

之前我们把连接的基本语义和基本原理讲了一下，今天开始正式来深入探索一下SQL关联语法的实现原理

首先，先给大家提出一个名词叫做：**嵌套循环关联（nested-loop join）**，这其实就是我们之前给大家提到的最基础的关联执行原理。

简单来说，假设有两个表要一起执行关联，此时会先在一个驱动表里根据他的where筛选条件找出一波数据，比如说找出10条数据吧

接着呢，就对这10条数据走一个循环，用每条数据都到另外一个被驱动表里去根据ON连接条件和WHERE里的被驱动表筛选条件去查找数据，找出来的数据就进行关联。

依次类推，假设驱动表里找出来10条数据，那么就要到被驱动表里去查询10次！

那么如果是三个表进行关联呢？那就更夸张了，你从表1里查出来10条数据，接着去表2里查10次，假设每次都查出来3条数据，然后关联起来，此时你会得到一个30条数据的结果集，接着再用这批数据去表3里去继续查询30次！

这种方法的伪代码有点类似下面这样：

```
1 t1Rows = queryFromt1() // 根据筛选条件对t1表进行查询
2 for t1Row in t1Rows { // 对t1里每一条符合条件的数据进行循环
3     t2Rows = queryFromt2(t1Row) // 拿t1里的数据去t2表里查询以及做关联
4     for t2Row in t2Rows { // 对t1和t2关联后的数据进行循环
5         t3Rows = queryFromt3(t2Row) // 拿t1和t2关联后的数据去t3表里查询和关联
6         for t3Row in t3Rows { // 遍历最终t1和t2和t3关联好的数据
7
8     }
9 }
10 }
```

上面那伪代码其实就是3个表关联的伪代码，用的就是最笨的嵌套循环关联方法，大家可以好好理解上面的伪代码。

不知道大家有没有发现上面那种多表关联方法的问题在哪里？

没错，就是我们往往从驱动表里查出来一波数据之后，要对每一条数据都循环一次去被驱动表里查询数据，所以万一你要是被驱动表的索引都没建好，总不能每次都全表扫描吧？这就是一个很大的问题！

另外一个，刚开始对你的驱动表根据WHERE条件进行查询的时候，也总不能全表扫描吧？这也是一个问题！

所以说，为什么有的时候多表关联很慢呢？答案就在这里了，你两个表关联，先从驱动表里根据WHERE条件去筛选一波数据，这个过程如果你没给驱动表加索引，万一走一个all全表扫描，岂不是速度很慢？

其次，假设你好不容易从驱动表里扫出来一波数据，接着又来一个for循环一条一条去被驱动表里根据ON连接条件和WHERE筛选条件去查，万一你对被驱动表又没加索引，难道又来几十次或者几百次全表扫描？那速度岂不是慢的跟蜗牛一样了！

所以说，通常而言，针对多表查询的语句，我们要尽量给两个表都加上索引，索引要确保从驱动表里查询也是通过索引去查找，接着对被驱动表查询也通过索引去查找。如果能做到这一点，你的多表关联语句性能就会很高！

End