华中科技大学

数据库系统原理实践报告

姓　　名：　熊宇飞

学　　院：　计算机科学与技术学院

专　　业：　计算机科学与技术

班　　级：　计科 1402

学　　号：　U201414602

指导教师：　吴海

分数	
教师签名	

2017 年 6 月 23 日

目　录

1 课程任务概述

数据库综合实践是为数据库原理及应用等系列课程而独立开设的实践性课程。数据库综合实践对于巩固数据库知识，加强学生的实际动手能力和提高学生综合素质十分必要。本课程分为软件学习、sql 语句练习和大型数据库系统实现三个阶段进行。数据库综合实践的主要目标是：

1) 加深对数据库系统、程序设计语言的理论知识的理解和应用水平。
2) 通过设计实际的数据库系统应用课题，进一步熟悉数据库管理系统的操作技术，提高动手能力，提高分析问题和解决问题的能力。

2 软件功能学习部分

2.1 任务要求

1. 练习 sqlserver 的两种完全备份方式：数据和日志文件的脱机备份、系统的备份功能。
2. 练习在新增的数据库上增加用户并配置权限的操作。

2.2 完成过程

2.2.1 数据和日志文件的脱机备份、系统备份

脱机步骤：

1) 关闭所有表和查询。
2) 鼠标右键点击创建的数据库->任务->脱机。

脱机演示图如下图 2.1 所示：

图 2.1 脱机演示图

3） 脱机之后数据库便不再连接。

备份操作：

1） 在数据库连接的状态下，点击备份按钮备份文件。备份类型选择“完整”。然后在所选目录下有名为<db_name>.bak 的文件生成。可以利用该文件在需要的时候还原数据库。

2） 如果要备份日志，仍按照 1）中步骤操作，唯一不同的是备份类型选择“事务日志”。然后有<db_name_logbackup_date>.bak 的文件生成。可以利用该文件回恢复日志。

2.2.2 在新增的数据库上增加用户并配置权限

1) 新增一个登录名。如下图 2.2 所示，新建一个登录名为 feifei 的 SQL server 身份验证的账号。

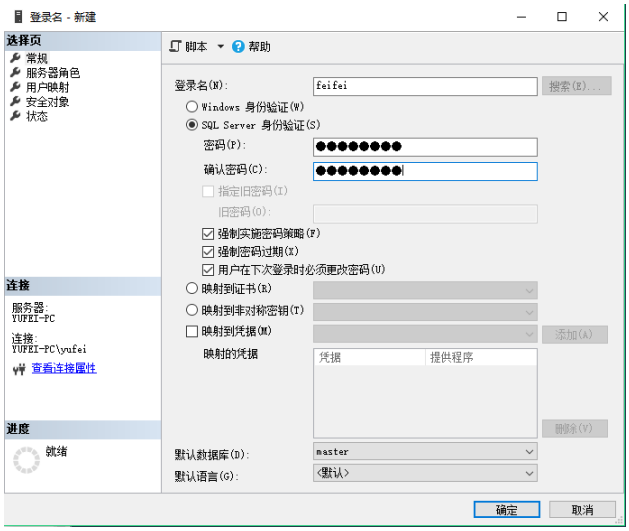


图 2.2 新增登录名演示图

2) 为数据库新增一个用户。如下图 2.3 所示，新用户名为 yuhan。

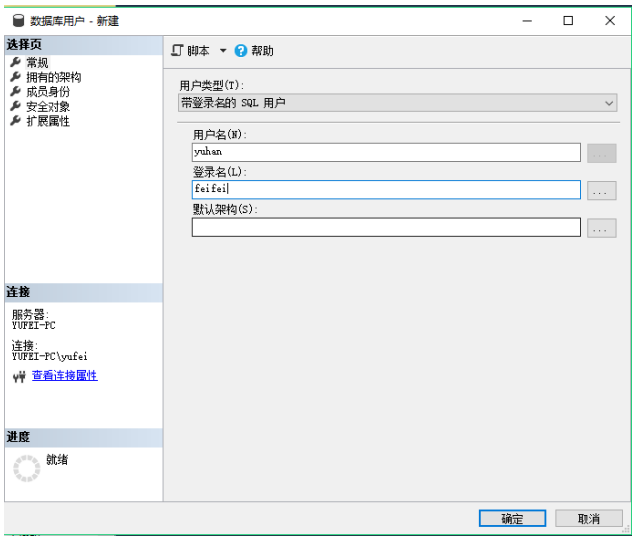


图 2.3 新增用户演示图

3) 为这个用户分配权限。如下图 2.4 所示，为新增用户分配权限。

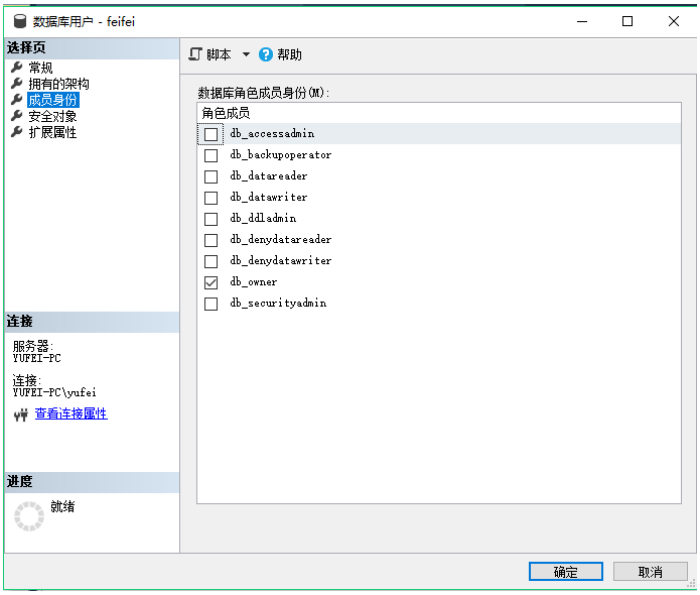


图 2.4 分配权限图

2.3 任务总结

实验一比较简单。主要是熟悉 sqlserver 的使用方法。在实践的过程中，我学到了如何建立数据库，如何给数据库添加表项，如何备份数据库以及如何从备份中还原数据库。操作简单，使用方便上手很快。在老师的指导下，我主要弄清楚了分离和脱机的区别与联系。分离和脱机都可以使数据库不能再被使用，但是分离后需要附加才能使用，而脱机后只需联机就可以用了

脱机与联机是相对的概念，它表示数据库所处的一种状态，脱机状态时数据库是存在的，只是被关闭了，用户不能访问而已，要想访问可以设为联机状态

分离与附加是相对的两个概念，分离后，数据库不存在，只存在数据库对应的安装地址下，要使用这些文件，就要附加他们。

3 SQL 语句练习

3.1 任务要求

3.1.1 建表

1）**创建如下三个关系，包括主码和外码的说明**

商品表【商品名称、商品类型】

GOODS【GNAME char（20）,GTYPE char（10）】

主关键字为（商品名称）。商品类型为（电器、文具、服装。。。）

商场【商场名称,所在地区】

PLAZA【PNAME char（20），PAREA char（20）】

主关键字为商场名称。所在地区为（洪山、汉口、汉阳、武昌。。。）

销售价格表【商品名称、商场名称、当前销售价格、目前举办活动类型】

SALE【GNAME char（20），PNAME char（20），PRICE FLOAT, ATYPE int】

主关键字为（商品名称、商场名称）。举办活动类型为（0，表示送券；大于 0 且小于 9 的整数，表示打折，例如 8 表示打八折；也可为空值，表示当前未举办任何活动）。例如，记录（‘jeep 男装’，‘大洋百货’，2000，9）表示大洋百货的 jeep 男装打 9 折，同一商场针对不同的商品可能采取不同的促销活动。

2）**观察性实验**

验证在建立外码时是否一定要参考被参照关系的主码，并在实验报告中简述过程和结果。

3.1.2 数据更新

1）向上述表格中用 sql 语句完成增、删、改的操作；

2）批处理操作

将 SALE 表中的打折记录插入到新表 SALE_CHEAP 中，并基于 SALE_CHEAP 表创建一个统计每个商场各自打折商品数的视图。

3）观察性实验

建立一个关系，但是不设置主码，然后向该关系中插入重复元组，然后观察在图形化交互界面中对已有数据进行删除和修改时所发生的现象。（全部删除，全部修改）。

4）触发器实验

编写一个触发器，用于实现对 SALE 表的完整性控制规则：当向该表插入一条记录时，若商品价格高于 2000 元，则该件商品最多打 9 折。

3.1.3 查询

1）查询所有“羽绒服”以“打折”方式的销售情况，并按照价格的升序排列；

2）查询所有在售的、没有任何活动的商品及其所在的商场，结果按照商品价格降序排列；
3）查询所有没有参加打折活动的商品；
4）查询价格在 200～500 元之间的商品名称、所在的商场名称、价格，结果按照商场名称降序排列，同一个商场的则按照价格的升序排列；
5）查询每种商品的最低价格、商品名称；
6）查询以“打折”方式销售的商品总数超过 30 种的商场名称；
7）查询以“打折”方式销售的商品总数超过 30 种的商场所在地区；
8）查询价格为下列取值之一的商品名称、所在商场名称、目前举办活动的类型，（9.8，98，998，9998）；
9）查询以“海尔”开头的所有商品的名称；
10）查询同时销售“剃须刀”和“理发器”的商场名称；
11）查询有打折但没有送券活动的商场；
12）查询所销售的商品包含了“海澜之家”所销售的所有商品的商场名称；
13）查询所有电器类商品及其销售情况，要求，即使某件电器商品没有销售信息也要能够输出其商品名称和商品类型。

3.1.4 了解系统的查询性能分析功能

选择上述第 3 题中较为复杂的 SQL 语句，查看其执行之前系统给出的分析计划和实际的执行计划，并进行简单的分析。

3.1.5 DBMS 函数调用

1）通过系统帮助文档学习系统关于时间、日期、字符串类型的函数，并将其应用于 SQL 语句。
2）编写自定义的函数，并将其应用于 SQL 语句。

3.2 完成过程

3.2.1 建表

SQL 建表语句如下：

(1) 建立商品表：

```
CREATE TABLE GOODS(GNAME CHAR(20) PRIMARY KEY, GTYPE CHAR(10) NOT NULL );
```

(2) 建立商场表：

```
CREATE TABLE PLAZA(PNAME CHAR(20) PRIMARY KEY, PAREA CHAR(20) NOT NULL );
```

(3) 建立销售价格表：CREATE TABLE SALE(GNAME CHAR(20) , PNAME

CHAR(20), PRICE FLOAT(10, 2) NOT NULL, ATYPE INT, PRIMARY KEY(GNAME, PNAME), FOREIGN KEY (GNAME) REFERENCES GOODS(GNAME), FOREIGN KEY (PNAME) REFERENCES PLAZA(PNAME));

2）观察性实验

在建立 SALE 表时将 GNAME 本应参照 GOODS 主码 GNAME 改为参照 GOODS 表的 GTYPE，执行 SQL 语句时数据库软件报错，所以一定要参照主码。但是另外一个测试使 GNAME 参照 PNAME、PNAME 参照 GNAME，这种虽然对应关系有误，但是可以建表。所以在建立外码时一定要参考被参照关系的主码，但是这个参照与被参照的关系可以与实际的关系不同，在语法上并不会出错。如下图 3.1 所示，提示 SALE 表建立出错。

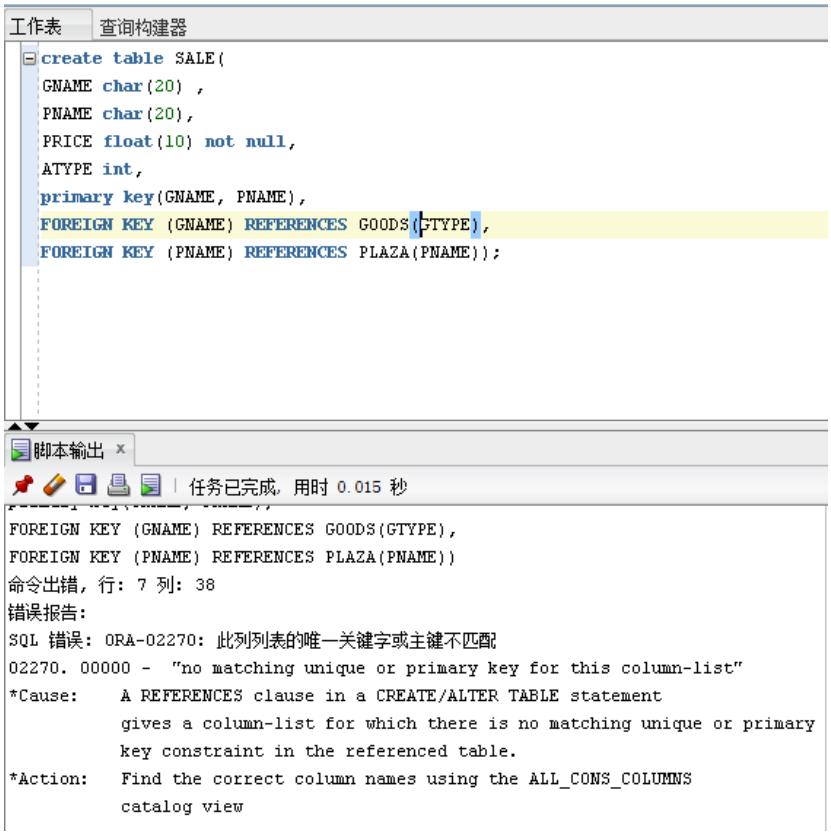


图 3.1 建表失败演示图

3.2.2 数据更新

(1) 向上述表格中用 sql 语句完成增、删、改的操作

使用自己创建的数据集完成数据的添加。添加代码请查看附录。

(2) 批处理操作

新建一个表：CREATE TABLE SALE_CHEAP (GNAME CHAR(20), PNAME CHAR(20) NOT NULL, ATYPE INT)；批处理操作演示图如下图 3.2 所示。

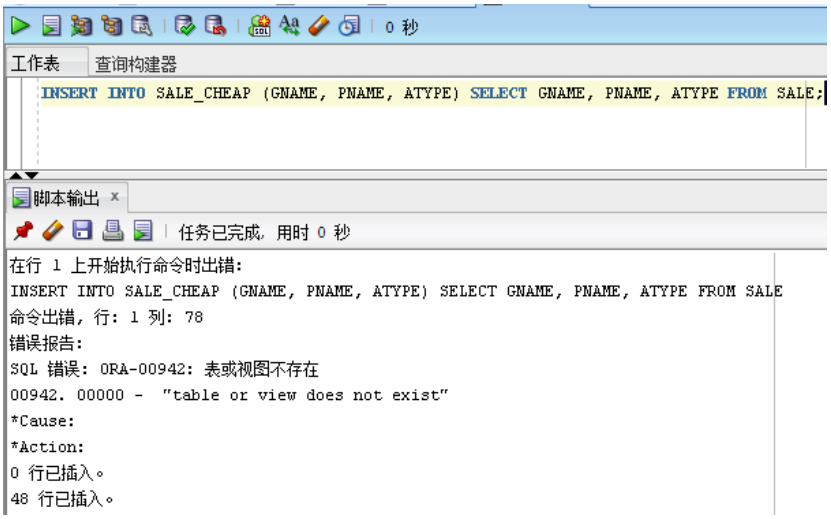


图 3.2 批处理操作演示图

建立视图：CREATE VIEW VSALE_CHEAP AS SELECT PNAME, COUNT(PNAME) AS GCOUNT FROM SALE_CHEAP WHERE ATYPE > 0 GROUP BY PNAME;

新建的视图如图 3.3 所示。

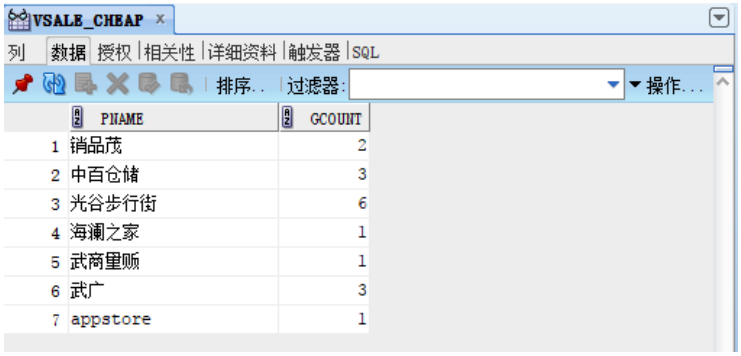


图 3.3 视图展示图

(3) 观察性实验

首先建立一个关系 NO_KEY，如图所示，该关系中没有主码。然后向该关系中插入重复元组，在图形化交互界面中对已有数据进行删除和修改时可以发现程序给出了警告，因为没有主码，程序使用图所示语句对表进行了更新。

(4) 触发器实验

触发器 SQL 语句如下：

```
CREATE TRIGGER SALE_CTRL BEFORE INSERT ON SALE
FOR EACH ROW
BEGIN
 IF NEW.PRICE > 2000 AND NEW.ATYPE < 9
```

```
    THEN SET NEW.ATYPE = 9;
 END IF;
END;
```

新建的触发器在插入表项时能正常的处理价格大于2000同时打折低于9折的商品，将其打折修改为9折。

3.2.3 查询

1）查询所有“羽绒服”以“打折”方式的销售情况，并按照价格的升序排列；

SQL语句：SELECT * FROM SALE WHERE GNAME = '羽绒服' AND ATYPE BETWEEN 1 AND 9 ORDER BY PRICE DESC;

查询结果如图3.4所示：

	GNAME	PNAME	PRICE	ATYPE
1	羽绒服	海澜之家	998	8
2	羽绒服	光谷步行街	888	9
3	羽绒服	销品茂	499	8

图3.4 查询结果1

2）查询所有在售的、没有任何活动的商品及其所在的商场，结果按照商品价格降序排列；

SQL语句：SELECT GNAME, PNAME FROM SALE WHERE ATYPE IS NULL ORDER BY PRICE DESC;

查询结果如图3.5所示：

	GNAME	PNAME
1	电脑	光谷步行街
2	电脑	武广
3	海尔热水器	武广
4	海尔空调	销品茂
5	海尔空调	光谷步行街
6	海尔冰箱	武广
7	海尔洗衣机	武广
8	裤子	海澜之家
9	电饭煲	武广
10	皮夹克	光谷步行街
11	卫衣	武商里贩
12	豆浆机	中百仓储
13	吹风机	武商里贩
14	加湿器	中百仓储
15	吹风机	中百仓储
16	剃须刀	光谷步行街
17	钢笔	武广
18	文房四宝	武广
19	毛笔	中百仓储
20	毛笔	武商里贩
21	钢笔	中百仓储
22	橡皮擦	武广
23	橡皮擦	中百仓储

图3.5 查询结果2

3）查询所有没有参加打折活动的商品；

SQL 语句：SELECT GNAME FROM GOODS WHERE GNAME NOT IN (SELECT GNAME FROM SALE WHERE ATYPE > 0 AND ATYPE <=9);

查询结果如图 3.6 所示：

	GNAME
1	铅笔
2	电脑
3	豆浆机
4	海尔空调
5	西装
6	长袍
7	风衣
8	笔袋
9	文具盒
10	理发器
11	长尾夹
12	水彩笔
13	打孔机
14	卫衣
15	雨衣
16	毛笔
17	橡皮擦
18	钢笔
19	皮夹克
20	中山装
21	运动服

图 3.6 查询结果 3

4）查询价格在 200～500 元之间的商品名称、所在的商场名称、价格，结果按照商场名称降序排列，同一个商场的则按照价格的升序排列；

SQL 语句：SELECT GNAME, PNAME ,PRICE FROM SALE WHERE PRICE BETWEEN 200 AND 500 ORDER BY PNAME DESC ,PRICE ASC;

查询结果如图 3.7 所示：

	GNAME	PNAME	PRICE
1	豆浆机	中百仓储	200
2	电饭煲	中百仓储	300
3	豆浆机	销品茂	260
4	裤子	销品茂	299
5	羽绒服	销品茂	499
6	卫衣	武商里贩	200
7	电饭煲	武广	299
8	加湿器	武广	499
9	裤子	海澜之家	350
10	皮夹克	光谷步行街	220

图 3.7 查询结果 4

5）查询每种商品的最低价格、商品名称；

SQL 语句：SELECT GNAME, MIN(DISTINCT PRICE) FROM SALE GROUP BY GNAME;

查询结果如图 3.8 所示：

	GNAME	MIN(DISTINCTPRICE)
1	海尔电视	5000
2	文房四宝	20
3	抽湿机	80
4	ipad	120
5	海尔洗衣机	800
6	剃须刀	50
7	电脑	4999
8	海尔空气进化器	1200
9	橡皮擦	3
10	加湿器	80
11	卫衣	150
12	电饭煲	299
13	豆浆机	200
14	海尔热水器	2999
15	裤子	299
16	皮夹克	220
17	羽绒服	499
18	吹风机	50
19	海尔冰箱	998
20	海尔空调	1500
21	钢笔	9.8
22	毛笔	9.8

图 3.8 查询结果 5

6）查询以“打折”方式销售的商品总数超过 30 种的商场名称；

SQL 语句：SELECT PNAME FROM SALE WHERE ATYPE > 0 AND ATYPE <= 9 GROUP BY PNAME HAVING COUNT(PNAME) > 1;

查询结果如图 3.9 所示：

	PNAME
1	销品茂
2	中百仓储
3	光谷步行街
4	武广

图 3.9 查询结果 6

7）查询以“打折”方式销售的商品总数超过 30 种的商场所在地区；

SQL 语句：SELECT DISTINCT PAREA FROM PLAZA WHERE PNAME IN (SELECT PNAME FROM SALE WHERE ATYPE > 0 AND ATYPE <= 9 GROUP BY PNAME HAVING COUNT(PNAME) > 1);

查询结果如图 3.10 所示：

	PAREA
1	汉口
2	洪山
3	武昌

图 3.10 查询结果 7

8）查询价格为下列取值之一的商品名称、所在商场名称、目前举办活动的类型，（9.8，98，998，9998）；

SQL 语句：SELECT GNAME, PNAME, ATYPE FROM SALE WHERE PRICE = 9.8 OR PRICE = 98 OR PRICE = 998 OR PRICE = 9998;

查询结果如图 3.11 所示：

	GNAME	PNAME	ATYPE
1	羽绒服	海澜之家	8
2	吹风机	武商里贩	(null)
3	裤子	光谷步行街	9
4	海尔冰箱	武广	(null)
5	海尔洗衣机	销品茂	9
6	剃须刀	武广	7
7	电脑	光谷步行街	(null)
8	钢笔	中百仓储	(null)
9	毛笔	武商里贩	(null)
10	毛笔	武广	0

图 3.11 查询结果 8

9）查询以“海尔”开头的所有商品的名称；

SQL 语句：SELECT GNAME FROM GOODS WHERE GNAME LIKE '海尔%';

查询结果如图 3.12 所示：

	GNAME
1	海尔冰箱
2	海尔电视
3	海尔空调
4	海尔空气进化器
5	海尔热水器
6	海尔洗衣机

图 3.12 查询结果 9

10）查询同时销售“剃须刀”和“理发器”的商场名称；

SQL 语句：SELECT PNAME FROM SALE A WHERE EXISTS (SELECT * FROM SALE B WHERE A.PNAME = B.PNAME AND B.GNAME = '皮夹克') AND A.GNAME = '剃须刀';

查询结果如图 3.13 所示：

	PNAME
1	光谷步行街

图 3.13 查询结果 10

11）查询有打折但没有送券活动的商场；

SQL 语句：SELECT DISTINCT PNAME FROM SALE A WHERE NOT EXISTS (SELECT * FROM SALE B WHERE A.PNAME = B.PNAME AND

B.ATYPE = 0) AND A.ATYPE BETWEEN 1 AND 9;

查询结果如图 3.14 所示：

	PNAME
1	中百仓储
2	海澜之家
3	appstore

图 3.14 查询结果 11

12）查询所销售的商品包含了“海澜之家”所销售的所有商品的商场名称；

SQL 语句：SELECT DISTINCT PNAME FROM SALE A WHERE NOT EXISTS (SELECT * FROM SALE B WHERE PNAME ='海澜之家' AND NOT EXISTS (SELECT * FROM SALE C WHERE C.GNAME = B.GNAME AND C.PNAME = A.PNAME));

查询结果如图 3.15 所示：

	PNAME
1	销品茂
2	光谷步行街
3	海澜之家

图 3.15 查询结果 12

13）查询所有电器类商品及其销售情况，要求，即使某件电器商品没有销售信息也要能够输出其商品名称和商品类型。

SQL 语句：SELECT GNAME, PNAME, PRICE FROM GOODS LEFT OUTER JOIN SALE USING(GNAME) WHERE GTYPE = '电器';

查询结果如图 3.16 所示：

	GNAME	PNAME	PRICE
1	抽湿机	光谷步行街	80
2	吹风机	光谷步行街	100
3	吹风机	中百仓储	50
4	吹风机	武商里贩	98
5	电饭煲	武广	299
6	电饭煲	中百仓储	300
7	电脑	武广	4999
8	电脑	光谷步行街	9998
9	豆浆机	中百仓储	200
10	豆浆机	销品茂	260
11	海尔冰箱	光谷步行街	1200
12	海尔冰箱	武广	998
13	海尔电视	光谷步行街	5000
14	海尔电视	武广	6999
15	海尔空调	销品茂	2000
16	海尔空调	光谷步行街	1500
17	海尔空气进化器	武广	1200
18	海尔空气进化器	光谷步行街	1699
19	海尔热水器	武广	2999
20	海尔热水器	武商里贩	3500
21	海尔洗衣机	销品茂	998
22	海尔洗衣机	武广	800
23	加湿器	中百仓储	80

图 3.16 查询结果 13

3.2.4 了解系统的查询性能分析功能

选择上述第 3 题中的第 12 个查询，在 ORACLE 中使用 explain 语句来分析 select 子句 。explain 返回一行记录，它包括了 select 语句中用到的各个表的信息。从记录中可以看出，该 select 子句分别在三个表上进行了三次查询，其中 type 字段是联合查询所使用的类型即访问类型，对应于该查询的三次查询，id 为 1 的查询所使用的访问类型是较差的一种，所以对于该查询，如果要考虑优化的话可以从这方面着手分析。

3.2.5 DBMS 函数调用

第二个实验并未涉及 DBMS 函数调用，但是我在第三个实验中频繁用到了 to_date()函数，因为我很多表中都存放日期信息。下面显示一个包含 to_date()函数的 sql 语句：

```
string sql = string.Format(@"update TREATINFO set TIME to_date('{0}','yyyy-MM-dd'),DISEASE = '{1}' where PATIENTID = '{2}' and DOCID = '{3}'and time = TO_DATE('{4}','yyyy-MM-dd')", date, disease,pid,user,previousDATE);
```

3.3 任务总结

第二个任务比第一个任务难上许多。不过通过这个 SQL 的练习任务，我进一步强化了编写 SQL 语句的能力。在课程中所学的理论是需要在实际应用中不断的锻炼的。比如在触发器实验中，课本上虽然给出了触发器的实现原理和语法，但是在不同的数据库软件中，触发器的实现还有很多差别，在 Sql server 中，有些触发器很难写成 before 类型的，而在 ORACLE 中，触发器的语法与课本上还是有很大不同。只有深入学习，才能很好的掌握 SQL。

另外，对于不同的数据库，查询性能的分析有不同的方法，但是在查询优化的手段上是基本类似的，高效的查询有助于加速软件的运行而带来很多好处，所以我们要重视对查询性能的分析。在本次实验中我仅仅简单的对于查询性能进行了分析，而在实际中，分析要通过各方面的信息进行对比，找出瓶颈部分并对其进行优化。

4 医院信息管理系统

4.1 系统设计目标

一个现代化医院的综合管理是否先进是直接通过其信息化水平来体现的，本实验开发的医院信息管理系统是经过简化的信息化管理系统，该系统包含账户管理、住院登记、医生站、价格管理、成本核算、药库管理等子系统，可以满足各个部门的业务信息处理和信息共享。

医院信息管理系统简单易用，方便医院管理员，医生，收费员和患者进行登录使用。

医院管理人员可以用该信息管理系统查询修改医院的各项内容。包括查询医生信息，查询病人信息，查询药品信息和病房信息等。

医生可以用该信息管理系统查询或修改个人信息，查询当日诊疗信息，查询自己的病人信息。

收费员可以用该信息管理系统管理病人购买药品，办理住院等。

病人可以用该信息管理系统了解医院信息，查找专家资料，方便查询各种费用收取情况。该系统还能为住院病人提供每日住院清单，使患者明白、放心治疗。

4.2 需求分析

为了设计好医院信息管理系统，我们必须首先对医院实际需求有所分析。下面我将从功能需求、性能需求、数据完整性需求、数据流图、数据字典等方面分析该系统的需求。

4.2.1 功能需求

基本要求：

1) 提供面向公众的导医和收费标准明细查询的功能。
2) 挂号、收费、诊疗人员等具有不同的查询和修改权限。
3) 按照看病的基本流程（例如：预约——挂号——诊断——检查——复诊——住院治疗——结算）进行信息管理。
4) 提供病人收费汇总清单，提供各种药品或检查项目的使用情况汇总；
5) 提供医院各部门财务报表及医院整体财务报表，并且分日明细表和月、年汇总表。
6) 提供各类用户的注册功能。

其他要求：

1) 该系统应该支持四类用户的登录使用，他们分别是医院管理员、医生、病人、收费员。

2) 对于医院管理员，又可细分为院长（根管理员）和各个部门的主管。院长具有最高权限，他可以利用该系统实现病人信息的查询，医生和医院其他工作人员（例如收费员）的信息的查询与修改，药品库和医院病房信息的查询与修改，还可以查询修改部门管理员的信息，查询医院财务信息。对于部门主管，他具有次高级权限，他可以查询医院相关的几乎所有信息，但是不能修改院长的信息，也不能修改其他部门主管和其他部门的医生的信息。他可以修改本部门的医生的信息和其他医院相关信息，可以查询本科室的财务信息。
3) 对于医生，他只能修改个人信息、病人的某些信息和诊疗信息。他可以查询几乎所有医院相关信息，包括医院管理员某些信息，其他医生某些信息，病人信息，药品信息和病房信息，但不支持医生查询财务信息。
4) 对于收费员，他只能修改个人信息，病人账单上的信息，挂号信息，药品信息，病房信息。可以查询药品信息，病房信息。

4.2.2 性能需求

本实验采用 C#语言，该语言是微软公司发布的一种面向对象的、运行于.NET Framework 之上的高级程序设计语言。它的运行速度较快，介于 C++和 java 之间（比 C++慢，比 java 稍快）。数据库采用 oracle 数据库。

数据完整性需求：

医院信息管理系统所管理的信息是病人与医院的各种信息，其对于数据完整性也有一定要求，比如：需要对实体的完整性进行控制，对每个病人分配的标识必须能唯一的表示那个病人；在医生为病人开药时，应该检查是否超出安全的用量，这就属于用户定义完整性的要求范围；当需要修改某个主体的信息时，对于各种参照完整性也要加以考虑。

4.2.3 数据流图

1. 医院信息管理系统流程图：

医院信息管理流程图如下图 4.1 所示。其中主要有四个角色，它们分别为病人，医生，管理员，收费员。病人产生的数据主要有预约挂号信息，医生产生的信息主要有诊疗信息，收费员产生信息主要有账单信息，更新药品信息，管理员主要产生的信息有财务报表信息。

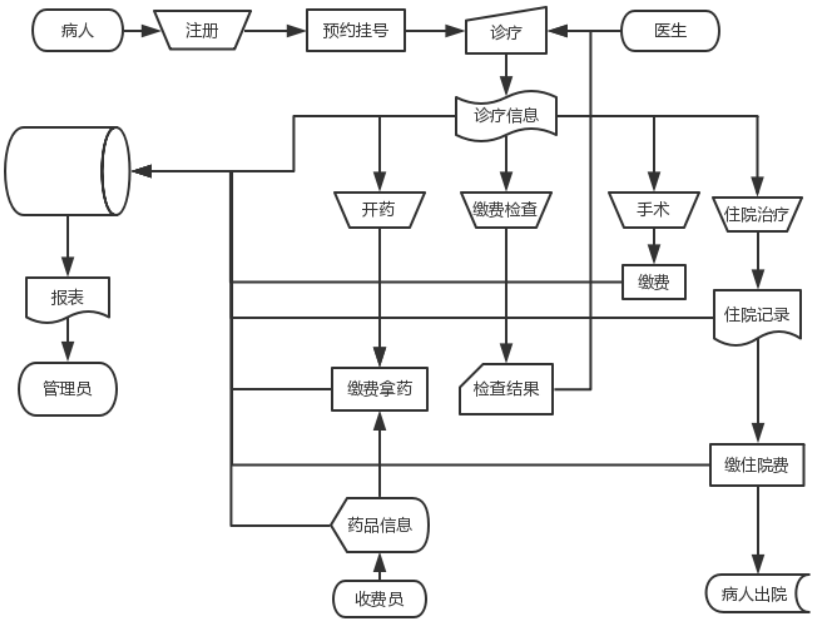


图 4.1 医院信息管理系统流程图

2. 病人治疗信息流图：

病人诊疗流程图如下图 4.2 所示。病人治疗的整个流程为：预约—挂号—门诊—检查—复诊—住院治疗—出院结算。

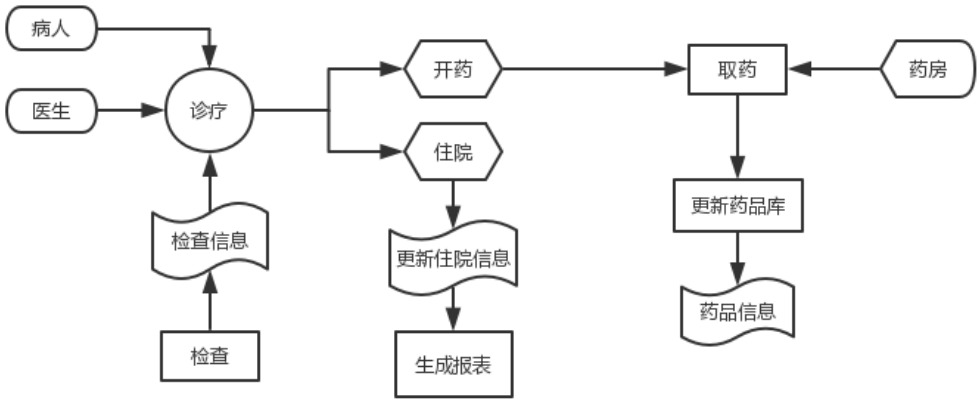


图 4.2 病人治疗信息数据流图

3. 医院财务数据流图：

医院财务数据流图如下图 4.3 所示。简单来讲，病人缴纳费用，收费员

收取费用，管理员可以查看医院的收入情况。

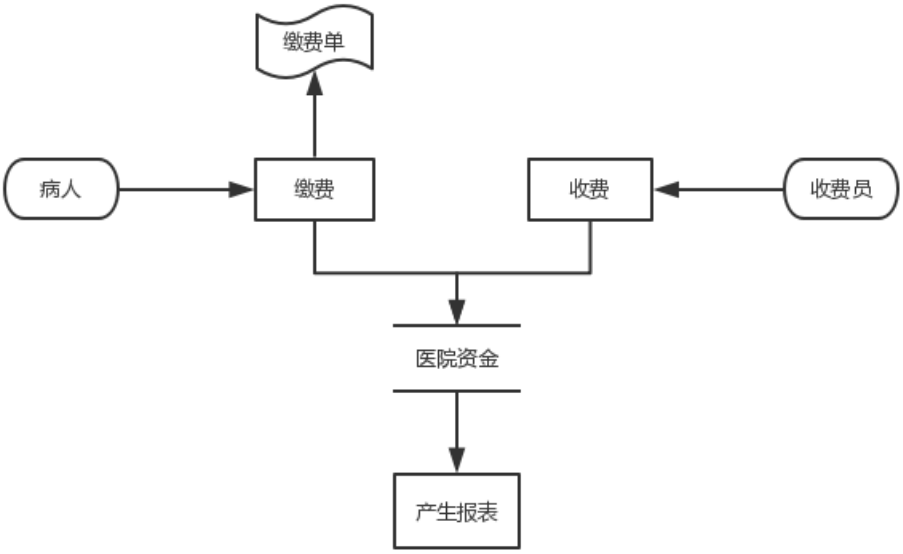


图 4.3 医院财务数据流图

4. 系统功能模块图：

系统功能模块图如下图 4.4 所示，本系统主要可以分为七大模块，分别为：登录模块、预约挂号模块、药品信息模块、诊疗模块，查询模块、住院信息管理模块、财务管理模块。

(1) 登录模块：控制各种用户的登录系统。

(2) 挂号模块：病人或者挂号处医生将病人信息录入系统选择对应科室进行挂号；

(3) 财务模块：病人缴费处理，挂号处医生收费，医院资金管理和各种财务报表的产生。

(4) 诊疗模块：诊治医生录入病人病情诊断、处方，检验科医生录入化验结果。

(5) 药品管理模块：药剂科医生管理药品信息，

(6) 住院信息维护模块：记录住院病人住院期间的各种信息，如：病房信息，用药信息，费用信息等。

(7) 查询模块：各种信息的查询，如：病人信息查询，费用查询等。

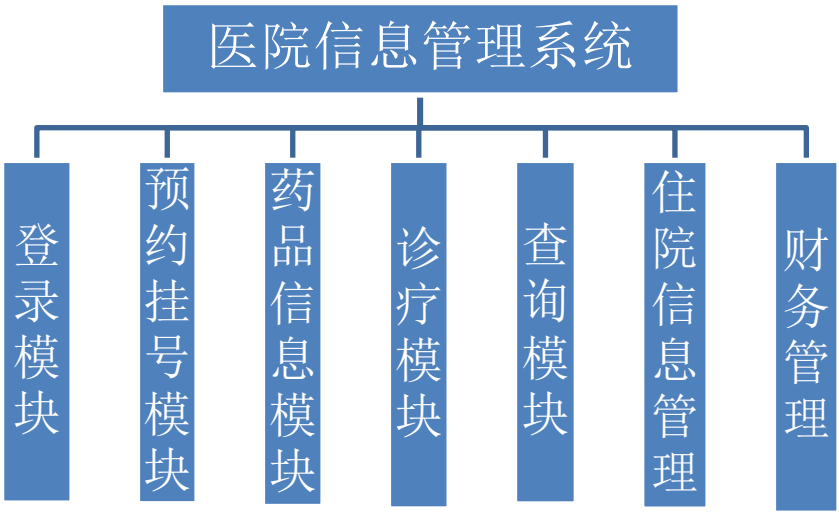


图 4.4 医院管理系统功能需求模块图

4.2.4 数据字典

建立数据库表的时候应充分考虑实际情况，尽量做到完整。下面将介绍我建立的十三个表。

一、 预约表：预约表见表 4.1。

表 4.1 预约表

APPOINT	主键	外键	类型	说明
PATIENTID	√	√	char	病人编号
DOCID	√	√	char	医生编号
DATETIME	√	×	date	日期

二、 账单信息表：账单信息表见表 4.2。

表 4.2 账单信息表

BILL	主键	外键	类型	说明
SERIAL	√	×	char	当日流水号
DATETIME	√	×	char	日期
PATIENTID	×	√	char	病人编号
ITEM	×	×	char	项目
AMOUNT	×	×	char	项目数量
UNITPRICE	×	×	int	单价
DOCID	×	√	char	医生编号
CASHERID	×	√	char	收费员

三、 收费员信息表：收费员信息表见表 4.3。

表 4.3 收费员信息表

CASHER	主键	外键	类型	说明
ID	√	×	char	员工编号
PASSWD	×	×	char	密码
NAME	×	×	char	姓名
GENDER	×	×	char	性别
ENTRY	×	×	char	入职时间
BIRTH	×	×	char	出生日期
SALARY	×	×	int	薪水

四、 检查项目表：检查项目表见表 4.4。

表 4.4 检查项目表

CHECKITEM	主键	外键	类型	说明
ID	√	×	char	检查项目编号
NAME	×	×	char	检查项目名称
PRICE	×	×	float	检查单价

五、 部门信息表：部门信息表见表 4.5。

表 4.5 部门信息表

DEPT	主键	外键	类型	说明
ID	√	×	char	部门编号
NAME	×	×	char	部门名称
TEL	×	×	char	部门电话
POS	×	×	char	部门位置

六、 医生信息表：医生信息表见表 4.6。

表 4.6 医生信息表

DOCTOR	主键	外键	类型	说明
ID	√	×	char	医生编号
PASSWD	×	×	char	医生密码
NAME	×	×	char	医生姓名
GENDER	×	×	char	医生性别
ENTRY	×	×	date	医生入职年份
BIRTH	×	×	date	医生生日
SALARY	×	×	int	医生薪水
DEPTID	×	√	char	医生所在部门
TITLE	×	×	char	医生职称

七、 管理员信息表：管理员信息表见表 4.7。

表 4.7 管理员信息表

MANAGEF	主键	外键	类型	说明
ID	√	×	char	主管编号
PASSWD	×	×	char	密码
NAME	×	×	char	姓名
TITLE	×	×	char	职称
GENDER	×	×	char	性别
ENTRY	×	×	char	入职时间
BIRTH	×	×	char	出生年月
SALARY	×	×	int	薪水
DEPTID	×	√	char	部门

八、 药品库表：药品库表见表 4.8。

表 4.8 药品库表

MEDICINE	主键	外键	类型	说明
ID	√	×	char	药品编号
NAME	×	×	char	药品名
TYPE	×	×	char	药品类型
FUNCTION	×	×	char	药品功能
PRODUCEF	×	×	char	药品厂商
COMPONE	×	×	char	药品成分
PRICE	×	×	char	药品价格
STOKE	×	×	char	药品库存

九、 病人信息表：病人信息表见表 4.9。

表 4.9 病人信息表

PATIENT	主键	外键	类型	说明
ID	√	×	char	病人编号
PASSWD	×	×	char	密码
NAME	×	×	char	姓名
GENDER	×	×	char	性别
BIRTH	×	×	char	出生年月
TEL	×	×	char	电话
NOTE	×	×	char	说明

十、 挂号信息表：挂号信息表见表 4.10。

表 4.10 挂号信息表

REGISTER	主键	外键	类型	说明
PATIENTID	×	√	char	病人编号
DOCID	√	√	char	医生编号
DATETIME	√	×	date	日期
SERIAL	√	×	char	序号
FLAG	×	×	char	就诊标志

十一、 收费员信息表：收费员信息表见表 4.11。

表 4.11 收费员信息表

TREATEINFO	主键	外键	类型	说明
DOCID	√	√	char	医生编号
TIME	√	×	date	日期
PATIENTID	√	√	char	病人编号
DISEASE	×	×	char	诊疗疾病

十二、 病房入住情况表：病房入住情况表见表 4.12。

表 4.12 病房入住情况表

WARD	主键	外键	类型	说明
ID	√	×	char	病房编号
PATIENTID	×	√	char	病人编号
STARTTIME	√	×	char	开始启用时间
ENDTIME	×	×	char	结束使用时间

十三、 病房信息表：病房信息表见表 4.13。

表 4.13 病房信息表

WARDINFO	主键	外键	类型	说明
ID	√	×	char	病房编号
POS	×	×	char	病房地点
STATE	×	×	char	病房状态
FEE	×	×	float	病房单日价格

4.3 总体设计

系统总体设计：这部分将给出系统的总体设计，包括 C/S 架构，系统的工作流程，系统的功能模块等等

4.3.1 C/S 架构

如图 4.4 所示，为系统 C/S 架构图，在客户端我们可以使用四种账号进行登录操作，分别为：管理员账号，医生账号，病人账号和收费员账号。然后将账号密码传给编写好的连接类，利用这个连接类连接到本地 oracle 数据库。然后我们根据发送过来的账号密码与数据库钟存放的账号密码进行比较，若账号能在数据库中找到，且密码能和账号对的上，那么就可以成功登陆。登陆后如果需要对数据库进行增删改查等操作，那么首先实例化连接类，并写好相应的 sql 语句，利用连接类对数据库直接操作。如下图 4.5 所示：

图 4.5 C/S 架构图

4.3.2 功能模块组成

医院管理系统按用户类型可以分为四个主要的模块，他们分别为：系统管理员模块，医生模块，病人模块，收费员模块。如下图 4.6 所示。

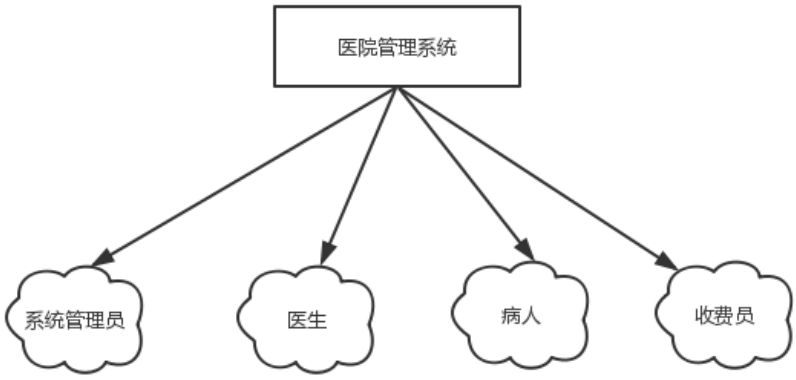


图 4.6 系统的功能模块图

系统管理员的功能模块可以划分为六大子模块，它们分别为：注册模块，个人信息模块，人事信息模块，药品信息模块，病床信息模块，财务信息模块。如下图 4.7 所示。

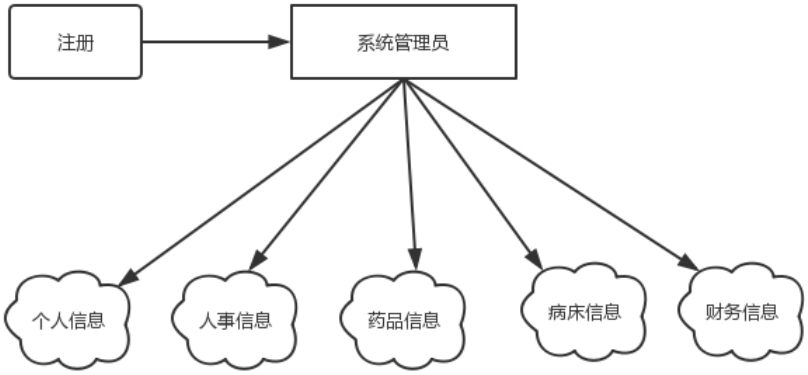


图 4.7 管理员的功能模块图

医生的功能模块可以划分为四大子模块，它们分别为：注册模块，诊疗模块，复诊模块，个人信息模块。如下图 4.8 所示。

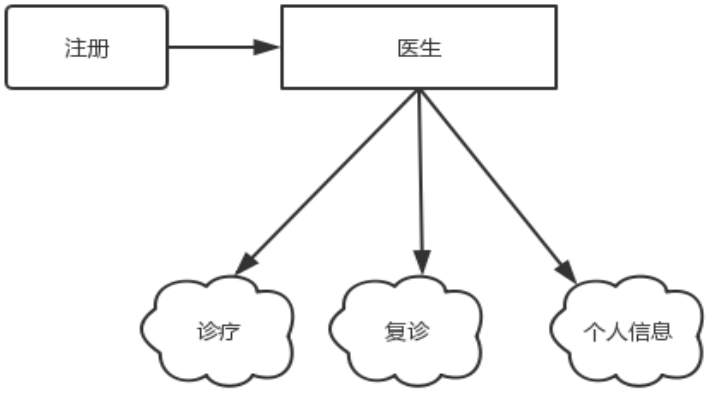


图 4.8 医生的功能模块图

病人的功能模块可以划分为四大子模块，它们分别为：注册模块，预约模块，个人信息模块，诊疗信息模块。如下图 4.9 所示。

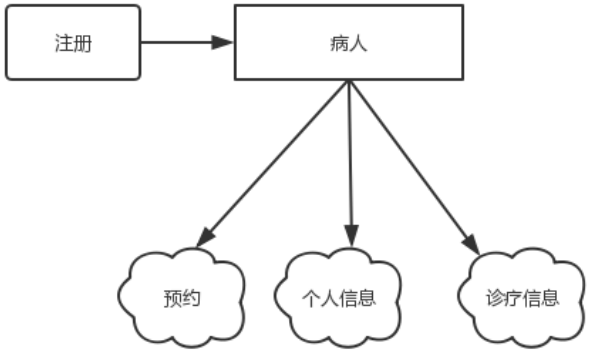


图 4.9 病人的功能模块图

收费员的功能模块可以划分为五大子模块，它们分别为：注册模块，挂号模块，取药模块，住院模块，个人信息模块。如下图 4.10 所示。

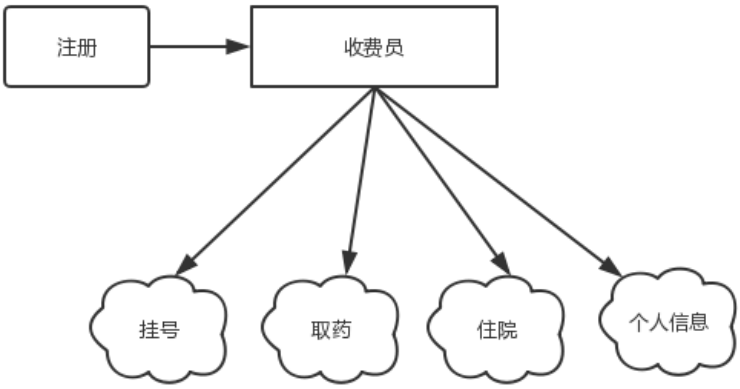


图 4.10 收费员的功能模块图

4.3.3 总体业务流程

医院管理系统比较复杂，简化后依然具有四个角色。每个角色的任务各不相同，总的来讲，可以分为两套业务流程，分别是：

1. 管理员对医院各个方面的管理。
2. 病人看病的整个流程。

下面具体介绍这两个流程：

管理员业务流程：管理员分为两类，一类是院长，具有最高权限，一类是部门管理，具有次高级权限。具体来讲，院长可以查看任何表，并管理医院所有人事信息。部门管理也可以查看任何表，但他只能管理自己部门的人事信息和收费人员的信息。管理员具体可以查看的医院相关信息有药库信息，病床信息和财务信息。药库信息和病房信息是直接显示出相关信息，财务信息由于需要分年月日

生成报表，所以需要管理员首先输入希望查询的日期，然后生成的报表就分部门显示各个部门的收入情况。最后一行再来一个医院总收入汇总。管理员业务流程图如下图 4.11 所示：

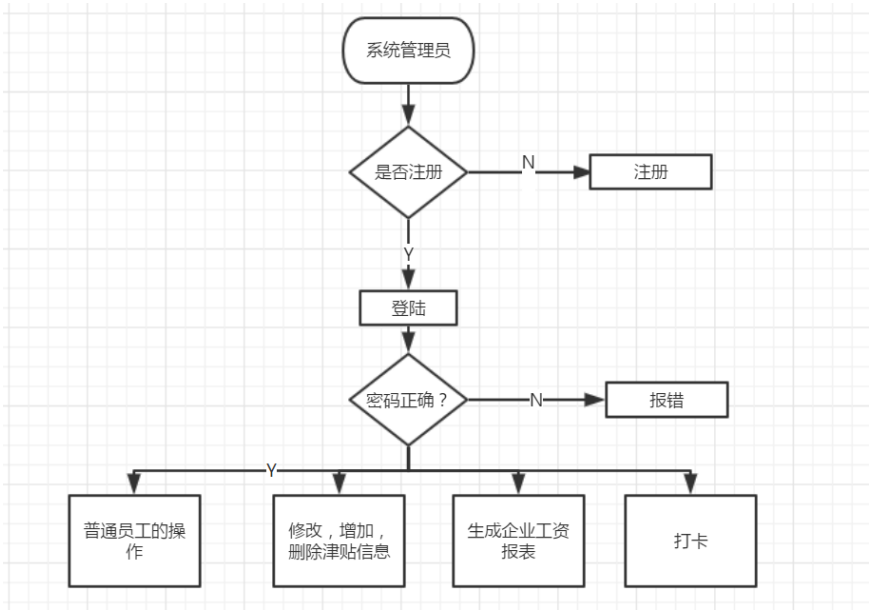


图 4.11 系统管理员业务流程

4.4 数据库设计

4.4.1 ER 图设计

四个角色之间的 ER 图：

如下图 4.12 所示，管理员主要属性有：编号、姓名，性别，年龄，职称，薪水等。病人主要属性有：编号、姓名、性别、年龄、电话、症状等。收费员主要具有的属性有：编号、姓名、性别、年龄、薪水等。医生主要具有的属性有：编号、姓名、性别、年龄、薪水、职称、部门等。

他们的关系如下：管理员可以查看病人的基本信息，可以管理医院的员工（收费员和医生）的信息并管理他们的职称或工资。病人可以预约医生，还可以到收费员处办理住院/出院手续，缴纳药品费和检查费。收费员可以办理缴费业务和住院/出院手续。医生可以为病人提供治疗，可以为病人开药，开检查，推荐其住院。

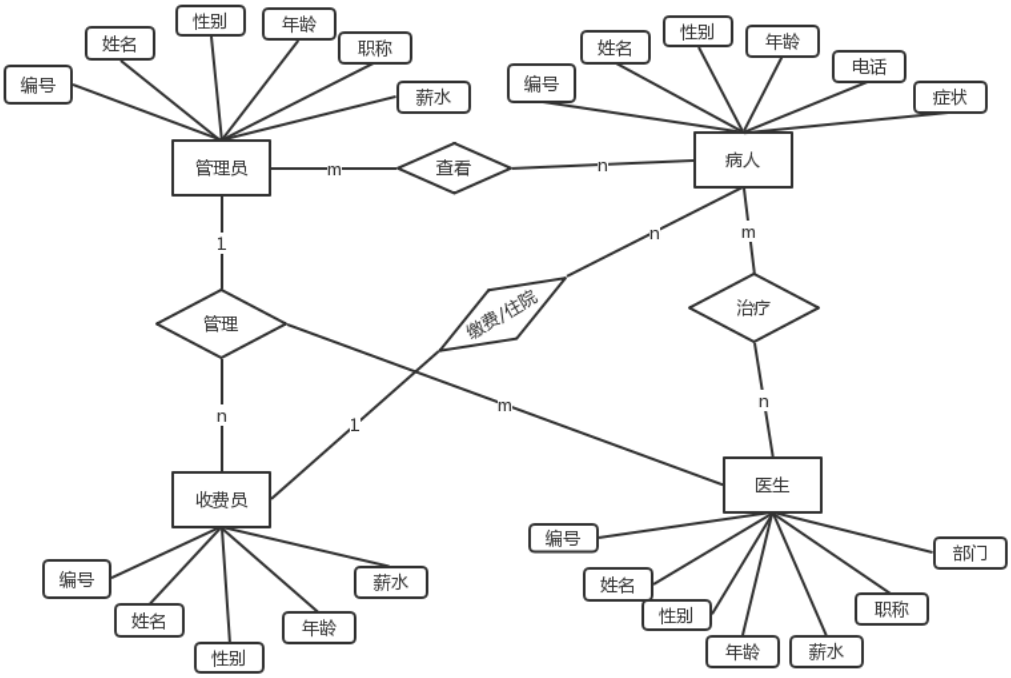


图 4.12 角色间 ER 图

管理员其他业务流程：

如下图 4.13 所示，管理员可以查看药库信息、医院财务信息、病房信息。药库信息。药库信息即为药品表中的内容；病房信息显示病房编号、病房地点、当前是否被占用、每日单价；医院财务信息可以分年、月、日查询，具体日期需要管理员进行输入，输入日期后可以查到各个部门的在特定日期的收入情况，在表的最后一行显示医院在该日期的总收入，实现财务信息的汇总。

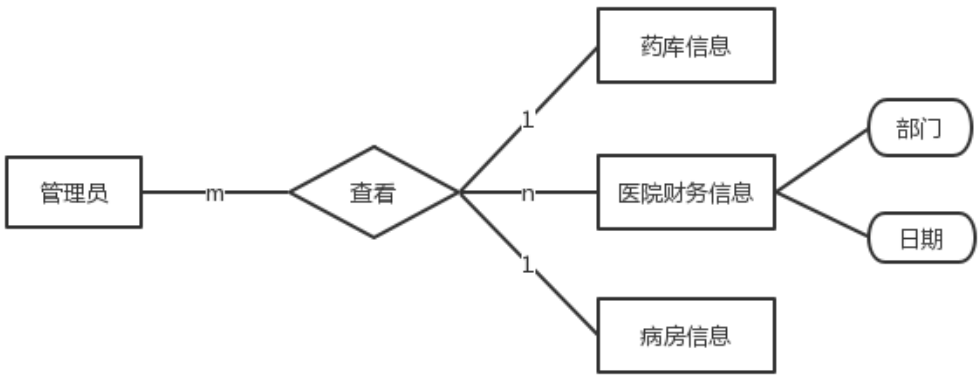


图 4.13 管理员业务流程图

4.4.2 数据库逻辑结构设计

表结构的说明：

本系统一共使用了十三张表，具体的逻辑结构说明，主码、外码说明已经在4.2.4 数据字典中做出了详细阐述，这里不再赘述。

视图的说明：

因为在很多表结构中，部门信息都存的是部门编号，而用户自使用的时候多用的是部门的名称，所以查找起来比较麻烦。为了简化查找，我创建了一个视图DOCINFO 。这个视图包含了医生的基本信息，有：医生编号、姓名、性别、部门名称和职称，这样查找起来就简便许多。视图结构见表 4.14。

表 4.14 视图结构

DOCINFO	说明
ID	医生编号
NAME	医生姓名
GENDER	性别
DEPT	部门名称
TITLE	医生职称

4.5 详细设计与实现

4.5.1 主干流程

下面将介绍本系统的三个主要流程：

本系统最主干的功能是病人看病的一系列流程。

病人看病业务流程如下：首先病人需要提前预约，在预约的时候，本系统提供按科室进行预约，为了简化，本系统一共提供十个科室，分别为：内科、外科、妇产科、放射科、儿科、脑科、骨科、神经科、麻醉科、五官科。病人点击按钮后 进入预约界面选择医生。科室选择界面如下图 4.14 所示：

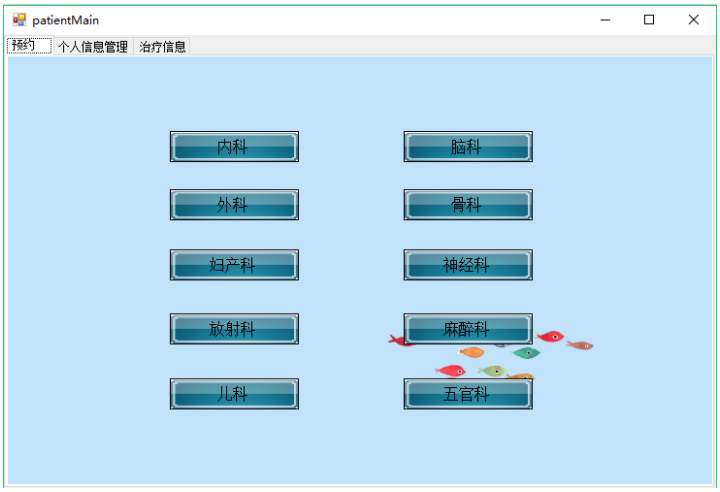


图 4.14 科室选择界面图

预约界面中，病人可以选择该科室的一个医生看病，或者按医生的职称进行分类后再进行选择。预约操作很简单，点击医生所在行，然后点击预约按钮即可。这里系统做了一些预约控制，就是每个病人一天之内只能预约某一位医生一次，避免病人误操作重复预约。当然病人如果想一天之内多次看同一名医生，直接去医生处复诊即可。这就是说，预约一天之内均有效。另一个控制是一个医生一天之内只能被预约 20 次，也就是说一个医生一天之内最多只能接待 20 名病人，当然 20 这个值也可以设置得更多。当某个医生被预约的人数多余 20 时，便不能再被新的病人预约。预约界面如下图 4.15 所示：

图 4.15 预约界面图

如果病人成功完成预约，那么他需要到医院收费处挂号，值得注意的是，预约的时候并不会规定病人就诊的次序，挂号的时候才会。也就是说，挂号的顺序决定了就诊的次序。病人到达挂号处后，需要向收费人员提供自己的就诊号和医生的工号，收费人员输入信息后会在预约表中查询，如果这名病人真的已经预约成功，那么就会将这条挂号信息录入，如果该病人未预约成功，那么挂号也将不成功。挂号界面图如下图 4.16 所示：

图 4.16 挂号界面图

如果病人挂号成功，那么他现在要做的就是排队等候医生看病。医生的就诊界面会依次载入已挂完号的病人信息，这些信息包括：就诊号，病人姓名，性别，年龄和病人症状。医生可以在这个界面执行就诊操作。其可以进行的操作有：填写该病人的诊疗结果，为病人开检查项目，为病人开药和让病人住院。一个病人诊疗之后医生可以点击右上方的 next 按钮，继续为挂号的另一个病人服务。医生诊疗界面图如下图 4.17 所示：

图 4.17 医生诊疗界面图

如果病人被开了检查项目，那么他会到医生处复诊。复诊界面与诊疗界面差别不大，唯一的区别就是复诊时首先需要输入病人的就诊号，输入就诊号后会显示病人的相关信息，便于医生医治。医生复诊界面图如下图 4.18 所示：

图 4.18 医生复诊界面图

到此，病人看医生的流程结束。

病人看病流程图如下图 4.19 所示：

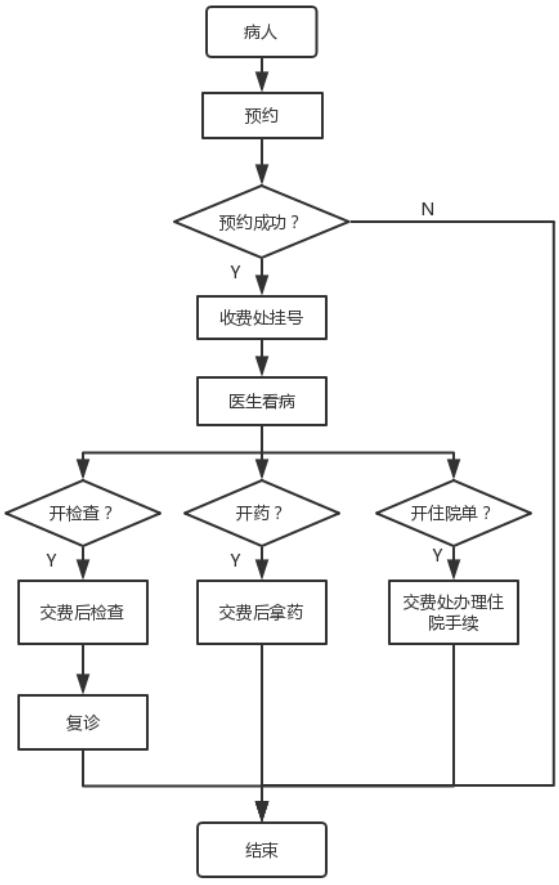


图 4.19 病人看病流程图

本系统中还有一些实用的功能，其中一个就是注册功能。

本系统支持所有四类角色的注册。注册界面灵活运用 c#图形构件库中的各个构件。下面分角色阐述注册功能。

管理员注册界面如下图 4.20 所示，管理员注册时需要填写的信息有：用户ID、密码、姓名、性别、出生日期、所在部门、职称和入职时间。管理员还可以上传自己的头像。上传时需要点击选择图片按钮，选择一张自己的照片上传。当然也可以选择不上传自己的照片，那样的话，系统会显示一张默认的头像。

图 4.20 管理员注册界面

病人注册界面如下图 4.21 所示，病人注册时需要填写的信息有：就诊卡号、密码、姓名、性别、年龄、电话。此外，为了方便医生诊疗，病人可以在既往病史框中选填自己的既往病史或者症状。这里的一个小 trick 是：当用户点击既往病史框时，“（选填）”会自动消失。

图 4.21 病人注册界面图

医生注册界面如下图 4.22 所示，医生注册时需要填写的信息有：工号、密码、姓名、性别、出生日期、所在部门、职称和入职时间。像管理员一样，医生还可以选择上传自己的头像。

图 4.22 医生注册界面

收费员的注册界面如下图 4.23 所示：这实际上和医生的注册界面并无两样，只是需要填写的信息种类不同。收费员在注册的时候需要填写的内容有：账号、密码、姓名、性别、出生日期、入职日期和选择照片。

图 4.23 收费员的注册界面图

本系统还有一个业务流程是管理员的人事信息管理流程：

当管理员登陆成功后点击人事信息按钮，会进入人事管理界面，如下图 4.24 所示。管理员在信息查看界面可以查看管理人员、医生、病人、收费员的信息。

	ID	NAME	GENDER	BIRTH	ENTRY	DEPTID	TITLE	SALARY
▶	001	李宇诩	女	1995/3/7	2008/8/8	01	内科主管	88888
	000	熊宇飞	男	1995/4/3	1999/4/9		院长	5000
	010	潘泽林	女	1996/11/3	2008/4/9	10	脑科主管	4565
	003	杨盈月	女	1995/5/3	2008/5/10	03	骨科主管	5345
	004	陈思雨	女	1996/4/4	2008/6/9	04	儿科主管	5346
	005	陈航	男	1996/5/6	2009/7/9	05	妇产科主管	4563
	006	张成龙	男	1994/6/3	2008/8/9	06	放射科主管	10000
	007	熊润龙	男	1995/2/3	2008/9/8	07	神经科主管	4444
	008	张润泽	女	1995/3/3	2009/10/3	08	麻醉科主管	4425
	009	卓振宇	男	1995/12/8	2008/4/2	09	五官科主管	6547
	002	海点	女	1996/4/3	2008/4/9	02	外科主管	10000
*								

图 4.24 信息查看页面图

在人事管理页面可以管理医院工作人员的职称或者薪水。如图 4.25，人事管理页面分为两类管理，一类是工资管理，另一类是职称管理。如果该管理员要进行工资管理，那么他首先要选择工作人员的类别（管理人员、医生、收费人员），然后输入该工作人员的工号。这时，如果点击查询按钮会弹出该员工当前的工资。然后，在修改工资框中输入修改后的工资，最后点击确认即可。当然，这里是有权限控制的。院长作为最高权限可以修改任何人的工资（包括他自己），但是部门管理只能修改收费人员和本部门的医生的工资。也就是说，他不能修改的工资包括：院长工资和管理员的工资（包括他自己），其他部门的工资。职称管理也是一样，院长可以修改医生的职称，本部门的管理也可以修改本部门医生的职称，但是管理员不能修改其他部门的医生的职称，否则就算越权。

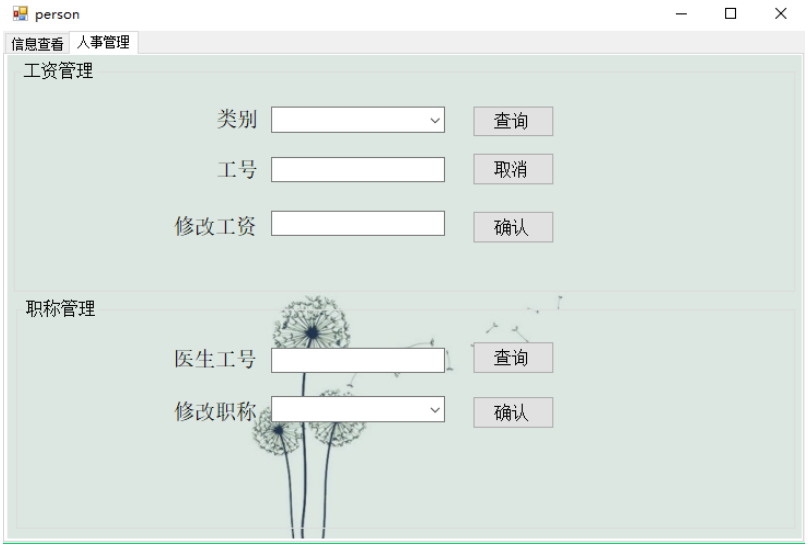


图 4.25 人事管理页面图

4.5.2 关键技术和算法

下面将介绍本系统的关键技术和实现算法：

连接类：

本系统各种信息都依托数据库储，所以许多操作需要连接数据库，但是如果让程序从头到尾一直连接数据库会造成系统资源不必要的浪费，这里解决办法就是创建一个连接类，需要对数据库操作的时候就实例化一个连接类，利用连接类中的函数进行数据库的一些操作，操作完成后断开系统与数据库的连接。连接的时候有一个需要注意的地方是，本实验的数据库是 ORACLE 数据库，要想在 VS 上连接上 ORACLE 数据库需要加入引用 Oracle.ManagedDataAccess。然后我们需要一个连接字，对我的系统来讲，连接字是"Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=LOCALHOST)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=orcl)));Persist Security Info=True;User ID=C##xxx;Password=xxx;"从连接字中，我们可以看出连接的数据库主机就是本机，当然如果需要连接外部数据库，我们可以输入对应的的 IP 地址。数据库的账号密码这里做了隐藏处理。

连接类代码如下所示：从代码中我们不难看出，这个连接类可以执行的功能有：数据库连接，断开连接，数据查询和一些非查询操作。其中查询函数返回类型是 DataSet 类型，即数据集。非查询操作函数返回的是受本次影响的行数，用以判断此次操作是否成功。

```
namespace HM
{
    class connect
    {   OracleConnection con;
        OracleCommand com;
        public void Link()
        {
            try
            {
                // 连接字
                string connstr = "Data
Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=LOCALHOST)
(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=orcl)));Persist Security
Info=True;User ID=C##yufei;Password=Yufei061000;";
                con = new OracleConnection(connstr);
                con.Open();
            }
            catch (Exception e)
            {
                Console.WriteLine("Error : {0}", e);
```

```
            }
        }
        public DataSet Search(string sql)
        {
            com = con.CreateCommand();
            com.CommandText = sql;//写好想执行的Sql语句
            //数据适配器，传输数据库数据
            OracleDataAdapter da = new OracleDataAdapter(com.CommandText,
con);
            DataSet ds = new DataSet();
            //填充数据集DataSet
            da.Fill(ds);
            return ds;
        }
        public int Execute(string sql)//返回受影响的行数
        {
            com = con.CreateCommand();
            com.CommandText = sql;//写好想执行的Sql语句
            Console.WriteLine("sql: {0}", sql);
            int cnt = com.ExecuteNonQuery();
            return cnt;
        }
        public void Disconn()//关闭数据库连接
        {
            con.Close();
            con.Dispose();
        }
    }
}
```

查询结果的显示方法：

在整个系统中，很多地方都需要显示查询到的数据。为了将这些数据很好地展示出来，我选用的是 C#中的 DataGridView 构件，这是 C# .NET Framework 中自带的构件，功能强大。为 DataGridView 添加需要显示的数据一般有两种方式：一种是直接绑定 dataset，另一种是手动逐行添加显示数据。本系统中两种方法都用到了。下面具体介绍两种数据绑定方法：

直接绑定法，代码如下：

```
private void drug_Load(object sender, EventArgs e)
        {
            connect conn = new connect();//连接类
            conn.Link();
            sql = string.Format(@"select * from medicine");//查询语句
            ds = conn.Search(sql);
```

```
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("未查到相关信息！");
                return;
            }
            dataGridView1.DataSource = ds.Tables[0];//数据绑定
           }
```

手动添加法，代码如下：

```
public void data()
        {
            //查到名称
            double all = 0;//总收入
            connect conn = new connect();//连接类
            conn.Link();
            ds = conn.Search(sql);//查询
            conn.Disconn();
            getName();//获取部门名称
            dataGridView1.Rows.Clear();
            for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {//将查询到的数据逐行逐列插入
                dataGridView1.Rows.Add();
                dataGridView1.Rows[i].Cells[0].Value =
ds.Tables[0].Rows[i][0].ToString();
                dataGridView1.Rows[i].Cells[1].Value = list[i];
                dataGridView1.Rows[i].Cells[2].Value =
ds.Tables[0].Rows[i][1].ToString();
                all += Convert.ToDouble(ds.Tables[0].Rows[i][1].ToString());
            }
            //获取总收入
            int lastRow = dataGridView1.RowCount;
            dataGridView1.Rows.Add();//增加一行
            dataGridView1.Rows[lastRow - 1].Cells[0].Value = "总收入";//将医
院总收入相关信息插入
            dataGridView1.Rows[lastRow-1].Cells[1].Value = "医院";
            dataGridView1.Rows[lastRow-1].Cells[2].Value = all.ToString();
            dataGridView1.ClearSelection();
           }
```

按钮的美化：

C#里的按钮不太美观，为了让按钮变得美观，我放弃使用按钮构件，转而使用 PictureBox 构件。不过 PictureBox 也是矩形，为了让它变为圆形，我们需要使用 GraphicsPath 类。利用这个类的 AddEllipse()函数，我们可以很容易将矩形的

PictureBox 构件变为圆形或者椭圆形。下面展示一段按钮美化代码。

```
private void beautify()
        {//美化  pictureBox界面，将其变为圆形
            GraphicsPath gp = new GraphicsPath();
            gp.AddEllipse(pictureBox1.ClientRectangle);
            Region region = new Region(gp);
            pictureBox1.Region = region;
            gp.Dispose();
            region.Dispose();

            GraphicsPath gp2 = new GraphicsPath();
            gp2.AddEllipse(pictureBox2.ClientRectangle);
            Region region2 = new Region(gp2);
            pictureBox2.Region = region2;
            gp2.Dispose();
            region2.Dispose();

            GraphicsPath gp3 = new GraphicsPath();
            gp3.AddEllipse(pictureBox3.ClientRectangle);
            Region region3 = new Region(gp3);
            pictureBox3.Region = region3;
            gp3.Dispose();
            region3.Dispose();

            GraphicsPath gp4 = new GraphicsPath();
            gp4.AddEllipse(pictureBox4.ClientRectangle);
            Region region4 = new Region(gp4);
            pictureBox4.Region = region4;
            gp4.Dispose();
            region4.Dispose();
        }
```

4.5.3 触发器的实现

为了使系统更加完善，我定义了一些触发器，这些触发器主要是用于工资管理。因此管理员、医生、收费员对应的表都有一个触发器。其中，我限制管理员的工资不得高于 10000，医生的工资不得高于 8000，收费员的工资不得高于 5000。这单个触发器的定义比较类似，这里只给出管理员的触发器创建代码。

管理员工资控制触发器定义如下：

```
create or replace
TRIGGER TRIGGER3
before update on MANAGER
referencing new as NEWEST
```

```
FOR EACH ROW
begin
if ( :NEWEST.SALARY > 10000)
        THEN    :NEWEST.salary := 5000;
    end if;
END;
```

4.6 系统测试

本系统功能丰富，很好地完成了实验要求的各项要求，除此之外，系统的界面也是经过精心设计的，美观大方，并且我还尝试用代码做出了圆形按钮，界面构件的摆放也是比较美观和谐，综合运用的 C#的许多构件。下面具体介绍系统的测试情况。

首先测试系统的注册功能，各个角色的注册功能比较相像，这里挑选两个比较有代表性的进行演示，分别是管理员的注册流程和病人的注册流程。

对于管理员来讲，界面上主要用到的输入构件有 TextBox，DateTimePicker 和 ComboBox。TextBox 是文本输入框，它可以接受输入的文本字符串；DateTimePicker 是日期选择构件，十分强大，通过它可以很方便地选择日期；ComboBox 是下拉菜单构件，可以在属性中添加下拉时显示的条目，运行时可以下拉选择条目。填写好注册信息后点击确认，系统会弹框显示“注册成功”。如果没有注册成功，系统会弹框显示“很遗憾，注册失败”。

如下图 4.26 所示，填写完整之后点击确定，可以看到已注册成功。

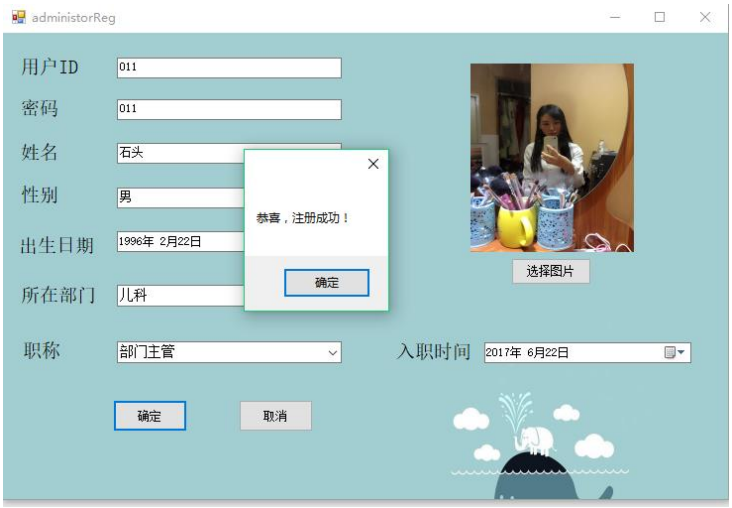


图 4.26 管理员注册演示图

接下来我们使用这个账户登录系统。点击个人信息按钮，可以看到我们刚才注册的信息完整的显示出来。这表明管理员的注册功能测试成功。管理员注册功

能验证图如图 4.27 所示：

图 4.27 管理员注册验证图

下面测试病人的注册功能。

对于病人注册界面来讲，与医院工作人员有所不同，这个界面主要用到了 TextBox 和 RichTextBox，TextBox 前面已经提到过，这里不再赘述。RichTextBox 控件允许用户输入和编辑文本的同时提供了比普通的 TextBox 控件更高级的格式特征。比如可以用它来设置文本粗体，文本颜色等，遗憾的是这些功能对于本系统并非必要，所以本系统未使用这些功能，只是简单使用了它多行编辑的功能。

同样，我们填写好注册信息，点击确定，弹出“恭喜，注册成功”的弹窗，这说明病人已注册成功。病人注册演示图如图 4.28 所示：

图 4.28 病人注册演示图

下面我们用这个新账户登录系统，以便进一步验证是否注册成功。登陆后，我们选择个人信息管理界面，可以看到账户的信息就是我们刚才注册的信息。这说明该病人注册成功。病人注册验证图如图 4.29 所示：

图 4.29 病人注册验证图

到此，注册功能测试结束。下面开始测试管理员的管理功能。我们随机选择一个管理员账户登录系统。这里我选择的是编号为 001 的内科管理员。登陆后，我们可以看到如下图 4.30 所示的界面。管理员的功能分为五大模块，分别为：个人信息模块，人事信息模块，药品信息模块，病床信息模块和财务信息模块。下面将分模块测试。

图 4.30 管理员主界面图

实际上系统中每个角色都有一个个人信息管理模块，这是为了让系统看起来更加完整。这里只展示管理员的个人信息管理流程，而其他角色的个人信息管理功能就不再赘述。

登录系统后点击个人信息按钮，我们可以看见如下图 4.31 所示的信息显示界面。

图 4.31 信息显示界面图

这时各个控件都是不能修改的，如果我们需要修改，则需要点击界面下方的修改按钮。当然，有的信息是不允许随便修改的，比如出生日期等。下面我将会把名字修改为李涵，照片也更换为其他照片，以验证程序的正确性。信息修改演示图如下图 4.32 所示：

图 4.32 信息修改演示图

再次进入个人信息界面，查看信息是否已经修改成功。如下图 4.33 所示，信息修改成功。

图 4.33 信息修改验证图

接着进行人事信息管理测试。人事信息管理界面分为两个页面，分别为信息查看和人事管理。信息查看页面的左上角有一个 ComboBox 下拉菜单栏，我们可以从这个菜单栏选择管理人员，医生，病人，收费人员中的一项，点击确认后查看相应信息。如图 4.34 所示，查看医院管理人员的信息。

ID	NAME	GENDER	BIRTH	ENTRY	DEPTID	TITLE	SALARY
001	李诵	女	1995/3/7	2008/8/8	01	内科主管	5000
000	熊宇飞	男	1995/4/3	1999/4/9		院长	5000
010	潘泽林	女	1996/11/3	2008/4/9	10	脑科主管	4565
003	杨盈月	女	1995/5/3	2008/5/10	03	骨科主管	5345
004	陈思雨	女	1996/4/4	2008/6/9	04	儿科主管	5346
005	陈航	男	1996/5/6	2009/7/9	05	妇产科主管	4563
006	张成龙	男	1994/6/3	2008/8/9	06	放射科主管	10000
007	熊润龙	男	1995/2/3	2008/9/8	07	神经科主管	4444
008	张润泽	女	1995/3/3	2009/10/3	08	麻醉科主管	4425
009	卓振宇	男	1995/12/8	2008/4/2	09	五官科主管	6547
002	海点	女	1996/4/3	2008/4/9	02	外科主管	10000
011	石头	男	1996/2/22	2017/6/22	04	儿科主管	

图 4.34 管理员信息查看演示图

另一个页面是人事管理页面。在这个页面中，我们可以修改权限内的工作人员的工资或者职称。下面我将演示工资管理过程。编号为 011 的医生是内科医生，刚好在本管理员管辖范围内。我们选择医生栏，然后输入 011，点击查询，然后弹出对话框，显示“该医生的工资为 3429”。工资查询演示图如图 4.35 所示：

图 4.35 工资查询演示图

下面我们将其工资修改为 3500。修改后再次查询，我们可以看到此时编号为 011 的医生工资已经成功修改为 3500。工资管理验证图如图 4.36 所示：

图 4.36 工资管理验证图

由于我们先前设置了工资修改的触发器，医生的工资不能高于 8000。如果高于 8000，那么工资就等于 8000。下面进行该触发器的测试。如下图所示，我们在修改工资栏输入 9000，点击确认。再次查询，工资为 8000。这说明工资修改触发器工作正常。触发器功能验证图如图 4.37 所示：

图 4.37 触发器功能验证图

如果医生不在该管理员所管理的部门呢？下面测试编号为 001 的医生，他不在当前管理员的部门，该管理员仍可以查询该医生的工资，但是却不能修改其工资。如下图 4.38 所示，强行修改会弹出“对不起，您无权修改其他部门医生工资”的弹窗。

图 4.38 工资修改权限验证图

职称管理与工资管理在实现方式上比较相似。权限管理上也是只能院长或者本部门的管理员修改其职称。这里不再赘述。

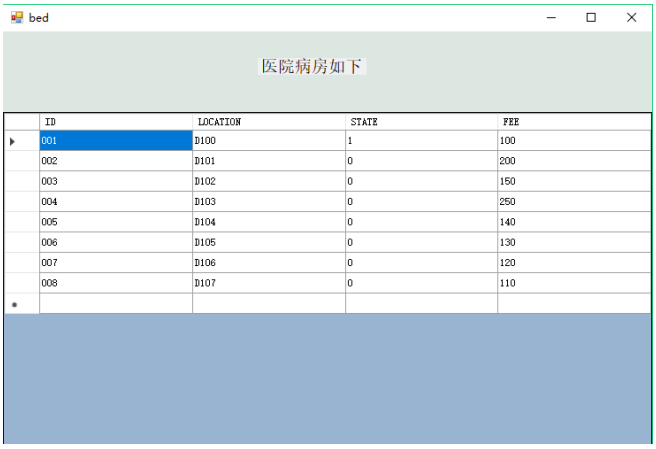
药品管理模块和病房管理模块实现方式类似，比较简单，这里只是将相关信息展示出来。

如下图 4.39 所示，药品管理界面上显示了药库中录入的各种药品的相关信息。

ID	NAME	TYPE	FUNCTION	PRODUCER	COMPONENT	PRICE	STOKE
001	A	处方药	中暑	石家庄制药	紫甘蓝	149	1275
002	B	处方药	呕吐	石家庄制药	鳖栗	141	1290
003	C	保健品	耳鸣	石家庄制药	调味剂	29	2
004	D	保健品	腹泻	石家庄制药	鳖栗	103	1380
005	E	保健品	中暑	石家庄制药	吊白块	40	1303
006	F	处方药	呕吐	石家庄制药	调味剂	47	1405
009	I	保健品	耳鸣	石家庄制药	紫甘蓝	181	1368
007	G	保健品	中暑	石家庄制药	紫甘蓝	64	1373
008	H	保健品	腹泻	石家庄制药	淀粉	94	1392

图 4.39 药品管理界面图

如下图 4.40 所示，病房管理界面显示了病房的相关信息。



ID	LOCATION	STATE	FEE
001	D100	1	100
002	D101	0	200
003	D102	0	150
004	D103	0	250
005	D104	0	140
006	D105	0	130
007	D106	0	120
008	D107	0	110

图 4.40 病房管理界面图

管理员的最后一个功能模块为财务信息查询。首先管理员需要在上面的输入框中输入想要查询的日期（支持年、月、日查询），然后点击确定，下面的显示框就会分部门显示各个部门的收入，最后一行会显示医院的总收入。如下图 4.41 所示，输入 2017，显示 2017 年的总收入情况。

部门编号	部门名称	收入
01	内科	1094
02	外科	1111
03	骨科	87
04	儿科	111
05	妇产科	87
06	放射科	282
07	神经科	149
08	麻醉科	149
09	五官科	100
10	脑科	50
总收入	医院	3220

图 4.41 医院年收入图

再输入 2017-6，查询 2017 年 6 月的收入情况。月收入情况图如下图 4.42 所示：

图 4.42 医院月收入图

最后输入 2017-6-10，查询 2017 年 6 月 10 号的收入情况。日收入情况图如下图 4.43 所示：

图 4.43 医院日收入图

到此，管理员的管理功能测试结束。下面通过病人的看病流程测试医院的看病功能。这将会涵盖病人，医生，收费员的所有功能。

首先，我们需要登录一个病人账号。为了看病，该病人需要先预约医生。预约界面如下图 4.44 所示，系统提供了十个科室，病人可以选择想看的科室，这里选择内科。

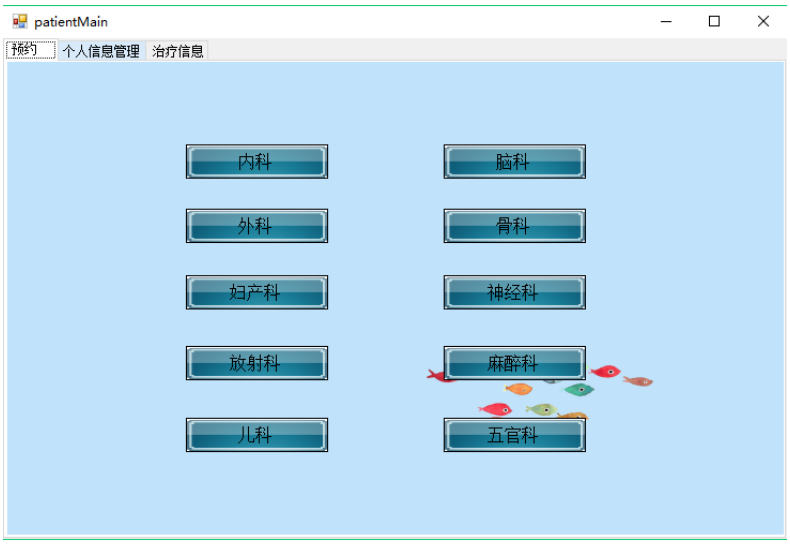


图 4.44 科室选择界面图

点击内科按钮后，我们进入医生信息显示界面。界面上显示该科室的全部医生。如果病人要预约某一个医生，他只需要点击该医生所在行，然后点击下方的预约按钮即可。病人此时可能需要记住医生的工号，因为以后会用到。预约医生界面图如下图 4.45 所示：

图 4.45 预约医生界面图

除此之外，为了便于病人查询，系统提供了按职称查询的功能。病人只需要点击右边的“主任医师”，“副主任医师”，“主治医师”或“住院医师”按钮，就可以很大程度上缩小查找范围。比如说点击主任医师按钮，如下图 4.46 所示，左边显示框中只出现职称为主任医师的医生列表。

图 4.46 预约演示图

不过就像 4.5.1 主干流程中提到的那样，病人并不是总能预约成功，如果病人当日已经预约过该医生或者该医生今日预约量已经满了，那么该病人就不能成功预约。如下图所示，该病人当日已经预约过 011 号医生了，那么当他再次预约时就会弹出“您今日已预约过该医生，请勿重复预约” 的对话框。预约失败图如下图 4.47 所示：

图 4.47 预约失败图

如果病人成功预约到想看的医生，那么接下来他要做的就是去收费员处挂号。挂号单上的号码就是他看病的顺序。我们现在登录一个收费员的号。如下图 4.48 所示，在挂号页面输入病人就诊号和医生工号后，点击确定，弹出对话框显示就诊序号，可以看出该病人的就诊号是 1 号。后面的病人在今日挂相同医生的号时，这个号码就依次增大。

图 4.48 挂号图

挂完号后，病人要做的就是等到医生叫自己的号，然后开始看病。那么现在我们需要登录一个医生的号。这个医生的工号为 011。登陆后我们便可以看到如下图 4.49 所示的界面。该界面的诊疗页面显示的就是我们之前挂号的 001 号病人的信息。图中病人信息显示在左上角，右上角是病人在注册的时候输入的既往病史或者症状。下方是医生的操作界面，分为四个部分：诊疗结果，检查，常规治疗，住院治疗。在诊疗结果页面中，医生可以填写病人的诊断结果并提交到数据库。右侧还有一个“清空”按钮，点击后输入框就会自动清空，目的是方便医生编辑。检查页面可以被医生用来开检查项目。而常规治疗一般是医生为病人开药。最后一个页面是住院，如果医生认为该病人需要住院，那么医生可以点击住院按钮，推荐病人住院治疗。

图 4.49 医生诊疗界面图

下面具体演示。首先我们将诊疗结果写为“头疼脑热”。诊疗图如下图 4.50 所示：

图 4.50 诊疗结果图

然后为其开一个检查“量体温”，然后点击确认将信息提交至数据库。这里还有几个按钮，分别是删除和清空。删除的作用是删除选中行的检查项目，清空是清除所有已开的项目。这里不再具体演示。检查项目图如图 4.51 所示：

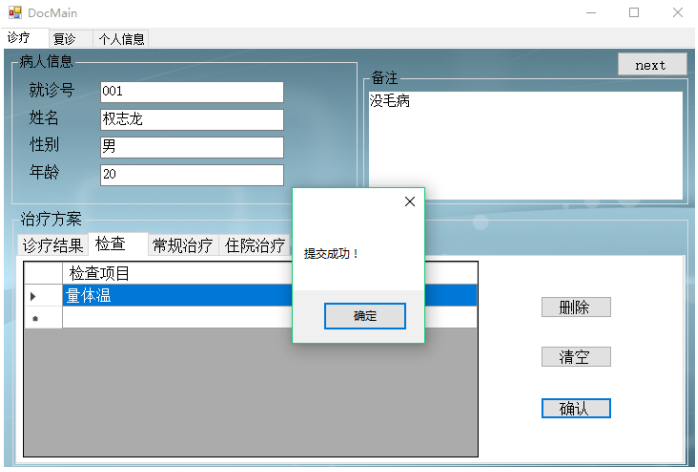


图 4.51 检查项目图

为病人开两种药，“A”四盒和“B”一盒。如下图所示，在输入框中输入相关信息，然后点击确定按钮。这里还有几个按钮，分别是删除和清空。删除的作用是删除选中行的药品，清空是清除所有已开的药品。这里不再具体演示。开药品图如图 4.52 所示：

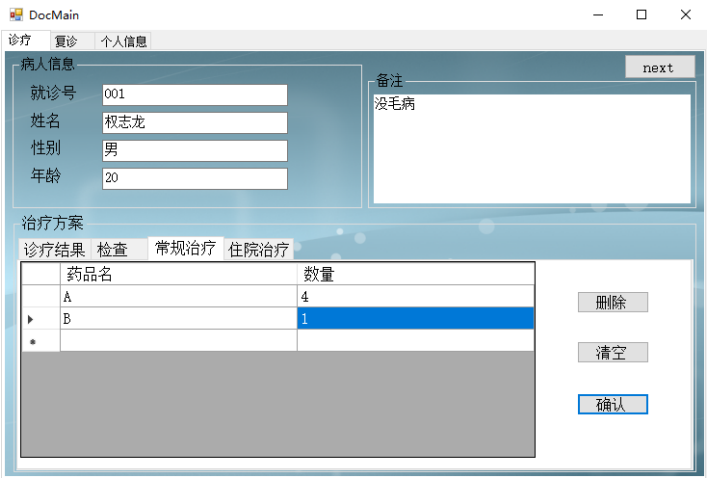


图 4.52 开药品图

最后推荐其住院治疗。如下图 4.53 所示，医生点击“住院按钮”，弹出对话框，提示让患者到收费处办理入院手续。

图 4.53 推荐住院演示图

到此，医生的工作告一段落。他可以点击右上角的 next 按钮，进入下一个病人的治疗过程，但我们现在只关心编号为 001 号的病人。接下来该病人需要到收费处进行一系列操作。现在我们使用收费员账号登录，并进入取药页面。输入病人的就诊号和医生的工号之后，点击查找，我们可以看见刚才医生为该病人开的检查和药品都显示在缴费清单中。并且页面右上角显示该病人此次应缴费用和他所有的未缴费用（因为该病人可能看过其他医生）。该病人交钱后，收费员点击缴费按钮，那么这部分钱就算缴纳了。当然，如果不输入医生工号进行查询，那么查到的将是该病人所有的未缴纳费用。这里我们演示缴纳 011 医生为其开的单子。缴纳费用图如图 4.54 所示：

图 4.54 缴纳费用图

点击缴费按钮后，如下图 4.55 所示，弹出缴费成功的窗口。

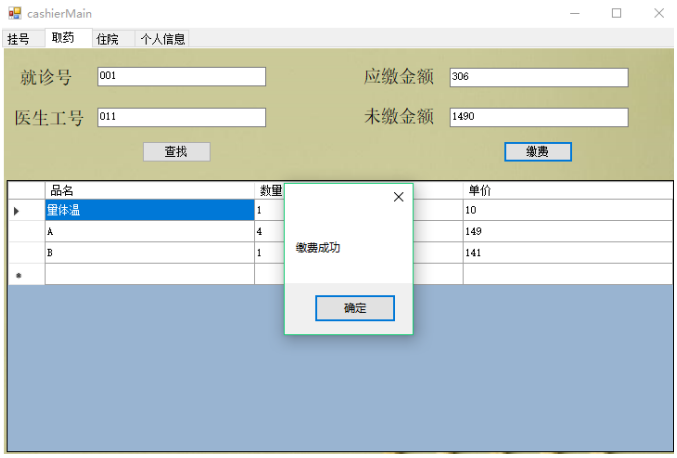


图 4.55 缴费成功演示图

再次查询，已经没有相关信息。这说明缴费成功。但还是有未缴纳金额，因为该病人还在其他医生处开了东西。我们可以不输入医生工号查询，然后缴费，这样就可以缴纳所有费用了。由于这个功能与只缴纳某个医生的费用功能原理上相同，这里不再重复演示。如下图 4.56 所示，病人已缴纳相关费用。

图 4.56 缴费完毕验证图

然后，开住院手续。如下图 4.57 所示，输入病人的就诊号 001，点击确定。右边床位框中便为该病人分配一个床位，从框中我们可以读到“该病人的床位号是 02，位置在 D101，每日住院费是 200。”

图 4.57 病房分配图

然后，病人检查完之后需要复诊，那么他还是得回到 011 号医生处。这时，我们回到医生界面，并进入到复诊页面。医生首先需要在右上角输入病人就诊号。然后点击确定。那么界面就显示该病人的相关信息。如下图 4.58 所示。这时右上角显示的是上次诊断结果，而不是病人自己的备注。医生操作部分与诊疗界面并无区别，这里不再赘述。

图 4.58 复诊演示图

到此，完成了系统所有主体功能的测试。但还有一些小功能没有演示，比如说医生预约人数多于 20 人就无法预约等等细节。为避免拖沓，这里不再演示这些细节功能。读者可以通过阅读代码查看实现细节。

4.7 系统设计与实现总结

医院管理系统表面上看比较简单，但实际深究的话非常复杂。角色比较多，需要注意的细节也不少。我的整个设计流程基本上是按照老师建议的流程来做的。

1. 需求分析。由于我平时去医院比较少，对医院了解不够，所以这一步花费了较多时间。尤其是表的设计，反反复复修改了很多次。为了将看病流程完美实现，我还咨询了我的学医学的朋友，向他们请教看病流程，最终大体上明白了如何设计系统。
2. 总体设计。我设计的是 C/S 架构的系统。在这一步，我并没有急于编写代码，而是先在脑海中思考我的系统的角色和各个角色承担的任务。最终，我确立了四个角色，这比很多同学的两个角色多出一倍，但是为了系统的完整性，我还是没有动摇。
3. 编写代码。由于系统比较复杂，代码量非常大，我在编写的时候尽量做到代码复用，但还是花费了非常多的时间编写。在编写的过程中，我不仅关注功能的实现，还十分注重界面的美观大方。每一个界面都经过精心设计，这也得到了助教和同学们的欣赏。
4. 系统测试。在测试阶段，时不时会蹦出几个小 BUG，但是无伤大雅。C#是一门比较成熟的语言，提供了比较准确的 BUG 提示，因此，调试时并不算难。

4 课程总结

在数据库的课程实践中，我们一共有三大任务。第一大任务是软件功能学习。第二大任务是 SQL 语句的练习，第三大任务就是数据库应用系统设计，三个任务循序渐进，难度依次递增，前一个任务是后一个任务的铺垫。下面分任务阐述主要工作。

1. 软件功能学习。由于大家都是第一次接触到数据库软件，所以需要首先学习如何使用软件。常用的数据库有 MS SQLServer 和 ORACLE，还有的同学使用的是达梦数据库。我最后选择的是 MS SQLServer。在第一个实验中，我学会了如何建表并插入数据，两种备份数据库的方法和在数据库上新增用户的技巧。第一个实验并不难，它旨在让我们熟悉软件的使用。
2. SQL 语句的练习。这个实验相对于第一个实验增加了难度。实验要求我们建立数据库之后再编写若干条查询语句对数据库进行查询，还要求我们创建触发器和了解系统的查询性能和分析功能。Sql 语句基本上涵盖

了课本上教的各种类型的操作，有嵌套查询，分组，排序等。因为当时对 SQL 语句的编写还不太熟练，所以检查那晚写了较长时间的 sql 语句。后来由于时间紧迫，就没有检查触发器。不过后来回去后，我没有放弃，最后完成了触发器的创建，这在后来的第三个实验中也有用到。

3. 数据库应用系统设计。第三个实验是本次课程实践的重头戏。也是我倾注心血最多的地方。为了完成它，我花了一周的时间。在选题的过程中，我没有跟风选择工资管理系统，而是选择了医院管理系统。这个题看似简单，实际上非常复杂。光是建表就花了一天半，还咨询了我的学医的朋友。我是在心中有了大概构想之后才开始编写代码。实际上，我之前并未接触过 C#，是因为这个实验才开始学习的，算是边学语言边完成实验。不过在学习 C#的时候，我发现 C#这门语言与 java 有很多相似之处，这极大地减轻了我的学习负担，因为我自学过 java。幸运的是，我花了不到三天时间就可以比较熟练地编写 C#代码了。开始编写代码后，进度就相当快了，很快我就绘制好了界面程序，然后再添加功能代码。在编写的过程中，我的确遇到了一些难题，比如我的数据库用到了 to_date（）函数。我们一般的日期格式是：yyyy-mm-dd，而 oracle 数据库在存储日期的时候使用的格式是：dd-mon-rr。这给我造成了一些困扰，不过在仔细查阅网上资料后很快就解决了。其次，我认为编写程序也同时复习了我们课上所学的 sql 语句。一开始，我的 sql 语句只能单表查询，后来熟练之后，不仅可以多表查询，还用到了分组，排序，统计求和等等较高级的操作。

总的来讲，这次课程实践不仅在很大程度上提升了我的数据库的专业知识，还促使我学习了一门新的编程语言，丰富了我组织编写大型应用程序的经验。

我对实验的一点小小的建议是：希望能够增加检查实验的时间或者增加检查实验的助教。一个助教要检查很多同学，这会导致检查的不仔细或者没时间检查剩下的同学。

最后感谢老师辛苦安排的实验！

附录

这里附上部分代码，完整代码请见文件夹。

logUI.cs

```
namespace HM
{
    public partial class logUI : Form
    {
        public logUI()
        {
            InitializeComponent();
            Show();
            beautify();
        }

        private void beautify()//美化  pictureBox界面，将其变为圆形
        {
            GraphicsPath gp = new GraphicsPath();
            gp.AddEllipse(pictureBox1.ClientRectangle);
            Region region = new Region(gp);
            pictureBox1.Region = region;
            gp.Dispose();
            region.Dispose();
            GraphicsPath gp2 = new GraphicsPath();
            gp2.AddEllipse(pictureBox2.ClientRectangle);
            Region region2 = new Region(gp2);
            pictureBox2.Region = region2;
            gp2.Dispose();
            region2.Dispose();
            GraphicsPath gp3 = new GraphicsPath();
            gp3.AddEllipse(pictureBox3.ClientRectangle);
            Region region3 = new Region(gp3);
            pictureBox3.Region = region3;
            gp3.Dispose();
            region3.Dispose();
            GraphicsPath gp4 = new GraphicsPath();
            gp4.AddEllipse(pictureBox4.ClientRectangle);
            Region region4 = new Region(gp4);
            pictureBox4.Region = region4;
            gp4.Dispose();
            region4.Dispose();
        }
        private void pictureBox1_Click(object sender, EventArgs e)
```

```
        {//点击进入管理员登录界面
            administrator ad = new administrator();
           }

        private void pictureBox2_Click(object sender, EventArgs e)
        {//点击进入病人登录界面
            //MessageBox.Show("hah,this is patient");
            patient pa = new patient();
        }

        private void pictureBox3_Click(object sender, EventArgs e)
        {//点击进入医生登录界面
            Doctor doc = new Doctor();
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {//点击进入收费员登录界面
            Cashier cash = new Cashier();
        }
    }
}
```

AdministratorMain.cs

```
namespace HM
{
    public partial class administratorMain : Form
    {
        string user;
        public administratorMain(string user)
        {
            InitializeComponent();
            Show();
            this.user = user;
            buttonBeauty();
        }
        public void buttonBeauty()//美化按钮
        {
            GraphicsPath gp = new GraphicsPath();
            gp.AddEllipse(pictureBox1.ClientRectangle);
            Region region = new Region(gp);
            pictureBox1.Region = region;
            gp.Dispose();
            region.Dispose();

            GraphicsPath gp2 = new GraphicsPath();
```

```
        gp2.AddEllipse(pictureBox2.ClientRectangle);
        Region region2 = new Region(gp2);
        pictureBox2.Region = region2;
        gp2.Dispose();
        region2.Dispose();

        GraphicsPath gp3 = new GraphicsPath();
        gp3.AddEllipse(pictureBox3.ClientRectangle);
        Region region3 = new Region(gp3);
        pictureBox3.Region = region3;
        gp3.Dispose();
        region3.Dispose();

        GraphicsPath gp4 = new GraphicsPath();
        gp4.AddEllipse(pictureBox4.ClientRectangle);
        Region region4 = new Region(gp4);
        pictureBox4.Region = region4;
        gp4.Dispose();
        region4.Dispose();

        GraphicsPath gp5 = new GraphicsPath();
        gp5.AddEllipse(pictureBox5.ClientRectangle);
        Region region5 = new Region(gp5);
        pictureBox5.Region = region5;
        gp5.Dispose();
        region5.Dispose();
}

private void pictureBox1_Click(object sender, EventArgs e)
{//个人信息
        adPersonal adp = new adPersonal(user);
}

private void pictureBox2_Click(object sender, EventArgs e)
{//人事信息
        person per = new person(user);
}

private void pictureBox3_Click(object sender, EventArgs e)
{//药品信息
        drug drug = new drug();
}

private void pictureBox4_Click(object sender, EventArgs e)
```

```
        {//病床信息
            bed bed = new bed();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {//财务信息
            finance fi = new finance();
        }
    }
}
```

PatientMain.cs

```
namespace HM
{
    public partial class patientMain : Form
    {
        string user;
        string ID;
        string passwd;
        string name;
        string gender;
        string birth;
        string tel;
        string note;
        string currentYear;

        public patientMain(string user)
        {
            this.user = user;
            InitializeComponent();
            Show();
        }
       private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
        {//选项卡切换
            if (tabControl1.SelectedIndex == 1) //个人信息
            {
                string sql = string.Format("SELECT * FROM PATIENT
      WHERE ID = '{0}'", user);
                connect conn = new connect();
                conn.Link();
                DataSet ds = conn.Search(sql);
                conn.Disconn();
                ID = ds.Tables[0].Rows[0]["ID"].ToString();
                textBoxID.Text = ID;
                textBoxPW.Text =ds.Tables[0].Rows[0]["PASSWD"].ToString();
```

```
                textBoxName.Text = ds.Tables[0].Rows[0]["NAME"].ToString();
                comboBoxGender.Text  =ds.Tables[0].Rows[0]["GENDER"].ToS
tring();
                birth = ds.Tables[0].Rows[0]["BIRTH"].ToString();
                currentYear = DateTime.Now.Year.ToString();
                int ageInt = Convert.ToInt32(currentYear) -
Convert.ToInt32(birth);
                birth = ageInt.ToString();
                textBoxAge.Text = birth;
                textBoxTel.Text = ds.Tables[0].Rows[0]["TEL"].ToString();
                richTextBox1.Text = ds.Tables[0].Rows[0]["NOTE"].ToString();
            }
        }
        public DataSet docInfo(string deptstr)
        {//获取医生信息
            string sql = @"select ID  编号  ,NAME  姓名,GENDER  性别,DEPT
部门,TITLE  职称
                                FROM DOCINFO WHERE DEPT = '" + deptstr + "'";
            connect conn = new connect();
            conn.Link();
            DataSet ds = conn.Search(sql);
            conn.Disconn();
            return ds;
        }
        private void button1_Click(object sender, EventArgs e)
        {//内科医生按钮
            DataSet ds = docInfo("内科");
            docInfo docinfo = new docInfo(ds,user);
        }
        private void button2_Click(object sender, EventArgs e)
        {//外科按钮
            DataSet ds = docInfo("外科");
            docInfo docinfo = new docInfo(ds, user);
        }
        private void button3_Click(object sender, EventArgs e)
        {//妇产科
            DataSet ds = docInfo("妇产科");
            docInfo docinfo = new docInfo(ds, user);
        }
        private void button4_Click(object sender, EventArgs e)
        {//放射科
            DataSet ds = docInfo("放射科");
            docInfo docinfo = new docInfo(ds, user);
        }
```

```
private void button5_Click(object sender, EventArgs e)
{//儿科
    DataSet ds = docInfo("儿科");
    docInfo docinfo = new docInfo(ds, user);
}
private void button6_Click(object sender, EventArgs e)
{//脑科
    DataSet ds = docInfo("脑科");
    docInfo docinfo = new docInfo(ds, user);
}
private void button7_Click(object sender, EventArgs e)
{//骨科
    DataSet ds = docInfo("骨科");
    docInfo docinfo = new docInfo(ds, user);
}
private void button8_Click(object sender, EventArgs e)
{//神经科
    DataSet ds = docInfo("神经科");
    docInfo docinfo = new docInfo(ds, user);
}
private void button9_Click(object sender, EventArgs e)
{//麻醉科
    DataSet ds = docInfo("麻醉科");
    docInfo docinfo = new docInfo(ds, user);
}
private void button10_Click(object sender, EventArgs e)
{//五官科
    DataSet ds = docInfo("五官科");
    docInfo docinfo = new docInfo(ds, user);
}
private void buttonModif_Click_1(object sender, EventArgs e)
{
    textBoxID.ReadOnly = false;
    textBoxPW.ReadOnly = false;
    textBoxName.ReadOnly = false;
    comboBoxGender.Enabled = true;
    textBoxAge.ReadOnly = false;
    textBoxTel.ReadOnly = false;
    richTextBox1.ReadOnly = false;
}
public void getInfo()
{
    ID = textBoxID.Text;
    passwd = textBoxPW.Text;
```

```
            name = textBoxName.Text;
            gender = comboBoxGender.Text;
            birth = textBoxAge.Text;
            tel = textBoxTel.Text;
            note = richTextBox1.Text;
            //计算年龄
            //currentYear = DateTime.Now.ToString("yyyy");
            Console.WriteLine("{0}",Convert.ToInt32(currentYear));
            int ageInt = Convert.ToInt32(currentYear) - Convert.ToInt32(birth);
            birth = ageInt.ToString();
        }
        public void update()
        {
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"update PATIENT set ID = '{0}',
PASSWD = '{1}',NAME = '{2}',GENDER = '{3}',BIRTH = '{4}',TEL = '{5}',NOTE
= '{6}' where ID = '{7}'",
                ID, passwd, name, gender, birth,tel,note,user);
            int cnt = conn.Execute(sql);
            conn.Disconn();
            if (cnt > 0)
            {
                MessageBox.Show("恭喜，修改成功！");
            }
            else if (cnt <= 0)
            {
                MessageBox.Show("很遗憾，修改失败，请重试！");
            }
        }
        private void buttonSubmit_Click(object sender, EventArgs e)
        {//提交
            getInfo();
            if (ID.Length == 0 || passwd.Length == 0 || name.Length == 0 ||
                gender.Length == 0 || birth.Length == 0 || tel.Length == 0)
            {
                MessageBox.Show("请将信息填写完整后再保存！");
                return;
            }
            update();
        }
        private void button12_Click(object sender, EventArgs e)
        {//按医生查询
            string doctorid = textBox1.Text;
```

```
                if(doctorid.Length == 0)
                {
                    MessageBox.Show("请输入医生工号！");
                    return;
                }
                string sql = string.Format(@"select item,amount,price,datetime fro
m bill where patientid = '{0}' and docid = '{1}'",user,doctorid);
                connect cc = new connect();
                cc.Link();
                DataSet ds = cc.Search(sql);
                cc.Disconn();
                if(ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("未查到先相关信息！");
                    return;
                }
                dataGridView1.DataSource = ds.Tables[0];
                cc.Disconn();
            }
        }
}
```

DocMain.cs

```
namespace HM
{
    public partial class DocMain : Form
    {
        string user;
        string patientId;
        string patientname;
        string patientgender;
        string patientage;
        string patientnote;
        DataGridViewRow currentROW;
        string date;
        string disease;
        ArrayList list1,list2,list3;
        private string picPath;
        private string targetPath;
        private string passwd;
```

```
 private string name;
 private string gender;
 private string entry;
 private string birth;
 DataSet ds;
 string pid;
 string previousDATE;

 public DocMain(string user)
 {
        this.user = user;
        InitializeComponent();
        Show();
        init();
 }
 public void init()
 {
        date = DateTime.Now.ToString("yyyy-MM-dd");
        list1 = new ArrayList();
        list2 = new ArrayList();
        list3 = new ArrayList();
 }
 private void DocMain_Load(object sender, EventArgs e)
 {//首先查找当前病人的 ID 号
        getPatientID();
        //根据病人 ID 号得到其基本信息
        getPatientINFO();
        textBoxID.Text = patientId;
        textBoxNAME.Text = patientname;
        textBoxGENDER.Text = patientgender;
        textBoxAGE.Text = patientage;
        richTextBox1.Text = patientnote;
 }
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
 {
```

```
            if (tabControl1.SelectedIndex == 0) //诊疗
            {   //首先查找当前病人的 ID 号
                getPatientID();
                //根据病人 ID 号得到其基本信息
                getPatientINFO();
                textBoxID.Text = patientId;
                textBoxNAME.Text = patientname;
                textBoxGENDER.Text = patientgender;
                textBoxAGE.Text = patientage;
                richTextBox1.Text = patientnote;
            }
            if (tabControl1.SelectedIndex == 2)
            {//个人信息
                //显示医生个人信息
                DisDocInfo();
            }
        }
        public void next()//下一个病人
        {
            //首先查找当前病人的 ID 号
            getPatientID();
            //根据病人 ID 号得到其基本信息
            getPatientINFO();
            textBoxID.Text = patientId;
            textBoxNAME.Text = patientname;
            textBoxGENDER.Text = patientgender;
            textBoxAGE.Text = patientage;
            richTextBox1.Text = patientnote;
        }
        public void getPatientID()
        {
            string sql = string.Format(@"SELECT  PATIENTID,serial from register where DOCID = '{0}' and SERIAL = (select min(SERIAL) from register where DOCID = '{1}' and FLAG = '0')", user,user);
            connect conn = new connect();
```

```
            conn.Link();
            DataSet ds = conn.Search(sql);
            conn.Disconn();
            if (ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("您当前没有就诊任务！");
                return;
            }
            if (ds.Tables[0].Rows[0][1].ToString().Equals(20))
            {
                MessageBox.Show("这是您今日最后一个病人！");
            }
            if (ds.Tables[0].Rows[0][0].ToString().Length > 0)
            {
                patientId = ds.Tables[0].Rows[0][0].ToString();
            }
        }
        public void getPatientINFO()//获取病人信息
        {
            string sql = string.Format(@"SELECT NAME,GENDER ,BIRTH,
NOTE  from PATIENT where ID = '{0}'",patientId);
            connect conn = new connect();
            conn.Link();
            DataSet ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count > 0)
            {
                patientname = ds.Tables[0].Rows[0]["NAME"].ToString();
                patientgender = ds.Tables[0].Rows[0]["GENDER"].ToString();
                string birth = ds.Tables[0].Rows[0]["BIRTH"].ToString();
                patientnote = ds.Tables[0].Rows[0]["NOTE"].ToString();
                string currentYear = DateTime.Now.Year.ToString();
                patientage = (Convert.ToInt32(currentYear) - Convert.ToInt32
(birth)).ToString();
            }
```

```
        }

        private void button3_Click(object sender, EventArgs e)//提交治疗信息
        {
            getMedicineInfo();
            connect conn = new connect();
            conn.Link();
            string sql0 = string.Format(@"select max(SERIAL) from BILL
                group by PATIENTID having PATIENTID = '{0}'",patientId);
            DataSet ds = conn.Search(sql0);
            string serial = ds.Tables[0].Rows[0][0].ToString();
            int sequence = Convert.ToInt32(serial);
            for(int i = 0; i < list1.Count; i++)
            {
                string sql = string.Format(@"insert into BILL(PATIENTID,D
OCID,ITEM,AMOUNT,PRICE,DATETIME,SERIAL,FLAG) VALUES('{0}','{1}','
{2}','{3}','{4}',to_date('{5}','yyyy-MM-dd'),{6},'{7}')",patientId, user, list1[i], list2
[i], list3[i], date,sequence+i+1,0);
                conn.Execute(sql);
            }
            conn.Disconn();
        }

        private void button1_Click(object sender, EventArgs e)//删除药品栏的一
行
        {
            currentROW = dataGridView1.CurrentRow;
            dataGridView1.Rows.Remove(currentROW);
        }

        private void button2_Click(object sender, EventArgs e)//清空药品栏
        {
            dataGridView1.Rows.Clear();
        }
        private void button4_Click(object sender, EventArgs e)//清空诊疗结果
```

```
        {
            richTextBox2.Clear();
        }
        private void button5_Click(object sender, EventArgs e)//提交诊疗结果
        {
            disease = richTextBox2.Text;
            string sql = string.Format(@"INSERT INTO TREATINFO(DOCID,
PATIENTID,TIME,DISEASE) VALUES('{0}','{1}',to_date('{2}','yyyy-MM-dd'),'{3}
') ", user, patientId,date,disease);
            connect conn = new connect();
            conn.Link();
            int cnt = conn.Execute(sql);
            conn.Disconn();
            if(cnt > 0)
            {
                MessageBox.Show("提交成功！");
            }
        }
        public void getMedicineInfo()//获取开药信息
        {
            list1.Clear();
            list2.Clear();
            list3.Clear();
            DataSet ds;
            int rows = dataGridView1.Rows.Count;
            for(int i = 0; i < rows-1; ++i)
            {
                list1.Add(dataGridView1.Rows[i].Cells[0].Value.ToString());
                list2.Add(dataGridView1.Rows[i].Cells[1].Value.ToString());
            }
            connect conn = new connect();
            conn.Link();
            foreach (string s in list1)
            {
                string sql = string.Format(@"select PRICE FROM MEDICINE
```

```
WHERE NAME = '{0}'",s);
                    ds = conn.Search(sql);
                    if (ds.Tables[0].Rows[0][0].ToString().Length > 0)
                    {
                        list3.Add(ds.Tables[0].Rows[0][0]);
                    }
                }
                conn.Disconn();
            }
        private void button6_Click(object sender, EventArgs e)//接待下一个病人
        {
            //首先在 register 表中将前一个病人置为已诊疗
            connect conn = new connect();
            conn.Link();
            string sql = string.Format("update register set flag = '1' where patientid
= '{0}'",patientId);
            int cnt = conn.Execute(sql);
            conn.Disconn();
            if(cnt == 0)
            {
                MessageBox.Show("失败！");
                return;
            }
            next();
        }


//*********************************************************************
        //个人信息
        public void DisDocInfo()//显示医生个人信息
        {
            //查找
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"select * from doctor where id =
```

```
'{0}'",user);
                DataSet ds = conn.Search(sql);
                conn.Disconn();
                if(ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("查找失败！ ");
                    return;
                }
                textBoxID_D.Text = ds.Tables[0].Rows[0]["ID"].ToString();
                textBoxPW_D.Text = ds.Tables[0].Rows[0]["PASSWD"].ToString();
                textBoxNAME_D.Text = ds.Tables[0].Rows[0]["NAME"].ToString();
                textBoxGENDER_D.Text                                    =
ds.Tables[0].Rows[0]["GENDER"].ToString();
                textBoxSalary.Text = ds.Tables[0].Rows[0]["SALARY"].ToString();
                //生日
                string birthstr = ds.Tables[0].Rows[0]["BIRTH"].ToString();
                int pos = birthstr.IndexOf(' ');
                birthstr = birthstr.Substring(0, pos);
                birthstr = birthstr.Replace('/', '-');
                dateTimePickerBirth.Value = Convert.ToDateTime(birthstr);
                //入职时间
                string entry = ds.Tables[0].Rows[0]["ENTRY"].ToString();
                int pos1 = entry.IndexOf(' ');
                entry = entry.Substring(0, pos1);
                entry = entry.Replace('/', '-');
                dateTimePickerEntry.Value = Convert.ToDateTime(entry);
                //部门
                Info info = new Info();
                comboBoxDept.Text                                       =
info.GetDeptName(ds.Tables[0].Rows[0]["DEPTID"].ToString());
                //职称
                comboBoxTitle.Text = ds.Tables[0].Rows[0]["TITLE"].ToString();
                //照片
                string path = Application.StartupPath;//获取当前应用程序目录路径
                int pos2 = path.LastIndexOf("\\");
```

```
            pos2 = path.LastIndexOf("\\", pos2 - 1);
            if (pos2 != -1)
            {
                path = path.Substring(0, pos2 + 1);
            }
            targetPath = path + "pictures\\D" + user + ".jpg";
            if (File.Exists(targetPath))
            {
                pictureBox_D.Image = Image.FromFile(targetPath);
            }
            else
            {
                string defaultPath = path + "pictures\\default.png";
                pictureBox_D.Image = Image.FromFile(defaultPath);
            }
        }
        public void getInfo()
        {//获取医生注册信息
            passwd = textBoxPW_D.Text;
            name = textBoxNAME_D.Text;
            gender = textBoxGENDER_D.Text;
            birth = dateTimePickerBirth.Text;
            entry = dateTimePickerEntry.Text;
        }
        private void button9_Click(object sender, EventArgs e)//确认提交个人信
息
        {
            //将信息提交到数据库
            getInfo();
            if (passwd.Length == 0 || name.Length == 0 ||
                gender.Length == 0 || birth.Length == 0 || entry.Length == 0)
            {
                MessageBox.Show("请填写完整个人信息！");
                return;
            }
```

```
            //将年龄和入职转换为日期格式
            birth = birth.Replace('年', '-');
            birth = birth.Replace('月', '-');
            birth = birth.Substring(0, birth.Length - 1);

            entry = entry.Replace('年', '-');
            entry = entry.Replace('月', '-');
            entry = entry.Substring(0, entry.Length - 1);
            //提交
            upload();
        }
        public void upload()//上传医生信息
        {
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"update DOCTOR set PASSWD = '{0}',NAME = '{1}',GENDER = '{2}', BIRTH = to_date('{3}','yyyy-MM-dd') , ENTRY = to_date('{4}','yyyy-MM-dd') where id = '{5}'",passwd,name,gender,birth,entry,user);
            int cnt = conn.Execute(sql);
            conn.Disconn();
            if (cnt > 0)
            {
                MessageBox.Show("恭喜，修改成功！");
            }
            else if (cnt <= 0)
            {
                MessageBox.Show("很遗憾，修改失败，请重试！");
            }
        }
        private void button8_Click(object sender, EventArgs e)//点击修改个人信息
        {
            textBoxPW_D.ReadOnly = false;
            textBoxNAME_D.ReadOnly = false;
```

```
        }
        private void button7_Click(object sender, EventArgs e)//点击修改个人图
片
        {
            openFileDialog1.ShowDialog();
            picPath = openFileDialog1.FileName;
            Console.WriteLine("path:{0}", picPath);
            if (!File.Exists(picPath))
            {//如果文件不存在
                return;
            }
            //
            Image image = pictureBox_D.Image;
            pictureBox_D.Image = null;
            image.Dispose();
            pictureBox_D.Image = Image.FromFile(picPath);//显示新图像在框中
            bool isrewrite = true; // true=覆盖已存在的同名文件,false 则反之
            try
            {
                Console.WriteLine("targetPath:{0}",targetPath);
                Console.WriteLine("picPath:{0}", picPath);

                File.Copy(picPath, targetPath, isrewrite);
            }
            catch (Exception e1)
            {
                MessageBox.Show("error{0}", e1.ToString());
            }
        }

        //******************************************************
        //复诊
        private void button15_Click(object sender, EventArgs e)//点击查找复诊病
人信息
        {
```

```
            pid = textBox1.Text;
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"select name,gender,birth,note from
patient where id = '{0}'", pid);
            ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("发生错误！");
                return;
            }
            DisPInfo();
        }
        public void DisPInfo()
        {
            //显示基本信息
            textBox4.Text = ds.Tables[0].Rows[0]["name"].ToString();
            textBox3.Text = ds.Tables[0].Rows[0]["gender"].ToString();
            textBox2.Text = ds.Tables[0].Rows[0]["birth"].ToString();
            textBox5.Text = ds.Tables[0].Rows[0]["note"].ToString();
            //显示上一次诊疗信息
            string sql = string.Format(@"select DISEASE, TIME from TREA
TINFO where DOCID = '{0}' and time = (select max(time) from TREATINFO
 where PATIENTID = '{1}' and docid = '{2}')",user,pid,user);
            connect conn = new connect();
            conn.Link();
            ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("发生错误！");
                return;
            }
            richTextBox3.Text = ds.Tables[0].Rows[0][0].ToString();
```

```
            previousDATE = ds.Tables[0].Rows[0][1].ToString();
            int pos = previousDATE.IndexOf(' ');
            previousDATE = previousDATE.Substring(0,pos);
            previousDATE = previousDATE.Replace('/','-');
            Console.WriteLine("pre:{0}",previousDATE);
        }
        private void button11_Click(object sender, EventArgs e)//清空诊断栏
        {
            richTextBox3.Clear();
        }

        private void button14_Click(object sender, EventArgs e)//删除药品栏一行
        {
            currentROW = dataGridView2.CurrentRow;
            dataGridView2.Rows.Remove(currentROW);
        }
        private void button13_Click(object sender, EventArgs e)//清空药品栏
        {
            dataGridView2.Rows.Clear();
        }
        private void button10_Click(object sender, EventArgs e)//提交诊断信息
        {
         //首先判断是不是今日复诊
            disease = richTextBox4.Text;
            if (!previousDATE.Equals(date))
            {
                string sql = string.Format(@"insert into TREATINFO(DOCID,
TIME,PATIENTID,DISEASE) VALUES('{0}',TO_DATE('{1}','yyyy-MM-dd'),'{2}','
{3}')", user, date, pid,disease);
                connect conn = new connect();
                conn.Link();
                int cnt = conn.Execute(sql);
                conn.Disconn();
                if (cnt > 0)
                {
```

```
                    MessageBox.Show("提交成功！");
                }
            }
            else
            {
                string sql = string.Format(@"update TREATINFO set TIME = to_date('{0}','yyyy-MM-dd'),DISEASE = '{1}' where PATIENTID = '{2}' and DOCID = '{3}' and time = TO_DATE('{4}','yyyy-MM-dd')", date, disease,pid,user,previousDATE);
                connect conn = new connect();
                conn.Link();
                int cnt = conn.Execute(sql);
                conn.Disconn();
                if (cnt > 0)
                {
                    MessageBox.Show("提交成功！");
                }
            }
        }
        public void getMedicineInfo2()//获取开药信息
        {
            list1.Clear();
            list2.Clear();
            list3.Clear();
            DataSet ds;
            int rows = dataGridView2.Rows.Count;
            for (int i = 0; i < rows - 1; ++i)
            {
                list1.Add(dataGridView2.Rows[i].Cells[0].Value.ToString());
                list2.Add(dataGridView2.Rows[i].Cells[1].Value.ToString());
            }
            connect conn = new connect();
            conn.Link();
            foreach (string s in list1)
            {
```

```
                Console.WriteLine("{0}", s);
                string sql = string.Format(@"select PRICE FROM MEDICINE
WHERE NAME = '{0}'", s);
                ds = conn.Search(sql);
                if (ds.Tables[0].Rows[0][0].ToString().Length > 0)
                {
                    list3.Add(ds.Tables[0].Rows[0][0]);
                }
            }
            conn.Disconn();
        }
        private void button12_Click(object sender, EventArgs e)//提交用药信息
        {
            getMedicineInfo2();
            connect conn = new connect();
            conn.Link();
            string sql0 = string.Format(@"select max(SERIAL) from BILL
                        group by PATIENTID having PATIENTID = '{0}'",
pid);
            DataSet ds = conn.Search(sql0);
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("error");
                return;
            }
            string serial = ds.Tables[0].Rows[0][0].ToString();
            //Console.WriteLine("serial:{0}",serial);
            int sequence = Convert.ToInt32(serial);
            for (int i = 0; i < list1.Count; i++)
            {
                string sql = string.Format(@"insert into BILL(PATIENTID,D
OCID,ITEM,AMOUNT,PRICE,DATETIME,SERIAL,FLAG) VALUES('{0}','{1}','
{2}','{3}','{4}',to_date('{5}','yyyy-MM-dd'),{6},'{7}')", pid, user, list1[i], list2[i], li
st3[i], date, sequence + i + 1, 0);
                conn.Execute(sql);
```

```
                MessageBox.Show("提交成功！");
            }
            conn.Disconn();
        }

        //**************************************************检查
        private void button18_Click(object sender, EventArgs e)//删除一项
        {
            currentROW = dataGridView3.CurrentRow;
            dataGridView3.Rows.Remove(currentROW);
        }
        private void button17_Click(object sender, EventArgs e)//清空
        {
            dataGridView3.Rows.Clear();
        }

        private void button16_Click(object sender, EventArgs e)//提交检查信息
        {
            getCheckInfo();
            connect conn = new connect();
            conn.Link();
            string sql0 = string.Format(@"select max(SERIAL) from BILL
group by PATIENTID having PATIENTID = '{0}'", patientId);
            DataSet ds = conn.Search(sql0);
            string serial = ds.Tables[0].Rows[0][0].ToString();
            //Console.WriteLine("serial:{0}",serial);
            int sequence = Convert.ToInt32(serial);
            for (int i = 0; i < list1.Count; i++)
            {
                string sql = string.Format(@"insert into BILL(PATIENTID,D
OCID,ITEM,AMOUNT,PRICE,DATETIME,SERIAL,FLAG) VALUES('{0}','{1}','
{2}','1','{3}',to_date('{4}','yyyy-MM-dd'),{5},'{6}')", patientId, user, list1[i], list2[i],
 date, sequence + i + 1, 0);
                conn.Execute(sql);
            }
```

```
            MessageBox.Show("提交成功！");
            conn.Disconn();
        }

        public void getCheckInfo()//获取检查信息
        {
            list1.Clear();
            list2.Clear();
            DataSet ds;
            int rows = dataGridView3.Rows.Count;
            for (int i = 0; i < rows - 1; ++i)
            {
                list1.Add(dataGridView3.Rows[i].Cells[0].Value.ToString());
            }
            connect conn = new connect();
            conn.Link();
            foreach (string s in list1)
            {
                string sql = string.Format(@"select PRICE FROM CHECKITEM
WHERE NAME = '{0}'", s);
                ds = conn.Search(sql);
                if(ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("不存在该项目！");
                    return;
                }
                if (ds.Tables[0].Rows[0][0].ToString().Length > 0)
                {
                    list2.Add(ds.Tables[0].Rows[0][0]);
                    Console.WriteLine("{0}", ds.Tables[0].Rows[0][0]);
                }
            }
            conn.Disconn();
        }
```

```
        //初诊住院
        private void button19_Click(object sender, EventArgs e)
        {
            MessageBox.Show("请该患者到收费处办理住院手续！");
        }
        //**************************************************复诊
        private void button22_Click(object sender, EventArgs e)//删除一行
        {
            currentROW = dataGridView4.CurrentRow;
            dataGridView4.Rows.Remove(currentROW);
        }
        private void button21_Click(object sender, EventArgs e)//清空
        {
            dataGridView4.Rows.Clear();
        }
        private void button20_Click(object sender, EventArgs e)//提交检查
        {
            getCheckInfo2();
            connect conn = new connect();
            conn.Link();
            string sql0 = string.Format(@"select max(SERIAL) from BILL
group by PATIENTID having PATIENTID = '{0}'", pid);
            DataSet ds = conn.Search(sql0);
            string serial = ds.Tables[0].Rows[0][0].ToString();
            int sequence = Convert.ToInt32(serial);
            for (int i = 0; i < list1.Count; i++)
            {
                string sql = string.Format(@"insert into BILL(PATIENTID,D
OCID,ITEM,AMOUNT,PRICE,DATETIME,SERIAL,FLAG) VALUES('{0}','{1}','
{2}','1','{3}',to_date('{4}','yyyy-MM-dd'),{5},'{6}')", pid, user, list1[i], list2[i], dat
e, sequence + i + 1, 0);
                conn.Execute(sql);
            }
            MessageBox.Show("提交成功！");
            conn.Disconn();
```

```
        }
        public void getCheckInfo2()//获取检查信息
        {
            list1.Clear();
            list2.Clear();
            DataSet ds;
            int rows = dataGridView4.Rows.Count;
            for (int i = 0; i < rows - 1; ++i)
            {
                list1.Add(dataGridView4.Rows[i].Cells[0].Value.ToString());
            }
            connect conn = new connect();
            conn.Link();
            foreach (string s in list1)
            {
                string sql = string.Format(@"select PRICE FROM CHECKITEM
WHERE NAME = '{0}'", s);
                ds = conn.Search(sql);
                if (ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("不存在该项目！");
                    return;
                }
                if (ds.Tables[0].Rows[0][0].ToString().Length > 0)
                {
                    list2.Add(ds.Tables[0].Rows[0][0]);
                }
            }
            conn.Disconn();
        }

        private void button23_Click(object sender, EventArgs e)//复诊住院
        {
            MessageBox.Show("请该病人到收费处办理住院手续！");
        }
```

```
    }
}
```

CashierMian.cs

```
namespace HM
{
    public partial class cashierMain : Form
    {
        string user;
        string date;
        string patientID;
        string docID;
        float sum = 0;
        float sum_ = 0;
        DataSet ds;
        private string targetPath;
        private string picPath;
        private string pid2;
        private string ward2;

        public cashierMain(string user)
        {
            this.user = user;
            InitializeComponent();
            Show();
            date = DateTime.Now.ToString("yyyy-MM-dd");
        }
        private void tabControl1_SelectedIndexChanged(object sender, EventArgs
e)
        {//分页控件
            if(tabControl1.SelectedIndex == 3)
            {
                DisCashierInfo();
            }
        }
```

```
//**************************************************挂号
private void button1_Click_1(object sender, EventArgs e)//点击确认
{
    patientID = textBox1.Text;
    docID = textBox2.Text;
    if (patientID.Length == 0 || docID.Length == 0)
    {
        MessageBox.Show("请填写完整！");
        return;
    }
    string sql = string.Format(@"select * from appoint where docid = '{0}' and patientid = '{1}' and datetime = to_date('{2}','yyyy-MM-dd')", docID, patientID, date);
    connect conn = new connect();
    conn.Link();
    ds = conn.Search(sql);
    if (ds.Tables[0].Rows.Count == 0)
    {
        MessageBox.Show("抱歉，您还未预约！");
    }
    else
    {
        //查找该医生当日最大挂号数
        string sqlMax = string.Format(@"select max(SERIAL) from register where DOCID = '{0}' and DATETIME = TO_DATE('{1}','yyyy-MM-dd')", docID, date);
        ds = conn.Search(sqlMax);
        if (ds.Tables[0].Rows.Count == 0 || ds.Tables[0].Rows[0][0].ToString().Length == 0)
        {//还没有该医生今日的挂号信息
            string sqlInsert = string.Format(@"insert into register(patientid, docid,datetime,serial,flag) values('{0}','{1}',to_date('{2}','yyyy-MM-dd'),'{3}','{4}')", patientID, docID, date, "1", "0");
            int cnt = conn.Execute(sqlInsert);
            if (cnt == 0)
```

```
                {
                    MessageBox.Show("error");
                    return;
                }
                MessageBox.Show("您的序号是:1");
            }
            else if(ds.Tables[0].Rows[0][0].ToString().Length > 0)
            {
                int max = Convert.ToInt32(ds.Tables[0].Rows[0][0].ToStri
ng());
                ++max;
                string sqlcheck = string.Format(@"select * from register
 where patientid = '{0}' and docid = '{1}'and datetime = to_date('{2}','yyyy-M
M-dd')",patientID,docID,date);
                ds = conn.Search(sqlcheck);
                if(ds.Tables[0].Rows.Count > 0)
                {
                    MessageBox.Show("您今日已挂号过该医生，请勿重
复挂号！");
                    return;
                }
                string sqlInsert = string.Format(@"insert into register(pati
entid, docid,datetime,serial,flag) values('{0}','{1}',to_date('{2}','yyyy-MM-dd'),'{3}',
'{4}')", patientID, docID, date, max.ToString(), "0");
                int cnt = conn.Execute(sqlInsert);
                if (cnt == 0)
                {
                    MessageBox.Show("error");
                    return;
                }
                MessageBox.Show("您的序号是:" + max.ToString());
            }
            else
            {
                MessageBox.Show("error");
```

```
                return;
            }
        }
    }

    private void button2_Click(object sender, EventArgs e)//点击取消
    {
        textBox1.Clear();
        textBox2.Clear();
    }
    //*****************************************************取药
    private void button3_Click(object sender, EventArgs e)//点击查找
    {
        patientID = textBox3.Text;
        docID = textBox6.Text;
        if(patientID.Length == 0 && docID.Length == 0)
        {
            MessageBox.Show("信息不完整！");
            return;
        }

        if(patientID.Length > 0 && docID.Length == 0)
        {//查找全部未交费信息
            AllFee_();//获取全部信息
            AllFee();
        }
        else if (patientID.Length > 0 && docID.Length > 0)
        {//查找部分信息
            AllFee_();//获取全部信息
            PartFee();
        }
    }
    public void AllFee_()//获取所有费用
    {
        dataGridView1.Rows.Clear();
```

```
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"select item,amount,price from BILL
                    where PATIENTID = '{0}' and FLAG = '0'", patientID);
            ds = conn.Search(sql);
            conn.Disconn();
            if (ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("该病人没有费用要交！");
                return;
            }
            getSum_();
        }
        public void AllFee()//获取所有费用
        {
            dataGridView1.Rows.Clear();
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"select item,amount,price from BILL
                    where PATIENTID = '{0}' and FLAG = '0',patientID);
            ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("该病人没有费用要交！");
                return;
            }
            //dataGridView1.DataSource = ds.Tables[0];
            for(int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {
                dataGridView1.Rows.Add();
                dataGridView1.Rows[i].Cells[0].Value =
ds.Tables[0].Rows[i][0].ToString();
                dataGridView1.Rows[i].Cells[1].Value =
ds.Tables[0].Rows[i][1].ToString();
```

```
                dataGridView1.Rows[i].Cells[2].Value =
ds.Tables[0].Rows[i][2].ToString();
            }
            textBox5.Text = sum_.ToString();
        }
        public void PartFee()//获取部分费用
        {
            dataGridView1.Rows.Clear();
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"select item,amount,price from BILL
                where PATIENTID = '{0}' and docid = '{1}' and FLAG = '0'",
patientID,docID);
            ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("未查到相关信息");
                return;
            }
            //dataGridView1.DataSource = ds.Tables[0];
            for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {
                dataGridView1.Rows.Add();
                dataGridView1.Rows[i].Cells[0].Value =
ds.Tables[0].Rows[i][0].ToString();
                dataGridView1.Rows[i].Cells[1].Value =
ds.Tables[0].Rows[i][1].ToString();
                dataGridView1.Rows[i].Cells[2].Value =
ds.Tables[0].Rows[i][2].ToString();
            }
            getSum();
        }

        public void getSum()//获得要交费用的总和
```

```
{
    sum = 0;
    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
    {
        sum += Convert.ToSingle(ds.Tables[0].Rows[i][1].ToString()) +
        Convert.ToSingle(ds.Tables[0].Rows[i][2].ToString());
    }
    textBox5.Text = sum.ToString();
}
public void getSum_()//获得总费用
{
    sum_ = 0;
    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
    {
        sum_ += Convert.ToSingle(ds.Tables[0].Rows[i][1].ToString()) +
       Convert.ToSingle(ds.Tables[0].Rows[i][2].ToString());
    }
    textBox4.Text = sum_.ToString();
}
private void button4_Click(object sender, EventArgs e)//点击确认缴费
{
    if (patientID.Length == 0 && docID.Length == 0)
    {
        MessageBox.Show("信息不完整！");
        return;
    }
    if (patientID.Length > 0 && docID.Length == 0)
    {//交完所有未交费用
        connect conn = new connect();
        conn.Link();
        string sql = string.Format(@"update BILL set flag = '1'
                                    where patientid = '{0}'",patientID);
        int cnt = conn.Execute(sql);
        conn.Disconn();
        if(cnt > 0)
```

```
                {
                    MessageBox.Show("缴费成功");
                    textBox5.Text = "0";
                    textBox4.Text = "0";
                }
                else
                {
                    MessageBox.Show("缴费失败");
                    return;
                }
            }
            else if (patientID.Length > 0 && docID.Length > 0)
            {//交部分费用用
                connect conn = new connect();
                conn.Link();
                string sql = string.Format(@"update BILL set flag = '1'
                                where patientid = '{0}' and docid = '{1}'",
patientID,docID);
                int cnt = conn.Execute(sql);
                conn.Disconn();
                if (cnt > 0)
                {
                    MessageBox.Show("缴费成功");
                    textBox5.Text = "0";
                    textBox4.Text = (sum_ - sum).ToString();
                }
                else
                {
                    MessageBox.Show("缴费失败");
                    return;
                }
            }
            connect cc = new connect();
            cc.Link();
            //将药品数量减少
```

```
            for(int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {
                string Mname = ds.Tables[0].Rows[i][0].ToString();
                string sql = string.Format(@"select stoke from medicine where
name = '{0}'",Mname);
                DataSet dd = cc.Search(sql);
                if(dd.Tables[0].Rows.Count > 0)
                {
                    int stoke = Convert.ToInt32(dd.Tables[0].Rows[0][0].ToStr
ing()) - Convert.ToInt32(ds.Tables[0].Rows[i][1].ToString());
                    if(stoke < 0)
                    {
                        MessageBox.Show("药品:" + Mname+ "数量不足！");
                    }
                    else
                    {
                        string sqldel = string.Format(@"update medicine set
 stoke = '{0}' where name = '{1}'",stoke.ToString(),Mname);
                        int cnt = cc.Execute(sqldel);
                        if(cnt == 0)
                        {
                            MessageBox.Show("失败！");
                            return;
                        }
                    }

                }
            }
            cc.Disconn();
        }
        //************************************************************ 显
示收费员信息
        public void DisCashierInfo()
        {
            string sql = string.Format(@"select * from  CASHIER where id =
```

```
'{0}'",user);
                connect conn = new connect();
                conn.Link();
                DataSet ds = conn.Search(sql);
                conn.Disconn();
                if(ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("发生错误！ ");
                    return;
                }
                textBoxID.Text = ds.Tables[0].Rows[0]["ID"].ToString();
                textBoxPW.Text = ds.Tables[0].Rows[0]["PASSWD"].ToString();
                textBoxName.Text = ds.Tables[0].Rows[0]["NAME"].ToString();
                comboBox1.Text = ds.Tables[0].Rows[0]["GENDER"].ToString();
                //生日
                string birth = ds.Tables[0].Rows[0]["BIRTH"].ToString();
                int pos = birth.IndexOf(' ');
                birth = birth.Substring(0, pos);
                birth = birth.Replace('/', '-');
                dateTimePickerBirth.Value = Convert.ToDateTime(birth);
                //入职时间
                string entry = ds.Tables[0].Rows[0]["ENTRY"].ToString();
                int pos1 = entry.IndexOf(' ');
                entry = entry.Substring(0, pos1);
                entry = entry.Replace('/', '-');
                dateTimePickerEntry.Value = Convert.ToDateTime(entry);
                //照片
                string path = Application.StartupPath;//获取当前应用程序目录路径
                int pos2 = path.LastIndexOf("\\");
                pos2 = path.LastIndexOf("\\", pos2 - 1);
                if (pos2 != -1)
                {
                    path = path.Substring(0, pos2 + 1);
                }
                targetPath = path + "pictures\\C" + user + ".jpg";
```

```
            //Console.WriteLine("str:{0}", targetPath);
            if (File.Exists(targetPath))
            {
                pictureBox1.Image = Image.FromFile(targetPath);
            }
            else
            {
                string defaultPath = path + "pictures\\default.png";
                pictureBox1.Image = Image.FromFile(defaultPath);
            }

        }

        private void button7_Click(object sender, EventArgs e)//提交修改
        {
            string passwd = textBoxPW.Text;
            string name = textBoxName.Text;
            string gender = comboBox1.Text;
            if (passwd.Length == 0 || name.Length == 0|| gender.Length == 0)
            {
                MessageBox.Show("请将信息填写完整！");
                return;
            }
            string sql = string.Format(@"update cashier set passwd = '{0}',name =
'{1}',gender = '{2}' ",
                                                passwd,name,gender);
            connect conn = new connect();
            conn.Link();
            int cnt = conn.Execute(sql);
            conn.Disconn();
            if(cnt > 0)
            {
                MessageBox.Show("修改成功！");
            }
            else
```

```
        {
            MessageBox.Show("修改失败");
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {//修改
        textBoxPW.ReadOnly = false;
        textBoxName.ReadOnly = false;
        comboBox1.Enabled = true;
    }
    private void button5_Click(object sender, EventArgs e)//修改图片
    {
        openFileDialog1.ShowDialog();
        picPath = openFileDialog1.FileName;
        //Console.WriteLine("path:{0}", picPath);
        if (!File.Exists(picPath))
        {//如果文件不存在
            return;
        }
        //
        Image image = pictureBox1.Image;
        pictureBox1.Image = null;
        image.Dispose();
        pictureBox1.Image = Image.FromFile(picPath);//显示新图像在框中
        bool isrewrite = true; // true=覆盖已存在的同名文件,false 则反之
        try
        {
            //Console.WriteLine("targetPath:{0}", targetPath);
            //Console.WriteLine("picPath:{0}", picPath);

            File.Copy(picPath, targetPath, isrewrite);
        }
        catch (Exception e1)
        {
```

```
                MessageBox.Show("error{0}", e1.ToString());
            }
        }
        //**************************************************住院
        private void button8_Click(object sender, EventArgs e)//住院确认按钮
        {
            string ward;
            string pid = textBox7.Text;
            if(pid.Length == 0)
            {
                MessageBox.Show("请输入就诊号！ ");
                return;
            }
            //查询该病人今日是否已经办过住院手续
            string sql = string.Format(@"select id from ward where patientid =
'{0}' and
                                        startdate                                    =
to_date('{1}','yyyy-MM-dd')",pid,date);
            connect conn = new connect();
            conn.Link();
            DataSet ds = conn.Search(sql);

            if(ds.Tables[0].Rows.Count == 0)
            {//未办理
                ward = getWard();
                if(ward != null)
                {
                    string sqlinsert = string.Format(@"insert into ward(id,
patientid,startdate)

values('{0}','{1}',to_date('{2}','yyyy-MM-dd'))",ward,pid,date);
                    int cnt = conn.Execute(sqlinsert);
                    if(cnt == 0)
                    {
                        MessageBox.Show("发生错误 1");
```

```
                    return;
                }
                //将该床位置为被占用
                string sqlupdate = string.Format(@"update wardinfo set s
tate = 1 where id = '{0}'",ward);
                cnt = conn.Execute(sqlupdate);
                if(cnt == 0)
                {
                    MessageBox.Show("发生错误 2");
                    return;
                }
                //查找床位信息
                string sqls = string.Format(@"select * from wardinfo where
id = '{0}'",ward);
                ds = conn.Search(sqls);
                if (ds.Tables[0].Rows.Count == 0)
                {
                    MessageBox.Show("发生错误 3");
                    return;
                }
                string pos = ds.Tables[0].Rows[0]["POS"].ToString();
                string fee = ds.Tables[0].Rows[0]["FEE"].ToString();
                richTextBox1.Text = string.Format("该病人的床位号是：
{0},病房的位置在{1},每日床费{2}。",
                                                  ward,pos,fee);
            }
        }
        else
        {//已办理
            MessageBox.Show("您今日已办理住院手续！");
            return;
        }
        conn.Disconn();
    }
    public string getWard()//得到病房
```

```
        {
            string ward = null;
            string sql = string.Format(@"select id from  WARDinfo where
STATE = '0'");
            connect conn = new connect();
            conn.Link();
            DataSet ds = conn.Search(sql);
            conn.Disconn();
            if(ds.Tables[0].Rows.Count == 0)
            {
                MessageBox.Show("抱歉，当前没有空床！");
            }
            else
            {
                ward = ds.Tables[0].Rows[0][0].ToString();
            }
            return ward;
        }
        private void button9_Click(object sender, EventArgs e)//住院取消按钮
        {
            textBox7.Clear();
        }
        private void button11_Click(object sender, EventArgs e)//出院确认按钮
        {
            pid2 = textBox8.Text;
            if(pid2.Length == 0)
            {
                MessageBox.Show("请输入就诊号！");
                return;
            }
            //查询该病人是否曾经办过住院手续
            string sql = string.Format(@"select * from WARD
                    where PATIENTID = '{0}' and ENDDATE is null", pid2);
            connect conn = new connect();
            conn.Link();
```

```
        DataSet ds = conn.Search(sql);

        if(ds.Tables[0].Rows.Count == 0)
        {
            MessageBox.Show("该病人未办理过住院手续！");
            return;
        }
        ward2 = ds.Tables[0].Rows[0]["ID"].ToString();
        string startDate = ds.Tables[0].Rows[0]["startdate"].ToString();
        int pos = startDate.IndexOf(' ');
        startDate = startDate.Substring(0, pos);
        startDate = startDate.Replace('/', '-');
        //查找单价
        sql = string.Format(@"select fee from WARDinfo
                    where ID = '{0}'", ward2);
        ds = conn.Search(sql);
        if (ds.Tables[0].Rows.Count == 0)
        {
            MessageBox.Show("出错");
            return;
        }
        string fee = ds.Tables[0].Rows[0][0].ToString();
        //计算住院天数
        DateTime t1 = DateTime.Parse(date);
        DateTime t2 = DateTime.Parse(startDate);
        TimeSpan t3 = t1 - t2;
        double getday = t3.TotalDays;
        //计算住院费
        double sum = getday * Convert.ToDouble(fee);
        textBox12.Text = ward2;
        textBox10.Text = startDate;
        textBox11.Text = date;
        textBox9.Text = sum.ToString();
        conn.Disconn();
    }
```

```
        private void button10_Click(object sender, EventArgs e)//出院取消按钮
        {
            textBox8.Clear();

        private void button12_Click(object sender, EventArgs e)//缴费按钮
        {
            //修改 ward 表信息
            connect conn = new connect();
            conn.Link();
            string sql = string.Format(@"update WARD set ENDDATE =
TO_DATE('{0}','yyyy-MM-dd')
                                        where PATIENTID = '{1}' and ENDDATE is
null", date, pid2);
            int cnt = conn.Execute(sql);
            if (cnt == 0)
            {
                MessageBox.Show("错误");
                return;
            }
              sql = string.Format(@"update WARDinfo set state = 0
                                        where ID = '{0}'", ward2);
            cnt = conn.Execute(sql);
            if (cnt == 0)
            {
                MessageBox.Show("错误");
                return;
            }
            conn.Disconn();
            MessageBox.Show("缴费成功！");
        }
    }
}
```