 未标题-1

**综合设计实验报告**

**题 目: 人事管理系统**

**院系名称： 信息科学与工程学院**

**专业班级： 空间信息与数字技术2001班**

**指导教师： 李 滨**

|  |  |
| --- | --- |
| 原培轩 | 曹睿明 |
| 黄超 | 周海鑫 |
| 程蒋睿 | 霍强森 |

**成 员：**

**2022年12月20日**

**目 录**

[0. 小组分工 1](#_Toc22202)

[1. 设计要求 1](#_Toc18020)

[1.1 数据要求 1](#_Toc23942)

[1.2 功能要求 1](#_Toc2667)

[1.2.1 系统功能的基本要求 1](#_Toc32457)

[1.2.2提高性要求 2](#_Toc31417)

[2. 概念结构设计 2](#_Toc23870)

[2.1 局部E-R模型 2](#_Toc20068)

[2.2 全局E-R模型 3](#_Toc7614)

[3. 逻辑结构设计 3](#_Toc16508)

[3.1 初始关系模式设计（E-R图转换关系模式） 3](#_Toc31316)

[3.2 关系模式规范化 4](#_Toc10569)

[4. 物理设计 6](#_Toc11998)

[4.1 设定的原则 6](#_Toc25139)

[4.2 根据以上的原则，我们进行了如表1的判断 7](#_Toc21492)

[5. 数据库实施与开发 8](#_Toc26081)

[5.1 数据库实施 8](#_Toc6970)

[5.2 模块开发（选做） 13](#_Toc6081)

[6. 总结 22](#_Toc29345)

[7. 组员个人总结（见附件） 23](#_Toc15892)

[附件1： 24](#_Toc7988)

[附件2： 25](#_Toc27224)

[附件3： 26](#_Toc21137)

[附件4： 27](#_Toc15718)

[附件5： 28](#_Toc20761)

[附件6： 29](#_Toc7519)

# 小组分工

黄超、周海鑫：数据库的实施（建表，加数据，写数据库索引语句)

原培轩：C#与数据库的连接以及C#的界面设计等相关代码

曹睿明、霍强森：实施增删改查功能

程蒋睿：组织润滑组员，明确实验目标，实验报告以及结构设计

# 设计要求

## 数据要求

① 员工基本信息表；

② 员工婚姻情况表，反映员工的配偶信息；

③ 员工学历信息表，反映员工的学历、专业、毕业时间、学校、外语情况等；

④ 企业工作岗位表；

⑤ 企业部门信息表。

## 功能要求

### 1.2.1 系统功能的基本要求

① 员工各种信息的输入，包括员工的基本信息、学历信息、婚姻状况信息、职称等；

② 员工各种信息的修改；

③ 对于转出、辞职、辞退、退休员工信息的删除；

④ 按照一定的条件，查询、统计符合条件的员工信息；至少应该包括每个员工详细信息的查询、按婚姻状况查询、按学历查询、按工作岗位查询等，至少应该包括按学历、婚姻状况、岗位、参加工作时间等统计各自的员工信息；

⑤ 对查询、统计的结果打印输出。

### 1.2.2提高性要求

① 结合实际业务需求设计并实现有关时空位置数据的字段、表格及功能。

② 基于构建的空间数据库，设计并编程实现一个空间数据库应用系统（至少包括数据库访问、时空数据可视化等功能）。

# 概念结构设计

## 局部E-R模型

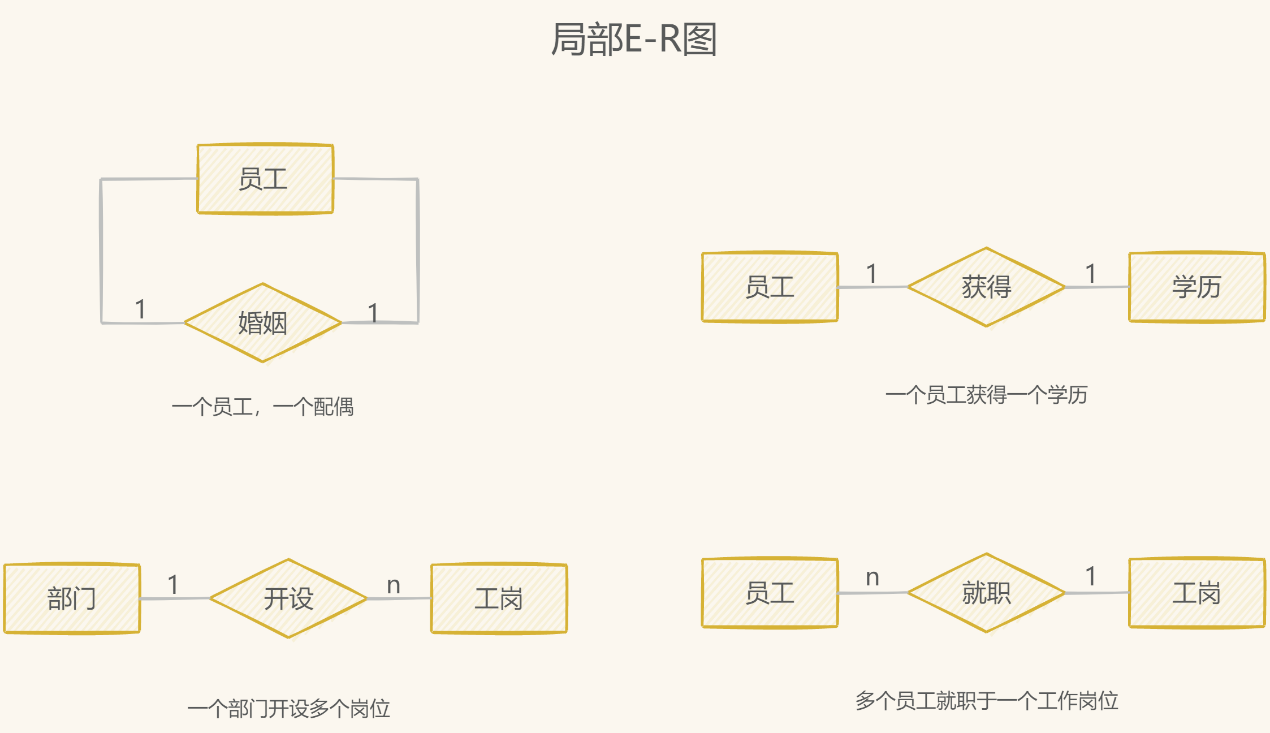


图1 局部 E-R模型图示

## 全局E-R模型

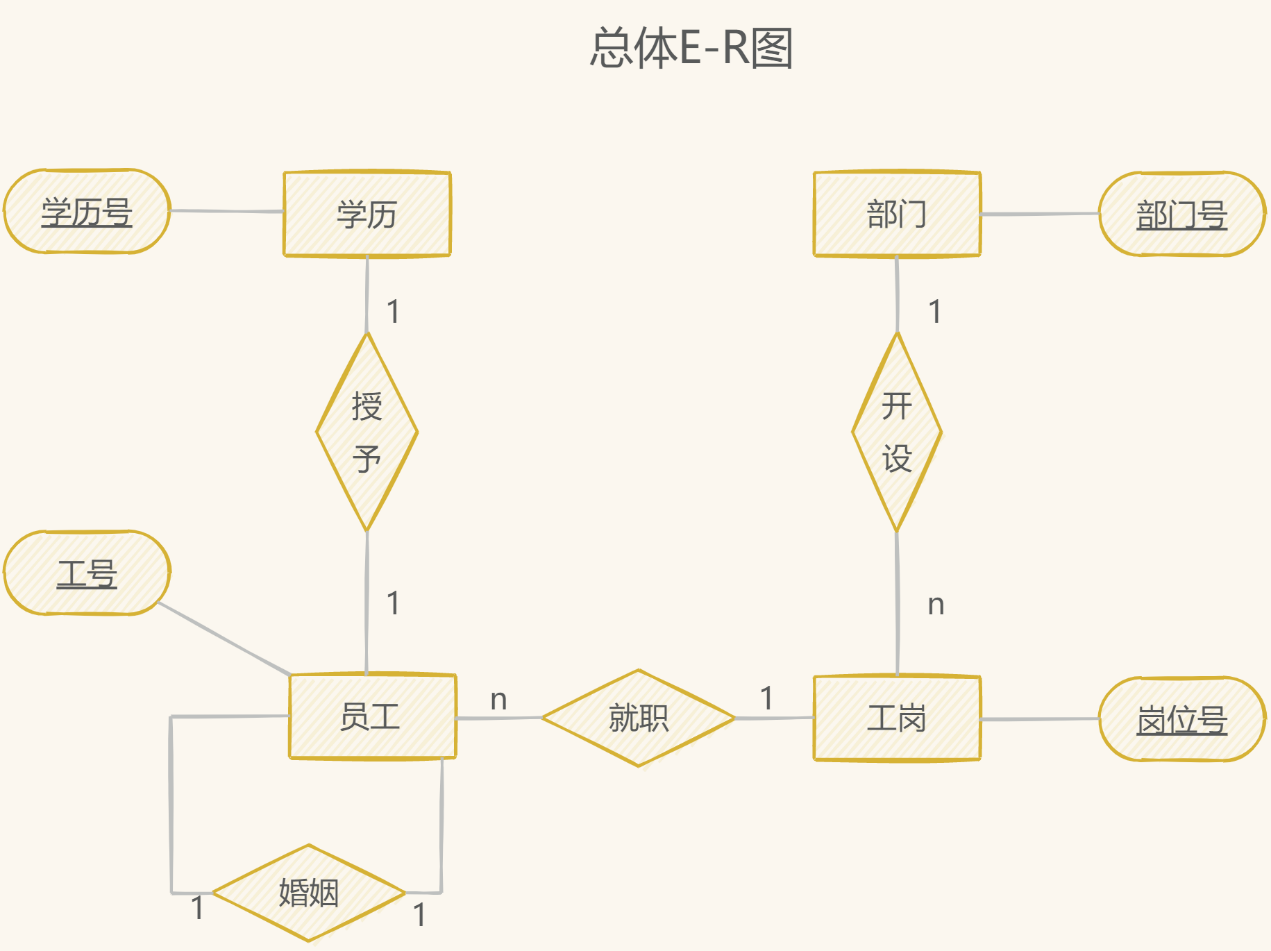


图2 全局 E-R模型图示

# 逻辑结构设计

逻辑结构设计是将概念结构设计阶段完成的概念模型，转换成能被选定的数据库管理系统(DBMS)支持的数据模型。这里主要将E-R模型转换为关系模式。需要具体说明把原始数据进行分解、合并后重新组织起来的数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构和文件结构、所建立的各个文件之间的相互关系，形成本数据库的数据库管理员视图。一般的逻辑设计分为以下3步：初始关系模式设计、关系模式规范化、模式的评价与改进。

## 初始关系模式设计（E-R图转换关系模式）

本实验的E-R关系模式转换图如图2所示

## 关系模式规范化

### 1、概念

数据库逻辑设计的结果不是唯一的。为了进一步提高数据库应用系统的性能，还应该根据应用需要适当修改、调整数据模型的结构，这就是数据模型的优化。关系数据模型的优化通常以规范化理论为指导，将关系模式规范化，使之达到较高的范式是设计好关系模式的唯一途径，否则，设计的关系数据库会产生一系列的问题。

一个“好”的关系模式应当不会发生插入和删除异常，冗余度要尽可能少。对于存在问题的关系模式，可以通过模式分解的方法使之规范化。“分解”是解决冗余的主要方法，也是规范化的一条原则，“关系模式有冗余问题，就分解它”。

函数依赖基本概念：

规范化是指用形式更为简洁、结构更加规范的关系模式取代原有关系模式的过程。

关系模式必须满足一定的完整性约束条件以达到现实世界对数据的要求。完整性约束条件主要包括以下两个方面：

（1）对属性取值范围的限定

属性值间的相互联系(主要体现在值的相等与否)，这种联系称为数据依赖。

客观世界的事物间存在着错综复杂的联系，实体间的联系有两类：一类是实体与实体之间的联系；另一类是实体内部各属性间的联系。而属性间的联系可分为3类：

①一对一联系(1:1)。设X和Y是关系R的两个属性(集)。如果对于X中的任一具体值，Y中至多有一值与之对应；反之亦然，则称X、Y两属性间是一对一联系。

②一对多联系(1:n)。设X和Y是关系R的两个属性(集)。如果对于X中的任意具体值，Y中至多有一个值与之对应，而Y中的一个值却可以在X中的n个值(n≥0)相对应，则称Y对X是一对多联系。

③多对多联系(m:n)。设X和Y是关系R的两个属性(集)。如果对于X中的任意具体值，Y中有m个值(m≥0)与之对应，而Y中的一个值也可以和X中的n个值(n≥0)相对应，则称Y对X是多对多联系。

数据依赖是指通过一个关系中属性间值的相等与否体现出来的数据间的相互关系，是现实世界属性间相互联系的抽象，是数据内在的性质。数据依赖共有3种：

函数依赖(Functional Dependency, FD)

多值依赖(Multivalued Dependency, MVD)

连接依赖(Join Dependency, JD)

其中最重要的是函数依赖和多值依赖。

在数据依赖中，函数依赖是最基本、最重要的一种依赖，它是属性之间的一种联系，假设给定一个属性的值，就可以唯一确定(查找到)另一个属性的值。这种唯一性并非指只有一个记录，而是指任何记录。

设有关系模式R(U)，X和Y均为U={A1, A2, …, An}的子集，r是R的任一具体关系，r中不可能存在两个元组在X上的属性值相等，而在Y上的属性值不等(也就是说，如果对于r中的任意两个元组t和s，只要有t[X]=s[X]，就有t[Y]=s[Y])，则称X函数决定Y，或称Y函数依赖于X，记作X→Y，其中X叫做决定因素(Determinant)，Y叫做依赖因素(Dependent)。

属性间联系决定函数依赖。

1:1联系。如果两属性集X、Y之间是1:1联系，则存在函数依赖X←→Y。

1:n联系。如果两属性集X、Y之间是n:1联系，则存在函数依赖X→Y。

m:n联系。如果两属性集X、Y之间是m:n联系，则不存在函数依赖。

识别函数依赖是理解数据语义的一个组成部分，依赖是关于现实世界的断言，它不能被证明，决定关系模式中函数依赖的唯一方法是仔细考察属性的含义。

### 2、范式

利用规范化理论，使关系模式的函数依赖集满足特定的要求，满足特定要求的关系模式称为范式(Normal Form)。关系按其规范化程度从低到高可分为5级范式，分别称为1NF、2NF、3NF(BCNF)、4NF、5NF。规范化程度较高者必是较低者的子集。一个低一级范式的关系模式，通过模式分解可以转换成若干个高一级范式的关系模式的集合，这个过程称为规范化。

（1）第一范式(1NF)

如果关系模式R中不包含多值属性(每个属性必须是不可分的数据项)，则R满足第一范式(First Normal Form)，记作R∈1NF 。1NF是规范化的最低要求，是关系模式要遵循的最基本的范式，不满足1NF的关系是非规范化的关系。关系模式如果仅仅满足1NF是不够的，只有对关系模式继续规范化，使之满足更高的范式，才能得到高性能的关系模式。

（2）第二范式(2NF)

如果关系模式R(U, F)∈1NF，且R中的每个非主属性完全函数依赖于R的某个候选码，则R满足第二范式(Second Normal Form)，记作R∈2NF。不满足2NF的关系模式，会产生以下几个问题：插入异常、删除异常、更新异常。解决的办法是用投影分解把关系模式分解为多个关系模式。投影分解是把非主属性及决定因素分解出来构成新的关系，决定因素在原关系中保持，函数依赖关系相应分开转化(将关系模式中部分依赖的属性去掉，将部分依赖的属性单组组成一个新的模式)。

（3）第三范式(3NF)

如果关系模式R(U, F)∈2NF，且每个非主属性都不传递函数依赖于任何候选码，则R满足第三范式(Third Normal Form)，记作R∈3NF。解决的办法同样是投影分解。3NF是一个可用的关系模式应满足的最低范式，也就是说，一个关系模式如果不满足3NF，则实际上它是不能使用的。

## 模式的评价与改进

关系模式的规范化不是目的而是手段，数据库设计的目的是最终满足应用需求。因此，为了进一步提高数据库应用系统的性能，还应该对规范化后产生的关系模式进行评价、改进，经过反复多次的尝试和比较，最后得到优化的关系模式。

# 物理设计

## 设定的原则

①在经常需要搜索的列上创建索引;

②在主键上创建索引;

③在经常用于连接的列上创建索引；也就是在外键上创建索引;

④在经常需要根据范围进行搜索的列上创建索引(因为索引已经排序，其指定的范围是连续的);

⑤在经常需要排序的列上创建索引(因为索引已经排序，这样查询可以利用索引的排序，加快排序查询时间);

⑥在经常用在WHERE子句中的列上创建索引；

⑦对于查询中很少涉及的列,或重复值比较多的列不要建索引；

⑧对于定义为text、image和bit数据类型的列上不要建立索引。

## 根据以上的原则，我们进行了如表1的判断

|  |  |  |
| --- | --- | --- |
| 动作描述 | 使用聚集索引 | 使用非聚集索引 |
| 外键列 | 应 | 应 |
| 主键列 | 应 | 应 |
| 列经常被分组排序(order by) | 应 | 应 |
| 返回某范围内的数据 | 应 | 不应 |
| 小数目的不同值 | 应 | 不应 |
| 大数目的不同值 | 不应 | 应 |
| 频繁更新的列 | 不应 | 应 |
| 烦繁修改索引列 | 不应 | 应 |
| —个或极少不同值 | 不应 | 不应 |

表1 索引判断

①因为随着数据和应用程序活动的改变，索引的性能和效能也会改变，所以要监视查询性能。如果表上有过度地插入、修改和删除操作，索引页被分成多块就形成了索引碎片，如果索引碎片严重，那扫描索引的时间就会变长，甚至导致索引不可用，因此数据检索操作就慢下来了并且降低查询性能，返回相同的结果需要更多的IO操作，因此通过重新生成、重新组织索引来保持索引碎片最小化。

②为只读文件组或数据库上的索引使用100%填充因子，这样使得完成查询需要访问的数据页更少，相应的IO操作也更少，效率自然得到提升。

③并且根据这些原则将以下几个主要的表设置索引（其他附表是为了优化系统，将其中很多个第二范式、第三范式的表分解，并且这些表不进行聚集索引的建设，转而在VS程序设计的时间直接调用，提高系统查询的性能）如图3

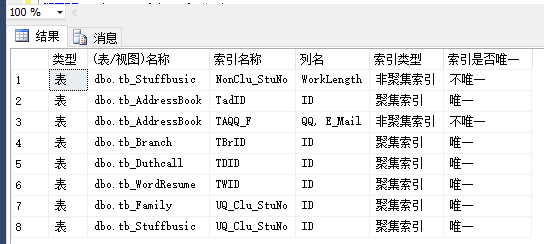


图3 几个表索引实例

# 数据库实施与开发

## 数据库实施

以表sys\_job的创建为例，该表的job\_location字段为geography类型，用于存放坐标点数据，以表示某工作岗位所处的地理位置。这部分的SQL语句如下：

USE InformationData

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(1,101,'销售员',geography::STGeomFromText('POINT(122.423401 47.827006)',4326))

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(2,101,'前台',geography::STGeomFromText('POINT(110.146687 34.547352)',4326))

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(3,101,'部门主管',geography::STGeomFromText('POINT(115.014697 23.479204)',4326))

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(4,102,'会计',geography::STGeomFromText('POINT(99.947938 22.672001)',4326))

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(5,102,'设计师',geography::STGeomFromText('POINT(86.549201 46.879246)',4326))

INSERT INTO sys\_job(job\_id,dept\_id,job\_name,job\_location)

VALUES(6,103,'工程师',geography::STGeomFromText('POINT(113.504921 35.001468)',4326))

GO

最终得到数据库InformationData及其各表结构与数据如下：

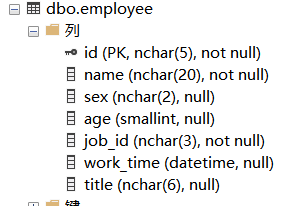


图4 员工基本信息表结构



图5 员工基本信息表数据

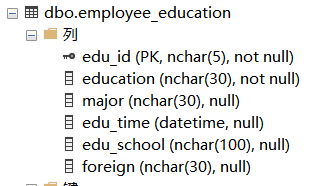


图6 员工学历信息表结构

 图7 员工学历信息表数据

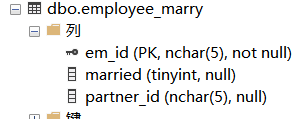


图8 员工婚姻情况表结构

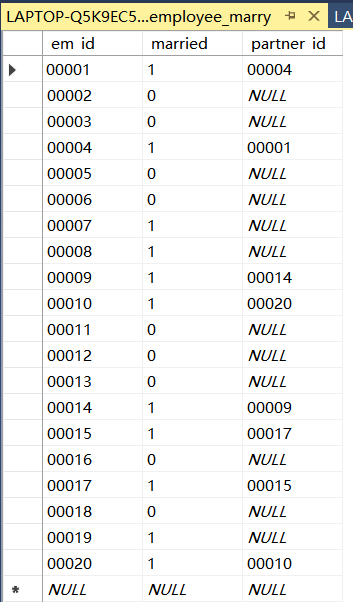


图9 员工婚姻情况表数据

企业部门信息表：

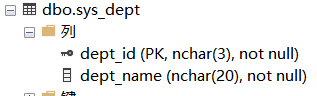
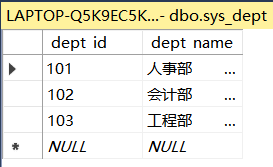
 

图10 企业部门信息表结构 图11 企业部门信息表数据

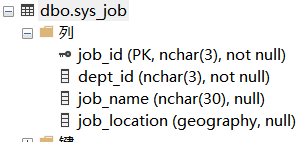


图12 企业工作岗位表结构

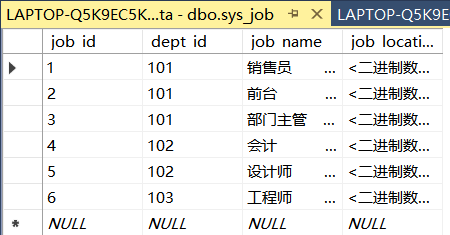


图13 企业工作岗位表数据

## 模块开发（选做）

根据建好的数据库，以C#设计可视化应用程序，其用户登录界面如下：



图14 登陆界面

其中，用户名为“user\_1”，密码为“123”。

程序主窗体MainForm如下：



图15 MainForm窗体

主窗体主要由两部分组成，上面部分为菜单栏，包含本程序主要功能，下面部分为工具栏，分别为“增、删、改、查”等功能提供快捷打开方式。

“操作”与“查看”按钮下分别实现“增、删、改、查”等功能。

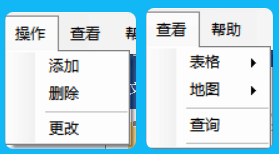


图16 MainForm窗体操作和查看功能示例

开始实行本程序数据库有关功能之前，首先应连接数据库。点击“文件”下的“打开数据库”以进行数据库连接。

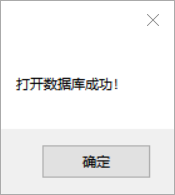


图17 MainForm窗体文件功能示例

“添加”功能可同时实现员工基本信息、学历信息、婚姻状况以及职称等信息的自动同步添加。该功能演示如下：

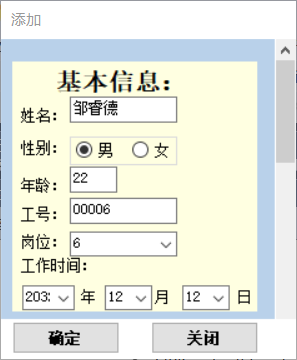
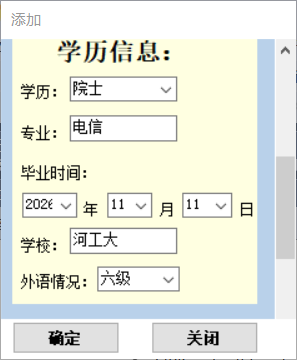
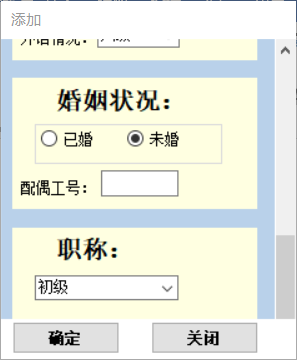








图18 添加功能演示

“删除”功能演示：

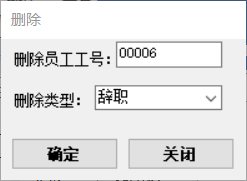




图19 “删除”功能演示

“更改”功能演示：

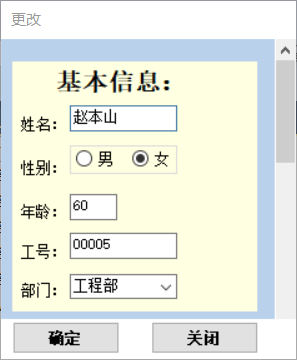




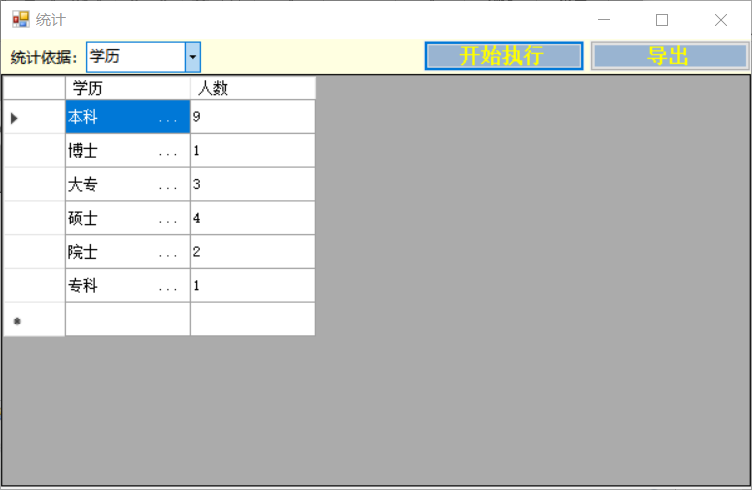
图20 “更改”功能演示

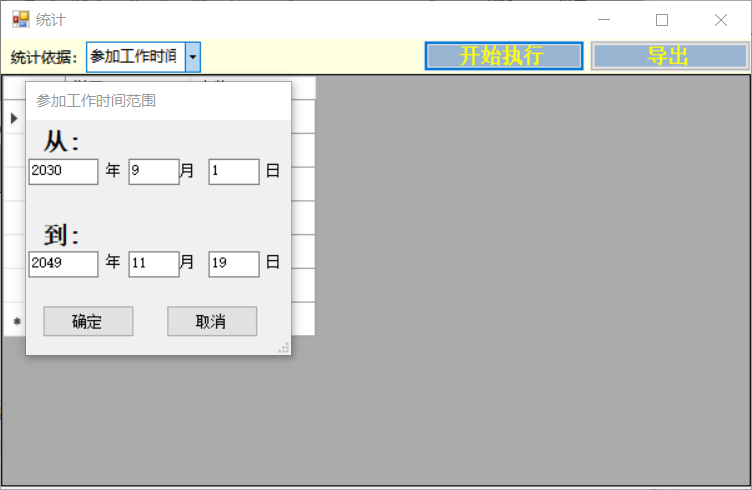
“查询”功能演示：



图21 “更改”功能演示

“统计”功能位于“查看”下。该功能演示如下：





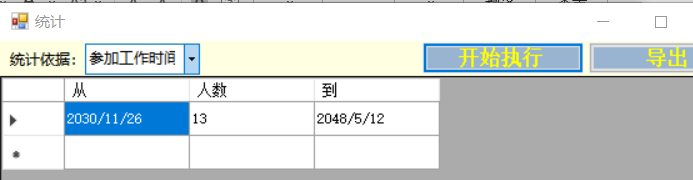
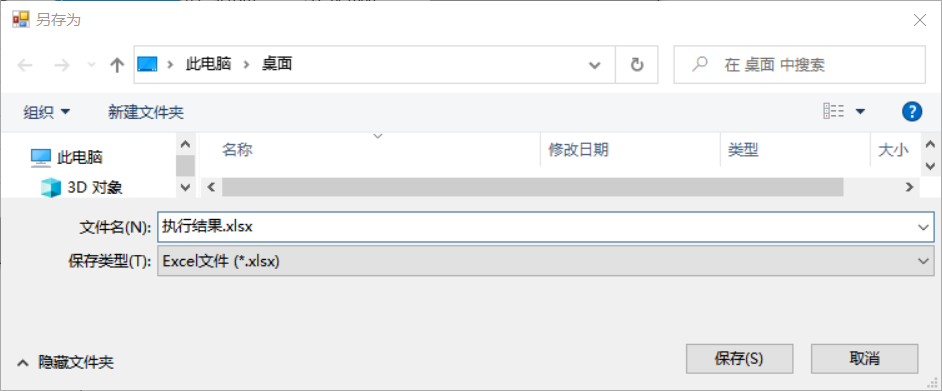


图22 “统计”功能演示

“导出”功能演示（以“查询”功能演示结果为例）：



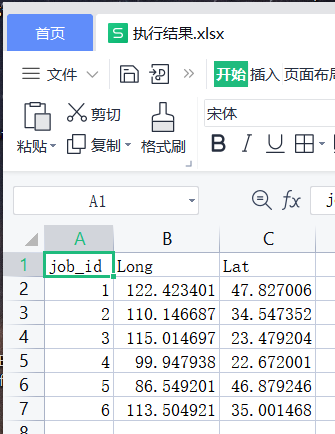
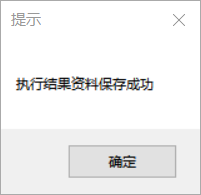
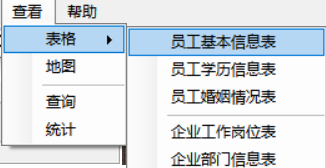


图23 “导出”功能演示

员工详细信息查询：









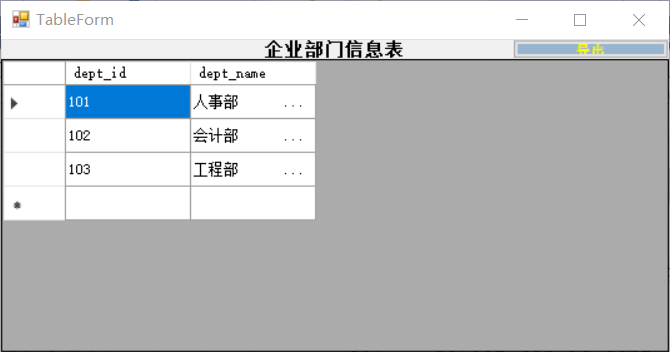
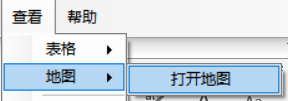


图24 员工详细查询

基于高德地图API，我们通过WinForm窗体WebBrowser控件实现了地理坐标的时空数据可视化。其中，内嵌web应用以JavaScript语言实现。空间数据库功能模块展示如下：





# 总结

通过本次实验，我们进行了从数据库和用户界面的设计到实施的各项工作。在此过程中，我们首先从数据库的概念模型设计出发，讨论了小组成员提供的各项预案的可行性，最终敲定本次实验E-R图，并将之自顶向下地实现为物理模型，即构建出5个相应的数据库表结构。我们为这些表添加合适的数据，并注重各表之间的协同与联系。

与此同时，我们小组的另一部分人员进行用户界面的设计与构建，综合考虑多种因素，我们决定采用基于VS平台的C#语言WinForm窗体应用开发模式。本用户界面程序灵感来自于遥感专业软件ENVI。

本实验难点在于数据库与用户界面之间数据的对接与小组负责各部组员之间的沟通协作。我们在实验过程中，特别是软件的开发部分，体会到软件工程的部分设计思想，即以需求为驱动的喷泉模型相关思想。喷泉模型主要用于面向对象的软件项目，其中包括许多程序复用，我们在实现各类信息表的显示功能时，也基于TableForm窗体实现了部分代码的复用，这既简化了部分工作量，又降低了在繁复的数据对接时出现错误的可能。

在实现空间数据库部分功能时，我们遇到一些问题。首先是在数据库部分，我们本来使用普通的数据类型float用以存储地理坐标数据，甚至在分别以经度字段和纬度字段储存地理坐标数据，和以一个数组的形式储存此数据这两种方案间游移，这显然不符合空间数据库设计思想，也徒增设计困难与复杂性。SQL Server已经提供了可供进行空间数据库设计的相关数据结构与方法，我们应该利用起来，使用geography数据类型来记载点坐标比较合理，在给这种数据类型添加坐标数据时，我们尝试使用SSMS交互方式直接添加，然而这种方式不具备可操作性，这是因为此数据类型是以二进制形式存储坐标数据，以SQL命令行模式为其添加数据较为妥当。

为了将记录的各工作岗位的地理坐标（代表某岗位所处地理位置）可视化至地图，我们查阅各种资料，尝试可能的方案，最终发现高德地图有相关的地图API，它是用JavaScript语言实现的WebApp，我们使用了一些代码将它嵌入到桌面主App中，并运用一些相应的前端开发知识，对WebApp做了一定的改造，以满足实验需求。

本次实验的综合性十分强，所用到的知识广泛且复杂，只掌握关于数据库相关内容是远远不够的。我们在进行实验时也是在现学现用，复习到了很多其它方面的知识。最终，我们基本实现了实验所要求的数据库结构及程序功能模块，对各模块的重要功能，以及关键的步骤和流程进行演示与截图记录。

# 组员个人总结（见附件）

附件1：

**个人总结**

班级：空信2001 学号：201916060105 姓名：黄超

一个简单的系统也需要相当庞大的数据库来组成，这样我们就有特别多的数据库表，而且表和表之间的依赖关系还挺复杂的，实体类之间的关系众多，设计起来有一定的难度。数据库的另外一个问题是数据表的属性，数据表属性的实际完全是依赖于我们刚开始做的需求分析文档，如果需求分析不完整的话，就会造成数据库表属性的缺失，从用户的角度看，就会觉得我们的系统提供的数据不够全面，信息量小。

在课堂上老师所讲的数据库索引语句的概念以及用法到了这时就可以让所谓的想法变成了实践。起初上课时即使老师说得再详细没有自己亲手多操作几次还真的体会不到索引的相关用法以及得到的结果是能给人怎样的惊喜，在实际操作中有多次的翻看老师所发的ppt作为指导书，再从网上查找一些知识来丰富相关用法以及作用，也通过此次学习让我对数据库索引的语句加深了印象，补足了当初在上理论课时对这些知识点的部分迷茫以及解除了对它们的一些误解。

附件2：

**个人总结**

班级：空信2001 学号：202012010101 姓名：原培轩

本次课程设计中，我承担C#与数据库的连接以及C#的界面设计相关代码的编写，主要运用DataGridView控件展示从数据库获取的各种数据，这个控件具有比较强大的适应性，也易于实现数据的导出。我考虑到输出数据多为表结构，故采用Excel容纳导出的数据，以供后期将之打印出来。为了将其他同学做出来的数据库用C#显示出来，我回顾了C#相关语法，并借鉴了我们GIS开发课的相关经验。

我的工作在整个实验中起到了承上启下与沟通全组各部分的作用，由于多采用线上沟通方式与其他组员进行联系，我时常感到沟通不畅，做出来的程序与其他人的多有冲突，比如在与其他同学的程序进行对接时，会出现命名冲突等问题，所以加强组员间的沟通交流，以及增强代码规范意识，对于大型项目的开发是至关重要的。在涉及到与数据库的连接部分，有时为了能够在程序中更好地展示数据，不得不优化一些表结构，这又涉及到新旧版本之间的冲突，我学习并掌握了为数据库生成脚本的方法，通过反复试验与摸索，解决诸如未给数据库的用户生成脚本，从而导致的C#程序界面连接数据库的错误等。遇到困难时，我一般选择上网搜寻相关解决方案或思路，可网上资源有限，所以时常受技术力的限制，不得不放弃一些功能及方案。

通过本次实验，我的SQL相关语法语句和C#代码能力得到了锻炼，收获颇丰。

附件3：

**个人总结**

班级：空信2001 学号：202012020111 姓名：曹睿明

通过本次实验让我获益匪浅，学到了许多的知识，也让我更为深刻的了解到了团队合作的重要性这次我的任务是增删改查和实现与时空有关的数据功能。

增删改查代码本身并不难实现实在不行可以上网跟教程，最重要的是和队友沟通，问黄超去了解他们写的数据库增删改查的索引思路与方法充分了解他们的想法，再者就是和原培轩交流了解他给我们的增删改查功能留的窗口与控件，尽量去兼顾美观与功能以免出现交流不充分导致有些控件与窗口变成摆设白白浪费了原培轩的努力。

在编写过程中因为霍强森对C#的一些不熟悉可能导致了一些小麻烦但在程蒋睿的交流下他一改之前的粗心熬夜在网上找教程和问我们问题出在那，那时候我也在熬夜在床上刷手机很乐意起床和他一起研究，时空数据有关的方面我和霍强森讨论了一下就要求数据库组再建一个表格存储人员工作地方的经纬度，再实现经纬度的点再地图上显示的功能，总的来说老师留的问题不难，难的是与队员的沟通与了解，实现他们想法中的奇思妙想更正他们的错误，这一点对我和霍强森也是如此。

附件4：

**个人总结**

班级：空信2001 学号：202016060102 姓名：周海鑫

本次的综合设计实验我们小组最终选择的题目是人事管理系统，该系统的目标是能够实现员工各种基本信息的输入，可以对输入的信息进行修改、删除等一系列的操作，还需要能够通过各项条件进行查询，精确的将目标信息查询，统计出来。

本次的综合设计实验是一项比较复杂的实验，尽管在这之前已经做过了空间数据库相关的几个实验，也对所需要的一些软件比如SuperMap、SQL Server等有了一定的认识学会了基本的应用，但最初看到实验要求还是有点不知从何下手，最终在小组讨论以及网上搜索资料之后，确定了本次综合设计实验的具体方向，并且进行分工，明确了每个人的任务。

在本次的实验当中我和组内的另一位同学所负责的主要部分是数据库的实施，主要的任务就是创建表，以及添加数据，还有写数据库索引语句，这些内容在之前的数据库相关实验当中也练习过，再结合着老师所发的一些教程以及网上查找的相关资料，最终完成了这一项内容。通过本次综合实验，我学到了很多，参与创建一个完整的系统相比之前零碎的实验内容更能让人产生深刻的理解，更容易将老师所讲授的内容与实际联系起来，在实验中我也发现了自身的不足，许多不大不小的错误需要花费很长的时间去调试，这表明之前的我太依赖实验教程，对于一些东西虽然做出来了却不知其所以然，在今后我也会尽力搞懂实验的每一个步骤，以及每一步的原理。

附件5：

**个人总结**

班级：空信2001 学号：202016060113 姓名：程蒋睿

经过全组人讨论本次综合实验我们制作了人事管理系统，人事管理工作是很重要的一项工作，它负责整个企业的日常人事安排，人员的人事管理等。随着企业规模变大，人事管理工作越发困难，利用数据库管理能大幅度减少人事管理工作，提高效率。

在讨论过程中我明确了实验目标，首先我们需要了解企业有那些部门，这些部门的部门号是多少，它开设了哪些岗位，这些岗位的岗位号是多少，其次我们需要了解员工的学历和学历号，员工的工号和婚姻状况。最后通过目标建立E-R模型，建立数据库同时在数据库中建立相关的表。在实验期间我了解各个任务的进度，组织润滑组员，调节任务进度。实验期间组织讨论，确定使用C#设计登录界面和可视化应用程序，确定了基于高德地图实现了地理坐标的时空数据可视化。在实验过程中我进行了实验报告以及结构的设计。

通过本次实验我明白了系统设计的困难，需要清晰的思路，考虑很多相关的事宜。我们小组制作的人事管理系统还不够完善，如员工工资的计算，人事的变化等。这次试验我了解到我在很多方面还有欠缺，在今后我会炉里学习相关专业知识，为之后的设计做好基础。

附件6：

**个人总结**

班级：空信2001 学号：202016060115 姓名：霍强森

通过本次实验让我获益匪浅，学到了许多的知识，也让我更为深刻的了解到了团队合作的重要性。在本次实验中我主要做的是实现程序中的增添，删除，修改和查询信息的这一方面内容。因为我个人在编程这一方面的能力不是很强，所以我在这一方面遇到的问题不少。

首先由于我很少将代码与数据库进行连接并设计代码，所以在刚开始的时候我无从下手，一点也不知道怎么写，但经过查找资料和向队员寻求帮助后，我慢慢的了解到了如何使用C#的代码实现数据库中信息的增添，删除，修改和查询，也慢慢地掌握了一些方法和技术。最后，在编写代码中也由于我的粗心，导致命名的变量的名称的大小写发生错误导致变量在运行使会发生错误，但经过查找修改后就可以正常运行了。

在团队合作中，原培轩和曹睿明两位同学对我的帮助甚巨，所以我也十分的感谢他们对我的帮助，同时也感谢黄超和周海鑫同学设计并建造数据库和程蒋睿同学帮助我们查找资料、设计实验报告的排版和撰写实验报告的内容。

最后的最后我很感谢李老师给我们这一次动手设计程序的实践，这使我将所学知识和动手实践联系到了一起，极大的丰富了我的课余生活，使我的动手能力和阅历都有了一定的增长。