

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的购物系统的数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 贺巍 | 学号 | 201810414113 |
| 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** | | |  |

1. 年 6 月 1 日

# 1表设计

1.1 数据表设计

（1）商品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| goods\_id | varchar2(10) | 否 | 主键 | 商品id |
| goods\_name | varchar2(50) | 否 |  | 商品名称 |
| class\_id | varchar2(10) | 否 | 外键 | 商品类别表外键 |
| goods\_price | number(10,2) | 否 |  | 不能为负 |
| goods\_info | varchar2(100) | 否 |  | 商品介绍 |

（2）商品库存表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| inventory\_id | varchar2 (10) | 否 | 主键 | 库存id |
| goods\_id | varchar2 (10) | 否 | 外键 | 商品表外键 |
| quantity | number(10,0) | 否 |  | 库存数量 |
| position | varchar2 (50) | 否 |  | 分表 |

（3）商品分类表

商品分类表（class）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| class\_id | varchar2 (10) | 否 | 主键 | 分类id |
| class\_name | varchar2 (50) | 否 |  | 分类名称 |

（4）管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| admin\_id | varchar2 (10) | 否 | 主键 | 管理员id |
| admin\_username | varchar2 (20) | 否 |  | 管理员用户名 |
| admin\_password | varchar2(20) | 否 |  | 管理员密码 |
| admin\_sex | varchar2 (10) | 否 |  | 男,女,未知 |

（5）用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| user\_id | varchar2 (10) | 否 | 主键 | 用户id |
| username | varchar2 (20) | 否 |  | 用户名 |
| password | varchar2 (20) | 否 |  | 密码 |
| name | varchar2 (20) | 否 |  | 用户名称 |
| sex | varchar2(10) | 否 |  | 性别 |
| phone | number(11,0) | 否 |  | 手机号码 |

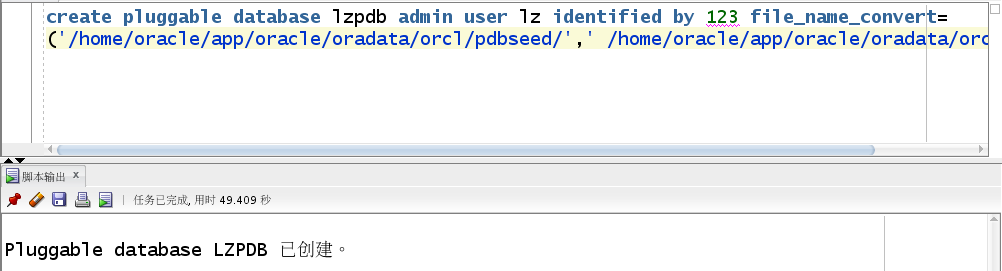
（6）订单表

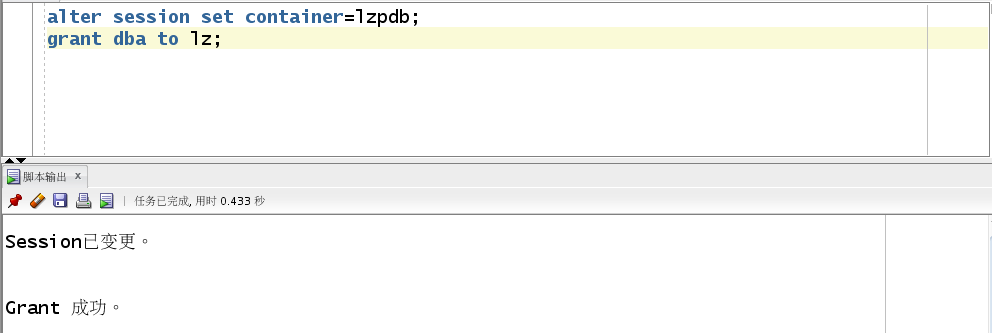
订单表（order）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 备注 |
| order\_id | varchar2(10) | 否 | 主键 | 订单id |
| order\_date | date | 否 |  | 下单时间 |
| order\_price | number (10,2) | 否 |  | 订单价格 |
| order\_info | varchar2 (100) | 否 |  | 订单详情 |
| user\_id | varchar2 (10) | 否 | 外键 | 用户id |
| class\_id | varchar2 (10) | 否 | 外键 | 类别id |

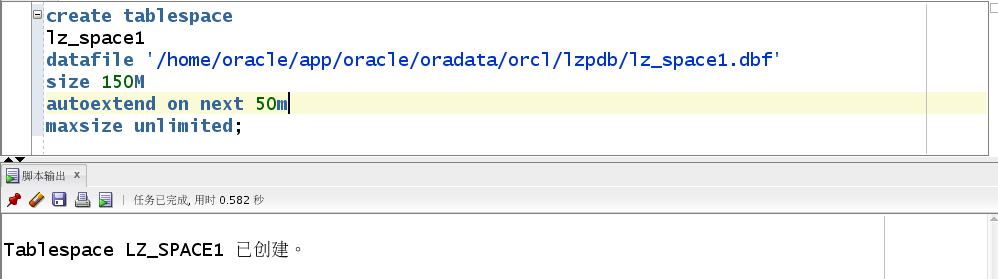
1.2 表空间设计

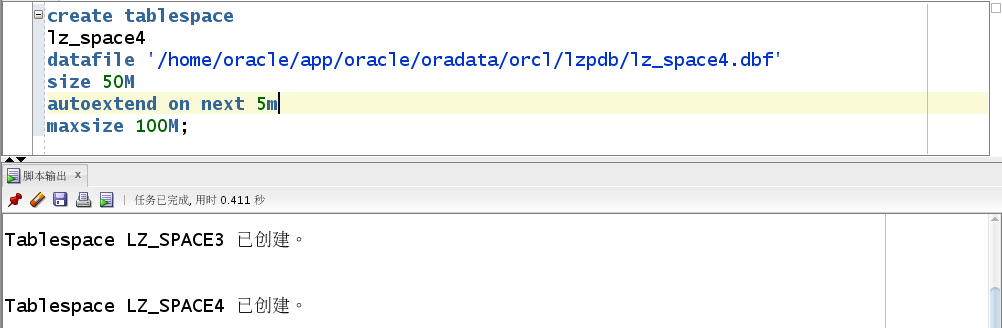
（1）创建数据库以及管理员





（2）创建表空间





(3)创建表

create table class (

class\_id varchar2 (10) not null primary key,

class\_name varchar2 (50) not null,

)TABLESPACE lz\_space1;

create table goods (

goods\_id varchar2(10) not null primary key,

goods\_name varchar2(50) not null,

goods\_price float not null CHECK(goods\_price > 0),

commodity\_info varchar2(100) not null,

class\_id varchar2(10) not null,

CONSTRAINT class\_id\_lz FOREIGN KEY (class\_id) REFERENCES class (class\_id)

)TABLESPACE lz\_space1;

create table user (

user\_id varchar2(10) not null primary key,

username varchar2(20) not null,

password varchar2(20) not null,

name varchar2(20) not null,

sex varchar2(10) not null,

phone number(11,0)

)TABLESPACE lz\_space1;

create table admin (

admin\_id varchar2 (10) not null primary key,

admin\_username varchar2 (20) not null,

admin\_password varchar2 (20) not null,

admin\_sex varchar2 (10) not null check(admin\_sex in ('男','女')),

)TABLESPACE lz\_space1;

create table inventory (

inventory\_id varchar2(10) not null primary key,

goods\_id varchar2(10) not null,

quantity number(10,0),

position varchar2(50),

CONSTRAINT inventory\_goods\_id FOREIGN KEY (goods\_id) REFERENCES goods (goods\_id)

)TABLESPACE lz\_space1;

CREATE TABLE order (

order\_id varchar2(10) NOT NULL primary key,

order\_info varchar2(100) NOT NULL,

order\_price number(10,2) NOT NULL,

user\_id char(10),

classr\_id char(10),

order\_date date,

CONSTRAINT order\_class\_id FOREIGN KEY (class\_id) REFERENCES class (class\_id),

CONSTRAINT order\_user\_id FOREIGN KEY (user\_id) REFERENCES user (user\_id)

) partition by range (order\_date)

(

partition p1 values LESS THAN (TO\_DATE(' 2021-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace lz\_space1,

partition p2 values LESS THAN (TO\_DATE(' 2022-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace lz\_space2,

partition p3 values LESS THAN (TO\_DATE(' 2023-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace lz\_space3,

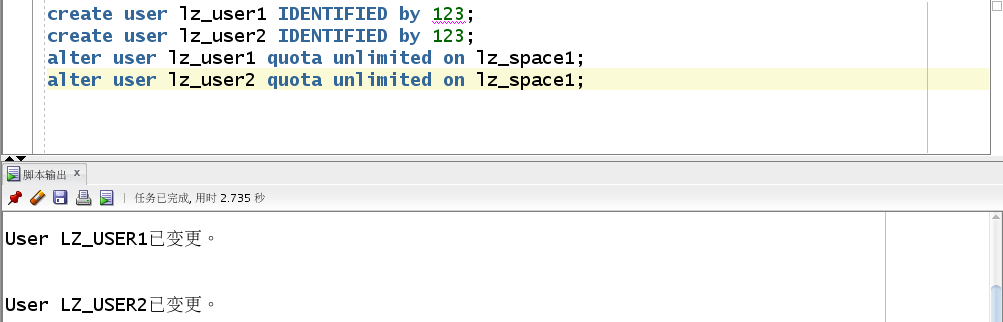
partition p4 values LESS THAN (MAXVALUE) tablespace lz\_space4

)

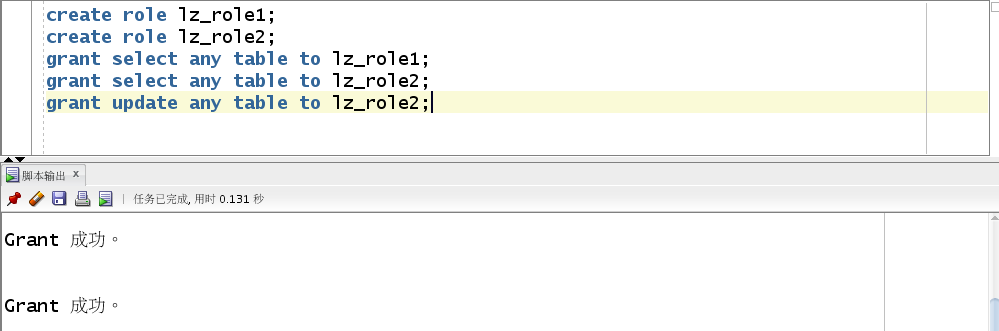
# 2 用户管理

2.1创建用户,授权,插入数据，以及对分区，授权的验证

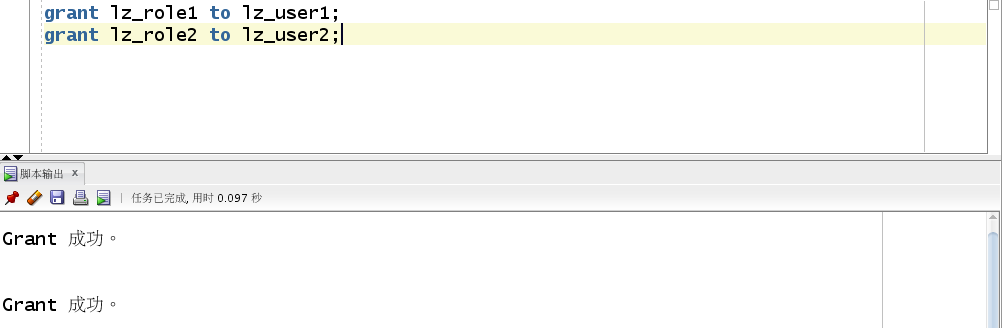
（1）分别创建两个用户lz\_user1和lz\_user2,并允许他们是使用lz\_space1表空间



1. 创建两个角色lz\_role1，lz\_role2，并分别赋予lz\_role1读任何表和读，lz\_role2读任何表和读、修改任何表的权利；



（3）将lz\_role1，lz\_role2分别赋予给用户lz\_user1，lz\_user2



（4）向表中插入数据：

declare

dt date;

order\_id varchar2(10);

order\_info varchar2(100);

order\_price number (10,2);

class\_id varchar2(10);

user\_id char(5);

begin

insert into class(class\_id,class\_name) values ('1','食品');

insert into class(class\_id,class\_name) values ('2','服饰');

insert into class(class\_id,class\_name) values ('3','数码');

Insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('101','手撕面包',20,'1');

insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('102','坚果',29,'1');

insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('103','T恤',65,'2');

insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('104','短裤',88,'2');

insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('105','iphone11',4999,'3');

insert into goods(goods\_id,goods\_name,goods\_price,class\_id) values ('106','小米11',3999,'3');

insert into admin(admin\_id,admin\_sex,admin\_username,admin\_password)

values ('1','男','0101','123456');

insert into user(user\_id,user\_name,user\_sex,user\_phone) values ('1','hw','男','17369066636');

for i in 1..50000

loop

if i mod 3=0 then

dt := to\_date('2018-01-01','yyyy-mm-dd')+(i mod 60);

--PARTITION\_2018

elsif i mod 6 = 1 then

dt := to\_date('2019-01-01','yyyy-mm-dd')+(i mod 60);

--PARTITON\_2019

elsif i mod 6 = 2 then

dt := to\_date('2020-01-01','yyyy-mm-dd')+(i mod 60);

--PARTITION\_2020

end if;

--开始向order表里插入数据

order\_id := i;

user\_id := '1';

class\_id := '1';

order\_info := '订单详情';

order\_price := dbms\_random.value(5,30);

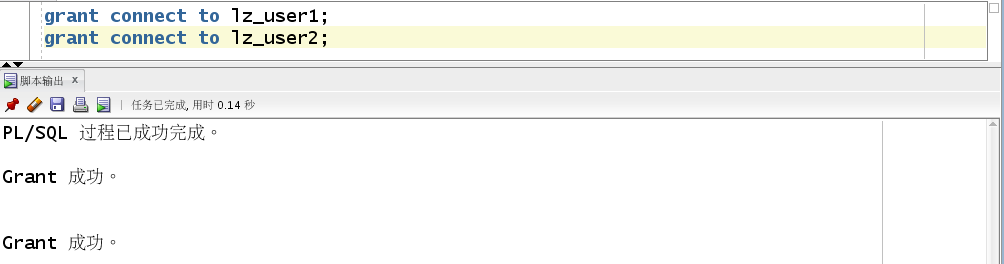
insert into orders (order\_id,order\_info,order\_price,user\_id,class\_id)

values (order\_id,order\_info,order\_price,user\_id,class\_id);

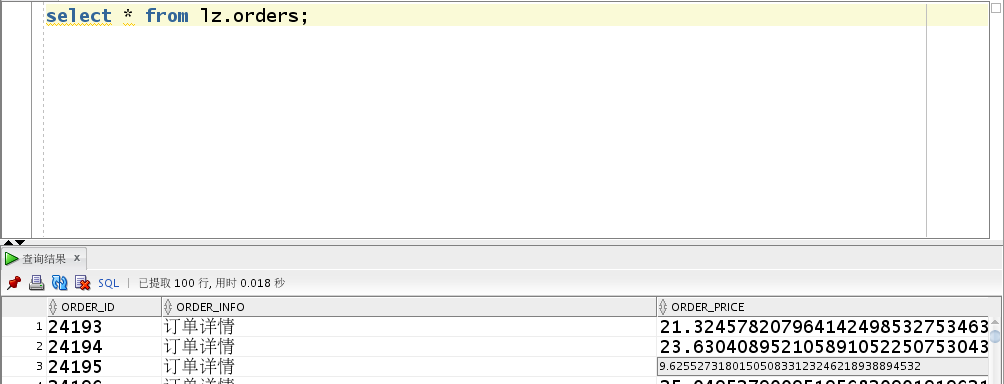
end loop;

end;

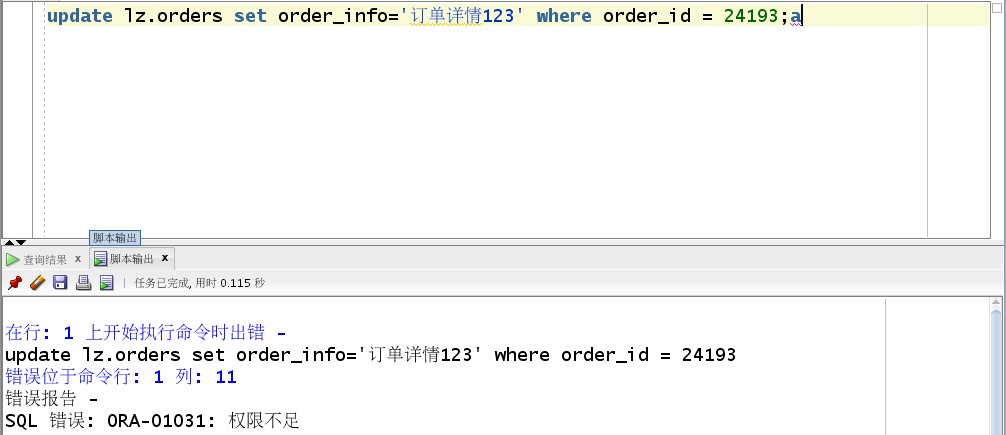
（5）之前创建了两个用户，一个是lz\_user1,一个是lz\_user2，并且给lz\_user1和lz\_user2分别授予了查看表，查看表以及更新表的权限，接下来进行验证：



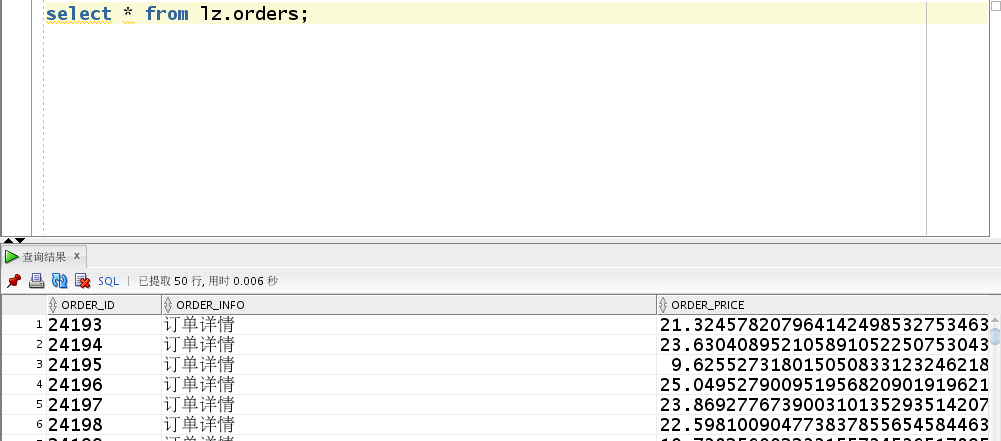
首先，让lz\_user1连接数据库lzpdb,让其查看lz的ORDER表，如下图所示，是可以看见的。

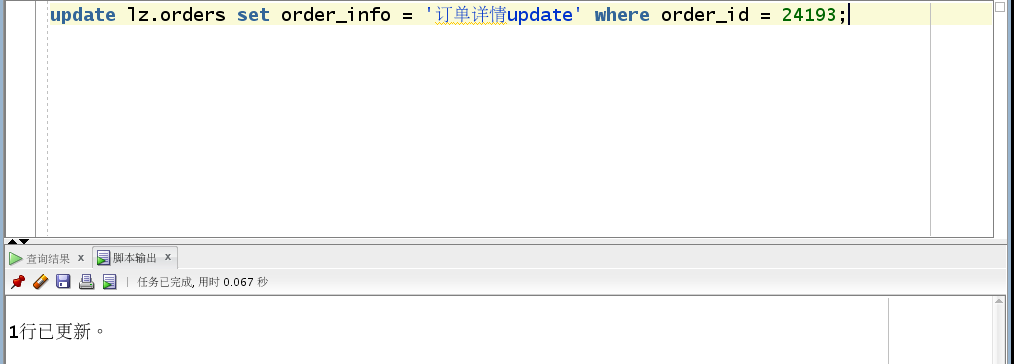


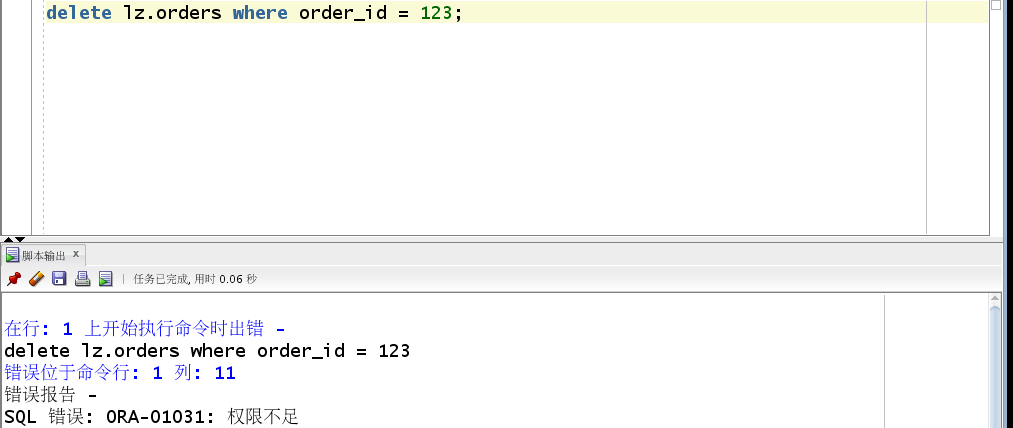
然后，让lz\_user1更新一条数据，会发现其权限不足，由此可以证明，授权成功。



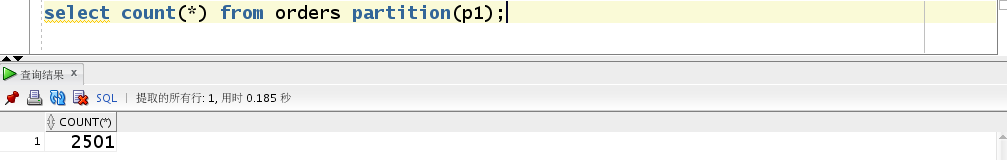
接着对lz\_user2(之前赋予了更新和查看权限）用户查看和更新lz的ORDERS表，发现均成功，但让其删除一条记录是会显示权限不足，由此可证明，授权无误。

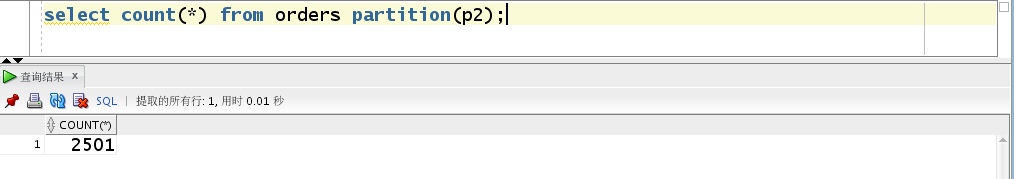


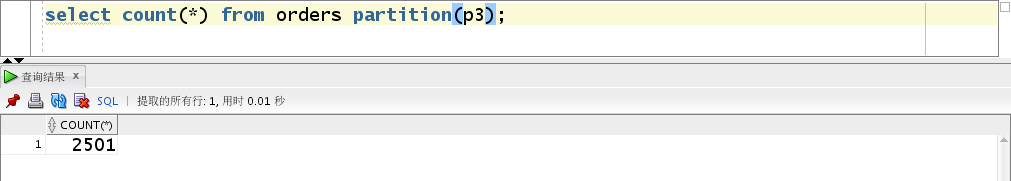


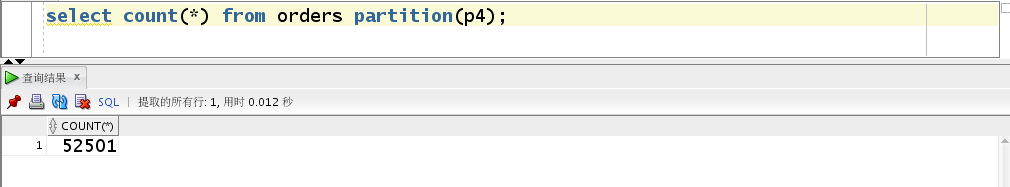


1. 根据REGISTRATION\_FORM表中的REGISTRATION\_DATE字段根据时间段的不同，将其放在了不同的表空间里面，由下图可知，每个表空间里面均有REGISTRATION\_FORM的数据，也可证明分区成功。









# 3 PL/SQL设计

3.1 程序包的建立

建立一个程序包，里面有一个函数和一个存储过程，其功能分别为：统计一段时间内的订单总收入；统计一段时间内，各个导购员的接待数。具体过程如下：

create or replace PACKAGE MyPack is

function get\_total(dt1 char,dt2 char) return number;

procedure get\_people(dt1 char,dt2 char);

END MyPack;

/

create or replace PACKAGE BODY MyPack is

function get\_total(dt1 char,dt2 char) return number

AS

n number;

begin

select sum(order\_price) into n from orders where order\_date >=to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')

and order\_date<=to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');

return n;

end;

procedure get\_people(dt1 char,dt2 char)

AS

a number;

b number;

cursor people is

select \* from orders where order\_date>= to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')

and order\_date<= to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');

begin

a:=0;

b:=0;

for v in people

loop

if v.class\_id = '1'

then a:= a+1;

elsif v.class\_id = '2'

then b:= b+1;

end if;

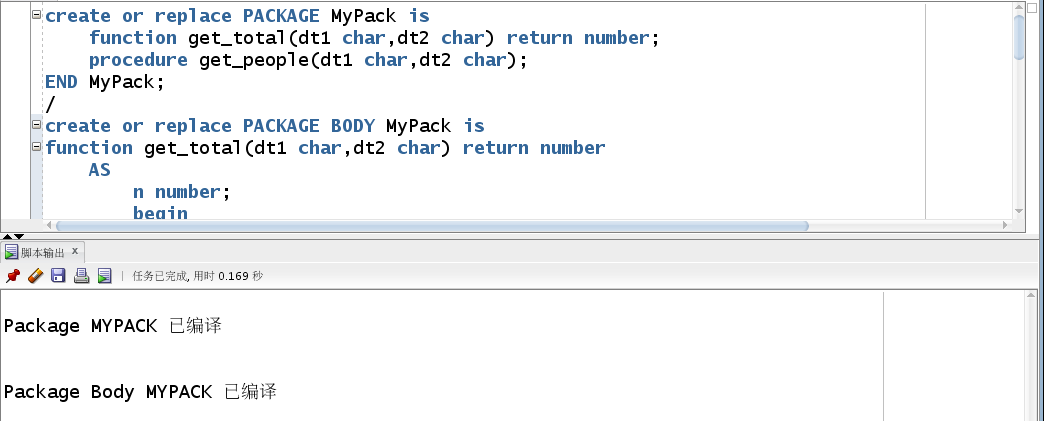
end loop;

dbms\_output.put\_line('零食类售出数量为:'|| a);

dbms\_output.put\_line('服饰类售出数量为:'||b);

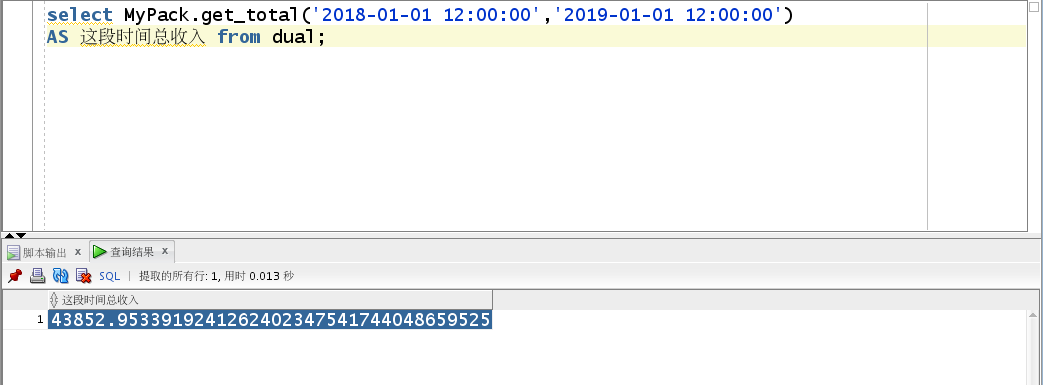
end;

END MyPack;

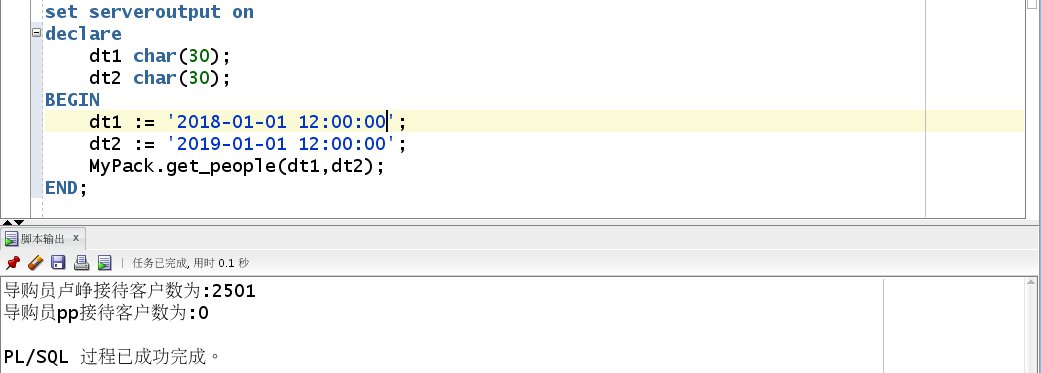


接下来是效果展示：

函数效果展示



存储过程效果展示



# 4备份方案

4.1 系统定时自动备份

（1）编写rman增量备份脚本

#rman\_level1.sh

#!/bin/sh

export NLS\_LANG='SIMPLIFIED CHINESE\_CHINA.AL32UTF8'

export ORACLE\_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome\_1

export ORACLE\_SID=orcl

export PATH=$ORACLE\_HOME/bin:$PATH

rman target / nocatalog msglog=/home/oracle/rman\_backup/lv1\_`date +%Y%m%d-%H%M%S`\_L0.log << EOF

run{

configure retention policy to redundancy 1;

configure controlfile autobackup on;

configure controlfile autobackup format for device type disk to '/home/oracle/rman\_backup/%F';

configure default device type to disk;

crosscheck backup;

crosscheck archivelog all;

allocate channel c1 device type disk;

backup as compressed backupset incremental level 1 database format '/home/oracle/rman\_backup/dblv1\_%d\_%T\_%U.bak'

plus archivelog format '/home/oracle/rman\_backup/arclv1\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

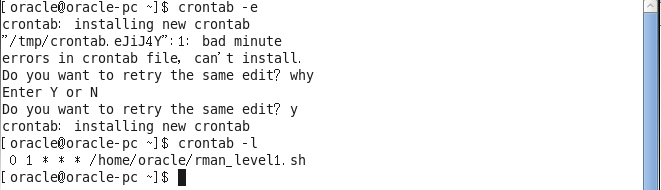
release channel c1;

}

EOF

exit

（2）启动linux的crontab定时任务，每天的凌晨一点自动进行备份



1. 演示备份与恢复：

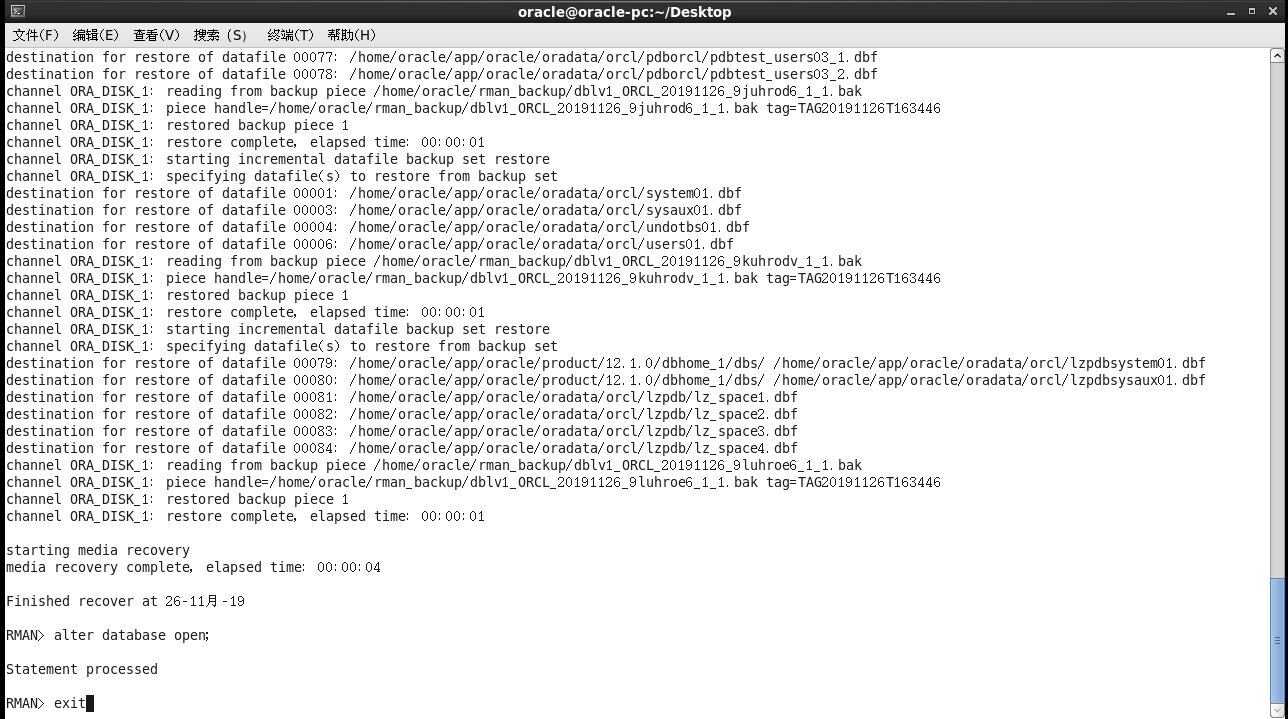
（3.1）执行rman\_level1.sh脚本，进行数据库备份



（3.2）删除数据库

（3.3）启用rman，进行数据库恢复





（3.4）查看删除的是否恢复



（3.5）恢复成功