## 1.python生成markdown (filetype=py)

import os

f = open("oracle复习.md", 'a')

filePath = '..\Oracle-BeninViolet\classnotes'

if \_\_name\_\_ == '\_\_main\_\_':

    for i,j,k in os.walk(filePath):

        for o in k:

            s="@import \"" + o + "\"\n"

            f.write(s)

## 2.markdown生成PDF (filetype=md)

<!-- @import "Oracle0525.txt" -->

<!-- @import "Oracle0527.txt" -->

@import "Oracle0605.sql"

@import "Oracle0608.sql"

@import "Oracle0610.sql"

@import "Oracle0612.sql"

@import "Oracle0615.sql"

@import "Oracle0619.sql"

@import "Oracle0622.sql"

@import "Oracle0624.sql"

@import "Oracle0628.sql"

@import "Oracle0629.sql"

@import "Oracle0703.sql"

@import "Oracle0706.sql"

@import "Oracle0708.sql"

@import "Oracle0710.sql"

@import "Oracle0713.sql"

@import "Oracle0717.sql"

## 合并封面和pdf (filetype=tex)

\documentclass[a4paper]{article}

\usepackage{pdfpages}

\begin{document}

\includepdfmerge{1.pdf}

\includepdfmerge{2.pdf,1-72}

\includepdfmerge{3.pdf}

\end{document}

**4.ppt代码 (filetype=md)**

---

author: Benin

title: Demo Slide

date: June 29, 2020

---

# Oracle practical training

By:Benin

---

## ❌ What I Did .

## ✌️ What I Learned ?

## ✔️ What I Done !

---

# ❌ What I Did .

---

## install and unstall

Oracle & PL/SQL developer for time to time.

---

# searching online

for any way to run my code.

<img src="img/bing.png" height="30px;" style="border: 0;" />

<img src="img/sougou.png" height="30px;" style="border: 0;" />

<img src="img/360.png" height="30px;" style="border: 0;" />

<img src="img/google.png" height="30px;" style="border: 0;" />

<img src="img/baidu.png " height="30px;" style="border: 0;" />

---

# ELSE in failure

<p></p>

<p></p>

<p></p>

---

---

<p>

    <font class="fragment">Maybe...&nbsp;</font>

    <font class="fragment">something&nbsp;</font>

    <font class="fragment">for</font>

</p>

<div class="fragment">

    <h1>oracle coding!

    <h1>

</div>

<p class="fragment ">\*Tips for effiective Oracle included.</p>

---

## Tips for Batch Coding</h2>

<img src="img/quickcode.gif " alt=" "> <br> Using HbuilderX to code rapidly.

---

## How to run PL/SQL code

<img src="img/before.png" height="260px;" style="border: 0;" />

<img src="img/after.png " height="260px;" style="border: 0;" />

just add a '/' after your sentences.

---

- Symmetric writing

- Slight difference about login

- Append "on delete cascade" on foreign constraint tail

---

<p>

    <font class="fragment">Maybe...&nbsp;</font>

    <font class="fragment">something&nbsp;</font>

    <font class="fragment">else&nbsp;</font>

    <font class="fragment">for</font>

</p>

<div class="fragment">

    <h1>my show time

    <h1>

</div>

---

# Markdown

for type setting

---

# Reveal.js

for PPT

Using Markdown language <br>to create PPT runned by explorer.

---

# Latex

for type setting

Combine two PDF file into one.

---

# ✔️ What I Done !

---

### create a review note

using python

<img src="img/pythoncode.png " alt=" ">

---

OUTPUT

<img src="img/pythonoutput.png " alt=" ">

---

## created a instance

using cloud service(<a href="https://console.cloud.tencent.com/cvm/instance/index?action=TerminateInstances&rid=33 ">cloud.tencent.com</a>) .

---

# create a simple instance

<a href="../execute.sql">execute.sql</a>

---

# thank you!

the end.

## 5.ORACLE代码 (filetype=sql)

1. **创建数据库**

-- /\*创建用户\*/

conn sys/1@orcl as sysdba;

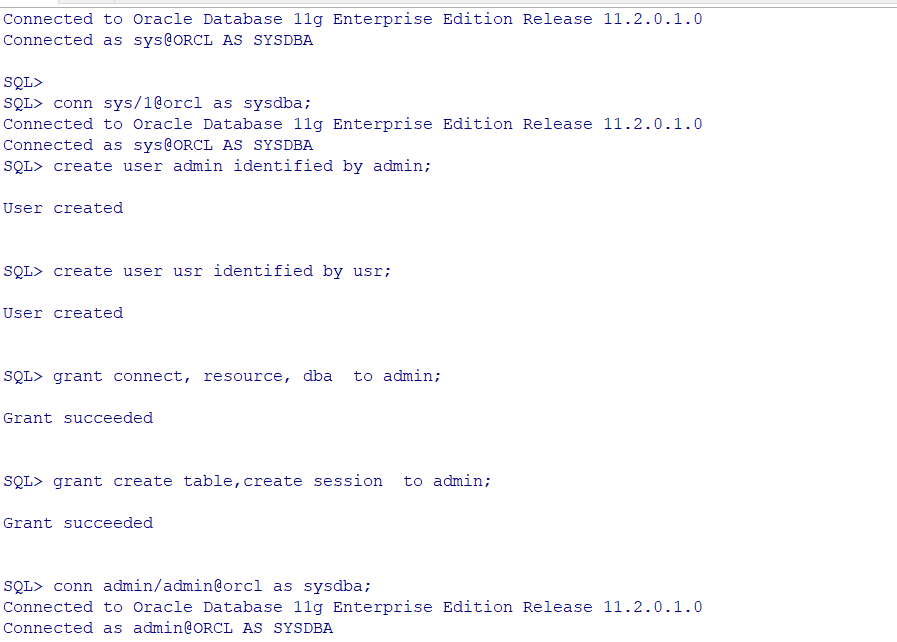
create user admin identified by admin;

create user usr identified by usr;

grant connect, resource, dba to admin;

grant create table,create session to admin;

conn admin/admin@orcl as sysdba;



1. **创建表**

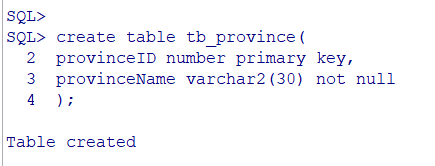
-- /\*创建tb\_province表\*/

create table tb\_province(

provinceID number primary key,

provinceName varchar2(30) not null

);



create sequence tb\_province\_provinceID\_seq

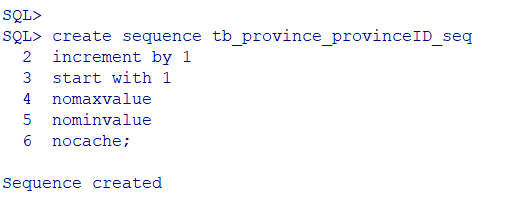
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_province\_provinceID

before insert on tb\_province

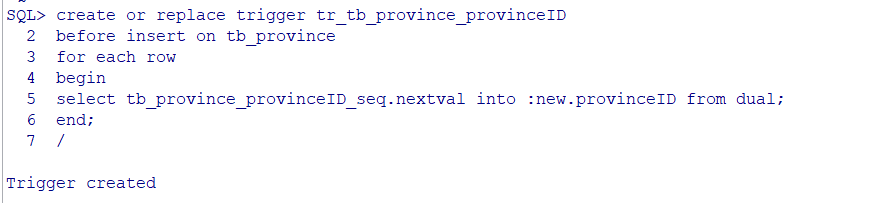
for each row

begin

select tb\_province\_provinceID\_seq.nextval into :new.provinceID from dual;

end;

/



-- /\*创建tb\_city表\*/

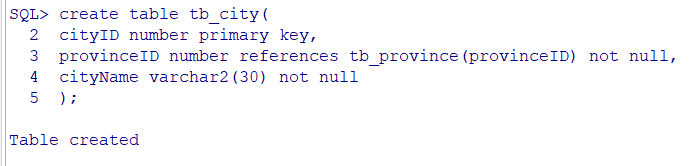
create table tb\_city(

cityID number primary key,

provinceID number references tb\_province(provinceID) not null,

cityName varchar2(30) not null,

);



create sequence tb\_city\_cityID\_seq

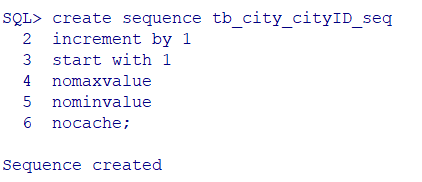
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_city\_cityID

before insert on tb\_city

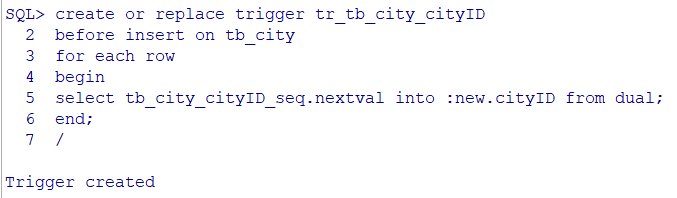
for each row

begin

select tb\_city\_cityID\_seq.nextval into :new.cityID from dual;

end;

/



-- /\*创建tb\_user表\*/

create table tb\_user(

userID number primary key,

name varchar2(10) unique,

password varchar2(30) not null,

email varchar2(30),

provinceID number,

cityID number references tb\_city(cityID),

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),

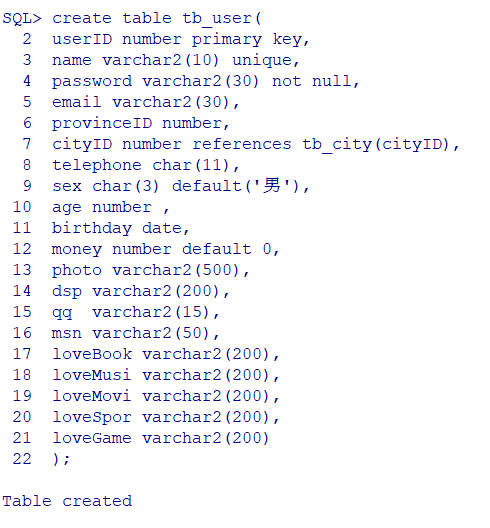
loveMusi varchar2(200),

loveMovi varchar2(200),

loveSpor varchar2(200),

loveGame varchar2(200)

);

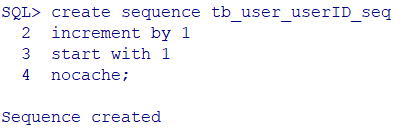


create sequence tb\_user\_userID\_seq

increment by 1

start with 1

nocache;



create or replace trigger tr\_tb\_user\_userID

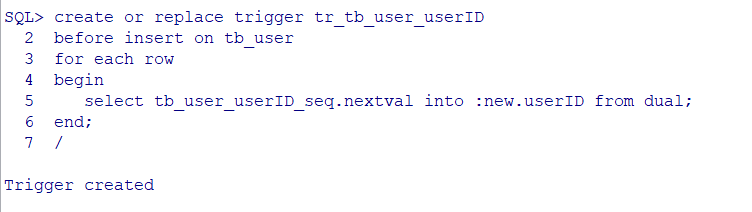
before insert on tb\_user

for each row

begin

select tb\_user\_userID\_seq.nextval into :new.userID from dual;

end;



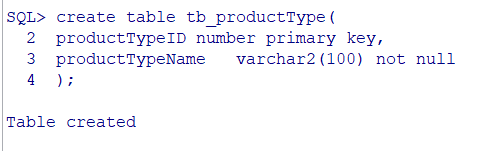
-- /\*创建tb\_productType表\*/

create table tb\_productType(

productTypeID number primary key,

productTypeName varchar2(100) not null

);



create sequence tb\_productType\_productTypeID\_seq

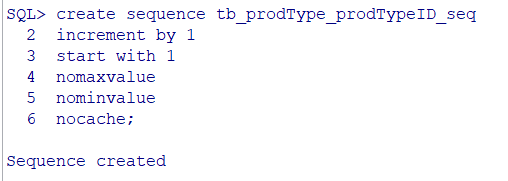
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_prodType\_prodTypeID

before insert on tb\_productType

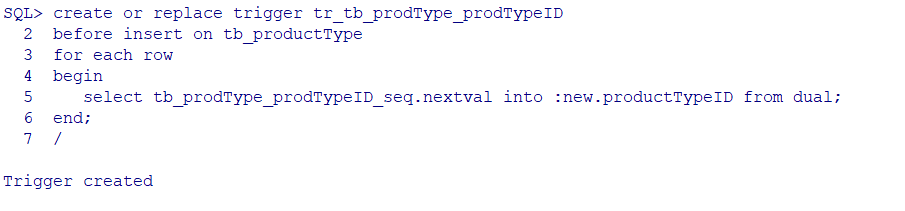
for each row

begin

select tb\_prodType\_prodTypeID\_seq.nextval into :new.productTypeID from dual;

end;

/



-- /\*创建tb\_product表\*/

create table tb\_product(

productID number primary key,

userID number references tb\_user(userID) not null,

productName varchar2(50) not null,

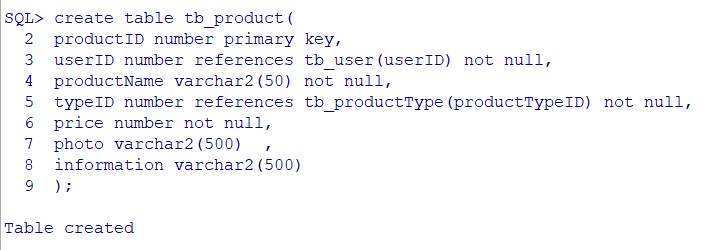
typeID number references tb\_productType(productTypeID) not null,

price number not null,

photo varchar2(500) ,

information varchar2(500)

);



create sequence tb\_product\_productID\_seq

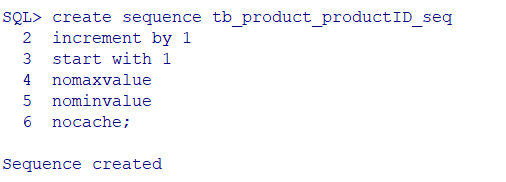
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_product\_productID

before insert on tb\_product

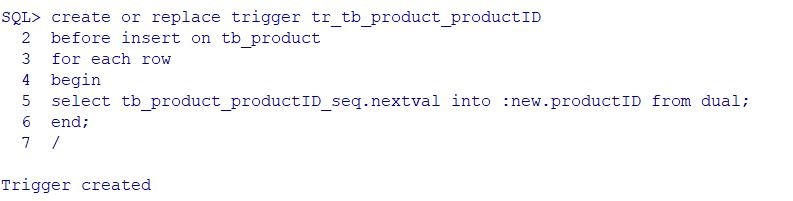
for each row

begin

select tb\_product\_productID\_seq.nextval into :new.productID from dual;

end;

/



-- /\*创建tb\_productSize表\*/

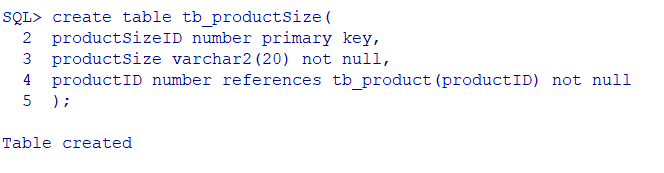
create table tb\_productSize(

productSizeID number primary key,

productSize varchar2(20) not null,

productID number references tb\_product(productID) not null,

);



create sequence tb\_productSize\_productSizeID\_seq

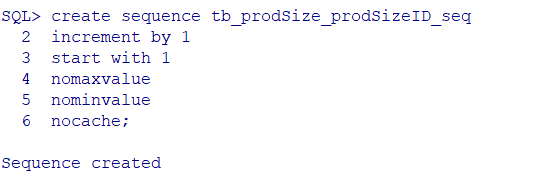
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_user\_userID

before insert on tb\_user

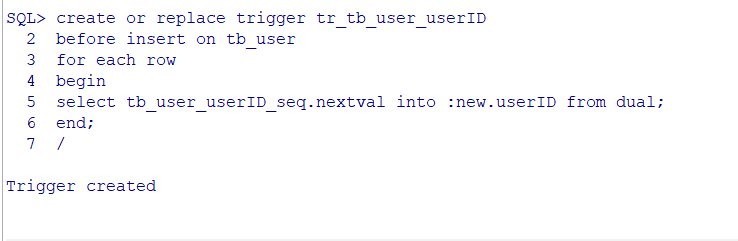
for each row

begin

select tb\_user\_userID\_seq.nextval into :new.userID from dual;

end;

/



-- /\*创建tb\_productColor表\*/

create table tb\_productColor(

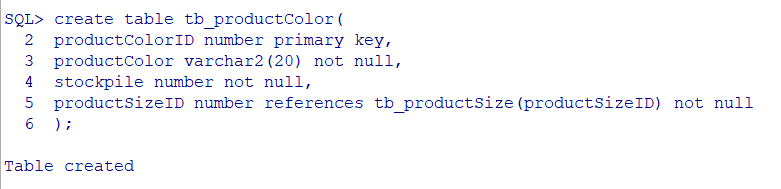
productColorID number primary key,

productColor varchar2(20) not null,

stockpile number not null,

productSizeID number references tb\_productSize(productSizeID) not null

);



create sequence tb\_prodColor\_prodColorID\_seq

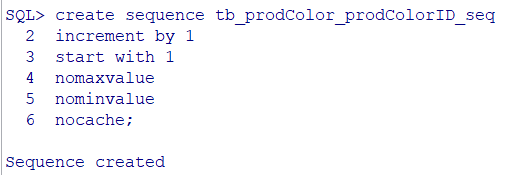
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_prodColor\_prodColorID

before insert on tb\_productColor

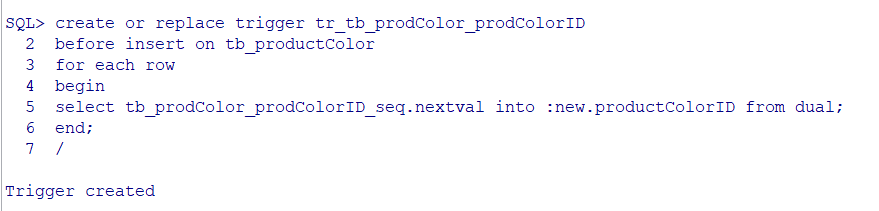
for each row

begin

select tb\_prodColor\_prodColorID\_seq.nextval into :new.productColorID from dual;

end;

/



-- /\*创建tb\_order表(orderState,orderID)\*/

create table tb\_order(

orderID number primary key,

toID number references tb\_user(userID) not null,

colorID number references tb\_productColor(productColorID) not null,

productI number references tb\_product(productID) not null,

address varchar2(100) not null,

telephone char(11) not null,

orderState number not null,

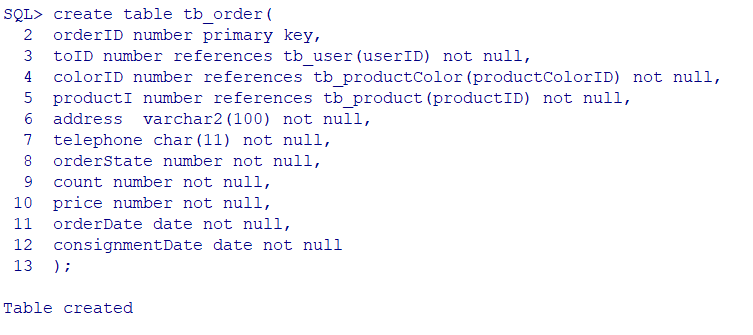
count number not null,

price number not null,

orderDate date not null,

consignmentDate date not null

);



create sequence tb\_order\_orderID\_seq

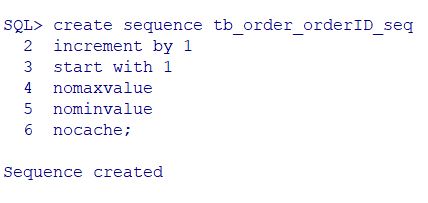
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_order\_orderID

before insert on tb\_order

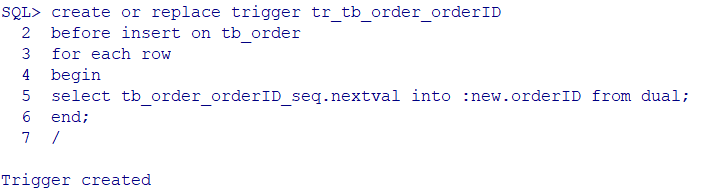
for each row

begin

select tb\_order\_orderID\_seq.nextval into :new.orderID from dual;

end;

/



/\*创建tb\_userAccount表\*/

create table tb\_userAccount(

userAccountID number primary key,

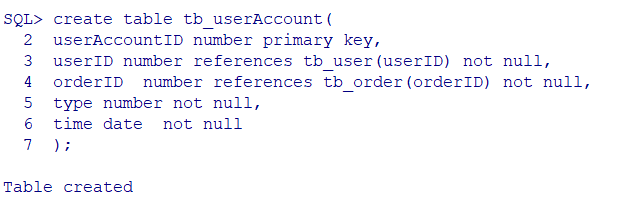
userID number references tb\_user(userID) not null,

orderID number references tb\_order(orderID) not null,

type number not null,

time date not null

);



create sequence tb\_userAccount\_userAccountID\_seq

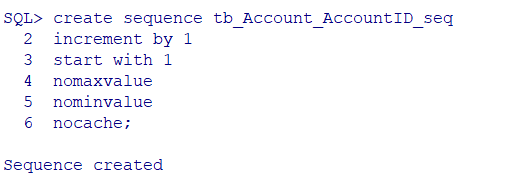
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



create or replace trigger tr\_tb\_Account\_AccountID

before insert on tb\_userAccount

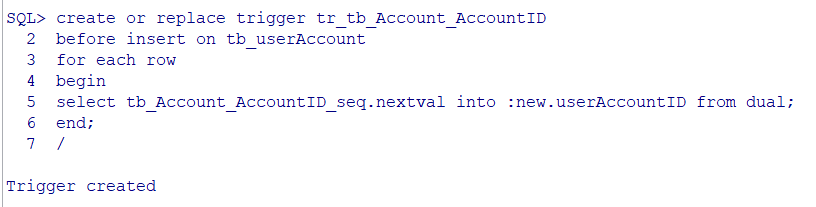
for each row

begin

select tb\_Account\_AccountID\_seq.nextval into :new.userAccountID from dual;

end;

/



-- /\*创建tb\_review表\*/

create table tb\_review(

reviewID number primary key,

review varchar2(250) not null,

isReply boolean default(false) not null,

isDel boolean default(false) not null,

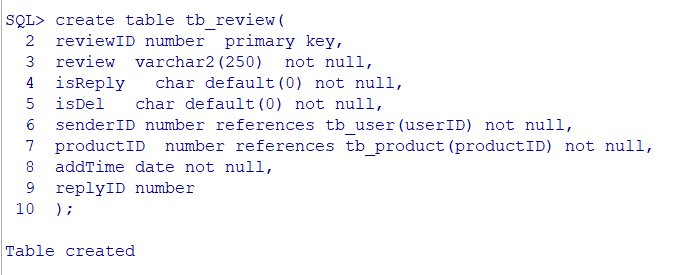
senderID number references tb\_user(userID) not null,

productID number references tb\_product(productID) not null,

addTime date not null,

replyID number

);



create sequence tb\_review\_reviewID\_seq

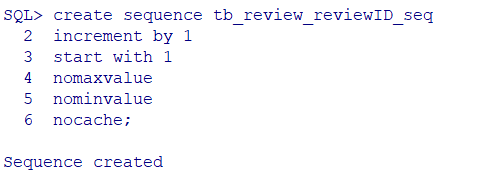
increment by 1

start with 1

nomaxvalue

nominvalue

nocache;



**三、添加约束**

/\*给每个表添加约束\*/

alter table tb\_city add constraint fk\_cityID\_provinceID foreign key(provinceID) references tb\_province(provinceID) on delete cascade;

alter table tb\_user add constraint fk\_userID\_cityID foreign key(cityID) references tb\_city(cityID) on delete cascade;

alter table tb\_product add constraint fk\_productID\_userID foreign key(userID) references tb\_user(userID) on delete cascade;

alter table tb\_product add constraint fk\_productID\_productTypeID foreign key(typeID) references tb\_productType(productTypeID) on delete cascade;

alter table tb\_productSize add constraint fk\_SizeID\_productID foreign key(productID) references tb\_product(productID) on delete cascade;

alter table tb\_productColor add constraint fk\_ColorID\_productSizeID foreign key(productSizeID) references tb\_productSize(productSizeID) on delete cascade;

alter table tb\_userAccount add constraint fk\_AccountID\_userID foreign key(userID) references tb\_user(userID) on delete cascade;

alter table tb\_userAccount add constraint fk\_AccountID\_orderID foreign key(orderID) references tb\_order(orderID) on delete cascade;

alter table tb\_review add constraint fk\_reviewID\_userID foreign key(senderID) references tb\_user(userID) on delete cascade;

alter table tb\_review add constraint fk\_reviewID\_productID foreign key(productID) references tb\_product(productID) on delete cascade;

**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\*/

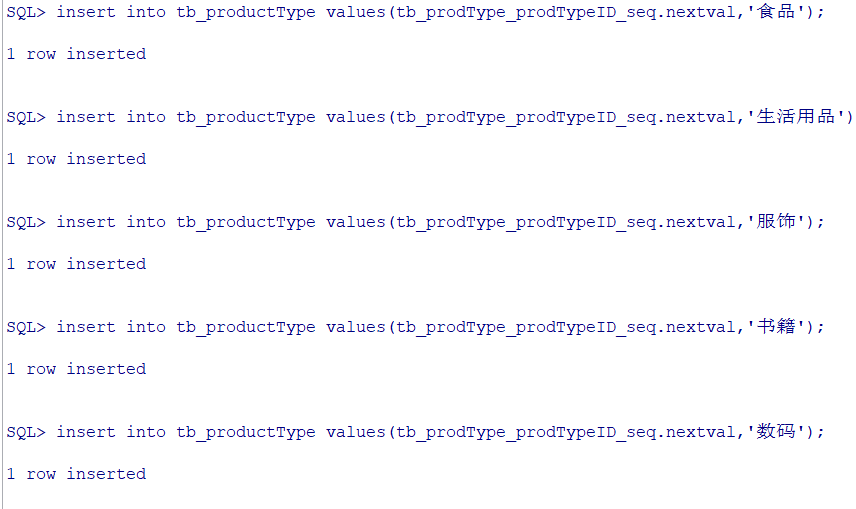
insert into tb\_productType values(tb\_prodType\_prodTypeID\_seq.nextval,'食品');

insert into tb\_productType values(tb\_prodType\_prodTypeID\_seq.nextval,'生活用品');

insert into tb\_productType values(tb\_prodType\_prodTypeID\_seq.nextval,'服饰');

insert into tb\_productType values(tb\_prodType\_prodTypeID\_seq.nextval,'书籍');

insert into tb\_productType values(tb\_prodType\_prodTypeID\_seq.nextval,'数码');



-- /\*tb\_province表添加省份名称\*/

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

insert into tb\_province values(tb\_province\_provinceID\_seq.nextval,'山西');

insert into tb\_province values(tb\_province\_provinceID\_seq.nextval,'河南');

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

insert into tb\_province values(tb\_province\_provinceID\_seq.nextval,'四川');



-- /\*tb\_city表给山东省添加城市名称\*/

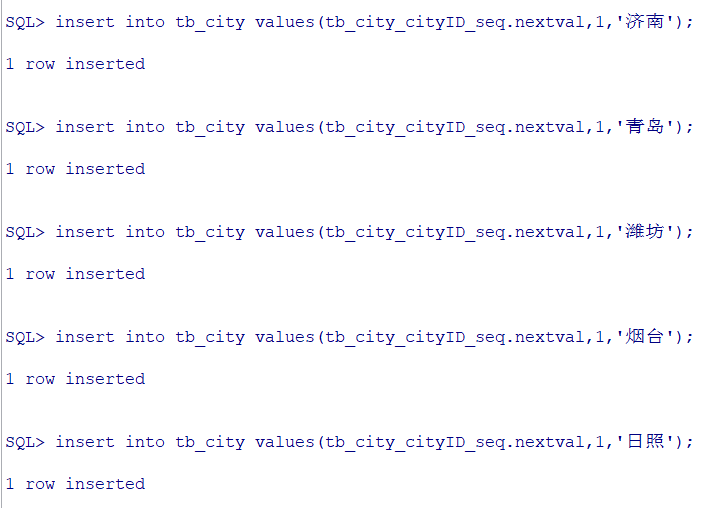
insert into tb\_city values(tb\_city\_cityID\_seq.nextval,1,'济南');

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

insert into tb\_city values(tb\_city\_cityID\_seq.nextval,1,'潍坊');

insert into tb\_city values(tb\_city\_cityID\_seq.nextval,1,'烟台');

insert into tb\_city values(tb\_city\_cityID\_seq.nextval,1,'日照');



**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\*/

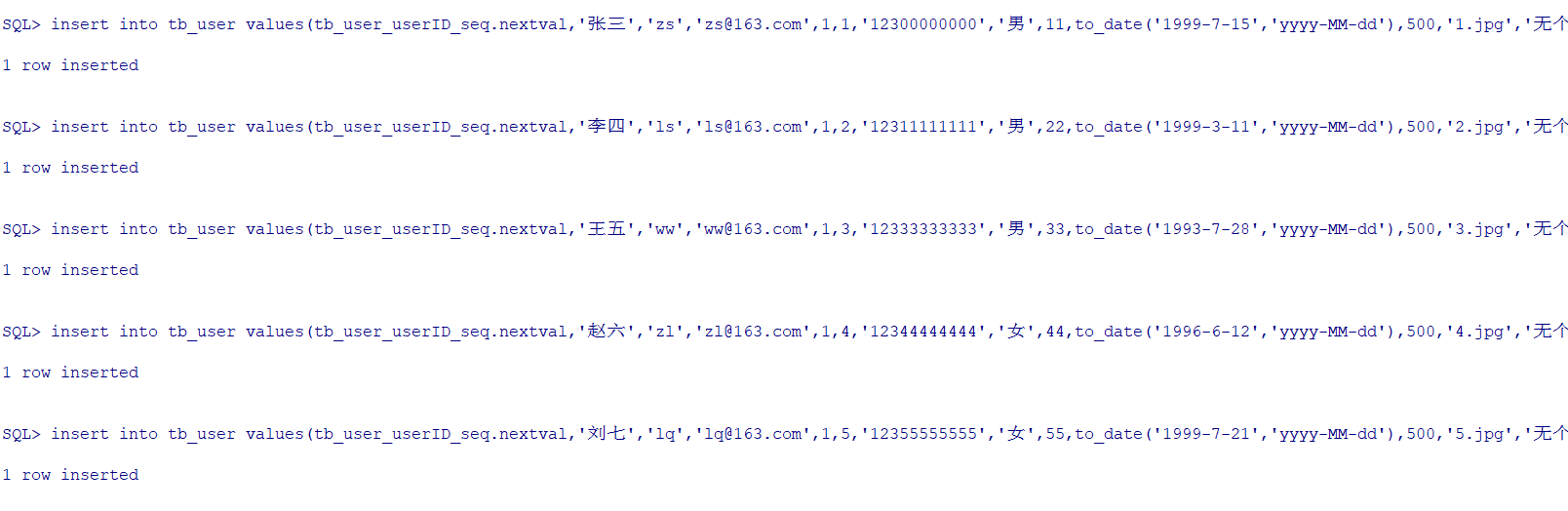
insert into tb\_user values(tb\_user\_userID\_seq.nextval,'张三','zs','zs@163.com',1,1,'12300000000','男',11,to\_date('1999-7-15','yyyy-MM-dd'),500,'1.jpg','无个性不签名','999999','44444','book1','music1','movie1','sport1','game1');

insert into tb\_user values(tb\_user\_userID\_seq.nextval,'李四','ls','ls@163.com',1,2,'12311111111','男',22,to\_date('1999-3-11','yyyy-MM-dd'),500,'2.jpg','无个性不签名','888888','44444','book2','music2','movie2','sport2','game2');

insert into tb\_user values(tb\_user\_userID\_seq.nextval,'王五','ww','ww@163.com',1,3,'12333333333','男',33,to\_date('1993-7-28','yyyy-MM-dd'),500,'3.jpg','无个性不签名','777777','44444','book3','music3','movie3','sport3','game3');

insert into tb\_user values(tb\_user\_userID\_seq.nextval,'赵六','zl','zl@163.com',1,4,'12344444444','女',44,to\_date('1996-6-12','yyyy-MM-dd'),500,'4.jpg','无个性不签名','666666','44444','book4','music4','movie4','sport4','game4');

insert into tb\_user values(tb\_user\_userID\_seq.nextval,'刘七','lq','lq@163.com',1,5,'12355555555','女',55,to\_date('1999-7-21','yyyy-MM-dd'),500,'5.jpg','无个性不签名','555555','44444','book5','music5','movie5','sport5



**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\*/

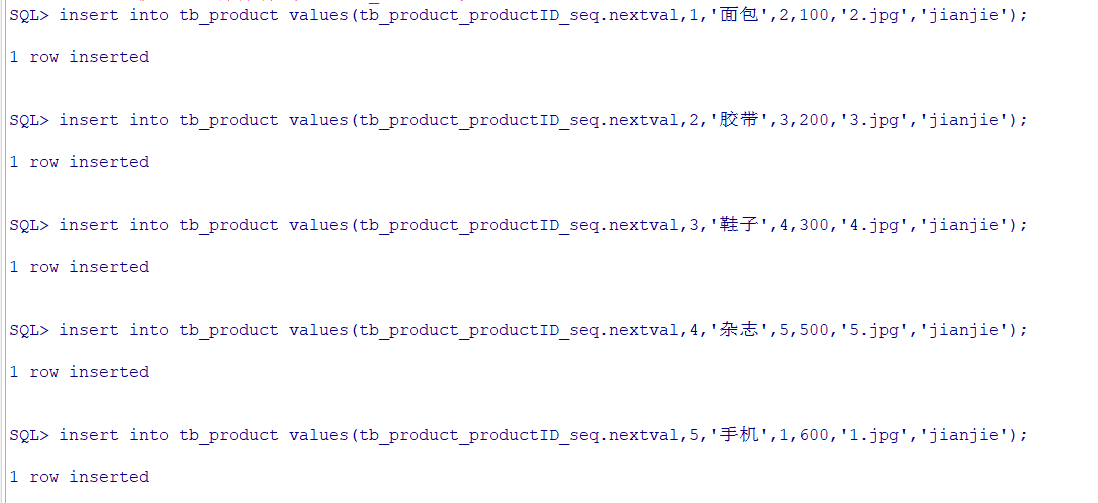
insert into tb\_product values(tb\_product\_productID\_seq.nextval,1,'面包',2,100,'2.jpg','jianjie');

insert into tb\_product values(tb\_product\_productID\_seq.nextval,2,'胶带',3,200,'3.jpg','jianjie');

insert into tb\_product values(tb\_product\_productID\_seq.nextval,3,'鞋子',4,300,'4.jpg','jianjie');

insert into tb\_product values(tb\_product\_productID\_seq.nextval,4,'杂志',5,500,'5.jpg','jianjie');

insert into tb\_product values(tb\_product\_productID\_seq.nextval,5,'手机',1,600,'1.jpg','jianjie');



-- /\*tb\_productSize表添加商品型号\*/

insert into tb\_productSize values(tb\_prodSize\_prodSizeID\_seq.nextval,'type1',1);

insert into tb\_productSize values(tb\_prodSize\_prodSizeID\_seq.nextval,'type2',2);

insert into tb\_productSize values(tb\_prodSize\_prodSizeID\_seq.nextval,'type3',3);

insert into tb\_productSize values(tb\_prodSize\_prodSizeID\_seq.nextval,'type4',4);

insert into tb\_productSize values(tb\_prodSize\_prodSizeID\_seq.nextval,'type5',5);



-- /\*tb\_productColor表添加商品型号\*/

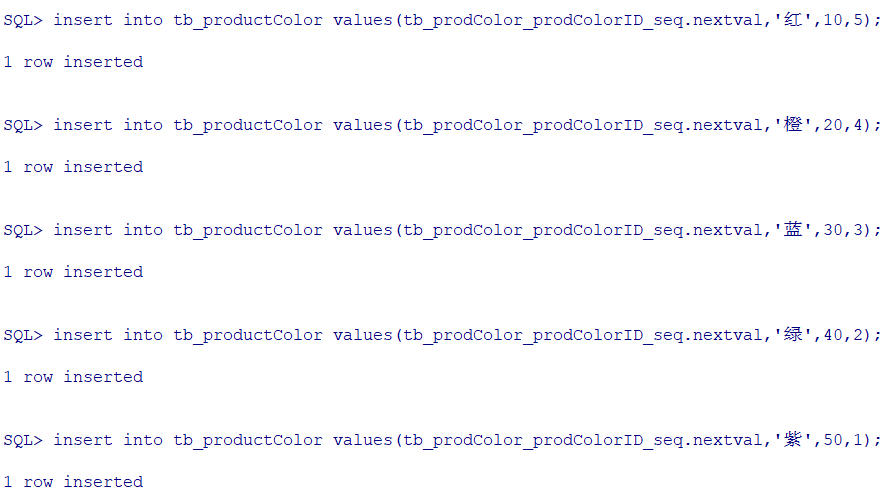
insert into tb\_productColor values(tb\_prodColor\_prodColorID\_seq.nextval,'红',10,5);

insert into tb\_productColor values(tb\_prodColor\_prodColorID\_seq.nextval,'橙',20,4);

insert into tb\_productColor values(tb\_prodColor\_prodColorID\_seq.nextval,'蓝',30,3);

insert into tb\_productColor values(tb\_prodColor\_prodColorID\_seq.nextval,'绿',40,2);

insert into tb\_productColor values(tb\_prodColor\_prodColorID\_seq.nextval,'紫',50,1);



**七、删除用户信息(使用plsql)**

-- /\*删除用户为XX的用户信息\*/

declare

v\_id number;

begin

v\_id:='&要删除的用户id';

delete from tb\_user where userID=v\_id;

-- alter table tb\_user disable novalidate constraint constraint\_name;

if sql%notfound then

dbms\_output.put\_line('删除失败!');

else

dbms\_output.put\_line('删除成功!');

end if;

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\*/

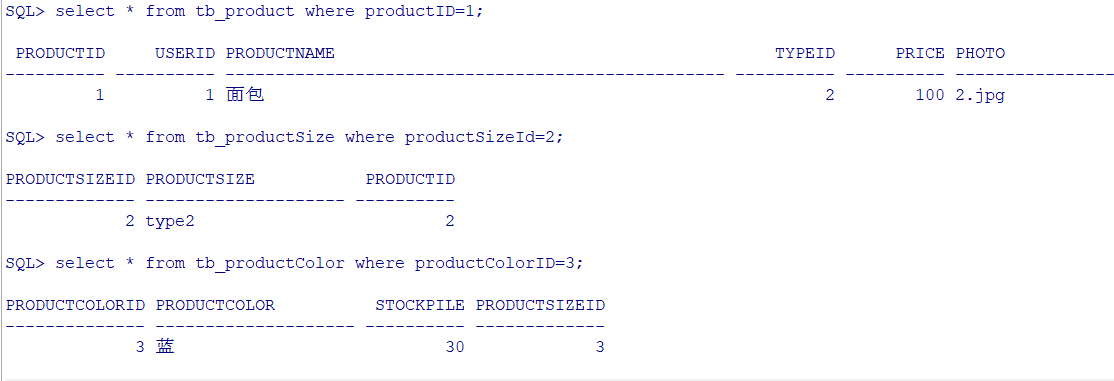
select \* from tb\_product where productID=1;

-- /\*查询tb\_productSize表中商品型号id为的型号信息\*/

select \* from tb\_productSize where productSizeId=2;

-- /\*查询tb\_productColor表中商品色号id为的颜色与库存信息\*/

select \* from tb\_productColor where productColorID=3;



-- /\*多表查询\*/

-- /\*用户在商城以商品类型查询一件商品的名称\*/

select productName from tb\_product tp join tb\_productType tpt on tp.typeId=tpt.productTypeId where productTypeName='服饰';

-- /\*用户在商城以商品名称查询一件商品的所有信息\*/

select \* from tb\_product tp

join tb\_productType tpt

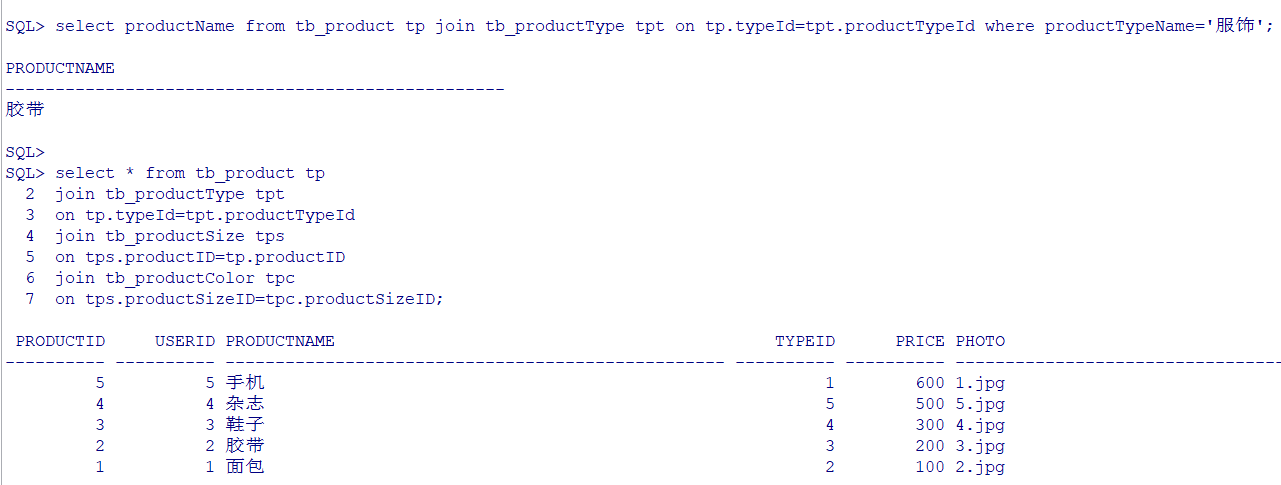
on tp.typeId=tpt.productTypeId

join tb\_productSize tps

on tps.productID=tp.productID

join tb\_productColor tpc

on tps.productSizeID=tpc.productSizeID;



**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\*/

declare

cursor c\_user is

select \* from tb\_user;

--定义记录变量

user\_info c\_user%rowtype;

begin

open c\_user;--打开游标

loop

FETCH c\_user

INTO user\_info;--获取记录值

EXIT WHEN c\_user%NOTFOUND;

dbms\_output.put\_line('学号：' || user\_info.userID ||

',姓名：' || user\_info.name||

',密码：' || user\_info.password||

',email：' || user\_info.email||

',省份：' || user\_info.provinceID||

',城市：' || user\_info.cityID||

',电话：' || user\_info.telephone||

',性别：' || user\_info.sex||

',年龄：' || user\_info.age||

',生日：' || user\_info.birthday||

',余额：' || user\_info.money||

',照片：' || user\_info.photo||

',签名：' || user\_info.dsp||

',qq：' || user\_info.qq||

'，喜欢的书：' || user\_info.loveBook||

'，喜欢的音乐：' || user\_info.loveMusi||

'，喜欢的电影：' || user\_info.loveMovi||

'，喜欢的运动：' || user\_info.loveSpor||

'，喜欢的游戏：' || user\_info.loveGame

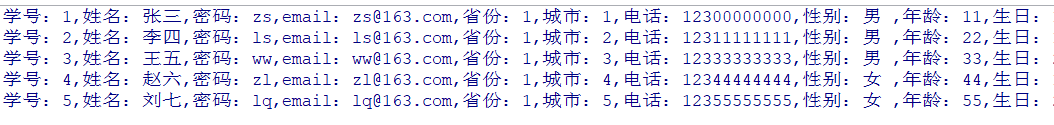
);

end loop;

close c\_user;--关闭游标

end;

/



**十、用户下订单并确认付款**

-- /\*用户下订单并确认付款\*/

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

-- /\*用户下订单并确认付款\*/

declare

this\_userId number;

this\_productColorId number;

this\_productId number;

this\_count number;

this\_stockPile number;

this\_price number;

this\_money number;

this\_phone char(11);

Ordernum number;

begin

this\_userid:='&买家id';

this\_productColorId:='&颜色Id';

this\_productId:='&商品id';

this\_count:='&数量';

select telephone into this\_phone from tb\_user where userId=this\_userId;

select price into this\_price from tb\_product where productId=this\_productId;

-- /\*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)

insert into tb\_order values

(tb\_order\_orderID\_seq.nextval,this\_userid,this\_productcolorid,this\_productid,'默认地址',this\_phone,0,this\_count,(this\_count\*this\_price),sysdate,(sysdate+3));

-- /\*下订单后减少商品库存\*/

select stockpile into this\_stockPile from tb\_productColor where productColorId=this\_productColorId;

if this\_stockPile>this\_count then

begin

update tb\_productColor set stockPile=stockPile-this\_count where productColorId=this\_productColorId;

end;

else

dbms\_output.put\_line('错误！库存不足');

end if;

-- /\*确认付款后减少用户余额\*/

select money into this\_money from tb\_user where userId=this\_userId;

if this\_money>this\_count\*this\_price then

begin

update tb\_user set money=this\_money-this\_count\*this\_price where userId=this\_userid;

dbms\_output.put\_line('交易成功');

end;

else

dbms\_output.put\_line( '错误！账户余额不足');

end if;

commit;

end;

--select \* from tb\_order;



**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）**

-- /\*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\*/

select \* from tb\_order;

declare

this\_orderId number;

this\_address varchar2(20);

this\_telephone char(11);

this\_orderState number;

this\_colorId number;

this\_price number;

this\_count number;

this\_bef\_price number;

this\_old\_productColorId number;

begin

this\_orderId:=&订单编号Id;

-- /\*修改tb\_order表中的送货地址\*/

this\_address:='&送货地址';

update tb\_order set address=this\_address where orderId=this\_orderId;

dbms\_output.put\_line('送货地址已修改');

-- /\*修改tb\_order表中的联系方式\*/

this\_telephone:='&联系方式';

update tb\_order set telephone=this\_telephone where orderId=this\_orderId;

dbms\_output.put\_line('联系方式已修改');

-- /\*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'

this\_orderState:=&修改状态;

update tb\_order set orderState=this\_orderState where orderId=this\_orderId;

dbms\_output.put\_line('订单状态已修改');

-- /\*修改tb\_order表中的商品颜色\*/

select colorId into this\_old\_productColorId from tb\_order where orderId=this\_orderId;

select count into this\_count from tb\_order where orderId=this\_orderId;

this\_colorId:='&颜色id';

update tb\_order set colorId=this\_colorId where orderId=this\_orderId;

-- /\*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

select price into this\_bef\_price from tb\_order where orderId=this\_orderId;

this\_price:='&当前价格';

-- /\*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\*/

update tb\_user set money=money+this\_bef\_price-this\_price where userId=(select toId from tb\_order where orderId=this\_orderId);

dbms\_output.put\_line('商品总价已修改');

dbms\_output.put\_line('操作成功');

-- /\*修改颜色后要将之前选择的颜色库存加一再将修改后的颜色库存减一\*/

update tb\_productcolor set stockPile =stockPile-this\_count where productColorId=this\_orderId;

update tb\_productColor set stockPile=stockPile+this\_count where productColorId=this\_old\_productColorId;

dbms\_output.put\_line('商品颜色已修改');

commit;

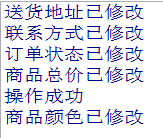
exception

when no\_data\_found then

dbms\_output.put\_line('数据未找到，操作失败');

rollback;

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\*/

declare

this\_userId number;

this\_sellerId number;

this\_price number;

this\_orderId number;

begin

this\_orderId:='&订单号';

-- /\*买家收到商品后，修改tb\_order表中的状态\*/

update tb\_order set orderState=3 where orderId=this\_orderId;

dbms\_output.put\_line('订单状态已完成');

-- /\*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\*/

select userId into this\_sellerId from tb\_product where productId=

(select productId from tb\_order where orderId=this\_orderId);

insert into tb\_useraccount values(tb\_Account\_AccountID\_seq.nextval,this\_sellerid,this\_orderId,1,sysdate);

-- dbms\_output.put\_line('用户资金流水表插入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\*/

select price into this\_price from tb\_order where orderId=this\_orderId;

update tb\_user set money=money+this\_price where userId=this\_sellerId;

dbms\_output.put\_line('卖家账户余额增加');

dbms\_output.put\_line('操作成功');

COMMIT;

exception

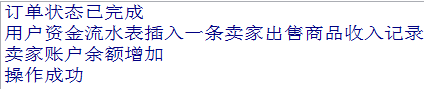
when no\_data\_found then

dbms\_output.put\_line('数据未找到，操作失败');

rollback;

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\*/

declare

this\_sellerId number;

this\_userId number;

this\_orderId number;

this\_price number;

begin

this\_orderId:='&订单编号';

-- /\*卖家确认商品丢件后，修改tb\_order表中的状态\*/

update tb\_order set orderState=5 where orderId=this\_orderId;

dbms\_output.put\_line('订单状态:撤销');

-- /\*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\*/

select toid into this\_userId from tb\_order where orderId=this\_orderId;

insert into tb\_useraccount values(tb\_Account\_AccountID\_seq.nextval,this\_userId,this\_orderId,1,sysdate);

dbms\_output.put\_line('用户资金流水表插入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\*/

select price into this\_price from tb\_order where orderId=this\_orderId;

update tb\_user set money=money+this\_price where userId=this\_userId;

dbms\_output.put\_line('买家账户余额恢复');

dbms\_output.put\_line('操作成功');

commit;

exception

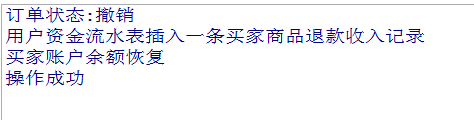
when no\_data\_found then

dbms\_output.put\_line('数据未找到，操作失败');

rollback;

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\*/

--drop view prodname\_view;

create view prodname\_view

as

select \* from tb\_product tp ,tb\_productType tpt where tp.typeid=tpt.producttypeid;

declare

name varchar2(20);

dname varchar2(20);

cursor cur(xname varchar2)

is

select productName from prodname\_view where productTypeName=xname;

begin

dname:='&商品类型';

open cur(dname);

loop

fetch cur into name;

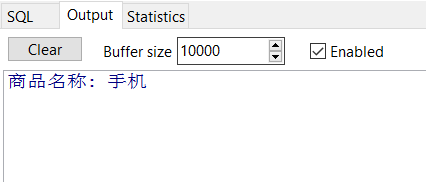
exit when cur%notfound;

dbms\_output.put\_line('商品名称: '||name);

end loop;

close cur;

end;



--/\*用户在商城以商品名称查询一件商品的所有信息\*/

--drop view allinfo\_view;

create view allinfo\_view

as

select productcolorid,productcolor,stockpile,

c.productsizeid,productsize,s.productid,userid,productname,typeid,price,photo,information

from tb\_productcolor c

join tb\_productsize s on c.productsizeid=s.productsizeid

join tb\_product p on p.productid=s.productid;

declare

dm allinfo\_view%rowtype;

pname varchar2(50);

cursor cur\_t (product\_name in varchar2)

is

select \* from allinfo\_view

where productname=product\_name;

begin

pname:='&商品名称';

open cur\_t(pname);

loop

fetch cur\_t into dm;

exit when cur\_t%notfound;

dbms\_output.put\_line(

'商品id：'||dm.productid||

'商品名字：'||dm.productname||

'商品颜色id：'||dm.productcolorid||

'商品颜色：'||dm.productcolor||

'商品库存：'||dm.stockpile||

'商品型号id：'||dm.productsizeid||

'商品型号：'||dm.productsize||

'商品卖家id：'||dm.userid||

'商品类型id：'||dm.typeid||

'商品价格：'||dm.price||

'商品图片路径：'||dm.photo||

'商品简介：'||dm.information

);

end loop;

close cur\_t;

exception

when no\_data\_found then

dbms\_output.put\_line('无数据！');

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\*/

declare

userId0 tb\_user.userId%type;

user tb\_user%rowtype;

begin

userId0:='&用户ID';

select \* into user from tb\_user where userId=userId0;

dbms\_output.put\_line(

'用户id：'||user.userid||

'，用户名：'||user.name||

'，邮箱：'||user.email||

'，所在省份id：'||user.provinceid||

'，所在城市id：'||user.cityid||

'，手机号：'||user.telephone||

'，性别：'||user.sex||

'，年龄：'||user.age||

'，生日：'||user.birthday

);

exception

when no\_data\_found then

dbms\_output.put\_line('没有发现您要找的数据！');

end;--/\*用户在商城可以查询已知用户(或卖家)的基本信息\*/

/



**十五、用户登录**

--/\*用户登录(应用存储过程)\*/

--自动登录

create or replace procedure login(login\_name in varchar2,login\_password in varchar2)

as

huoqu\_password varchar2(30);

begin

select password into huoqu\_password from tb\_user where name=login\_name;

if huoqu\_password=login\_password then

dbms\_output.put\_line(login\_name||',欢迎登陆！');

else

dbms\_output.put\_line('密码错误！');

end if;

EXCEPTION

when no\_data\_found then

dbms\_output.put\_line('该用户尚未注册！');

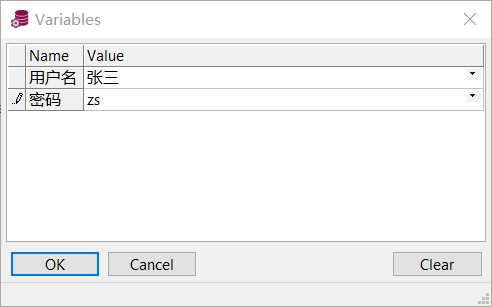
end login;

/

begin

login('张三','zs');

end;



--手动登录

declare

login\_name varchar2(10);

login\_password varchar2(30);

get\_password varchar2(30);

begin

login\_name:='&用户名';

login\_password:='&密码';

select password into get\_password from tb\_user where name=login\_name;

if get\_password=login\_password then

dbms\_output.put\_line(login\_name||',欢迎登陆！');

else

dbms\_output.put\_line('密码错误！');

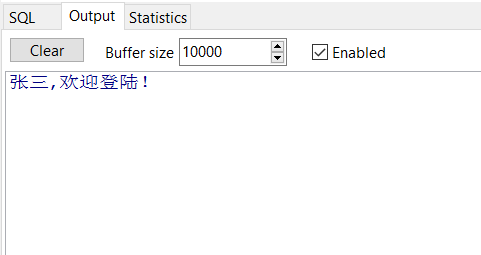
end if;

EXCEPTION

when no\_data\_found then

dbms\_output.put\_line('该用户尚未注册！');

end login;



## 作业代码

<!-- using markdown-preview-enhance to create pdf file. -->

# Excise 1.1

## 理论练习

1、Oracle是\_\_ \*\*对象关系型\*\* \_\_数据库管理系统。

2、Oracle 9i中的i和Oracle 10g、Oracle11g中的g各代表什么意思。

\*\*I:Internet,因特网

G:grid,网格计算\*\*

3、Oracle11g中的g表示（\*\*C\*\*）。

A 网络   B 数据库   C 网格计算   D 版本

## 面试笔试题

1、Oracle跟SQL Server 2005的区别？

\*\*1. 最大的区别在于平台，oracle可以运行在不同的平台上，sql server只能运行在windows平台上，由于windows平台的稳定性和安全性影响了sql server的稳定性和安全性

2. oracle使用的脚本语言为PL-SQL，而sql server使用的脚本为T-SQL\*\*

2、对数据库SQL2005、Oracle熟悉吗？

3、数据库DB、数据库系统DBS、数据库管理系统DBMS三者之间的关系是\_\_\_\*\*A\*\*\_\_\_。(选择1项)

A)DBS包括DB和DBMS      B)DBMS包括DB和DBS

C)DB包括DBS和DBMS      D)DBS就是DB，也就是DBMS

# Excise 1.2

## 理论练习

1、  完全卸载Oracle 11g时，需要进行的第一步操作是（ \*\*A\*\* ）。

A 停止所有的Oracle服务     B 启动Oracle的卸载向导

C 删除磁盘上的Oracle文件   D 删除数据库Orcl

2、  安装Oracle数据库过程中SID指的是什么（ \*\*B\*\* ）。

A 系统标识号   B 数据库名   C 用户名   D 用户口令

## 上机练习

1、  下载Oracle安装包，在本机安装Oracle数据库

\*\*登录U+账号后，点击[此处](https://eec-cn.com/batchDownStu/L2F0dGFjaG1lbnRzL2JhdGNoRG93bmxvYWRDb3Vyc2VNYXRlcmlhbHMvc3R1ZGVudD9pZHM9ZmY4MDgwODE3MjQxYzQyZjAxNzI0YmZjODAwYzA3ZDQ=/share/0)下载。\*\*

2、  在本机卸载Oracle数据库

\*\*1.  停用oracle服务：进入计算机管理，在服务中，找到oracle开头的所有服务，右击选择停止

2.  在开始菜单中，找到Universal Installer，运行Oracle Universal Installer，单击卸载产品

3.  在产品清单窗口中，单击全部展开，除了OraDb11g\_home1外，勾选其他项目，单击删除

4.  按Windows徽标键和R键，打开运行窗口，输入regedit，打开注册表，依次展开HKEY\_LOCAL\_MACHINE\SOFTWARE，找到oracle，删除

5.  依次展开HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services中，删除所有oracle开头的项

6.  依次展开HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application，删除所有oracle开头的项；

7.  重启电脑，删除oracle目录，删除Oracle的安装目录app等\*\*

# Excise 1.3

## 理论练习

1、  Oracle网络配置分为服务器端和客户端，监听程序的配置是在\_\_\*\*服务器端\*\*\_\_\_ 。

2、  监听程序与Net服务名的关系是怎样的？

![](img/p1.3.1.png)

\*\*监听程序：Oracle...TNSListener

Net服务名：OracleService...\*\*

## 上机练习

1、完成Oracle数据库监听器及客户端的配置

# Excise 1.4

## 理论练习

1、  要开启Oracle服务器，必须开启的两个服务是（ \*\*D\*\* ）。

A．OracleHOME\_NAMETNSListener和OracleHOME\_NAMEAgent

B．OracleServiceSID和OracleHOME\_NAMEHTTPServer

C．OracleHOME\_NAMEAgent和OracleServiceSID

D．OracleHOME\_NAMETNSListener和OracleServiceSID

## 上机练习

1、 查看Oracle数据库安装后，系统的服务增加了哪些，开启相关服务，启动Oracle数据库

2、 关闭相关服务，关闭Oracle数据库

# Excise 1.5

## 理论练习

1、  Oracle服务器的两个主要组件：\_ \*\*文件\*\*\_、\_\_ \*\*内存\*\*\_\_ 。

2、  实例由\_\_\*\*Oracle DB\*\*\_\_\_ 和\_\_\_\*\*Oralce Server\*\*\_\_ 组成。

3、  Oracle的物理结构包括\_\_\_\*\*数据文件\*\*\_\_\_、\_\_\_\*\*控制文件\*\*\_\_\_、\_\_\_\*\*日志文件\*\*\_\_\_。

4、  当Oracle服务器启动时，下列哪种文件不是必须的（ \*\*D\*\* ）。

A 数据文件   B 控制文件   C 日志文件   D 归档日志文件

5、  在Oracle中，一个用户拥有的所有数据库对象统称为（ \*\*B\*\* ）。

A 数据库   B 模式   C 表空间   D 实例

6、  在Oracle数据库的逻辑结构中有以下组件：\*\*ABCD\*\*

A 表空间   B 数据库   C 区   D 段

这些组件从大到小依次是（ \*\*B-A-D-C\*\* ）

A  A->B->C->D   B  A->D->C->B   C  A->C->B->D   D  D->A->C->B

7、  在Windows操作系统中，Oracle的（ \*\*B\*\* ）服务器监听并接受来自客户端应用程序的连接请求。

A  OracleHOME\_NAMETNSListener   B  OracleServiceSID

C  OracleHOME\_NAMEAgent            D  OracleHOME\_NAMEHTTPServer

8、  关于模式的描述下列哪一项不正确？（ \*\*C\*\* ）

A 表或索引等模式对象一定属于某一个模式

B 在Oracle数据库中，模式与数据库用户是一一对应的

C 一个表可以属于多个模式

D 一个模式可以拥有多个表

9、判断：Oracle数据库的表空间内存的逻辑对象为段，段由盘区组成，而盘区又由多个数据块组成。（ \*\*对\*\* ）

## 面试笔试题

1、介绍一下oracle的体系结构？

![](img/p1.5.1.png)

2、下列不属于Oracle逻辑结构的是（ \*\*C\*\* ）。

   A 区   B 段   C 数据文件   D 表空间

3、Oracle中有哪几种文件？

\*\*数据文件、控制文件、日志文件\*\*

4、解释data block , extent , segment和tablespace的区别？

@import "img\数据库存储总图.png"

\*\*block：块，最小的数据管理单位\*\*

\*\*extent：区：存储分配的最小单位\*\*

\*\*segment：段，一个或多个连续的区组成\*\*

\*\*tablespace：表空间\*\*

\*\*数据库由一个或多个表空间组成

表空间由一个或多个数据文件组成，一个表空间包含多个段

段由一个或多个区组成

区是数据文件中一个连续的分配空间，由一个或多个块组成

块是数据库中最小、最基本的单位，是数据库使用的最小的I/O单元

每个用户都有一个对应的方案\*\*

 # Excise 1.6

 ## 上机练习

1、 使用Oracle 11g特有的企业管理器连接Oracle数据库

2、 使用Oracle自带的客户端工具sqlplus连接Oracle数据库

@import "img/p1.6.2.png"

3、 安装PLSQL Developer图形化工具，并使用该客户端工具连接Oracle数据库

@import "img/p1.6.3.png"

 # Excise 2.1

 ## 理论练习

1、  判断：表空间是Oracle最大的逻辑组成部分。Oracle数据库由一个或多个表空间组成。一个表空间由一个或多个数据文件组成，但一个数据文件只能属于一个表空间。（ \*\*错\*\* ）\*\*数据库\*\*

2、   判断：表空间分为永久表空间和临时表空间两种类型。（ \*\*错\*\* ）\*\*系统/非系统\*\*

3、   判断：Oracle数据库的逻辑存储结构主要由表空间构成。（ \*\*？\*\* ）

## 上机练习

1、 在Web版企业管理器中，创建名称为mytablespace的表空间。为表空间增加一个名为datafile01的数据文件，大小为100M。然后编辑该表空间，为其添加一个datafile02的数据文件，大小为50M。

2、 在PLSQL Developer中创建用户myname,密码为123456，将其默认的表空间修改为mytablespace。

@import "code\2.1.1.ans.sql"

3、 在sqlplus中创建名称为user\_data的表空间，为表空间增加一个名为user\_data的数据文件，大小为50M；创建名称为user\_temp的临时表空间，并增加一个名为user\_temp的数据文件，大小为50M。

删除创建成功的表空间与临时表空间。

@import "code\2.1.2.ans.sql"

面试笔试题

1、在Oracle中创建用户时，若未提及DEFAULT TABLESPACE关键字，则Oracle就将\_\_\_\*\*C\*\*\_\_\_

表空间分配给用户作为默认表空间。

A)HR     B)SCOTT     C)SYSTME     D)SYS

# Excise 2.2

## 理论练习

 1、  下面哪个用户不是Oracle缺省安装后就存在的用户（ \*\*A\*\* ）。

A  SYSDBA     B  SYSTEM

C  SCOTT      D  SYS

## 上机练习

1、 在SQL\*PLUS下创建用户qst，密码为123456，默认表空间为user\_data，临时表空间为user\_temp；

修改用户qst密码为123；

锁定和解锁用户qst；

删除用户qst

@import "code/2.2.2.sql"

2、 在PLSQL Developer中创建用户qst1，密码为123456，默认表空间为user\_data，临时表空间为user\_temp；

修改用户qst1密码为123；

锁定和解锁用户qst1；

删除用户qst1

@import "code/2.2.2.ans.sql"

面试笔试题

1、  Oracle有哪些用户?其默认口令和权限是什么?

|sys|system|scott|

|---|---|---|

|manager|manager|tiger|

# Excise 2.3

## 理论练习

1、 判断：DBA角色具有全部系统权限。（ \*\*错\*\* ）

2、 判断：每一个数据库用户必须具有RESOURCE角色才能登录到Oracle数据库中，所以RESOURCE角色用户是权限最低的用户。（ \*\*错\*\* ）

## 上机练习

1、  在SQL\*PLUS下为用户qst赋予连接数据库的权限，以及访问scott中emp表的权限；

@import "code/2.3.1.ans.sql"

2、  在SQL\*PLUS下对用户qst回收连接数据库的权限，以及访问scott中emp表的权限；

@import "code/2.3.2.ans.sql"

3、  在SQL\*PLUS下为用户qst赋予连接并访问数据库的角色：connect，resource；

@import "code/2.3.3.ans.sql"

4、  在SQL\*PLUS下实现权限的传递：scott用户中对emp表的访问权限通过权限传递授予qst，用户qst再把该权限授予qst1；

@import "code/2.3.4.ans.sql"

5、  在SQL\*PLUS下创建角色、删除角色；

@import "code/2.3.5.ans.sql"

6、  在PLSQL Developer中为用户qst1赋予连接并访问数据库的角色：connect，resource

```sql

SQL> grant connect,resource to qst1;

Grant succeeded

```

## 面试笔试题

1、下面哪个操作会导致用户连接到Oracle数据库，但不能创建表（ \*\*A\*\* ）。

A 授予了CONNECT的角色，但没有授予RESOURCE的角色

B 没有授予用户系统管理员的角色

C 数据库实例没有启动

D 数据库监听没有启动

2、授予sa用户在SCOTT.EMP表中SAL列的更新权限的语句是（ \*\*C\*\* ）。

A GRANT CHANGE ON SCOTT.EMP TO SA

B GRANT UPDATE ON SCOTT.EMP(SAL) TO SA

C GRANT UPDATE (SAL) ON SCOTT.EMP TO SA

D GRANT MODIFY ON SCOTT.EMP(SAL) TO SA

# Excise 3.1

## 理论练习

1、      SQL缩写的含义是\_\_\*\*结构化查询语言，\_Structured\_Query\_Language\*\*\_\_\_。

2、      SQL语言的组成包括\_\_\_\*\*数据定义语言（DDL）\*\* \_\_\_\_\_、\_\_\_\*\*数据操作语言（DML）\*\*\_\_\_、\_\_\_\*\*数据控制语言（DCL）\*\*\_\_\_、\_\_\_\*\*事务控制语言（TCL）\*\*\_\_\_四种。

3、      简述DDL、DML、DCL、TCL缩写分别代表的含义是什么，其包含的SQL语句分别有哪些？

\*\*数据定义语言（DDL，Data Definition Language）:创建或删除表格。也可以定义索引（键），规定表之间的链接，以及施加表间的约束。\*\*

\*\*CREATE/ALTER DATABASE、CREATE/DROP/ALTER TABLE、CREATE/DROP INDEX\*\*

\*\*数据操作语言（DML，Data Manipulation Languag）：供用户实现对数据的增删改查等操作。\*\*

\*\*SELECT、UPDATE、DELETE、INSERT INTO\*\*

\*\*数据控制语言（DCL）:\*\*

\*\*事务控制语言（TCL）:\*\*

4、常用的伪列有哪两个，他们之间的区别是怎样的？

## 面试笔试题

1、 用来存放可变长度字符串的类型是：\_\_\_\_\_\_\_\*\*VARCHA2\*\*\_\_\_。

2、 关于类型定义Number(9,2)说法正确的有\_\_\_\*\*B\*\*\_\_\_。

A)整数部分9位，小数部分2位，共11位   B)整数部分7位，小数部分2位，共9位

C)整数部分6位，小数点一位，小数部分2位，共9位   D)以上说法均不正确

3、 在PL/SQL块中定义一个名为PI值为3.14的常量的语法是\_\_\_\_\*\*D\*\*\_\_\_。

A)PI Const number＝3.14;      B)PI Real number ＝3.14;

C)Constant PI number:＝3.14   D)PI Constant number:=3.14

4、 \_\_\_\*\*C\*\*\_\_consta\_是指组成表的各列的名称及数据类型，也就是日常表格的“栏目信息”。

A)表型     B)记录     C)字段     D)关键字

# Excise 3.2

## 理论练习

1、 数据定义语言，缩写为\_\_\_\_\*\*DDL\*\*\_\_\_\_ ，用于改变数据库结构，包括创建、\_\_\_\_\_\*\*删除\*\*\_\_\_\_ 和 \_\_\_\_\_ \*\*更改\*\*\_\_\_\_数据库对象。

2、 用于操纵表结构的数据定义语言命令有哪些？

\*\*create/drop/alter/truncate\*\*

3、 Truncate与Drop的区别是什么？

\*\*TRUNCATE TABLE 删除表中的所有行，但表结构及其列、约束、索引等保持不变。DROP删除表定义及其数据。\*\*

## 上机练习

1、 某学校要建立一个学生选课系统，分为以下三张表：

1.s学生表：

|字段名称    |      数据类型        |是否可为空|

| :---: | :---: | :---: |

|学号（PK）sno       | varchar2(10)      |      否|

|学生姓名sname       | varchar2(20)      |      是|

2.c课程表：

|字段名称         |     数据类型          |是否可为空  |

| :---: | :---: | :---: |

|课程号（PK）cno   |    varchar2(5)    |         否|

|课程名称cname    |    varchar2(20)   |          是|

|任课老师cteacher |       varchar2(20)|          是|

3.sc学生选课表

|字段名称        |      数据类型         | 是否可为空 |

| :---: | :---: | :---: |

|学号sno(FK)   |       varchar2(10)     |        |

|课程号cno(FK)  |      varchar2(5)      |        |

|成绩scgrade   |         varchar2(20)    |    否 |

 ①请依次创建这三张表。

```sql

SQL>

SQL> create table s(

  2      sno varchar2(10) primary key,

  3      sname varchar2(20)

  4  );

Table created

SQL> create table c(

  2      cno varchar2(5) primary key,

  3      cname varchar2(20),

  4      cteacher varchar2(20)

  5  );

Table created

SQL> create table sc(

  2      sno varchar2(10),

  3      cno varchar2(5),

  4      scgrade varchar2(20) not null,

  5      foreign key(sno) references s(sno),

  6      foreign key(cno) references c(cno)

  7  );

Table created

```

②修改表s，为其增加一列stel（电话）；

```sql

SQL>  alter table s add tel varchar2(11);

Table altered

```

③修改表c，将其“课程名称”列的数据类型长度增加一倍；

```sql

SQL> alter table c modify cname varchar2(40);

Table altered

```

④修改表sc，将其“成绩”列名改为“总分”；

@import "code\3.2.4.sql"

⑤删除表c;

@import "code\3.2.5.sql"

⑥重命名表s为ss;

@import "code\3.2.6.sql"

## 面试笔试题

1、 表userInfo中有三个字段（userID、userName、userAddress），现在要删除字段 userAddress，正确的命令是\_\_\_\_\*\*B\*\*\_\_\_。

A)UPDATE  userInfo  DROP  COLUMN  userAddress

B)ALTER  TABLE  userInfo  DROP  COLUMN  userAddress

C)ALTER  userInfo  DROP  COLUMN  userAddress

D)ALTER  TABLE  userInfo  DELETE  COLUMN  userAddress

2、要将表userInfo从数据库中删除，所用的命令是：\_\_\_\_\_\_\*\*drop table userInfo;\*\*\_\_\_\_\_\_\_\_\_。

3、SQL语言中，用来创建、修改及删除数据库对象的语言被称为：\_\_\_\_\_\_\*\*数据库操作语言\*\*\_\_\_\_\_\_\_。

4、比较truncate、delete和drop命令。

\*\*TRUNCATE TABLE 删除表中的所有行，但表结构及其列、约束、索引等保持不变。\*\*

\*\*新行标识所用的计数值重置为该列的种子。如果想保留标识计数值，请改用 DELETE。\*\*

\*\*如果要删除表定义及其数据，请使用 DROP TABLE 语句。\*\*

# Excise 3.3

## 理论练习

1、  根据数据完整性机制所作用的数据库对象和范围不同，数据完整性可分为\_\_\_\_ \*\*实体完整性\*\* \_\_\_\_、\_\_\_\_\_\_ \*\*域完整性\*\* \_\_\_\_\_\_、 \_\_\_\_ \*\*参照完整性\*\* \_\_\_\_ 、\_\_\_\_\_\_\*\*用户自定义完整性\*\*\_\_\_\_四种类型。

2、  对实体完整性进行约束的方法有\_\_\_\*\*非空约束\*\*\_\_\_、\_\_\_\*\*唯一约束\*\*\_\_\_。

3、  对引用完整性进行约束的方法可以是\_\_\_\_ \*\*添加外键约束\*\*\_\_\_\_。

4、  在数据完整性当中，\_\_\_\_\*\*域完整性\*\*\_\_\_\_可以用检查约束、非空约束，或者默认值方法进行约束。

5、  在数据完整性当中，可以用触发器进行约束的是\_\_\_\_\_\_\_\_。

6、  常见的五大约束分别为\_\_\_\_\*\*主键约束\*\*\_\_\_、\_\_\_\_\_\*\*非空约束\*\*\_\_\_、\_\_\_\*\*唯一约束\*\*\_\_\_\_\_、\_\_\*\*检查约束\*\*\_\_\_、默认约束。

##  上机练习

1、  结合视频和综合设计题1，完成表的创建，并添加相应的约束。

2、  在3.2章节“上机练习”中，为表s的sno，表c的cno添加主键约束；为表sc中的sno和cno添加外键约束，scgrade列添加非空约束；为 表s中新添加的的stel列添加唯一约束。

```sql

SQL> alter table s add constraint PK\_s\_sno primary key(sno);

Table altered

SQL> alter table c add constraint PK\_c\_cno primary key(cno);

Table altered

SQL> alter table sc add constraint FK\_sc\_sno foreign key(sno) references s(sno);

Table altered

SQL> alter table sc add constraint FK\_sc\_cno foreign key(cno) references c(cno);

Table altered

SQL> alter table sc modify scgrade constraint NN\_sc\_scgrade not null;

Table altered

SQL> alter table s add constraint UN\_s\_tel unique(tel);

Table altered

```

3、  新创建一张带有约束的雇员表emptest，表结构如下：

|属性 |    数据类型  |  相应约束|

| :---: | :---: | :---: |

|员工编号 |number(3)  |  主键|

|姓名   |  varchar2(20) |  非空|

|性别    | varchar2(2)  |  默认为“男”，且性别只能为“男”或“女”|

|生日   |  date|

请问该如何创建这张表？

@import "code\3.3.3.sql"

## 面试笔试题

 1、唯一约束与主键约束的一个区别是\_\_\_\*\*D\*\*\_\_\_\_？

A)唯一约束的列的值不可以有重复值     B)唯一约束的列的值可以不是唯一的

C)唯一约束的列不可以为空值         D)唯一约束的列可以为空值

2、在关系数据库中，建立数据库表时，将年龄字段值限制在12～40岁之间的这种约束属于\_\_\_\*\*B\*\*\_\_\_。

A)视图完整性约束   B)域完整性约束    C)参照完整性约束    D)实体完整性约束

3、假设需要为表 customer 添加主键约束,主键列为 customer\_id，可以使用如下\_\_\_\_\*\*BD\*\*\_\_\_\_

方式。(选择2项)

A)Alter table CUSTOMER ADD pk\_customer primary key ("customer\_id");

B)Alter table CUSTOMER ADD primary key ("customer\_id");

C)Alter table CUSTOMER ADD constraint pk\_customer ("customer\_id");

D)Alter table CUSTOMER ADD constraint pk\_customer primary key ("customer\_id");

4、Oracle会为\_\_\_\_\_\_\*\*主键\*\*\_\_\_\_\_\_\_\_约束自动建立索引。

5、 已知关系：厂商（厂商号，厂名）PK=厂商号

产品（产品号，颜色，厂商号）PK=产品号，FK=厂商号，表如下：

|厂商号|厂名|

| :---: | :---: |

|C01|宏达|

|C02|利仁|

|C03|广源|

|产品号|颜色|厂商号|

| :---: | :---: | :---: |

|P01|红|C01|

|P02|黄|C03|

若再往产品表中插入如下记录：

I（P03，红，C02）

II（P01，蓝，C01）

III（P04，白，C04）

IV（P05，黑，null）

能够插入的记录是\_\_\_\*\*D\*\*\_\_\_\_。

A)I，II，IV    B)I，III    C)I，II    D)I，IV

6、在以下哪几种情况下使用唯一约束而不使用主键约束：\_\_\_\*\*AB\*\*\_\_\_\_ (选择2项)

A)列或几个列的组合不是主键    B)列的值允许为空值

C)列有外键引用          D)列的值不允许为空值

# Excise 3.4

## 理论练习

1、  数据操纵语言，缩写为\_\_\_\_\*\*DML\*\*\_\_\_\_\_，主要用于\_\_\_\*\*增加\*\*\_\_\_\_\_、\_\_\_\_\*\*删除\*\*\_\_\_\_和\_\_\_\_\*\*更改\*\*\_\_\_\_数据。

2、  \_\_\_\_\_\_\*\*查询\*\*\_\_\_\_\_是最常见的SQL命令。

3、  数据操纵语言命令包括\_\_\_\*\*INSERT\*\*\_\_\_ 、\_\_ \*\*DELETE\*\* \_\_、\_\*\*UPDATE\*\*\_\_、\_\_\_。

## 上机练习

1、 在3.2章节的“上机练习”中，向s表中插入数据。插入内容如下：

|学号    |姓名  |  电话|

| :---: | :---: | :---: |

|10001   |王晓明 |  13112341118|

|10002   |王涵  |  13212341118|

|10003   |李有才 |  13312341118|

|10004   |张小小 |  13412341118|

|10005   |董强  |  13512341118|

```sql

SQL> insert into s values(10001,'王晓明',1312341118);

1 row inserted

SQL> insert into s values(10002,'王涵',1312341118);

1 row inserted

SQL> insert into s values(10003,'李有才',1312341118);

1 row inserted

SQL> insert into s values(10004,'张小小',1312341118);

1 row inserted

SQL> insert into s values(10005,'董强',1312341118);

1 row inserted

```

2、 生成一个新的表s1，表结构与s表基本一致，有“学号”和“姓名”两列，表中内容如下：

|学号   | 姓名   |

| :---: | :---: | :---: |

|10001   |王晓明 |

|10002   |王涵  |

要求用旧表s生成新表s1去实现。

    从旧表变成新表，需要从三列变两列。旧表非空，所以需要先使用truncate清空表。删除的列无外部约束，可以直接删除。

```sql

SQL> create table s1 as select \* from s;

Table created

SQL> truncate table s1;

Table truncated

SQL> alter table s1 drop column tel;

Table altered

-- ORA-12988: 无法删除属于 SYS 的表中的列(可能报错，换个用户就好了)

```

3、 修改s表中的数据，将董强的电话改为13612341118。

```sql

SQL> update s set tel='13612341118' where SNAME='董强';

1 row updated

```

4、 查询修改后的s表中的全部信息。

```sql

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10001      王晓明               1312341118

10002      王涵                 1312341118

10003      李有才               1312341118

10004      张小小               1312341118

10005      董强                 13612341118

```

5、 删除s1表中的学号列。

```sql

SQL> alter table s1 drop column sno;

Table altered

```

## 面试笔试题

1、 比较Truncate和Delete命令。

\*\*delete：删除表中的某些记录

trancate：删除表中的所有记录，仅保留表结构。\*\*

2、 简述oracle中，dml、ddl、dcl的使用。

\*\*DML：SELECT、INSERT、UPDATE、DELETE\*\*

\*\*DDL：CREATE TABLE、 ALTER TABLE、DROP TABLE、CREATE INDEX、DROP INDEX\*\*

\*\*DCL ：ALTER、GRANT、REVOKE、CREATE\*\*

3、下列哪些语句属于DML语句\_\_\_\*\*AD\*\*\_\_\_\_？(选择2项)

A)select count(\*) from dba\_tables;  B)create table test (a number);

C)alter table test nologging;    D)delete from test;

4、UPDATE-SQL语句的功能是\_\_\_\*\*D\*\*\_\_\_\_。

A)属于数据定义功能             B)属于数据查询功能

C)可以修改表中某些列的属性         D)可以修改表中某些列的内容

5、使用SQL命令将STUDENT中的学生年龄AGE字段值增加1岁，应该使用的命令是\_\_\_\*\*D\*\*\_\_\_\_。

A)REPLACE AGE WITH AGE+1       B)UPDATE STUDENT AGE WITH AGE+1

C)UPDATE SET AGE WITH AGE+1      D)UPDATE STUDENT SET AGE=AGE+1

6、要删除表userInfo中的所有记录，用\_\_\_\_\_\_\_\_\_\*\*trancate table userInfo;\*\*\_\_\_\_\_\_和\_\_\_\_\_\_\_\*\*delete \* from userInto;\*\*\_\_\_\_\_\_\_\_两个命令实现？

# Excise 3.5

## 理论练习

1、  为用户提供权限控制命令的语言是\_\_\_\*\*数据控制语言\*\*\_\_\_\_\_，其缩写为\_\_\_\_\*\*DCL\*\* \_\_\_\_\_。

2、  用于权限控制的命令有\_\_\_\_\*\*REVOKR\*\*\_\_\_\_和\_\_\_\_\*\*GRANT\*\*\_\_\_\_。

## 上机练习

1、  假设用户student2创建了表s，表s的表结构和数据参见3.3、3.5章节。现要求把查询、修改表s的权限赋予普通用户student1，该如何操作？

切换用户sys,

```sql

grant select,update on s to studente1;

```

2、  收回student1对表s进行修改的权限，该如何操作？

```sql

revoke select,update on s from studente1;

```

## 面试笔试题

1、REVOKE 是属于\_\_\*\*C\*\*\_\_\_语句？

A)DQL     B)DML     C)DDL     D)DCL

2、GRANT 语句是\_\_\_\*\*C\*\*\_\_语句？

A)DQL     B)DML     C)DDL     D)DCL

# Excise 3.6

## 理论练习

1、  \_\_\_\_\_\*\*事务\*\*\_\_\_\_是指作为一个逻辑单元整体进行工作的一系列操作。

2、  用于事务控制的语句有\_\_\_\*\*BEGIN\*\*\_\_\_\_\_\_ 、\_\_\_\_ \*\*ROLLBACK\*\* \_\_\_\_\_ 、\_\_\_\_\*\*COMMIT\*\*\_\_\_\_\_。

3、  事务应该具有四个属性，分别为\_\_\_\*\*原子性\*\*\_\_\_\_ 、 \_\_\_\_ \*\*一致性\*\*\_\_\_ 、 \_\_\_ \*\*隔离性\*\* \_\_\_\_ 、 \_\_\_ \*\*持续性\*\*\_\_\_\_。这四个属性通常称为事务的 \_\_\_\_ \*\*四大\*\* \_\_\_\_特性。

## 上机练习

1、 向3.2章节“上机练习”中的s表添加两行数据，添加后设置保存点p1，应该如何操作？添加的数据内容如下

|学号     |姓名     |电话 |

| --- | --- |--- |

|10010   | 孙燕姿   |18600001111 |

|10020   | 周杰伦   |18900002222|

```sql

SQL> insert into s values(10010,'孙燕姿','18600001111');

1 row inserted

SQL> insert into s values(10020,'周杰伦','18900002222');

1 row inserted

SQL> savepoint p1;

Savepoint created

```

2、 更新学号为10020的电话为18000000000，更新后设置保存点p2，应该如何操作？

```sql

SQL>  update s set tel='18000000000' where sno='10020';

1 row updated

SQL> savepoint p2;

Savepoint created

```

3、 删除学号为10010的学生记录，删除后设置保存点p3，应该如何操作？

```sql

SQL>  delete from s where sno='10010';

1 row deleted

SQL>  savepoint p3;

Savepoint created

```

  4、 分别回滚到p1,p2,p3的状态，查看表s中数据的变化；

    错了，应该是分别回到p3,p2,p1的状态。这是p1,p2,p3的情况：因为回到了p1，所以p2和p3就不存在了。

    报错：  ORA-01086: 从未在此会话中创建保存点 'P2' 或者该保存点无效

```sql

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10020      周杰伦               18000000000

SQL> rollback to p3;

Rollback complete

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10020      周杰伦               18000000000

SQL> rollback to p2;

Rollback complete

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10010      孙燕姿               18600001111

10020      周杰伦               18000000000

SQL> rollback to p1;

Rollback complete

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10010      孙燕姿               18600001111

10020      周杰伦               18900002222

```

5、 最后提交操作，查看表s中数据的变化。

```sql

SQL> commit;

Commit complete

SQL> select \* from s;

SNO        SNAME                TEL

---------- -------------------- -----------

10010      孙燕姿               18600001111

10020      周杰伦               18900002222

```

## 面试笔试题

1、\_\_\_\_\_\_\_\_包含了一组数据库操作命令,并且所有的命令作为一个整体一起向系统提交或撤消操作请求。

# Excise 4.1

## 理论练习

1、在属性前添加\_\_\_\_\_\*\*unique\*\*\_\_\_\_\_关键字可以保证表中该属性下没有重复的字段。

2、对查询结果进行排序的关键字是\_\_\_\_\_\*\*order by\*\*\_\_\_\_\_。

3、模糊查询中，’%’和’\_’的区别是什么？

\*\*%:代表一或多个字符

\_:代表一个字符\*\*

## 上机练习

1、 下面的语句是否可以执行成功？

select ename , job , sal as salary from emp; \*\*对\*\*

2、 下面的语句是否可以执行成功？

select \* from emp; \*\*对\*\*

3、 找出下面语句中的错误：

select empno , ename sal \* 12 ANNUAL SALARY from emp

\*\*改：

select empno , ename,sal\*12 "ANNUAL SALARY" from emp;\*\*

4、 显示表dept的结构，并查询其中的全部数据。

@import "code/4.1.4.sql"

5、 显示出表emp中的不重复的岗位job。

@import "code/4.1.5.sql"

 6、 选择员工姓名的第三个字母是A的员工姓名。

@import "code/4.1.6.sql"

7、 选择姓名中有字母A和E的员工姓名。

@import "code/4.1.7.sql"

## 面试笔试题

1、在表authors中查找以 ean 结尾的所有4个字母的作者所在的行。

请选择以下正确的语句\_\_\_\_\*\*A\*\*\_\_\_。

A)SELECT \* FROM authors  WHERE au\_fname LIKE '\_ean'

B)SELECT \* FROM authors  WHERE au\_fname LIKE '%ean'

C)SELECT \* FROM authors  WHERE au\_fname LIKE '[\_ean]'

D)SELECT \* FROM authors  WHERE au\_fname LIKE '[%]ean'

2、下列的哪个子句在SELECT语句中用于排序结果集\_\_\_\*\*D\*\*\_\_\_\_？

A)Having子句   B)Where子句   C)From子句   D)Order by子句

3、使用SQL语句从表STUDENT中查询所有姓王的同学的信息，正确的命令是\_\_\_\*\*A\*\*\_\_\_\_。

A)SELECT \* FROM STUDENT WHERE LEFT(姓名，2)＝’王’

B)SELECT \* FROM STUDENT WHERE RIGHT(姓名，2)＝’王’

C)SELECT \* FROM STUDENT WHERE TRIM(姓名，2)＝’王’

D)SELECT \* FROM STUDENT WHERE STR(姓名，2)＝’王’

4、\_\_\_\*\*unique\*\*\_\_\_\_\_能够去掉查询中重复的行。

5、‘\_P%’表示\_\_\_\*\*C\*\*\_\_\_\_。

A)以P开头     B)以P结束    C)第二个字母是P    D)右面第二个字母是P。

6、在表emp中查找字段empno中以两个数字开头第三个字符是下画线“\_”的所有记录。

请选择以下正确的语句\_\_\_\*\*D\*\*\_\_\_\_：

A)SELECT \* FROM employee  WHERE empid LIKE '[0-9][0-9]\_%'

B)SELECT \* FROM employee  WHERE empid LIKE '[0-9][0-9]\_[%]'

C)SELECT \* FROM employee  WHERE empid LIKE '[0-9]9[\_]%'

D)SELECT \* FROM employee  WHERE empid LIKE '[0-9][0-9][\_]%'

# Excise 4.2

## 上机练习

1、 对scott用户下的表进行多表链接查询：

①   在一张表中，显示所有员工的姓名ename，部门号deptno和部门名称dname。

@import "code\4.2.1.sql"

②   查询20号部门员工的job和20号部门的loc。

@import "code\4.2.2.sql"

③   选择所有有奖金comm的员工的ename , dname , loc。

@import "code\4.2.3.sql"

④   选择在DALLAS工作的员工的ename , job , deptno, dname。

@import "code\4.2.4.sql"

⑤   选择所有员工的姓名ename，员工号deptno，以及他的管理者mgr的姓名ename和员工号deptno，结果类似于下面的格式

| employees | Emp |manager | Mgr|

| --- | --- | --- | --- |

|SMITH|7369|FORD|7902|

@import "code\4.2.5.sql"

⑥   查询各部门员工姓名和他们所在位置，结果类似于下面的格式

|Deptno|Ename|Loc|

| --- | ---  | --- |

|20|SMITH|DALLAS|

@import "code\4.2.6.sql"

# Excise 4.3

理论练习

1、 聚合函数中，表示最小值的函数是\_\_\_\_ \*\*MIN()\*\* \_\_\_\_ ，表示最大值的函数是\_\_\_\_\*\*MAX()\*\*\_\_\_\_\_ ，求和函数是\_\_\_\*\*SUM()\*\*\_\_\_\_\_ ，求平均值的函数是\_\_\_\_\*\*AVG()\*\*\_\_\_\_ ，用来计数的函数是\_\_\_\*\*COUNT()\*\*\_\_\_\_\_ 。

2、 对记录的分组是通过关键字\_\_\_\_\_\*\*GROUP BY\*\*\_\_\_\_\_实现的。

3、 分组查询时，用于过滤掉不符合条件的组的子句是\_\_\_\_\*\*HAVING\*\*\_\_\_\_。

## 上机练习

1、  基于scott.emp表进行一下查询：

①  查询公司员工工资的最大值，最小值，平均值，总和

@import "code\4.3.1.sql"

②  查询各种job的员工工资的最大值，最小值，平均值，总和

@import "code\4.3.2.sql"

③  查询各种job的员工人数(提示:对job进行分组)

@import "code\4.3.3.sql"

④  查询员工最高工资和最低工资的差距（DIFFERENCE）

@import "code\4.3.4.sql"

⑤  查询各个管理者手下员工的最低工资，其中最低工资不能低于800，没有管理者的员工不计算在内

@import "code\4.3.5.sql"

⑥  查询所有部门的名字dname，所在位置loc，员工数量和工资平均值

@import "code\4.3.6.sql"

⑦  查询公司的人数，以及在1980-1987年之间，每年雇用的人数，结果类似下面的格式

|total|1980|1981|1982|1987|

|---|---|---|---|---|

|30|3|4|6|7|

@import "code\4.3.7.sql"

## 面试笔试题

1、\_\_\_\_\*\*GROUP BY\*\*\_\_\_\_子句实现对一个结果集进行分组和汇总。

2、假定Emp表中共有14行记录，有三条记录Comm字段为NULL，则Select count(\*) from emp和Select count(Comm) from emp分别返回\_\_\_\*\*14\*\*\_\_\_\_\_和\_\_\_\_\*\*11\*\*\_\_\_\_。

3、\_\_\_\_\*\*AVG()\*\*\_\_\_\_函数返回的是满足给定条件的平均值。

4、下面有关HAVING子句描述错误的是\_\_\_\*\*B\*\*\_\_\_\_。

A)HAVING子句必须与GROUPBY子句同时使用，不能单独使用。

B)使用HAVING子句的同时不能使用WHERE子句。

C)使用HAVING子句的同时可以使用WHERE子句。

D)使用HAVING子句的作用是限定分组的条件。

5、\_\_\_\_\*\*COUNT\*\*\_\_\_\_函数返回的是满足给定条件的记录数。

# Excise 4.4

## 上机练习

1、 查询和scott相同部门的员工姓名ename和雇用日期hiredate；

@import "code\4.4.1.sql"

2、 查询工资比公司平均工资高的所有员工的员工号empno，姓名ename和工资sal；

@import "code\4.4.2.sql"

3、 查询和姓名中包含字母u的员工在相同部门的员工的员工号empno和姓名ename；

@import "code\4.4.3.sql"

4、 查询在部门的loc为NewYork的部门工作的员工的员工姓名ename，部门名称dname和岗位名称job；

@import "code\4.4.4.sql"

5、 查询管理者是king的员工姓名ename和工资sal。

@import "code\4.4.5.sql"

## 面试笔试题

1、 rowid，rownum的定义分别是什么？

\*\*rowid 是物理地址，用于定位 oracle 中具体数据的物理存储位置， 查询中不会发生变化

rownum 是根据 sql 查询出的结果给每行分配一个逻辑编号，sql 不同可能会导致 rownum 不同\*\*

# Excise 5.1

## 理论练习

1、  在Oracle中，表示不等于的操作符有两种，分别是\_\_\_<>

\_\_\_\_、\_\_\_!=\_\_\_\_\_。

2、在Oracle中，连接操作符用\_\_\_\_||\_\_\_表示。

3、在Oracle中，集合操作符有四种，分别为\_\_\_\*\*MINUS\*\* \_\_\_\_ 、\_\_\_\_  \*\*INTERSECT\*\* \_\_\_\_、 \_\_\_ \*\*UNION ALL\*\* \_\_\_\_\_\_\_、 \_\_\_ \*\*UNION\*\*\_\_\_\_\_。

## 上机练习

1、  查询工资大于1600的员工姓名和工资；

```sql

select ename, sal from emp  where sal>1600;

```

2、  选择工资不在4000到5000的员工的姓名和工资；

```sql

select ename,sal  from emp  where sal  not between 4000 and 5000;

```

3、  选择雇用时间在1981-2-19到1981-12-3之间的员工姓名，JOB和雇用时间HIREDATE；

```sql

Select ename, job, hiredate

from emp  where hiredate

between to\_date('1981-2-19','YYYY-mm-dd')

and  to\_date('1981-12-3','YYYY-mm-dd');

```

4、  选择在20和30号部门工作的员工姓名和部门号；

```sql

select ename,deptno from emp where deptno in(20,30);

```

5、  选择在1987年雇用的员工的姓名和雇用时间；

```sql

Select ename,hiredate

from emp

where hiredate  between to\_date('1987-1-1','YYYY-mm-dd')

and  to\_date('1987-12-31','YYYY-mm-dd');

```

6、  选择公司中没有管理者的员工姓名及JOB；

```sql

select ename,job

from emp

where mgr is null;

```

7、  选择公司中有奖金的员工姓名，工资和奖金级别；

```sql

select ename,sal,comm

from emp

where comm is not null

and comm>0;

```

8、 连接表emp的全部列，各个列之间用逗号连接，列头显示成OUT\_PUT(提示:使用连接符||)。

```sql

select EMPNO|| ',' || ENAME || ',' || JOB || ',' || MGR || ',' || HIREDATE || ',' || SAL || ',' || COMM || ',' || DEPTNO as "OUT\_PUT" from emp;

```

 ## 面试笔试题

1、  Oracle中字符串用什么符号链接？

\*\*||\*\*

2、设字段变量"工作日期"为日期型，"工资"为数值型，则要表达"工龄大于30年，工资高于1500，且低于1800元"这命令，其表达式是\_\_\_\*\*C\*\*\_\_\_\_。

A)工龄>30.AND.工资>1500.AND.工资<1800

B)工龄>30.AND.工资>1500.OR.工资<1800

C)INT((DATE()-工作日期)/365)>30.AND.工资>1500.AND.工资

D)INT((DATE()－工作日期)/365)>30.AND.(工资>1500.OR.工资<1800)

3、oracle中的操作符包括哪些类型？

\*\*算术，逻辑，比较，集合，连接\*\*

# Excise 5.2

## 理论练习

1、  SQL函数分为单行函数、\_\_\_\_ \*\*分析函数\*\* \_\_\_\_ 、\_\_\_\*\*聚合函数\*\*\_\_\_\_\_三种。

2、  常用的分析函数有\_\_\_\_ \*\*row\_number()\*\* \_\_\_\_ 、\_\_\_\_ \*\*rank()\*\* \_\_\_\_ 、 \_\_\_\_ \*\*dense\_rank()\*\* \_\_\_\_三种。

## 面试笔试题

1、 说说Oracle中经常使用到的函数。

\*\*字符串常用函数、处理数值类型的函数、格式转换函数、日期函数\*\*

2、获得当前系统时间的查询语句是\_\_\_\*\*C\*\*\_\_\_\_。

A)Sysdate               B)Select Sysdate

C)Select Sysdate from dual   D)以上都可以

# Excise 6.1

## 理论练习

1、数据库用来控制共享资源并发访问的机制是\_\_\_\_\*\*数据库并发控制\*\*\_\_\_\_。

# Excise 6.2

## 理论练习

1、  按照锁的粒度进行分类，锁分为\_\_\_\_\*\*表级锁\*\*\_\_\_\_、 \_\_\_\_\*\*行级锁\*\*\_\_\_\_和数据库级锁。

2、  锁的粒度和系统开销成\_\_\_\_\_\*\*反\*\*\_\_\_比例关系。

## 面试笔试题

1、 Oracle的锁有几种，定义分别是什么？

\*\*按照锁的粒度进行分类，锁分为表级锁和行级锁和数据库级锁。\*\*

2、关于For update 子句和For update of 字段说法正确是\_\_\*\*C\*\*\_\_。

A)没有区别

B)前者只锁定当前表中的相关行，而后者锁定所有相关表的行

C)前者锁定所有相关表中的行，而后者锁定当前表的相关行

D)以上说法均不正确

# Excise 6.3

## 理论练习

1、  两个事务相互等待对方释放资源时，就会形成\_\_\_\*\*死锁\*\*\_\_\_\_\_。

## 面试笔试题

1、  解释什么是死锁，如何解决Oracle中的死锁？

 \*\*是指两个会话，每个会话都持有另外一个会话想要的资源，因争夺资源而造成的一种互相等待的现象，此时就会出现死锁，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。Oracle对于“死锁”采取的策略是回滚其中一个事务，让另外一个事务顺利进行。\*\*

2、Oracle中用来释放锁的语句有\_\_\_\*\*A\*\*\_\_\_\_。(选择2项)

A)Rollback     B)Commit     C)Drop Lock     D)Unlock

# Excise 6.4

## 理论练习

1、 Oracle表有哪几种类型？

\*\*1）堆表：heap table ：数据存储时，行是无序的，对它的访问采用全表扫描。

   2）分区表 表>2G

   3）索引组织表（IOT）

   4）簇表

   5）临时表

   6）压缩表

   7）嵌套表\*\*

# Excise 6.5

## 理论练习

1、  什么是表分区技术？

\*\*数据分区是一种物理数据库的设计技术，它的目的是为了在特定的SQL操作中减少数据读写的总量以缩减响应时间。\*\*

\*\*分区并不是生成新的数据表，而是将表的数据均衡分摊到不同的硬盘，系统或是不同服务器存储介子中，实际上还是一张表。\*\*

# Excise 6.6

## 理论练习

1、  常用的分区方法有4中，分别为\_\_\_\*\*范围分区\*\*\_\_\_\_\_、\_\_\_ \*\*哈希分区\*\*\_\_\_\_、\_\_\_\_\*\*间隔分区\*\*\_\_\_\_、\_\_\_\_\*\*列表分区\*\*\_\_\_\_。

2、  依据HASH函数执行操作的表分区技术为\_\_\_\*\*哈希分区\*\*\_\_\_\_\_。

## 面试笔试题

1、Oracle分区是怎样优化数据库的？

\*\*Oracle的分区可以分为：列表分区、范围分区、散列分区、复合分区。

1).  增强可用性：如果表的一个分区由于系统故障而不能使用，表的其余好的分区仍可以使用；

2).  减少关闭时间：如果系统故障只影响表的一部份分区，那么只有这部份分区需要修复，可能比整个大表修复花的时间更少；

3).  维护轻松：如果需要得建表，独产管理每个公区比管理单个大表要轻松得多；

4).  均衡I/O：可以把表的不同分区分配到不同的磁盘来平衡I/O改善性能；

5).  改善性能：对大表的查询、增加、修改等操作可以分解到表的不同分区来并行执行，可使运行速度更快

6).  分区对用户透明，最终用户感觉不到分区的存在。\*\*

# Excise 6.7

## 理论练习

1、  有哪几种常用的分区维护操作？

\*\*添加分区、删除分区、截断分区、合并分区、拆分分区、接合分区、重命名分区\*\*

# Excise 7.1

## 理论练习

1、  Oracle数据库对象又称为\_\_\_\*\*模式对象\*\*\_\_\_\_\_对象。

2、数据库对象是逻辑结构的集合，最基本的数据库对象是\_\_\_\_\*\*表和视图\*\*\_\_\_\_，其他常用的数据库对象包括\_\_\_\*\*约束\*\*\_\_\_、\_\_\_\*\*序列\*\*、\_\_\_、\_\_\_\*\*函数\*\*\_\_\_\_、\_\_\_\*\*存储过程\*\*\_\_\_。

# Excise 7.2

## 理论练习

1、 同义词有两种类型，其中，只能在模式内访问，且不能与当前模式对象同名的同义词称为\_\_\_\_\*\*私有同义词\*\*\_\_\_\_，可被所有的数据库用户访问的是 \_\_\_ \*\*公有同义词\*\*\_\_\_\_\_。

2、 创建同义词的语法关键字是\_\_\_\*\*synonym\*\*\_\_\_\_\_。

# Excise 7.3

## 理论练习

1、 用来生成唯一、连续序号的对象是\_\_\_\*\*序列\*\*\_\_\_\_\_。

2、  创建序列的语法关键字是\_\_\_\_\*\*sequence\*\*\_\_\_\_。

## 面试笔试题

1、在下列各选项中，关于序列的描述哪一项是不正确的\_\_\_\*\*D\*\*\_\_\_？

A)序列是Oracle提供的用于产生一系列唯一数字的数据库对象

B)序列并不占用实际的存储空间

C)使用序列时，需要用到序列的两个伪列NEXTVAL与CURRVAL。其中，NEXTVAL 将返回序列生成的下一个值，而CURRVAL返回序列的当前值

D)在任何时候都可以使用序列的伪列CURRVAL，以返回当前序列值

# Excise 7.4

## 理论练习

1、 视图的本质是一个\_\_\_\_\*\*select\*\*\_\_\_\_语句，被编译后存储在数据字典里。

2、 创建视图所依据的表称为\_\_\_\_\*\*视图\*\*\_\_\_\_。

3、 视图的优点有哪些？

4、 创建视图的语法关键字是\_\_\_\*\*create view\*\*\_\_\_\_\_。

5、 \_\_\_\_\_\*\*with check option\*\*\_\_\_\_\_\_语句可以帮助检查更新视图数据时，是否符合视图查询事实的一些规范。

## 面试笔试题

1、 在视图上使用DML语句有哪些限制？

\*\*a)可以在简单视图中执行 DML操作

b)当视图定义中包含以下元素之一时不能使用delete：

  i.组函数

 ii.GROUP BY子句

iii.DISTINCT关键字

 iv.ROWNUM 伪列 DUAL伪表

c)当视图定义中包含以下元素之一时不能使用update：

  i.组函数

 ii.GROUP BY子句

iii.DISTINCT关键字

 iv.ROWNUM 伪列

  v.列的定义为表达式

d)当视图定义中包含以下元素之一时不能使用insert:

  i.组函数

 ii.GROUP BY子句

iii.DISTINCT关键字

 iv.ROWNUM 伪列

  v.列的定义为表达式

 vi. 中非空的列在视图定义中未包括\*\*

2、 怎样创建一个视图，视图的好处是什么，视图可以控制权限么？

\*\*创建：

create or replace view [viewname] as ...

好处：

1．封装查询

2．灵活的控制安全性

控制权限：

可以\*\*

3、关于创建视图的正确的说法是\_\_\_\*\*BC\*\*\_\_\_\_。（选择2项）

A)只能基于基表创建视图

B)定义视图的查询不能包含Order By, Compute By 和 Into子句

C)不能将默认值、规则和触发器与视图相关联

D)在视图中修改列名会影响基表的列名

# Excise 7.6

## 理论练习

1、  从索引的特性上进行划分，索引类型分为普通索引、\_\_\_\*\*反向索引\*\*\_\_\_\_\_、\_\_\_\*\*唯一索引\*\*\_\_\_\_\_、\_\_\_\_\*\*全文索引\*\*\_\_\_\_、函数索引、位图索引。

2、  创建索引的语法关键字是\_\_\_\*\*create index\*\*\_\_\_\_\_。

3、  \_\_\_\_\_\*\*唯一\*\*\_\_\_索引确保在定义索引的列中没有重复值。

4、  在低基数的列上，以及经常使用聚合函数的操作上创建索引，通常选择创建\_\_\_\_\*\*位图索引\*\*\_\_\_\_。

## 面试笔试题

1、  怎样创建一个索引？索引使用的原则有哪些？有什么优点和缺点？

\*\*创建索引：create index\*\*

\*\*使用原则：\*\*

\*\*1. 最左前缀匹配原则，非常重要的原则，mysql会一直向右匹配直到遇到范围查询(>、<、between、like)就停止匹配，比如a = 1 and b = 2 and c > 3 and d = 4 如果建立(a,b,c,d)顺序的索引，d是用不到索引的，如果建立(a,b,d,c)的索引则都可以用到，a,b,d的顺序可以任意调整；\*\*

\*\*2. =和in可以乱序，比如a = 1 and b = 2 and c = 3 建立(a,b,c)索引可以任意顺序，mysql的查询优化器会帮你优化成索引可以识别的形式；\*\*

\*\*3. 尽量选择区分度高的列作为索引,区分度的公式是count(distinct col)/count(\*)，表示字段不重复的比例，比例越大我们扫描的记录数越少，唯一键的区分度是1，而一些状态、性别字段可能在大数据面前区分度就是0，那可能有人会问，这个比例有什么经验值吗？使用场景不同，这个值也很难确定，一般需要join的字段我们都要求是0.1以上，即平均1条扫描10条记录；\*\*

\*\*4. 索引列不能参与计算，保持列“干净”，比如from\_unixtime(create\_time) = ’2014-05-29’就不能使用到索引，原因很简单，b+树中存的都是数据表中的字段值，但进行检索时，需要把所有元素都应用函数才能比较，显然成本太大。所以语句应该写成create\_time = unix\_timestamp(’2014-05-29’);\*\*

\*\*5. 尽量的扩展索引，不要新建索引。比如表中已经有a的索引，现在要加(a,b)的索引，那么只需要修改原来的索引即可。\*\*

\*\*优点：\*\*

- \*\*通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。\*\*

- \*\*可以大大加快数据的检索速度，这也是创建索引的最主要的原因。\*\*

- \*\*可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。\*\*

- \*\*在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。\*\*

- \*\*通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。\*\*

\*\*缺点：\*\*

- \*\*创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。\*\*

- \*\*索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。\*\*

- \*\*当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。\*\*

# Excise 8.1

## 理论练习

1、 过程语言与结构化查询语言结合而成的编程语言简称为\_\_\_\_ \*\*PL/SQL\*\* \_\_\_\_ 。

2、 \_\_\_\_\*\*PL / SQL块\*\*\_\_\_\_是构成PL/SQL程序的基本单元。

3、 PL/SQL块分为三部分，包括\_\_\_\_ \*\*declare\*\* \_\_\_\_\_\_ 、 \_\_\_\_ \*\*begin\*\* \_\_\_\_\_、 \_\_\_\_ \*\*exception\*\* \_\_\_\_\_。

4、 在PL/SQL中，赋值操作符用\_\_\_\_\_\_:\_\_\_\_表示，’||’表示\_\_\_\*\*连接\*\*\_\_\_\_操作符，单行注释用\_\_\_\_\_#\_\_\_表示，多行注释用\_\_\_\_/\* \*/\_\_\_\_\_表示，标签用\_\_\_\_<< >>\_\_\_\_表示，’..’表示\_\_\_\_\*\*范围\*\*\_\_\_\_操作符。

5、 判断：PL/SQL区分大小写。（\*\*错\*\*）

## 面试笔试题

1、PL/SQL块中不能直接使用的SQL命令是\_\_\_\*\*A\*\*\_\_\_\_。(选择1项)

A)SELECT      B)INSERT      C)UPDATE      D)DROP

# Excise 8.2

## 理论练习

1、  在PL/SQL当中，\_\_\_\_\_\*\*属性类型\*\*\_\_\_\_\_用于引用数据库列的数据类型，以及表示表中一行的记录类型的。

2、  属性类型有两种，其中，引用变量和数据库列的数据类型称为\_\_\_ \*\*列类型\*\* \_\_\_\_\_ ，用\_\_\_\_ \*\*%type\*\* \_\_\_\_ 表示；提供表示表中一行的记录类型称为\_\_\_\_ \*\*行类型\*\* \_\_\_\_ ，用\_\_\_\_\_ \*\*%row type\*\* \_\_\_表示。

3、  使用属性类型的优点有哪些？

\*\*不需要知道被引用的表列的具体类型\*\*

\*\*如果被引用对象的数据类型发生改变， PL/SQL   变量的数据类型也随之改变\*\*

## 面试笔试题

1、声明%TPYE类型的变量时，服务器将会做什么操作\_\_\_\_ \*\*A\*\* \_\_\_？

A)为该变量检索数据库列的数据类型     B)复制一个变量

C)检索数据库中的数据             D)为该变量检索列的数据类型和值

2、定义SQL块%TYPE有什么好处

\*\*使得 PL/SQL 更加灵活，更加适应于对数据库定义的更新\*\*

3、在PL/SQL块中定义一个名为PI值为3.14的Real型常量的语法是\_\_\_\_\*\*D\*\*\_\_\_。

A)Pi Const Real＝3.14;      B)Pi Real Const ＝3.14;

C)Constant Pi Real:＝3.14     D)Pi Constant Real:=3.14

# Excise 8.3

## 理论练习

1、  PL/SQL支持的流程控制结构包括\_\_\_\*\*顺序控制\*\*\_\_\_\_\_、 \_\_\_\_ \*\*条件控制\*\* \_\_\_\_ 、\_\_\_\_\*\*循环控制\*\*\_\_\_\_三种。

2、  在PL/SQL语言当中，条件控制包括\_\_\_\*\*if\*\* \_\_\_\_ 和 \_\_\_\_ \*\*case\*\* \_\_\_\_ 两种语句；循环控制包括 \_\_\_ \*\*loop\*\* \_\_\_\_\_ 、 \_\_\_\_ \*\*for\*\* \_\_\_\_ 、\_\_\_ \*\*while\*\* \_\_\_\_\_ 三种循环结构。

3、  跳转结构语句包括\_\_\_\_  \*\*goto\*\*  \_\_\_\_ 、 \_\_\_\_ \*\*null\*\* \_\_\_\_ 两种，其中，\_\_\_\_  \*\*goto\*\* \_\_\_\_语句实现无条件转到标签指定的语句， \_\_\_ \*\*null\*\* \_\_\_\_语句是什么也不做的空语句。

## 面试笔试题

1、举例说明while..loop/for..loop循环语句,如插入1..10的值到一个单字段数据表(结构自定义)。

\*\*i:=1;

while i<=10 loop

insert into test(id) values(i);

i:=i+1;

end loop;

commit;\*\*

2、判断下面不完整的 loop语句:

LOOP

INSERT INTO inventory (id\_number, description)

VALUES (v\_id\_number, v\_description);

v\_counter := v\_counter +1;

\_\_\_\_\_\_\_\_\*\*END LOOP\*\*\_\_\_\_\_\_\_\_\_命令应被加入，为了有条件的停止 loop 执行?

3、有一段PL/SQL程序如下所示：

```sql

var1 := 10;

LOOP

EXIT WHEN var1 > 12;

var1 := var1 +1;

DBMS\_OUTPUT.PUT\_LINE(‘A’);

END LOOP;

```

输出为\_\_\_\_\_\_\_\_\*\*AAA\*\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

# Excise 8.4

## 理论练习

1、  在PL/SQL中，运行程序时出现的错误称为\_\_\_\*\*异常\*\*\_\_\_\_\_。

2、  PL/SQL的异常有两种类型，其中，当程序违反Oracle规则或者超越系统限制时隐式引发的异常称为\_\_\_\_\*\*预定义异常\*\*\_\_\_\_\_\_；用户在声明部分定义的异常称为 \_\_\_\_\_ \*\*用户定义异常\*\* \_\_\_\_\_ ，该异常可以通过 \_\_\_\_ \*\*RAISE\*\*\_\_\_\_语句显式引发。

3、  表示“除数为0错误”的系统异常的关键字为 \_\_\_ \*\*ZERO\_DIVIDE\*\*\_\_\_\_\_ ，表示“查询多行错误”的系统异常的关键字为 \_\_\_ \*\*No\_rows\_found\*\* \_\_\_\_\_ ，表示“非法数字”的系统异常关键字为\_\_\_\*\*Invalid\_Number\*\* \_\_\_\_\_ ，表示“未查到数据”的异常关键字为\_\_\_ \*\*No\_Data\_rows\_found\*\* \_\_\_\_\_。

## 面试笔试题

1、当Select语句没有返回行时，将引发下列哪个异常\_\_\_\*\*A\*\*\_\_\_\_？

A)No\_rows\_found   B)No\_data\_found   C)No\_Data\_rows\_found    D)Invalid\_Number

2、以零作除数时会引发\_\_\_\*\*B\*\*\_\_\_\_异常。

A)VALUE\_ERROR  B)ZERO\_DIVIDE  C)STORAGE\_ERROR  D)SELF\_IS\_NUL

# Excise 8.5

## 理论练习

1、  PL/SQL语言的优点有哪些？

\*\*1、SQL 的支持

2、面向对象的支持

3、良好的性能

4、高效性

5、可移植性

6、与SQL 紧密结合

7、高度安全\*\*

## 面试笔试题

1、  下面哪个操作会导致用户连接到Oracle数据库，但不能创建表（ \*\*A\*\* ）。

A 授予了CONNECT的角色，但没有授予RESOURCE的角色

B 没有授予用户系统管理员的角色

C 数据库实例没有启动

D 数据库监听没有启动

# Excise 9.1

## 理论练习

1、PL/SQL用\_\_\_\_\*\*游标\*\*\_\_\_\_来管理SQL的SELECT语句。

2、什么是游标？

\*\*L/SQL  是用游标来管理  SQL  的  SELECT  语句的  .  游标是为了处理这些语句而分配的一大块内存  .  它提供了对一个结果集进行逐行处理的能力  ,  可看作是一种特殊的指针  .  它与某个查询结果集相关联  ,  可以指向结果集的任意位置  ,  以便对指定位置的数据进行处理  .  使用它可以在查询数据的同时对数据进行处理\*\*

3、游标与某个查询结果集相关联，可以指向结果集的任意位置，可以看作是一种特殊的\_\_\_\_ \*\*指针\*\* \_\_\_\_。

4、游标的类型分为\_\_\_ \*\*隐式游标\*\* \_\_\_\_ 、 \_\_\_\_ \*\*显式游标\*\* \_\_\_\_ 、 \_\_ \*\*REF 游标\*\* \_\_\_\_\_\_\_三种。

## 面试笔试题

1、如何使用Oracle的游标？

\*\*声明游标

打开游标（将数据存储到游标）

根据游标提取数据

关闭游标\*\*

2、在定义游标时使用的FOR UPDATE子句的作用是\_\_\_\_\*\*C\*\*\_\_\_。

A)执行游标           B)执行SQL语句的UPDATE语句

C)对要更新表的列进行加锁  D)都不对

# Excise 9.2

## 理论练习

1、  自动声明、打开和关闭的游标称为\_\_\_ \*\*隐式游标\*\* \_\_\_\_\_。

2、  什么情况下自动创建隐式游标？

\*\*使用 DML 语句（增删改）和单行查询语句（赋值）时\*\*

3、  隐式游标的属性有哪些？

\*\*%FOUND  –  SQL  语句影响了一行或多行时为  TRUE

%NOTFOUND  –  SQL  语句没有影响任何行时为 TRUE

%ROWCOUNT  –  SQL  语句影响的行数，没有影响任何行，返回 0 ，在执行任何 DML 语句前，值为 NULL

%ISOPEN  -  游标是否打开，隐式游标始终为 FALSE( 隐式游标执行时打开，结束时立即关闭 )\*\*

# Excise 9.3

## 理论练习

1、  由PL/SQL程序员定义和命名的游标称为\_\_\_ \*\*显式游标\*\* \_\_\_\_\_。

2、  需要手动打开，手动关闭的游标是\_\_\_\_ \*\*显式游标\*\*\_\_\_\_。

3、  当用户需要从游标中提取多行记录时，经常使用的游标是\_\_\_\_ \*\*显式游标\*\*\_\_\_\_。

## 面试笔试题

1、  简述使用带参数显式游标的四个步骤。

\*\*声明游标、打开游标、提取游标指向数据给变量、关闭游标\*\*

-->

2、  作家信息表Auths包含两个字段

|Author\_Code|VARCHAR2(50)|

|:---:|:---:|

|Salary|NUMBER(10)|

其中Author\_Code为作家代码，主键。Salary为作家工资额，非空。

现由于工资调整，对于作家代码在‘A00001′和‘A00006′;之间的作家工资少于或等于五百，则给该作家增加15%的工资，对于大于500到1000之间则增加12%，其它则增加10%。如下程序是通过游标方式实现的，请填写程序中相应的空格。

DECLARE

–  声明两个变量，用来接收游标返回的工资和作家代码

v\_Salary \_\_ \*\*NUMBER(10)\*\* \_\_\_\_\_\_

v\_Code \_\_\_ \*\*VARCHAR2(50)\*\*\_\_\_\_\_

v\_ErrText \_\_ \*\*VARHCAR(200)\*\* \_\_ ;

–  声明游标 c\_Salary ，该游标的结果集是 Auths 表中作家代码 (Author\_Code) 值是

– ‘A00001′ 到 ‘A00006′ 的工资值 (Salary ) 和作家代码值 (Author\_Code)

CURSOR c\_Salary IS

SELECT Salary,

Author\_Code

FROM Auths

WHERE Author\_Code >=’A00001′

AND Author\_Code <=’A00006′;

BEGIN

–  打开游标

\_open \_\_ \*\*c\_Salary\*\* ； \_\_\_\_\_

LOOP

– 游标循环，将游标查询结果集中的一行保存到两个临时变量中

\_\_select\_\_\_ \*\*c\_Salary INTO v\_Salary, v\_Code\*\* \_\_\_;

–  当结果集中没有行是退出

EXIT WHEN \_\_\_ \*\*NO\_DATA\_FOUND\*\* \_\_\_\_\_ ;

–  如果该作家的工作少于或等于五百，则给该作家增加 15% 的工资

– 500 到 1000 之间则增加 12%, 其他增加 10%

IF v\_Salary <= 500 THEN

UPDATE Auths

SET Salary = \_ \*\*Salary+Salary\*0.15\*\* \_\_\_\_\_\_\_

WHERE Author\_Code = v\_Code;

ELSIF v\_Salary <= 1000 THEN

UPDATE Auths

SET  Salary = Salary + Salary \* 0.12

WHERE Author\_Code = v\_Code;

\_\_\_ \*\*THEN\*\* \_\_\_\_\_UPDATE Auths

SET Salary = Salary + Salary \* 0.1

WHERE Author\_Code = v\_Code;

END IF;

END LOOP;

\_ \*\*CLOSE \_c\_Salary\*\*\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_;

EXCEPTION

WHEN OTHERS THEN

CLOSE c\_Salary;

ROLLBACK;

v\_E rrText := \_\_\_\_\_\_\_\_;

DBMS\_OUTPUT.PUT\_LINE(‘ 程序异常终止，出现一下错误 :’||v\_ErrText);

END;

<!--

– 声明两个变量，用来接收游标返回的工资和作家代码

v\_Salary \_\_\_\_\_\_\_\_

v\_Code \_\_\_\_\_\_\_\_

v\_ErrText VARHCAR(200);

– 声明游标c\_Salary，该游标的结果集是Auths表中作家代码(Author\_Code)值是

– ‘A00001′到‘A00006′的工资值(Salary)和作家代码值(Author\_Code)

CURSOR c\_Salary IS

SELECT Salary,

Author\_Code

FROM Auths

WHERE Author\_Code >=’A00001′

AND Author\_Code <=’A00006′;

BEGIN

– 打开游标

\_\_\_\_\_\_\_\_

LOOP

–游标循环，将游标查询结果集中的一行保存到两个临时变量中

\_\_\_\_\_\_\_\_c\_Salary INTO v\_Salary, v\_Code;

– 当结果集中没有行是退出

EXIT WHEN \_\_\_\_\_\_\_\_;

– 如果该作家的工作少于或等于五百，则给该作家增加15%的工资

– 500到1000之间则增加12%,其他增加10%

IF v\_Salary <= 500 THEN

UPDATE Auths

SET Salary = \_\_\_\_\_\_\_\_

WHERE Author\_Code = v\_Code;

ELSIF v\_Salary <= 1000 THEN

UPDATE Auths

SET Salary = Salary + Salary \* 0.12

WHERE Author\_Code = v\_Code;

\_\_\_\_\_\_\_\_UPDATE Auths

SET Salary = Salary + Salary \* 0.1

WHERE Author\_Code = v\_Code;

END IF;

END LOOP;

\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_;

EXCEPTION

WHEN OTHERS THEN

CLOSE c\_Salary;

ROLLBACK;

v\_ErrText := \_\_\_\_\_\_\_\_;

DBMS\_OUTPUT.PUT\_LINE(‘程序异常终止，出现一下错误:’||v\_ErrText);

END; -->

# Excise 9.5

## 理论练习

1、  游标变量的优点和限制有哪些？

\*\*游标变量的优点有：

可从不同的  SELECT  语句中提取结果集

可以作为过程的参数进行传递

可以引用游标的所有属性

可以进行赋值运算

使用游标变量的限制：

不能在程序包中声明游标变量

FOR UPDATE 子句不能与游标变量一起使用

不能使用比较运算符\*\*

# Excise 10.1

## 理论练习

1、\_\_\_\_\*\*存储过程\*\*\_\_\_\_是一组为了完成\_\_\_\_\*\*为了完成特定功能的SQL语句集\*\* \_\_\_\_的符合数据库程序脚本规范的程序，经过编译后存储在\_\_\_\*\*数据库\*\*\_\_\_\_中，然后由一个应用程序或者其他PL/SQL程序\_\_\_\*\*执行\*\*\_\_\_\_\_。

2、从根本上讲，过程就是\_\_\_\*\*存储在数据库中的\*\*\_\_\_\_的PL/SQL程序块。

3、创建过程的语法关键字是\_\_\_\_\*\*create procedure\*\*\_\_\_\_\_。

4、创建带有输出参数的过程，参数前面要加参数类型为\_\_\_\_\*\*out\*\*\_\_\_\_。

## 面试笔试题

1、  谈谈存储过程的优缺点。

\*\*优点：

1.执行速度比普通sql快。

2.减少网络传输

3.可维护性更好

4.安全性更强

5.可扩展性更强

缺点:

1.面对复杂的业务逻辑，过程化的处理会很吃力

2.开发调试复杂

3.无法应用缓存。

4.不支持群集\*\*

2、下面哪些关于存储过程的描述是正确的\_\_\_\*\*ABD\*\*\_\_\_\_？(选择3项)

A)存储过程是一组预编译的SQL语句   B)存储过程可加快查询的执行速度

C)不可以在存储过程内引用临时表     D)帮助实现模块化、编程

3、PL/SQL过程的参数模式有\_\_\_\_\*\*ABC\*\*\_\_\_。(选择3项)

A)In       B)Out      C)In Out      D)Ref

4、存储过程中的传出参数使用\_\_\_\_\*\*out/inout\*\*\_\_\_\_关键字。

# Excise 10.2

## 理论练习

1、  有返回值的，命名的PL/SQL子程序称为\_\_\_\_\*\*函数\*\*\_\_\_\_。

2、  定义函数的限制有哪些？

3、  访问函数有哪两种方式，对应的语法是怎样的？

\*\*plsql语句调用 fun()

dual调用：用dual伪表进行显示，select 函数名（参数） from dual；\*\*

4、  创建函数的语法关键字是\_\_\_\_\*\*create function\*\*\_\_\_\_\_。

## 面试笔试题

1、Oracle中function和procedure，以

及游标的区别？

\*\*1). 可以理解函数是存储过程的一种

2). 函数可以没有参数,但是一定需要一个返回值，存储过程可以没有参数,不需要返回值

3). 函数return返回值没有返回参数模式，存储过程通过out参数返回值, 如果需要返回多个参数则建议使用存储过程

4). 在sql数据操纵语句中只能调用函数而不能调用存储过程\*\*

# Excise 10.3

## 理论练习

1、  将相关的过程、函数、变量、游标和异常等对象进行封装，得到\_\_\_\*\*程序包\*\*\_\_\_\_\_。

2、  程序包由\_\_\_\_\_ \*\*包元素\*\* \_\_\_ 和\_\_\_\_ \*\*包体\*\* \_\_\_ 两部分组成，其中，在 \_\_\_ \*\*包元素\*\* \_\_\_\_\_ 中声明程序包中的公共对象，在\_\_\_ \*\*包体\*\* \_\_\_\_中声明程序包中的私有对象，以及实现在包规范中声明的子程序和游标。

3、  创建程序包的语法关键字是\_\_\_\*\*create package\*\*\_\_\_\_\_。

4、  判断：语句：drop package package\_name;，删除的是程序包的主体。（\*\*对\*\*）

5、  程序包的优点有哪些？

\*\*简化应用设计、提高应用性能、实现信息隐藏、子程序重载。\*\*

# Excise 11.1

## 理论练习

1 、特定事件出现时，自动执行的存储过程称为 \_\_ \*\*触发器\*\* \_\_\_\_\_\_ 。

2 、判断：触发器可以自动触发，也可以被显式调用。（ \*\*错\*\*  ）

3 、判断：触发触发器的特定事件可以是执行查询的 DML 语句。（ \*\*对\*\*  ）

4 、触发器由 \_\_\_ \*\*触发事件\*\* \_\_\_\_\_ 、 \_\_\_\_ \*\*触发条件\*\* \_\_\_\_ 、 \_\_ \*\*触发操作\*\* \_\_\_\_\_\_ 三部分组成。

5 、按触发事件的不同，触发器分为 \_\_\_\*\*DML触发器\*\* \_\_\_\_\_ 、 \_\_\_ \*\*INSTEAD OF 触发器\*\* \_\_\_\_\_ 、 \_\_\_ \*\*系统触发器\*\* \_\_\_\_\_ 三大类。

## 面试笔试题

1、能够自动执行操作的存储对象是： \_\_\_ \*\*触发器\*\* \_\_\_ 。

# Excise 11.2

## 理论练习

1、  DML触发器分为\_\_\_\_ \*\*行级触发器\*\* \_\_\_\_和 \_\_\_\_ \*\*语句级触发器\*\* \_\_\_\_两类。

## 面试笔试题

1、  怎样创建一个触发器？触发器的定义是怎样的？触发器的游标怎样定义？

\*\*创建触发器:

CREATE TRIGGER trigger\_name {INSERT | DELETE | UPDATE} ON table\_name

PL/SQL\_BLOCK | CALL procedure\_name;\*\*

\*\*触发器的定义:

当某个事件发生时自动地隐式运行\*\*

\*\*定义触发器的游标:

ALTER TRIGGER [dbo].[deleteClassSetTrigger]    --新建触发器

ON [dbo].[t\_d\_ExercitationClassSet]  --在某个表中新建的触发器

for delete  --做的什么操作触发触发器

AS

BEGIN

declare @id int --定义变量id

declare cur\_delete cursor  --定义游标

for

select ExercitationClassSetId from deleted --从删除的数据中找到某个字段值

open cur\_delete  --打开游标

fetch next from cur\_delete into @id

while @@fetch\_status=0

begin

delete t\_d\_ExercitationClass where ExercitationClassSetId=@id --执行符合条件的SQL语句

fetch next from cur\_delete into @id --查找下一条数据

end

close cur\_delete --关闭游标

deallocate cur\_delete --删除游标引用

end\*\*

# Excise 11.3

## 理论练习

1 、INSTEAD OF 触发器是定义在 \_\_ \*\*视图\*\* \_\_\_\_\_\_ 上的触发器，该触发器只能是 \_\_\_\_ \*\*行级\*\* \_\_\_\_ （行级 or 语句级）触发器，定义该触发器必须加上 \_\_\_\_ \*\*FOR EACH ROW\*\* \_\_\_\_\_\_ 语句。

2、 INSTEAD OF 触发器不能包含 \_\_\_ \*\*WHEN\*\* \_\_\_\_\_ 子句，不能包含 \_\_ \*\*BEFORE\*\* \_\_\_\_\_\_ 和 \_\_\_\_ \*\*AFTER\*\* \_\_\_\_ 选项。

# Excise 11.4

## 理论练习

1、  被Oracle系统事件自动触发的触发器称为\_\_\_\_\*\*系统触发器\*\* \_\_\_\_\_。

2、  执行哪些操作可以触发系统触发器？

\*\*启动和关闭触发器

用户登录和退出

DDL操作\*\*

# 综合设计题

主观题 (100.0)

1、在某软件公司里，要建立一个数据库来管理员工和项目，其中职员的信息包括(员工编号、姓名、性别、生日、祖籍、工资)，要求员工编号不能重复，姓名必须填写，性别如果不做特别说明默认为男，而且不能填写除‘男’或‘女’之外的其它任何字；工资必须大与0；项目的信息包括(项目编号、项目名称、起始日期、结束日期、预算)，要求项目编号从1001号起每个项目递增1，项目名称不能重复且不可空着。注意：职员是要做项目的！！

要求：

1.  在sqlplus环境中使用system用户登录到服务器

```sql

C:\Users\62531>sqlplus sys/1@orcl as sysdba;

SQL\*Plus: Release 11.2.0.2.0 Production on 星期四 12月 24 16:34:16 2020

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

连接到:

Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

```

2.  使用自己名字的缩写创建一个数据库用户（如张三丰：zhangsf第一个字全拼，其它字只写第一个字母）

3.  授予用户create table 、create session 权限和resource 角色

4.  在PL/SQL Developer 中使用创建的用户登录并打开一个SQL窗口

5. 创建用户及表，并创建相应的约束

```sql

SQL> create user xvbn identified by 1;

User created

SQL> grant create table,create session  to xvbn;

Grant succeeded

SQL> grant resource to xvbn;

Grant succeeded

```

的信息包括(员工编号、姓名、性别、生日、祖籍、工资)，要求员工编号不能重复，姓名必须填写，性别如果不做特别说明默认为男，而且不能填写除‘男’或‘女’之外的其它任何字；工资必须大与0；项目的信息包括(项目编号、项目名称、起始日期、结束日期、预算)，要求项目编号从1001号起每个项目递增1，项目名称不能重

```sql

create table clerk(

  cno number unique,

  cname varchar2(10) not null,

  csex varchar2(2) default '男' check(csex in ('男','女')),

  cbir date,

  chome varchar(10),

  csalary check(csalary>0)

);

create seqeunce PNO\_SEQ

  start with 1

  increment by 1;

create table project(

  no varchar2,

  cname,

  csex,

  cbir,

  chome,

  csalary

);

create or replace trigger "TRIG\_SEQ\_PNO" before insert on project

for each row

 when (new."pno" is null)

begin

  :new.pno := PNO\_SEQ.nextval;

end;

create table cp(

cno number;

pno number;

);

```

6. 使用语句插入以下员工信息和项目信息

|EMPNO|ENAME|ESEX|BIRTHDAY|ADDRESS|SALARY|

|---|---|---|---|---|---|

|1|王晓明|女|1980/1/3|山东青岛|8900|

|2|王涵|男|1978/6/12|广东佛山|760|

|3|李有才|男|1978/5/23|上海|7800|

|4|张小小|女|1982/9/7|山东济南|4500|

```sql

insert into clerk values(1,'王晓明','女',1980/1/3,'山东青岛',8900)

insert into clerk values(2,'王涵  ' ,'男',1978/6/12,'广东佛山',760)

insert into clerk values(3,'李有才','男',1978/5/23,'上海' ,7800)

insert into clerk values(4,'张小小','女',1982/9/7,'山东济南',4500)

```

|ITEMID|ITEMNAME|STARTDATE|ENDDATE|MONEY|

|---|---|---|---|---|---|

|1001|小型监控系统|2009/8/6|2010/1/1|100000|

|1002|办公网络安全|2009/12/30|2010/10/1|450000|

|1003|电子购物广场|2009/11/30|2010/12/31|12000|

```sql

insert into project values(1001,'小型监控系统',2009/8/6,2010/1/1,100000);

insert into project values(1002,'办公网络安全',2009/12/30,2010/10/1,450000);

insert into project values(1003,'电子购物广场',2009/11/30,2010/12/31,12000);

```

 9.员工参与项目的信息如下：王晓明三个项目都参加了，王涵参与了电子购物广场和办公室安全项目，李有才参与了小型监控系统和办公网络安全项目，张小小只参与了小型监控系统一个项目

|EMPNO|ITEMID|

|---|---|

|1|1001|

|1|1002|

|1|1003|

|2|1002|

|2|1003|

|3|1001|

|3|1002|

|4|1001|

```sql

insert into values cp(1,1001);

insert into values cp(1,1002);

insert into values cp(1,1003);

insert into values cp(2,1002);

insert into values cp(2,1003);

insert into values cp(3,1001);

insert into values cp(3,1002);

insert into values cp(4,1001);

```

2、一个简单图书管理系统包括图书馆内书籍信息、学校在校学生信息以及学生借阅信息此系统功能。分为以下三张表:|

1．图书信息表

|字段名称        |   数据类型             |  是否可为空|

|---|---|---|

|图书编号(PK)    | varchar2(10) |             否|

|书名            | varchar2(50)       |       否|

|作者            | varchar2(12)       |       否|

|出版社          |  varchar2(50)      |            是|

|出版日期        |   date             |     是|

|介绍            | varchar2(200)      |            是|

<!--

```sql

create table book(

bno varchar2(10),

bname varchar2(50),

bwriter varchar2(12),

bpuber varchar2(50),

bpubtime date,

bintro varchar2(200)

);

```

 -->

2．读者信息表

|字段名称          | 数据类型            |   是否可为空     |

|---|---|---|

|读者学号(PK)      |     number      |                否|

|读者姓名          | varchar2(10)    |          否|

|读者性别(男/女)   |      varchar2(2)|               否|

|所在系            |varchar2(12)     |             否|

|生效时间          | date            |            是|

|失效时间          | date            |            是|

|累计借书          | number          |       是|

<!--

```sql

create table reader(

rno number not null,

rsex varchar2(10) not null,

rsex varchar2(2) not null,

rmaj varchar2(12) not null,

rdstart date,

rdend date,

rsum number

);

```

 -->

3.借阅表

|字段名称          |数据类型        |        是否可为空                    |

|---|---|---|

|图书编号          |varchar2(10)|               否                             |

|读者学号          |number      |            否                                |

|借书日期          |date        |                 否                           |

|还书日期          |date        |                 否  |

<!--

```sql

create table borrow(

  bno varchar2(10) not null,

  rno number not null,

  bdate date not null,

  pdate date not null,

);

```

 -->

题目:(假设表已经创建,数据已经添加)

1)  查询没有借过书的学生有哪些

```sql

select distinct 读者姓名 from 读者信息表 join 借阅表 on 借阅表.读者学号=读者信息表.读者学号 where 借书日期 is null;

```

2)  将计算机专业的读者借书的还书日期再加30天

```sql

update 借阅表 set 还书日期=还书日期+30 where 所在系='计算机';

```

3)  查询借书最多的读者有哪些

```sql

select 读者姓名

from 读者学号,读者信息表 join 借阅表

on 借阅表.读者学号=读者信息表.读者学号

group by 读者学号 count(图书编号)=max(count(图书编号) );

```