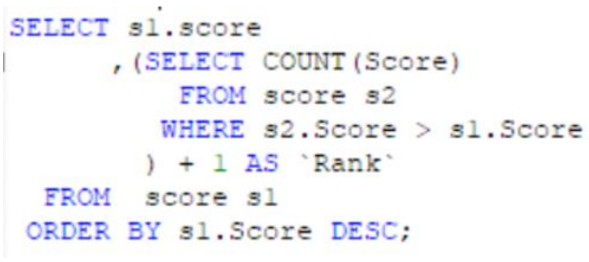
MySQL任务4作业参考答案

项目十六 分数排名 不连续

和项目九的类似，有个小改动。给个眼神，自己体会下。



```
SELECT s1.score
       ,(SELECT COUNT(Score)
          FROM score s2
         WHERE s2.Score > s1.Score
        ) + 1 AS `Rank`
  FROM  score s1
 ORDER BY s1.Score DESC;
```

项目十七 查询回答率最高的问题 （难度：中等）

根据question_id分组，然后根据回答率降序排序，并输出第一条记录。回答率就是*action*字段中’answer’的次数除以’show’的次数。要计算回答率,就需要统计action中’answer’和’show’的个数。可以根据question_id分组，然后分别统计出每道题’answer’和’show’的次数。
首先，按question_id分组，然后用SUM()和IF()统计每道题’answer’和’show’的次数：

```
SELECT question_id,
       SUM(IF(action=’show’, 1, 0)) AS num_show,
       SUM(IF(action=’answer’, 1, 0)) AS num_answer
FROM survey_log GROUP BY question_id
```

再按question_id分组，并根据回答率降序排序：

```
SELECT question_id AS survey_log FROM
(SELECT question_id,
        SUM(IF(action=’show’, 1, 0)) AS num_show,
        SUM(IF(action=’answer’, 1, 0)) AS num_answer
 FROM survey_log GROUP BY question_id) AS t
ORDER BY (num_answer/num_show) DESC;
```

最后输出第一条记录：

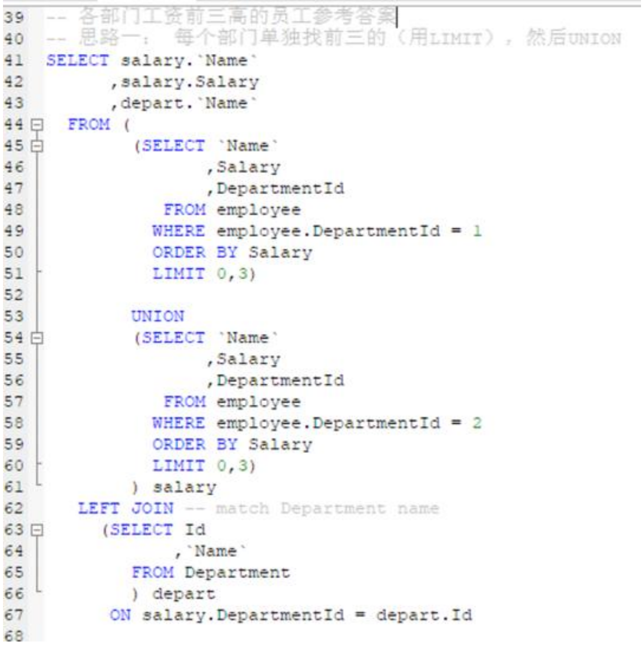
```
SELECT question_id AS survey_log FROM
(SELECT question_id,
        SUM(IF(action=’show’, 1, 0)) AS num_show,
        SUM(IF(action=’answer’, 1, 0)) AS num_answer
 FROM survey_log GROUP BY question_id) AS t
ORDER BY (num_answer/num_show) DESC LIMIT 1;
```

项目十八 各部门工资第三高的员工

思路一： 因为只有两个部门，我们可以取巧分别对每个部门按工资降序排名，取前三行，然后UNION。
需要注意的是，ORDER BY 和 LIMIT本身不支持在子查询中使用。所以需要加上括号形成独立的几个表而不是UNION的子查询。

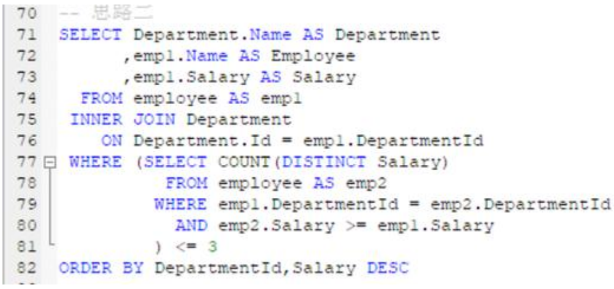
思路二：也是大家普遍在网上搜到的答案。
和分数排名的思想类似，用到了辅助表。
emp1是我们的基础表，emp2是辅助表。
将emp1里的每个salary和整张emp2比较。下面来捋下过程。
以IT部门为例，emp1的salary有 6.9万，7万，8.5万，9万四个数
① emp1工资是6.9万的时候，emp2表里的 count是3，说明有三个大于它的（间接说明它是第四大）
② emp1工资是7万的时候，emp2表里的count是2，说明有两个大于它的（间接说明它是第三大）
③emp1工资是8.5万的时候，emp2表里的count是1，说明有1个大于它的（间接说明它是第二大）
④emp1工资是9万的时候，emp2表里的count是0，说明没有大于它的（间接说明它是最大的）
在code里就是emp2.Salary > （emp1.Salary =6.9）

然后我们要求的是前三大，所以是COUNT() < 3。



```
-- 各部门工资前三高的员工参考答案
-- 思路一：  每个部门单独找前三的（用LIMIT），然后UNION
SELECT salary.`Name`
      ,salary.Salary
      ,depart.`Name`
  FROM (
        (SELECT `Name`
               ,Salary
               ,DepartmentId
           FROM employee
          WHERE employee.DepartmentId = 1
          ORDER BY Salary
          LIMIT 0,3)

        UNION
        (SELECT `Name`
               ,Salary
               ,DepartmentId
           FROM employee
          WHERE employee.DepartmentId = 2
          ORDER BY Salary
          LIMIT 0,3)
        ) salary
   LEFT JOIN -- match Department name
      (SELECT Id
             ,`Name`
         FROM Department
        ) depart
     ON salary.DepartmentId = depart.Id
```



```
-- 思路二
SELECT Department.Name AS Department
      ,emp1.Name AS Employee
      ,emp1.Salary AS Salary
  FROM employee AS emp1
 INNER JOIN Department
    ON Department.Id = emp1.DepartmentId
 WHERE (SELECT COUNT(DISTINCT Salary)
          FROM employee AS emp2
         WHERE emp1.DepartmentId = emp2.DepartmentId
           AND emp2.Salary >= emp1.Salary
         ) <= 3
ORDER BY DepartmentId,Salary DESC
```

项目十九 平面上最近距离

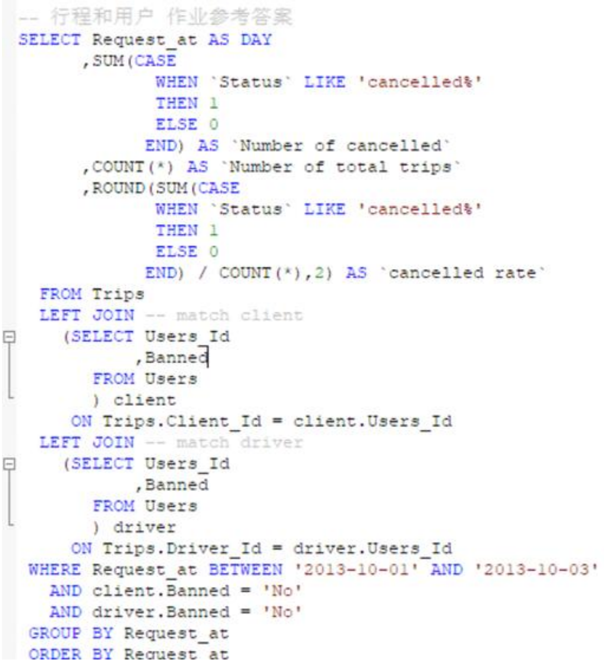
连接（join）两张 **point_2d** 表生成所有可能的点对，然后计算距离，求最小距离并保留小数点后2位。

```
SELECT ROUND(SQRT(MIN(POW(p1.x–p2.x,2)+POW(p1.y–p2.y,2))),2) AS
shortest
FROM point_2d AS p1, point_2d AS p2
WHERE p1.x <> p2.x OR p1.y <> p2.y;
```

项目二十 行程和用户

题目意思本身很简单，难点在于如何同时计算取消的数量和总数量。
总数量很简单，就是COUNT(*)，
同时计算取消数量可以用SUM(CASE END)来实现。对status列计算数量，如果是cancel就记1，complete记为0.

P.S. 如果需要userID 匹配 clientID和driverID两列，需要写两个Lef Join。 但是可以通过CASE ...END 在最终结果只显示一列。



```
-- 行程和用户 作业参考答案
SELECT Request_at AS DAY
      ,SUM(CASE
             WHEN `Status` LIKE 'cancelled%'
             THEN 1
             ELSE 0
            END) AS `Number of cancelled`
      ,COUNT(*) AS `Number of total trips`
      ,ROUND(SUM(CASE
             WHEN `Status` LIKE 'cancelled%'
             THEN 1
             ELSE 0
            END) / COUNT(*),2) AS `cancelled rate`
  FROM Trips
  LEFT JOIN -- match client
    (SELECT Users_Id
           ,Banned
       FROM Users
      ) client
     ON Trips.Client_Id = client.Users_Id
  LEFT JOIN -- match driver
    (SELECT Users_Id
           ,Banned
       FROM Users
      ) driver
     ON Trips.Driver_Id = driver.Users_Id
 WHERE Request_at BETWEEN '2013-10-01' AND '2013-10-03'
   AND client.Banned = 'No'
   AND driver.Banned = 'No'
 GROUP BY Request_at
 ORDER BY Request_at
```