上海理工大学

实验课程设计报告

**设计名称：** **数据库课程设计**

**序号**  **姓 名：**  **学 号：**

**序号**  **姓 名：**  **学 号：**

**序号**  **姓 名：**  **学 号：**

**专业班级：**

**系 （院）： 管理学院**

**设计时间：** **2017~2018学年第二学期**

**成绩：**

目录

[一． 研究背景 3](#_Toc77103297)

[1.1商业前景 3](#_Toc77103298)

[1.2经营现象 3](#_Toc77103299)

[1.3可行性分析定义 3](#_Toc77103300)

[1.4可行性分析目的与意义 3](#_Toc77103301)

[1.5技术可行性分析 4](#_Toc77103302)

[二．功能及需求分析 4](#_Toc77103303)

[2.1需求分析 4](#_Toc77103304)

[2.2数据流图 4](#_Toc77103305)

[2.2数据字典 6](#_Toc77103306)

[三．数据库概念结构设计 7](#_Toc77103307)

[3.1数据库中的实体 7](#_Toc77103308)

[3.2 E-R图 7](#_Toc77103309)

[四.数据库逻辑结构设计 8](#_Toc77103310)

[4.1关系模型的设计 8](#_Toc77103311)

[4.2视图设计 9](#_Toc77103312)

[五.数据库物理结构设计 9](#_Toc77103313)

[5.1索引的创建 9](#_Toc77103314)

[5.2经常作为查询条件的属性 10](#_Toc77103315)

[5.3经常作为聚集函数参数的属性 10](#_Toc77103316)

[5.4系统的物理结构评价 10](#_Toc77103317)

[六．软件结构设计 10](#_Toc77103318)

[6.1编写目的 10](#_Toc77103319)

[6.2术语定义 11](#_Toc77103320)

[6.3设计目标 11](#_Toc77103321)

[6.4运行环境 11](#_Toc77103322)

[6.5条件限制 11](#_Toc77103323)

[6.6详细设计的目的 11](#_Toc77103324)

[七．系统实现 11](#_Toc77103325)

[八．数据库设计实现代码 21](#_Toc77103326)

[九．心得体会与总结 25](#_Toc77103327)

# 研究背景

## 1.1商业前景

中国面包烘焙市场到底有多大?据资料统计目前我国面包产量为160万吨;而德国面包消费量为692万吨，人均84kg;在欧洲国家中居最高;最低的为意大利，年人均消费量50kg;日本面包年消费量127万吨，人均10kg;我国台湾省面包年消费量约为20万吨，人均约9kg;而我国内地估算最多为年人均1.2kg，如将面包产量仅按5亿城镇人口消费来推算，也仅为3.2kg。由此可见，中国面包消费还属于发展中国家，从饮食习惯相近的日本和我国台湾省发展历程来看，预测我国内地面包的产量在2010年将达到480万吨。

在亚洲国家烘焙业发展最成熟的日本，光是一家知名面包连锁店品牌yamazaki山崎面包的年营业额就高达7000亿日币，而类似yamazaki的面包连锁店在日本就有近300家，市场之大，由此可见。而日本约有1亿2000多万人口，不到中国人口的1/10，若将日本经验移到中国，若中国面包烘焙市场发展到像日本一样成熟的阶段，市场无疑会获利多多!

其实，这样的发展不无可能，因为中国商店发展向来就和日本有着相近的脚步，烘焙业也是如此，所以未来中国面包烘焙业的荣景可期。

## 1.2经营现象

面包屋经营最重要的是产品质量。首先面包要新鲜，一般要求面包出炉到消费不超过6小时，尽量不卖隔夜的东西;这也决定了配送半径的控制，决定了其中央工厂、区域经营的特点;其次，需要一流点心师主持产品工艺制作及培训，才能保证产品品质、口感上乘一流;其实面包产品质量容易控制，只要标准统一，配方固定，标准作业，经营更为简单方便。

其次要有自己的主导产品。从目前情况看，整个产品质量的档次不高。一方面因为国内居民不把面包当主食，影响了面包屋的生意，面包屋也不把产品质量看得十分重要;另一方面产品的整体档次不高又影响了人们把面包当主食来用。

## 1.3可行性分析定义

可行性分析是通过对项目的市场需求、资源供应、建设规模、工艺路线、设备选型、环境影响、资金筹措、盈利能力等方面的研究，从技术、经济、工程等角度对项目进行调查研究和分析比较，并对项目建成以后可能取得的财务、经济效益及社会环境影响进行科学预测，为项目决策提供公正、可靠、科学的软件咨询意见。主要从经济、技术、社会环境等方面分析所给出的解决方案是否可行，当解决方案可行并有一定的经济效益和社会效益是才开始真正的基于计算机的系统的开发。

## 1.4可行性分析目的与意义

在现代软件技术下, 如果不考虑“成本”、“进度”、可获取的开发资源的限制, 任何软件问题都是有解的。而这些限制是客观存在、不可回避的。

“可行性研究”的目的在于针对一个确定的系统目标并确认了其“初步可行性”的软件问题, 在对软件问题进行总体性的需求分析、由此确定软件“高层逻辑模型”的基础上, 探寻总体性的、面向问题关键性技术需求的、可同时满足“技术可行性”、“经济可行性”、“实践可行性”要求的“软件开发方案”。

实践可知, 当一个软件的“高层逻辑模型”被正确地确定后, 人们对于软件问题的总体特性就有了准确的把握, 对软件问题的关键性技术需求有着系统的了解。面向各关键性技术需求, 可有着不同的技术对策。

通过“可行性研究”, 可以建立起对软件问题的总体性把握, 这是确保软件开发成功的关键;通过“可行性研究”, 软件的“高层逻辑模型”被确定, 软件定义问题由此被总体确立, 由此为通过工程化的、面向软件问题细节的“需求分析”提供了直接的工作基础;通过“可行性研究”, 可把握软件问题的关键性技术需求, 为确定软件的“系统开发方案”提供支持, 为软件的设计实现提供了总体约束;通过“可行性研究”, 可对各备选“系统开发方案”的开发成本与进度进行较准确的估算、开发资源需求有着系统了解, 为软件的设计实现确立基础。

## 1.5技术可行性分析

根据已经设想出的功能，经过实现方法的思考，已经基本可以确定其可行性，并不断发掘新的功能，使得软件更有趣，更贴合用户群体。

# 二．功能及需求分析

## 2.1需求分析

面包店为了提高销售量，需要开发一款面包销售管理系统。具体功能包括，店员身份信息登录，面包库存管理，面包销售管理，销售订单管理，订单打印等功能。方便了员工记录库存和订单信息，提高动作效率。

## 2.2数据流图

图片包含 图示

描述已自动生成

图形用户界面, 图示, 应用程序

描述已自动生成

图形用户界面, 图示

描述已自动生成

手机屏幕的截图

描述已自动生成

## 2.2数据字典

（1）员工信息表User

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 默认值 | | 允许为空 | | 自动递增 | | 备注 | |
| 编号 | | varchar | |  | | 否 | | 是 | | 主键 | |
| 密码 | | varchar | | NULL | | 是 | |  | |  | |

（2）面包信息表info

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 默认值 | | 允许为空 | | 自动递增 | | 备注 | |
| 面包编号 | | int | |  | | 否 | | 是 | | 主键 | |
| 面包名称 | | varchar | | NULL | | 是 | |  | |  | |
| 库存数量 | | int | | NULL | | 是 | |  | |  | |
| 单价 | | money | | NULL | | 是 | |  | |  | |

（3）订单信息表info

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 默认值 | | 允许为空 | | 自动递增 | | 备注 | |
| 订单编号 | | int | |  | | 否 | | 是 | | 主键 | |
| 面包名称 | | varchar | | NULL | | 是 | |  | |  | |
| 销售数量 | | int | | NULL | | 是 | |  | |  | |
| 开单时间 | | Datetime | | NULL | | 是 | |  | |  | |
| 开单员工号 | | varchar | | NULL | | 是 | |  | |  | |
| 总价 | | Money | | NULL | | 是 | |  | |  | |

# 三．数据库概念结构设计

## 3.1数据库中的实体

（1）员工信息表User

（2）面包信息表info

（3）订单信息表info

## 3.2 E-R图

日程表

描述已自动生成

卡通人物

中度可信度描述已自动生成

图示

描述已自动生成

图示

描述已自动生成

# 四.数据库逻辑结构设计

## 4.1关系模型的设计

员工信息表（账号，密码）

面包信息表（编号，面包名称，库存数，单价）

订单信息表（订单号，商品名，数量，开单时间，开单员工号，总价）

## 4.2视图设计

（1）员工登录界面视图

创建员工登录验证信息子模式，用户需要用到登陆验证子模式普通员工信息表（账号，密码）。子模式LOGIN对应的数据来自普通员工信息表。

（2）员工维护商品信息视图

创建员工维护商品信息子模式，员工需要用到员工维护商品信息子模式面包信息表（编号，面包名称，库存数，单价）。子模式Apply对应的数据来自面包信息表。

（3）员工新增销售订单视图

创建员工新增销售订单子模式，员工根据面包信息新增到订单信息表。子模式Add对应的数据来自面包信息表和员工信息表。

# 五.数据库物理结构设计

## 5.1索引的创建

创建索引可以大大提高系统的性能。

第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。

第二，可以大大加快数据的检索速度，这也是创建索引的最主要的原因。

第三，可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

第四，在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。

第五，通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。

增加索引也有许多不利的方面。

第一，创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。

第二，索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。

第三，当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。

索引是建立在数据库表中的某些列的上面。因此，在创建索引的时候，应该仔细考虑在哪些列上可以创建索引，在哪些列上不能创建索引。一般来说，应该在这些列上创建索引，例如：在经常需要搜索的列上，可以加快搜索的速度；在作为主键的列上，强制该列的唯一性和组织表中数据的排列结构；在经常用在连接的列上，这些列主要是一些外键，可以加快连接的速度；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；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；在经常使用在WHERE子句中的列上面创建索引，加快条件的判断速度。

根据对索引优缺点的分析，我们在数据库中斟酌进行索引的建立。

## 5.2经常作为查询条件的属性

本数据库的内容大部分由员工编号、商品编号来进行查询操作。

## 5.3经常作为聚集函数参数的属性

本数据库由员工编号、商品编号以及订单编号属性来作为聚集函数参数。

## 5.4系统的物理结构评价

（1）存储记录结构设计

数据库完整的体现了数据项的类型、长度以及逻辑记录到存储记录的映射。

（2）确定数据存放位置

按照设计要求，将相应的数据存放的对应的表中。还把经常同时被访问的数据组合，如账号和密码。

（3）存取路径的设计

数据库可以完美实现主存储路径，这是主要的存取方式。也可以实现辅存取路径，用辅键存取数据。如数据库系统中部门负责人信息表与请假信息表表中确认人的使用。

（4）完整性和安全性考虑

对每个数据项我们都设置了域完整性约束、实体完整性约束、参照完整性约束，并且设置了编号为主要信息存储表的主键，建立了安全性。

（5）程序设计

数据库建立完成后，C#可视化程序设计就相应开始。在窗体设计完成后，连接数据库实现具体的设想功能，并进入系统的测试阶段。

# 六．软件结构设计

## 6.1编写目的

概要设计的主要任务是设计程序的体系结构，也就是确定程序有哪些模块组成以及模块计的关系。概要设计过程首先寻找实现目标系统的各种不同的方案，需求分析阶段得到的数据流图是设想各种可能方案的基础。然后分析员从这些供选择的方案中选取若干个合理的方案，为每个合理的方案都准备一份系统流程图，列出组成系统的所有物理元素，进行成本/效益分析，并且制定实现这个方案的进度计划。分析员应该综合分析比较这些合理的方案，从中选出一个最佳方案向用户和使用部门负责人推荐。如果用户和使用部门的负责人接受了推荐的方案，分析员应该进一步为这个最佳方案设计软结构，通常，进行必要的数据库设计，确定测试要求并且是定测试计划。

## 6.2术语定义

实体─联系图（E-R图):包含实体（即数据对象)、关系和属性。作为用户与分

析员之间有效交流的工具。

流程图:由一些特定意义的图形、流程线及简要的文字说明构成，能清晰明确地

表示程序的运行过程

盒图:又称为N-S图、CHAPIN图，它把整个程序写在一个大框图内，大框图由

若干个小的基本框图构成

## 6.3设计目标

该阶段目的在于明确系统的数据结构和软件结构，此外总体设计还将给出内部软件和外部系统部件之间的接口定义，各个软件模块的功能说明，数据结构的细节以及具体的装配要求。

## 6.4运行环境

软件基本运行环境为Windows环境。

## 6.5条件限制

软件设计应当表现出层次结构，它应巧妙地利用各个软件部件之间的控制关系。

设计应当是模块化的，即该软件应当从逻辑上被划分成多个部件，分别实现各种特定功能和子功能。

设计最终应当给出具体的模块（例如子程序或过程)，这些模块就具有独立的功能特性。

应当应用在软件需求分析期间得到的信息，采取循环反复的方法来获得设计。

## 6.6详细设计的目的

详细设计的目的是为系统结构图中的每一个模块确定采用的算法（包括数据结构），用某种选定的表达工具给出清晰的描述，使程序员可以将这种描述直接翻译为某种语言程序。

# 七．系统实现

首先是数据库连接类的编写：

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Text;

namespace 面包销售

{

class DBHelper

{

public static string a;

public static string b1, b2, b3, b4, b5, b6;

public string connString = @"server=.;database=breadsale; integrated security=SSPI";

private SqlConnection connction;

public SqlConnection Connction

{

get

{

if (connction == null)

{

connction = new SqlConnection(connString);

}

return connction;

}

}

public void OpenConnection()

{

if (connction.State == ConnectionState.Closed)

{

Connction.Open();

}

else if (connction.State == ConnectionState.Broken)

{

Connction.Close();

Connction.Open();

}

}

public void CloseConnection()

{

if (connction.State == ConnectionState.Open || connction.State == ConnectionState.Broken)

{

Connction.Close();

}

}

}

}

然后是登录注册界面的实现：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace 面包销售

{

public partial class 登录注册 : Form

{

public 登录注册()

{

InitializeComponent();

}

private void btna\_Click(object sender, EventArgs e)

{

string userid = this.tba.Text;

string psword = this.tbb.Text;

if (userid.Equals("") || psword.Equals(""))//用户名或密码为空

{

MessageBox.Show("用户名或密码不能为空");

}

else

{

string sql = string.Format(@"select \* from [dbo].[user] where [id] = '{0}' and [password] = '{1}'", userid, psword);

DBHelper dbHelper = new DBHelper();

SqlCommand cmd = new SqlCommand(sql, dbHelper.Connction);

dbHelper.OpenConnection();

SqlDataReader sdr = cmd.ExecuteReader();

if (sdr.Read())

{

DBHelper.a = userid;

MessageBox.Show("信息验证成功");

main m = new main();

m.Show();

this.Hide();

}

else

{

MessageBox.Show("用户名或密码错误");

}

}

}

private void btnb\_Click(object sender, EventArgs e)

{

string userid = this.tba.Text;

string psword = this.tbb.Text;

if (userid.Equals("") || psword.Equals(""))//用户名或密码为空

{

MessageBox.Show("用户名或密码不能为空");

}

else

{

string sql = string.Format(@"insert into [dbo].[user]([id],[password]) values('{0}','{1}')", userid, psword);

DBHelper dbHelper = new DBHelper();

try

{

//执行工具

SqlCommand cmd = new SqlCommand(sql, dbHelper.Connction);

//打开数据库

dbHelper.OpenConnection();

//执行

int result = cmd.ExecuteNonQuery();

//判断

if (result > 0)

{

MessageBox.Show("注册成功，请登录！");

}

else

{

MessageBox.Show("注册失败，请重试！");

}

}

catch (Exception ex)

{

MessageBox.Show("发生错误，请联系管理员,错误原因是：" + ex.Message);

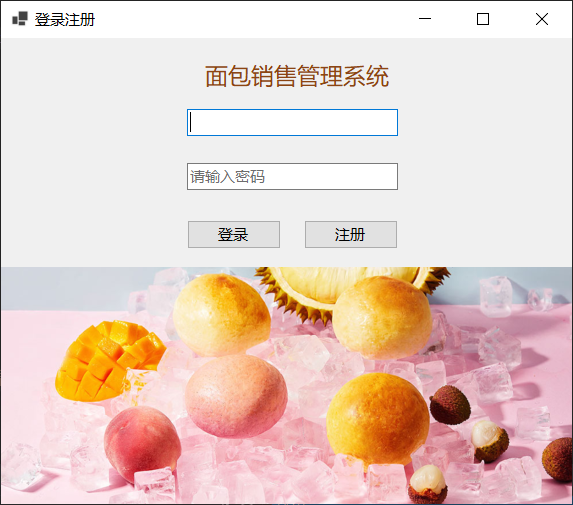
}

}

}

}

}



然后编写销售管理系统主界面：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace 面包销售

{

public partial class main : Form

{

public main()

{

InitializeComponent();

show();

show1();

}

void show()

{

DBHelper dbHelper = new DBHelper();

string sql = string.Format(@"select \* from [dbo].[info] where [num]>0;");

SqlDataAdapter adapter = new SqlDataAdapter(sql, dbHelper.Connction);

DataSet ds = new DataSet();

adapter.Fill(ds, "info");

this.dgv\_1.DataSource = ds.Tables["info"];

this.dgv\_2.DataSource= ds.Tables["info"];

}

void show1()

{

DBHelper dbHelper = new DBHelper();

string sql = string.Format(@"select \* from [dbo].[sale];");

SqlDataAdapter adapter = new SqlDataAdapter(sql, dbHelper.Connction);

DataSet ds = new DataSet();

adapter.Fill(ds, "sale");

this.dgv\_3.DataSource = ds.Tables["sale"];

}

private void btnsell\_Click(object sender, EventArgs e)

{

if (dgv\_1.SelectedRows.Count < 1)

{

MessageBox.Show("请选择面包种类");

}

else

{

if (Convert.ToInt32(dgv\_1.SelectedRows[0].Cells[2].Value) < Convert.ToInt32(tb.Text))

{

MessageBox.Show("库存不足");

}

else

{

Random ran = new Random();

string id = DBHelper.a + DateTime.Now.ToString("yyyyMMddHHmmss") + ran.Next(999);

string sql = string.Format(@"insert into [dbo].[sale] ([id],[name],[num],[time],[salerid],[price]) values('{0}','{1}',{2},'{3}','{4}','{5}');update [dbo].[info] set num=num-{6} where [id]='{7}';", id, dgv\_1.SelectedRows[0].Cells[1].Value.ToString(), Convert.ToInt32(tb.Text), DateTime.Now, DBHelper.a,Convert.ToDecimal(dgv\_1.SelectedRows[0].Cells[3].Value)\*Convert.ToInt32(tb.Text), Convert.ToInt32(tb.Text), Convert.ToInt32(dgv\_1.SelectedRows[0].Cells[0].Value));

DBHelper dbHelper = new DBHelper();

try

{

SqlCommand cmd = new SqlCommand(sql, dbHelper.Connction);

dbHelper.OpenConnection();

cmd.ExecuteNonQuery();

MessageBox.Show("出库成功");

dbHelper.CloseConnection();

show();

show1();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

private void btninsert\_Click(object sender, EventArgs e)

{

if (dgv\_2.SelectedRows.Count < 1)

{

MessageBox.Show("请选择面包种类");

}

else

{

if (Convert.ToInt32(tb2.Text)<1)

{

MessageBox.Show("数量错误");

}

else

{

string sql = string.Format(@"update [dbo].[info] set num=num+{0} where [id]='{1}';", Convert.ToInt32(tb2.Text), Convert.ToInt32(dgv\_2.SelectedRows[0].Cells[0].Value));

DBHelper dbHelper = new DBHelper();

try

{

SqlCommand cmd = new SqlCommand(sql, dbHelper.Connction);

dbHelper.OpenConnection();

cmd.ExecuteNonQuery();

MessageBox.Show("入库成功");

dbHelper.CloseConnection();

show();

show1();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

DBHelper.b1 = dgv\_3.SelectedRows[0].Cells[0].Value.ToString();

DBHelper.b2 = dgv\_3.SelectedRows[0].Cells[1].Value.ToString();

DBHelper.b3 = dgv\_3.SelectedRows[0].Cells[2].Value.ToString();

DBHelper.b4 = dgv\_3.SelectedRows[0].Cells[3].Value.ToString();

DBHelper.b5 = dgv\_3.SelectedRows[0].Cells[4].Value.ToString();

DBHelper.b6 = dgv\_3.SelectedRows[0].Cells[5].Value.ToString();

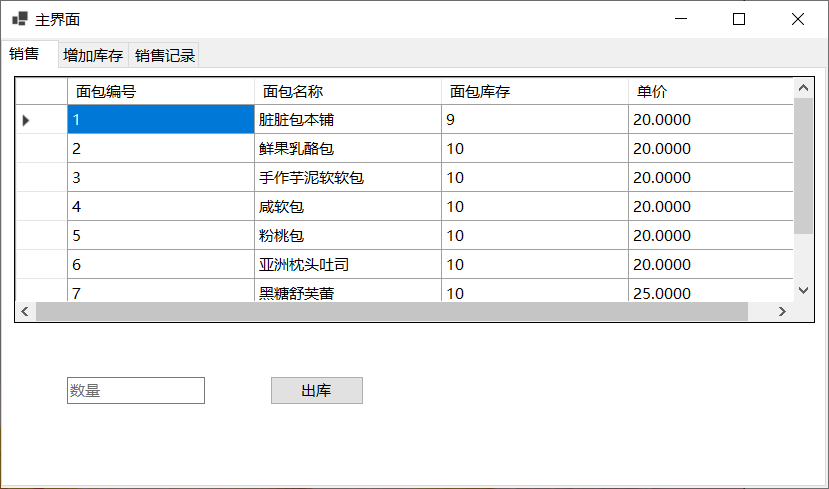
order od = new order();

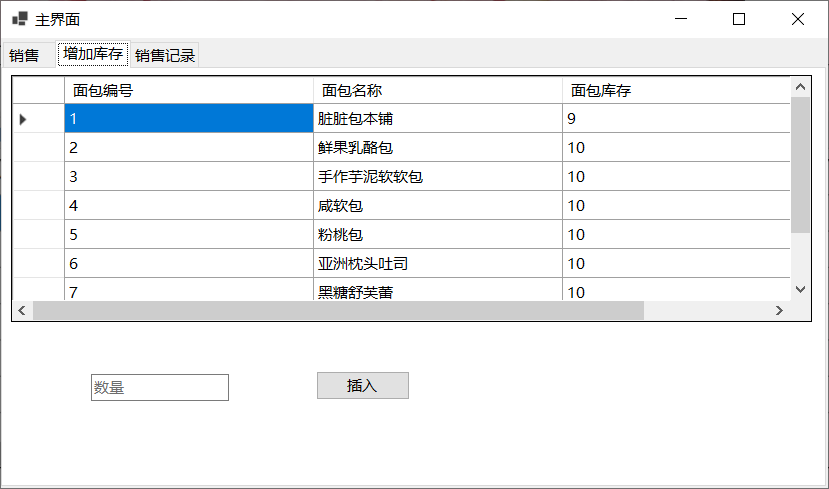
od.Show();

}

}

}







最后编写打印订单界面内容：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace 面包销售

{

public partial class order : Form

{

public order()

{

InitializeComponent();

labeldate.Text = "单据日期：" + DBHelper.b4;

labelid.Text = "订单编号：" + DBHelper.b1;

labelname.Text = DBHelper.b2;

labelnum.Text = DBHelper.b3;

labeltotal.Text = DBHelper.b6;

labelyingshou.Text= "应收金额：" + DBHelper.b6;

labelshishou.Text = "实收金额："+ DBHelper.b6;

label11.Text= DBHelper.b6;

}

}

}



# 八．数据库设计实现代码

-- ----------------------------

-- Table structure for info

-- ----------------------------

IF EXISTS (SELECT \* FROM sys.all\_objects WHERE object\_id = OBJECT\_ID(N'[dbo].[info]') AND type IN ('U'))

DROP TABLE [dbo].[info]

GO

CREATE TABLE [dbo].[info] (

[id] int NOT NULL,

[name] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NULL,

[num] int NULL,

[price] money NULL

)

GO

ALTER TABLE [dbo].[info] SET (LOCK\_ESCALATION = TABLE)

GO

-- ----------------------------

-- Records of info

-- ----------------------------

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'1', N'脏脏包本铺', N'9', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'2', N'鲜果乳酪包', N'10', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'3', N'手作芋泥软软包', N'10', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'4', N'咸软包', N'10', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'5', N'粉桃包', N'10', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'6', N'亚洲枕头吐司', N'10', N'20.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'7', N'黑糖舒芙蕾', N'10', N'25.0000')

GO

INSERT INTO [dbo].[info] ([id], [name], [num], [price]) VALUES (N'8', N'麻辣小龙虾包', N'10', N'25.0000')

GO

-- ----------------------------

-- Table structure for sale

-- ----------------------------

IF EXISTS (SELECT \* FROM sys.all\_objects WHERE object\_id = OBJECT\_ID(N'[dbo].[sale]') AND type IN ('U'))

DROP TABLE [dbo].[sale]

GO

CREATE TABLE [dbo].[sale] (

[id] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NOT NULL,

[name] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NULL,

[num] int NULL,

[time] datetime2(7) NULL,

[salerid] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NULL,

[price] decimal(18) NULL

)

GO

ALTER TABLE [dbo].[sale] SET (LOCK\_ESCALATION = TABLE)

GO

-- ----------------------------

-- Records of sale

-- ----------------------------

INSERT INTO [dbo].[sale] ([id], [name], [num], [time], [salerid], [price]) VALUES (N'120210713152322527', N'脏脏包本铺', N'1', N'2021-07-13 15:23:22.0000000', N'1', N'20')

GO

INSERT INTO [dbo].[sale] ([id], [name], [num], [time], [salerid], [price]) VALUES (N'120210713152343958', N'脏脏包本铺', N'2', N'2021-07-13 15:23:43.0000000', N'1', N'40')

GO

INSERT INTO [dbo].[sale] ([id], [name], [num], [time], [salerid], [price]) VALUES (N'120210713164322464', N'脏脏包本铺', N'1', N'2021-07-13 16:43:22.0000000', N'1', N'20')

GO

INSERT INTO [dbo].[sale] ([id], [name], [num], [time], [salerid], [price]) VALUES (N'220210713164132533', N'鲜果乳酪包', N'2', N'2021-07-13 16:41:32.0000000', N'2', N'40')

GO

-- ----------------------------

-- Table structure for user

-- ----------------------------

IF EXISTS (SELECT \* FROM sys.all\_objects WHERE object\_id = OBJECT\_ID(N'[dbo].[user]') AND type IN ('U'))

DROP TABLE [dbo].[user]

GO

CREATE TABLE [dbo].[user] (

[id] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NOT NULL,

[password] varchar(255) COLLATE Chinese\_PRC\_CI\_AS NULL

)

GO

ALTER TABLE [dbo].[user] SET (LOCK\_ESCALATION = TABLE)

GO

-- ----------------------------

-- Records of user

-- ----------------------------

INSERT INTO [dbo].[user] ([id], [password]) VALUES (N'1', N'1')

GO

INSERT INTO [dbo].[user] ([id], [password]) VALUES (N'2', N'2')

GO

-- ----------------------------

-- Primary Key structure for table info

-- ----------------------------

ALTER TABLE [dbo].[info] ADD CONSTRAINT [PK\_\_info\_\_3213E83F2645ED70] PRIMARY KEY CLUSTERED ([id])

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

ON [PRIMARY]

GO

-- ----------------------------

-- Primary Key structure for table sale

-- ----------------------------

ALTER TABLE [dbo].[sale] ADD CONSTRAINT [PK\_\_sale\_\_3213E83F843A6C65] PRIMARY KEY CLUSTERED ([id])

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

ON [PRIMARY]

GO

-- ----------------------------

-- Primary Key structure for table user

-- ----------------------------

ALTER TABLE [dbo].[user] ADD CONSTRAINT [PK\_\_user\_\_3213E83F73F159A1] PRIMARY KEY CLUSTERED ([id])

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

ON [PRIMARY]

GO

# 九．心得体会与总结

通过这次课程设计发现这其中需要的很多知识我们没有接触过，上网查找资料的时候发现我们以前所学到的仅仅是皮毛，还有很多需要我们掌握的东西我们根本不知道。同时也发现有很多已经学过的东西我们没有理解到位，不能灵活运用于实际，不能很好的用来解决问题，这就需要自己不断的大量的实践，通过不断的自学，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的东西。这次的数据库课程设计，我们组负责的人员请假管理系统的设计。从各种文档的阅读到需求分析、概要设计、数据库总体设计、代码编写与调试，我们都准备了好长时间。组内分工合作的整个过程，我亲身体验了一回系统的设计开发过程，分工合作的好处。很多东西书上写的很清楚，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的时候才发现其中的难度。经常做到后面突然就发现自己一开始的设计有问题，然后又回去翻工，在各种反复中不断完善自己的想法。

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，结合我们的生活经验，根据可行性研究的结果和用户的需要，分析现有情况及问题。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，经过组内讨论。最终把它解决了。通过这次课程设计,我对数据库的设计更加熟练了。

通过本次课程设计，对SQL语言，数据库的创建、修改、删除方法有了一定的了解，通过导入表和删除表、更改表，学会了数据库的基本操作。很多事情不是想象中的那么简单的，它涉及到的各种实体、属性、数据流程、数据处理等等。很多时候感觉后面的设计根本无法继续，感觉像是被前面做的各种图限制了。在做关系模型转换的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。

通过本次课程设计，我觉得不管做什么，始终都要相信自己。不要畏惧，不要怕遇到困难，什么都要去尝试，有些你开始认为很难的事在你尝试之后你可能会发现原来它并没有那么难，只要钻研就一定会有收获。