**表关系**

表关系：表与表之间（实体）有什么样的关系，每种关系应该如何设计表结构。

**一对一**

一对一：一张表中的一条记录与另外一张表中最多有一条明确的关系：通常，此设计方案保证两张表中使用同样的主键即可

学生表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生ID（PRI） | 姓名 | 年龄 | 性别 | 籍贯 | 婚否 | 住址 |
|  |  |  |  |  |  |  |

表的使用过程中：常用的信息会经常去查询，而不常用的信息会偶尔才会用到。

解决方案：将两张表拆分，常见的放一张表，不常见的放一张表

常用表

|  |  |  |  |
| --- | --- | --- | --- |
| 学生ID（PRI） | 姓名 | 年龄 | 性别 |
|  |  |  |  |

不常用表

|  |  |  |  |
| --- | --- | --- | --- |
| 学生ID（PRI） | 籍贯 | 婚否 | 住址 |
|  |  |  |  |

**一对多**

一对多，通常也叫作多对一的关系。通常一对多的关系设计的方案，在“多”关系的表中去维护一个字段，这个字段是“一”关系的主键。

母亲表

|  |  |  |  |
| --- | --- | --- | --- |
| 母亲ID | 姓名 | 年龄 | 身高 |
| M1 |  |  |  |
| M2 |  |  |  |

孩子表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 孩子ID | 姓名 | 年龄 | 身高 | 母亲ID |
| K1 |  |  |  | M1 |
| K2 |  |  |  | M1 |

**多对多**

多对多：一张表中的一条记录在另外一张表中可以匹配到多条记录，反过来也一样。

多对多的关系如果按照多对一的关系维护：就会出现一个字段中有多个其他表的主键，在访问的时候就会带来不便。

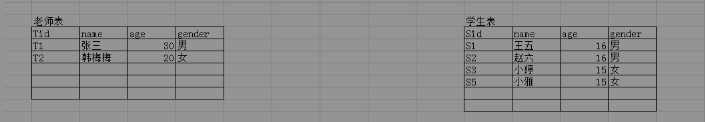
既然通过两张表自己增加字段解决不了问题，那么就通过第三张表来解决。

师生关系

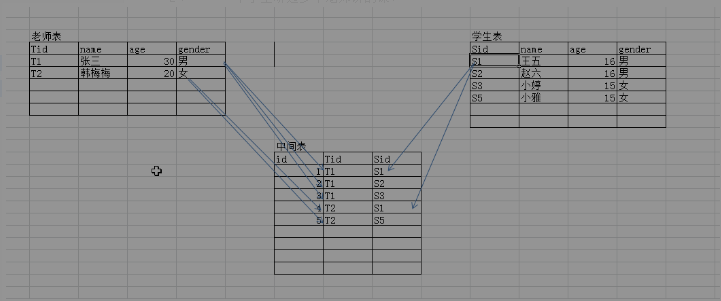
1、 一个老师教过多个班级的学生；

2、 一个学生听过多个老师讲的课；

首先得有两个实体：老师表和学生表



从中间设计一张表：维护两张表对应的联系：每一种联系都包含



多对多解决方案；增加一个中间表，让中间表与对应的其他表形成两个多对一的关系：多对一的解决方案是在“多”表中增加“一”表对应的主键字段。