

如何学好数据库编程？PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/264/2021\\_2022\\_\\_E5\\_A6\\_82\\_E4\\_BD\\_95\\_E5\\_AD\\_A6\\_E5\\_c97\\_264258.htm](https://www.100test.com/kao_ti2020/264/2021_2022__E5_A6_82_E4_BD_95_E5_AD_A6_E5_c97_264258.htm) 好像武侠小说里边说的：“你的招式忘了没有？”，回答：“差不多忘了”，“忘了就好”。这与数据库编程有什么关系？关系可大了。同志们学过Pascal、BASIC、C(C)没有？如果没有，FOXBASE、FOXPRO应该学过吧？按以上这些语言编程，都是过程化的，说白一点就是一个数据一个数据、一条记录一条记录去处理(FOXBASE、FOXPRO不完全这样，但书上也经常是这样介绍的)，当初我接触Access 97时，一下子没有了IF、FOR这些语句(指数据处理)，都用SQL语句，真是找不到北了，好在我学SQL语言时，也尽量忘掉这个IF、FOR，到我忘得差不多时，功夫也进了一大步，原来要编一大段程序，现在一两条SQL语句搞定，就算用多几条SQL语句，由于是在图形界面下做，可视化操作，拉拉扯扯，再修改一下生成的SQL语句，也就省事多了。由于ACCESS具备完整的SQL语言(FOXBASE没有、FOXPRO不完整)，我从ACCESS 97开始用ACCESS编程，到现在为止，DAO、ADO很少用，加上最近从爱赛思上接触的一些技术，基本上不用DAO、ADO都可以了，可以从我的“未完工的库存管理”中看出，只是在特殊情况下才偶尔用一下。(少用，但不是不用，还得学，不要误解) 如何学好数据库编程？下面介绍一下本人的一些经验，仅供参考：1.首先要把原来一个数据一个数据、一条记录一条记录的数据处理方式忘掉，越彻底越好。现在用成批处理了。少用记录集一条记录一条记录地处理，尽量用SQL语句

。 2.学好关系数据库的理论，尤其是规范化理论，表的设计一定要规范化，最起码要规范化到第三范式。集合运算(并、交、差)。关系运算(选择、投影、连接)。其中连接与规范化是紧密结合的。 3.运用面向对象的技术：面向对象的分析OOA、面向对象的设计OOD、面向对象的编程OOP，根据表的关系，用窗体和子窗体、报表和子报表，仿真面向对象，这样可以增加程序的可读性和可维护性。(这是高级技术，同志们不要轻视，做大项目你就知道有用了) 4.用查询时，通常一步做不出来，可以分几步做，本人通常是这么做的，从我给网友回复的例子中也可以看得出。为什么要这样做？(1)有些是SQL语言的限制，没办法一步做出来，逼的；(2)可以检查每一步查询的结果，容易调试；(3)增加可读性，便于日后维护。 5.查询的结果用窗体显示或用报表打印，两者的技术差不多。通常改变打开窗体或报表的条件就可控制显示或打印的记录范围。另外用查询做数据源时，动态改变查询中的SQL语句，比在查询中引用窗体的控件要方便，因为SQL语句生成是在VBA中，可以先存放在字符变量中，然后再更新查询的SQL语句，这样就可以用断点来检查变量值对不对，如果在查询中引用窗体的控件，尤其是包含IIF()函数时，调试是很困难的。 6.开发一个系统，首先要解决技术问题，即算法，用简单例子，把算法弄懂了，再详细设计，这一点从网友的提问中可以看出，有很多人问题表达不清楚，有的人其中夹了很多与算法无关的东西，尤其是很专业的东西，别人不容易看得明白，由于算法没搞清楚，程序就无法编了。 7.不要使用过多的工具特性，使用过多的工具特性会使可读性降低，可维护性差，要别人帮忙时难以得到帮助，更要命

的是可移植性差，从MDB到ADP就可以体会到了，所以在编程时可读性很重要，可移植性更重要，你甘心自己的程序就固定在一个环境下运行吗？你甘心永远用一个工具搞开发吗？为了你的“钱途”，不要玩弄太多的技巧，当然技术研究是另外一码事。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)