东 莞 理 工 学 院

数据库设计综合实训报告

课程名称：数据库系统原理

设计题目：图书管理系统

院系名称：网络空间安全学院

班　　级：2017 网络工程 3 班

学生姓名：冯海涛

学　　号：201741111136

一． 题目

1、背景材料

用户有以下需求：建立读者档案；建立图书档案，建立书目索引；完成日常图书检索、借还工作，对读者档案、图书档案、借还系统的访问，必需进行身份验证。具体要求如下：

⑴ 读者档案

数据包括：借书证号、姓名、性别、出生日期、身份证号、单位、通讯地址、邮政编码、联系电话、办证日期、借阅范围（书库）、允许最多借书册数、借书期限、职业等。

操作要求：能办证、修改、注销；访问时要进行身份验证，办证、修改、注销应记录操作员编号、操作日期、理由、审批记录等。

输出数据：打印借书证卡片、借书证清单。

⑵ 图书档案

数据包括：书号、书名、作者、出版单位、出版日期、版次、单价、内容提要、分类号、索书号、藏书册数、每册图书馆藏注册号、所在书库、入库日期等。

操作要求：输入、修改、注销等操作必需进行身份验证，记录操作员号、操作日期、理由等。

输出数据：按入库日期时间段打印图书清单。

⑶ 检索系统

能根据书号、书名、作者、出版单位、内容提要关键字、分类号、索书号、每册图书馆藏注册号等进行简单查询和组合查询，对内容提要进行模糊查询。查询输出内容必需是只读的，没有访问身份验证要求。根据需要打印借书索书条。

⑷ 借书系统

输入索书条后，能根据借书证号判断该读者可以借书的书库，借书是否超出最大允许借书册数，书库中是否还有该书可借。满足条件的进行借书登记，不满足条件的给出提示信息，提示读者为什么不能借书。另外，还能查找以前所借图书情况。进入该模块具有身份验证要求。

⑸ 还书系统

对过期未还图书，能打印出催还图书通知单。对归还的图书能从借书登记表中取消。进入该模块具有身份验证要求。

2、设计要求：

⑴ 进行需求分析，编写数据字典。

⑵ 设计 E-R 图。

⑶ 进行数据库的逻辑设计。

⑷ 完成物理数据库的设计，包括数据库、表、索引、视图、完整性约束的物理设计。

⑸创建存储过程，创建触发器等。

二. 系统需求分析

1. 需求概述和系统边界

随着用户群体数量，以及图书馆书籍的库存的增加，图书管理系统为了实现系统管理员对用户和图书的管理，用户可以实现浏览、借阅、归还图书，同时可以查看、修改自己的相关信息等基本功能。

这个系统支持 3 类用户：游客、用户、系统管理员。游客可以随意浏览图书但只有注册为用户才能借阅图书。用户可以实现的操作：通过索引（书号、书名、作者、出版社…）查询图书信息，还可以借阅图书，归还图书，修改用户信息，打印借书证、打印借书清单。系统管理员可以实现对图书入库操作，图书信息相关（藏书册数、每册图书馆藏注册号、所在书库等等）修改操作，图书的删除操作；对用户某些属性（借阅范围（书库）、允许最多借书册数、借书期限）修改。

2. 数据需求分析

（1） 读者基本信息（借书证号，姓名、密码、性别、出生日期、身份证号、单位、通讯地址、邮政编码、联系电话、职业、办证日期、借阅范围（书库）、允许最多借书册数、借书期限），借书证号、身份证号是主码 ，用

户注册成功后返回一个借书证编号，与读者基本信息相关联，办证日期可以获取当前系统时间（年月日），读者分等级，不同等级的读者借书数量和范围限制不同

（2） 图书基本信息（书号、书名、作者、出版单位、出版日期、版次、单价、内容提要、分类号、索书号、藏书册数、每册图书馆藏注册号、所在书库、入库日期）书号是唯一标识，库存>=0

（3） 借书信息（读者借书证号、书号、借阅日期、数量），读者借书证号唯一标识

（4） 还书信息（读者借书证号、书号、归还日期、数量），读者借书证号唯一标识

（5） 系统管理员信息（管理员编号、姓名、密码、联系电话），管理员编号唯一标识

（6） 图书操作信息（管理员编号、操作日期、理由），管理员编号唯一标识

（7） 审核信息（管理员编号、操作日期、理由、审批理由）管理员编号唯一标识

（8） 读者等级信息（读者等级、借阅范围（书库）、允许最多借书册数、借书期限）读者等级唯一标识

（9） 分类信息（分类号、类别名）分类号唯一标识

（10） 出版社信息（出版社编号、出版社名称）出版社编号唯一标识

3. 功能需求分析

（1）用户（读者）管理。

用户注册、修改、查询自己的信息以及打印自己的借书清单

系统管理员修改及删除（注销）用户

管理员可以打印借书证清单

（2）图书管理。主要提供图书基本信息的录入、修改、查询，基本信息也包括了库存，所存仓库号等等

图书信息的录入、修改、查询

管理员可以对图书进行入库和删除

（3）借书管理，主要提供读者借阅图书

当该图书超出读者的借阅权限范围不允许办理该操作

当该图书库存不足的时候不能提供借阅

借阅图书后根据用户的借阅期限更新借阅表的应归还时间等，还应更新库存

管理员可以打印借阅清单

（4）还书管理，提供给用户进行还书操作

当用户还书后，更新还书信息归还时间和图书信息表的库存

4. 业务规则及完整性约束分析

（1） 游客、用户、管理员都可以搜索、浏览图书，但只用注册为用户后游客才能进行借书还书操作。

（2） 用户进行还书、借书、修改个人信息等操作时要进行登陆验证，同时管理员对用户有和图书进行相关的查询、修改、注销等操作的时候也要进行登陆验证

（3） 当某图书库存不足，不再不提供借阅

（4） 图书借阅和归还后都应该自动修改相应的库存数量

（5） 当读者所借的书没还超出了期限会提示催还信息单

（6） 用户只能借阅在规定的借阅范围的图书，超出范围的可以浏览、查询，但不允许借阅

（7） 用户修改自己的信息时，只能修改自己的基本信息，不能修改读者等级的属性，该读者等级只有管理可以修改

（8） 一种图书只能由一个出版社出版，而一个出版社可以有多种图书

（9） 借阅书的数量不能超过库存，也不能超过最多允许借阅数量，还书的数量不能超过借阅的数量

5. 数据字典

（1）读者实体集。其属性有：借书证号(LCNumber)，姓名(userName)、密码(userPassword)、性别(sex)、出生日期(birthday)、身份证号(Id_Number)、单位(workplace)、通讯地址(address)、邮政编码(postcode)、联系电话(phone)、

职业(work)、办证日期(date)、等级(level)、借阅范围（书库）(borrowRange)、允许最多借书册数(count)、借书期限(dateDeadline)

属性名	含义	类别	域及约束
readerNumber	借书证号	主码	char(10),不允许为空
userPassword	用户密码		Char(10)，不允许为空
userName	用户姓名		varchar(20)，不允许为空
sex	性别		char(2),取值范围：｛‘男’,’女’｝
birthday	出生日期		datetime
Id_Number	身份证号		Varchar(20),不允许为空
workplace	工作单位		varchar(30)
address	通讯地址		varchar(30)
postcode	邮政编码		varchar(10),数字组成
phone	联系电话		char(13),数字组成
work	职业		varchar(30)
level	等级		varchar(20),{“学生”、“老师”、”普通读者”}
date	办证日期		datetime，不允许为空
borrowRange	借阅范围（书库）		varchar(20)
count	允许最多借书册数		numeric,Default 10
dateDeadline	借书期限(天)		numeric,Default 180

读者信息数据字典

(2)图书实体集。其属性：书号(bookNumber)、书名(bookName)、作者(author)、出版单位(publish)、出版日期(publishDate)、版次(version)、单价(price)、内容提要(contents)、分类号(classNum)、索书号(bookNo)、藏书册数(bookCount)、每册图书馆藏注册号(regNum)、所在书库(stackRoom)、入库日期(intoDate)

属性名	含义	类别	域及约束
bookNumber	书号	主码	Char(10)，不允许为空
bookName	书名		varchar(20),not null
author	作者		varchar(20)
publish	出版社		varchar(30)

publishDate	出版日期		datetime
version	版次		numeric
price	单价		numeric
contents	内容提要		varchar(100)
classNum	分类号		char(10),NOT NULL
bookNo	索书号		char(10),NOT NULL
bookCount	藏书册数		numeric,NOT NULL
regNum	馆藏注册号		char(10),NOT NULL
stackRoom	所在书库		nuneric
intoDate	入库日期		datetime

图书信息实体集数据字典

（3）管理员实体集。其属性：管理员编号（adminNo）、管理员密码(adminPassword)、管理员姓名(adminName)、联系电话(phone)

属性名	含义	类别	域及约束
adminNo	管理员编号	主码	char(10),不允许为空
adminPassword	管理员密码		Char(10), 不允许为空
adminName	管理员姓名		varchar(20)，不允许为空
phone	联系电话		char(13),数字组成

管理员实体集数据字典

(4)借书清单实体集。其属性：读者借书证号（readerNumber）、书号（bookNumber）、借阅日期（borrowDate）、数量（count）

属性名	含义	类别	域及约束
bookNumber	书号	主码	Char(10), 不允许为空
borrowDate	借阅时间		datetime，不允许为空
count	借阅数量		numeric

借书清单实体集数据字典

(5)还书清单实体集. 其属性：读者借书证号（readerNumber）、书号（bookNumber）、借阅日期（returnDate）、数量（count）

属性名	含义	类别	域及约束
bookNumber	书号	主码	Char(10), 不允许为空

returnDate	归还时间		datetime，不允许为空
count	借阅数量		numeric

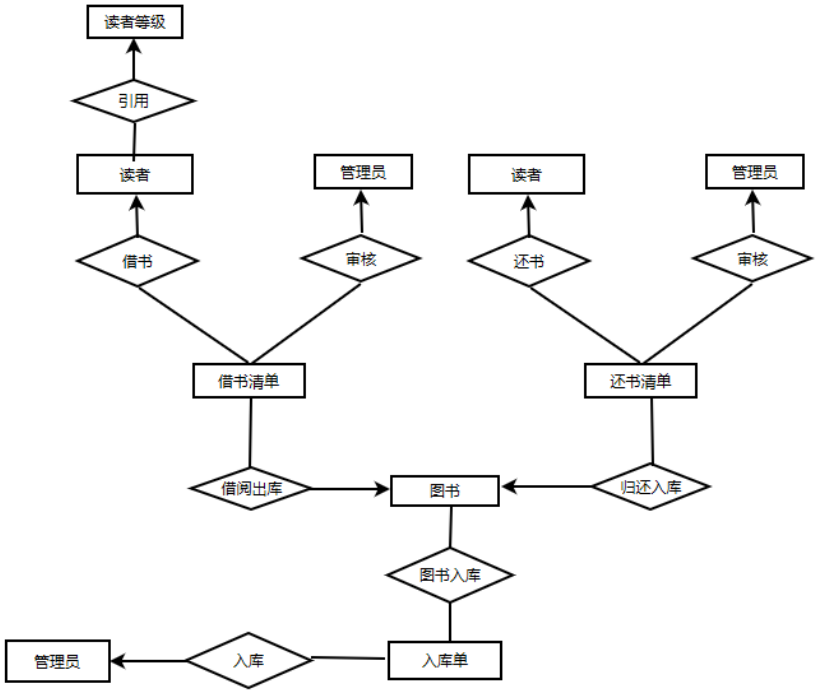
还书清单实体集数据字典

三. 系统概念设计

1. 确定联系集及属性

（1）引用联系集：它是用户等级实体集和用户（读者）实体集之间的多对一 联系集，没有联系属性
（2）借书联系集：它是用户实体集和借书清单实体集之间的一对多联系集，没有联系属性
（3）还书联系集：它是用户实体集和还书清单实体集之间的一对多联系集，没有联系属性
（4）审核联系集：它是管理员和借书清单、还书清单实体集之间一对多联系集，没有联系属性
（5）借阅出库联系集：它是借书清单实体集和和图书实体集之间的多对一联系集，没有联系属性
（6）归还入库联系集：它是还书清单实体集和图书实体集之间多对一联系集，没有联系属性
（7）图书入库联系集：它是图书和入库单之间多对多联系集，联系属性有入库数量
（8）入库联系集：它是管理员和入库单之间一对多联系集，没有联系属性

2. E-R 图



图书管理系统总 E-R 图

四. 数据库逻辑设计

（1） 管理员 Administrator 表：由管理员实体集转化而来

属性名称	数据类型	属性描述
adminNo	char(10)	管理员编号
adminPassword	char(10)	管理员密码
adminName	varchar(20)	管理员姓名
phone	varchar(15)	联系电话

（2） 读者等级表 ReaderLevel 表：有读者等级实体集转换而来

属性名称	数据类型	属性描述
readerLevel	varchar(10)	等级
borrowRange	varchar(20)	借阅范围（书库）
maxCount	numeric	允许最多借书册数

dateDeadline	numeric	借书期限(天)

（3） 读者信息表 Reader 表：由读者实体集和读者等级表转换而来

属性名称	数据类型	属性描述
readerNumber	char(10)	借书证号
create_date	datetime	办证日期
userPassword	Char(10)	用户密码
userName	varchar(20)	用户姓名
sex	char(2)	性别
birthday	datetime	出生日期
Id_Number	Varchar(20)	身份证号
workplace	varchar(30)	工作单位
address	varchar(30)	通讯地址
postcode	varchar(10)	邮政编码
phone	char(13)	联系电话
work	varchar(30)	职业
Readerlevel	varchar(20)	等级

（4） 图书信息表 Book 表：由图书实体集转换而来

属性名	数据类型	属性描述
bookNumber	Char(10)	书号
bookName	varchar(20)	书名
author	varchar(20)	作者
publish	varchar(30)	出版社名称
publishDate	datetime	出版日期
version	numeric	版次
price	numeric	单价
contents	varchar(300)	内容提要
classNum	char(10)	分类号
bookNo	char(10)	索书号
bookCount	numeric	藏书册数
regNum	char(10)	馆藏注册号
stackRoom	varchar(20)	所在书库
intoDate	datetime	入库日期

（5） 图书分类表 Store:由图书实体集分化而来

属性名	数据类型	属性描述

classNum	char(10)	分类号
className	varchar(20)	分类名称

（6） 借书清单表 Borrow 表：由图书实体集、读者实体集和管理员实体集转换而来

属性名	数据类型	属性描述
readerNumber	varchar(20)	借书证号
adminNo	varchar(20)	管理员编号
bookNumber	Char(10)	书号
borrowDate	datetime	借阅时间
borrowCount	numeric	借阅数量

（7） 还书清单表 ReturnBook 表：由图书实体集、读者实体集和管理员实体集转换而来

属性名	数据类型	属性描述
readerNumber	varchar(20)	借书证号
adminNo	varchar(20)	管理员编号
bookNumber	Char(10)	书号
returnDate	datetime	归还时间
rerurnCount	numeric	归还数量

（8） 入库单表 StoreSheet 表：由入库单实体集和管理员实体集转换而来

属性名	数据类型	属性描述
adminNo	varchar(20)	管理员编号
storeNo	Char(10)	入库单号
storeDate	datetime	入库时间

（9） 图书入库表 BookIn 表:由入库单实体集和图书实体集转换而来

属性名	数据类型	属性描述
bookNumber	varchar(20)	书号
storeNo	Char(10)	入库单号
storeCount	numeric	入库数量

（10） 管理员_读者表 Admin_Reader 表：根据功能需求转换而来

属性名	数据类型	属性描述

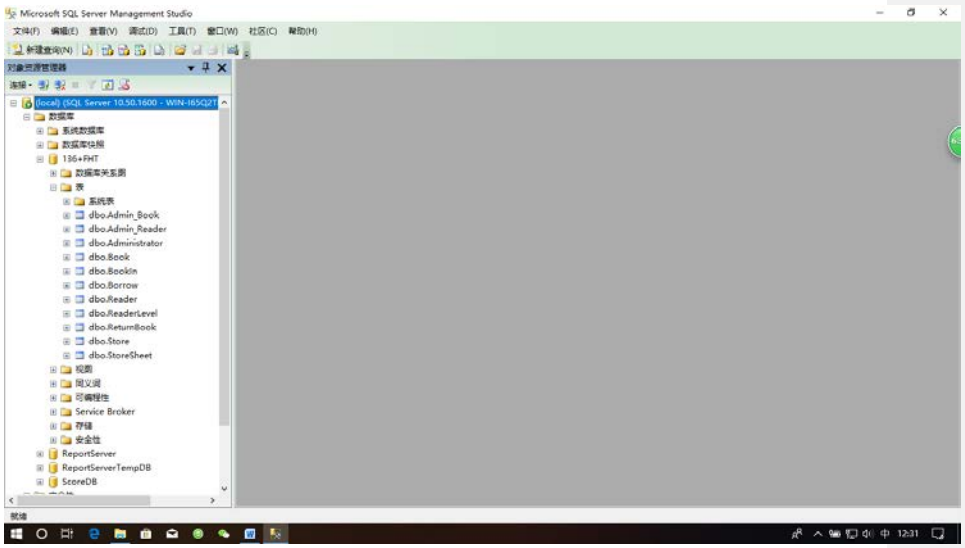
adminNo	char(10)	操作员编号
readerNo	char(20)	读者证号
operationDate	datetime	操作日期
reason	varchar(90)	理由
records	varchar(10)	审批记录

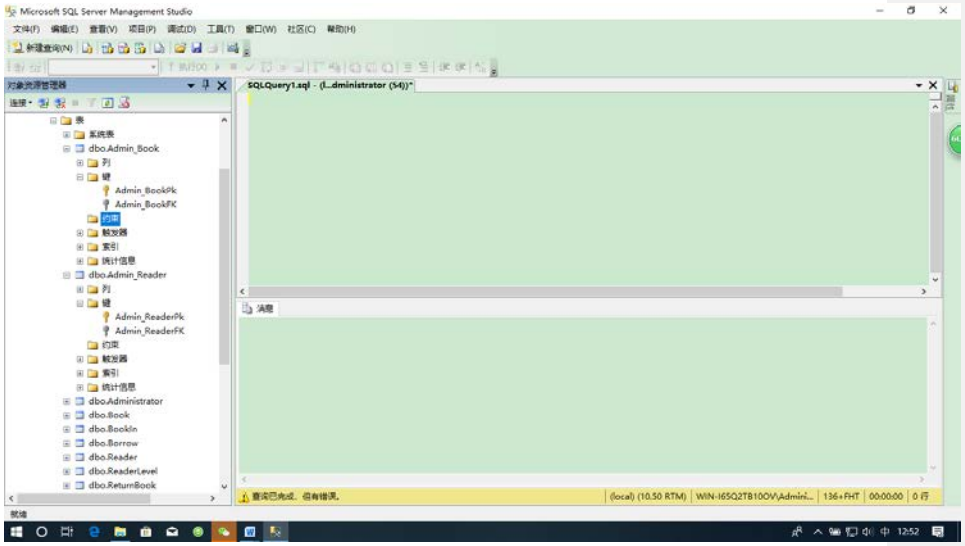
（11） 管理员_图书表 Admin_Book 表：由功能需求中分化而来，用来了管理员都图书操作

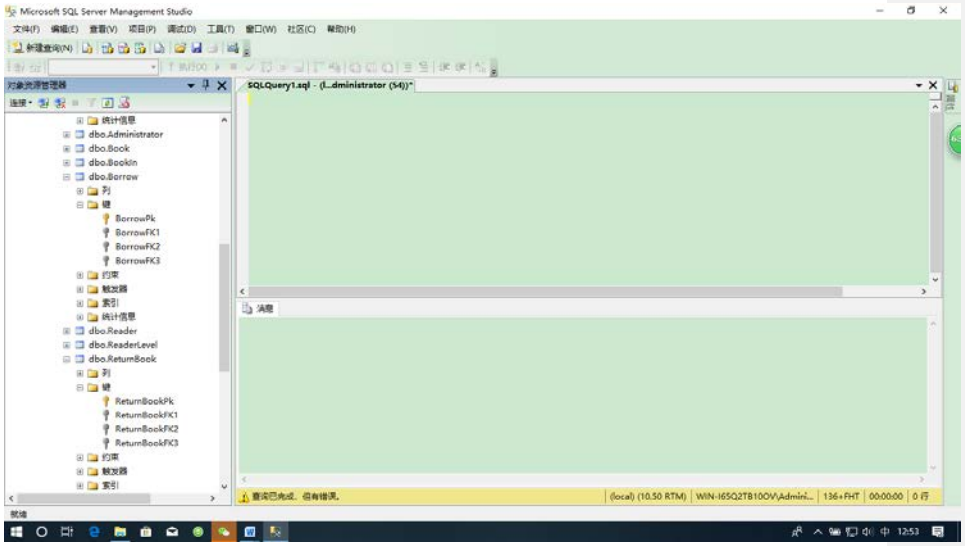
属性名	数据类型	属性描述
adminNo	char(10)	操作员编号
bookNo	char(20)	书号
operationDate	datetime	操作日期
reason	varchar(90)	理由

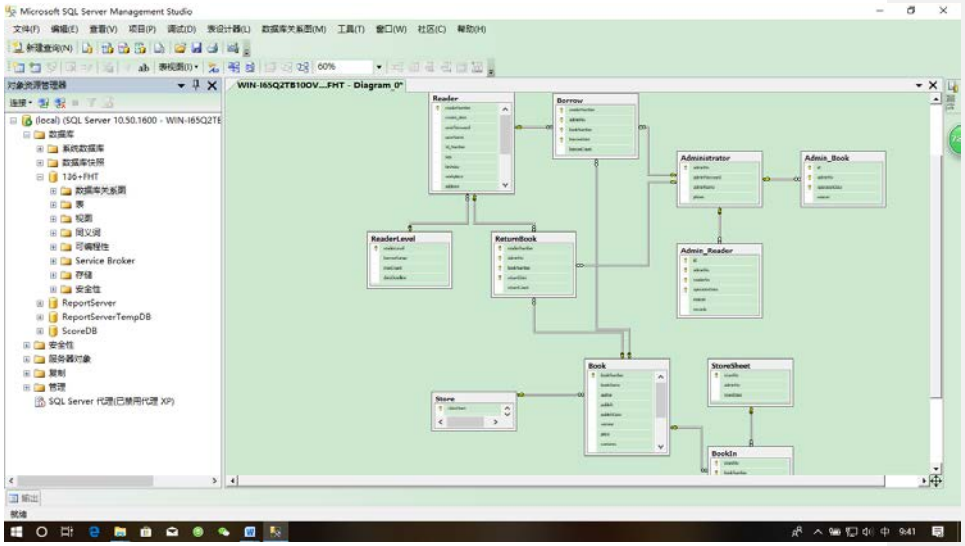
五． 数据库物理设计与运行代码及结果

1. 建表




Microsoft SQL Server Management Studio
文件(F) 编辑(E) 查看(V) 项目(P) 调试(D) 工具(T) 窗口(W) 社区(C) 帮助(H)
新建查询(N)
对象资源管理器
连接
SQLQuery1.sql - (l...dministrator (54))*
表
系统表
dbo.Admin_Book
列
键
Admin_BookPk
Admin_BookFK
约束
触发器
索引
统计信息
dbo.Admin_Reader
列
键
Admin_ReaderPk
Admin_ReaderFK
约束
触发器
索引
统计信息
dbo.Administrator
dbo.Book
dbo.BookIn
dbo.Borrow
dbo.Reader
dbo.ReaderLevel
dbo.ReturnBook
消息
查询已完成，但有错误。
(local) (10.50 RTM)
WIN-I65Q2TB10OV\Admini...
136+FHT
00:00:00
0 行
就绪
12:52


Microsoft SQL Server Management Studio
文件(F) 编辑(E) 查看(V) 项目(P) 调试(D) 工具(T) 窗口(W) 社区(C) 帮助(H)
新建查询(N)
对象资源管理器
连接
SQLQuery1.sql - (l...dministrator (54))*
统计信息
dbo.Administrator
dbo.Book
dbo.BookIn
dbo.Borrow
列
键
BorrowPk
BorrowFK1
BorrowFK2
BorrowFK3
约束
触发器
索引
统计信息
dbo.Reader
dbo.ReaderLevel
dbo.ReturnBook
列
键
ReturnBookPk
ReturnBookFK1
ReturnBookFK2
ReturnBookFK3
约束
触发器
索引
消息
查询已完成，但有错误。
(local) (10.50 RTM)
WIN-I65Q2TB10OV\Admini...
136+FHT
00:00:00
0 行
就绪
12:53



Admin_Book 表及其外键约束：

```
create table Admin_Book
(   adminNo         char(10)        not null,
    bookNo          char(10)        not null,          --书号
    operationDate   datetime   not null,
    reason          varchar(90)    not null,
    constraint Admin_BookPk primary key(adminNo,bookNo,operationDate)
)
alter table Admin_Book
add   constraint   Admin_BookFK   FOREIgn   key(adminNo)   references
Administrator(adminNo)
```

Admin_Reader 表及其外键：

```
create table Admin_Reader
(     adminNo              char(10)              not null,
      readerNo             char(10)              not null,
      operationDate        datetime      not null,
      reason               varchar(90)       not null,
      records              varchar(15)           not null,
      constraint Admin_ReaderPk primary key(adminNo,readerNo,operationDate )
)
alter table Admin_Reader
add constraint Admin_ReaderFK foreign key(adminNo) references Administrator(adminNo)
```

Administrator 表：

```
create table Administrator
(     adminNo       char(10)       not null,
      adminPassword       char(10) not null,
      adminName       varchar(20)    not null,
      phone           varchar(15)    null,
```

```
    constraint AdministratorPk primary key(adminNo)
)
```

Book 表及其外键约束：

```
create table Book
(   bookNumber      char(10)        not null,           --书号
    bookName        varchar(20)     not null,           --书名
    author          varchar(20)     not null,           --作者
    publish         varchar(30)     not null,           --出版社名称
    publishDate     datetime        not null,           --出版日期
    version         numeric         not null,           --版次
    price           numeric         not null,           --单价
    contents        varchar(300)    not null,           --内容简介
    classNum        char(10)        not null,           --分类号
    bookNo          char(10)        not null,           --索书号
    bookCount       numeric         default 0   not null,
    regNum          char(10)        not null,
    stackRoom       varchar(20)     not null            --书库
    check(stackRoom in ('一号书库','二号书库','三号书库','全部书库')),
    constraint ReaderPk primary key(bookNumber)
)

alter table Book
add constraint BookFK1 foreign key(classNum) references Store(classNum)
```

图书入库 BookIn 表及外键约束：

```
create table BookIn
(
   storeNo     integer identity(1,1)     not null,           --入库单号
   bookNumber  char(10)          not null,           --书号
   storeCount      numeric     not null,           --入库数量
   constraint BookInPk primary key (storeNo,bookNumber)
）
alter table BookIn
add constraint BookInFk1 foreign key(bookNumber) references Book(bookNumber),
constraint BookInFk2 foreign key(storeNo) references StoreSheet(storeNo)
```

借书清单表 Borrow 及其外键约束：

```
create table Borrow
(
       readerNumber     char(10)    not null,           --借书证号
       adminNo      char(10)    not null,           --管理员编号
       bookNumber       char(10)        not null,           --书号
       borrowDate       datetime        not null,           --借阅时间
       borrowCount      numeric         not null,
       constraint BorrowPk primary key(readerNumber,adminNo,bookNumber, borrowDate )
)
```

```
alter table Borrow
add constraint BorrowFK1 foreign key(readerNumber) references Reader(readerNumber),
constraint BorrowFK2 foreign key(adminNo) references Administrator(adminNo),
constraint BorrowFK3 foreign key(bookNumber) references Book(bookNumber)
```

读者表 Reader 及其外键约束：

```
create table Reader
(     readerNumber       char(10)       not null,                --借书证号
      create_date        datetime       not null,                --办证日期
      userPassword       char(10)       not null,                --读者密码
      userName           varchar(20)    not null,                --读者姓名
      Id_Number          char(18)       not null,                --身份证号码
      sex                char(2)        null                     --性别
      check(sex in ('男','女')),
      birthday           datetime       null,                    --生日
      workplace          varchar(30)    null,                    --工作单位
      address            varchar(30)    null,                    --通讯地址
      postcode           varchar(10)    null,                    --邮政编码
      phone              char(13)       null,
      work               varchar(30)    null,
      readerLevel        varchar(10)    not null                 --读者等级
check(readerLevel in ('学生','老师','普通读者'))
constraint ReaderPk primary key(readerNumber)
)
ALTER TABLE Reader
  ADD    CONSTRAINT ReaderFk1 FOREIGN KEY(readerLevel)
REFERENCES    ReaderLevel (readerLevel)
```

ReaderLevel 表：

```
create table ReaderLevel
(     readerLevel        varchar(10)       not null              --读者等级
        check(readerLevel in ('学生','老师','普通读者')),
      borrowRange        varchar(20)    not null                 --借书范围
      check(borrowRange in ('一号书库','二号书库','三号书库','全部书库')),
      maxCount           numeric        not null,                --最大借阅量
      dateDeadline       numeric        not null,                --借书期限（天）
      constraint LevelPk primary key(readerLevel)
)
```

还书清单表 ReturnBook 及其外键约束：

```
create table ReturnBook
(
        readerNumber       char(10)       not null,              --借书证号
        adminNo            char(10)       not null,              --管理员编号
        bookNumber         char(10)       not null,              --书号
        returnDate         datetime       not null,              --归还时间
        returnCount        numeric        not null,
```

```
        constraint ReturnBookPk primary key(readerNumber,adminNo,bookNumber,returnDate)
)
alter table ReturnBook
add constraint ReturnBookFK1 foreign key(readerNumber) references Reader(readerNumber),
  constraint ReturnBookFK2 foreign key(adminNo) references Administrator(adminNo),
    constraint ReturnBookFK3 foreign key(bookNumber) references Book(bookNumber)
```

图书分类表：

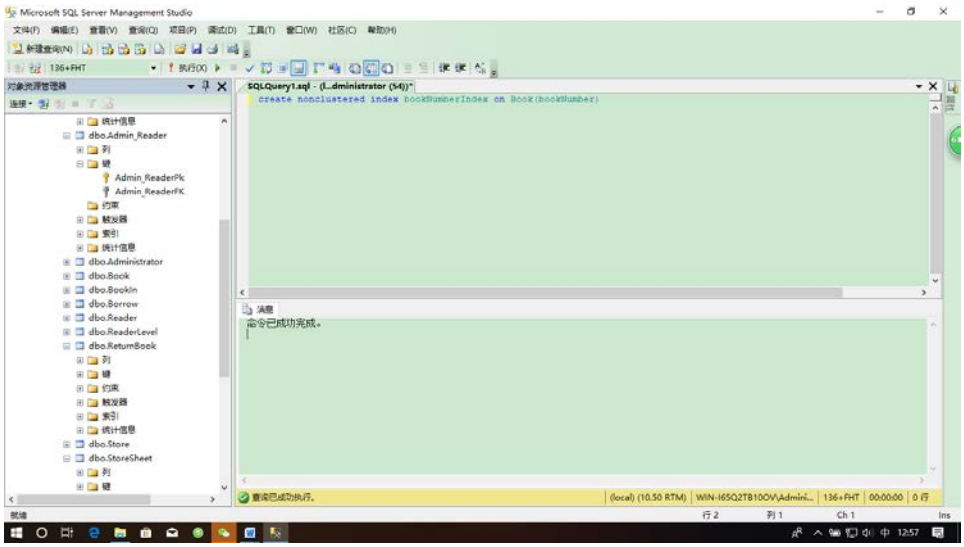
```
create table Store
(     className          char(10)       not null,
     className          varchar(20) not null,
     constraint StorePk primary key(className)
)
```

入库单表：

```
create table StoreSheet
(
storeNo     integer identity(1,1)       not null,       --入库单号
      adminNo     char(10)       not null,       --操作员编号
      storeDate   datetime       not null,       --入库时间
      constraint StoreSheetPk primary key(storeNo )
)
```

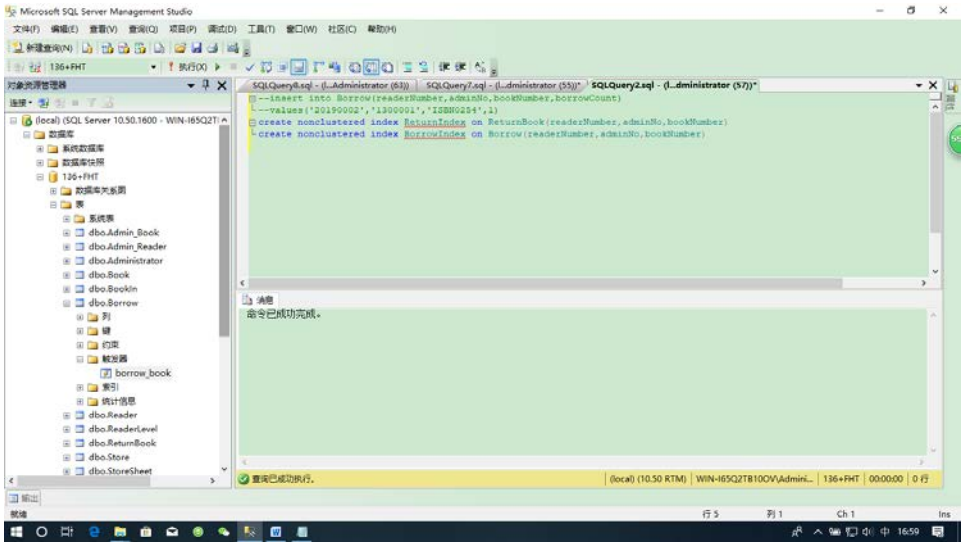
2. 索引的建立

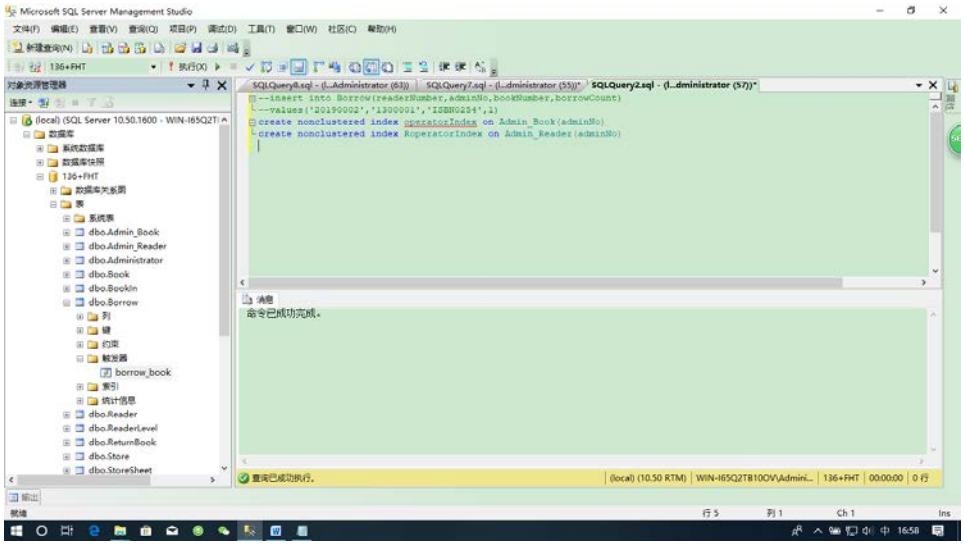
```
create nonclustered index bookNumberIndex on Book(bookNumber)
```



```
create nonclustered index operatorIndex on Admin_Book(adminNo)
create nonclustered index RoperatorIndex on Admin_Reader(adminNo)
create nonclustered index ReturnIndex on
ReturnBook(readerNumber,adminNo,bookNumber)
create        nonclustered        index        BorrowIndex        on
```

```
Borrow(readerNumber,adminNo,bookNumber)
```





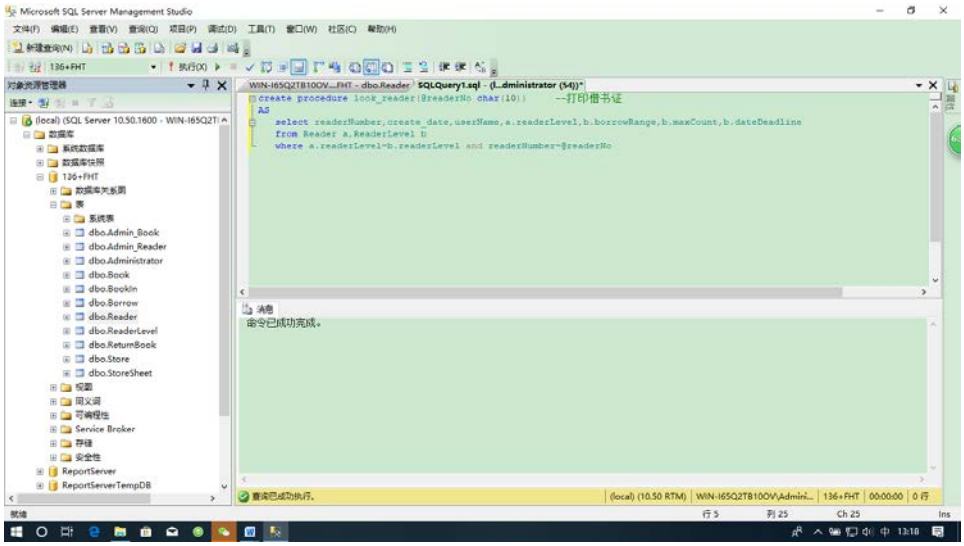
3. 存储过程的建立

由于复习时间的紧迫，存储过程就弄了几个，其实还可以通过书名、作者等等来检索图书的信息

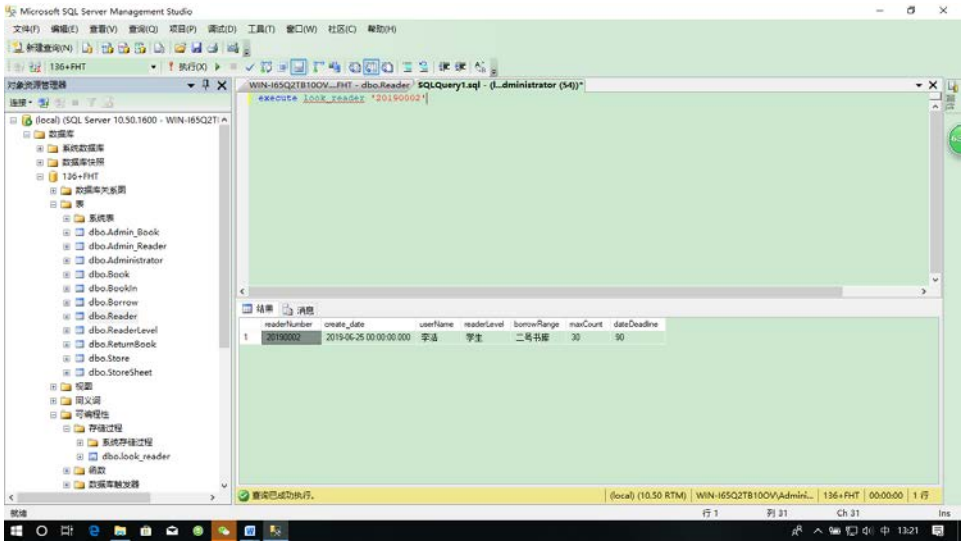
根据借书证号打印借书证：

```
create procedure look_reader(@readerNo char(10))     --打印借书证
AS
   select
readerNumber,create_date,userName,a.readerLevel,b.borrowRange,b.maxCo
unt,b.dateDeadline
```

```
   from Reader a,ReaderLevel b
   where a.readerLevel=b.readerLevel and readerNumber=@readerNo
```

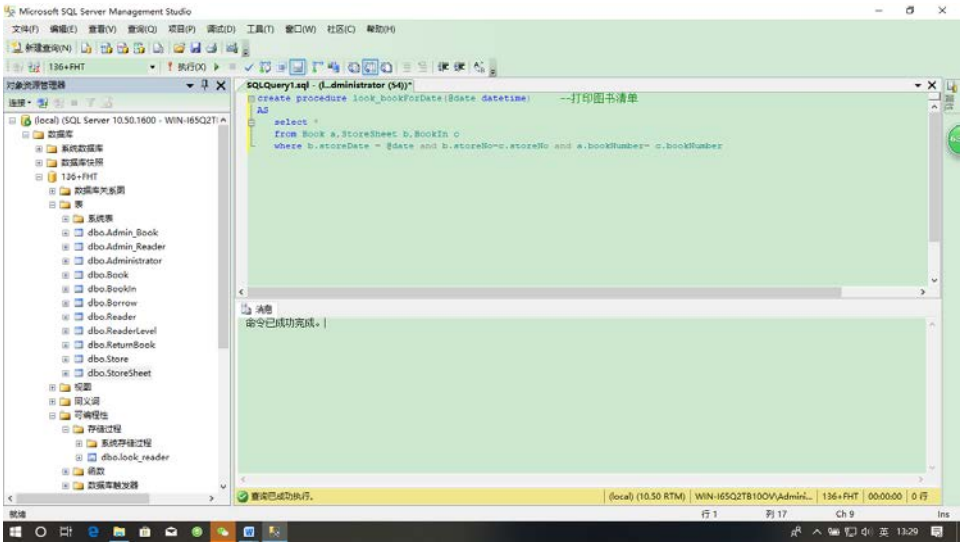


```
execute look_reader '20190002'
```

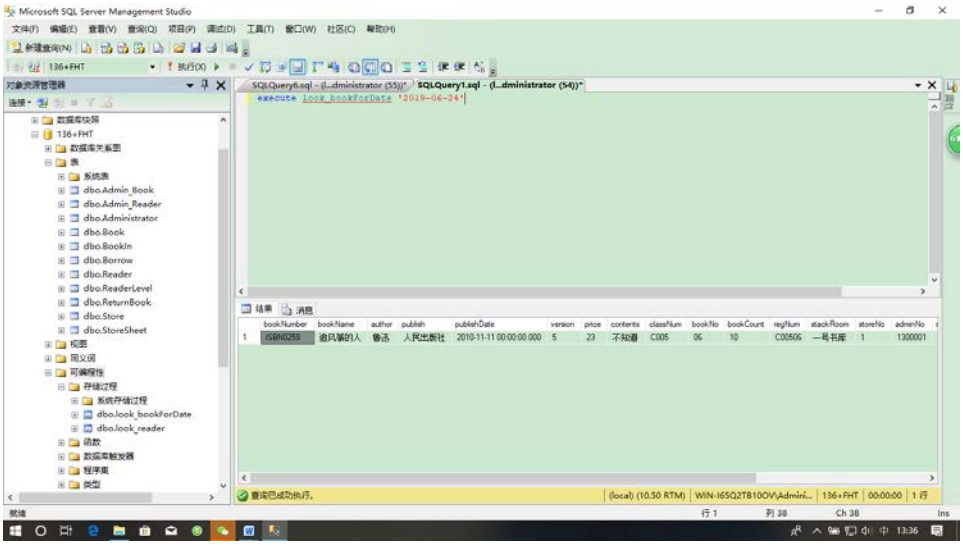


根据入库时间打印图书清单：

```
create procedure look_bookForDate(@date datetime)     --打印图书清单
AS
   select *
   from Book a,StoreSheet b,BookIn c
   where b.storeDate = @date and b.storeNo=c.storeNo and a.bookNumber=
c.bookNumber
```

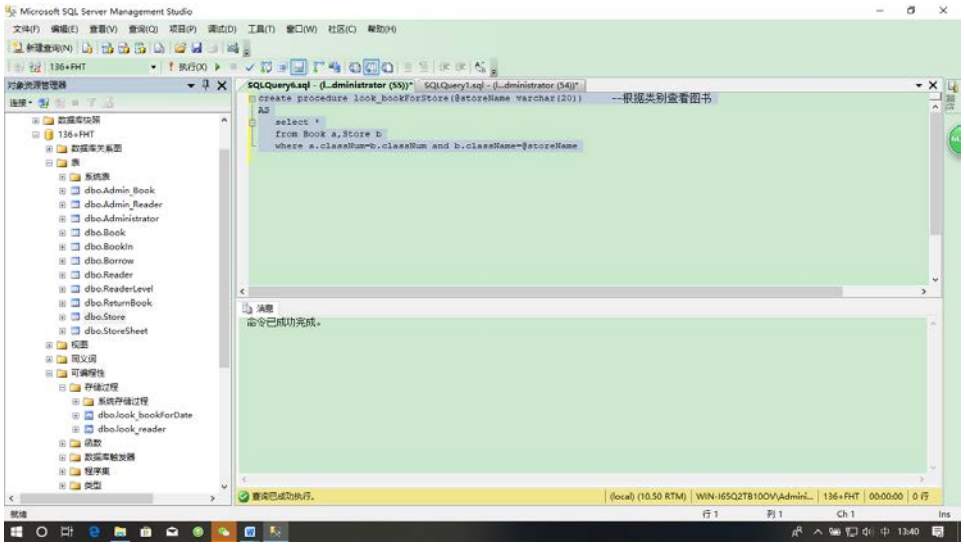


```
execute look_bookForDate '2019-06-24'
```

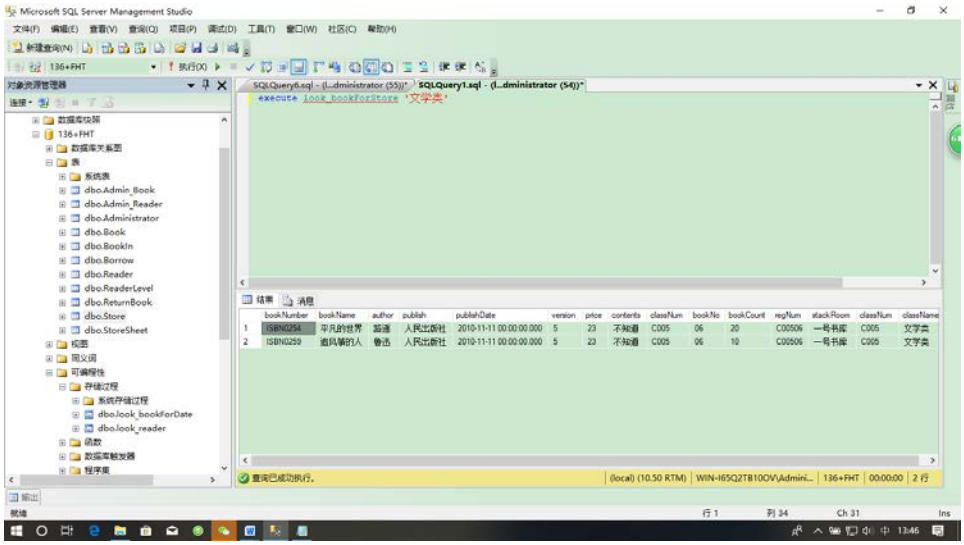


根据图书分类查看图书：

```
create procedure look_bookForStore(@storeName varchar(20))     --根据类别查看图书
AS
   select *
   from Book a,Store b
   where a.className=b.className and b.className=@storeName
```

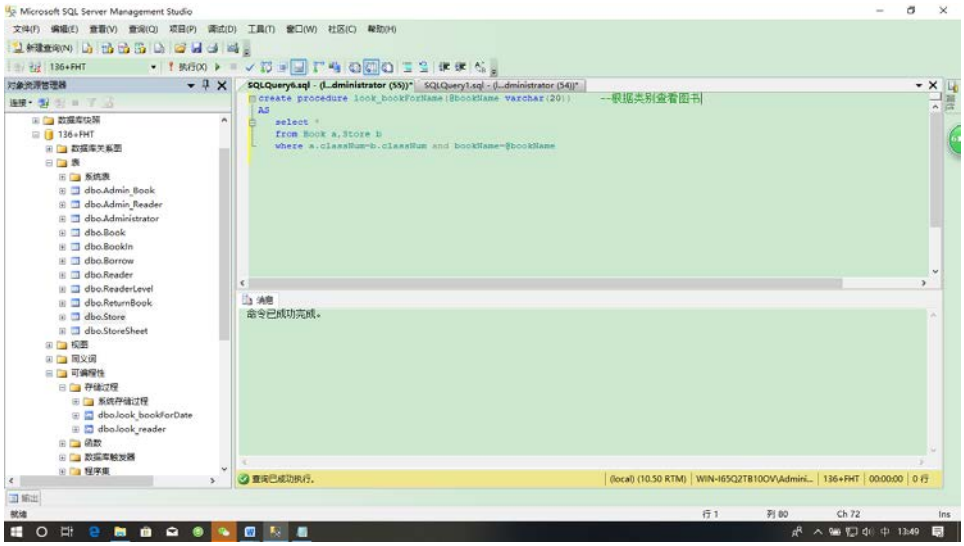


```
execute look_bookForStore '文学类'
```

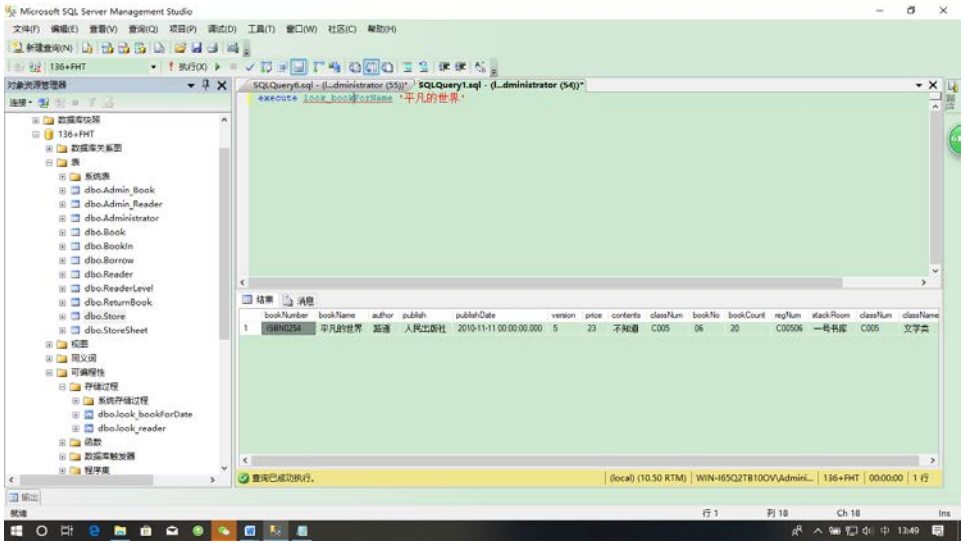


根据书名查询书：

```
create procedure look_bookForName(@bookName varchar(20))     --根据书名查看图书
AS
   select *
   from Book a,Store b
   where a.classNum=b.classNum and bookName=@bookName
```

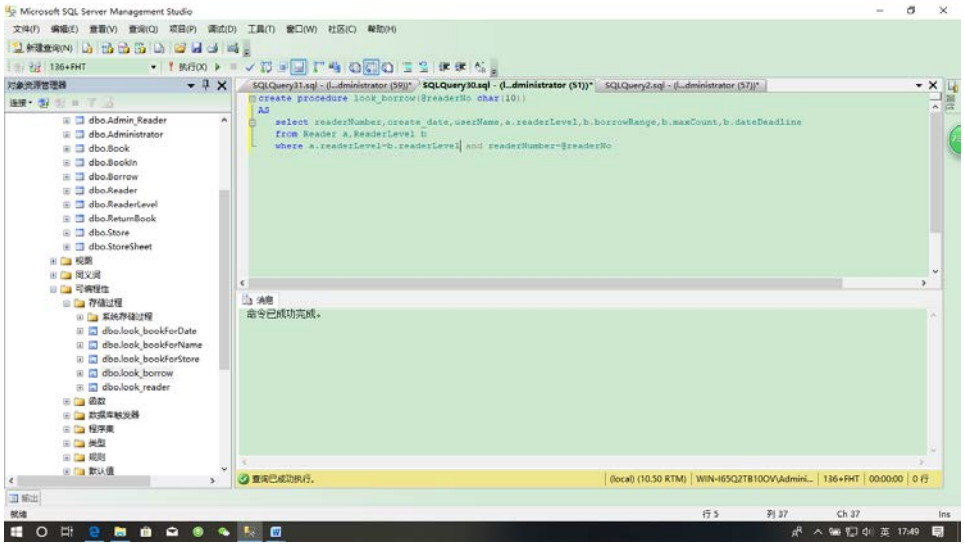


```
execute look_bookForName '平凡的世界'
```



根据读者借书证查看以前借书的情况：

```
create procedure look_borrow(@readerNo char(10))
AS
   select
readerNumber,create_date,userName,a.readerLevel,b.borrowRange,b.maxCo
unt,b.dateDeadline
   from Reader a,ReaderLevel b
   where a.readerLevel=b.readerLevel and readerNumber=@readerNo
```



批注 [CU1]: 触发器后面紧跟着是实例运行结果好一些

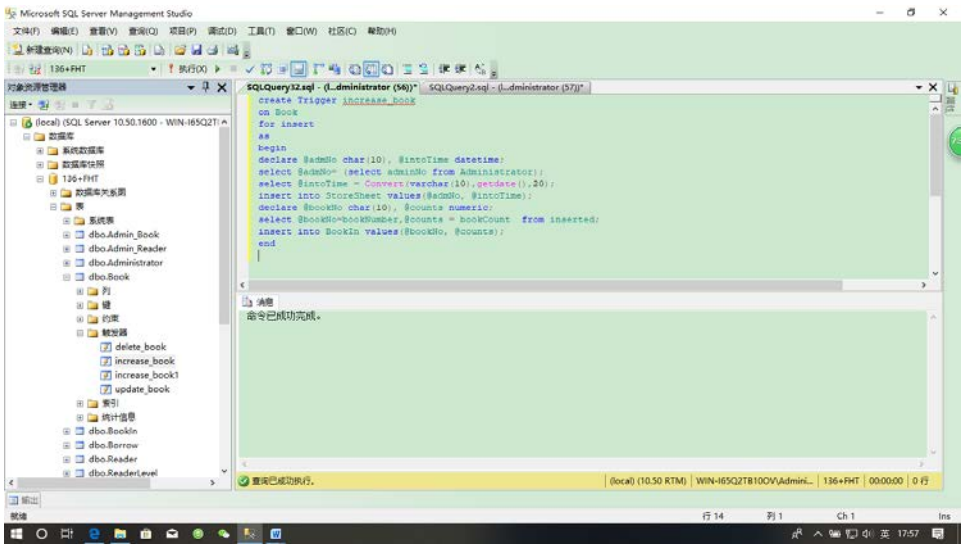
4. 触发器建立和测试

两个图书的录入触发器：

```
create Trigger increase_book
on Book
for insert
as
begin
declare @admNo char(10), @intoTime datetime;
select @admNo= (select adminNo from Administrator);
select @intoTime = Convert(varchar(10),getdate(),20);
insert into StoreSheet values(@admNo, @intoTime);
declare @bookNo char(10), @counts numeric;
select @bookNo=bookNumber,@counts = bookCount  from inserted;
insert into BookIn values(@bookNo, @counts);
end


ALTER Trigger [dbo].[increase_book1]
on [dbo].[Book]
for insert
as
begin
declare @admNo char(10), @operateTime datetime;
select @admNo= (select adminNo from Administrator);
select @operateTime= Convert(varchar(10),getdate(),20);
insert into Admin_Book values(@admNo,@operateTime,'进行了图书信息的录入!
```

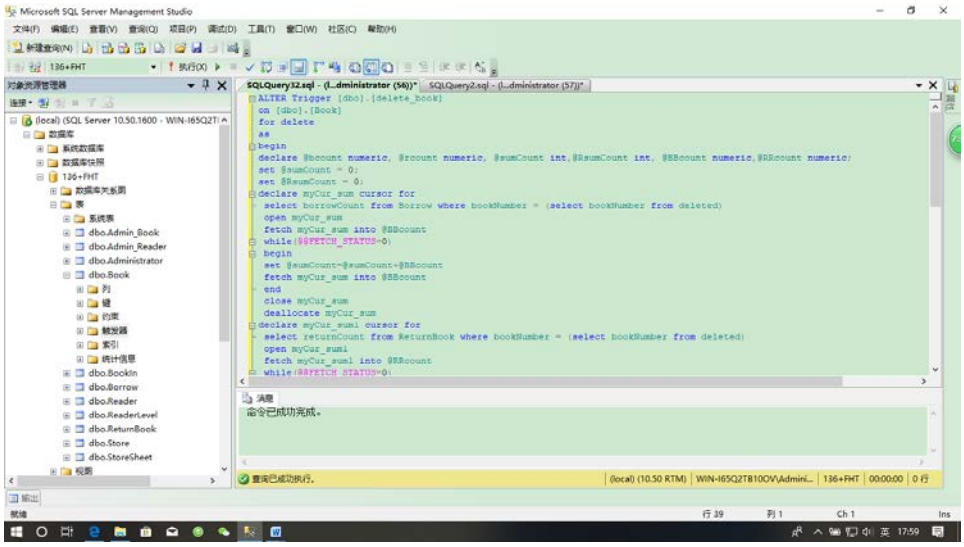
')
end



图书删除触发器：

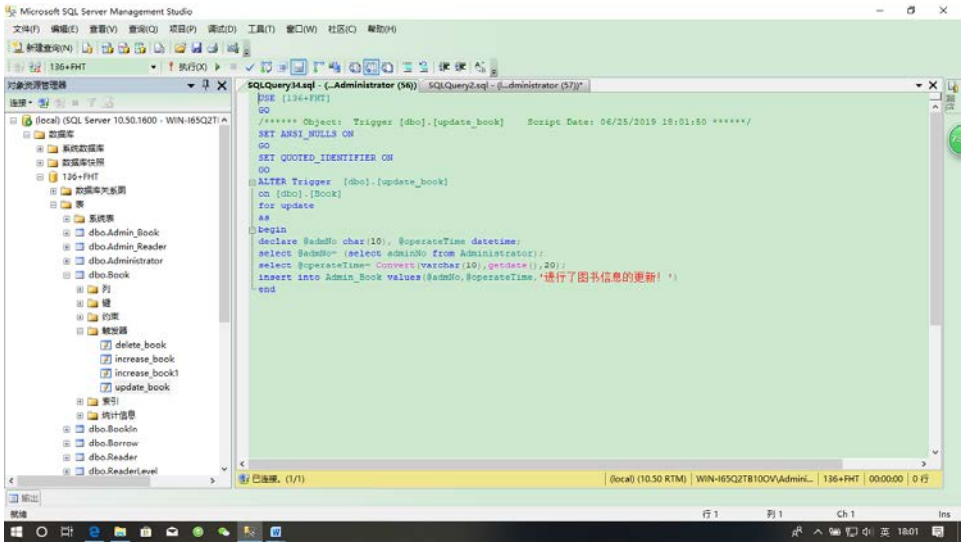
```
ALTER Trigger [dbo].[delete_book]
on [dbo].[Book]
for delete
as
begin
declare @bcount numeric, @rcount numeric, @sumCount int,@RsumCount int, @BBcount numeric,@RRcount numeric;
set @sumCount = 0;
set @RsumCount = 0;
declare myCur_sum cursor for
  select borrowCount from Borrow where bookNumber = (select bookNumber from deleted)
  open myCur_sum
  fetch myCur_sum into @BBcount
  while(@@FETCH_STATUS=0)
  begin
  set @sumCount=@sumCount+@BBcount
  fetch myCur_sum into @BBcount
  end
  close myCur_sum
  deallocate myCur_sum
declare myCur_sum1 cursor for
  select returnCount from ReturnBook where bookNumber = (select bookNumber from deleted)
  open myCur_sum1
  fetch myCur_sum1 into @RRcount
  while(@@FETCH_STATUS=0)
```

```
 begin
 set @RsumCount=@RsumCount+@RRcount
 fetch myCur_sum1 into @RRcount
 end
 close myCur_sum1
 deallocate myCur_sum1
if @RsumCount<@sumCount
rollback
else
declare @admNo char(10), @operateTime datetime;
select @admNo= (select adminNo from Administrator);
select @operateTime= Convert(varchar(10),getdate(),20);
insert into Admin_Book values(@admNo,@operateTime,'进行了图书信息的删除！')
end
```



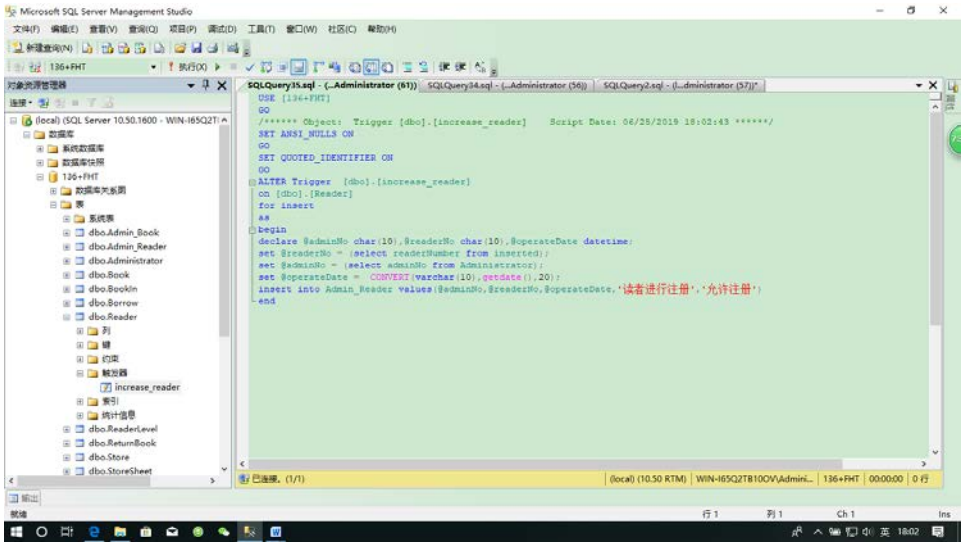
图书更新触发器：

```
ALTER Trigger    [dbo].[update_book]
on [dbo].[Book]
for update
as
begin
declare @admNo char(10), @operateTime datetime;
select @admNo= (select adminNo from Administrator);
select @operateTime= Convert(varchar(10),getdate(),20);
insert into Admin_Book values(@admNo,@operateTime,'进行了图书信息的更新！')
end
```



读者注册触发器：

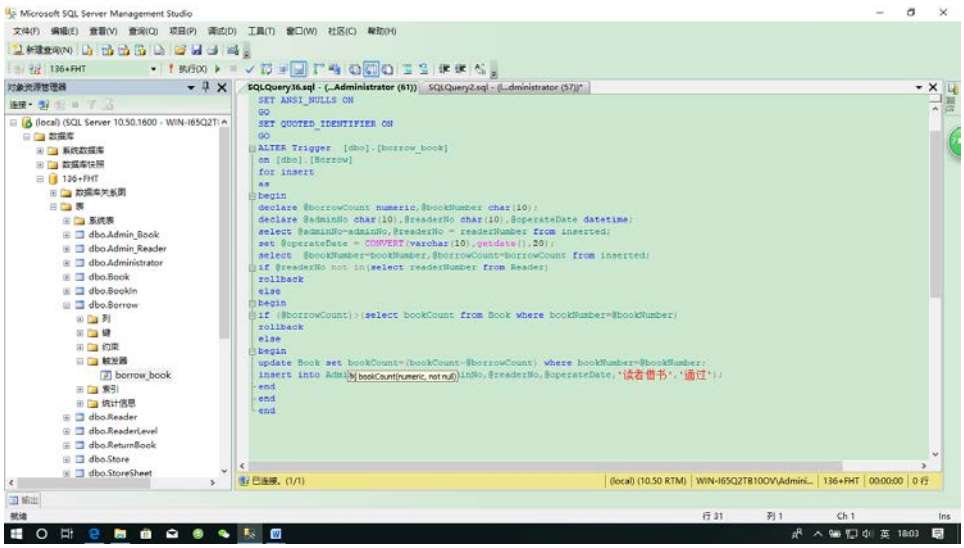
```
ALTER Trigger    [dbo].[increase_reader]
on [dbo].[Reader]
for insert
as
begin
declare @adminNo char(10),@readerNo char(10),@operateDate datetime;
set @readerNo = (select readerNumber from inserted);
set @adminNo = (select adminNo from Administrator);
set @operateDate =    CONVERT(varchar(10),getdate(),20);
insert into Admin_Reader values(@adminNo,@readerNo,@operateDate,'读者进行注册','允许注册')
end
```



读者借书的两个触发器：

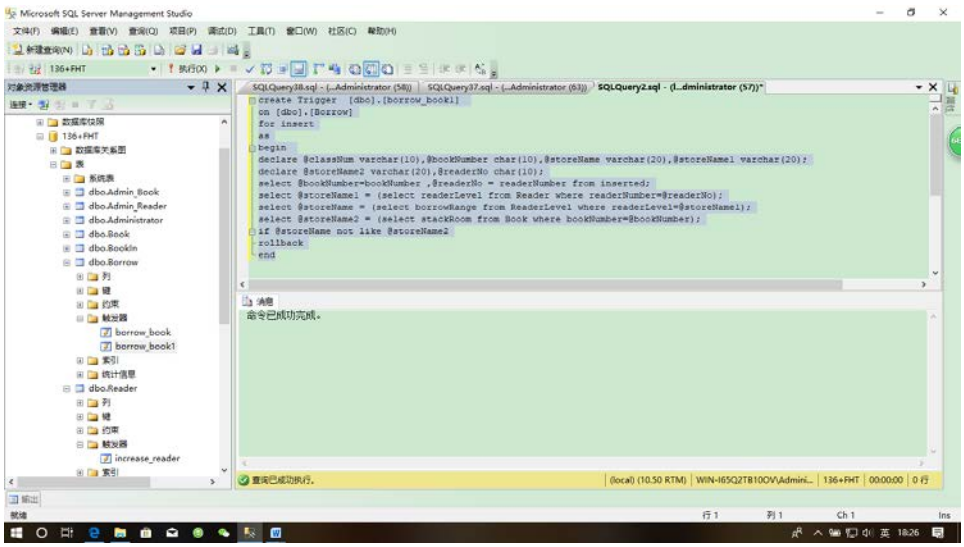
借书一定要先注册，未注册的是不允许借书的，借书的数量不能超过库存容量，否则不允许借书，而且根据读者等级限制结束范围和借书数量

```
ALTER Trigger    [dbo].[borrow_book]
on [dbo].[Borrow]
for insert
as
begin
declare @borrowCount numeric,@bookNumber char(10);
declare @adminNo char(10),@readerNo char(10),@operateDate datetime;
select @adminNo=adminNo,@readerNo = readerNumber from inserted;
set @operateDate = CONVERT(varchar(10),getdate(),20);
select    @bookNumber=bookNumber,@borrowCount=borrowCount from inserted;
if @readerNo not in(select readerNumber from Reader)
rollback
else
begin
if (@borrowCount)>(select bookCount from Book where bookNumber=@bookNumber)
rollback
else
begin
update Book set bookCount=(bookCount-@borrowCount) where bookNumber=@bookNumber;
insert into Admin_Reader values(@adminNo,@readerNo,@operateDate,'读者借书','通过');
end
end
end
```



```
create Trigger  [dbo].[borrow_book1]
on [dbo].[Borrow]
```

```
for insert
as
begin
declare @className varchar(10),@bookNumber char(10),@storeName
varchar(20),@storeName1 varchar(20);
declare @storeName2 varchar(20),@readerNo char(10);
select @bookNumber=bookNumber ,@readerNo = readerNumber from inserted;
select @storeName1 = (select readerLevel from Reader where
readerNumber=@readerNo);
select @storeName = (select borrowRange from ReaderLevel where
readerLevel=@storeName1);
select @storeName2 = (select stackRoom from Book where
bookNumber=@bookNumber);
if @storeName not like @storeName2
begin
print '该书超出借书范围'
rollback
end
end
```

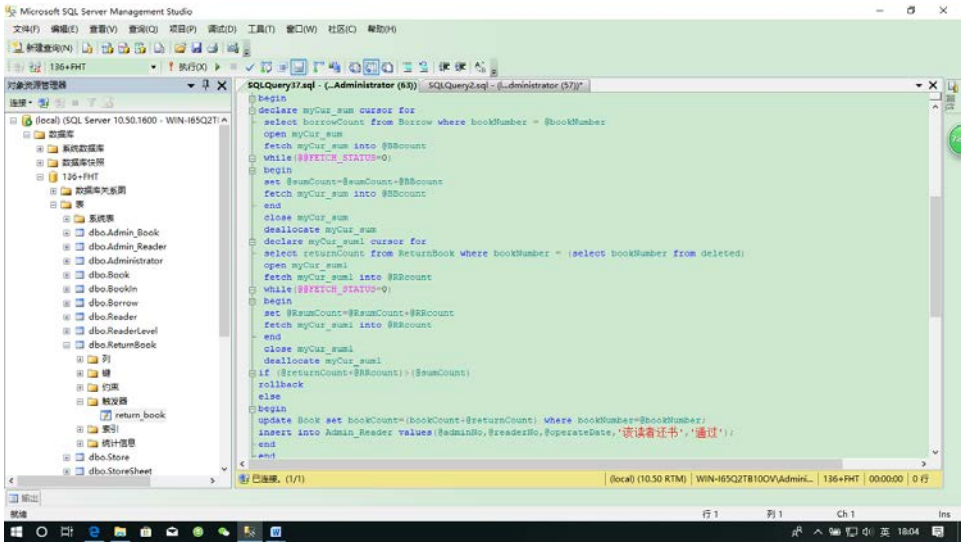


读者还书触发器：

还书的数量不能超过借书的数量，然后不用考虑还书时间比结束时间早的问题，因为获取的是系统时间，还书要是没这读者的相关信息，也无法还书

```
ALTER Trigger  [dbo].[return_book]
on [dbo].[ReturnBook]
for insert
as
begin
declare  @returnCount  numeric,@bookNumber  char(10),@sumCount  numeric,@BBcount
```

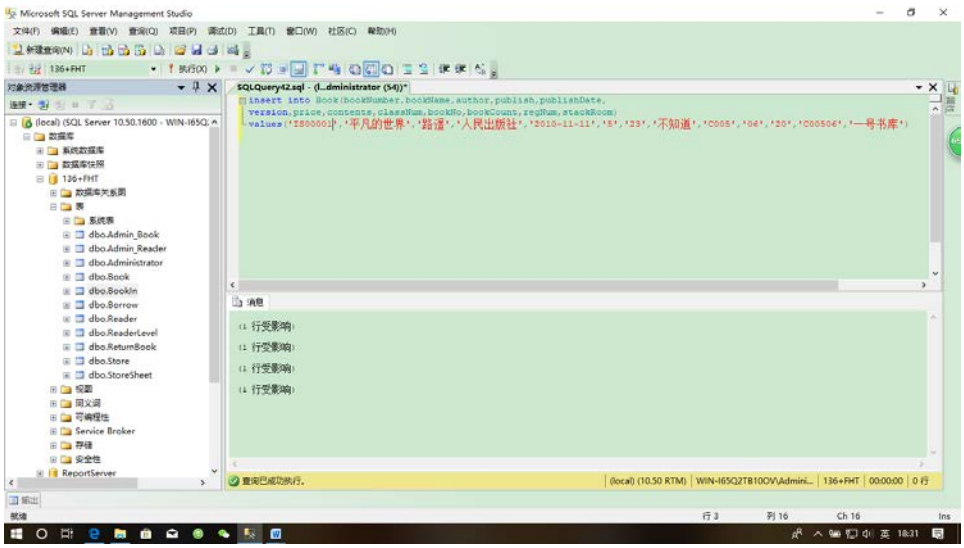
```
numeric;
declare @RsumCount numeric,@RRcount numeric;
declare @adminNo char(10),@readerNo char(10),@operateDate datetime;
select @adminNo=adminNo,@readerNo = readerNumber from inserted;
set @operateDate = CONVERT(varchar(10),getdate(),20);
select    @bookNumber=bookNumber,@returnCount=returnCount from inserted;
if @readerNo not in(select readerNumber from Reader)
rollback
else
begin
declare myCur_sum cursor for
  select borrowCount from Borrow where bookNumber = @bookNumber
  open myCur_sum
  fetch myCur_sum into @BBcount
  while(@@FETCH_STATUS=0)
  begin
  set @sumCount=@sumCount+@BBcount
  fetch myCur_sum into @BBcount
  end
  close myCur_sum
  deallocate myCur_sum
  declare myCur_sum1 cursor for
  select returnCount from ReturnBook where bookNumber = (select bookNumber from deleted)
  open myCur_sum1
  fetch myCur_sum1 into @RRcount
  while(@@FETCH_STATUS=0)
  begin
  set @RsumCount=@RsumCount+@RRcount
  fetch myCur_sum1 into @RRcount
  end
  close myCur_sum1
  deallocate myCur_sum1
if (@returnCount+@RRcount)>(@sumCount)
rollback
else
begin
update Book set bookCount=(bookCount+@returnCount) where bookNumber=@bookNumber;
insert into Admin_Reader values(@adminNo,@readerNo,@operateDate,'该读者还书','通过');
end
end
end
```

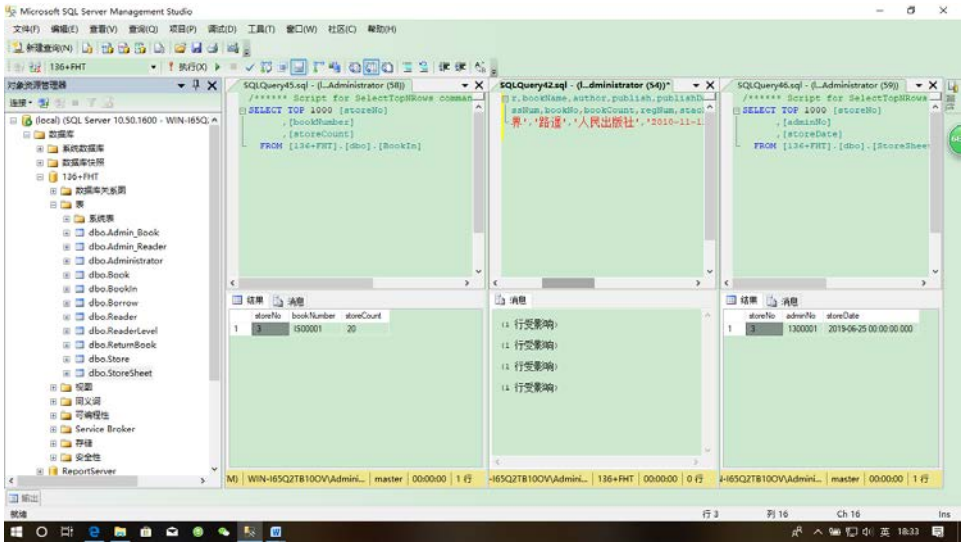


触发器的触发：

图书录入：

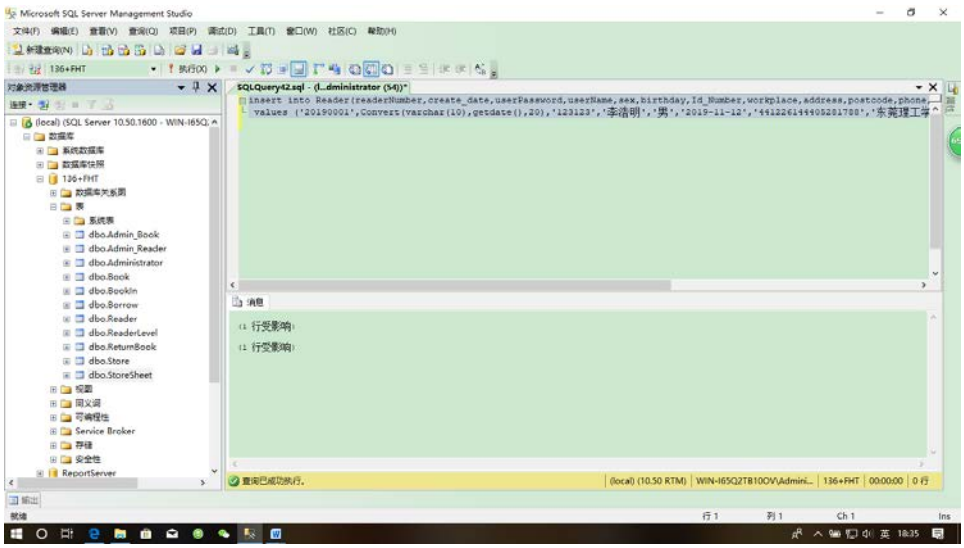
```
insert into Book(bookNumber,bookName,author,publish,publishDate,
version,price,contents,classNum,bookNo,bookCount,regNum,stackRoom)
values('IS00001','平凡的世界','路遥','人民出版社','2010-11-11','5','23','
不知道','C005','06','20','C00506','一号书库')
```

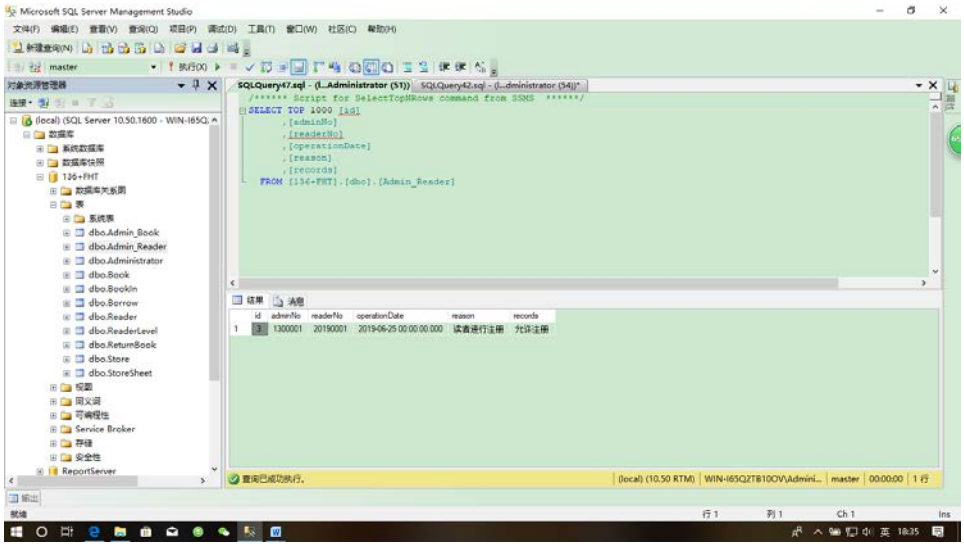


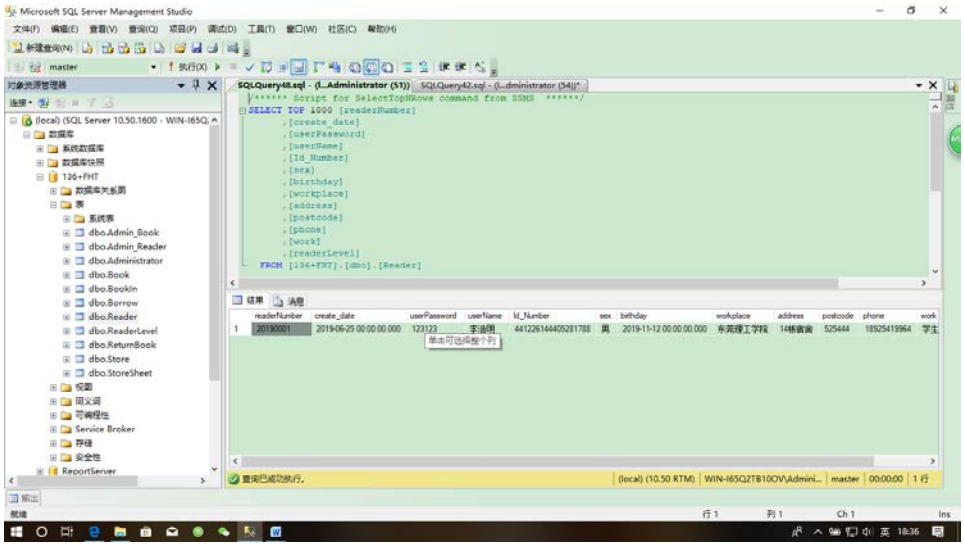


读者注册：

```
insert into
Reader(readerNumber,create_date,userPassword,userName,sex,birthday,Id_Number,workplace,address,postcode,phone,work,readerLevel)
 values ('20190001',Convert(varchar(10),getdate(),20),'123123','李浩明','男','2019-11-12','441226144405281788','东莞理工学院','14栋宿舍','525444','18925419964','学生','学生')
```



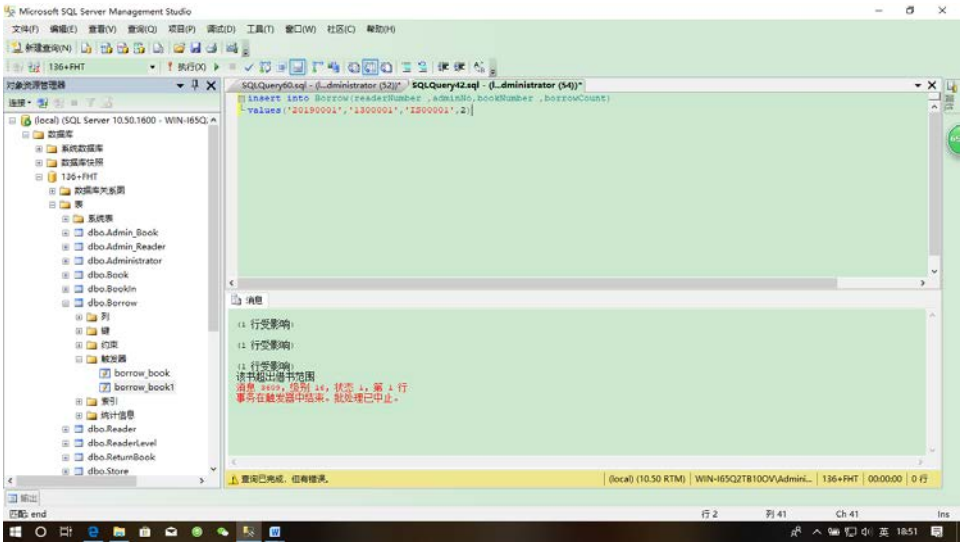




读者借书：

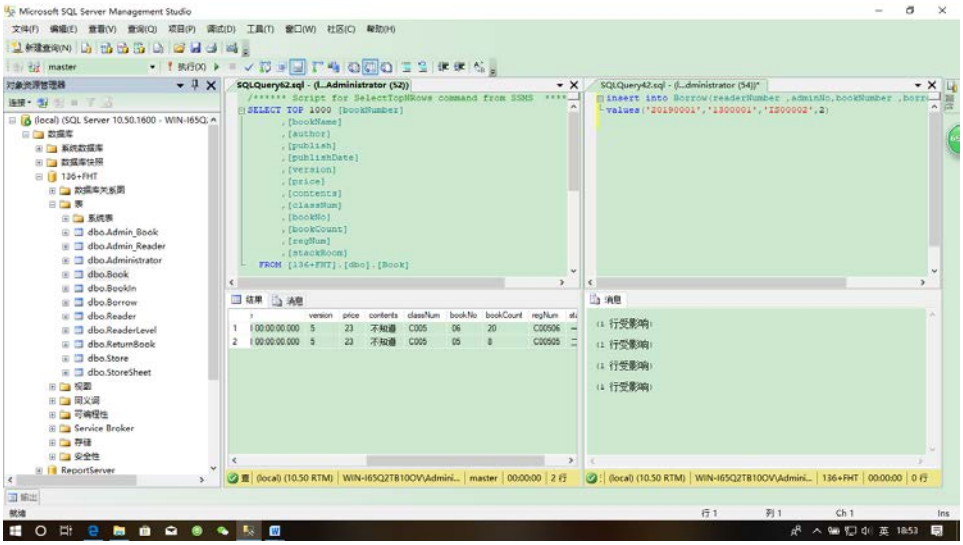
由于触发器根据学生课借书范围，该图书在一号书库，而学生的借书范围为二号书库，所以不能允许借书

```
insert into Borrow(readerNumber ,adminNo,bookNumber ,borrowCount)
values('20190001','1300001','IS00001',2)
```



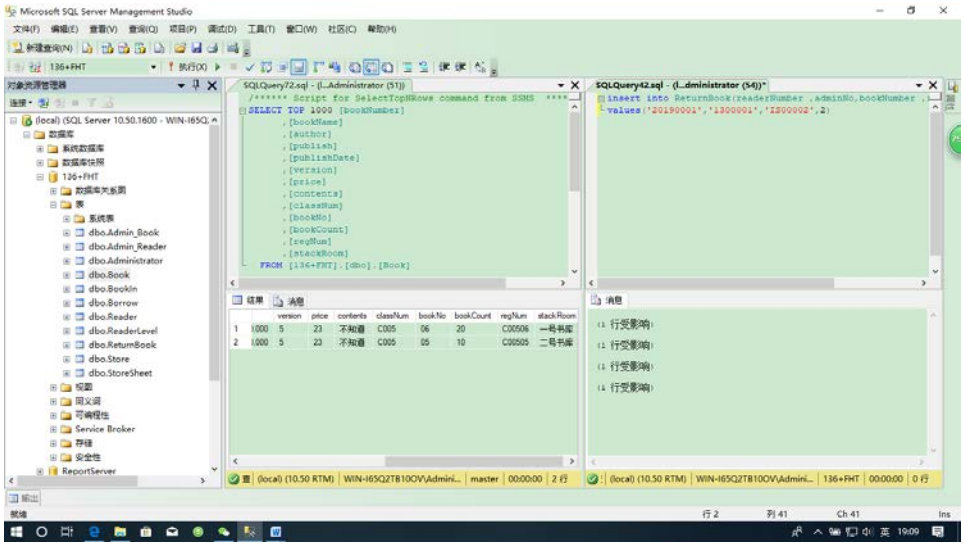
```
insert into Borrow(readerNumber ,adminNo,bookNumber ,borrowCount)
values('20190001','1300001','IS00002',2)
```

自动修改库存和在图书操作记录增加一条记录，和读者操作表增加记录，该记录是管理员审批通过借书



读者还书：

```
insert into ReturnBook(readerNumber ,adminNo,bookNumber ,returnCount)
values('20190001','1300001','IS00002',2)
```



六.　总结与体会

这一次实验基本都运用了老师所教的知识，涉及范围广，然后可能是因为我对课本不是很熟悉，在做实验的时候还是得翻书查看一下语句什么的概念等等，建数据库的时候先不要建立外键，因外这可能会使你后面建立触发器，存储过程等测试的时候会产生冲突，经过这一次实验，感觉还是得重复看书理解课本内容，不然就很容易产生忘记，希望以后能努力再努力吧！