

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 图书借阅系统Oracle数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 计算机学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 王涛 | 学号 | 201810414319 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 10 |  |
| 表设计 | 表，表空间设计合理，数据合理 | 20 |  |
| 用户管理 | 权限及用户分配方案设计正确 | 20 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 30 |  |
| 备份方案 | 备份方案设计正确 | 20 |  |
| **得分合计** | | |  |

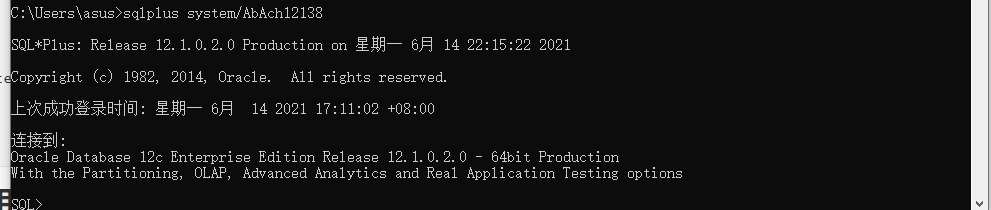
2021 年 6 月 1 日

## 期末考核要求

* 设计项目涉及的表及表空间使用方案。至少5张表和5万条数据，两个表空间。
* 设计权限及用户分配方案。至少两类角色，两个用户。
* 在数据库中建立一个程序包，在包中用PL/SQL语言设计一些存储过程和函数，实现比较复杂的业务逻辑，用模拟数据进行执行计划分析。
* 设计自动备份方案或则手工备份方案。

# 一：搭建本地Oracle环境，登录进入Oracle

//测试登录  
//使用sqlplus，输入用户名和密码  
sqlplus system/AbAch12138

运行结果：  


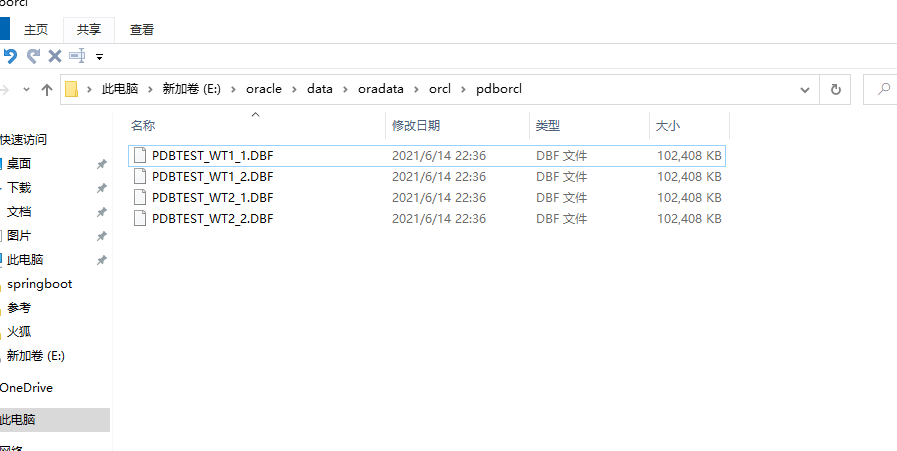
# 二：创建表空间

space\_wt1

Create Tablespace space\_wt1  
datafile  
'E:/oracle/data/oradata/orcl/pdborcl/pdbtest\_wt1\_1.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
'E:/oracle/data/oradata/orcl/pdborcl/pdbtest\_wt1\_2.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

space\_wt2

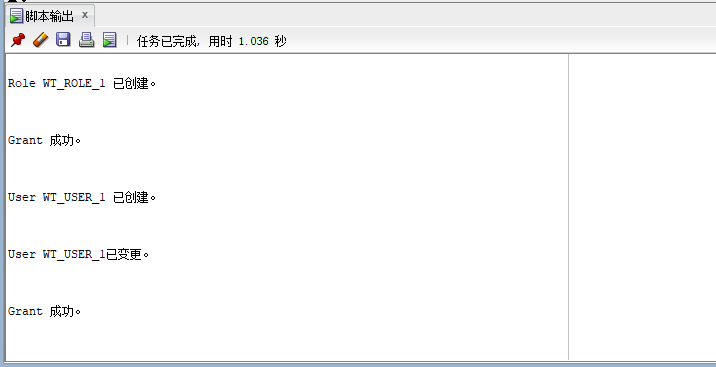
Create Tablespace space\_wt2  
datafile  
'E:/oracle/data/oradata/orcl/pdborcl/pdbtest\_wt2\_1.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
'E:/oracle/data/oradata/orcl/pdborcl/pdbtest\_wt2\_2.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

运行结果：  


# 三、创建用户及角色

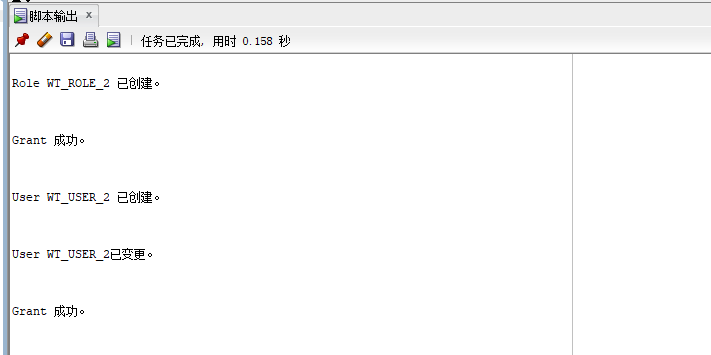
用户默认使用表空间space*wt1*  
*创建角色wt*role*1及用户wt*user\_1

CREATE ROLE wt\_role\_1;  
 GRANT connect,resource,CREATE VIEW TO wt\_role\_1;  
 CREATE USER wt\_user\_1 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_wt1 TEMPORARY TABLESPACE temp;  
 ALTER USER wt\_user\_1 QUOTA 60M ON space\_wt1;  
 GRANT wt\_role\_1 TO wt\_user\_1;

运行结果：  


创建角色wt*role*2及用户wt*user*2

CREATE ROLE wt\_role\_2;  
 GRANT connect,resource,CREATE VIEW TO wt\_role\_2;  
 CREATE USER wt\_user\_2 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_wt2 TEMPORARY TABLESPACE temp;  
 ALTER USER wt\_user\_2 QUOTA 60M ON space\_wt1;  
 GRANT wt\_role\_2 TO wt\_user\_2;

运行结果：  


# 四、创建数据表

### 1.数据表分析：

1.用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| u\_id | number(10) | 否 | 主键 |  | 用户id |
| pwd | VARCHAR2(30 BYTE) | 否 |  |  | 密码 |
| status | number(1) | 否 |  |  | 状态 |
| name | VARCHAR2(50 BYTE) | 否 |  |  | 名字 |
| limit | number(3) | 否 |  | 20 | 借书限额 |

2.图书表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| b\_id | number(10) | 否 | 主键 |  | 图书id |
| writer | VARCHAR2(50 BYTE) | 否 |  |  | 作者 |
| name | VARCHAR2(50 BYTE) | 否 |  |  | 名字 |
| publisher | VARCHAR2(50 BYTE) | 否 |  |  | 出版社 |
| price | number(10,2) | 否 |  |  | 价格 |
| all\_num | int | 否 |  |  | 图书总数量 |
| borrow\_num | int | 否 |  |  | 可借数量 |
| c\_id | number(10) | 否 | 外键 |  | 类别id |

3.管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| m\_id | number(10) | 否 | 主键 |  | 管理员id |
| name | VARCHAR2(50 BYTE) | 否 |  |  | 管理员名字 |
| tel | VARCHAR2(20 BYTE) | 否 |  |  | 管理员联系方式 |
| pwd | VARCHAR2(30 BYTE) | 否 |  |  | 密码 |

4.图书类别表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| c\_id | number(10) | 否 | 主键 |  | 图书类别id |
| text | VARCHAR2(20 BYTE) | 否 |  |  | 图书类别 |

5.借阅记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| br\_id | number(10) | 否 | 主键 |  | 借阅记录id |
| b\_id | number(10) | 否 | 外键 |  | 图书id |
| u\_id | number(10) | 否 | 外键 |  | 用户id |
| borrow\_time | Date | 否 |  |  | 借书时间 |
| return\_time | Date | 是 |  |  | 还书时间 |
| status | int | 否 |  |  | 状态（1.未还，2.已还） |

6.预定图书表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| rr\_id | number(10) | 否 | 主键 |  | 还书记录id |
| b\_id | number(10) | 否 | 外键 |  | 图书id |
| u\_id | number(10) | 否 | 外键 |  | 用户id |
| time | Date | 否 |  |  | 预定时间 |
| status | int | 否 |  |  | 状态（1.预定中，2.预定成功） |

7.消息记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主/外键 | 默认值 | 说明 |
| me\_id | number(10) | 否 | 主键 |  | 消息记录id |
| u\_id | number(10) | 否 | 外键 |  | 用户id |
| text | VARCHAR2(200 BYTE) | 否 |  |  | 消息内容 |
| time | Date | 否 |  |  | 发送时间 |

### 2.创建表

1.用户表

create table T\_USER  
(  
u\_id  number(10) primary key,  
pwd  VARCHAR2(30 BYTE) not null,  
status number(1) not null,  
name VARCHAR2(50 BYTE) not null,  
limit number(3)default 20 not null   
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

2.图书类别表

create table T\_CATEGORY  
(  
c\_id  number(10) primary key,  
text  VARCHAR2(20 BYTE) not null  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

3.图书表

create table T\_BOOK  
(  
b\_id  number(10) primary key,  
writer  VARCHAR2(50 BYTE) not null,  
name VARCHAR2(50 BYTE) not null,  
publisher VARCHAR2(50 BYTE) not null,  
price number(10,2) not null,  
all\_num int not null,  
borrow\_num int not null,  
c\_id number(10),  
constraint c\_id foreign key(c\_id) references T\_CATEGORY(c\_id)  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

4.管理员表

create table T\_MANAGER  
(  
m\_id  number(10) primary key,  
name  VARCHAR2(50 BYTE) not null,  
tel VARCHAR2(20 BYTE) not null,  
pwd VARCHAR2(30 BYTE) not null  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

5.消息记录表

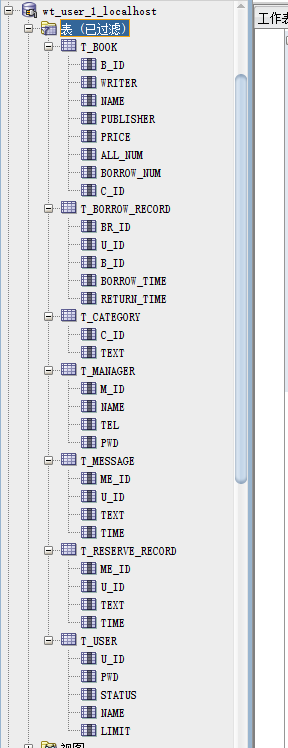
create table T\_MESSAGE  
(  
me\_id  number(10) primary key,  
u\_id  number(10),   
text VARCHAR2(200 BYTE) not null,  
time date not null,  
constraint u\_id foreign key(u\_id) references T\_USER(u\_id)  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

6.借阅记录表

create table T\_BORROW\_RECORD  
(  
br\_id  number(10) primary key,  
u\_id  number(10),   
b\_id number(10),  
borrow\_time date not null,  
return\_time date,  
constraint br\_u\_id foreign key(u\_id) references T\_USER(u\_id),  
constraint b\_id foreign key(b\_id) references T\_BOOK(b\_id)  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

7.预定记录表

create table T\_RESERVE\_RECORD  
(  
rr\_id  number(10) primary key,  
u\_id  number(10),   
b\_id number(10),  
time date not null,  
status number(1),  
constraint me\_u\_id foreign key(u\_id) references T\_USER(u\_id),  
constraint me\_b\_id foreign key(b\_id) references T\_BOOK(b\_id)  
)  
tablespace space\_wt1  
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

运行结果：  


# 五、模拟插入数据

### 插入数据62200条

1.用户表：5000条  
2.图书类别表：2000条  
3.图书表：5000条  
4.管理员类：200条  
5.消息记录表：10000条  
6.借阅记录表：20000条  
7.预定记录表：20000条

### 1.用户表

declare  
i int;   
u\_id  number(10);  
pwd  VARCHAR2(30 BYTE);  
status number(1);  
name VARCHAR2(50 BYTE);  
limit number(3);  
begin  
i:=1;  
while i<=5000  
loop  
u\_id:=i;  
name:= 'user'|| i;  
pwd:= '123';  
status:=1;  
limit:=20;  
insert into T\_USER(u\_id,pwd,status,name,limit) values (u\_id,pwd,status,name,limit);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 2.图书类别表

declare  
i int;   
c\_id   number(10);  
text   VARCHAR2(20 BYTE);  
begin  
i:=1;  
while i<=2000  
loop  
c\_id:=i;  
text:='第'||i||'类';  
insert into T\_CATEGORY(c\_id,text) values (c\_id,text);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 3.图书表

declare  
i int;   
j int;  
k int;  
b\_id  number(10);  
writer VARCHAR2(50 BYTE);  
name VARCHAR2(50 BYTE);  
publisher VARCHAR2(50 BYTE);  
price number(10,2);  
all\_num int;  
borrow\_num int;  
c\_id number(10);  
begin  
i:=1;  
while i<=5000  
loop  
b\_id:=i;  
j := dbms\_random.value()\*50;  
writer:= '作者'||j;  
name:='图书'||i;  
k := dbms\_random.value()\*30;  
publisher :='出版社'||k;  
price :=dbms\_random.value()\*100;  
all\_num :=20;  
borrow\_num:=dbms\_random.value()\*20;  
c\_id:=dbms\_random.value()\*2000;  
insert into T\_BOOK(b\_id,writer,name,publisher,price,all\_num,borrow\_num,c\_id) values (b\_id,writer,name,publisher,price,all\_num,borrow\_num,c\_id);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 4.管理员表

declare  
i int;   
j int;  
k int;  
m\_id   number(10);  
name  VARCHAR2(50 BYTE);  
tel VARCHAR2(20 BYTE);  
pwd VARCHAR2(30 BYTE);  
begin  
i:=1;  
while i<=200  
loop  
m\_id:=i;  
name:='管理员'||i;  
tel:='13333333333';  
pwd:='123456';  
insert into T\_MANAGER(m\_id,name,tel,pwd) values (m\_id,name,tel,pwd);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 5.消息记录表

declare  
i int;  
me\_id number(10);  
u\_id number(10);  
text VARCHAR2(200 BYTE);  
time date;  
begin  
i:=1;  
while i<=10000  
loop  
me\_id:=i;  
text := '您预定的图书已有剩余';  
u\_id := dbms\_random.value()\*4000+1;  
if i mod 6 = 0 then  
  time:= to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =1 then  
  time:=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =2 then  
  time:=to\_date('2017-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =3 then  
  time:=to\_date('2018-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =4 then  
  time:=to\_date('2019-3-2','yyyy-mm-dd')+(i mod 60);  
else  
  time:=to\_date('2020-3-2','yyyy-mm-dd')+(i mod 60);  
end if;  
insert into T\_MESSAGE(me\_id,u\_id,text,time) values (me\_id,u\_id,text,time);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 6.借阅记录表

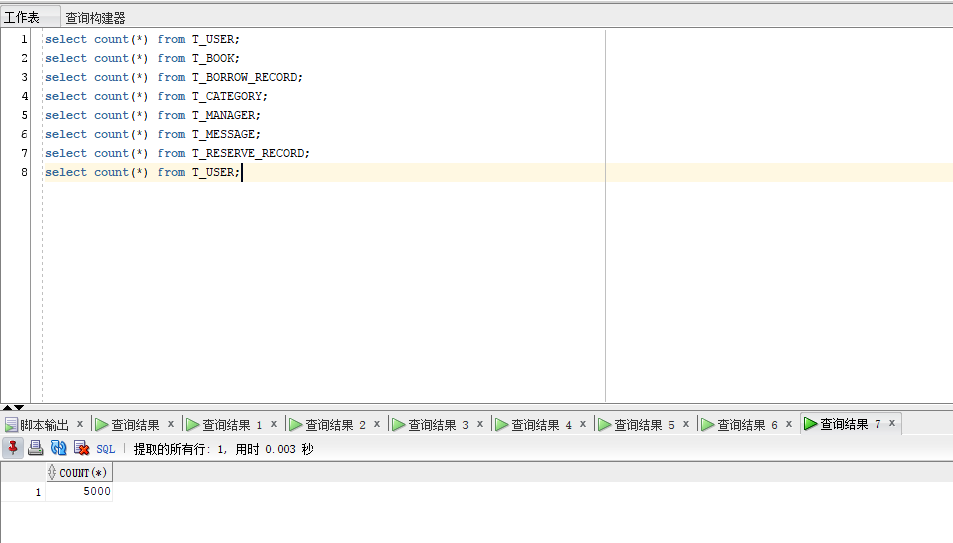
declare  
i int;  
br\_id  number(10);  
u\_id number(10);  
b\_id number(10);  
borrow\_time date;  
return\_time date;  
begin  
i:=1;  
while i<=20000  
loop  
br\_id:=i;  
u\_id:=dbms\_random.value()\*4000+1;  
b\_id:=dbms\_random.value()\*4000+1;  
if i mod 6 = 0 then  
  borrow\_time:= to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);  
 return\_time:= to\_date('2015-3-5','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =1 then  
  borrow\_time:=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =2 then  
  borrow\_time:=to\_date('2017-3-2','yyyy-mm-dd')+(i mod 60);  
 return\_time:= to\_date('2017-3-5','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =3 then  
  borrow\_time:=to\_date('2018-3-2','yyyy-mm-dd')+(i mod 60);  
 return\_time:= to\_date('2018-3-5','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =4 then  
  borrow\_time:=to\_date('2019-3-2','yyyy-mm-dd')+(i mod 60);  
 return\_time:= to\_date('2019-3-10','yyyy-mm-dd')+(i mod 60);  
else  
  borrow\_time:=to\_date('2020-3-2','yyyy-mm-dd')+(i mod 60);  
 return\_time:= to\_date('2020-4-10','yyyy-mm-dd')+(i mod 60);  
end if;  
insert into T\_BORROW\_RECORD(br\_id,u\_id,b\_id,borrow\_time,return\_time) values (br\_id,u\_id,b\_id,borrow\_time,return\_time);  
i:=i+1;  
end loop;  
commit;  
end;  
/

### 7.预定记录表

declare  
i int;  
rr\_id  number(10);  
u\_id number(10);  
b\_id number(10);  
time date;  
status number(1);  
begin  
i:=1;  
while i<=20000  
loop  
rr\_id:=i;  
u\_id:=dbms\_random.value()\*4000+1;  
b\_id:=dbms\_random.value()\*4000+1;  
status:=dbms\_random.value()\*2+1;  
if i mod 6 = 0 then  
  time:= to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =1 then  
  time:=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =2 then  
  time:=to\_date('2017-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =3 then  
  time:=to\_date('2018-3-2','yyyy-mm-dd')+(i mod 60);  
elsif i mod 6 =4 then  
  time:=to\_date('2019-3-2','yyyy-mm-dd')+(i mod 60);  
else  
  time:=to\_date('2020-3-2','yyyy-mm-dd')+(i mod 60);  
end if;  
insert into T\_RESERVE\_RECORD(rr\_id,u\_id,b\_id,time,status) values (rr\_id,u\_id,b\_id,time,status);  
i:=i+1;  
end loop;  
commit;  
end;  
/

## 8.查询是否插入成功

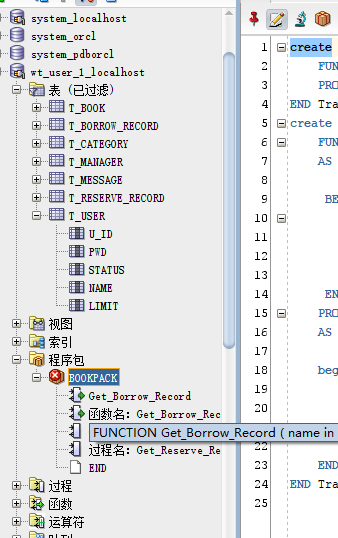
select count(\*) from T\_USER;  
select count(\*) from T\_BOOK;  
select count(\*) from T\_BORROW\_RECORD;  
select count(\*) from T\_CATEGORY;  
select count(\*) from T\_MANAGER;  
select count(\*) from T\_MESSAGE;  
select count(\*) from T\_RESERVE\_RECORD;  
select count(\*) from T\_USER;

运行结果：  


# 六、创建程序包及存储过程，函数

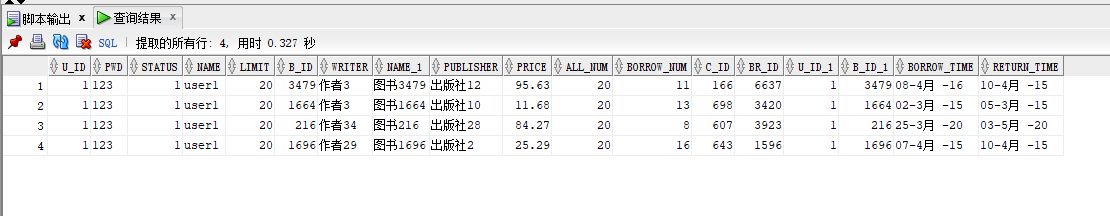
包名：BookPack  
函数名：Get\_Borrow\_Record 功能：输入用户名字，获取该用户借书的图书信息  
过程名：Get\_Reserve\_Record 功能：输入用户名字，获取该用户预定图书的图书信息

--创建包  
create or replace PACKAGE BookPack IS  
 FUNCTION 函数名：Get\_Borrow\_Record(name VARCHAR2);  
 PROCEDURE 过程名：Get\_Reserve\_Record(name VARCHAR2);  
END TrainPack;  
--创建函数和存储过程  
create or replace PACKAGE BODY BookPack IS  
 FUNCTION Get\_Borrow\_Record(name in VARCHAR2)  
 AS  
  
 BEGIN  
 select \* from T\_USER u,T\_BOOK b,T\_BORROW\_RECORD br   
 where u.u\_id=br.u\_id   
 and b.b\_id=br.b\_id   
 and u.name=name;   
 END;  
 PROCEDURE Get\_orders(name VARCHAR2)  
 AS  
  
 begin  
 select \* T\_USER u,T\_BOOK b,T\_RESERVE\_RECORD rr   
 where u.u\_id=br.u\_id   
 and b.b\_id=rr.b\_id   
 and u.name=name;   
 END;  
END TrainPack;

运行结果：  


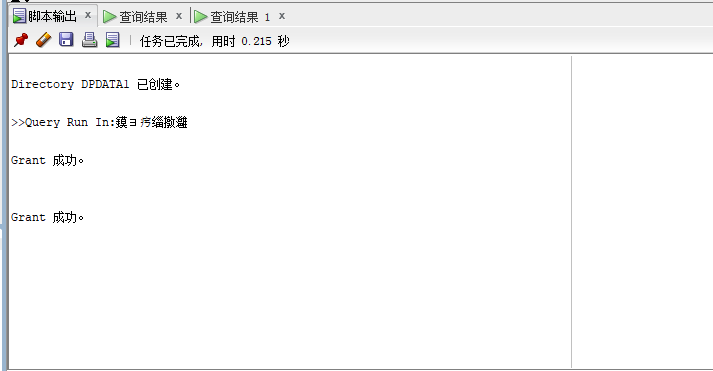
### 测试函数和存储过程

DECLARE  
begin  
 Get\_Borrow\_Record('user1');  
end;  
DECLARE  
begin  
 Get\_Borrow\_Record('user1');  
end;

运行结果:  


# 七、设计手动备份

-- 数据备份  
create directory dpdata1 as 'F:\ORACLE\copy';  
select \* from dba\_directories;  
-- 开始备份  
-- 授权给用户  
grant read,write on directory dpdata1 to wt\_user\_1;

运行结果：  


备份：

expdp system/AbAch12138@ORCL directory=dpdata1 dumpfile=beifen.dmp logfile=beifen.log schemas=wt\_user\_1

运行结果：  
