School of Data and Computer Science, Sun Yat-sen University

功能简介和数据库部分实现

18308045 Zhengyang Gu

January 15, 2021

Contents

1 功能简介

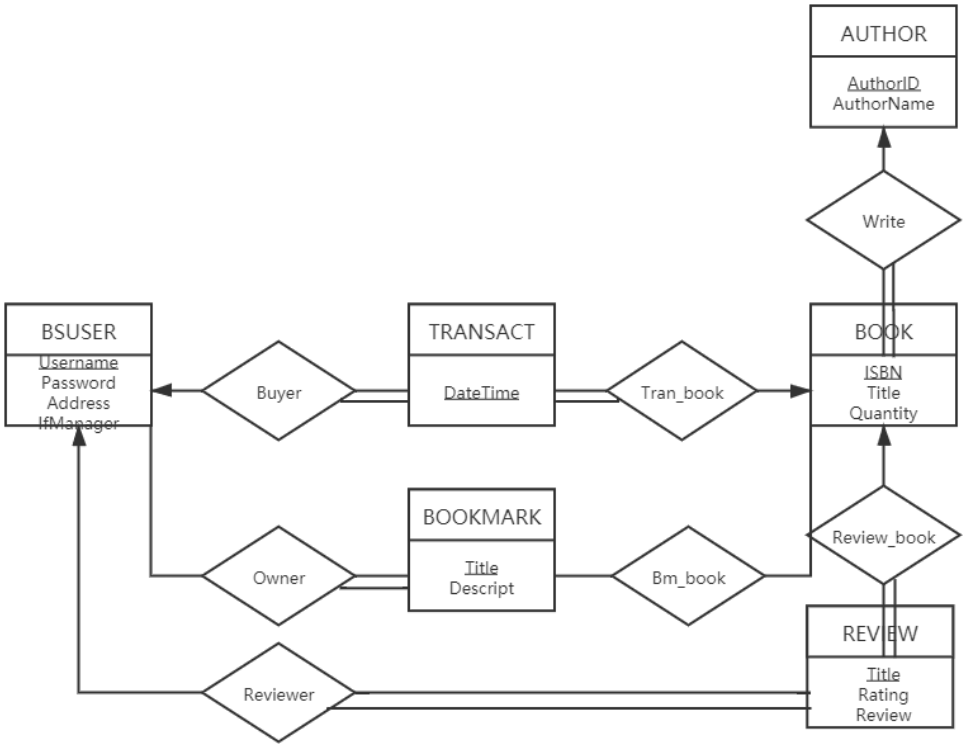
总体来说是图书售卖系统。具体来说，包含以下几个功能：

1. 创建管理员用户，管理员用户可以管理数据库。
2. 创建普通用户。
3. 如果图书还有余量，则可购买此图书。
4. 如果用户已经购买过图书，则可评价此图书。
5. 创建书单其中有标题和描述，所有用户都可以查看，并按照书单买书（好书推荐）。

2 具体实现

2.1 模式设计

2.1.1 ER 图



2.1.2 转化为模式

一对多的，让多的实体包含一的实体的主码；多对多的，增加一个表包含两个实体。另外值得注意的是对书的数目即 quantity 设了一个约束大于等于 0。

```
1 CREATE TABLE BSUSER
```

```
(
    Username VARCHAR(15) NOT NULL,
    Password VARCHAR(15) NOT NULL,
    Address VARCHAR(20) NOT NULL,
    IfManager INT,
    PRIMARY KEY(Username)
);

CREATE TABLE AUTHOR
(
    AuthorID VARCHAR(15) NOT NULL,
    AuthorName VARCHAR(15) NOT NULL,
    PRIMARY KEY(AuthorID)
);

CREATE TABLE BOOK
(
    ISBN CHAR(14) NOT NULL,
    Title VARCHAR(50) NOT NULL,
    AuthorID VARCHAR(15) NOT NULL,
    Quantity INT NOT NULL,
    CHECK (Quantity >= 0),
    PRIMARY KEY(ISBN),
    FOREIGN KEY(AuthorID) references AUTHOR(AuthorID)
);

CREATE TABLE REVIEW
(
    Title VARCHAR(15),
    Reviewer VARCHAR(15) NOT NULL,
    BookID CHAR(14) NOT NULL,
    Rating INT NOT NULL,
    Review TEXT,
    CHECK (Rating >= 1 OR Rating <= 5),
    PRIMARY KEY(Title),
    FOREIGN KEY(Reviewer) references BSUSER(Username),
    FOREIGN KEY(BookID) references BOOK(ISBN)
);

CREATE TABLE TRANSACT
(
```

```
    DateTime CHAR(23) NOT NULL,
    Buyer VARCHAR(15) NOT NULL,
    BookID CHAR(14) NOT NULL,
    PRIMARY KEY(DateTime),
    FOREIGN KEY(Buyer) references BSUSER(Username),
    FOREIGN KEY(BookID) references BOOK(ISBN)
);

create table BOOKMARK
(
    Title VARCHAR(15),
    Descript TEXT,
    primary key(Title)
)

create table BOOKMARK_BOOK
(
    Title VARCHAR(15),
    Owner VARCHAR(15),
    BookID CHAR(14),
    Owned Bit,
    primary key(Title, BookID, Owned),
    foreign key(Owner) references BSUSER(Username),
    foreign key(BookID) references BOOK(ISBN)
)
```

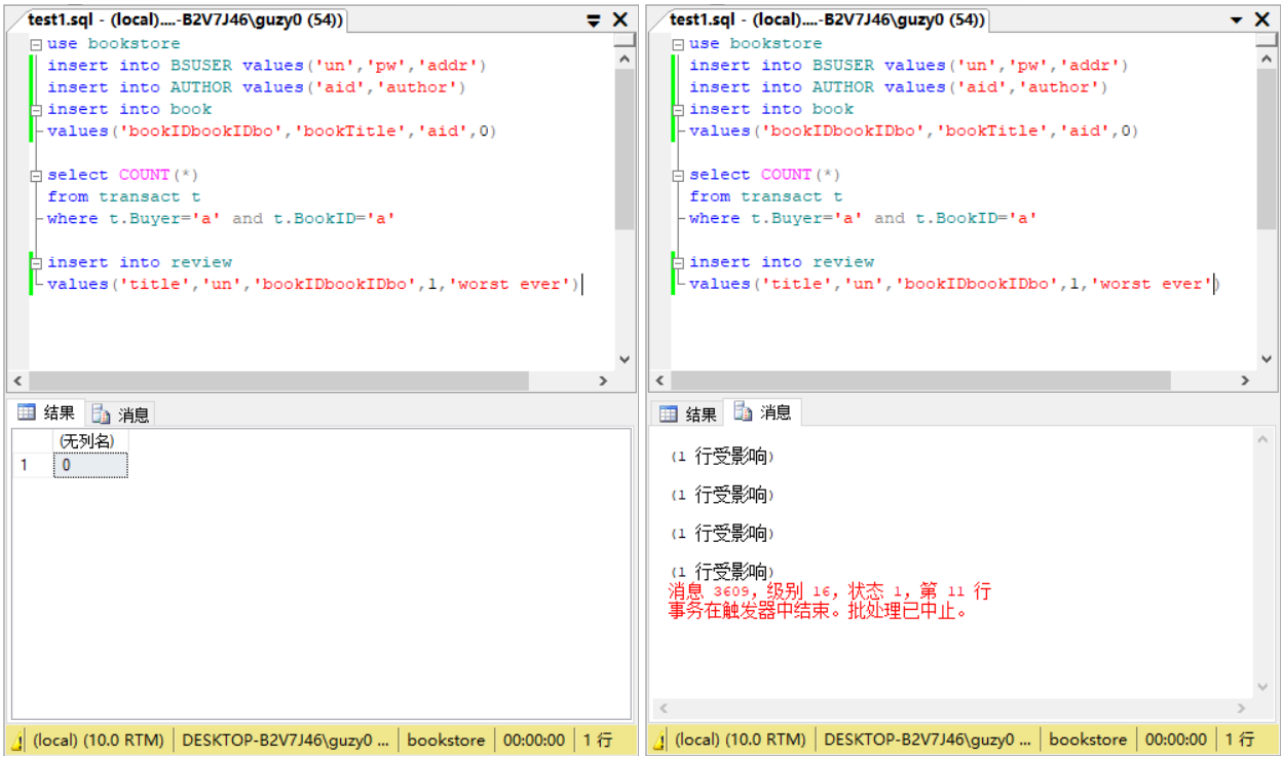
2.2 触发器

2.2.1 评论

要求评论书前，需要已经购买过此书：

```
create trigger t_reivew on REVIEW for insert as
if (select count(*)
    from TRANSACT t, inserted i
    where t.Buyer=i.Reviewer and t.BookID=i.BookID)=0
begin
rollback TRANSACTION
end
```

测试恶意评论结果如下：



可以看出，用户'un' 对书'bookIDbookIDbo' 的交易为 0，触发器致使'un' 评论失败。

2.2.2 交易

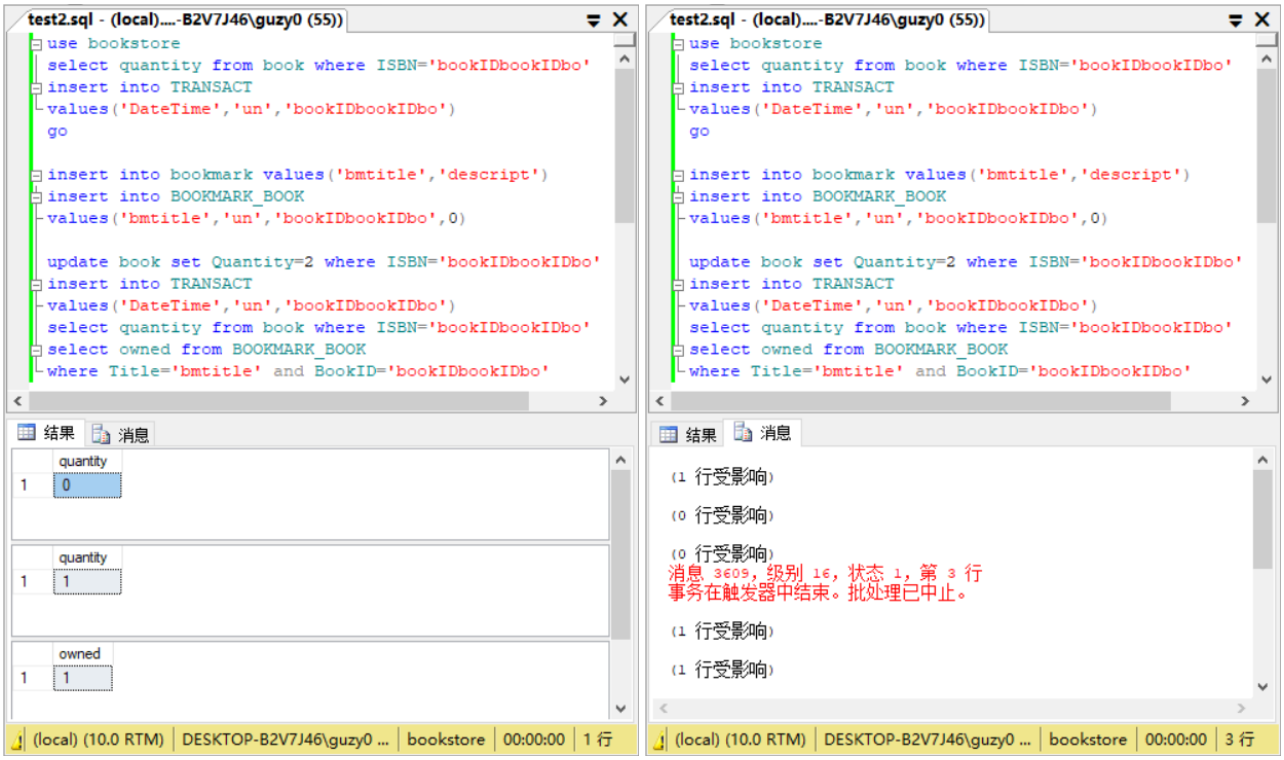
购买书的时候，有以下几点：

1. 书的数量要大于 0。
2. 买完后书的数量减 1。
3. 书单上标记已购此书。

实现如下：

```
create trigger t_transact on TRANSACT for insert as
if (select quantity from book, inserted where ISBN=inserted.BookID) <= 0
begin
rollback transaction
end
update book set quantity=quantity-1 where ISBN=(select BookID from inserted)
update bookmark_book set Owned=1 where BookID=(select BookID from inserted)
and Owner=(select Buyer from inserted)
```

测试如下：



可以看出，书的数量是 0，触发器致使'un' 交易失败。设置此书于书单'bmtitle' 中且标记为无此书，更新书的数量为 2。之后再插入交易，发现成功，且交易后书的数量变为 1，书单中标记为拥有此书。

2.3 可能出现的不一致问题

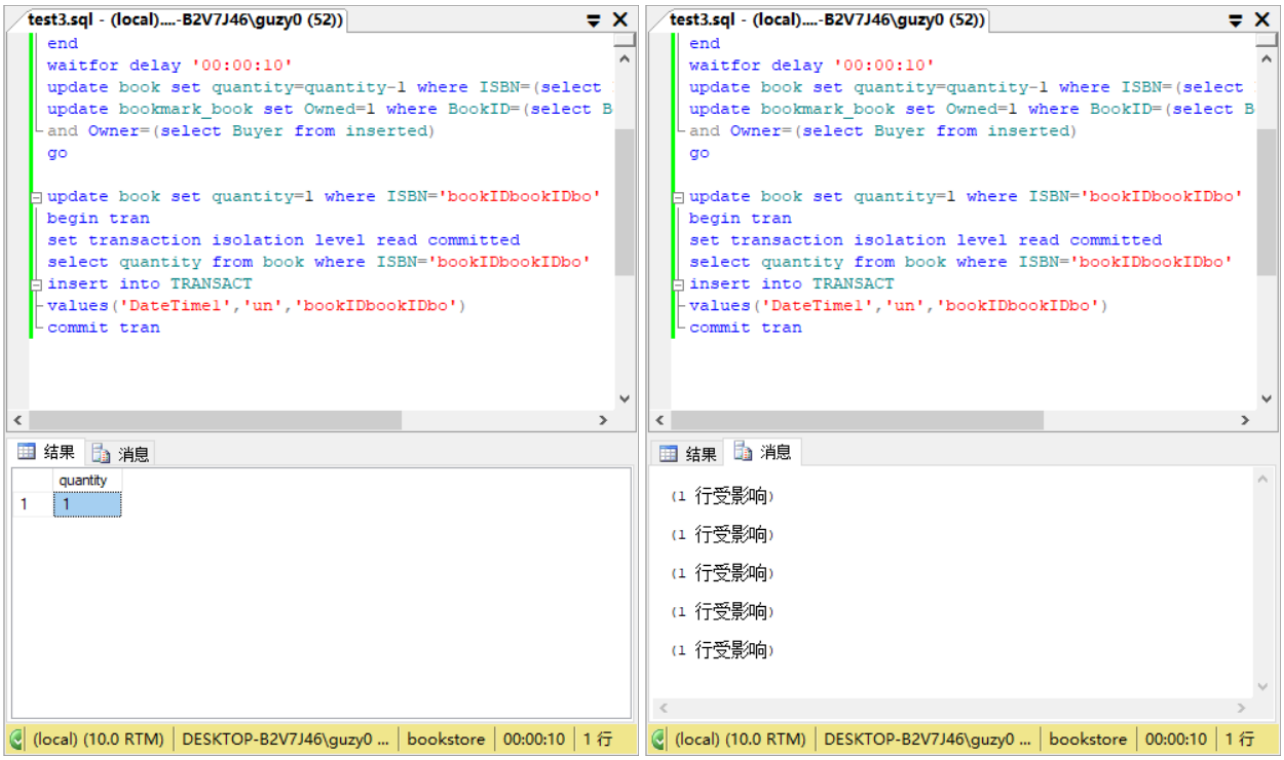
由于触发器并不是原子操作，在多用户多事务并发的环境下插入交易时可能出现不一致的情况。考虑此时某本书只有一个，事务 A 在刚刚判断完 $quantity > 0$ 而未修改 quantity 并提交，此时事务 B 也判断会认为 $quantity > 0$，最终结果是一本书插入了两条交易。下面在触发器判断后和插入前加一个 10 秒延迟来模拟这种现象。test3.sql 代码截图不全，因而在此放出：

```
use bookstore
drop trigger t_transact
go
create trigger t_transact on TRANSACT for insert as
if (select quantity from book, inserted where ISBN=inserted.BookID) <= 0
begin
rollback transaction
end
waitfor delay '00:00:10'
update book set quantity=quantity-1 where ISBN=(select BookID from inserted)
update bookmark_book set Owned=1 where BookID=(select BookID from inserted)
```

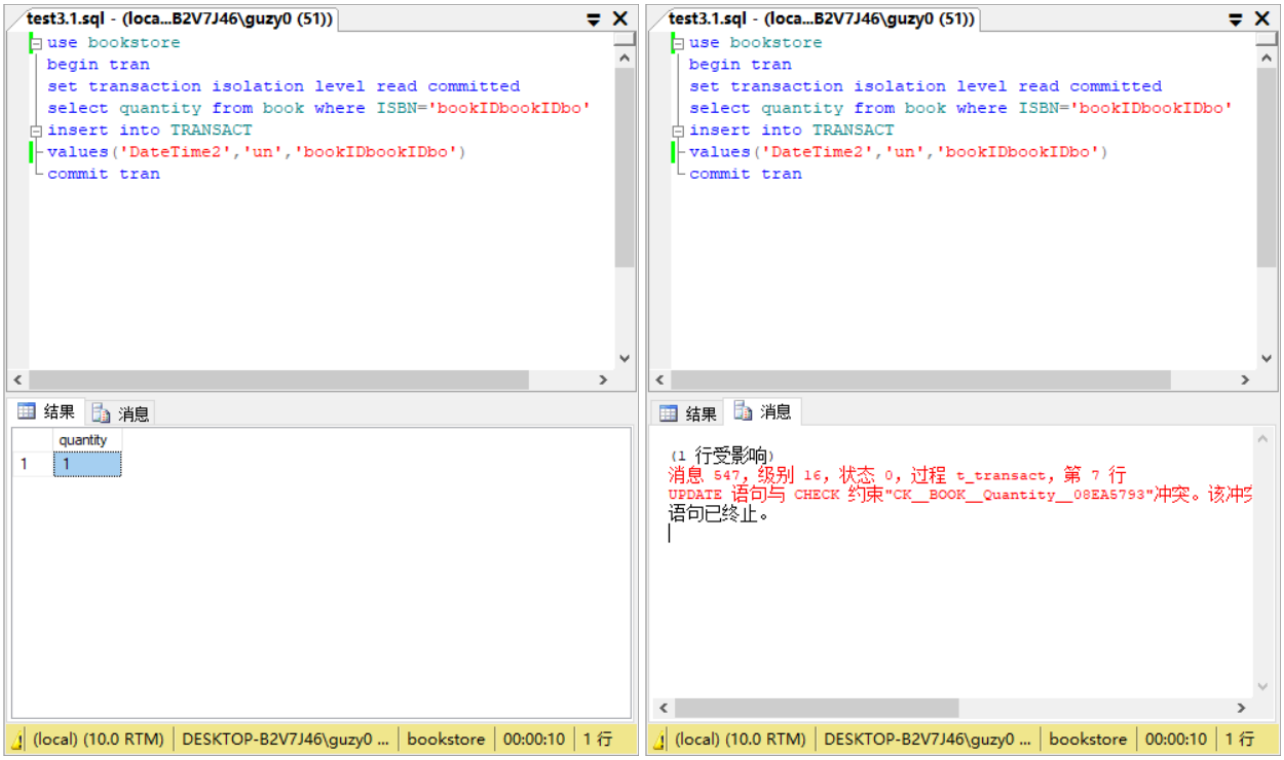
```
and Owner=(select Buyer from inserted)
go

update book set quantity=1 where ISBN='bookIDbookIDbo'
begin tran
set transaction isolation level read committed
select quantity from book where ISBN='bookIDbookIDbo'
insert into TRANSACT
values('DateTime1','un','bookIDbookIDbo')
commit tran
```

其主要功能就是在 t_transact 的判断语句和插入语句之间加一个延迟，将书 quantity 设为 1 并执行事务插入一条交易。运行结果如下：



上面事务执行的同时，执行如下事务：



可以发现，两个事务都读到书的 quantity 为 1，因此两个事务都没被回滚，执行了插入语句。但是由于 quantity 的约束要求大于等于 0，其中一个事务的 trigger 更新会失败，导致 trigger 失败进而导致插入语句失败。因此 update 语句和对 quantity 的约束已经防止了此类不一致的发生，因而提前判断 quantity 也是没必要的，因而改成：

```
create trigger t_transact on TRANSACT for insert as
update book set quantity=quantity-1 where ISBN=(select BookID from inserted)
update bookmark_book set Owned=1 where BookID=(select BookID from inserted)
and Owner=(select Buyer from inserted)
```