关系

在数据库中，将实体与实体的关系反应到表的设计上来，可以细分为 3 种，分别为：一对一(1:1)，一对多(1:N)（或多对一(N:1)）和多对多(N:N)。

在此，所有的关系都是指表与表之间的关系。

一对一

一对一，即一张表的一条记录只能与另外一张表的一条记录相对应，反之亦然。

例如，咱们设计一张「个人信息表」，其字段包含：姓名、性别、年龄、身高、体重、籍贯和居住地等。

[](https://camo.githubusercontent.com/42fbed062310a4aa46db4f2f5e9be6d1b39c66ac/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835343532363337)

如上表所示，基本满足咱们的要求，其中姓名、性别和年龄属于常用数据，但是身高、体重、籍贯和居住地为不常用数据。如果每次查询都要查询所有数据的话，那么不常用数据就会影响效率，而且又不常用。因此，咱们可以将常用的数据和不常用的数据分离存储，即分为两张表，例如：

表 1：常用数据

[](https://camo.githubusercontent.com/ecb615eea64fe320eb30b982f9cc8cbf9c53835a/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835353131303630)

表 2：不常用数据

[](https://camo.githubusercontent.com/f893414613110a03649f18cb3b6a710aac03f61c/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835353237343335)

如上面表1和表2所示，通过字段ID，表1中的一条记录只能匹配表2中的一条记录，反之亦然，这就是一对一的关系。

一对多/多对一

一对多，即一张表中的记录可以对应另外一张表中的多条记录，但是反过来，另外一张表中的一条记录只能对应第一张表中的一条记录。

例如，咱们设计「国家城市表」，其包含两个实体，即国家和城市。

表 3：国家表

[](https://camo.githubusercontent.com/4260d2bbc6853c99febcdbe86f88b73fe5c85a9e/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835353431383230)

表 4：城市表

[](https://camo.githubusercontent.com/339d031277e20c5df9667b83719cc8561f357752/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835353535323333)

如上面表3和表4所示，通过字段国家，表3中的一条记录可以匹配表4中的多条记录，但反过来，表4中的一条记录只能匹配表3中的一条记录，这就是典型的一对多的关系。

多对多

多对多，即一张表中的记录可以对应另外一张表中的多条记录，反过来，另外一张表中的一条记录也可以对应第一张表中的多条记录。

例如，咱们设计「教师学生表」，其包含两个实体，即教师和学生。

表 5：教师表

[](https://camo.githubusercontent.com/722e61f7fa3849ac4c075991c120000dec18521d/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835363130353234)

表 6：学生表

[](https://camo.githubusercontent.com/7bc6ce03140b6bfbc194dea19109067dda3462cc/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835363234303134)

观察上面的表5和表6，咱们会发现：表5和表6的设计满足了实体的属性，但没有维护实体之间的关系，即一个老师教过多个学生，一个学生也被多个老师教过。但是无论咱们在表5中还是在表6中增加字段，都会出现一个问题，那就是：该字段要保存多个数据，并且还是与其他表有关系的字段，不符合设计规范。因此，咱们可以再设计一张「中间表」，专门用来维护表5和表6的关系。

表 7：中间表

[](https://camo.githubusercontent.com/42a11122eaff099e13e69cddf5a433e8e35ff741/687474703a2f2f696d672e626c6f672e6373646e2e6e65742f3230313730363031303835363336393532)

观察上面的表5、表6和表7，咱们会发现增加表7之后，咱们维护表5和表6的关系更加方便啦！无论是想从表5通过表7查到表6，还是想从表6通过表7查到表5，都非常容易啦！这就是典型的多对多的关系。