

数据库课程设计有哪些 数据库课程设计 心得体会(精选5篇)

在日常的学习、工作、生活中，肯定对各类范文都很熟悉吧。相信许多人会觉得范文很难写？下面是小编帮大家整理的优质范文，仅供参考，大家一起来看看吧。

数据库课程设计有哪些篇一

在这次课程设计中，我们小组设计的是一个仓库管理系统，通过设计这个系统，可以更方便地实现信息的浏览，查询，添加等操作和报表输出等功能，具体包括能实现产品入库管理，可以填写入库单，确认产品入库；产品出库管理，可以填写出库单，确认出库；借出管理，凭借条借出，然后能够还库；初始库存设置，设置库存的初始值，库存的上下警戒限；可以进行盘库，反映每月，年的库存情况；可以查询产品入库情况，出库情况，当前库存情况，可以按出库单，入库单，产品，时间进行查询；虽然这个系统的功能还不是很完善，还有很多功能尚未实现，但这也是我们共同努力的结果，只要有更多的时间，我相信我们能完成的更好。

通过这次将近一个星期的课程设计实训，我对数据库在生活中的广泛应用有了更深的体会，同时也加深了对access20xx的认识，了解了相关的access的知识，尤其是对数据库的原理与开发程序的应用有了更进一步的认识和了解。由于不会vb编程，很多功能实现起来很困难，但好在access有查询、宏和窗体，特别是宏操作可以将几个操作结合在一起，可以执行多个查询，可以调用多次窗体。例如，在某个窗体中要显示设备代码表信息，所以这个窗体中建立一个子窗体用于设备代码表的显示，我们打开窗体后又对设备代码表进行了跟新，所以窗体中显示的并不是最新数据，我们往往需要及时看到设备代码表跟新后的最新数据，窗体不会自动跟新我

们就不会看到最新数据，所以这时候我们就会用一个宏操作在对设备代码表跟新之后关闭窗体在打开窗体，这样窗体就相当于完成了一次跟新操作。又例如，在一个表a中刚刚追加了一条记录，我们要将每次刚刚新添加的记录跟新到另一个表b中，用一个查询是很难完成的，经过认真思考，得到了解决的办法是，需要三个步骤，第一步，在表a中添加标志字段，设置默认值为‘否’。第二步，编写跟新查询将标志字段为‘否’的数据跟新到表b中。第三步，编写跟新查询将表a中标志字段为‘否’的跟新为‘是’。

读书破万卷下笔如有神，以上就是为大家带来的10篇《数据库课程设计心得体会》，希望对您的写作有一定的参考作用，更多精彩的范文样本、模板格式尽在。

数据库课程设计有哪些篇二

两个星期时间非常快就过去了，这两个星期不敢说自己有多大进步，获得了多少知识，但起码是了解了项目开发部分过程。虽说上过数据库上过管理信息系统等相关课程，但是没有亲身经历过相关设计工作细节。这次实习证实提供了一个很好机会。

通过这次课程设计发现这其中需要很多知识我们没有接触过，去图书馆查资料时候发现我们前边所学到仅仅是皮毛，还有很多需要我们掌握东西我们根本不知道。同时也发现有很多已经学过东西我们没有理解到位，不能灵活运用于实际，不能很好用来解决问题，这就需要不断大量实践，通过不断自学，不断地发现问题，思考问题，进而解决问题。在这个过程中我们将深刻理解所学知识，同时也可以学到不少很实用东西。从各种文档阅读到开始需求分析、概念结构设计、逻辑结构设计、物理结构设计。亲身体会了一回系统设计开发过程。很多东西书上写很清楚，貌似看着也很简单，思路非常清晰。但真正需要自己想办法去设计一个系统时候才发

现其中难度。经常做到后面突然就发现自己一开始设计有问题，然后又回去翻工，在各种反复中不断完善自己想法。

我想有这样问题不止我一个，事后想想是一开始着手做时候下手过于轻快，或者说是根本不了解自己要做这个系统是给谁用。因为没有事先做过仔细用户调查，不知道整个业务流程，也不知道用户需要什么功能就忙着开发，这是作为设计开发人员需要特别警惕避免，不然会给后来工作带来很大的麻烦，甚至可能会需要全盘推倒重来。所以以后课程设计要特别注意这一块设计。

按照要求，我们做是机票预订系统。说实话，我对这个是一无所知，没有订过机票，也不知道航空公司是怎么一个流程。盲目开始设计下场我已经尝过了，结果就是出来一个四不像设计方案，没有什么实际用处。没有前期调查，仅从指导书上那几条要求着手是不够。

在需求分析过程中，我们通过上网查资料，去图书馆查阅相关资料，结合我们生活经验，根据可行性研究结果和客户要求，分析现有情况及问题，采用client/server结构，将机票预定系统划分为两个子系统：客户端子系统，服务器端子系统。在两周时间里，不断地对程序及各模块进行修改、编译、调试、运行，其间遇到很多问题：由于忘记了一些java语言规范使得在调试过程中一些错误没有发现，通过这次课程设计，我对调试掌握得更加熟练了，意识到了程序语言规范性以及我们在编程时要有严谨态度，同时在写程序时如有一定量注释，既增加了程序可读性，也可以使自己在读程序时更容易。

我们学习并应用了sql语言，对数据库创建、修改、删除方法有了一定了解，通过导入表和删除表、更改表学会了对于表一些操作，为了建立一个关系数据库信息管理系统，必须得经过系统调研、需求分析、概念设计、逻辑设计、物理设计、系统调试、维护以及系统评价一般过程，为毕业设计打下基础。

很多事情不是想象中那么简单，它涉及到各种实体、属性、数据流程、数据处理等等。很多时候感觉后面设计根本无法继续，感觉像是被前面做各种图限制了。在做关系模型转换时候碰到有些实体即可以认为是实体又可以作为属性，为了避免冗余，尽量按照属性处理了。

物理结构设计基本没有碰到问题，这一块和安全性、完整性不觉就会在物理结构设计中添加一些安全设置：主键约束、check约束、default定义等。最后才做索引部分，对一些比较经常使用搜索列，外键上建立索引，这样可以明显加快检索速度，最后别忘记重要安全性设置，限制用户访问权限，新建用户并和数据库用户做相应映射。

不管做什么，我们都要相信自己，不能畏惧，不能怕遇到困难，什么都需要去尝试，有些你开始认为很难事在你尝试之后你可能会发现原来她并没有你以前觉得那样，自己也是可以。如果没有自信，没有目标，没有信心就不可能把事情做好，当其他人都在迷茫时候，自己一定要坚信目标，大学毕业出去即面临找工作，从学习这个专业，到以后从事这方面工作都需要不断地去学习去实践，这次实践可以给我们敲一个警钟，我们面临毕业，面临择业，需要这些实践经验，在困难面前要勇于尝试，这是这次课程设计给我最大感想！

以上基本是这次实习体会了，设计进行非常艰难，编码非常不容易，才发现做一个项目最重要不在于如何实现，而是实现之前需求分析和模块设计。创新很难，有些流行系统其实现并不难，难在于对市场分析和准确定位。设计，是一个任重道远过程。

数据库课程设计有哪些篇三

由于平时接触的都是是一些私人项目，这些项目大都是一些类库，其他人的交流相对可以忽略不计，因此也就不考虑规范化的文档。实际上从学习的经历来看，我们接触的知识体系

都是属于比较老或比较传统的，与现在发展迅速的it行业相比很多情况已不再适用，尤其是当开源模式逐渐走近开发者后更是如此。

虽然这次是一个数据库课程设计，由于本人在选择项目的时候是本着对自己有实际应用价值的角度考虑的，所以其中也涉及到一些数据库以外的设计。对于ooa/ood的开发模式有时不免要提出一些疑问□uml是设计阶段的工具，而它基本涵盖了软件设计的方方面面，也就是说按照这一软件工程的正常流程，在动手写第一句代码之前，开发人员已经非常熟悉软件产品了，这对于相当有经验的架构师一类人说可能会很容易，但是我们作为学生，连足够的编码经验都没有，却首先被教授并要求先ooa再oop□这样直接导致的问题就是文档与编码对不上号，在修改代码的时候基本不会再去审查文档和先前的分析。甚至根本就是现有代码再有文档，即便是这种情况，代码与文档还是不对应。不可否认，在传统软件工程的详细设计之前的项目过程中还是有很多利于项目开发的部分的。所以我就一直在寻找适合我——针对探究型项目——的开发模式，这次的项目也算是一次尝试，当然这个过程并不会太短。

回到数据库设计上了，这次的数据库设计我是严格按照数据库建模的步骤来进行的，老实说我并没有感觉这样的流程对开发带来多大的帮助，反倒是觉得将思维转化为图表很浪费时间。总体上来说这次的项目也不是很大，而且在数据库的设计上比较保守，也就是说实际上数据库设计还可以再完善完善的。随着我对计算机领域的拓宽和加深，我也会静下心来思考在接触计算机之前的行为，很多次我能深切感觉到，其实我的大脑（未于别人比较）本身就是在使用一种更接近关系数据库的方式来记忆，所以我很可恨自然的设计出符合三范式的表结构来，即便我不知道这些范式的确切含义。可能就像“范式不太容易用通俗易懂的方式解释”一样，在“让工具用图标表述我的思维”时费了一番力气。

从我作为项目的提出人和实现者来看，这是个失败的项目，结合几次教学项目的实践，发现这也已经不是第一次了。主观原因占多数，比如，尝试新的开发方式，根据设计花了太多的时间来抽象出公用的库而忽略业务逻辑。就这次项目而言，失败的原因有以下几点：

使用了新的开发环境[vim]这是首次在脱离高级ide的情况下编码。

使用了新的开发语言[python][actionscript3]因为我一直比较喜欢“学以致用”，而且这样的“数据驱动型”软件的整套自实现的库都已经完成了，但是由于语言本身的差异，迁移时问题很多，当发现这一点是，已没有多少有效剩余时间了。

编码流程的不妥，我比较喜欢从底层的库开始开发，因为一旦库测试通过，将很容易将它放到不同的表示层下。但如果库没有测试成功，将导致整个项目没有任何可视化模型，所以这次的项目无法提交“可运行的代码”。

实践目的的不同，我轻易不放弃锻炼的机会，事实上，有机会就一定要比以前有所突破，总是照搬以前的做法还不如就不做呢。这个前提是因为现在能完全用来学习的时间比较多，等到工作时再这样做的可能性就很小了，因此当然要抓紧机会了。不过还有一个隐藏原因，总以为自己很了不起，其实“遇到的问题数跟人的能力是成正比的”。

数据库课程设计有哪些篇四

本次课程设计，使我对《数据结构》这门课程有了更深入理解。《数据结构》是一门实践性较强课程，为了学好这门课程，必须在掌握理论知识同时，加强上机实践。

我课程设计题目是线索二叉树运算。刚开始做这个程序时候，感到完全无从下手，甚至让我觉得完成这次程序设计根本就

是不可能，于是开始查阅各种资料以及参考文献，之后便开始着手写程序，写完运行时有很多问题。特别是实现线索二叉树删除运算时很多情况没有考虑周全，经常运行出现错误，但通过同学间帮助最终基本解决问题。

在本课程设计中，我明白了理论与实际应用相结合重要性，并提高了自己组织数据及编写大型程序能力。培养了基本、良好程序设计技能以及合作能力。这次课程设计同样提高了我综合运用所学知识能力。并对vc有了更深入了解。《数据结构》是一门实践性很强课程，上机实习是对学生全面综合素质进行训练一种最基本方法，是与课堂听讲、自学和练习相辅相成、必不可少一个教学环节。

上机实习一方面能使书本上知识变“活”，起到深化理解和灵活掌握教学内容目；另一方面，上机实习是对学生软件设计综合能力训练，包括问题分析，总体结构设计，程序设计基本技能和技巧训练。此外，还有更重要一点是：机器是比任何教师更严厉检查者。因此，在“数据结构”学习过程中，必须严格按照老师要求，主动地、积极地、认真地做好每一个实验，以不断提高自己编程能力与专业素质。

通过这段时间课程设计，我认识到数据结构是一门比较难课程。需要多花时间上机练习。这次程序训练培养了我实际分析问题、编程和动手能力，使我掌握了程序设计基本技能，提高了我适应实际，实践编程能力。总的来说，这次课程设计让我获益匪浅，对数据结构也有了进一步理解和认识。

一周的课程设计结束了，在这次的课程设计中不仅检验了我所学习的知识，也培养了我如何去把握一件事情，如何去做一件事情，又如何完成一件事情的方法和技巧。在设计过程中，和同学们相互探讨，相互学习，相互监督。我学会了运筹帷幄，学会了宽容，学会了理解，也学会了做人与处世，这次课程设计对我来说受益良多。

课程设计是我们专业课程知识综合应用的实践训练，这是我们迈向职业工作前一个必不可少的过程。“千里之行始于足下”，通过这次课程设计，我深深体会到这句千古名言的真正含义。我今天认真的进行课程设计，学会脚踏实地迈开这一步，就是为明天能稳健地在社会大潮中奔跑打下坚实的基础。我这次设计的科目是数据结。

数据结构，是一门研究非数值计算的程序设计问题中计算机的操作对象（数据元素）以及它们之间的关系和运算等的学科，而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。“数据结构”在计算机科学中是一门综合性的专业基础课。数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。数据结构这一门课的内容不仅是一般程序设计（特别是非数值性程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序的重要基础。通过这次模具设计，我在多方面都有所提高。

在界面设置中使用函数调用while其中文本显示颜色和背景颜色都可以任意按照自己的喜好，任意改变，但改变的时候必须采用标准英文大写，同时在制作显示菜单的窗口，大小根据菜单条数设计。最后采用printf输出程序设计界面。

这次的程序软件基本上运行成功，可以简单的建立链式循环链表，并进行输出，及循环语句的运用和选择语句的控制。由于时间和知识上的限制，使得程序规模相对较小，即功能还不很全面，应用也不很普遍。原来c语言可是涉及很多知识，而不是枯燥无聊的简单的代码部分而已，利用c语言方面的知识，我们可以设计出更完善的软件。

通过这次的课程设计，更是让我深刻认识到自己在学习中的不足，同时也找到了克服这些不足的方法，这也是一笔很大的资源。在以后的时间中，我们应该利用更多的时间去上机实验，加强自学的的能力，多编写程序，相信不久后我们的编程能力都会有很大的提高能设计出更多的更有创新的作品。

数据库课程设计有哪些篇五

由于平时接触的都是一些私人项目，这些项目大都是一些类库，其他人的交流相对可以忽略不计，因此也就不考虑规范化的文档。实际上从学习的经历来看，我们接触的知识体系都是属于比较老或比较传统的，与现在发展迅速的it行业相比很多情况已不再适用，尤其是当开源模式逐渐走近开发者后更是如此。

虽然这次是一个数据库课程设计，由于本人在选择项目的时候是本着对自己有实际应用价值的角度考虑的，所以其中也涉及到一些数据库以外的设计。对于ooa/ood的开发模式有时不免要提出一些疑问。uml是设计阶段的工具，而它基本涵盖了软件设计的方方面面，也就是说按照这一软件工程的正常流程，在动手写第一句代码之前，开发人员已经非常熟悉软件产品了，这对于相当有经验的架构师一类人说可能会很容易，但是我们作为学生，连足够的编码经验都没有，却首先被教授并要求先ooa再oop。这样直接导致的。问题就是文档与编码对不上号，在修改代码的时候基本不会再去审查文档和先前的分析。甚至根本就是现有代码再有文档，即便是这种情况，代码与文档还是不对应。不可否认，在传统软件工程的详细设计之前的项目过程中还是有很多利于项目开发的部分的。所以我就一直在寻找适合我针对探究型项目的开发模式，这次的项目也算是一次尝试，当然这个过程并不会太短。

回到数据库设计上了，这次的数据库设计我是严格按照数据库建模的步骤来进行的，老实说我并没有感觉这样的流程对开发带来多大的帮助，反倒是觉得将思维转化为图表很浪费时间。总体上来说这次的项目也不是很大，而且在数据库的设计上比较保守，也就是说实际上数据库设计还可以再完善完善的。随着我对计算机领域的拓宽和加深，我也会静下心来思考在接触计算机之前的行为，很多次我能深切感觉到，

其实我的大脑(未于别人比较)本身就是在使用一种更接近关系数据库的方式来记忆，所以我很可恨自然的设计出符合三范式的表结构来，即便我不知道这些范式的确切含义。可能就像范式不太容易用通俗易懂的方式解释一样，在让工具用图标表述我的思维时费了一番力气。

从我作为项目的提出人和实现者来看，这是个失败的项目，结合几次教学项目的实践，发现这也已经不是第一次了。主观原因占多数，比如，尝试新的开发方式，根据设计花了太多的时间来抽象出公用的库而忽略业务逻辑。就这次项目而言，失败的原因有以下几点：

使用了新的开发环境(vim)[]这是首次在脱离高级ide的情况下编码。

使用了新的开发语言(python[]actionscript3)[]因为我一直比较喜欢学以致用，而且这样的数据驱动型软件的整套自实现的库都已经完成了，但是由于语言本身的差异，迁移时问题很多，当发现这一点是，已没有多少有效剩余时间了。编码流程的不妥，我比较喜欢从底层的库开始开发，因为一旦库测试通过，将很容易将它放到不同的表示层下。但如果库没有测试成功，将导致整个项目没有任何可视化模型，所以这次的项目无法提交可运行的代码。

实践目的的不同，我轻易不放弃锻炼的机会，事实上，有机会就一定要比以前有所突破，总是照搬以前的做法还不如就不做呢。这个前提是因为现在能完全用来学习的时间比较多，等到工作时再这样做的可能性就很小了，因此当然要抓紧机会了。不过还有一个隐藏原因，总以为自己很了不起，其实遇到的问题数跟人的能力是成正比的。