

数据库设计注意事项 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/449/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E8_c97_449158.htm

数据库设计注意事项
老实说很多人不是很注重数据库的设计，今天看到一篇不错的文章，自己记下，如果有需要的人可以看看。强调：数据库设计看似简单，但是绝对不能轻视！一个成功的管理系统，是由：[50%的业务 50%的软件]所组成，而50%的成功软件又有[25%的数据库 25%的程序]所组成，数据库设计的好坏是一个关键。如果把企业的数据库比做生命所必需的血液，那么数据库的设计就是应用中最重要的一部分。有关数据库设计的材料汗牛充栋，大学学位课程里也有专门的讲述。不过，就如我们反复强调的那样，再好的老师也比不过经验的教诲。所以我归纳历年来所走的弯路及体会，并在网上找了些对数据库设计颇有造诣的专业人士给大家传授一些设计数据库的技巧和经验。精选了其中的60个最佳技巧，并把这些技巧编写成了本文，为了方便索引其内容划分为5个部分：
第1部分 - 设计数据库之前这一部分罗列了12个基本技巧，包括命名规范和明确业务需求等。
第2部分 - 设计数据库表总共24个指南性技巧，涵盖表内字段设计以及应该避免的常见问题等。
第3部分 - 选择键怎么选键呢？这里有10个技巧专门涉及系统生成的主键的正确用法，还有何时以及如何索引字段以获得最佳性能等。
第4部分 - 保证数据完整性讨论如何保持数据库的清晰和健壮，如何把有害数据降低到最小程度。
第5部分 - 各种小技巧不包括在以上4个部分中的其他技巧，五花八门，有了它们希望你的数据库开发工作会

更轻松一些。第 1 部分 - 设计数据库之前考察现有环境在设计一个新数据库时，你不但应该仔细研究业务需求而且还要考察现有的系统。大多数数据库项目都不是从头开始建立的；通常，机构内总会存在用来满足特定需求的现有系统（可能没有实现自动计算）。显然，现有系统并不完美，否则你就不必再建立新系统了。但是对旧系统的研究可以让你发现一些可能会忽略的细微问题。一般来说，考察现有系统对你绝对有好处。定义标准的对象命名规范一定要定义数据库对象的命名规范。对数据库表来说，从项目一开始就要确定表名是采用复数还是单数形式。此外还要给表的别名定义简单规则（比方说，如果表名是一个单词，别名就取单词的前 4 个字母；如果表名是两个单词，就各取两个单词的前两个字母组成 4 个字母长的别名；如果表的名称由 3 个单词组成，你不妨从头两个单词中各取一个然后从最后一个单词中再取出两个字母，结果还是组成 4 字母长的别名，其余依次类推）对工作表来说，表名可以加上前缀 WORK_ 后面附上采用该表的应用程序的名字。表内的列[字段]要针对键采用一整套设计规则。比如，如果键是数字类型，你可以用 _N 作为后缀；如果是字符类型则可以采用 _C 后缀。对列[字段]名应该采用标准的前缀和后缀。再如，假如你的表里有好多“money”字段，你不妨给每个列[字段]增加一个 _M 后缀。还有，日期列[字段]最好以 D_ 作为名字打头。检查表名、报表明和查询名之间的命名规范。你可能会很快就被这些不同的数据库要素的名称搞糊涂了。假如你坚持统一地命名这些数据库的不同组成部分，至少你应该在这些对象名字的开头用 Table、Query 或者 Report 等前缀加以区别。如果采用了

Microsoft access，你可以用 qry、rpt、tbl 和 mod 等符号来标识对象（比如 tbl_Employees）。我在和 SQL Server 打交道的时候还用过 tbl 来索引表，但我用 sp_company（现在用 sp_left_）标识存储过程，因为在有的时候如果我发现了更好的处理办法往往会保存好几个拷贝。我在实现 SQL Server 2000 时用 udf_（或者类似的标记）标识我编写的函数。工欲善其事，必先利其器采用理想的数据库设计工具，比如：SyBase 公司的 PowerDesign，她支持 PB、VB、Delphe 等语言，通过 ODBC 可以连接市面上流行的 30 多个数据库，包括 dBase、FoxPro、VFP、SQL Server 等，今后有机会我将着重介绍 PowerDesign 的使用。获取数据模式资源手册正在寻求示例模式的人可以阅读《数据模式资源手册》一书，该书由 Len Silverston、W. H. Inmon 和 Kent Graziano 编写，是一本值得拥有的最佳数据建模图书。该书包括的章节涵盖多种数据领域，比如人员、机构和工作效能等。其他的你还可以参考：[1] 萨师煊 王珊著 数据库系统概论(第二版)高等教育出版社 1991、[2][美] Steven M.Bobrowski 著 Oracle 7 与客户 / 服务器计算技术从入门到精通 刘建元等译 电子工业出版社，1996、[3]周中元 信息系统建模方法(下) 电子与信息化 1999年第3期，1999 畅想未来，但不可忘了过去的教训我发现询问用户如何看待未来需求变化非常有用。这样做可以达到两个目的：首先，你可以清楚地了解应用设计在哪个地方应该更具灵活性以及如何避免性能瓶颈；其次，你知道发生事先没有确定的需求变更时用户将和你一样感到吃惊。一定要记住过去的经验教训！我们开发人员还应该通过分享自己的体会和经验互相帮助。即使用户认为他们再也不需要什么支持了，我们也应该

对他们进行这方面的教育，我们都曾经面临过这样的时刻“当初要是这么做了该多好..”。在物理实践之前进行逻辑设计在深入物理设计之前要先进行逻辑设计。随着大量的 CASE 工具不断涌现出来，你的设计也可以达到相当高的逻辑水准，你通常可以从整体上更好地了解数据库设计所需要的方方面面。了解你的业务在你百分百地确定系统从客户角度满足其需求之前不要在你的 ER（实体关系）模式中加入哪怕一个数据表（怎么，你还没有模式？那请你参看技巧 9）。了解你的企业业务可以在以后的开发阶段节约大量的时间。一旦你明确了业务需求，你就可以自己做出许多决策了。一旦你认为你已经明确了业务内容，你最好同客户进行一次系统的交流。采用客户的术语并且向他们解释你所想到的和你所听到的。同时还应该用可能、将会和必须等词汇表达出系统的关系基数。这样你就可以让你的客户纠正你自己的理解然后做好下一步的 ER 设计。创建数据字典和 ER 图表一定要花点时间创建 ER 图表和数据字典。其中至少应该包含每个字段的数据类型和在每个表内的主外键。创建 ER 图表和数据字典确实有点费时但对其他开发人员要了解整个设计却是完全必要的。越早创建越能有助于避免今后面临的可能混乱，从而可以让任何了解数据库的人都明确如何从数据库中获得数据。有一份诸如 ER 图表等最新文档其重要性如何强调都不过分，这对表明表之间关系很有用，而数据字典则说明了每个字段的用途以及任何可能存在的别名。对 SQL 表达式的文档化来说这是完全必要的。创建模式一张图表胜过千言万语：开发人员不仅要阅读和实现它，而且还要用它来帮助自己和用户对话。模式有助于提高协作效能，这样在先期的数据库设计

中几乎不可能出现大的问题。模式不必弄的很复杂；甚至可以简单到手写在一张纸上就可以了。只是要保证其上的逻辑关系今后能产生效益。从输入输出下手在定义数据库表和字段需求（输入）时，首先应检查现有的或者已经设计出的报表、查询和视图（输出）以决定为了支持这些输出哪些是必要的表和字段。举个简单的例子：假如客户需要一个报表按照邮政编码排序、分段和求和，你要保证其中包括了单独的邮政编码字段而不要把邮政编码糅进地址字段里。报表技巧要了解用户通常是如何报告数据的：批处理还是在线提交报表？时间间隔是每天、每周、每月、每个季度还是每年？如果需要的话还可以考虑创建总结表。系统生成的主键在报表中很难管理。用户在具有系统生成主键的表内用副键进行检索往往会返回许多重复数据。这样的检索性能比较低而且容易引起混乱。理解客户需求看起来这应该是显而易见的事，但需求就是来自客户（这里要从内部和外部客户的角度考虑）。不要依赖用户写下来的需求，真正的需求在客户的脑袋里。你要让客户解释其需求，而且随着开发的继续，还要经常询问客户保证其需求仍然在开发的目的之中。一个不变的真理是：“只有我看见了，我才知道我想要的是什么”必然会导致大量的返工，因为数据库没有达到客户从来没有写下来的需求标准。而更糟的是你对他们需求的解释只属于你自己，而且可能是完全错误的。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