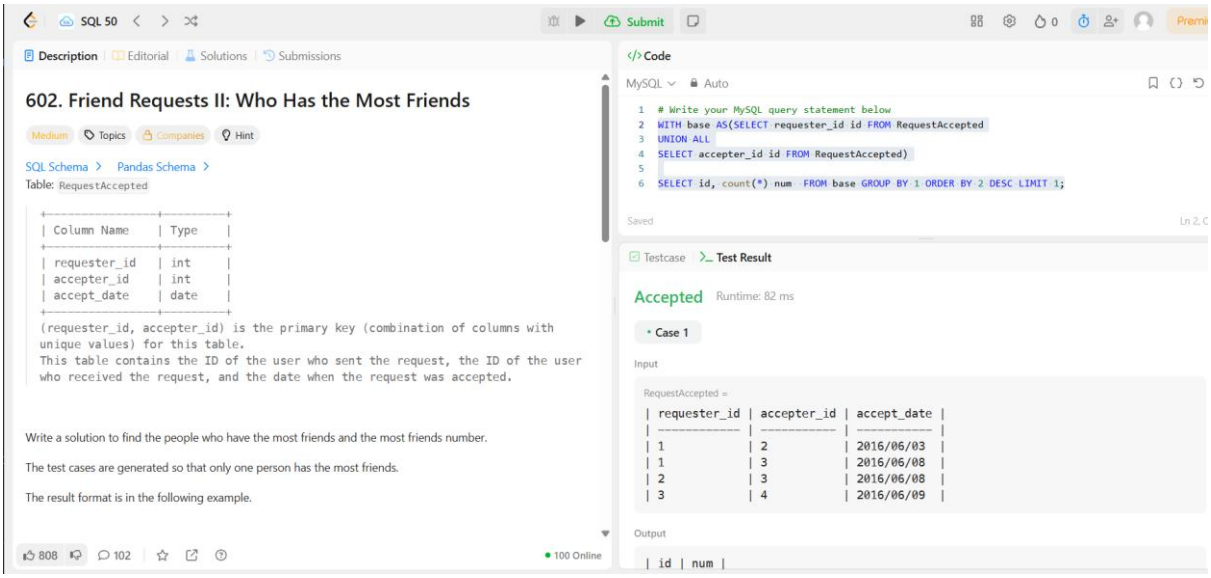
39) [1341. Movie Rating]



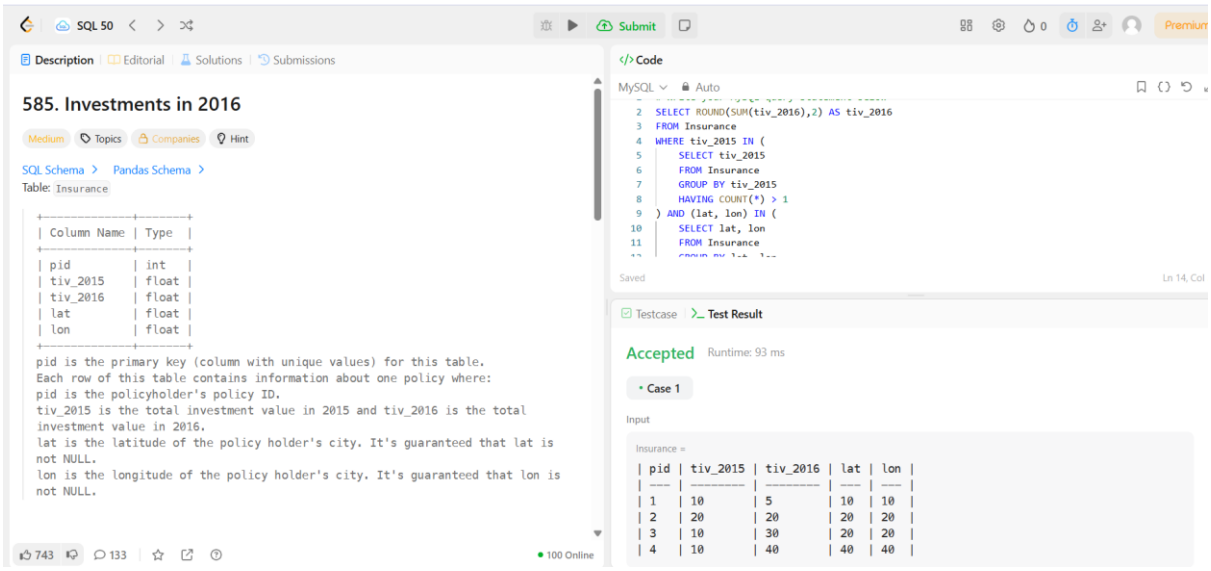
40) [1321. Restaurant Growth]



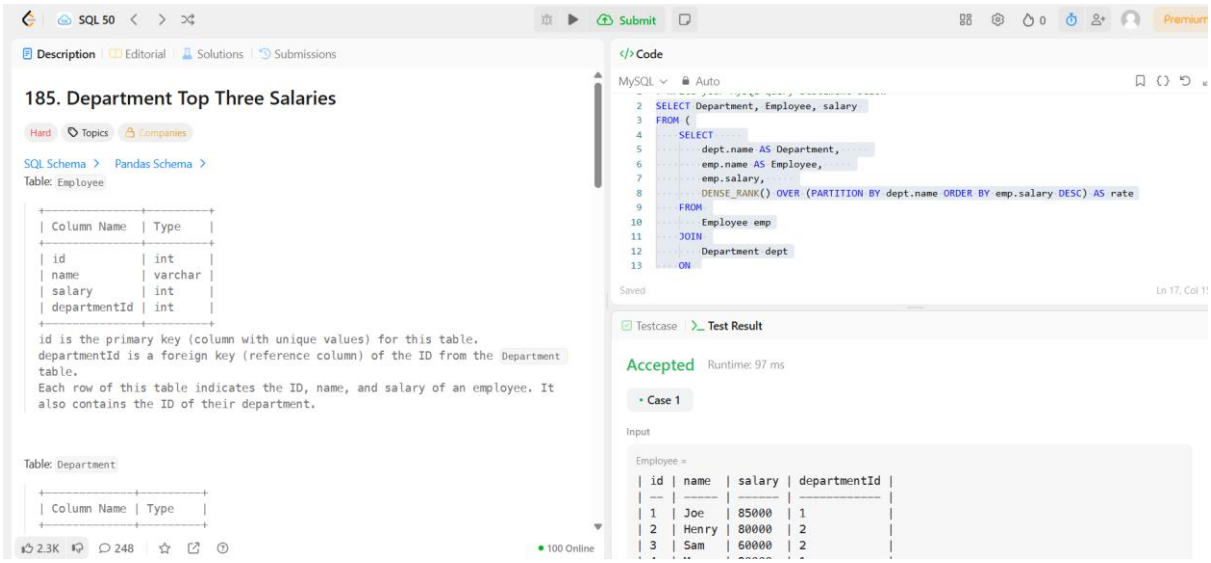
41) 602. Friend Requests II: Who Has the Most Friends



42) 585. Investments in 2016

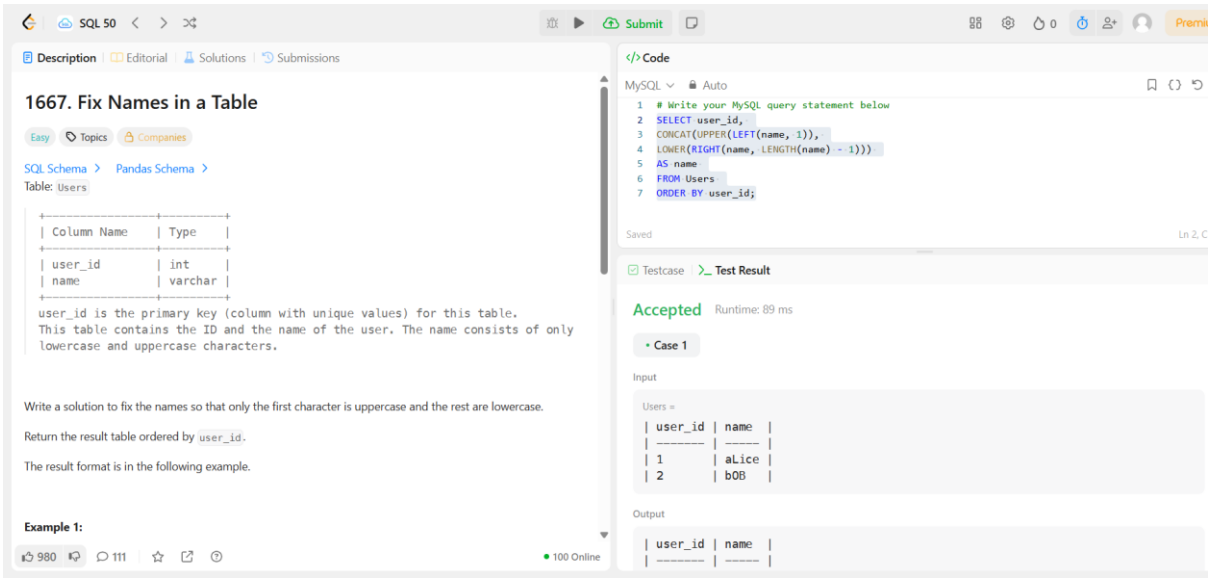


43) 185. Department Top Three Salaries

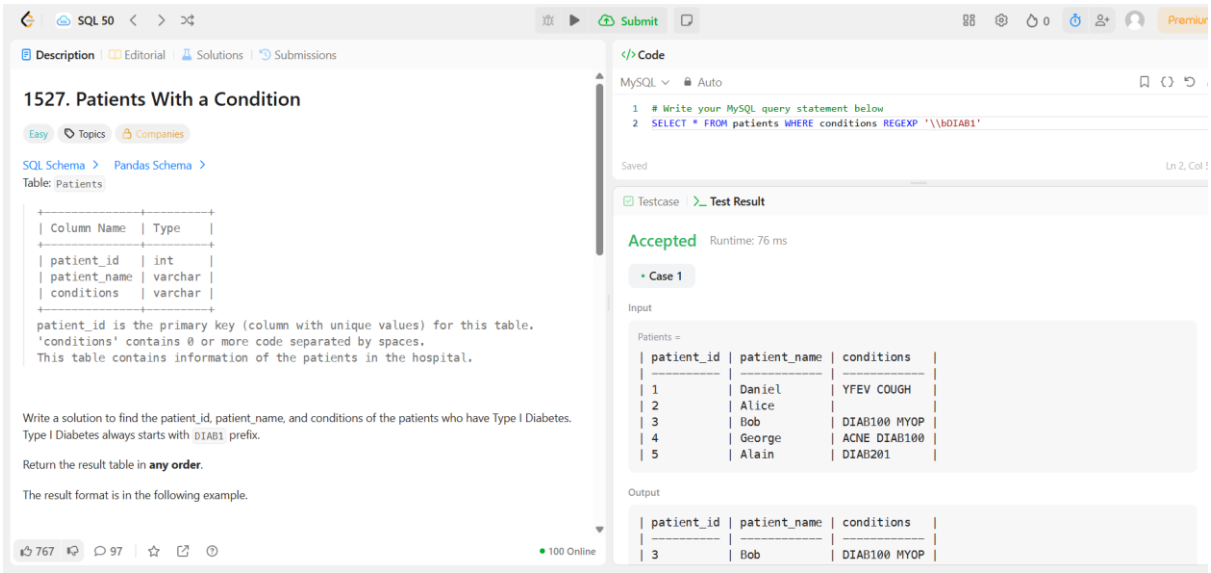


➢ Advanced String Functions / Regex / Clause

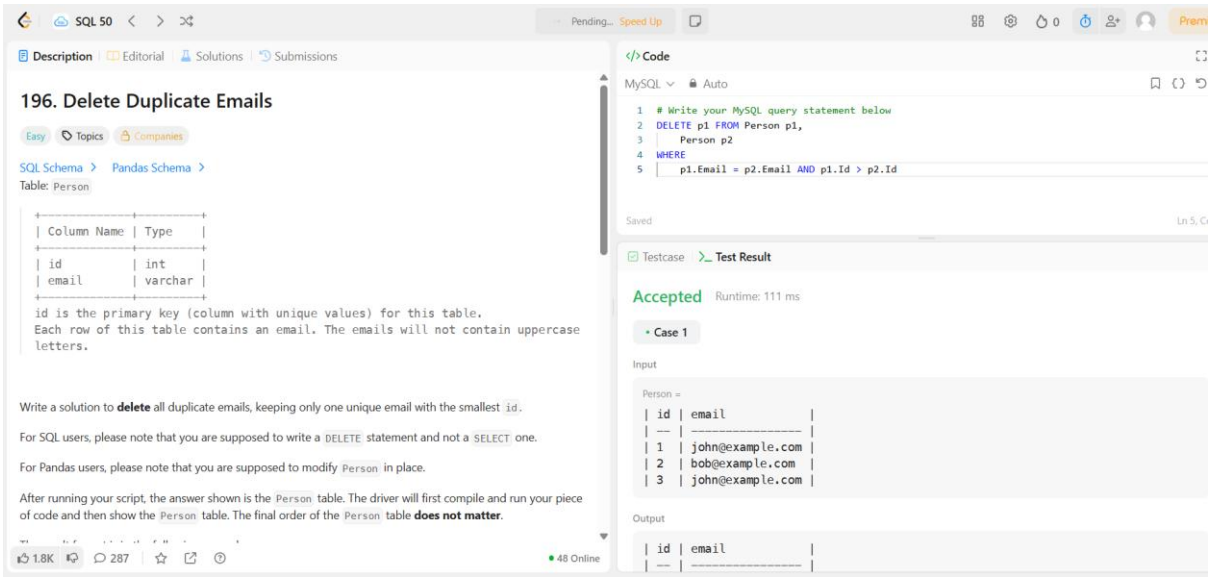
44) 1667. Fix Names in a Table



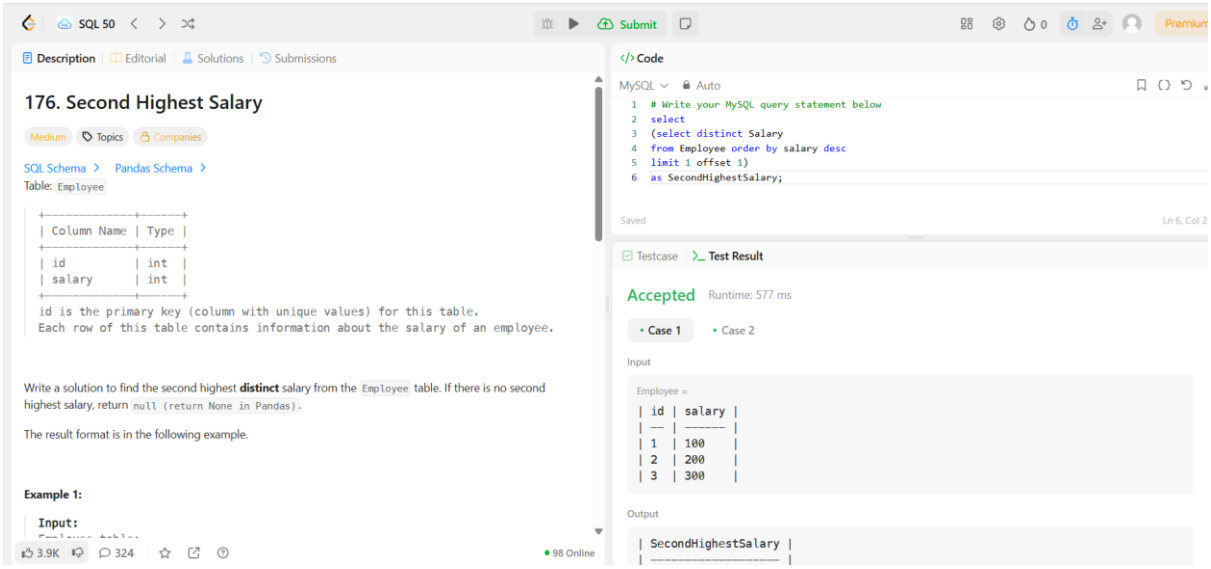
45) [1527. Patients With a Condition]



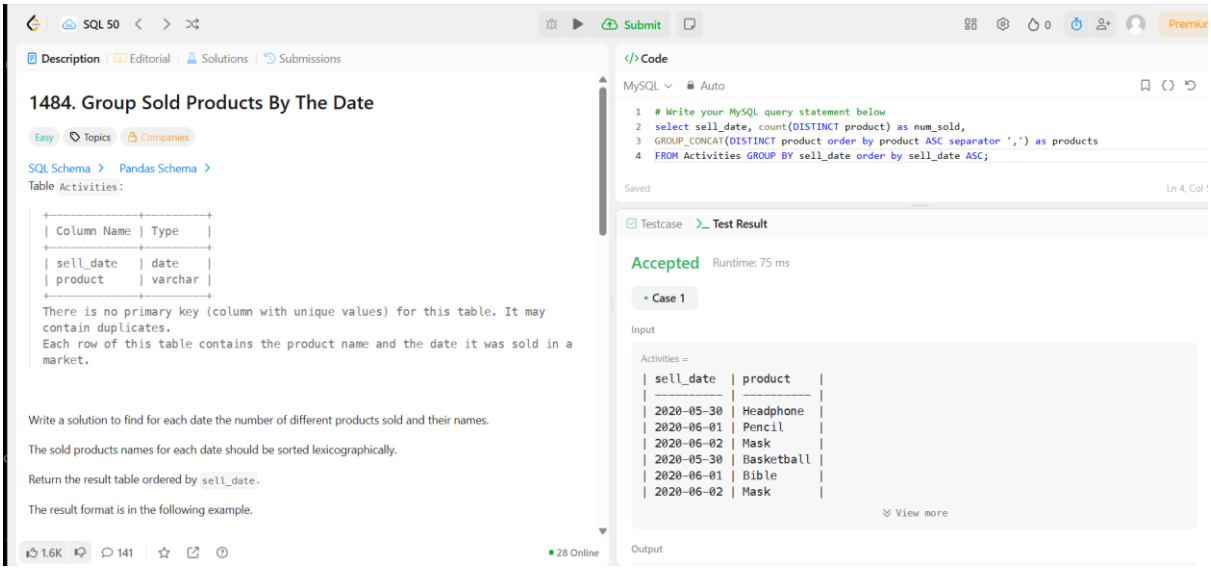
46) [196. Delete Duplicate Emails]



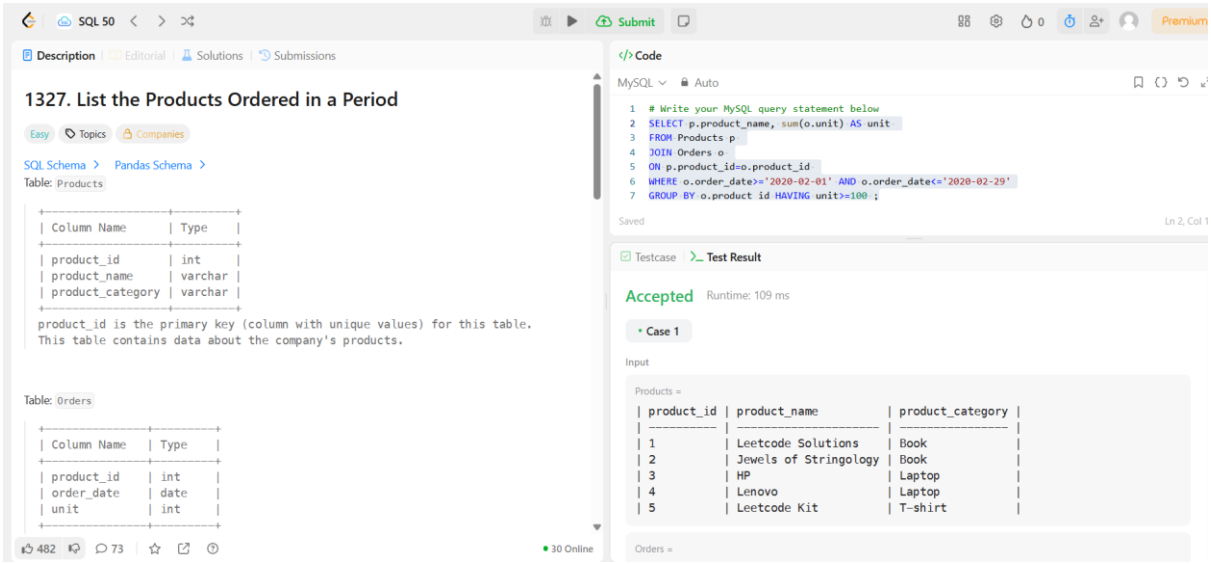
47) [176. Second Highest Salary]



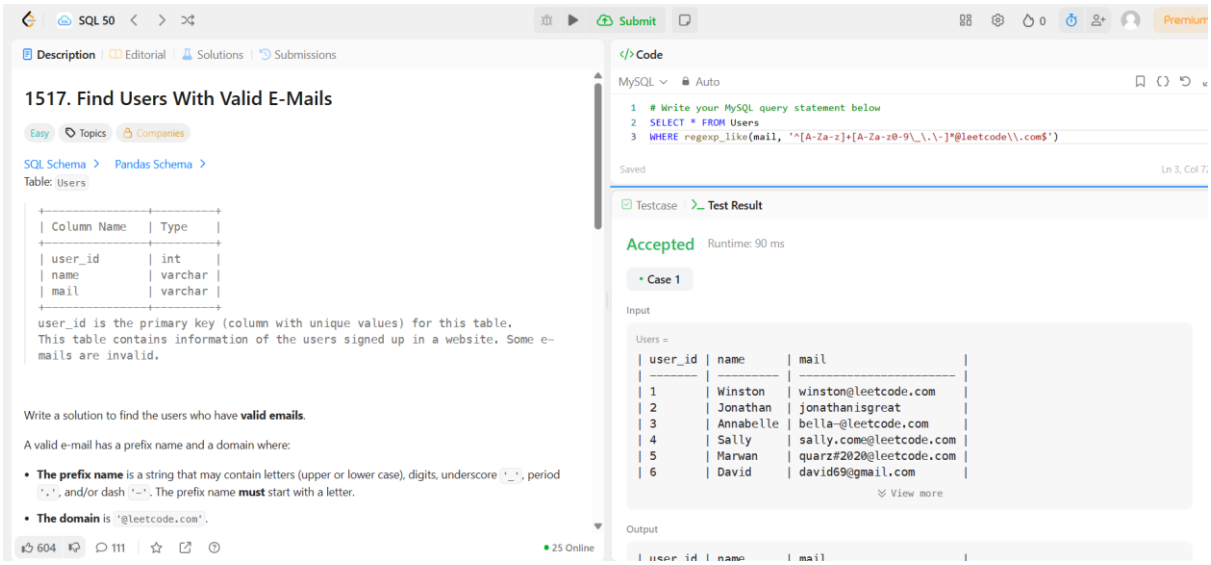
48) [1484. Group Sold Products By The Date]



49) [1327. List the Products Ordered in a Period]



50) [1517. Find Users With Valid E-Mails]


SQL 50
Submit
Premium
Description
Editorial
Solutions
Submissions
1517. Find Users With Valid E-Mails
Easy
Topics
Companies
SQL Schema
Pandas Schema
Table: Users
+----------------+---------+
| Column Name | Type |
+----------------+---------+
user_id	int
name	varchar
mail	varchar
+----------------+---------+
user_id is the primary key (column with unique values) for this table.
This table contains information of the users signed up in a website. Some e-mails are invalid.
Write a solution to find the users who have valid emails.
A valid e-mail has a prefix name and a domain where:
The prefix name is a string that may contain letters (upper or lower case), digits, underscore '_', period '.', and/or dash '-'. The prefix name must start with a letter.
The domain is '@leetcode.com'.
604
111
25 Online
Code
MySQL
Auto
1 # Write your MySQL query statement below
2 SELECT * FROM Users
3 WHERE regexp_like(mail, '^[A-Za-z]+[A-Za-z0-9_\.\-]*@leetcode\\.com$')
Saved
Ln 3, Col 7
Testcase
Test Result
Accepted
Runtime: 90 ms
Case 1
Input
Users =
user_id	name	mail
1	Winston	winston@leetcode.com
2	Jonathan	jonathanisgreat
3	Annabelle	bella-@leetcode.com
4	Sally	sally.come@leetcode.com
5	Marwan	quarz#2020@leetcode.com
6	David	david69@gmail.com
View more
Output