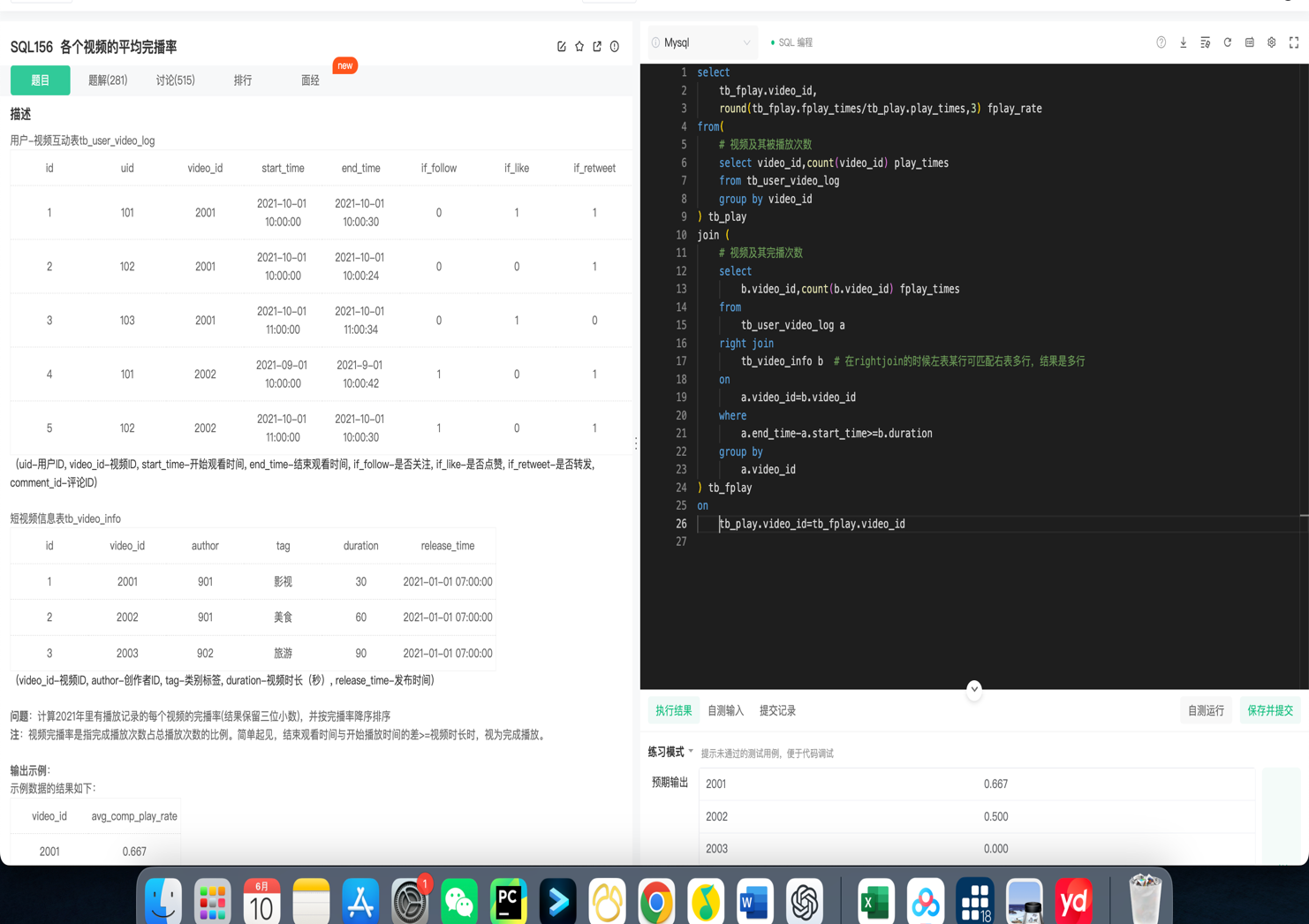
# 1.计算视频完播率



表格

中度可信度描述已自动生成

文本, 信件, 电子邮件

描述已自动生成

# 2. 在北京市接单至少3次的司机的平均接单数和平均兼职收入

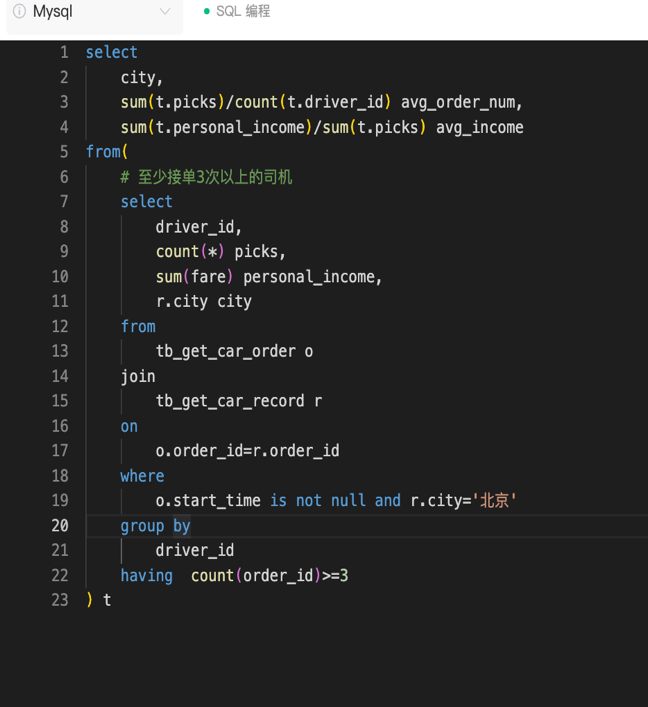
图形用户界面, 表格

中度可信度描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

我自己的答案vs正确答案：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成文本

描述已自动生成

DATE\_FORMAT() 函数的基本语法如下：

DATE\_FORMAT(date, format)

• date：要格式化的日期或日期时间值。

• format：指定输出格式的字符串。这个格式字符串可以包含日期格式化符号，用来指定如何显示日期的各个部分。

* %Y：四位年份

• %y：两位年份

• %m：月份（01到12）

• %d：月中的天（01到31）

• %H：小时（24小时制，00到23）

• %h：小时（12小时制，01到12）

• %i：分钟（00到59）

• %s：秒（00到59）

• %p：AM或PM

假设你有一个日期 "2023-06-15"，并希望以不同的格式显示这个日期：

SELECT DATE\_FORMAT('2023-06-15', '%Y-%m-%d') AS formatted\_date;

输出：2023-06-15

# 3. 有取消订单记录的司机平均评分

数据库表同上

图形用户界面, 文本, 应用程序

描述已自动生成

我的答案和结果如下就是先找出有取消订单的司机再去查评分，其存在的问题是没有汇总信息

图形用户界面, 文本, 网站

描述已自动生成

使用带有 ROLLUP 的 GROUP BY 子句来生成一个包含汇总行的结果集：

ROLLUP 与任何聚合函数（包括 AVG）一起使用时，都会在分组层次上自动生成汇总行。

本题中就是会在司机id层次上生成一个汇总

但是汇总行的主键是null，也就是说本体中汇总行的driver\_id值为null,举个例子如下所示

图形用户界面, 文本, 应用程序

描述已自动生成

所以需要搭配ifnull函数来使用，

IFNULL 函数用于检查一个表达式是否为 NULL，如果是 NULL，则返回一个指定的替代值，否则返回该表达式的值。

图形用户界面, 应用程序

描述已自动生成

所以本题目最终的正确答案是（在原基础上加个rollup和ifnull）：  
文本

描述已自动生成

# 4.统计每个城市中评分最高的司机的聚合信息



这道题的关键就是，既要分城市，还要算司机的聚合信息，所以这道题的关键就是group by后面既要有city又要有driver\_id,分好组之后，就根据要求写聚合函数或者窗口函数即可

正确答案如下

文本

描述已自动生成

**6.国庆期间前3日近7日日均取消订单量**

图形用户界面, 文本, 表格

中度可信度描述已自动生成

分析：

1. 只要订单表有记录，那就是有订单，所以只用看订单表，所以不用管打车记录表的所有字段
2. 订单只有两个状态：完成or取消；（判断依据：有start time就是接上了，就算“完成单”）
3. 这类题的模板：先求每一天（也就是某一天的），用窗口函数order by去圈定一个窗口，在这个窗口里做聚合，可以分组也可以不分组

文本

描述已自动生成

ROWS 6 PRECEDING 是窗口函数中的一个子句，用于定义窗口的范围。在这个查询中，它指定了窗口包含当前行及其之前的6行，共7行。

文本

描述已自动生成

# 5.每篇文章同一时刻最大在看人数

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

本题的难点在于：如何求取或计算计算瞬时的最大计数（在看人数）

这类取数问题的解法就是对原表进行编码+联立

所以，在此对原表in\_time和out\_time进行编码，in为观看人数+1， out为观看人数-1，进行两次SELECT联立（这里对应的代码要认真看，编码的本质就是新增一列再筛选出需要的内容，而且因为需要计算瞬时变化，这里需要用union all得到两个查询的并集而不是用union得到去重后的结果），并按artical\_id升序，时间戳升序，此时的结果如下：

图形用户界面, 应用程序

中度可信度描述已自动生成

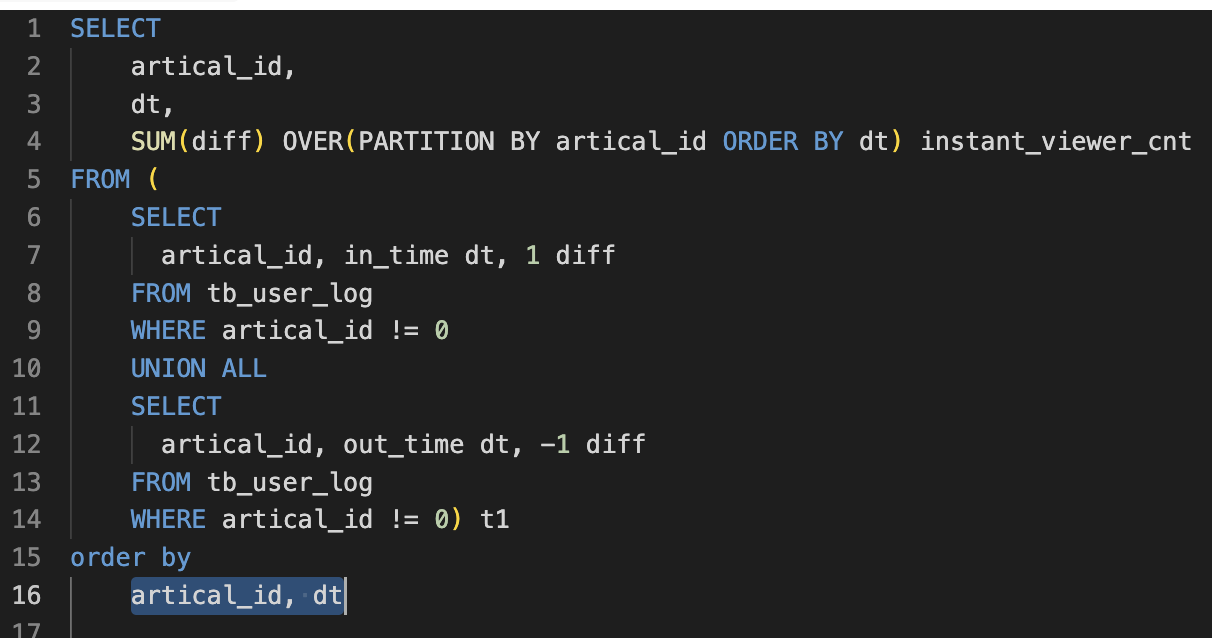
此时我们得到的是，某个内容在给定的时间戳下的瞬时变化动作，有了瞬时的变化动作之后，就可以通过聚合的方式得到某个瞬间具体的观看人数，所以这里使用sum窗口函数来处理

所以，窗口内按文章id分区，按时间顺序排序

文本, 表格

描述已自动生成

这里注意窗口函数的意义，over中无论是partition by还是order by 都是在定义窗口函数sum的计算方式而不是对外部查询的group by和order by产生影响，这里有一个小问题，既然窗口函数只是定义计算方式，为什么输出结果是按照artical\_id, dt进行排序的呢，这可能是因为某些数据库系统（如PostgreSQL和MySQL）在处理 UNION ALL 后的结果集时，会保持输入结果的顺序，特别是在 ORDER BY 子句明确指定时。所以最好还是显式地将artical\_id, dt的排序写出来



然后到了本题的坑点。

题目要求在瞬时统计时遵循【先进后出】：如果同一时刻有进入也有离开时，先记录用户数增加，再记录减少。

因此在ORDER BY层面，在遵循dt升序的同时，还要遵循先+1，再-1的原则，即diff DESC：

SUM(diff) OVER(PARTITION BY artical\_id ORDER BY dt, diff DESC)

最后再把这个搞成子查询，分组聚合找最大即可：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（这里没有修正刚才说的排序）

# 6.一月内每天的新用户次日留存率



1.先查询出每个用户第一次登陆时间（最小登陆时间）--每天新用户表

2.因为涉及到跨天活跃，所以要进行并集操作，将登录时间和登出时间取并集，这里union会去重，也就是说用户的登入和登出各自算，都在一张表中，这样才能满足夸天的要求--用户活跃表

3.将每天新用户表和用户活跃表左连接，只有是同一用户并且该用户第2天依旧登陆才会保留整个记录，否则右表记录为空

4.得到每天新用户第二天是否登陆表后,开始计算每天的次日留存率：根据日期分组计算，次日活跃用户个数/当天新用户个数

文本

描述已自动生成

用with ……as的时候注意 ，如果连续定义两个临时表（with as）那么中间需要用逗号隔开且第二个临时表不需要再用with

文本

描述已自动生成

最后左连接联表查询即可

注意函数date\_sub date\_add的用法