**数据库结构设计：**

**主、外键**

主键Primary key，唯一标示一个实体。是保证[数据库](http://lib.csdn.net/base/14" \o "MySQL知识库" \t "http://blog.csdn.net/hejingyuan6/article/details/_blank)的实体完整性，取值非空唯一。

外键Foreign，是用来使表与表之间联系。用来保证数据库的参照完整性，外键的取值必须来自参照表参照列的值，可以为空也可不为空。

**主外键的关系结构：**

①一对一，不用引用主外键，把它们放一个表中即可

     例如：一个学生只能有一个卡号，那么学生跟卡号放在一个表中即可

②一对多，引用主外键，‘一’相当于主键，‘多’即是引用主键的外键。

     例如：一个班级可以有多个学生，并且一个学生只能属于一个班级，这就是一对多的关系；

③多对多关系，需要创建一个表，表中需要两个字段，分别是引用，两个多对多字段主键的外键。

**垂直分割依据：**

“**垂直分割**”是一种把数据库中的表按列变成几张表的方法，这样可以降低表的复杂度和字段的数目，从而达到优化的目的。

示例一：在Users表中有一个字段是家庭地址，这个字段是可选字段，相比起，而且你在数据库操作的时候除了个人信息外，你并不需要经常读取或是改写这个字段。只有用户ID，用户名，口令，用户角色等会被经常使用。

示例二： 你有一个叫 “last\_login” 的字段，它会在每次用户登录时被更新。但是，每次更新时会导致该表的查询缓存被清空。所以，你可以把这个字段放到另一个表中，这样就不会影响你对用户ID，用户名，用户角色的不停地读取了，因为查询缓存会帮你增加很多性能。

所以垂直分割的依据是：

1. 这些字段不经常用到，不会经常性地去Join他们，不然的话，这样的性能会比不分割时还要差
2. 这些字段的更新更为频繁