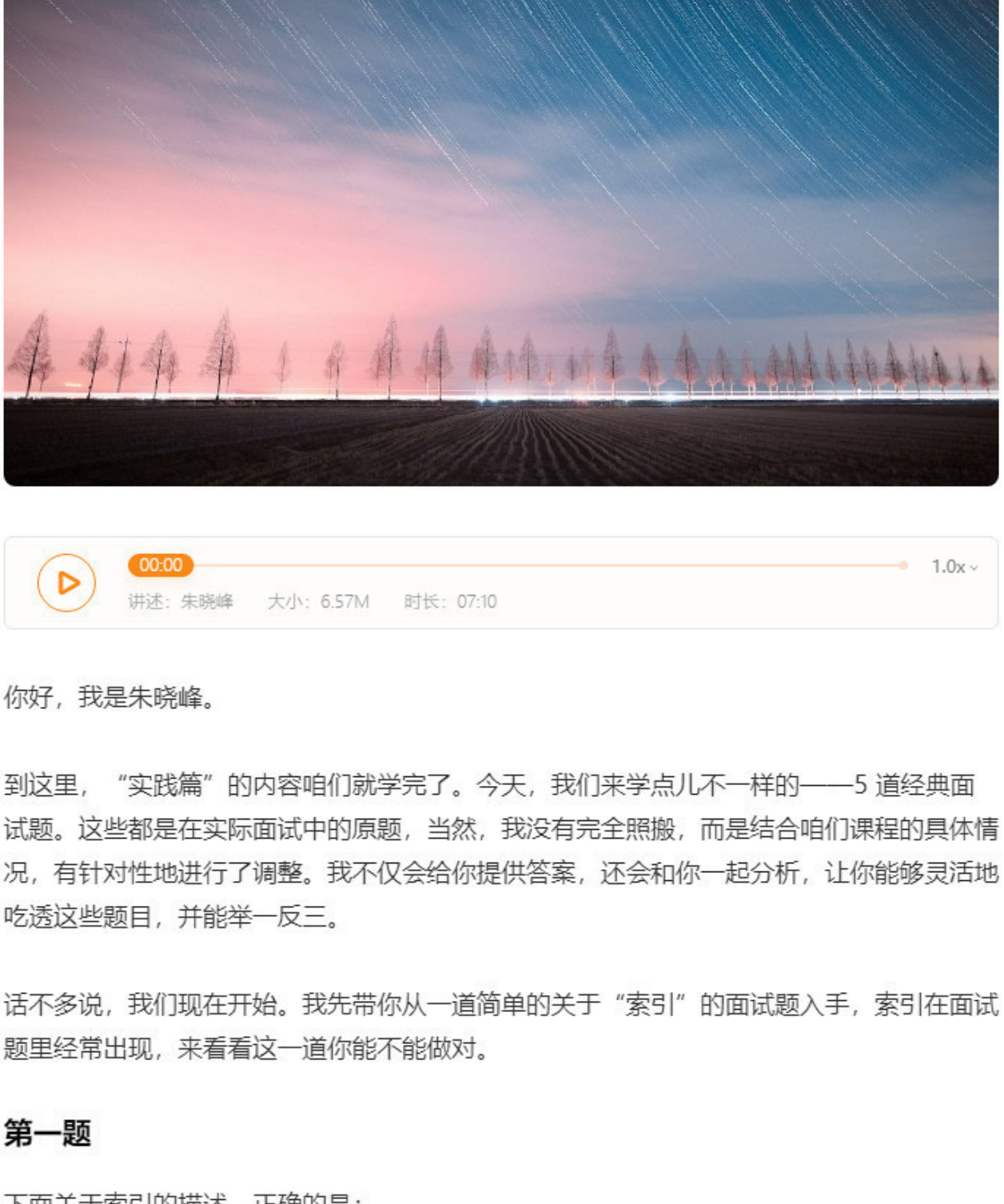


特别发送（一） | 经典面试题讲解第一弹

朱晓峰 2021-04-08



00:00 讲话: 朱晓峰 大小: 6.57M 时长: 07:10 1.0x

你好，我是朱晓峰。

到这里，“实践篇”的内容咱们就学完了。今天，我们来学点儿不一样的——5 道经典面试题。这些都是在实际面试中的原题，当然，我没有完全照搬，而是结合咱们课程的具体情况，有针对性地进行了调整。你不仅会给你提供答案，还会和你一起分析，让你能够灵活地吃透这些题目，并能举一反三。

话不多说，我们现在开始。我先带你从一道简单的关于“索引”的面试题入手，索引在面试题里经常出现，来看看这一道题你能不能做对。

第一题

下面关于索引的描述，正确的是：

1. 建立索引的主要目的是减少冗余数据，使数据表占用更少的空间，并且提高查询的速度
2. 一个表上可以建立一个或者多个索引
3. 组合索引可以有效提高查询效率，比单字段索引更高效，所以，我们应该创建一个由所有的字段组成的组合索引，这样就可以解决所有问题了
4. 因为索引可以提高查询效率，所以索引建得越多越好

解析：这道题的正确答案是选项 2，我们来分析一下其他选项。

- 选项 1 说对了一半，索引可以提高查询效率，但是创建索引不能减少冗余数据，而且索引还要占用额外的存储空间，所以选项 1 不对。
- 选项 3 不对的原因有 2 个。第一，组合索引不一定比单字段索引高效，因为组合索引的字段是有序的，遵循左对齐的原则。如果查询的筛选条件不包含组合索引中最左边的字段，那么组合索引就完全不能用。第二，创建索引也是有成本的，需要占用额外的存储空间。用所有的字段创建组合索引的存储成本比较高，而且利用率比较低，完全上的可能性几乎不存在，所以很少有人会这样做。而且一旦更改任何一个字段的数据，就必须修改索引，这样操作成本也比较高。
- 选项 4 错误，因为索引有成本，很少作为筛选条件的字段，没有必要创建索引。

如果这道题你回答错了，一定要回去复习下[第 11 讲](#)的内容。

第二题

假设我们有这样一份学生成绩单，所有同学的成绩都各不相同，请编写一个简单的 SQL 语句，查询分数排在第三名的同学的成绩：

| id (编号) | Name (名称) | Score (成绩) |
|------------|--------------|---------------|
| 1 | 张三 | 80 |
| 2 | 李四 | 90 |
| 3 | 王五 | 76 |
| 4 | 赵六 | 88 |
| 5 | 孙七 | 67 |

解析：这道题考查的是我们对查询语句的掌握情况。针对题目中的场景，可以分两步来进行查询。

第一步，按照成绩高低进行排序：

```
1 mysql> SELECT *
2 -> FROM demo.person AS a
3 -> ORDER BY score DESC; -- DESC表示降序排列
4
5 | id | name | score |
6 |----|-----|
7 | 2 | 李四 | 90.00 |
8 | 4 | 赵六 | 88.00 |
9 | 1 | 张三 | 80.00 |
10 | 3 | 王五 | 76.00 |
11 | 5 | 孙七 | 67.00 |
12 |----|-----|
13 5 rows in set (0.00 sec)
```

第二步，找出排名第三的同学和对应的成绩。我们可以用[第 4 讲](#)里学过的对返回记录进行限定的关键字 LIMIT：

```
1 mysql> SELECT *
2 -> FROM demo.test1
3 -> ORDER BY score DESC
4 -> LIMIT 2,1;
5
6 | id | name | score |
7 |----|-----|
8 | 1 | 张三 | 80.00 |
9 |----|-----|
10 1 row in set (0.00 sec)
```

在 MySQL 中，LIMIT 后面可以跟 2 个参数，第一个参数表示记录的起始位置（第一个记录的位置是 0），第二个参数表示返回几条记录。因此，“LIMIT 2,1”就表示从第 3 条记录开始，返回 1 条记录。这样，就可以查出排名第三的同学的成绩了。

第三题

现在我们有二个表，分别是人员表（demo.person）和地址表（demo.address），要求你使用关联查询查出完整信息。无论有没有地址信息，人员的信息必须全部包含在结果集中。

人员表：

| Id (编号) | Fname (姓) | Lname (名) |
|------------|--------------|--------------|
| 1 | 张 | 三 |
| 2 | 李 | 四 |
| 3 | 王 | 五 |

地址表：

| Id (编号) | Country (国家) | City (城市) | Address (地址) |
|------------|-----------------|--------------|-----------------|
| 1 | 中国 | 北京 | 海淀区123号 |
| 2 | 美国 | 纽约 | 奥巴马333号 |

解析：这个是典型的外查询，咱们在[第 6 讲](#)里学过。题目要求我们查出人员表中的全部信息，而地址表中信息则可以为空，就可以用下面的查询代码：

```
1 mysql> SELECT *
2 -> FROM demo.person AS a
3 -> LEFT JOIN demo.address AS b -- 左连接，确保demo.person中的记录全部包括在结果集中
4 -> ON (a.id=b.id);
5
6 | id | fname | lname | id | country | city | address |
7 |----|-----|
8 | 1 | 张 | 三 | 1 | 中国 | 北京 | 海淀区123 |
9 | 2 | 李 | 四 | 2 | 美国 | 纽约 | 奥巴马333 |
10 | 3 | 王 | 五 | NULL | NULL | NULL |
11 |----|-----|
12 3 rows in set (0.02 sec)
```

第四题

假设有这样一个教学表（demo.teach），里面包含了人员编号（id）、姓名（fname）和对应的老师的人员编号（teacherid）。如果一个人是学生，那么他一定有对应的老师编号，通过这个编号，就可以找到老师的信息；如果一个人是老师，那么他对应的老师编号就是空。比如说，下表李四的老师编号是 101，我们就可以通过搜索人员编号，找到 101 的名称是张三，那么李四的老师就是张三；而张三自己就是老师，所以他对应的老师编号是空。

| Id (人员编号) | Fname (姓名) | Teacherid (老师编号) |
|--------------|---------------|---------------------|
| 101 | 张三 | NULL |
| 102 | 李四 | 101 |
| 103 | 王五 | 101 |
| 104 | 赵六 | NULL |
| 105 | 孙七 | 104 |
| 106 | 周八 | 101 |

要求：请写一个 SQL 语句，查询出至少有 2 名学生的老师姓名。

说明一下，在刚刚的数据表中，张三有 3 名学生，分别是李四、王五和周八。赵六有一名学

生，是孙七。因此，正确的 SQL 语句的查询结果应该是：

| Id | Fname |
|-----|-------|
| 101 | 张三 |

解析：

针对这道题，我们可以按照这样的思路去做：

1. 通过统计学生对应的老师编号，就可以获取至少有 2 个学生的老师的编号。
2. 通过关联查询和自连接获取需要的信息。所谓的自连接，就是数据表与自身进行连接。你可以认为是把数据表复制成一模一样的 2 个表，通过给数据表起不同的名字来区分它们，这样方便对表进行操作，然后就可以对这 2 个表进行连接操作了。
3. 通过使用条件语句 WHERE 和 HAVING 对数据进行筛选：先用 WHERE 筛选出所有的老师编号，再用 HAVING 筛选出有 2 个以上学生的老师编号。

首先，我们来获取老师编号，如下：

```
1 mysql> SELECT teacherid
2 -> FROM demo.teach
3 -> WHERE teacherid IS NOT NULL
4 -> GROUP BY teacherid
5 -> HAVING COUNT(*) >= 2;
6
7 | teacherid |
8 |----|
9 | 101 |
10 |----|
11 1 row in set (0.00 sec)
```

接着，通过自连接，来获取老师的姓名：

```
1 mysql> SELECT a.id,a.fname
2 -> FROM demo.teach AS a
3 -> JOIN
4 -> {
5 -> SELECT teacherid
6 -> FROM demo.teach
7 -> WHERE teacherid IS NOT NULL
8 -> GROUP BY teacherid
9 -> HAVING COUNT(*) >= 2
10 -> } AS b
11 -> ON (a.id = b.teacherid);
12
13 | id | fname |
14 |----|
15 | 101 | 张三 |
16 |----|
17 1 row in set (0.00 sec)
```

第五题

假设某中学高三年级有多位同学，分成多个班，我们有统一记录学生成绩的表（demo.student）和班级信息表（demo.class），具体信息如下所示：

学生成绩表：

| Id (编号) | Stdname (名称) | Points (成绩) | Classid (班级编号) |
|------------|-----------------|----------------|-------------------|
| 1 | 张三 | 85 | 1 |
| 2 | 李四 | 80 | 2 |
| 3 | 王五 | 65 | 2 |
| 4 | 赵六 | 90 | 1 |
| 5 | 孙七 | 85 | 1 |
| 6 | 周八 | 69 | 1 |
| 7 | 郑九 | 70 | 1 |

班级信息表：

| Id (编号) | Classname (班级名称) |
|------------|---------------------|
| 1 | 创新班 |
| 2 | 普通班 |

要求：写一个 SQL 查询语句，查出每个班级前三名的同学。

说明一下，针对上面的数据，正确的 SQL 查询应该得出下面的结果：

| Classname (班级) | Stdname (名称) | Points (成绩) |
|-------------------|-----------------|----------------|
| 创新班 | 赵六 | 90 |
| 创新班 | 张三 | 85 |
| 创新班 | 周八 | 85 |
| 创新班 | 郑九 | 70 |
| 普通班 | 李四 | 80 |
| 普通班 | 王五 | 65 |

解析：

1. 从题目给出的查询结果看，不需要考虑并列的情况。那么，现在要选出分数排名前三的同学，其实只要找出 3 个最好的分数以及对应的同学就可以了。
2. 这道题需要用到我们讲过的关联查询和子查询的知识。
3. WHERE 语句的筛选条件表达式中，也可以包括一个子查询的结果。

第一步，我们假设有一个分数 X，就是那个第 N 好的分数，算一下有多少个同学的成绩优于这个分数：

```
1 SELECT COUNT(DISTINCT b.points)
2 FROM demo.student AS b
3 WHERE b.points > X;
```

这个查询的结果小于 3 的话，就代表这个分数 X 是排名第三的分数了。

第二步，查询出哪些同学满足成绩排名前三的这个档次：

```
1 mysql> SELECT a.stdname,a.points
2 -> FROM demo.student AS a
3 -> WHERE 3 >
4 -> (
5 -> SELECT COUNT(DISTINCT b.points) -- 统计一下有几个成绩
6 -> FROM demo.student AS b
7 -> WHERE b.points > a.points -- 比这个成绩好
8 -> );
9
10 | stdname | points |
11 |----|
12 | 张三 | 85 |
13 | 李四 | 80 |
14 | 赵六 | 90 |
15 | 周八 | 85 |
16 |----|
17 4 rows in set (0.00 sec)
```

第三步，与班级表关联，按班级统计前三名同学的成绩，并且获取班级信息：

```
1 mysql> SELECT c.classname,a.stdname,a.points
2 -> FROM demo.student AS a
3 -> JOIN demo.class AS c
4 -> ON (a.classid = c.id) -- 关联班级信息
5 -> WHERE 3 >
6 -> (
7 -> SELECT COUNT(DISTINCT b.points)
8 -> FROM demo.student AS b
9 -> WHERE b.points > a.points
10 -> AND b.classid = a.classid -- 按班级分别查询
11 -> );
12 -> ORDER BY c.classname,a.points DESC;
13
14 | classname | stdname | points |
15 |----|
16 | 创新班 | 赵六 | 90 |
17 | 创新班 | 张三 | 85 |
18 | 创新班 | 周八 | 85 |
19 | 创新班 | 郑九 | 70 |
20 | 普通班 | 李四 | 80 |
21 | 普通班 | 王五 | 65 |
22 |----|
23 6 rows in set (0.00 sec)
```

总结

今天，我们借助几个面试题，回顾了索引的概念、查询、子查询和关联查询的知识，以及条件语句 WHERE 和 HAVING 的不同使用方法。所以，在学习每一节课时，你一定要结合我给出的过程中，对知识的灵活运用能力。所以，在学习每一节课时，你一定要结合我给出的实际项目，去真正实践一下，这样才能以不变应万变，在面试中有好的表现。

在真正的面试中，很少有单纯考查知识点本身的题目，更多的是考查你在解决实际问题的过程中，对知识的灵活运用能力。所以，在学习每一节课时，你一定要结合我给出的实际项目，去真正实践一下，这样才能以不变应万变，在面试中有好的表现。

更多学习推荐

175 道 Go 工程师大厂常考面试题

限量免费领取

极客大学

版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

“内部学习使用” 699250

精选留言 (6)

Jun

如果mysql版本大于8.0，第二题和第五题都可以用row_number来解决，更加简洁和容易理解。

2021-05-05

右耳猫咪咪

第三题貌似有问题，不应该根据两个表的主键关联，应该在地址表创建一个人员表的外键，这个外键与人员表的主键关联才对。

2021-04-08

freshswq

之前@June第二题和第五题都可以使用mysql8 新功能窗口函数，试了一下，比较好理解。相关SQL代码如下（含建表SQL）

第二题：

建表和数据SQL:

```
create table if exists test1;
create table test1
(id int primary key,
Name varchar(20),
Score decimal(4,2))
;
```

```
Insert into test1 values(1,'张三',80);
Insert into test1 values(2,'李四',90);
Insert into test1 values(3,'王五',76);
insert into test1 values(4,'赵六',88);
insert into test1 values(5,'孙七',67);
```

查询SQL:

```
select * from (
SELECT
FROM
( SELECT rank() over ( ORDER BY Score DESC ) AS row_num,Id, NAME, Score FROM test1 ) t
WHERE
row_num = 3)
```

第五题：

```
drop table if exists student;
create table student
(
Id int primary key,
Stdname varchar(20),
Points decimal(4,2),
Classid tinyint
)
```

```
insert into student values(1,'张三',85,1);
insert into student values(2,'李四',80,2);
insert into student values(3,'王五',65,2);
insert into student values(4,'赵六',90,1);
insert into student values(5,'孙七',69,1);
insert into student values(6,'周八',85,1);
insert into student values(7,'郑九',70,1);
```

```
drop table if exists class;
create table class
(
Classid tinyint primary key,
Classname varchar(20))
;
```

```
insert into class values(1,'创新班');
insert into class values(2,'普通班');
```

查询SQL:

```
SELECT
Classname,
Stdname,
Points
FROM
(
SELECT
dense_rank() over ( PARTITION BY Classname ORDER BY Points DESC ) AS row_num,
Classname,
Stdname,
Points
FROM
student a
JOIN class b
WHERE
a.Classid = b.id
)t
WHERE
row_num <= 3
```

2021-05-12

Jun

老师，请问能否提供创建这些面试题里面的表的代码？

2021-05-05

lesserror

偶尔做SQL题目，和做算法题目一样，都可以预防脑退化。

这一讲中，第五题是一道好题，想了一会儿才想明白。要是面试让我写，不一定能写出来，可能面试官直接会看第五题的结果来筛选候选人。

最近在忙着弄docker搭建新公司的环境问题，弄的头大，还是抽空跟上专栏的脚步！

2021-04-08

Harry

第二题中，排序，限定返回的记录，取其中分数最小的那一条，偶偶就没想到直接用 limit 就解决问题了。

第四题，思路跟老师一致，但在通过分组统计出学生对应的老师编号后，就卡住了，不知道利用自连接就可以直接获取老师的姓名。

在第五题，我想到要先创建一个临时表用来保存最终的查询结果，然后关联查询两张表，分别把不同班的学生成绩插入到这个临时表中，但如何取每个班的前三名，这是个问题。

然后再回头去看题目要求，只写一个 SQL 查询语句，这也太准了...

2021-04-08