MYSQL实验报告

Create database booksalesmanagement character set utf8;

//创建用户列表

create table UserTable(

username varchar(100) NOT NULL,

password varchar(100) ,

rank ENUM('职工','供货商','管理员'),

Primary key(username)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table BookStorage(

ISBN char(13) NOT NULL,

bookname varchar(300) NOT NULL,

author varchar(200),

bookconcern varchar(200),

price float(8) NOT NULL,

storecount int(4) NOT NULL,

cost float(8) NOT NULL,

Primary Key(ISBN)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table supplier(

sno int(8) NOT NULL AUTO\_INCREMENT,

supplytime timestamp NOT NULL,

ISBN char(13) NOT NULL,

bookname varchar(300) NOT NULL,

author varchar(200) ,

supplier varchar(300) NOT NULL,

supplycost float(8) NOT NULL,

supplycount int(4) NOT NULL,

Primary Key(sno,supplytime)

)Engine=InnoDB DEFAULT CHARSET=utf8;

create table BookStock(

sno int(8) NOT NULL AUTO\_INCREMENT,

stocktime timestamp NOT NULL,

ISBN char(13) NOT NULL,

bookname varchar(300),

author varchar(200),

bookconcern varchar(200),

cost float(8) NOT NULL,

stockcount int(4) NOT NULL,

Primary Key(sno)

)Engine=InnoDB DEFAULT CHARSET=utf8;

create table BooKReport(

ISBN char(13) NOT NULL,

bookname varchar(300) NOT NULL,

author varchar(200),

bookconcern varchar(200),

price float(8) NOT NULL,

soldcount int(4) NOT NULL,

soldsum float(8) NOT NULL,

Primary Key(ISBN)

)Engine=InnoDB DEFAULT CHARSET=utf8;

create table booksold(

sno int(8) NOT NULL AUTO\_INCREMENT,

soldtime timestamp NOT NULL,

ISBN char(13) NOT NULL,

bookname varchar(300) NOT NULL,

author varchar(200),

bookconcern varchar(200),

price float(8) NOT NULL,

soldcount int(4) NOT NULL,

Primary Key(sno)

)Engine=InnoDB DEFAULT CHARSET=utf8;

create table bookrefund(

sno int(8) NOT NULL AUTO\_INCREMENT,

refundtime timestamp NOT NULL,

ISBN char(13) NOT NULL,

bookname varchar(300) NOT NULL,

author varchar(200),

bookconcern varchar(200),

price float(8) NOT NULL,

refundcount int(4) NOT NULL,

Primary Key(sno)

)Engine=InnoDB DEFAULT CHARSET=utf8;

delimiter //

create trigger update\_after\_refund after insert on bookrefund

for each row

begin

update bookstorage set storecount=storecount+new.refundcount where ISBN=new.ISBN;

update bookreport set soldcount=soldcount-new.refundcount where ISBN=new.ISBN;

update bookreport set soldsum=soldcount\*price;

end//

delimiter //

create procedure insert\_into\_refund(refundtime timestamp,ISBN char(13),bookname varchar(50),author varchar(20), bookconcern varchar(20),price float(8),refundcount int(4))

Begin

if(refundcount <= (select bookreport.soldcount from bookreport where bookreport.ISBN=ISBN)) then

insert into bookrefund (refundtime, ISBN, bookname, author, bookconcern, price, refundcount) values(refundtime,ISBN,bookname,author,bookconcern,price,refundcount);

end if;

End;//

delimiter //

create trigger insert\_after\_stock after insert on bookstock

for each row

begin

if((select count(\*) from bookstorage where bookstorage.ISBN = new.ISBN) <> 0) then

update bookstorage set storecount=storecount+new.stockcount,cost=new.cost where ISBN=new.ISBN;

else

insert into bookstorage values(new.ISBN, new.bookname, new.author, new.bookconcern, new.cost\*1.12,new.stockcount,new.cost);

end if;

end//

delimiter //

create trigger insert\_after\_vend after insert on booksold

for each row

begin

update bookstorage set storecount=storecount-new.soldcount where ISBN=new.ISBN;

if((select count(\*) from bookreport where bookreport.ISBN = new.ISBN) <> 0) then

update bookreport set soldcount=soldcount+new.soldcount,soldsum=soldcount\*price where ISBN=new.ISBN;

else

insert into bookreport values(new.ISBN, new.bookname, new.author, new.bookconcern, new.price,new.soldcount,new.price\*new.soldcount);

end if;

end//