MYSQL开发技巧 3

如何在子查询中匹配俩个值 解决同属性多值过滤的问题 如果计算累计税的问题

常见的子查询使用场景

1 使用子查询可以避免由于子查询中的数据产生的重复 in的作用是

select *

from test
where id in (1,2,3)
就是查询id=1 or id=2 or id=3的意思
也有
select *
from test1
where id in (select id from test2)
这样的,意思是一样的

2 使用子查询的语句更好理解, 更符合语句

特殊情况: 取出多列的场景 多列过滤

MySQL中独有的多例过滤方式

SELECT a.user_name,b.timestr,kills
FROM user1 a
JOIN user_kills b ON a.id =b.user_id
WHERE (b.user_id,b.kills) IN (
SELECT user_id,MAX(kills)
FROM user_kills
GROUP BY user_id)

select a,user_name,b.timestr,kills from user1 a join user_kills b on a.id=b.user_id join (select user_id,max (kills) as cnt from user_kills group by user_id) c on b.use id = c.user id and b.kills = c.cnt;

二如何解决同属性多值过滤问题 什么是同一属性的多值过滤 为了解释在我们示例库中添加一个技能表

user_name	skill	skill_level 5	
唐僧	紧箍咒		
唐僧	打坐	4	
唐僧	念经	5	
唐僧	变化	0	
猪八戒	变化	4	
猪八戒	腾云	3	
猪八戒	浮水	5	
猪八戒	念经	0	
猪八戒	紧箍咒	0	
孙悟空	变化	5	
孙悟空	腾云	5	
孙悟空	浮水	3	
孙悟空	念经	2	
孙悟空	请神	5	
孙悟空	紧箍咒	0	
沙僧	变化	2	
沙僧	腾云	2	
沙僧	浮水	4	
沙僧	念经	1	

如何查询出同时具有变化和念经的俩项技能的取经人

select a.user_name,b.skill,c.skill from user1 a join user1_skills b on a.id=b.user_id and b.skill="念经" join user_skills c on c.user_id=b.user_id and c.skills="变化" where b.skills_level>0 and c.skills_level>0;

select a.user_name,b.skill,c.skill,d.skill from user1 a join user1_skill b on a.id =b.user_id and b.skill='念经' join

日常工作中也要注意到,只需要查询到我们需要的值

这种方法

查询出技能的大于等于2的取经人

SELECT a.'user_name',b.skill,c.skill,d.skill,e.skill-

FROM user1 a +

LEFT JOIN user1_skills b ON a.'id'=b.user_id AND b.skill='念经' AND

b.skill_level>0-

LEFT JOIN user1_skills c ON a.id= c.user_id AND c.skill ='变化' AND

c.skill_level>0+

LEFT JOIN user1_skills d ON a.id = d.user_id AND d.skill= '騰云' AND

d.skill level>0

LEFT JOIN user1_skills e ON a.id = e.user_id AND e.skill='浮水' AND

e.skill_level>0

Where (case when b.skill is not null then 1 else 0 end).

+(case when c.skill is not null then 1 else 0 end)-

+(case when d.skill is not null then 1 else 0 end)-

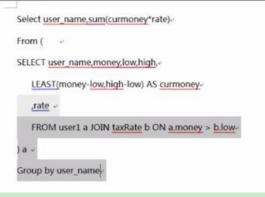
+(case when e.skill is not null then 1 else 0 end)

这种方法比较简单

select a.user_name from user1 a join user1_skils b on a.id=b_user_id where b.skil in('念经','变化','腾云','浮水') and b.skils >0, group by a.user name having count(*) >=2;

利用LEAST()函数来确定每个区间的纳税额

user_name	money	low	high	curmoney	rate
唐僧	35000.00	0	1500	1500	0.03
唐僧	35000.00	1500	4500	3000	0.1
唐僧	35000.00	4500	9000	4500	0.2
唐僧	35000.00	9000	35000	26000	0.25
孙悟空	28000.00	0	1500	1500	0.03
孙悟空	28000.00	1500	4500	3000	0.1
孙悟空	28000.00	4500	9000	4500	0.2



Least()可以返回参数中最小的值