

每日一练3 分页查询

编程要求

根据提示，在右侧编辑器补充代码：

- 1. 按 prod_id 升序查询表 products 中第 6-10 行数据,只要求 prod_id 字段即可（prod_id 从 1 开始）；
- 2. 利用子查询优化查询出按 prod_id 升序的第 10-15 行数据，只要求 prod_id 字段即可。数据表结构如下：

Field	Type	Null	Key	Default	Extra
prod_id	varchar(20)	NO	PRI	NULL	
prod_name	varchar(30)	NO		NULL	
prod_price	varchar(20)	NO		NULL	

测试说明

我们会根据表中数据对你的结果进行测试：

预期输出： prod_id 6 7 8 9 10 prod_id 10 11 12 13 14 15

代码

```
USE Products;
#请在此处添加实现代码
##### Begin #####

#1.分页查询
SELECT prod_id FROM products limit 5, 5;

#2.用子查询优化分页查询语句
SELECT prod_id FROM products WHERE prod_id >= (SELECT prod_id FROM products
limit 10, 1)
limit 5;

##### End #####
```

解析

什么是分页查询

当一个表中有很多符合查询条件的数据的时候，我们往往不需要把他们全部一次性取出来，那样对查询效率或者服务器性能来说都会是一个极大的挑战:例如最简单的商城，假设商城中有一万个数据，但我们在前端可能只会每次看到一页 10 个左右的数据，当我们浏览完这十个商品后，点击下一页浏览其他的商品，这样的分页浏览，对实际存储商品数据的 MySQL 数据库来说，实际上就是使用了 MySQL 的分页查询功能。

limit 和 offset 实现的分页查询

limit 在之前的学习中应该有学习到，它可以返回符合条件的 limit 条数据：

```
select * from table where xxx="xxx" limit 10;
```

输出：检索记录行符合条件的10条数据

那么我们也可以利用 limit 和 offset 来进行分页查询:

```
select * from table where xxx="xxx" limit 10 offset 10;
```

输出：检索记录行符合条件的 11-20 条数据

在实际使用中，我们可以直接把 offset 直接省略掉：

```
select * from table where xxx="xxx" limit 10,10;
```

该语句效果与带 offset 实际上是一模一样的。

大数据量下分页查询优化

```
select * from table where xxx="xxx" limit 100,100;select * from table where  
xxx="xxx" limit 1000,100;select * from table where xxx="xxx" limit  
10000,100;select * from table where xxx="xxx" limit 100000,100;select * from  
table where xxx="xxx" limit 1000000,100;
```

以上语句在实际使用中实际上查询速度会是越来越慢的，这是因为分页查询方式会从数据库第一条记录开始扫描，所以越往后，查询速度越慢，而且查询的数据越多，也会拖慢总查询速度。

如果可以直接定位到偏移量所在记录，那么查询速度肯定能得到一个巨大的提升，所以可以考虑添加一个嵌套子查询，先查询到偏移量位置，再进行分页：

```
select * from table where xxx="xxx" and id>=(select id from table where  
xxx="xxx"limit 100000,1)limit 100;
```