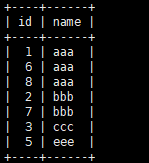
**例如表中数据:**



**表中有7条数据**

**那么name 的索引选择性为 2/7;**

**那么id 的索引选择性为 7/7 = 1**

**1 . 当我们创建复合索引(多列索引)的时候把id索引放再第一个;**

**select \* from aa where id = 3 and name = aaa;**

**2. 当我们创建复合索引(多列索引)的时候把name索引放再第一个;**

**Select \* from aa where id = 3 and name =aaa;**

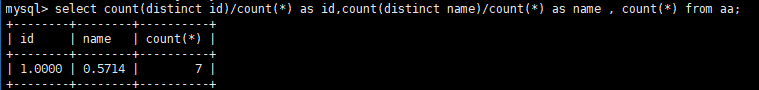
**当第一种情况的下 从复合索引第一个字段开始查的话 一次就拿到了数据;**

**当第二种情况下从复合索引第一个字段开始查的话 要先找到aaa的所在行 3行，然后在去找id=3 所在行;**

**所以可以看得出 : 索引选择性越高越好**

**可以这样看一下索引选择性的高低**

哪个字段的选择性更接近1就把哪个字段索引前面就好。



**当然也是看业务的:**

**比如:** 举个简单的例子，比如要查询某个用户组下有过交易的用户信息

select user\_id from trade where user\_group\_id = 1 and trade\_amount > 0

MySQL为这个查询选择了索引(user\_group\_id,trade\_amount)，如果不考虑特殊情况，这看起来没有任何问题，但实际情况是这张表的大多数数据都是从老系统中迁移过来的，由于新老系统的数据不兼容，所以就给老系统迁移过来的数据赋予了一个默认的用户组。这种情况下，通过索引扫描的行数跟全表扫描基本没什么区别，索引也就起不到任何作用。