# 112 案例实战：亿级数据量商品系统的SQL调优实战（1）

今天开始进入我们的SQL语句调优的第二个案例实战，这个案例讲的是一个电商平台的亿级数据量的商品系统的SQL语句的性能调优。对于这个案例，我们同样会拆分为三讲来讲解。

上次的案例大家会看到，主要问题在于MySQL内部自动使用了半连接优化，结果半连接的时候导致大量无索引的全表扫描，引发了性能的急剧下降；

而这次的这个案例，其实也是类似的，是我们的MySQL数据库在选择索引的时候，选择了一个不太合适的索引，导致了性能极差，引发了慢查询。

先从当时线上的商品系统出现的一个慢查询告警开始讲起，某一天晚上，我们突然收到了线上数据库的频繁报警，这个报警的意思大致就是说，数据库突然涌现出了大量的慢查询，而且因为大量的慢查询，导致每一个数据库连接执行一个慢查询都要耗费很久。

那这样的话，必然会导致突然过来的很多查询需要让数据库开辟出来更多的连接，因此这个时候报警也告诉我们，数据库的连接突然也暴增了，而且每个连接都打满，每个连接都要执行一个慢查询，慢查询还跑的特别慢。

接着引发的问题，就是数据库的连接全部打满，没法开辟新的连接了，但是还持续的有新的查询发送过来，导致数据库没法处理新的查询，很多查询发到数据库直接就阻塞然后超时了，这也直接导致线上的商品系统频繁的报警，出现了大量的数据库查询超时报错的异常！

当时看到这一幕报警，让人是非常揪心的，因为这种情况，基本意味着你的商品数据库以及商品系统濒临于崩溃了，大量慢查询耗尽了数据库的连接资源，而且一直阻塞在数据库里执行，数据库没法执行新的查询，商品数据库没法执行查询，用户没法使用商品系统，也就没法查询和筛选电商网站里的商品了！

而且大家要知道，当时正好是晚上晚高峰的时候！也就是一个电商网站比较繁忙的时候，虽说商品数据是有多级缓存架构的，但是实际上在下单等过程中，还是会大量的请求商品系统的，所以晚高峰的时候，商品系统本身TPS大致是在每秒几千的。

因此这个时候，发现数据库的监控里显示，每分钟的慢查询超过了10w+！！！也就是说商品系统大量的查询都变成了慢查询！！！

那么慢查询的都是一些什么语句呢？其实主要就是下面这条语句，大家可以看一下，我们做了一个简化：

```
select * from products where category='xx' and sub_category='xx' order by id desc limit xx,xx
```

这其实是一个很稀松平常的SQL语句，他就是用户在电商网站上根据商品的品类以及子类在进行筛选，当然真实的SQL语句里，可能还包含其他的一些字段的筛选，比如什么品牌以及销售属性之类的，我们这里是做了一个简化，然后按id倒序排序，最后是分页，就这么一个语句。

这个语句执行的商品表里大致是1亿左右的数据量，这个量级已经稳定了很长时间了，主要也就是这么多商品，但是上面的那个语句居然一执行就是几十秒！

几十秒，这还得了？基本上数据库的连接全部被慢查询打满，一个连接要执行几十秒的SQL，然后才能执行下一个SQL，此时数据库基本就废了，没法执行什么查询了！！！

所以难怪商品系统本身也大量的报警说查询数据库超时异常了！

**End**