**注意:代码下附上图示**

**一、创建数据库**

/\*创建用户\*/

create user rxy identified by 123456;

grant dba to rxy;

**二、创建表**

/\*创建tb\_user表\*/

create sequence SEQ\_userID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_user(

userID number,

name varchar2(20)unique not null,

password varchar2(30)not null,

email varchar2(30),

provinceID number,

cityID number,

telephone char(11),

sex char(3)default '男',

age number,

birthday date,

money number default 0,

photo varchar2(500),

dsp varchar2(200),

qq varchar2(15),

msn varchar2(50),

loveBook varchar2(200),

loveMusic varchar2(200),

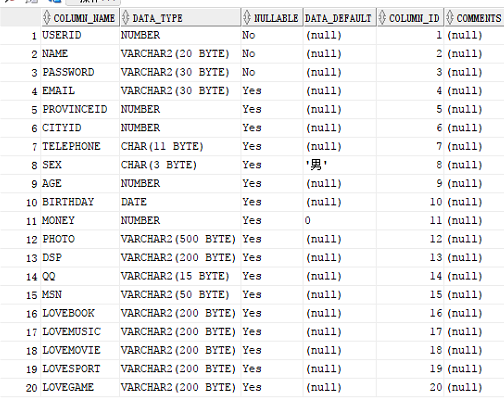
loveMovie varchar2(200),

loveSport varchar2(200),

loveGame varchar2(200),

primary key(userID)

);



/\*创建tb\_product表\*/

create sequence SEQ\_productID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_product(

productID number not null,

userID number not null,

productName varchar2(50)not null,

typeID number not null,

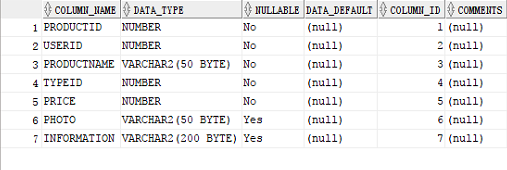
price number not null,

photo varchar2(50),

information varchar2(200),

primary key(productID)

);



/\*创建tb\_productType表\*/

create sequence SEQ\_productTypeID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_productType(

productTypeID number not null,

productTypeName varchar2(30)not null,

primary key(productType)

);



/\*创建tb\_order表\*/

create sequence SEQ\_orderID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_order(

orderID number not null,

toID number not null,

colorID number not null,

productID number not null,

address varchar2(100) not null,

telephone char(11)not null,

orderState number not null,

count number not null,

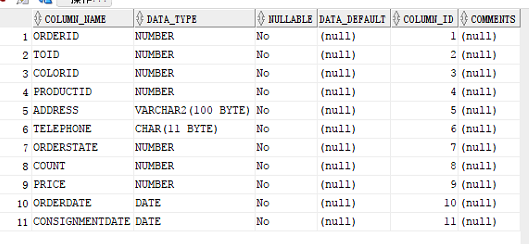
price number not null,

orderDate date not null,

consignmentDate date not null,

primary key(orderID)

);



(orderState,orderID)

/\*创建tb\_userAccount表\*/

create sequence SEQ\_userAccountID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_userAccount(

userAccountID number not null,

userID number not null,

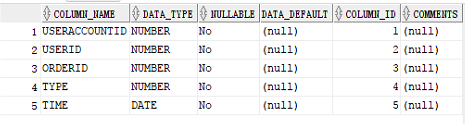
orderID number not null,

type number not null,

time date not null,

primary key(userAccountID)

);



/\*创建tb\_review表\*/

create sequence SEQ\_reviewID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_review(

reviewID number not null,

review varchar2(200)not null,

isReply number default 0 not null,

isDel number default 0 not null,

senderID number not null,

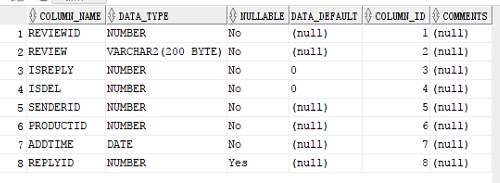
productID number not null,

addTime date not null,

replyID number,

primary key(reviewID)

);



/\*创建tb\_province表\*/

create sequence SEQ\_provinceID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

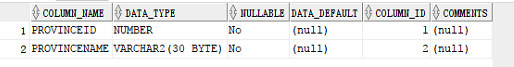
create table tb\_province(

provinceID number not null,

provinceName varchar2(30)not null,

primary key(provinceID)

);



/\*创建tb\_city表\*/

create sequence SEQ\_cityID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_city(

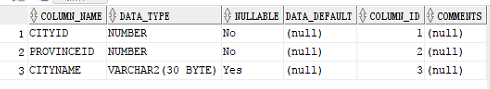
cityID number not null,

provinceID number not null,

cityName varchar2(30),

primary key(cityID)

);



/\*创建tb\_productSize表\*/

create sequence SEQ\_productSizeID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_productSize(

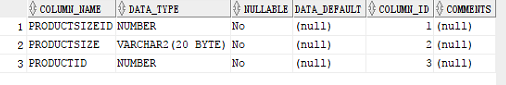
productSizeID number not null,

productSize varchar2(20)not null,

productID number not null,

primary key(productSizeID)

);



/\*创建tb\_productColor表\*/

create sequence SEQ\_productColorID

minvalue 1

maxvalue 99999999

start with 1

increment by 1

nocache;

create table tb\_productColor(

productColorID number not null,

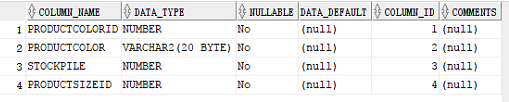
productColor varchar2(20)not null,

stockpile number not null,

productSizeID number not null,

primary key(productColorID)

);



**三、添加约束**

/\*给每个表添加约束\*/

alter table tb\_user add constraint cityID\_limit foreign key(cityID) references tb\_city (cityID);

alter table tb\_product add constraint userID\_limit foreign key(userID) references tb\_user (userID);

alter table tb\_product add constraint typeID\_limit foreign key(typeID) references tb\_productType (productTypeID);

alter table tb\_order add constraint toID\_limit foreign key(toID) references tb\_user (userID);

alter table tb\_order add constraint colorID\_limit foreign key(colorID) references tb\_productColor (productColorID);

alter table tb\_order add constraint productID\_limit foreign key(productID) references tb\_product (productID);

alter table tb\_userAccount add constraint userID2\_limit foreign key(userID) references tb\_user (userID);

alter table tb\_userAccount add constraint orderID\_limit foreign key(orderID) references tb\_order (orderID);

alter table tb\_review add constraint senderID\_limit foreign key(senderID) references tb\_user (userID);

alter table tb\_review add constraint productID2\_limit foreign key(productID) references tb\_product (productID);

alter table tb\_city add constraint provinceID\_limit foreign key(provinceID) references tb\_province (provinceID);

alter table tb\_productSize add constraint productID3\_limit foreign key(productID) references tb\_product (productID);

alter table tb\_productColor add constraint productSizeID\_limit foreign key(productSizeID) references tb\_productSize (productSizeID);

**四、基础数据(商品类型+省份+城市)添加**

/\*基础数据(商品类型+省份+城市)添加\*/

/\*tb\_productType表添加商品类型\*/

insert into tb\_productType values(seq\_producttypeID.nextval,'数码相机');

insert into tb\_productType values(seq\_producttypeID.nextval,'笔记本电脑');

insert into tb\_productType values(seq\_producttypeID.nextval,'手机');



/\*tb\_province表添加省份名称\*/

insert into tb\_province values(seq\_provinceID.nextval,'山东');

insert into tb\_province values(seq\_provinceID.nextval,'北京');

insert into tb\_province values(seq\_provinceID.nextval,'河北');

insert into tb\_province values(seq\_provinceID.nextval,'内蒙古');

insert into tb\_province values(seq\_provinceID.nextval,'浙江');

insert into tb\_province values(seq\_provinceID.nextval,'上海');



/\*tb\_city表给山东省添加城市名称\*/

insert into tb\_city values(seq\_cityID.nextval,1,'青岛');

insert into tb\_city values(seq\_cityID.nextval,1,'菏泽');

insert into tb\_city values(seq\_cityID.nextval,1,'济南');

insert into tb\_city values(seq\_cityID.nextval,2,'北京');

insert into tb\_city values(seq\_cityID.nextval,3,'保定');

insert into tb\_city values(seq\_cityID.nextval,4,'呼伦贝尔');

insert into tb\_city values(seq\_cityID.nextval,5,'武汉');

insert into tb\_city values(seq\_cityID.nextval,6,'上海');



**五、用户数据(注册+信息完善)添加**

/\*用户数据(注册+信息完善)添加\*/

/\*tb\_user表添加用户信息\*/

insert into tb\_user values(seq\_userID.nextval, '任晓宇','123','unmei@baka9.com',1,1,'15063011597','男',20,to\_date('1996.12.19','yyyy-mm-dd'),50000,null,null,'496071360',null,null,null,null,null,null);

insert into tb\_user values(seq\_userID.nextval, '单良' ,'222','2单良@163.com',1,2,'222222','男',20,to\_date('1997.05.22','yyyy-mm-dd'),2222 ,null,null,null ,null,null,null,null,null,null);

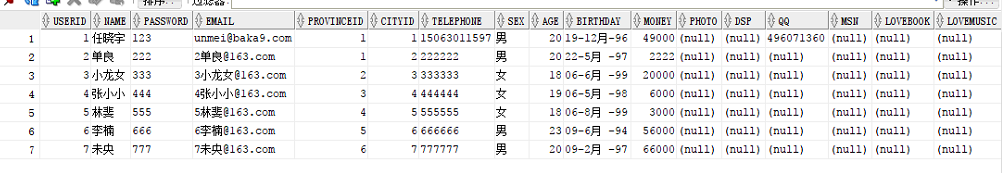
insert into tb\_user values(seq\_userID.nextval, '小龙女' ,'333','3小龙女@163.com',2,3,'333333','女',18,to\_date('1999.06.06 ','yyyy-mm-dd'),20000 ,null,null,null ,null,null,null,null,null,null);

insert into tb\_user values(seq\_userID.nextval, '张小小' ,'444','4张小小@163.com',3,4,'444444','女',19,to\_date('1998.05.06 ','yyyy-mm-dd'),6000 ,null,null,null ,null,null,null,null,null,null);

insert into tb\_user values(seq\_userID.nextval, '林斐' ,'555','5林斐@163.com',4,5,'555555','女',18,to\_date('1999.08.06 ','yyyy-mm-dd'),3000 ,null,null,null ,null,null,null,null,null,null);

insert into tb\_user values(seq\_userID.nextval, '李楠' ,'666','6李楠@163.com',5,6,'666666','男',23,to\_date('1994.06.09 ','yyyy-mm-dd'),56000 ,null,null,null ,null,null,null,null,null,null);

insert into tb\_user values(seq\_userID.nextval, '未央' ,'777','7未央@163.com',6,7,'777777','男',20,to\_date('1997.02.09 ','yyyy-mm-dd'),66000 ,null,null,null ,null,null,null,null,null,null);



**六、商品信息(商品信息+型号+颜色)添加**

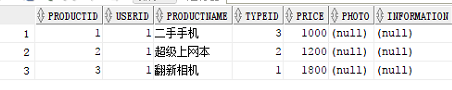
/\*商品信息(商品信息+型号+颜色)添加\*/

/\*tb\_product表添加商品信息\*/

insert into tb\_product values(seq\_productID.nextval,1,'二手手机',3,1000,null,null);

insert into tb\_product values(seq\_productID.nextval,1,'超级上网本',2,1000,null,null);

insert into tb\_product values(seq\_productID.nextval,1,'翻新相机',1,1000,null,null);



/\*tb\_productSize表添加商品型号\*/

insert into tb\_productSize values(seq\_productSizeID.nextval,'魅族Pro7',3);

insert into tb\_productSize values(seq\_productSizeID.nextval,'华为荣耀9',3);

insert into tb\_productSize values(seq\_productSizeID.nextval,'OnePlus 5',3);

insert into tb\_productSize values(seq\_productSizeID.nextval,'小米6',3);

insert into tb\_productSize values(seq\_productSizeID.nextval,'ThinkPad X1',2);

insert into tb\_productSize values(seq\_productSizeID.nextval,'Dell灵越15',2);

insert into tb\_productSize values(seq\_productSizeID.nextval,'HP pavilion15',2);

insert into tb\_productSize values(seq\_productSizeID.nextval,'佳能EOS',1);

insert into tb\_productSize values(seq\_productSizeID.nextval,'尼康D5300',1);

insert into tb\_productSize values(seq\_productSizeID.nextval,'徕卡SL',1);

insert into tb\_productSize values(seq\_productSizeID.nextval,'尼康D3400',1);



/\*tb\_productColor表添加商品型号\*/

insert into tb\_productColor values(seq\_productcolorID.nextval,'靛蓝',90,1);

insert into tb\_productColor values(seq\_productcolorID.nextval,'碳黑',100,2);

insert into tb\_productColor values(seq\_productcolorID.nextval,'品红',10,9);

insert into tb\_productColor values(seq\_productcolorID.nextval,'金黄',30,6);

insert into tb\_productColor values(seq\_productcolorID.nextval,'亮银',200,4);



**七、删除用户信息**

/\*删除用户\*\*\*\*/

create or replace trigger tr\_delOnUser\_delOrder

after delete on tb\_user

for each row

begin

delete from tb\_order where toID=:old.userID;

end;

create or replace trigger tr\_delOnOrder\_delUserAccount

after delete on tb\_order

for each row

begin

delete from tb\_userAccount where orderID=:old.orderId;

end;

**八、商品信息查询**

/\*商品信息查询\*/

/\*单表查询\*/

/\*查询tb\_product表中商品id为的商品信息\*/

set serveroutput on;

declare

gno number;

ansRow tb\_product%rowtype;

begin

gno:=&商品号;

select \* into ansRow from tb\_product

where productID=gno;

dbms\_output.put\_line('商品ID:'||ansRow.productID||'卖家ID:'||ansRow.userID||'商品名:'||ansRow.productName||'商品类型ID:'||ansRow.typeID||'价格:'||ansRow.price);

end;



/\*查询tb\_productSize表中商品id为的型号信息\*/

set serveroutput on;

declare

gno number;

begin

gno:=&商品号;

declare

cursor cs\_ans is

select \* from tb\_productSize

where productID=gno;

cs\_row cs\_ans%rowtype;

begin

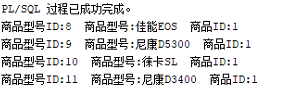
for cs\_row in cs\_ans loop

dbms\_output.put\_line('商品型号ID:'||cs\_row.productSizeID||' 商品型号:'||cs\_row.productSize||' 商品ID:'||cs\_row.productID);

end loop;

end;

end;



/\*查询tb\_productColor表中商品型号id为、、的颜色与库存信息\*/

set serveroutput on;

declare

gno number;

ansRow tb\_productColor%rowtype;

cursor cur1(gno number)

is

select \* from tb\_productColor

where productSizeID=gno;

begin

gno:=&商品类型ID;

open cur1(gno);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('颜色:'||ansRow.productColor||' 库存:'||ansRow.stockpile);

end loop;

close cur1;

end;



/\*多表查询\*/

/\*用户在商城以商品类型查询一件商品的名称\*/

set serveroutput on;

declare

gno number;

ansRow tb\_product%rowtype;

cursor cur1(gno number)is

select \* from tb\_product

where typeID=gno;

begin

gno:=&商品类ID;

open cur1(gno);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('商品名:'||ansRow.productName);

end loop;

close cur1;

end;



/\*用户在商城以商品名称查询一件商品的所有信息\*/

set serveroutput on;

declare

gname tb\_product.productName%type;

ansRow tb\_product%rowtype;

cursor cur1(gname tb\_product.productName%type)

is

select \* from tb\_product

where productName=gname;

begin

gname:='&商品名';

open cur1(gname);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('商品ID:'||ansRow.productID||'卖家ID:'||ansRow.userID||'商品名:'||ansRow.productName||'商品类型ID:'||ansRow.typeID||'价格:'||ansRow.price||'商品图片路径:'||ansRow.photo||'商品简介:'||ansRow.information);

end loop;

close cur1;

end;



**九、用户信息查询**

/\*用户信息查询\*/

/\*用户在商城可以查询已知用户(或卖家)的基本信息\*/

set serveroutput on;

declare

username tb\_user.name%type;

ansRow tb\_user%rowtype;

cursor cur1(username tb\_user.name%type)is

select \* from tb\_user

where name=username;

begin

username:='&用户姓名';

open cur1(username);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('用户ID:'||ansRow.userID||' 用户名:'||ansRow.name||' 邮箱:'||ansRow.email||' 省ID:'||ansRow.provinceID||' 市ID:'||ansRow.cityID||' 手机号:'||ansRow.telephone||' 性别:'||ansRow.sex||' 年龄:'||ansRow.age||' 生日:'||ansRow.birthday);

end loop;

close cur1;

end; 

**十、用户下订单并确认付款（应用）**

/\*用户下订单并确认付款(应用事务)\*/

/\*用户\*\*\*想要购买卖家\*\*\*的产品颜色-大小-产品名\*/

/\*定义变量,用于存储本次订单的物品单价\*/

/\*定义变量,用于存储本次订单的物品ID\*/

/\*定义变量,用于存储本次订单的颜色ID\*/

/\*定义变量,用于存储本次订单的买家ID\*/

/\*定义变量,用于存储流水表的用户ID\*/

Declare

Order\_toID number;

Order\_colorID number;

Order\_productID number;

Order\_conut number;

Order\_Sum number;

Order\_address varchar2(100);

Order\_tel char(11);

order\_date date;

stock number;

begin

Order\_toID:='&输入本次订单的买家ID';

Order\_productID:='&输入本次订单的物品ID';

Order\_colorID:='&输入本次订单的颜色ID';

Order\_conut:='&输入订购的数量';

Select price\*Order\_conut into order\_sum from tb\_product where productID= Order\_productID;

Select cityName into Order\_address from tb\_user,tb\_city where tb\_user.cityID =tb\_city.cityID and userID= Order\_toID;

Select telephone into order\_tel from tb\_user where userID= Order\_toID;

select sysdate into order\_date from dual;

select stockpile into stock from tb\_productColor where productColorID=Order\_colorID;

if stock>=Order\_conut then

begin

update tb\_productColor set stockpile=stockpile-Order\_conut where productColorID=Order\_colorID;

Insert into tb\_order(toID,colorID,productID,address,telephone,orderstate,count,price,orderDate,consignmentDate)

values(Order\_toID, Order\_productID, Order\_colorID, order\_address,order\_tel,0,Order\_conut,order\_sum,order\_date,order\_date+2);

end;

else

dbms\_output.put\_line('库存不足');

end if;

end;

/\*用户\*\*\*\*下订单\*/

/\*用户\*\*\*\*将订单信息填入到tb\_order表中\*/

--orderState订单状态为用户选择填入下订单(买家付款后自动转为已付款,买家付款后卖家可选择修改为延后,卖家发货后选择修改为发货,买家未付款时卖家可选择修改为撤销,买家收到货物后可选择修改为完成)

/\*下订单后减少商品库存\*/

/\*确认付款后减少用户余额\*/

/\*将用户付款的资金冻结后订单状态改为已付款\*/

/\*用户账户上的钱转账完成后建立用户资金流水表一条买家购买商品支出记录\*/

declare

order\_userID number;

Order\_orderID number;

order\_time date;

fk number;

pri number;

psum number;

m number;

begin

order\_userID:=&买家ID;

Order\_orderID:=&订单ID;

fk:=&是否付款;

select sysdate into order\_time from dual;

select price into pri from tb\_order where orderID=Order\_orderID;

select money-pri into psum from tb\_user where userid=order\_userID;

if fk=1 then

if psum>=0 then

begin

update tb\_user set money=money-pri where userid=order\_userID;

insert into tb\_userAccount values(seq\_useraccountID.nextval,order\_userID,order\_orderid,1,order\_time);

update tb\_order set orderState=1 where orderID=Order\_orderID;

end;

else

dbms\_output.put\_line('余额不足，付款失败');

end if;

end if;

end;





**十一、买家对订单信息进行修改（包括发货后的订单状态修改及修改商品颜色时应用事务）**

/\*卖家对订单信息进行修改\*/

/\*假设买家\*\*\*想修改商品的颜色，则需要应用事务，对相应颜色的库存进行修改\*/

/\*修改信息是各项信息均为可选择项，如不想修改某项信息，则选择以前的选项，但是会将数据重复的修改一次，即用相同的数据来覆盖之前的数据\*/

/\*除此之外卖家发货后可将订单状态改为发货\*/

/\*定义变量,用于存储本次订单的物品单价\*/

/\*修改tb\_order表中的送货地址\*/

/\*修改tb\_order表中的联系方式\*/

/\*修改tb\_order表中的状态\*/

--若卖家因故不能发货，但已付款则卖家可修改订单状态为'延后'

/\*修改tb\_order表中的商品颜色\*/

/\*修改tb\_order表中的商品总价\*/

--若和买家商议后买家同意降价，但商品信息上还是不修改的，就可以在订单总价上修改，付款后卖家修改总价，差价打回买家账户余额内

/\*将与之前的差价加回买家账户余额内\*/

/\*修改tb\_user表中买家的账户余额\*/

/\*修改颜色后要将之前选择的颜色库存加一再讲修改后的颜色库存减一\*/

declare

order\_sprice number;

order\_address varchar2(100);

order\_tel char(11);

order\_statement number;

order\_orderid number;

order\_if number;

order\_if1 number;

order\_color varchar2(20);

order\_userid number;

num number;

cid number;

cha number;

sum number;

begin

order\_orderid:=&输入订单ID;

order\_if:=&是否已经发货;

order\_address:='&输入修改后的地址';

order\_tel:='&输入修改后的联系方式';

order\_color:=&输入修改后的颜色id;

select count into num from tb\_order where orderID=order\_orderid;

select colorID into CID from tb\_order where orderID=order\_orderid;

if order\_if=0 then

begin

update tb\_order set orderState=3 where orderId=order\_orderid;

update tb\_order set address=order\_address where orderId=order\_orderid;

update tb\_order set telephone=order\_tel where orderId=order\_orderid;

update tb\_order set colorID=order\_color where orderId=order\_orderid;

update tb\_productColor set stockpile=stockpile+num where productColorID=Cid;

update tb\_productColor set stockpile=stockpile-num where productColorID=order\_color;

order\_if1:=&是否同意降价;--0不同意,1同意

if order\_if1=1 then

begin

order\_sprice:=&输入降价后的单价;

select toID into order\_userid from tb\_order where orderId=order\_orderid;

select price-order\_sprice into cha from tb\_order where orderId=order\_orderid;

update tb\_order set price=num\*order\_sprice where orderId=order\_orderid;

update tb\_user set money=money+num\*cha where userID=order\_userid;

end;

end if;

end;

else

update tb\_order set orderState=4 where orderId=order\_orderid;

end if;

end;

**十二、订单完成进行转账**

/\*订单完成进行转账\*/

/\*定义变量,用于存储流水表的用户ID\*/

/\*定义变量,用于存储本次订单的物品单价\*/

/\*买家收到商品后，修改tb\_order表中的状态\*/

/\*订单状态为'完成'后建立用户资金流水表一条卖家出售商品收入记录\*/

/\*交易成功后卖家账户余额增加\*/

declare

userAccount\_ID number;

price number;

order\_state number;

user\_id number;

order\_orderid number;

sum number;

num number;

selluserID number;

userAccount\_time date;

begin

userAccount\_ID:=&流水表用户的ID;

order\_state:=&订单状态;

select orderID into order\_orderid from tb\_userAccount where userAccountID=userAccount\_ID;

select userID into selluserID from tb\_product,tb\_order where tb\_product.productID=tb\_order.productID and orderID=order\_orderid;

select price into sum from tb\_order where orderID=order\_orderid;

select count into num from tb\_order where orderID=order\_orderid;

select sysdate into userAccount\_time from dual;

if order\_state=5 then

begin

update tb\_order set orderState=5 where orderID=order\_orderid;

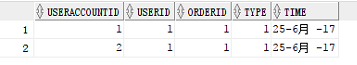
insert into tb\_userAccount values(seq\_useraccountID.nextval,selluserID,order\_orderid,1,userAccount\_time);

update tb\_user set money=money+sum where userID=selluserID;

end;

end if;

end;



**十三、货物丢件进行退款**

/\*货物丢件进行退款\*/

/\*前提条件:订单为发货状态，但物流出现丢件情况，由物流赔偿卖家，卖家确认丢件情况发生后，可进行退款操作\*/

/\*定义变量,用于存储流水表的用户ID\*/

/\*卖家确认商品丢件后，修改tb\_order表中的状态\*/

/\*订单状态为'撤销'后建立用户资金流水表一条买家商品退款收入记录\*/

/\*撤销成功后买家账户余额恢复\*/

declare

iflose number;

userAccount\_ID number;

order\_state number;

order\_orderid number;

userAccount\_time date;

buyuserID number;

refund number;

begin

iflose:=&是否丢失;--(0为是,1为否)

userAccount\_ID:=&流水表ID;

select orderID into order\_orderid from tb\_userAccount where userAccountID=userAccount\_ID;

select orderstate into order\_state from tb\_order where orderID=order\_orderId;

select sysdate into userAccount\_time from dual;

select toID into buyuserID from tb\_order where orderID=order\_orderid;

select price into refund from tb\_order where orderID=order\_orderid;

begin

if iflose=0 and order\_state=4 then

begin

update tb\_order set orderState=6;

insert into tb\_userAccount values(seq\_useraccountID.nextval,buyuserID,order\_orderId,2,userAccount\_time);

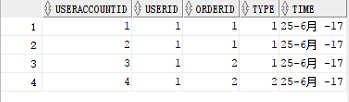
update tb\_user set money=money+refund where userid=buyuserID;

end;

end if;

end;

end;



**十四、视图**

--/\*多表查询\*/

--/\*用户在商城以商品类型查询一件商品的名称\*/

create view view\_product as

select \* from tb\_product join tb\_productType on typeID=productTypeID

with read only;

set serveroutput on;

declare

tno number;

ansRow view\_product%rowtype;

cursor cur1(tno number)

is

select \* from view\_product

where typeID=tno;

begin

tno:=&商品类ID;

open cur1(tno);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('商品名:'||ansRow.productName);

end loop;

close cur1;

end;



--/\*用户在商城以商品名称查询一件商品的所有信息\*/

set serveroutput on;

declare

gname number;

ansRow view\_product%rowtype;

begin

gname:=&商品号;

select \* into ansRow from view\_product

where productID=gname;

dbms\_output.put\_line('商品ID:'||ansRow.productID||'卖家ID:'||ansRow.userID||'商品名:'||ansRow.productName||'商品类型ID:'||ansRow.typeID||'价格:'||ansRow.price);

end;



--/\*用户信息查询\*/

--/\*用户在商城可以查询已知用户(或卖家)的基本信息\*/

create view view\_user as

select \* from tb\_user

with read only;

declare

username view\_user.name%type;

ansRow view\_user%rowtype;

cursor cur1(username view\_user.name%type)is

select \* from tb\_user

where name=username;

begin

username:='&用户姓名';

open cur1(username);

loop

fetch cur1 into ansRow;

exit when cur1%notfound;

dbms\_output.put\_line('用户ID:'||ansRow.userID||' 用户名:'||ansRow.name||' 邮箱:'||ansRow.email||' 省ID:'||ansRow.provinceID||' 市ID:'||ansRow.cityID||' 手机号:'||ansRow.telephone||' 性别:'||ansRow.sex||' 年龄:'||ansRow.age||' 生日:'||ansRow.birthday);

end loop;

close cur1;

end;



**十五、用户登录(选做) 例如:sql server 格式:**

/\*用户登录(应用存储过程)\*/

create procedure proc\_land

@resultName int = 0, --定义为表示用户名错误

@resultPassWord int =0, --定义为表示密码错误

@userName nvarchar(10) = '\*\*\*',

@PassWord nvarchar(30) = 'jhc123'

as

select @resultName = COUNT(1) from tb\_user where name=@userName --只要表中有输入的用户名则为表示用户名存在

select @resultPassWord = COUNT(1) from tb\_user where name=@userName and [password]=@PassWord --若存在的这个用户的信息里的密码等与输入的密码则为表示密码核对无误

begin

if @resultName = 1 and @resultPassWord = 1

print '用户登录成功！'

else if @resultName = 0

print '用户名错误！'

else if @resultPassWord = 0

print '密码错误！'

end

create or replace procedure proc\_land(username varchar2,userpwd varchar2,result out varchar2)

as

resultName number;

resultPassWord number;

begin

select Count(\*) into resultName from tb\_user where name=userName;

select Count(\*) into resultPassWord from tb\_user where name=userName and password= userpwd;

result:='赋值成功';

if resultName >0 and resultPassWord>0 then

result:='用户登录成功！';

else if resultName <=0 then

result:='用户名错误！';

else

result:='密码错误';

end if;

end if;

end;

set serveroutput on;

declare

ans varchar2(60);

begin

proc\_land('任晓宇','123',ans);

dbms\_output.put\_line(ans);

proc\_land('任晓宇1','123',ans);

dbms\_output.put\_line(ans);

proc\_land('任晓宇','1233',ans);

dbms\_output.put\_line(ans);

end;

