

能做对这几道SQL题，你的面试可能就稳了

OriginalBitTiger BitTiger 1周前



SQL在Business Analyst, Data Analyst, 甚至Data Scientist面试中都占据了非常重要的部分。大部分人都会觉得SQL很简单，但一到面试就挂。在这里我们列举几个SQL题目，熟练掌握这些题，你的面试基本不会被难倒。

这里有两张表：

HR_ID	Name	Hire_Date	Title
121212	Mingche	10/28/2015	Analyst
131415	小明	08/01/2012	Engineer
...

Table a: HR Hires Table (记录员工入职时的信息)

HR_ID	Promotion_Date	Title
121212	09/21/2016	Sr. Analyst
...

Table b: Promotions Table (记录员工的Promotion和新的Title)

热身问题：小王，你来说一下哪些员工从来没有升过职？

要求是：在以下两种不同条件下，分别解决这个问题。

条件一：不能Join table

条件二：必须Join table

大家先自己想一下。

防偷看

防偷看

防偷看

防偷看

防偷看

防偷看

以下为答案：

1) 不能Join table

```
Select * from a where a.HR_ID not in (select distinct HR_ID from b)
```

2) 必须Join table

```
Select
```

```
a.*
```

```
from a
```

```
left join b
```

```
on a.HR_ID = b.HR_ID
```

```
Where b.HR_ID is null
```

这道题考察了Left join的使用。

我们进一步思考，如果最后一行的Where b.HR_ID is null 中，“Is null”改成“is not null”，是不是就能得到有过Promotion的员工呢？

答案是肯定的，但并不全对，因为会得到所有重复的员工信息，所以写这段代码时不要忘了Distinct。

```
Select
```

distinct

a.*

from a

left join b

on a.HR_ID = b.HR_ID

Where b.HR_ID is null

在做SQL题时，一定要注意Distinct，Distinct可以帮助去重，是一个需要注意的重要细节，是一个习惯。宁可多写也不要被面试官抓住把柄。

我们再看下一个问题：

HR_ID	Name	Hire_Date	Title
121212	Mingche	10/28/2015	Analyst
131415	小明	08/01/2012	Engineer
...

Table a

Promotion_ID	HR_ID	Promotion_Date	Title
1002	121212	09/21/2016	Sr. Analyst
1025

Table b

这里我们给Table b加了一列作为Primary key。

小王，你来说一下：哪位员工升职次数最多？

Select b.HR_ID, count(distinct Promotion_ID)

From b

Group by 1

Order by 2 desc

Limit 1

还可以这样写：

Table c:

```
Select b.HR_ID, count(distinct Promotion_ID) as promo_cnt
```

```
From b
```

```
Group by 1
```

```
Order by 2 desc
```

建一个Table c，每一行表示这个人升职了多少次。通过Table c找出谁升职次数最多：

```
Select distinct c.HR_ID
```

```
From c
```

```
Where c.promo_cnt = (select max(promo_cnt) from c)
```

那么，哪位员工升职次数第二多？

```
Select distinct c.HR_ID, c.promo_cnt
```

```
From c
```

```
Where c.promo_cnt <> (select max(promo_cnt) from c)
```

```
Order by 2 desc
```

```
Limit 1
```

哪位员工升职次数第五多？

HR_ID	Name	Hire_Date	Title
121212	Mingche	10/28/2015	Analyst
131415	小明	08/01/2012	Engineer
...

Promotion_ID	HR_ID	Promotion_Date	Title
1002	121212	09/21/2016	Sr. Analyst
1025

在这里推荐大家使用Rank函数，写出来简单易懂。你会用Rank，很多难题会迎刃而解。如果你在解决“第二多”的问题时就直接用Rank，面试官可能就会觉得你的SQL技能在一个比较好的水平。

```
Rank() over(partition by XXX order by XXX)
```

Rank函数是给元素排序，不需要用Group by，就能给每个元素一个index。需要注意的是，如果用Rank，Query中要用Qualify而不是Where，

```
Select distinct c.HR_ID, c.promo_cnt,
```

```
Rank() over(order by c.promo_cnt desc) as rnk
```

```
From c
```

```
Qualify rnk = 5
```

现在，如果不允许你用Raw number，Limit，以及Rank，怎么找出升职次数第五多的员工呢？

这考察的是Self join，代码如下：

```
Select c1.HR_ID, Count(distinct c2.promo_cnt) as rnk
```

```
From c c1 left join c c2
```

```
On c1.promo_cnt <= c2.promo_cnt
```

```
Group by 1
```

```
Having rnk = 5
```

用这种方法，找第n个人也是可以实现的。

最后再看一个拓展问题：

如果Table里有重复值，你如何解决重复值？

比如现在有100个人，升职次数有两个人并列第一，他们都升职了10次。而我只想要Top 1的这一个值，这时Self join可能没法解决这个问题，但用Rank可以解决。

```
Select distinct c.HR_ID, c.promo_cnt,
```

```
Rank() over(order by c.promo_cnt, c.hr_id desc) as rnk
```

```
From c
```

```
Qualify rnk = 5
```

这样每次只会取出一条记录。

最后给大家留一个思考题：

哪一种Promotion最耗时间？（比如从Analyst升到Sr. Analyst，为一种Promotion）

以上总结起来有两个要点：

- 熟练掌握Rank以应对测试
- 如果不用Rank，知道使用Cross join来完成排序

为你详细讲解十二个常见算法问题，不仅让你学会如何掌握这些问题的解题代码，还会让你学到解决算法问题所必需的四类解题模板，让你轻松应对各类新问题。

↓↓↓

0 to 1 高频算法面试题

金牌讲师：冯沁原

BitTiger.io 联合创始人、算法与系统设计主讲老师与教学总监。前阿里巴巴高级专家、前酷我音乐总监，北京大学博士、微软亚洲学者。



原价：\$129

限时\$98

关注我们



后台回复“**我要进群**”
加入BitTiger讨论群



<https://www.bittiger.io>



YouTube



Read more