#### 多对多，多对一，一对多三种联系中的键怎么定义

1. 一对多（1:N）

定义：一个实体可以关联多个另一个实体，但反向只能关联一个。  
例子：一个 用户（User） 可以发布多个 订单（Order），但一个订单只能属于一个用户。

2. 多对一（N:1）

定义：多对一是 一对多 关系的反向表达，例如 多个订单属于同一个用户。  
实现方式：与 一对多 相同，Orders 表的 user\_id 作为外键指向 Users 表。

3. 多对多（M:N）

定义：一个实体可以关联多个另一个实体，反向同样如此。  
例子：学生和课程的关系，一个学生可以选多门课程，一门课程也可以被多个学生选修。

键的定义：

Students.id 和 Courses.id 是 主键（PK）。

Student\_Course 作为 中间表，包含：

student\_id 和 course\_id 组成 联合主键（PK），确保唯一性。

student\_id 作为外键（FK），指向 Students.id。

course\_id 作为外键（FK），指向 Courses.id。

#### 讨论三种方式的优缺点，ER，OO，NULL

1. Object-oriented : One relation per subset of subclasses, with all relevant attributes.

优点：1，查询性能高：每个表只存自己需要的数据，不需要 JOIN 操作。  
 2，避免 NULL 值：没有无用字段，节省存储空间。  
 3，符合面向对象思想，适合 ORM（对象关系映射）。

缺点：1， 数据冗余：不同子类的表都有 id 和 name 等字段，重复存储。  
2，扩展困难：如果要新增一个子类（如 Vodka），需要创建新的表。  
3， 跨子类查询不方便：如果要查询所有酒类（不管是 Wine 还是 Beer），需要 UNION 多个表，性能较低。

适用场景：

1，子类之间的属性差异很大，有很多独特字段，避免无意义的 NULL。

2，查询时通常只访问一个子类的数据，比如查询所有葡萄酒信息，而不是所有 酒的信息。

1. Use nulls : One relation; entities have NULL in attributes that don’t belong to them.

优点：1，查询简单：所有数据在一个表中，不需要 JOIN 或 UNION。  
 2，结构简单：只需要维护一张表，不用处理多个子类表。  
 3，易扩展：增加新类别时，只需添加新 type 值，无需新建表。

缺点：1，NULL 值多：字段多但大部分为空，浪费存储空间。  
 2，数据完整性问题：无法强制某些字段仅适用于某个 type，可能导致错误数据 （如 Beer 竟然填了 grape\_variety）。  
 3，索引效率低：如果查询 Wine，需要加 WHERE type='Wine'，影响索引优化。

适用场景：

1，子类字段差异较小，例如大部分字段通用，只有少量特有字段。

2，查询经常跨类别，比如查询所有酒，不管是 Wine 还是 Beer。

3，子类变化较多，比如系统支持的酒类会经常扩展。

1. E/R style : One relation for each subclass:

优点：1，数据完整性高：Wine 只能存 Wine 的字段，Beer 只能存 Beer 的字段。  
2，节省存储空间：避免 NULL，数据结构更加规范。  
3，易扩展：增加 Vodka 只需新建表，而不影响已有表。

缺点：1，查询复杂：查询一个具体的子类数据需要 JOIN。  
2，写入操作较多：插入一个 Wine，需要同时插入 Drinks 和 Wine 两张表。  
3，跨子类查询稍麻烦：查询所有酒类需要 JOIN 或 UNION。

适用场景：

1，适用于 字段差异较大，但仍然有共同属性的情况。

2，需要保证数据完整性，避免无效字段填充。

3，查询性能比写入更重要，因为查询涉及 JOIN，但结构规范。

1. 设计图书馆管理的数据库，包含以下信息：图书名称、ISBN（International Standard Book Number，国际标准书号）、图书类别、书条形码（条形码可以唯一的对应一本书）、是否已经出借、学生的信息（学号、姓名、班级、性别等）、学生已经借阅的图书、借书时间 、还书时间、图书证号码。

其中语义有：每位同学的学号就是图书证号码 ，每位同学最多可以借阅5本图书，要登记学生借阅的具体书（即同时登记ISBN和条形码，每本书有若干册，ISBN相同，但条形码不同） ，当归还图书时需要把该借阅信息作为历史信息保存起来，借书期限为一个月（不续借），如果学生未能借到书，可以进行预约登记，一旦有预约的图书归还，数据库将自动使用电子邮件通知预约学生来借书（假设数据库系统中有函数mailto(email\_address, text)），当借书完毕，删除预约信息。

画ER图

