

浅谈数据库设计技巧(上) 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\\_B5\\_85\\_E8\\_B0\\_88\\_E6\\_95\\_B0\\_E6\\_c98\\_137799.htm](https://www.100test.com/kao_ti2020/137/2021_2022__E6_B5_85_E8_B0_88_E6_95_B0_E6_c98_137799.htm) 说到数据库，我认为不能不先谈数据结构。1996年，在我初入大学学习计算机编程时，当时的老师就告诉我们说：计算机程序 = 数据结构 + 算法。尽管现在的程序开发已由面向过程为主逐步过渡到面向对象为主，但我还是深深赞同8年前老师的告诉我们的公式：计算机程序 = 数据结构 + 算法。面向对象的程序开发，要做的第一件事就是，先分析整个程序中需处理的数据，从中提取出抽象模板，以这个抽象模板设计类，再在其中逐步添加处理其数据的函数(即算法)，最后，再给类中的数据成员和函数划分访问权限，从而实现封装。数据库的最初雏形据说源自美国一个奶牛场的记账簿(纸质的，由此可见，数据库并不一定是存储在电脑里的数据^\_^)，里面记录的是该奶牛场的收支账目，程序员在将其整理、录入到电脑中时从中受到启发。当按照规定好的数据结构所采集到的数据量大到一定程度后，出于程序执行效率的考虑，程序员将其中的检索、更新维护等功能分离出来，做成单独调用的模块，这个模块后来就慢慢发展、演变成现在我们所接触到的数据库管理系统(DBMS)程序开发中的一个重要分支。下面进入正题，首先按我个人所接触过的程序给数据库设计人员的功底分一下类：  
： 1、没有系统学习过数据结构的程序员。这类程序员的作品往往只是他们的即兴玩具，他们往往习惯只设计有限的几个表，实现某类功能的数据全部塞在一个表中，各表之间几乎毫无关联。网上不少的免费管理软件都是这样的东西，当

程序功能有限，数据量不多的时候，其程序运行起来没有什么问题，但是如果用其管理比较重要的数据，风险性非常大。

2、系统学习过数据结构，但是还没有开发过对程序效率要求比较高的管理软件的程序员。这类人多半刚从学校毕业不久，他们在设计数据库表结构时，严格按照教科书上的规定，死扣E-R图和3NF(别灰心，所有的数据库设计高手都是从这一步开始的)。他们的作品，对于一般的access型轻量级的管理软件，已经够用。但是一旦该系统需要添加新功能，原有的数据库表差不多得进行大换血。

3、第二类程序员，在经历过数次程序效率的提升，以及功能升级的折腾后，终于升级成为数据库设计的老鸟，第一类程序员眼中的高人。这类程序员可以胜任二十个表以上的中型商业数据管理系统的开发工作。他们知道该在什么样的情况下保留一定的冗余数据来提高程序效率，而且其设计的数据库可拓展性较好，当用户需要添加新功能时，原有数据库表只需做少量修改即可。

4、在经历过上十个类似数据库管理软件的重复设计后，第三类程序员中坚持下来没有转行，而是希望从中找出“偷懒”窍门的有心人会慢慢觉悟，从而完成量变到质变的转换。他们所设计的数据库表结构有一定的远见，能够预测到未来功能升级所需要的数据，从而预先留下伏笔。这类程序员目前大多晋级成数据挖掘方面的高级软件开发人员。

5、第三类程序员或第四类程序员，在对现有的各家数据库管理系统的原理和开发都有一定的钻研后，要么在其基础上进行二次开发，要么自行开发一套有自主知识产权的通用数据库管理系统。我个人正处于第三类的末期，所以下面所列出的一些设计技巧只适合第二类和部分第三类数据库设计人员。同时

，由于我很少碰到有兴趣在这方面深钻下去的同行，所以文中难免出现错误和遗漏，在此先行声明，欢迎大家指正，不要藏私哦8) 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)