

XI`AN TECHNOLOGICAL UNIVERSITY

**数据库原理与应用课程设计**

**二手房中介管理系统**

专 业： 物流管理

班 级： 17050718

学 号： 17050718116

姓 名： 李 诺

指导教师： 杨 敏

2019 年 12 月 27 日

摘要

随着科学技术的不断提高计算机科学的日渐成熟其强大的功能已为人们深刻认识它已进入人类社会的各个领域并发挥着越来越重要的作用。作为计算机应用的一部分,使用计算机对房屋中介的信息进行管理具有着手工管理所无法比拟的优点。这些优点能够极大地提高房屋中介和房地产公司营运的管理效率也是中介和公司的科学化、正规化管理,与世界接轨的重要条件。

管理信息系统从单位、中介到大型房产公司都是很必要的,它的内容对于买卖房屋的顾客和管理者来说都至关重要所以房屋中介管理信息系统应该能够为用户提供充足的信息和快捷的查询手段。一直以来人们使用传统人工的方式管理文件档案,或者使用人工和电脑统计相结合的方式,这些管理方式存在着许多缺点如效率低、保密性差,易出错,另外时间一长将产生大量的文件和数据这对于查找、更新和维护都带来了不少的困难。

因此,开发这样一个房屋中介管理信息系统成为很有必要的事情在下面的各章中我们将以开发一个房屋中介管理信息系统为例谈谈其开发过程和所涉及到的问题及解决方法。

**关键词：**房屋租赁；数据库；管理信息系统

目录

一 建立数据库的目的1

二 需求分析1

2.1 设计任务2

2.2 设计要求2

2.3 需求描述3

2.3.1 业务需求3

2.3.2 用户需求3

2.3.3 功能需求3

三 结构概念设计4

3.1 局部E-R图4

3.2 全局E-R图8

四 逻辑结构设计9

4.1 关系数据模式9

4.2 视图10

4.3 触发器10

五 数据库物理设计与实施12

5.1 物理结构设计12

5.1.1 确定数据库的存放位置12

5.1.2 存取方式的分析12

5.2 建表13

5.3 建立数据库17

5.4 加载测试数据18

六 数据操作要求及实现20

6.1 数据的查询操作20

6.2 数据维护操作22

6.3 索引24

七 个人总结25

**一 建立数据库的目的**

新中国成立后,特别是改革开放以来,居民住房问题一直是一个很棘手的问题,国家对于这个问题一再地修改关于房地产方面的议案,特别是在上海等比较发达的地区出现炒房现象之后,国家对于房地产的问题更加重视了,所以在居民和买卖房屋之间起桥梁作用的房产中介等机构就愈发显得重要了,而与此同时,由于计算机的普及以及互联网的飞速发展,使得很多原来由人力来完成的工作可以借由计算机来实现,因此,房屋中介管理信息系统渐渐发展起来了。

**二 需求分析**

**2.1 设计任务**

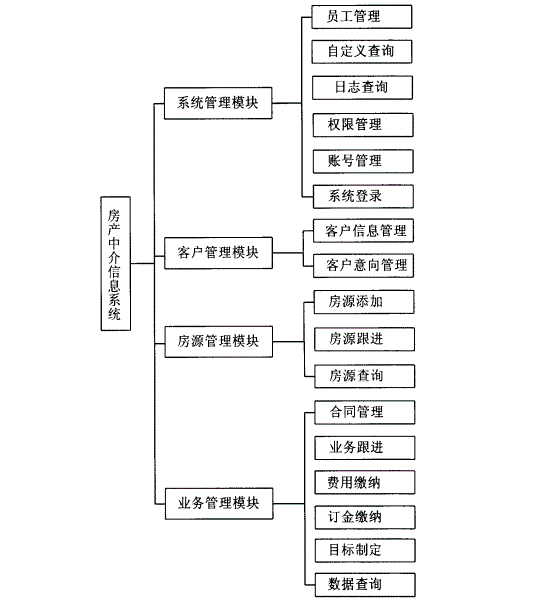
1、系统在功能上应在不同程度上实现以下几个方面的功能:

系统使用前必须先给用户授权,权限分:管理员权限和卖房者权限以及买房者权限;有关卖房者各种信息的输入,包括姓名,年龄,房源的描述,联系方式等:有关买房者各种信息的输入,包括姓名,年龄,对房源需求的描述,联系方式等:中介或公司工作人员管理的各种信息查询,修改和维护房屋中介管理信息系统的使用帮助。

2、系统功能的模块设计在系统功能分析基础上,考虑到程序的特点,编制系统功能模块图。

**2.2 设计要求**

对于系统参与者的功能用例需求进行归纳总结,得到系统功能需求分析结果。嘉信房产中介公司信息管理系统功能需求总体分析分析结果如图：

****

由图可知,房产中介信息管理系统的功能需求主要划分为系统管理、客户管理、房源管理和业务管理四个功能模块。

系统管理模块包括员工管理、自定义查询日志管理、权限管理、账号管理、系统登录功能。员工管理模块包括客户信息登记、客户意向管理功能。房源管理模块包括房源添加、房源跟进、房源查询功能。业务管理模块包括合同管理、业务跟进、费用缴纳、订金缴纳、目标制定和数据查询功能。功能需求分析所形成的功能方案,是系统设计和实现的依据。功能需求方案能够满足嘉信房产中介公司日常业务处理的要求。

**2.3 需求描述**

**2.3.1业务需求**

随着计算计硬件和软件的发展,现代社会早已经进入信息化高速运行时代。计算机在各中行业和领域的应用也越来越广泛,对于房屋中介和一些大型的房地产公司也不例外。现代社会的房产信息管理也以进入系统化、信息化阶段。大量的库存信息管理和快节奏的更新速度是传统的管理方式显得力不从心,而使用计算机管理就成了惟一的解决方案,然而计算机的应用是需强大的系统软件和行业软件来支持的,房产信息的管理亦不例外。

所以开发出一款功能强大的房屋中介管理信息系统是软件支持的需要,更何况此系统既要是功能强大,也所以,开发出一款功能强大,而又操作简洁、方便、快捷,适应用户要求的房屋中介管理信息系统是业务的需要。

**2.3.2用户需求**

1.软件具有简洁美观的界面。

2.操作简单、快捷,适合用户使用习惯。

3.要求系统安全、稳定。

4.要求能够注册多个用户,有不同的用户权限。

5.要求能够方便的使用增删改查等功能。

6.要求实现面向3个用户分别实现以下功能:卖房者:注册,登录,发布房源,查询购买者意向,确认成交。买房者:注册,登录,查询空闲房源,对中意房源报价,查询成交情况。管理员:查询买卖双方基本信息,查询房源情况,对卖房者登记的信息进行审核,并能根据房源情况进行分类统计。

**2.3.3功能需求**

1.创建系统用户、修该用户密码、设置用户权限。要求操作简单,易实现。

2.卖房者注册,录。

3.卖房者发布房源。

4.卖房者查询购买者意向,确认成交。

5.买房者注册,登录。

6.买房者查询空闲房源。

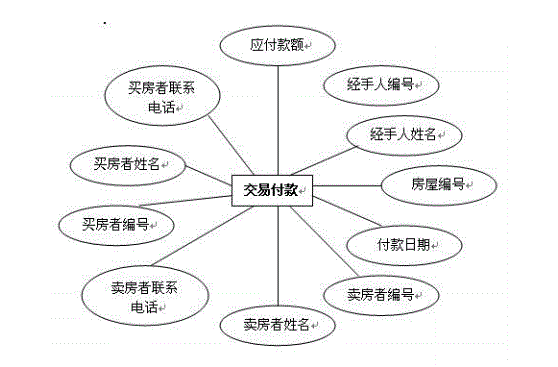
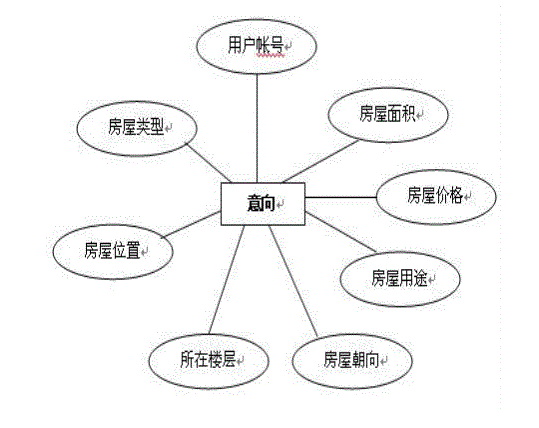
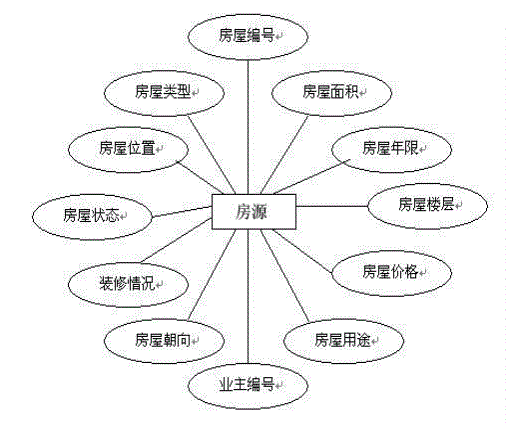
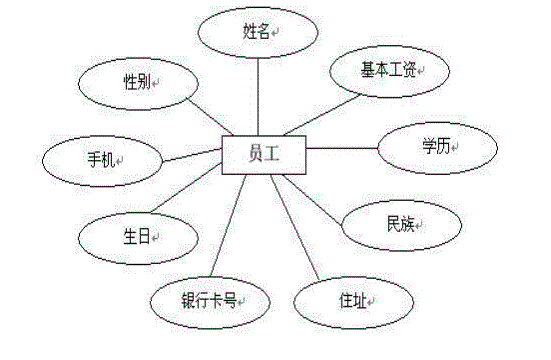
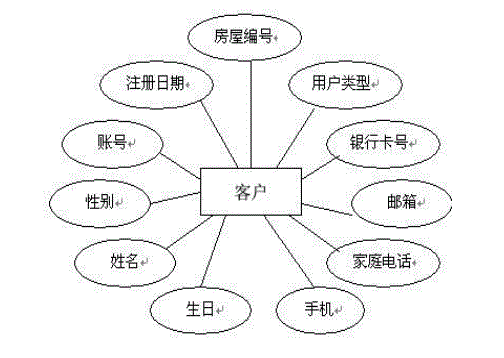
7.买房者对中意房源报价,查询成交情况。

8.管理员查询买卖双方基本信息。

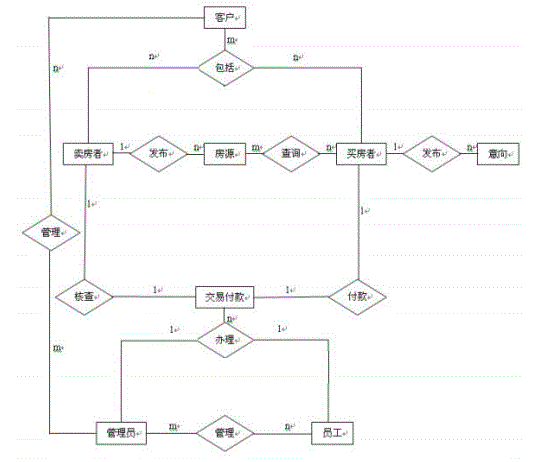
9.管理员查询房源情况。

10.管理员对卖房者登记的信息进行审核。

**三 结构概念设计**

**3.1 局部E-R图**

**3.2 全局E-R图**

****

**四 逻辑结构设计**

**4.1 关系数据模式**

E-R图转换成关系模式如下（其中，主码用下划直线表示，外码用下划波浪线表示）：

（1）房主关系

（房主编号，用户名，密码，性别，电话号码，家庭住址，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是房主编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

（2）房屋关系

（房屋编号，房主编号，房屋地点，面积，户型，照片，价格，出租状态，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是房屋编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

（3）租房者关系

（租房者编号，姓名，密码，性别，电话号码，职业，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是租房者编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

（4）看房关系

（看房信息编号，房主编号，租房者编号，看房日期，地点，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是看房信息编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

（5）租房关系

（租房信息编号，租房者编号，房屋编号，租房日期，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是租房信息编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

（6）交易关系

（交易信息编号，房主编号，房屋编号，金额，记录创建时间，记录更新时间）

该关系中，每一个属性值都是不可分的数据项，所有此关系模式为1NF；

该关系中，主码是交易信息编号，不存在非主属性对主码的部分函数依赖，所以此关系模式为2NF；

该关系中，不存在非主属性对主码的传递函数依赖，所以此关系模式为3NF；

该关系中，每个属性不存在部分函数依赖也不存在传递函数依赖于候选码，所以此关系为BCNF。

**4.2 视图**

样本数据库 Northwind 拥有一些被默认安装的视图。

视图 "Current Product List" 会从 "house" 表列出所有正在使用的产品（未停产的产品）。这个视图使用下面的 SQL 创建：

CREATE VIEW [Current Product List] AS

SELECT houseID,houseName

FROM houses

WHERE Discontinued=No

我们可以像这样查询上面这个视图：

SELECT \* FROM [Current Product List]

Northwind 样本数据库的另一个视图会选取 "house" 表中所有单位价格高于平均单位价格的房源：

CREATE VIEW [house Above Average Price] AS

SELECT ProductName,UnitPrice

FROM Products

WHERE UnitPrice>(SELECT AVG(UnitPrice) FROM Products)

我们可以像这样查询上面这个视图：

SELECT \* FROM [Products Above Average Price]

Northwind 样本数据库的另一个视图会计算在 1997 年每个种类的销售总数。请注意，这个视图会从另一个名为 "housing resource" 的视图那里选取数据：

CREATE VIEW [housing resource] AS

SELECT DISTINCT CategoryName,Sum(ProductSales) AS CategorySales

FROM [housing resoure]

GROUP BY CategoryName

我们可以像这样查询上面这个视图：

SELECT \* FROM [Category Sales For 1997]

我们也可以向查询添加条件。

SELECT \* FROM [Category Sales For 1997]

WHERE CategoryName='Beverages'

**4.3 触发器**

当确定完成租房交易之后，需要将对应房屋的出租状态设置为“已出租”，即对应数据“Y”：

****

**五 数据库物理设计与实施**

**5.1 物理结构设计**

**5.1.1确定数据库的存放位置**

为了提高系统的性能,应该根据应用情况将数据的易变部分,稳定部分、经常存取的部分和存取频率低的部分分开存放因此我可以将表和索引分别存放在不同的磁盘上,在查询时由于两个磁盘驱动器并行工作,可以提高物理读写的速度。

**5.2.2存取方式的分析**

从时间和空间上来说这种并行的设计方式是可以达到用户的需要的。存取方式的分析:对收费、报修的各个关系最经常的操作是查找,假设现有n个住宿房间的信息,如果采取顺序查找,平均查找n/2次;建立B+树索引,则平均查找次数为B+树的层数log2n+1

**5.4 建表**

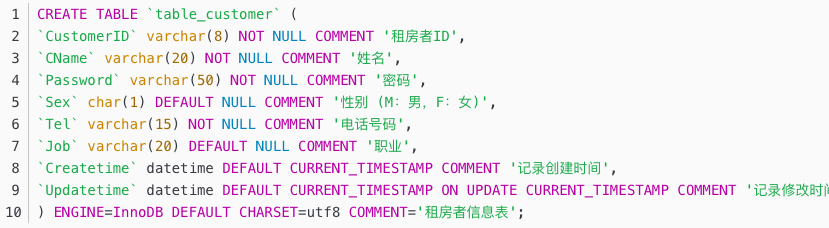
（1）房主  
创建数据表Table\_Provider,执行以下语句：

****

（2）房屋  
创建数据表Table\_House，执行以下语句：

****

（3）租房者  
创建数据表Table\_Customer，执行以下语句：

****

（4）看房  
创建数据表Table\_Showing，执行以下语句：

****

（5）租房  
创建数据表Table\_Renting，执行以下语句：

****

（6）交易  
创建数据库Table\_Deal，执行以下语句：

****

**5.3 索引的设计**

（1）房主  
在表Table\_Provider中，为列ProviderID创建索引：

****

2）房屋  
在表Table\_House中，为列HouseID创建索引：

****

（3）租房者  
在表Table\_Customer中，为列CustomerID创建索引：

****

（4）看房  
在表Table\_Showing中，为列ShowID创建索引：

****

（4）看房  
在表Table\_Showing中，为列ShowID创建索引：

****

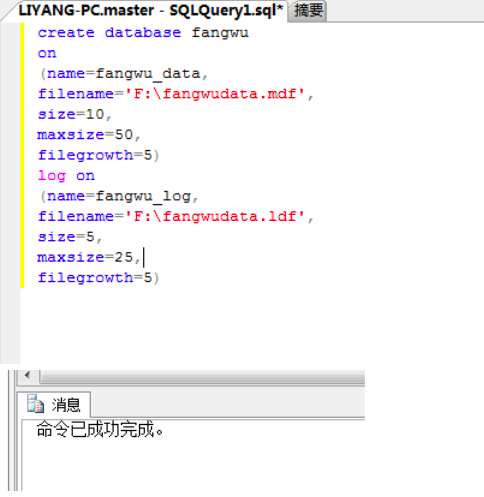
（5）租房  
在表Table\_Rentign中，为列RentID创建索引：

****

（6）交易  
在表Table\_Deal中，为列DealID创建索引：

****

**5.4 建立数据库**

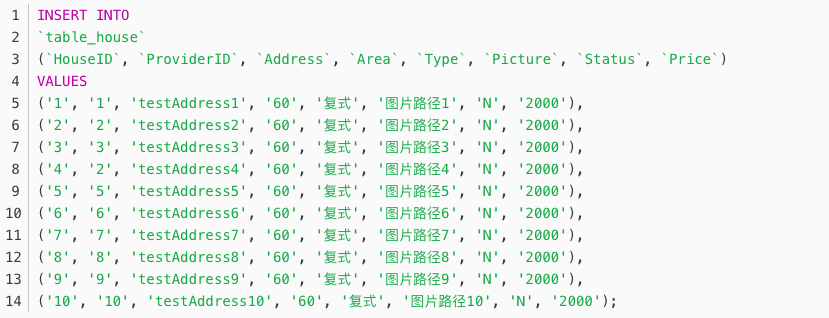
****

**5.4 加载测试数据**

（1）房主  
假设现在有10位房主准备把房子交给中介公司进行出租，这时候应该把10位房主的信息添加到数据表Table\_Provider里面：

****

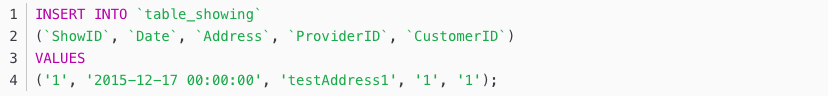
2）房屋  
假设10位房主拥有的房屋总共有10套，则应该把房屋信息添加到数据表Table\_House里面：

****

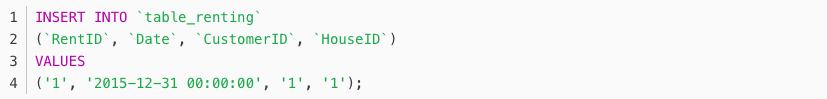
（3）租房者  
假设已经有10位顾客准备租房，现向数据表Table\_Customer中插入这10位顾客的信息，执行以下语句：

****

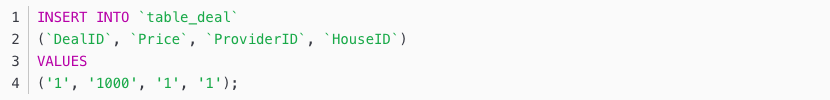
（4）看房  
假如租房顾客testCustomer1想要联系房主testprovider1一起看房子，则应该往数据表Table\_Showing添加数据：

****

（5）租房  
假如租房顾客testCustomer1决定要租用编号为1的房屋，则应该将信息插入数据表Table\_Renting：

****

（6）交易  
中介公司根据编号为1的房屋已经被租用的信息，需要与房主完成租金交易，此时应该将交易信息添加到数据表Table\_Deal：

****

**六 数据操作要求及实现**

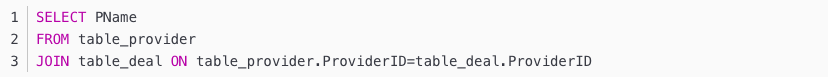
**6.1 数据查询操作**

（1）子查询  
假如需要查询编号为1的租房订单中的顾客的姓名和职业，则执行以下语句：

****

查询结果为：testCustomer1，学生

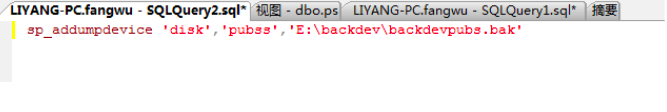
（2）连接查询  
假如想要查询已经将房屋出租的所有房主的姓名，可以对房主信息表和交易信息表进行内连接查询，执行以下语句：

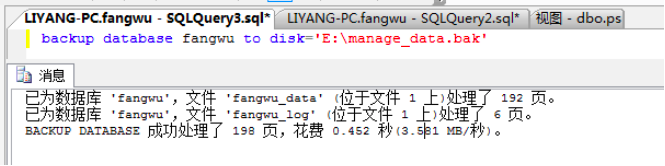
****

查询结果为：superProvider

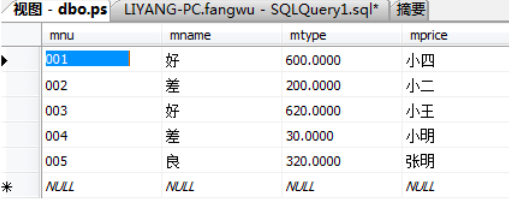
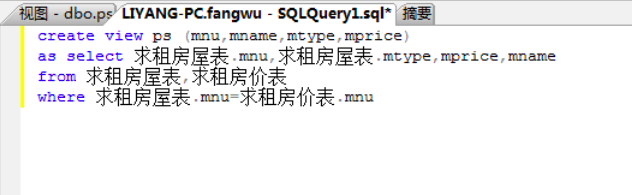
**6.2 数据维护操作**

按照设计阶段提供的安全规范和故障恢复规范,要经常检测系统的安全是否受到侵犯,要根据用户的实际需要授予用户不同的操作权限,还在防止数据库运行中受到环境的变化,对安全性的要求也是有变化的要根据实际的情况做出相应的调整,另外,为了确定系统在发生故障时,能够及时的进行恢复,管理员要针对不同的应用要求制定不同的转储计划,定期对数据库和日志文件进行备分,以使数据库在发生故障后恢复到某种一致性的,为了解决上述可能出现的问题,我感觉实行备分与还原还是比较可行的,备分与、还原的代码如下:

****

****

同样我们也可以进行还原:USEmasterRESTOREDATABASEfangwuFROMDISK=E\Manage\_Data.BAK'WITHREPLACE同样我们还可能利用数据库的三级模式,仅普通用户使用外模式,这样就可以使视图来达到这样的要求的啊,如下:

****

这样用户看到的就不是存储在物理上的真实的表,而是一张虚表。从而保证了数据的安全性。

**6.3 索引**

（1）房主  
在表Table\_Provider中，为列ProviderID创建索引：

****

（2）房屋  
在表Table\_House中，为列HouseID创建索引：

****

（3）租房者  
在表Table\_Customer中，为列CustomerID创建索引：

****

（4）看房  
在表Table\_Showing中，为列ShowID创建索引：

****

（5）租房  
在表Table\_Rentign中，为列RentID创建索引：

****

（6）交易  
在表Table\_Deal中，为列DealID创建索引：

****

**七 个人总结**

这次的课程设计我一开始感觉很难,因为对这个房屋的问题是一点了解也没有的,之后就找了许多的资料,看了好久,有了一点感觉,同时也参考了许多人的建议,有了很好的想法之后,才发现最难的不是敲代码,而是一种思想,你必须掌握一种数据库创建的思想,和一定的模式,不能天马行空,那样就会出现各种问题,这次是有很深的感触的,一开始,我就是天马行空做到那是那,最后发现全乱了,真的不知道怎么办了,后来没有办法,按照顺序一步步的来,最后我开始了漫长的设计生活,这其中的各种不适,还有就是对word的操作不是很熟练没有办法,只好边学边用,最大收获就是自己对于这次的互相帮助,这使有感觉自己有了很大的长进,也使我认识到自己的严重不足,还有就是自己的眼高手低,没有办法,自己想的很快,可是却是很难实现的,有时候在网上找了半天也是一无所获,只有自己苦苦的找,要不就是放弃了原来的想法,另改一种方法来达到同样的目地,感觉到了原来达到同样的目的是有很多不同的方法的啊,就是这样的,自己也不能一条路走到黑啊,我感觉自己的想法是那样的不够严谨。