

数据库设计中的14个技巧 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/137/2021\\_2022\\_\\_E6\\_95\\_B0\\_E6\\_8D\\_AE\\_E5\\_BA\\_93\\_E8\\_c98\\_137765.htm](https://www.100test.com/kao_ti2020/137/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E8_c98_137765.htm)

1. 原始单据与实体之间的关系 可以是一对一、一对多、多对多的关系。在一般情况下，它们是一对一的关系：即一张原始单据对应且只对应一个实体。在特殊情况下，它们可能是一对多或多对一的关系，即一张原始单据对应多个实体，或多张原始单据对应一个实体。这里的实体可以理解为基本表。明确这种对应关系后，对我们设计录入界面大有好处。【例1】：一份员工履历资料，在人力资源信息系统中，就对应三个基本表：员工基本情况表、社会关系表、工作简历表。这就是“一张原始单据对应多个实体”的典型例子。

2. 主键与外键 一般而言，一个实体不能既无主键又无外键。在E - R图中，处于叶子部位的实体，可以定义主键，也可以不定义主键(因为它无子孙)，但必须要有外键(因为它有父亲)。主键与外键的设计，在全局数据库的设计中，占有重要地位。当全局数据库的设计完成以后，有个美国数据库设计专家说：“键，到处都是键，除了键之外，什么也没有”，这就是他的数据库设计经验之谈，也反映了他对信息系统核心(数据模型)的高度抽象思想。因为：主键是实体的高度抽象，主键与外键的配对，表示实体之间的连接。

3. 基本表的性质 基本表与中间表、临时表不同，因为它具有如下四个特性：(1) 原子性。基本表中的字段是不可再分解的。(2) 原始性。基本表中的记录是原始数据(基础数据)的记录。(3) 演绎性。由基本表与代码表中的数据，可以派生出所有的输出数据。(4) 稳定性。基本表的结构

是相对稳定的，表中的记录是要长期保存的。理解基本表的性质后，在设计数据库时，就能将基本表与中间表、临时表区分开来。

#### 4. 范式标准

基本表及其字段之间的关系，应尽量满足第三范式。但是，满足第三范式的数据库设计，往往不是最好的设计。为了提高数据库的运行效率，常常需要降低范式标准：适当增加冗余，达到以空间换时间的目的。

【例2】：有一张存放商品的基本表，如表1所示。“金额”这个字段的存 在，表明该表的设计不满足第三范式，因为“金额”可以由“单价”乘以“数量”得到，说明“金额”是冗余字段。但是，增加“金额”这个冗余字段，可以提高查询统计的速度，这就是以空间换时间的作法。在Rose 2002中，规定列有两种类型：数据列和计算列。“金额”这样的列被称为“计算列”，而“单价”和“数量”这样的列被称为“数据列”。

商品名称	商品型号	单价	数量	金额
电视机	29	2,500	40	100,000

#### 5. 通俗地理解三个范式

通俗地理解三个范式，对于数据库设计大有好处。在数据库设计中，为了更好地应用三个范式，就必须通俗地理解三个范式(通俗地理解是够用的理解，并不是最科学最准确的理解)：

- 第一范式：1NF是对属性的原子性约束，要求属性具有原子性，不可再分解；
- 第二范式：2NF是对记录的惟一性约束，要求记录有惟一标识，即实体的惟一性；
- 第三范式：3NF是对字段冗余性的约束，即任何字段不能由其他字段派生出来，它要求字段没有冗余。没有冗余的数据库设计可以做到。但是，没有冗余的数据库未必是最好的数据库，有时为了提高运行效率，就必须降低范式标准，适当保留冗余数据。具体做法是：在概念数据模型设计时遵守第三范式，降低范式标

准的工作放到物理数据模型设计时考虑。降低范式就是增加字段，允许冗余。

6. 要善于识别与正确处理多对多的关系 若两个实体之间存在多对多的关系，则应消除这种关系。消除的办法是，在两者之间增加第三个实体。这样，原来一个多对多的关系，现在变为两个一对多的关系。要将原来两个实体的属性合理地分配到三个实体中去。这里的第三个实体，实质上是一个较复杂的关系，它对应一张基本表。一般来讲，数据库设计工具不能识别多对多的关系，但能处理多对多的关系。〔例3〕：在“图书馆信息系统”中，“图书”是一个实体，“读者”也是一个实体。这两个实体之间的关系，是一个典型的多对多关系：一本图书在不同时间可以被多个读者借阅，一个读者又可以借多本图书。为此，要在二者之间增加第三个实体，该实体取名为“借还书”，它的属性为：借还时间、借还标志(0表示借书，1表示还书)，另外，它还应该有两个外键(“图书”的主键，“读者”的主键)，使它能与“图书”和“读者”连接。

7. 主键PK的取值方法 PK是供程序员使用的表间连接工具，可以是一无物理意义的数字串，由程序自动加1来实现。也可以是有物理意义的字段名或字段名的组合。不过前者比后者好。当PK是字段名的组合时，建议字段的个数不要太多，多了不但索引占用空间大，而且速度也慢。

8. 正确认识数据冗余 主键与外键在多表中的重复出现，不属于数据冗余，这个概念必须清楚，事实上有许多人还不清楚。非键字段的重复出现，才是数据冗余！而且是一种低级冗余，即重复性的冗余。高级冗余不是字段的重复出现，而是字段的派生出现。〔例4〕：商品中的“单价、数量、金额”三个字段，“金额”就是由“单价”乘以“

数量”派生出来的，它就是冗余，而且是一种高级冗余。冗余的目的是为了提高处理速度。只有低级冗余才会增加数据的不一致性，因为同一数据，可能从不同时间、地点、角色上多次录入。因此，我们提倡高级冗余(派生性冗余)，反对低级冗余(重复性冗余)。

9. E - R图没有标准答案 信息系统的E - R图没有标准答案，因为它的设计与画法不是惟一的，只要它覆盖了系统需求的业务范围和功能内容，就是可行的。反之要修改E - R图。尽管它没有惟一的标准答案，并不意味着可以随意设计。好的E - 图的标准是：结构清晰、关联简洁、实体个数适中、属性分配合理、没有低级冗余。

10. 视图技术在数据库设计中很有用 与基本表、代码表、中间表不同，视图是一种虚表，它依赖数据源的实表而存在。视图是供程序员使用数据库的一个窗口，是基表数据综合的一种形式，是数据处理的一种方法，是用户数据保密的一种手段。为了进行复杂处理、提高运算速度和节省存储空间，视图的定义深度一般不得超过三层。若三层视图仍不够用，则应在视图上定义临时表，在临时表上再定义视图。这样反复交迭定义，视图的深度就不受限制了。对于某些与国家政治、经济、技术、军事和安全利益有关的信息系统，视图的作用更加重要。这些系统的基本表完成物理设计之后，立即在基本表上建立第一层视图，这层视图的个数和结构，与基本表的个数和结构是完全相同。并且规定，所有的程序员，一律只准在视图上操作。只有数据库管理员，带着多个人员共同掌握的“安全钥匙”，才能直接在基本表上操作。请读者想想：这是为什么？

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)