电子科技大

实验报告

课程名称：数据库原理及应用

学　　院：信息与软件工程学院

专　　业：软件工程

指导教师：　胡成华

学生姓名：　韩政君

学　　号：2021090922022

实验成绩：

日　　期：　　2023 年　5　月　17　日

电 子 科 技 大 学

实 验 报 告

一、实验一：图书销售管理系统数据库 SQL 应用编程

二、实验室名称：信软学院楼西 400　　实验时间：2023-04-27，2023-05-04

三、实验目的

结合图书销售管理系统数据库开发项目案例，开展数据库 SQL 应用编程实践，培养数据库 SQL 操作访问、存储过程与触发器处理的数据库编程能力。

四、实验原理

首先对图书销售管理系统进行数据需求分析，定义组成系统数据结构的实体、实体属性以及实体之间的关系。

采用实体关系图（E-R 模型图）方法来展示图书销售管理系统的概念数据模型与逻辑数据模型。

利用 PowerDesigner 数据库软件系统进行系统物理数据模型设计，对设计的图书销售管理系统数据库模型进行检验与完善，并对系统进行数据库设计，给出设计方案。

基于数据库设计方案，通过 SQL 编程执行来完成对数据库的创建与数据访问操作以及相应的后端编程操作。

在本实验中，使用 SQL 语句完成对数据库、关系表、索引、视图、触发器、存储过程的创建，

并编写 SQL 语句对数据库表进行数据的增删查改操作，以及利用视图、存储过程、触发器实现业务数据处理。

五、实验内容

针对图书销售管理系统基本需求，开发实现图书销售管理系统数据库，具体实验内容如下:

1.基于图书销售管理系统基本数据需求，给出图书销售管理系统数据库设计方案。

2.在数据库服务器中，执行 SQL 创建图书销售管理系统数据库 BookSale。

3.在数据库 BookSale 中，执行 SQL 创建数据库表、视图、索引等对象。

4.在数据库 BookSale 中，执行 SQL 进行数据增、删、查、改访问操作。

5.在数据库 BookSale 中，采用 PL/pgSQL 语言编写存储过程函数 Pro_CurrentSale，实现当日图书销售量及销售金额汇总统计。

6.在数据库 BookSale 中，采用 PL/pgSQL 语言编写过程语句块，实现对存储过程函数 Pro_CurrentSale 的调用，并输出统计结果。

7.在数据库 BookSale 中，采用 PL/pgSQL 语言编写编写图书销售表 Insert 触发器 Tri_InsertSale，实现图书库存数据同步修改处理。

8.在数据库 BookSale 中，对图书销售表 Insert 触发器 Tri_InsertSale 程序进行功能验证。

在实验计算机上，利用 pgAdmin4 数据库管理工具及 SQL、PL/pgSQL 语言，完成图书销售管理系统数据库应用编程操作，同时记录实验过程的步骤、操作、运行结果界面等数据，为撰写实验报告提供素材。

六、实验设备及环境

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows10 / Windows 11

管理工具： pgAdmin4

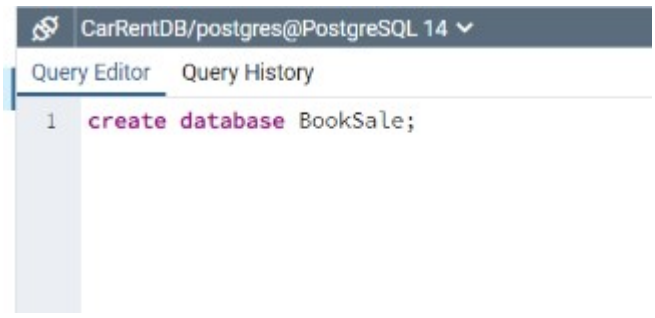
DBMS 系统： PostgreSQL 15.1

七、实验步骤

（1）图书销售管理系统数据库 BookSale 创建操作。

采用 SQL 语句执行方式，创建图书销售管理系统数据库 BookSale。

```
create database BookSale;
```



（2）在图书销售管理系统数据库 BookSale 中创建数据库表、视图、索引等对象

采用 SQL 语句执行方式，创建图书表 Book、作者表 Author、出版社表 Publisher、库存

流水表 Bookstock、客户表 Customer、销售表 Sale，以及各表主键和外键的创建，并为各表创建索引。

A．创建表

```
create table Author
(
    Author_ID       char(18) not null,
    Author_Name         varchar(20)  not null,
    Author_Gender   char(2)      not null,
    constraint Author_PK primary key(Author_ID)
);
create table Publisher
(
    Publisher_ID    char(11) not null,
    Publisher_Name varchar(20)  not null,
    Publisher_phone varchar(15)  not null,
    constraint Publisher_PK primary key(Publisher_ID)
);
create table Book
(
    Book_ISBN       char(13) not null,
    Book_Name       varchar(50)  not null,
    Book_Pubdate    date     not null,
    Book_Price      money        not null,
    Book_Stock      int4     not null,
    Author_ID       char(18) null,
    Publisher_ID    char(11) null,
    constraint Book_PK primary key(Book_ISBN)
);
create table Customer
(
    Customer_ID         char(18) not null,
    Customer_Name varchar(20)  not null,
    Customer_phone varchar(15)  not null,
    constraint Customer_PK primary key(Customer_ID)
```

```
);
create table Sale
(
    Sale_ID          varchar(11)  not null,
    Sale_Date        date      not null,
    Sale_Number          int4      not null,
    Sale_Amount          money         not null,
    Book_ISBN      char(13)  null,
    Customer_ID          char(18)  null,
    constraint Sale_PK primary key(Sale_ID)
);
create table Bookstock
(
    Stock_ID         varchar(10)  not null,
    Stock_Date       date      not null,
    Stock_Operation char(4)       not null,
    Stock_Change     int4      not null,
    Book_ISBN      char(13)  null,
    constraint Stock_PK primary key(Stock_ID)
);
```

B. 键的设计

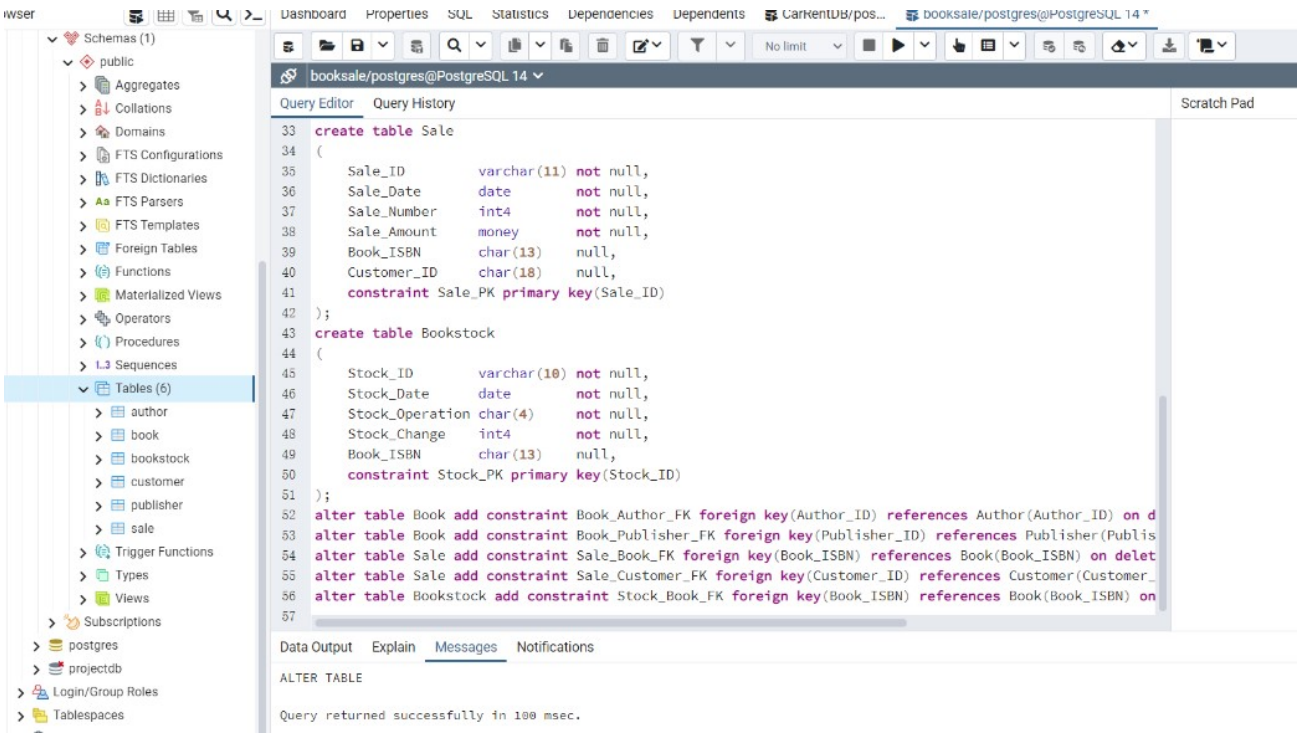
alter table Book add constraint Book_Author_FK foreign key(Author_ID) references Author(Author_ID) on delete restrict on update restrict;

alter table Book add constraint Book_Publisher_FK foreign key(Publisher_ID) references Publisher(Publisher_ID) on delete restrict on update restrict;

alter table Sale add constraint Sale_Book_FK foreign key(Book_ISBN) references Book(Book_ISBN) on delete restrict on update restrict;

alter table Sale add constraint Sale_Customer_FK foreign key(Customer_ID) references Customer(Customer_ID) on delete restrict on update restrict;

alter table Bookstock add constraint Stock_Book_FK foreign key(Book_ISBN) references Book(Book_ISBN) on delete restrict on update restrict;



C. 创建索引

create index Author_Name_Idx on Author(Author_Name);

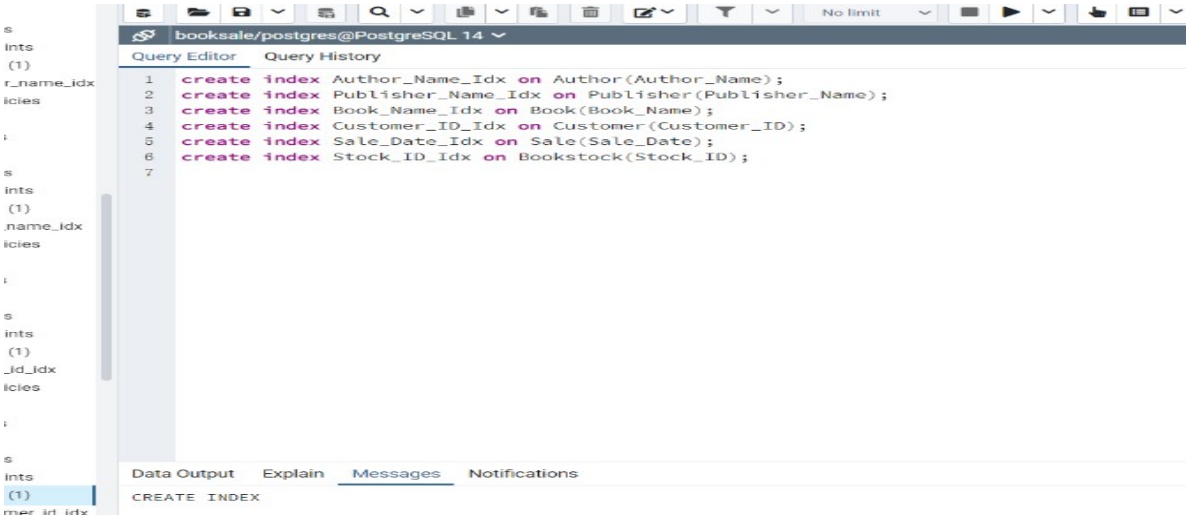
create index Publisher_Name_Idx on Publisher(Publisher_Name);

create index Book_Name_Idx on Book(Book_Name);

create index Customer_ID_Idx on Customer(Customer_ID);

create index Sale_Date_Idx on Sale(Sale_Date);

create index Stock_ID_Idx on Bookstock(Stock_ID);



（3）对图书销售管理系统数据库表进行数据增、删、查、改 SQL 操作。

为 Book、Author、Publisher、Customer 表准备样本数据，采用 SQL 语句执行方式，将样本数据插入到表中。

对各表进行数据修改、删除、查询、统计等访问操作。

A. 插入操作

```
insert into Author values('111111111111111111','张伟','男');
insert into Author values('222222222222222222','王秀英','女');
insert into Author values('333333333333333333','李军','男');
insert into Publisher values('PUB00000001','人民文学出版社','010-11111111');
insert into Publisher values('PUB00000002','上海译文出版社','010-22222222');
insert into Publisher values('PUB00000003','中华书局出版社','010-33333333');
insert into Book values('9787121021961','操作系统:精髓与设计原理','2006-02-01','58','20','111111111111111111
insert into Book values('9787559202253','西方建筑史','2019-12-02','328','30','222222222222222222','PUB0000
insert into Book values('9787532786831','克拉拉与太阳','2021-03-03','68','98','222222222222222222','PUB0000
insert into Book values('9787555911425','字母表谜案','2021-05-07','42','150','333333333333333333','PUB0000
insert into Customer values('77777777777777777','刘洋','12378952016');
insert into Customer values('888888888888888888','李桂英','19618346059');
insert into Customer values('999999999999999999','王涛','19204562817');
select * from Publisher;
```

Data Output | Explain | Messages | Notifications

	publisher_id [PK] character (11)	publisher_name character varying (20)	publisher_phone character varying (15)
1	PUB00000001	人民文学出版社	010-11111111
2	PUB00000002	上海译文出版社	010-22222222
3	PUB00000003	中华书局出版社	010-33333333

B．更新操作

```
update Publisher set Publisher_Phone = '010-55555555' where Publisher_ID = 'PUB00000004';
select * from Publisher;
```

	publisher_id [PK] character (11)	publisher_name character varying (20)	publisher_phone character varying (15)
1	PUB00000001	人民文学出版社	010-11111111
2	PUB00000002	上海译文出版社	010-22222222
3	PUB00000003	中华书局出版社	010-33333333
4	PUB00000004	电子科技大学出版社	010-55555555

C．删除操作

```
delete from Publisher where Publisher_ID = 'PUB00000004';
select * from Publisher;
```

	publisher_id [PK] character (11)	publisher_name character varying (20)	publisher_phone character varying (15)
1	PUB00000001	人民文学出版社	010-11111111
2	PUB00000002	上海译文出版社	010-22222222
3	PUB00000003	中华书局出版社	010-33333333

（4）编写存储过程 Pro_CurrentSale，实现当日图书销售量及销售金额汇总统计。

```
create or replace function Pro_CurrentSale(out amount int4, out allmoney money) as $count$
    begin
        select sum(Sale_Number) into amount from Sale where Sale_Date = '2022-05-05';
        select sum(Sale_Amount) into allmoney from Sale where Sale_Date = '2022-05-05';
    end;
$count$ LANGUAGE plpgsql;
```

（5）编写过程语句块，实现对存储过程 Pro_CurrentSale 的调用，并输出统计结果。

```
select * from Pro_CurrentSale();
```

（6）编写图书销售表 Insert 触发器 Tri_InsertSale，实现图书库存数据同步修改处理。

编写图书销售表 Insert 触发器 Tri_InsertSale，实现在 Sale 表数据插入时，级联操作 Bookstock 表，将图书的库存流水进行记录，同时级联更新 Book 表中对应图书的库存数据。

```
create or replace function InsertSale()
    returns trigger as $$
    begin
        insert into Bookstock values(new.Sale_ID,new.Sale_Date,'出库',new.Sale_Number,new.Book_ISBN);
        update Book set Book_Stock = Book_Stock - new.Sale_Number where Book.Book_ISBN = new.Book_ISBN;
        return new;
    end;
$$ language plpgsql;
```

创建触发器

```
1  create trigger Tri_InsertSale after insert on Sale
2  for each row execute procedure InsertSale();
```

（7）对图书销售表 Insert 触发器 Tri_InsertSale 程序进行功能验证。

为 Sale 表准备样本数据，将样本数据插入到表中之后查看 Bookstock 表是否有对应的更新，并对比插入数据前后 Bookstock 表中对应数据的修改情况。

```
insert into Sale
values('SA00001','2022-05-02','8','464','9787121021961','777777777777777777');
insert into Sale
values('SA00002','2022-05-03','5','1640','9787559202253','888888888888888888');
insert into Sale
values('SA00003','2022-05-05','10','680','9787532786831','999999999999999999');
insert into Sale
values('SA00004','2022-05-05','22','924','9787555911425','999999999999999999');
insert into Sale
values('SA00005','2022-05-05','33','2244','9787532786831','888888888888888888');
insert into Sale
values('SA00006','2022-05-05','2','656','9787559202253','777777777777777777');
select * from Bookstock;
select * from Book;
```

再分别查看 Bookstock 和 Book 表

	stock_id [PK] character varying (10)	stock_date date	stock_operation character (4)	stock_change integer	book_isbn character (13)
1	SA00001	2022-05-02	出库	8	9787121021961
2	SA00002	2022-05-03	出库	5	9787559202253
3	SA00003	2022-05-05	出库	10	9787532786831
4	SA00004	2022-05-05	出库	22	9787555911425
5	SA00005	2022-05-05	出库	33	9787532786831
6	SA00006	2022-05-05	出库	2	9787559202253

	book_isbn [PK] character (13)	book_name character varying (50)	book_pubdate date	book_price money	book_stock integer	author_id character (18)	publisher_id character (11)
1	9787121021961	操作系统:精髓与设计原理	2006-02-01	$58.00	12	111111111111111111	PUB00000001
2	9787555911425	字母表谜案	2021-05-07	$42.00	128	333333333333333333	PUB00000003
3	9787532786831	克拉拉与太阳	2021-03-03	$68.00	55	222222222222222222	PUB00000002
4	9787559202253	西方建筑史	2019-12-02	$328.00	23	222222222222222222	PUB00000002

八、实验数据及结果分析

Bookstock 表中本来没有数据，在 Sale 表插入数据之后，将库存数据进行记录，记录数据正确，同时发现 Book 表中库存量都发生了变化，

分别减少了与销售表中记录的销售件数相应值的大小，更新数据正确，可以证明触发器的功能都实现了，触发器功能正确。

九、总结及心得体会

通过这次学习，发现自己的相关 SQL 语句的标准模式掌握的不够熟练，还需要查找相关资料才能完成，通过这次练习，提高了相关语句的熟练度，所以勤学多练才是掌握一门技能的好方法。

电 子 科 技 大 学

实 验 报 告

一、实验二：图书销售管理系统数据库安全管理

二、实验室名称：信软学院楼西 400　　实验时间：2023-05-11，2023-05-18

三、实验目的

了解该 DBMS 系统对数据库管理的内容与方法，特别是理解数据库安全机制和作用，以及 PostgreSQL 数据库角色管理、用户管理、权限管理的基本方法，培养数据库管理能力。在图书销售管理系统数据库中，创建必要的角色和用户，并完成上述角色与用户的权限管理。

四、实验原理

设计数据存取权限控制模型，对各角色进行不同权限的赋予，保证数据库数据的安全性。使用 SQL 语句进行角色、用户的创建、对角色进行权限赋予、对用户分派角色。

五、实验内容

使用 pgAdmin4 数据库管理工具对图书销售管理系统数据库进行数据库安全管理，具体实验内容如下:

1.针对图书销售管理系统数据库应用需求，设计数据存取权限控制模型。

2.在数据库中，创建客户（R_Customer）、商家（R_Seller）角色。

3.在数据库中，根据业务规则为客户（R_Customer）、商家（R_Seller）角色赋予数据库对象权限。

4.在数据库中，分别创建客户用户 U_Customer、商家用户 U_Seller。

5.分别为客户用户 U_Customer、商家用户 U_Seller 分派客户（R_Customer）、商家（R_Seller）角色。

6.分别以客户用户 U_Customer、商家用户 U_Seller 身份访问图书销售管理数据库，验

证所实现数据存取权限控制模型的正确性。

在实验计算机上，利用 pgAdmin4 数据库管理工具及 SQL 语句，完成图书销售管理系统数据库安全管理，同时记录实验过程的步骤、操作、运行结果界面等数据，为撰写实验报告提供素材。

六、实验设备及环境

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows10 / Windows 11

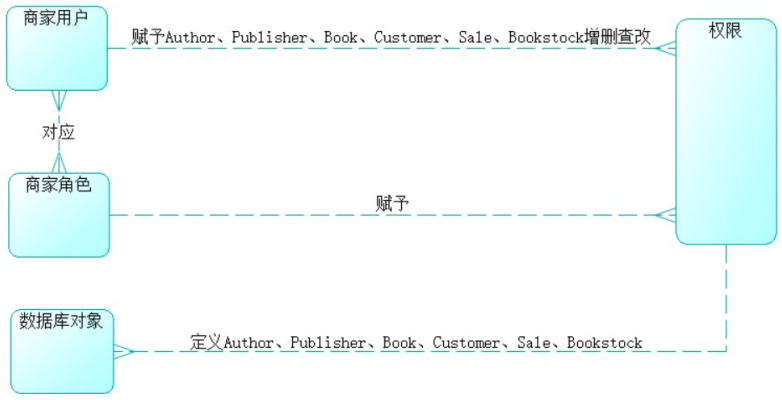
管理工具： pgAdmin4

DBMS 系统： PostgreSQL 15.1

七、实验步骤

（1）针对图书销售管理系统数据库，设计数据存取权限控制模型。

根据业务逻辑及实际需要为商家、客户、系统管理员分配在每个数据库表上的操作权限。



（2）在数据库中，创建客户（R_Customer）、商家（R_Seller）角色。

采用 SQL 语句执行方式，创建客户 R_Customer、商家 R_Seller 角色。

```
create role R_Customer with
login
nosuperuser
nocreatedb
nocreaterole
noinherit
noreplication
connection limit -1
password '123456';
create role R_Seller with
login
nosuperuser
nocreatedb
nocreaterole
noinherit
noreplication
connection limit -1
password '123456';
```

（3）在数据库中，根据业务规则为客户（R_Customer）、商家（R_Seller）角色赋予数据库对象权限。

利用 Grant 关键词对客户 R_Customer、商家 R_Seller 角色，参考设计的数据存取权限控制模型，赋予对应的所定义的数据库对象权限。

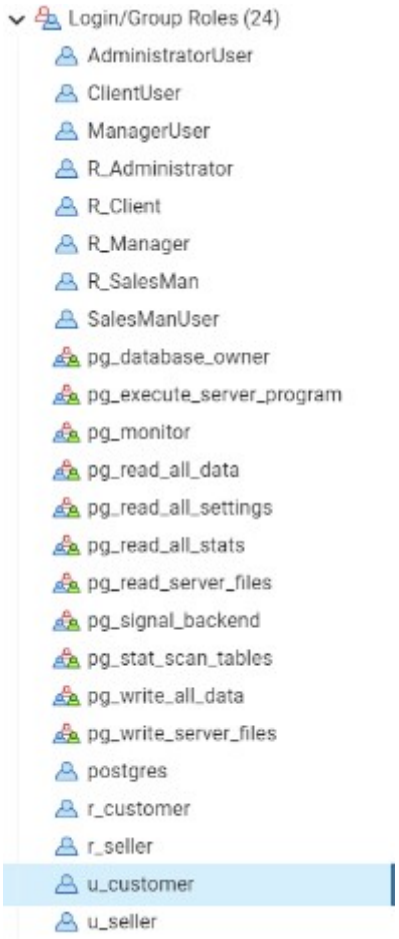
```
grant select on Author to R_Customer;
grant select on Publisher to R_Customer;
grant select on Book to R_Customer;
grant select,insert,update,delete on Author to R_Seller;
grant select,insert,update,delete on Publisher to R_Seller;
grant select,insert,update,delete on Book to R_Seller;
grant select,insert,update,delete on Customer to R_Seller;
grant select,insert,update,delete on Sale to R_Seller;
grant select on Bookstock to R_Seller;
```

（4）在数据库中，分别创建客户用户 U_Customer、商家用户 U_Seller。

（5）分别为客户用户 U_Customer、商家用户 U_Seller 分派客户（R_Client）、商家

（R_Seller）角色。

```
create user U_Customer with
login
connection limit -1
in role R_Customer
password '123456';
create user U_Seller with
login
connection limit -1
in role R_Seller
password '123456';
```



（6）分别以客户用户 U_Customer、商家用户 U_Seller 身份访问图书销售管理数据库，验证所实现数据存取权限控制模型的正确性。

分别以客户用户 U_Customer、商家用户 U_Seller 身份访问图书销售管理数据库，并分别以这两个用户对各数据库表进行操作，以验证是否正确分配了两用户不同的角色权限。

以客户用户身份登陆数据库：

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: booksale
Port [5432]:
Username [postgres]: u_customer
psql (14.2)
输入 "help" 来获取帮助信息.

booksale=>
```

A. 查看 Book 表：

```
SQL Shell (psql)

booksale=> select * from Book;
   book_isbn    |        book_name        | book_pubdate | book_price | book_stock |     author_id     | publisher_i
----------------+-------------------------+--------------+------------+------------+-------------------+-----------
 9787121021961 | 操作系统:精髓与设计原理 | 2006-02-01   |     $58.00 |         12 | 1111111111111111  | PUB0000000
 9787555911425 | 字母表谜案              | 2021-05-07   |     $42.00 |        128 | 333333333333333   | PUB0000000
 9787532786831 | 克拉拉与太阳            | 2021-03-03   |     $68.00 |         55 | 2222222222222222  | PUB0000000
 9787559202253 | 西方建筑史              | 2019-12-02   |    $328.00 |         23 | 2222222222222222  | PUB0000000
-- More --
```

删除数据：

```
booksale=> delete from Book where Book_ISBN = '9787121021961';
ERROR:  permission denied for table book
booksale=>
```

B. 查看 Author 表

```
booksale=> select * from Author;
     author_id      | author_name | author_gender
--------------------+-------------+---------------
 1111111111111111   | 张伟        | 男
 2222222222222222   | 王秀英      | 女
 333333333333333    | 李军        | 男
(3 行记录)
```

插入数据：

```
booksale=> insert into Author values('4444444444444444','大锤','男');
ERROR:  permission denied for table author
booksale=>
```

查看 sale 表；

```
booksale=> select * from Sale;
ERROR:  permission denied for table sale
booksale=>
```

以商家用户身份登陆数据库：

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: booksale
Port [5432]:
Username [postgres]: u_seller
psql (14.2)
输入 "help" 来获取帮助信息.
```

A.　　查看 Customer 表

```
booksale=> select * from Customer;
    customer_id     | customer_name | customer_phone
--------------------+---------------+----------------
 7777777777777777   | 刘洋          | 12378952016
 8888888888888888   | 李桂英        | 19618346059
 9999999999999999   | 王涛          | 19204562817
(3 行记录)
```

插入数据

```
booksale=> insert into Author values('4444444444444444','大锤'
INSERT 0 1
booksale=> select * from Author;
     author_id       | author_name | author_gender
---------------------+-------------+---------------
 1111111111111111    | 张伟        | 男
 2222222222222222    | 王秀英      | 女
 3333333333333333    | 李军        | 男
 4444444444444444    | 大锤        | 男
(4 行记录)
```

从 Bookstock 表中删除一条数据

```
booksale=> delete from Bookstock where Stock_ID = 'SA0
ERROR:  permission denied for table bookstock
booksale=>
```

更新 Sale 表中的一条数据

```
booksale=> update Sale set Sale_Date = '2022-05-01' where Sale_ID = 'SA00001';
UPDATE 1
booksale=> select * from Sale;
 sale_id | sale_date  | sale_number | sale_amount |   book_isbn   |    customer_id
---------+------------+-------------+-------------+---------------+--------------------
 SA00002 | 2022-05-03 |           5 | $1,640.00   | 9787559202253 | 888888888888888888
 SA00003 | 2022-05-05 |          10 |   $680.00   | 9787532786831 | 999999999999999999
 SA00004 | 2022-05-05 |          22 |   $924.00   | 9787555911425 | 999999999999999999
 SA00005 | 2022-05-05 |          33 | $2,244.00   | 9787532786831 | 888888888888888888
 SA00006 | 2022-05-05 |           2 |   $656.00   | 9787559202253 | 777777777777777777
 SA00001 | 2022-05-01 |           8 |   $464.00   | 9787121021961 | 777777777777777777
```

八、实验数据及结果分析

客户：

客户用户 U_Customer 可查看 Book 表中的数据。

在删除 Book 表中的数据时，结果显示失败，客户用户不能完成删除操作。

客户用户可以查看 Author 表中的数据。

在向 Author 表中插入一条数据时，结果显示失败，客户用户不能完成插入操作。

在查看 Sale 表中的数据时，结果显示失败，客户用户不能完成查看操作

客户用户对数据库表的权限的测试都正确，即创建的客户用户赋予权限的操作是正确的。

商家：

商家用户 U_Seller 可查看 Customer 表中的数据。

通过与客户用户步骤中查询的作家信息表进行对比，Author 表中的数据新增加了一条，商家用户具有对 Author 表的插入权限。

在从 Bookstock 表中删除一条数据时，结果显示失败，商家用户不能完成删除操作。

通过对比，Sale 表中的数据发生了更新，商家用户具有对 Sale 表的更新权限。

商家用户对数据库表的权限的测试都正确，即创建的商家用户赋予权限的操作是正确的。

九、总结及心得体会

角色和用户创建后，使用用户登陆时只能进行创建时赋予的操作权限，对于未授予权限的操作，数据库系统将会报错。

在对数据库安全设计时，考虑使用者的权限管理。