

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的网络购物的数据库设计 | | |
| 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 | 2017级 |
| 学生姓名 | 钟港 | 学号 | 201710414130 |
| 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 | 10 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 25 |  |
| 备份方案 | 备份方案设计正确 | 25 |  |
| 容灾方案 | DataGuard设计正确 | 10 |  |
| **得分合计** | | |  |

2019 年 11 月 18 日

目录

1. 系统概述---------------------------------------------------------------------3
2. 业务流程---------------------------------------------------------------------3
3. 实体分析---------------------------------------------------------------------4

3.1 实体模型---------------------------------------------------------------------------------4

3.2 实体联系模型-----------------------------------------------------------------------------5

1. 数据表设计-------------------------------------------------------------------6

4.1商品表---------------------------------------------------------------------6

4.2 库存（仓库）表--------------------------------------------------------------6

4.3 商家表----------------------------------------------------------------------7

4.4 订单表----------------------------------------------------------------------7

4.5 顾客表----------------------------------------------------------------------7

1. 创建空间表和用户-------------------------------------------------------------7

5.1 创建数据库--------------------------------------------------------------------------------8

5.2 创建表空间--------------------------------------------------------------------------------8

5.3 创建表------------------------------------------------------------------------------------9

5.4 创建用户及授权----------------------------------------------------------------------------13

5.5 验证用户授权是否正确----------------------------------------------------------------------16

1. 创建程序包-------------------------------------------------------------------18
2. 数据备份---------------------------------------------------------------------20

7.1 开始全备份--------------------------------------------------------------------------------20

7.2 每天定时开启备份--------------------------------------------------------------------------21

7.3 执行脚本备份数据库------------------------------------------------------------------------21

7.4 删除数据库文件----------------------------------------------------------------------------21

7.5 重启损坏的数据库到mount状态-------------------------------------------------------------22

7.6 删除备份集--------------------------------------------------------------------------------23

1. **系统概述**

电子商务是指以[信息网络](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E7%BD%91%E7%BB%9C" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)技术为手段，以商品交换为中心的[商务活动](https://baike.baidu.com/item/%E5%95%86%E5%8A%A1%E6%B4%BB%E5%8A%A8/3153331" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)；也可理解为在[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91/199186" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)、企业内部网和[增值网](https://baike.baidu.com/item/%E5%A2%9E%E5%80%BC%E7%BD%91/145452" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)上以电子交易方式进行交易活动和相关服务的活动，是传统商业活动各环节的电子化、[网络化](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8C%96/5810347" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)、信息化；以互联网为媒介的商业行为均属于电子商务的范畴。

在[因特网](https://baike.baidu.com/item/%E5%9B%A0%E7%89%B9%E7%BD%91/114119" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)开放的网络环境下，基于客户端/服务端应用方式，买卖双方不谋面地进行各种商贸活动，实现消费者的[网上购物](https://baike.baidu.com/item/%E7%BD%91%E4%B8%8A%E8%B4%AD%E7%89%A9/214710" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)、商户之间的[网上交易](https://baike.baidu.com/item/%E7%BD%91%E4%B8%8A%E4%BA%A4%E6%98%93/420646" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)和在线电子支付以及各种商务活动、交易活动、金融活动和相关的综合服务活动的一种新型的商业运营模式。

电子商务是因特网爆炸式发展的直接产物，是[网络技术](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%8A%80%E6%9C%AF" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)应用的全新发展方向。因特网本身所具有的开放性、[全球性](https://baike.baidu.com/item/%E5%85%A8%E7%90%83%E6%80%A7" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)、[低成本](https://baike.baidu.com/item/%E4%BD%8E%E6%88%90%E6%9C%AC" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)、高效率的特点，也成为电子商务的内在特征，并使得电子商务大大超越了作为一种新的[贸易形式](https://baike.baidu.com/item/%E8%B4%B8%E6%98%93%E5%BD%A2%E5%BC%8F" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)所具有的价值，它不仅会改变企业本身的生产、经营、管理活动，而且将影响到整个[社会的经济](https://baike.baidu.com/item/%E7%A4%BE%E4%BC%9A%E7%9A%84%E7%BB%8F%E6%B5%8E" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)运行与结构。以互联网为依托的“电子”技术平台为传统商务活动提供了一个无比宽阔的发展空间，其突出的优越性是传统[媒介](https://baike.baidu.com/item/%E5%AA%92%E4%BB%8B" \t "https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1/_blank)手段根本无法比拟的。

近年来，随着科技水平的不断发展以及电子经济的迅速发展，网络购物开始出现并且迅速被人们所接受，所以网络购物的建设尤为重要，需要确保可以保证大家的操作是稳定的、安全的。并且不断完善发展，发展为让大家可以有更好的体验模式。

1. **业务流程**

业务或业务活动是对组织或企业的一切专业工作和活动的总称。业务流程图是根据系统详细调查过程中所得的资料和问卷调查的结果，按业务实际处理过程绘制成的图，在网络购物系统，主要分为三个部分：商品的存取管理、商家的调配管理、顾客购买管理。

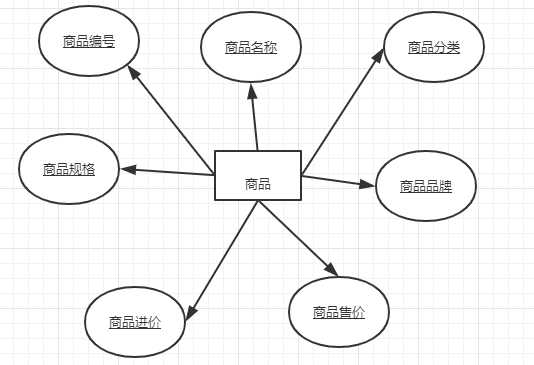
1. 顾客购买管理：顾客购买时，一般是先看好商品，然后填写收货地址，完成之后进行支付购买，之后就可以等待商家发货。
2. 商家调配管理：商家在发货之前，需要了解订单详情，然后根据大家不同的需求，发送不同的货物。
3. 商品存取管理：一个网店的开关，首先需要拥有一个发送货源的地方，没有足够的货源便无法顺利的完成顾客的需求。在经营之中如果货已经销售完了便需要进货，进来的货都存放在一个地方，当顾客下单了，便需要根据顾客的要求去取出满足顾客需求的商品，通过快递等方式，确保货物可以达到顾客的手上。

**第三章 实体分析**

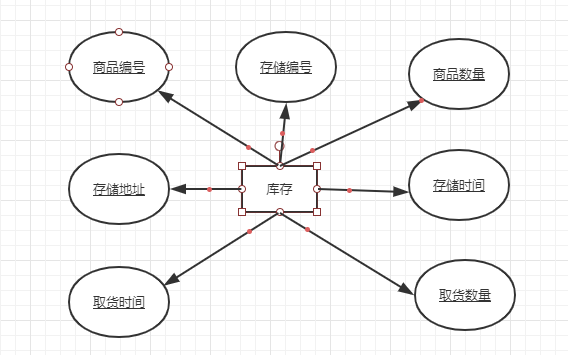
**3.1实体模型**

根据业务流程分析一共有5个实体，这5个实体分别是：商品、库存（仓库）、商家、订单、顾客。

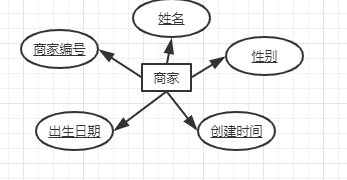
1.商品（commodity）：商品编号（commodity\_no）、商品名称（commodity\_name）、商品分类（commodity\_class）、商品规格（commodity\_spec）、商品品牌（commodity\_brand）、商品进价（intake\_price）、商品售价（outtake\_price）



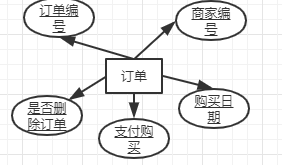
1. 库存（Stock）：商品编号（commodity\_no）、存储编号（stock\_no）、商品数量（size）、存储地址（position）、存储时间（store\_time）、取货时间（take\_time）、取货数量（take\_size）



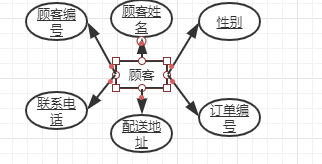
1. 商家（business）：商家编号（business\_no）、商家姓名（business\_name）、性别（sex）、出生日期（birthday）、创建时间（Creation\_ time）



1. 订单（Order）：订单编号（Order\_no）、商家编号（business\_no）、是否删除订单（delete）、支付购买（price）、购买日期（date）



1. 顾客（Customer）：顾客编号（Customer\_no）、顾客姓名（Customer\_name）、性别（sex）、联系电话（tel）、配送地址（address）、订单编号（Order\_no）



**3.2 实体联系模型**

1.商品需要在仓库里进行存取



1. 商家可以接受多个订单



1. 一个顾客对应一个订单



**第四章 数据表的设计**

1.商品表

商品表（commodity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| commodity\_no | char(5) | 否 | 主键 |  |
| commodity\_name | char(20) | 否 |  |  |
| commodity\_class | char(8) |  |  |  |
| commodity\_spec | char(10) |  |  |  |
| commodity\_brand | char(10) |  |  |  |
| intake\_price | Float | 否 |  | >0 |
| outtake\_price | Float | 否 |  | >0 |

1. 库存（仓库）表

库存（仓库）表（stock）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| stock\_no | char(5) | 否 |  |  |
| commodity\_no | char(5) | 否 | 外键 |  |
| Size | char(10) |  |  |  |
| position | char(20) |  |  |  |
| store\_time | char(10) |  |  |  |
| take\_time | char(10) |  |  |  |
| take\_size | char(10) |  |  |  |

1. 商家表

商家表（business）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| business\_no | char(5) | 否 | 主键 |  |
| business\_name | char(10) | 否 |  |  |
| sex | char(2) | 否 |  | 男/女 |
| birthday | char(10) |  |  |  |
| Creation\_ time | char(10) | 否 |  |  |

1. 订单表

订单表（Orderby）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| Order\_no | char(5) | 否 | 主键 |  |
| business\_no | char(5) | 否 | 外键 |  |
| delete | char(2) |  |  |  |
| price | float |  |  |  |
| Order\_date | char(10) |  |  |  |

1. 顾客表

顾客表（Customer）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| Customer\_no | char(5) | 否 | 主键 |  |
| Customer\_name | char(10) | 否 |  |  |
| sex | char(2) | 否 |  | 男/女 |
| tel | char(11) |  |  |  |
| address | char(20) | 否 |  |  |
| Order\_no | char(5) | 否 |  |  |

**第五章 创建zgmail数据库，表空间，表，以及用户**

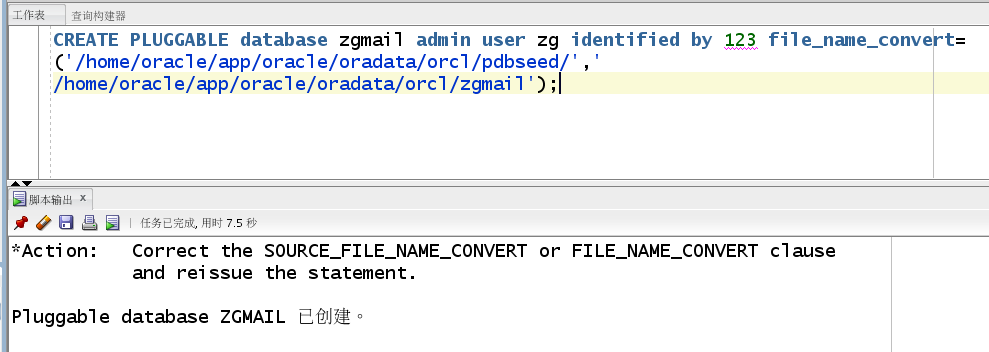
**5.1创建数据库zgmail，指定创建位置，并且创建数据库管理员zg**

Sql如下：

CREATE PLUGGABLE database zgmail admin user zg identified by 123 file\_name\_convert=

('/home/oracle/app/oracle/oradata/orcl/pdbseed/','

/home/oracle/app/oracle/oradata/orcl/zgmail');



**5.2 创建表空间**

在zgmail数据库中分别创建表空间zg-space1、表空间zg-space2、表空间zg\_space3和表空间zg\_space4.同时分别设置表空间的相关大小。设置zg\_space1时，将表空间1的大小（size）设置为150M，另外没有最大限制。表空间2的空间大小设置数据和表空间1的一样。表空间3和表空间4的空间大小设置参数一致，都为50m同时最大的大小限制为100m。根据条件完成设置。

create tablespace

zg\_space1

datafile '/home/oracle/app/oracle/oradata/orcl/zgmail/zg\_space1.dbf'

size 150M

autoextend on next 50m

maxsize unlimited;

create tablespace

zg\_space2

datafile '/home/oracle/app/oracle/oradata/orcl/zgmail/zg\_space2.dbf'

size 150M

autoextend on next 50m

maxsize unlimited;

create tablespace

zg\_space3

datafile '/home/oracle/app/oracle/oradata/orcl/zgmail/zg\_space3.dbf'

size 50M

autoextend on next 5m

maxsize 100m;

create tablespace

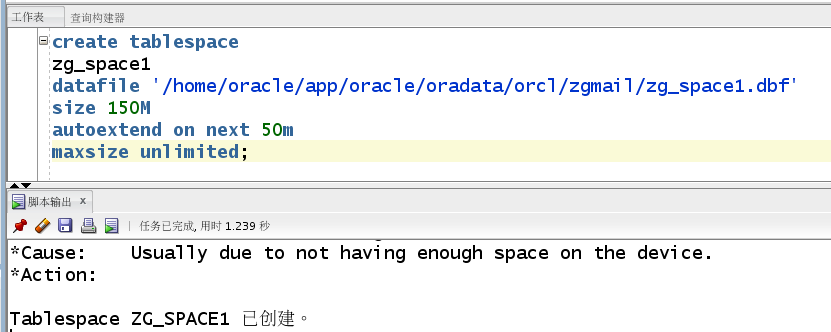
zg\_space4

datafile '/home/oracle/app/oracle/oradata/orcl/zgmail/zg\_space4.dbf'

size 50M

autoextend on next 5m

maxsize 100m;



创建成功的图片基本相似，所以这里只使用了表空间zg\_space1的结果图片。

**5.3创建表以及将订单表（Orderby）进行分区**

1. 创建商品表（commodity）

create table commodity(

commodity\_no char(5) not null primary key,

commodity\_name char(25) not null,

commodity\_class char(8),

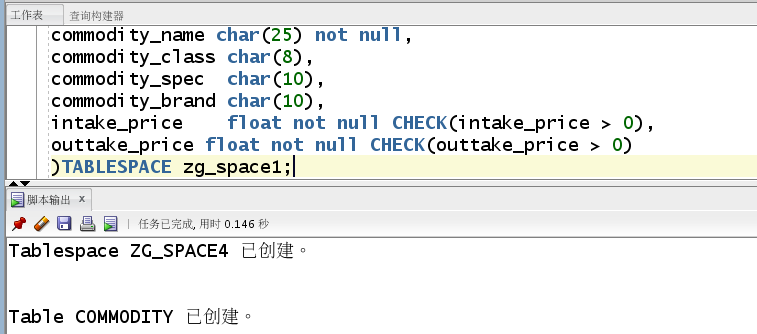
commodity\_spec char(10),

commodity\_brand char(10),

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

outtake\_price float not null CHECK(outtake\_price > 0)

)TABLESPACE zg\_space1;



1. 创建商品库存表（stock）

create table stock(

stock\_no char(5),

commodity\_no char(5),

size\_ char(10),

position\_ char(20),

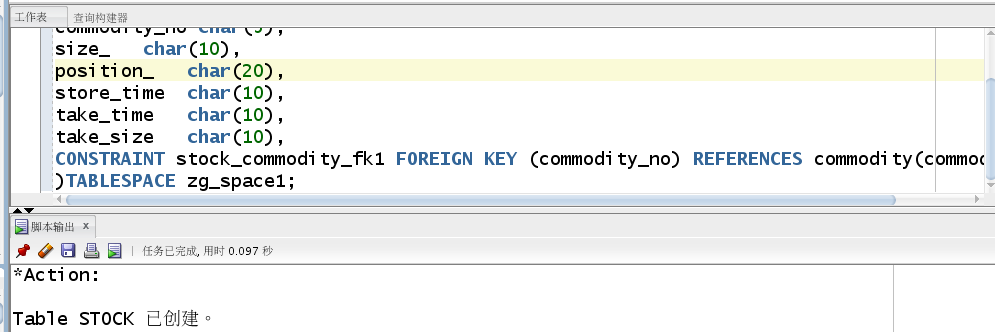
store\_time char(10),

take\_time char(10),

take\_size char(10),

CONSTRAINT stock\_commodity\_fk1 FOREIGN KEY (commodity\_no) REFERENCES commodity(commodity\_no)

)TABLESPACE zg\_space1;



1. 创建商家表（business）

create table bussiness(

business\_no char(5) not null PRIMARY KEY,

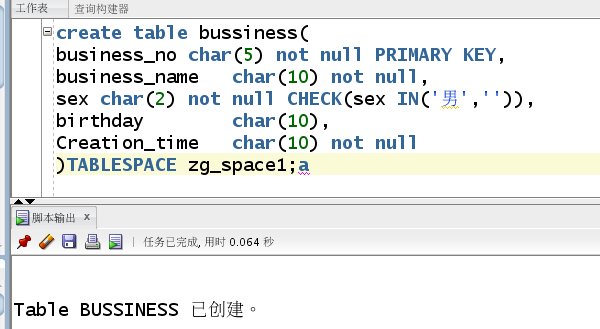
business\_name char(10) not null,

sex char(2) not null CHECK(sex IN('男','')),

birthday char(10),

Creation\_time char(10) not null

)TABLESPACE zg\_space1;



1. 对订单表进行分区（Orderby）

创建订单（Order），将订单的订单时间（Order\_date）进行分区，分别将2017年之前的数据放在表空间zg\_space1, 将2018 到 2017年的数据放在了表空间zg\_space2，将2019年 到 2018年的数据放在了表空间zg\_space3,最后将大于2019年的数据放在了zg\_spcae4里面。

create table orderby(

order\_no char(5) not null PRIMARY KEY,

business\_no char(5) not null,

delete\_ char(2 byte),

price float(126),

order\_date date not null,

CONSTRAINT order\_FORM\_bussiness\_fk1 FOREIGN KEY (business\_no) REFERENCES bussiness (business\_no)

) partition by range (order\_date)

(

partition p1 values LESS THAN (TO\_DATE(' 2017-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace zg\_space1,

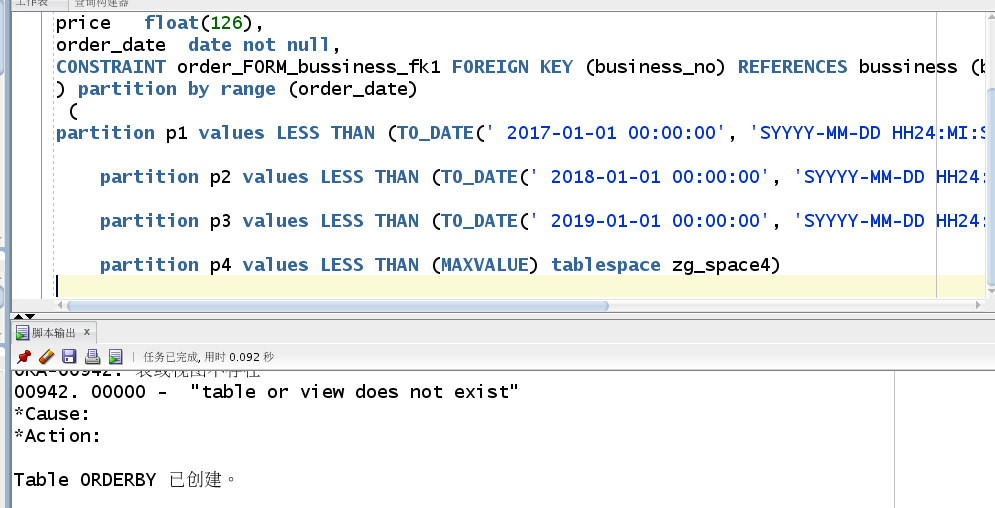
partition p2 values LESS THAN (TO\_DATE(' 2018-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace zg\_space2,

partition p3 values LESS THAN (TO\_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace zg\_space3,

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

partition p3 values LESS THAN (TO\_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN')) tablespace zg\_space3,

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



1. 创建顾客表（Customer）

create table customer(

customer\_no char(5) not null PRIMARY KEY,

customer\_name char(10) not null,

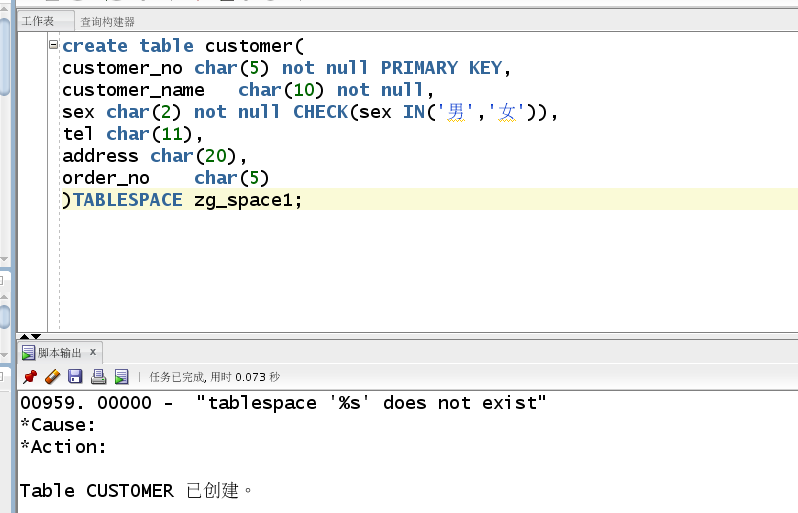
sex char(2) not null CHECK(sex IN('男','女')),

tel char(11),

address char(20),

order\_no char(5)

)TABLESPACE zg\_space1;



**5.4 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zg1和zg2，两个用户的密码都是123，并且允许他们使用zg\_space1的表空间.sql语句如图所示：

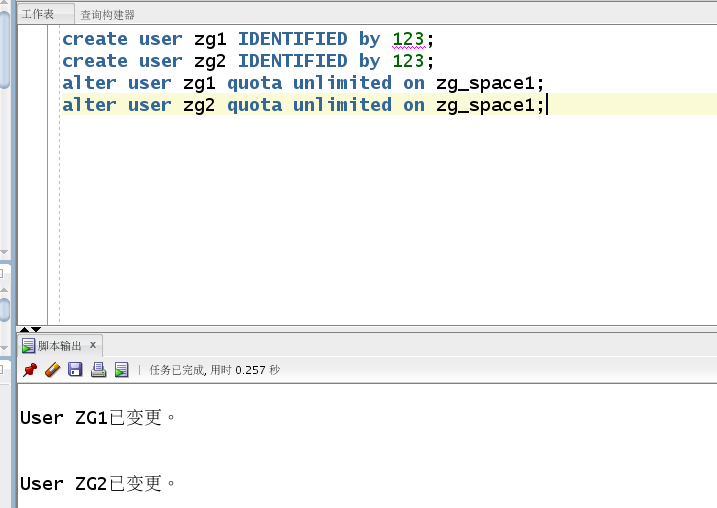
create user zg1 IDENTIFIED by 123;

create user zg2 IDENTIFIED by 123;

alter user zg1 quota unlimited on zg\_space1;

alter user zg2 quota unlimited on zg\_space1;

创建sql语句以及结果如图所示



1. 创建两个角色role1和role2，并且给这两个角色分别赋予查看以及修改的权限。Sql语句如下图所示，通过grant any table to sql语句完成给角色赋予权限的操作。

create role role1;

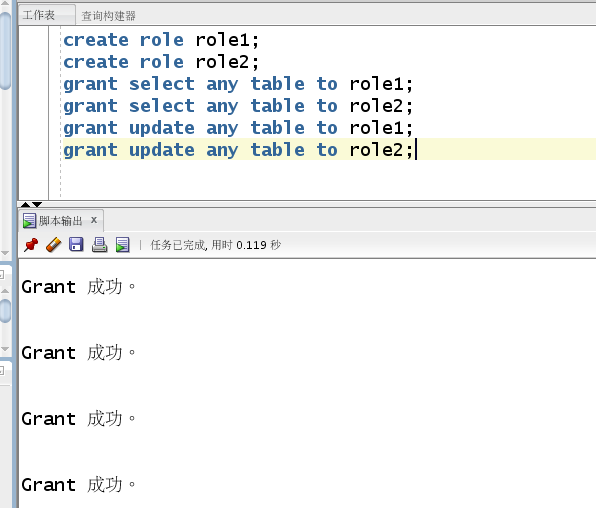
create role role2;

grant select any table to role1;

grant select any table to role2;

grant update any table to role1;

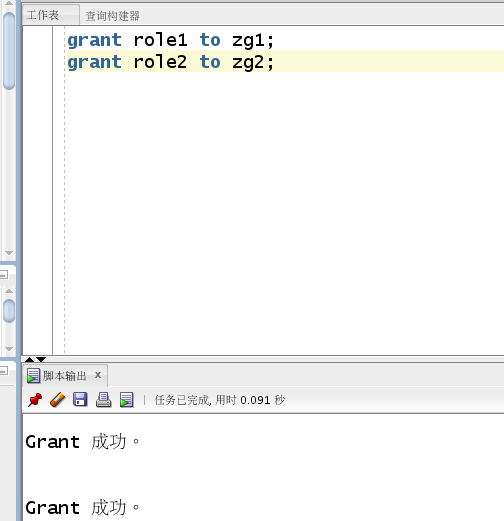
grant update any table to role2;



1. 将两个角色role1和role2分别交给两个用户zg1和zg2，使用grant to 的sql语句完成这个过程的操作。

grant role1 to zg1;

grant role2 to zg2;



1. 向表中插入数据

declare

dt date;

order\_no char(5);

business\_no char(5);

delete\_ char(2);

price float;

begin

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('1','小米1','电产','发烧机','小米',1000,2000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('2','小米2','电产','发烧机','小米',2000,3000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('3','小米3','电产','发烧机','小米',3000,4000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('4','小米4','电产','发烧机','小米',3000,4000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('5','小米5','电产','发烧机','小米',4000,5000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('1','小米1','电产','发烧机','小米',1000,2000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('2','小米2','电产','发烧机','小米',2000,3000);

Insert into

commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('3','小米3','电产','发烧机','小米',3000,4000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('4','小米4','电产','发烧机','小米',3000,4000);

insert into commodity(commodity\_no,commodity\_name,commodity\_class,commodity\_spec,commodity\_brand,intake\_price,outtake\_price)

values('5','小米5','电产','发烧机','小米',4000,5000);

insert into bussiness(business\_no,business\_name,sex,birthday,Creation\_time)

values('1001','海尔','1','1998','1');

insert into bussiness(business\_no,business\_name,sex,birthday,Creation\_time)

values('1002','海尔','1','1999','2');

insert into bussiness(business\_no,business\_name,sex,birthday,Creation\_time)

values('1003','海尔','1','2000','3');

insert into bussiness(business\_no,business\_name,sex,birthday,Creation\_time)

values('1004','海尔','1','2001','4');

insert into bussiness(business\_no,business\_name,sex,birthday,Creation\_time)

values('1005','海尔','1','2002','5');

for i in 1..50000

loop

if i mod 3 =0 then

dt:=to\_date('2017-01-01','yyyy-mm-dd')+(i mod 60);--PARTITION\_2017

elsif i mod 6 =1 then

dt:=to\_date('2018-01-01','yyyy-mm-dd')+(i mod 60);--PARTITION\_2018

elsif i mod 6 =2 then

dt:=to\_date('2019-01-01','yyyy-mm-dd')+(i mod 60);--PARTITION\_2019

end if;

--插入订单

order\_no := i;

business\_no := case i mod 6 when 0 then '1001' when 1 then '1002' when 2 then '1003' when 3 then '1004' else '1005' end;

delete\_ := case i mod 2 when 0 then '1' else '0' end;

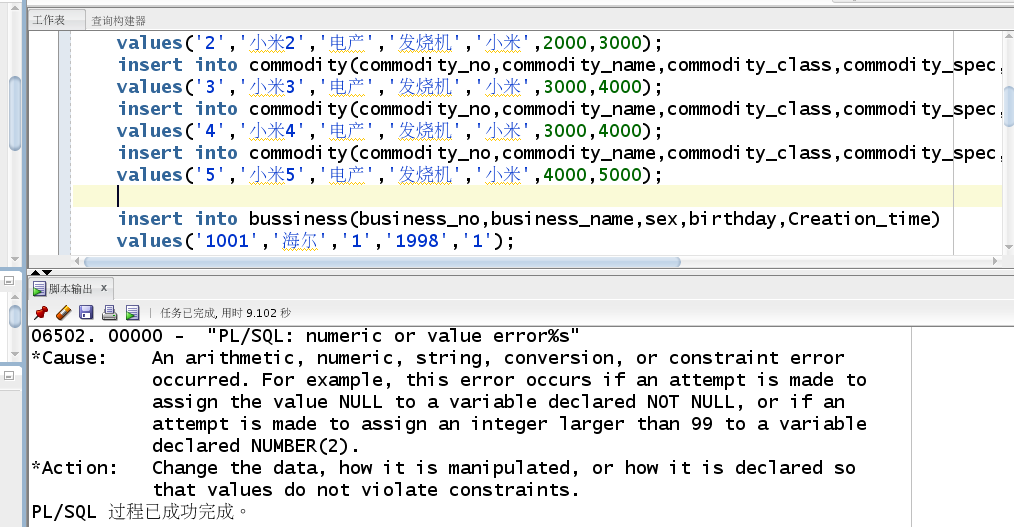
price := dbms\_random.value(5,30);

insert into orderby (order\_no,business\_no,delete\_,price,order\_date)

values(order\_no,business\_no,delete\_,price,dt);

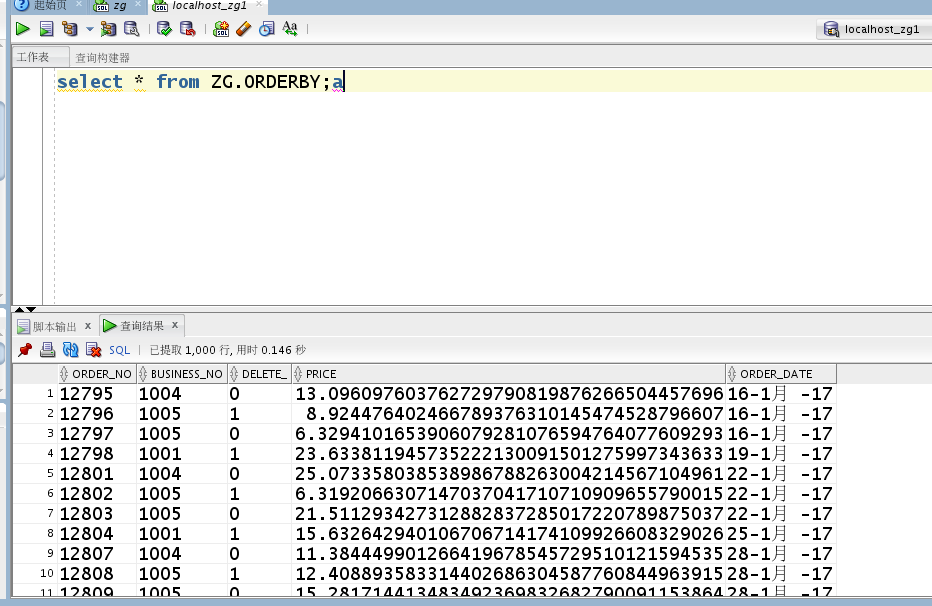
end loop;

end;

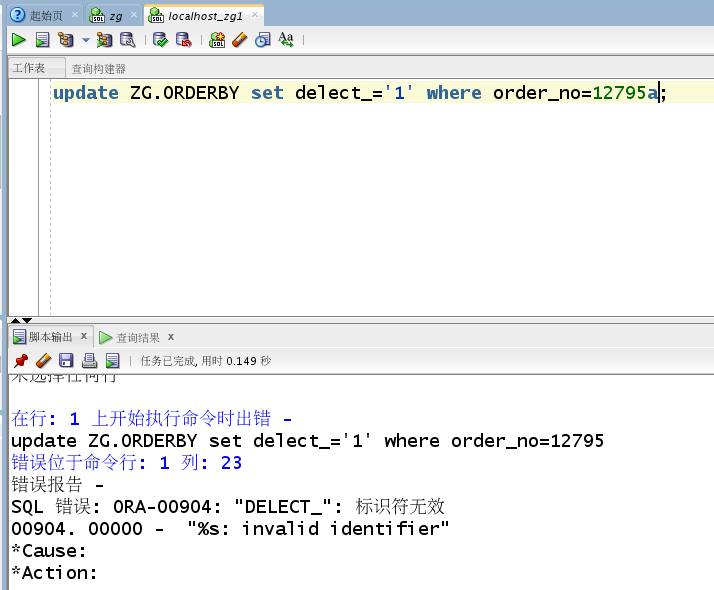


1. 验证用户zg1和zg2授权权限操作是否成功
2. 使用zg1连接数据库zgmail，然后查看zg的orderby表

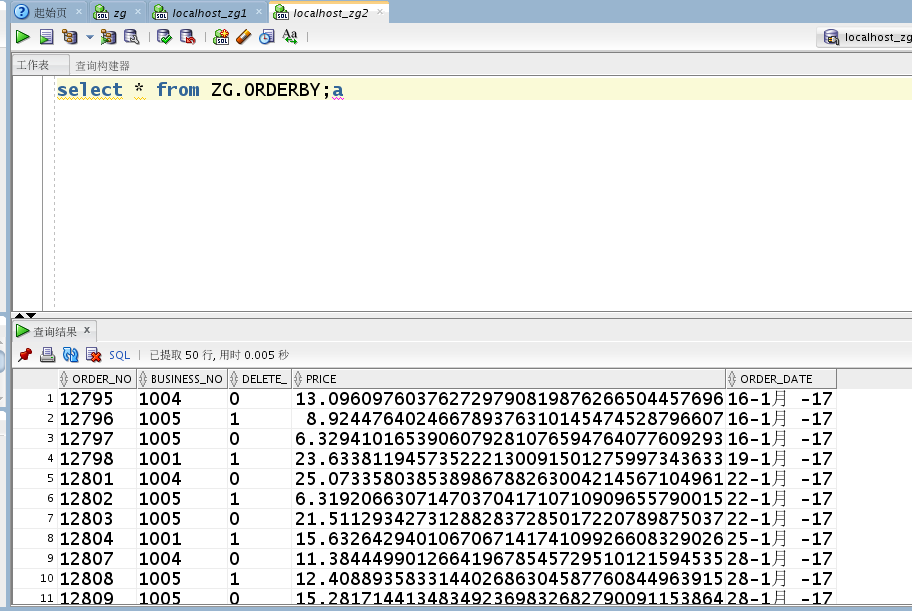
结果如下图

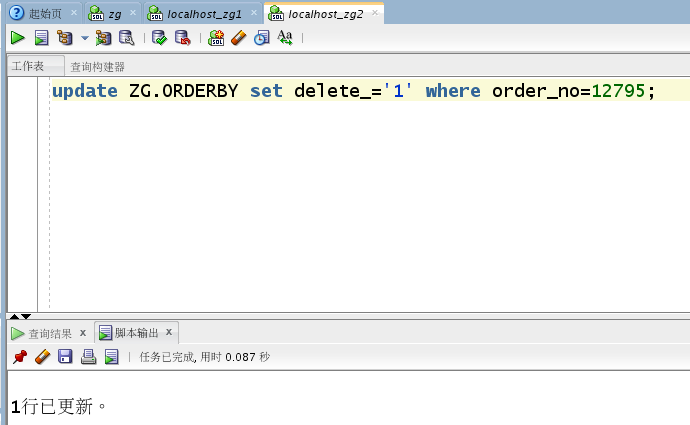


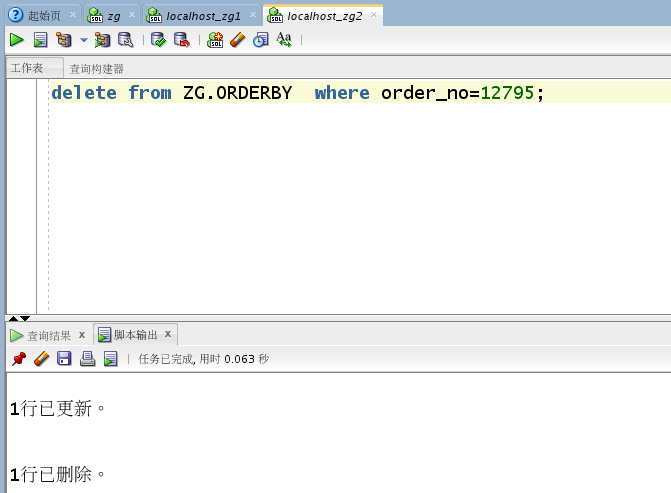
然后使用zg1更新一条数据，会发现其权限不足，由此证明，授权成功



1. 接下来对zg2分别赋予查看和更新orderby的权限，可以发现可以查看和更新同时还可以删除一条数据，说明赋予了删除权限，同时可以证明赋予权限是很成功的。

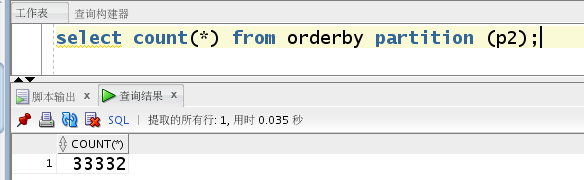






1. 根据orderby表中的order\_date根据不同的时间分区，将不同的数据分在了不同的表空间中，下面将对不同的表空间进行查看，看是否成功将数据插入到了不同的表空间中。

结果如下图







1. **建立程序包**

建立一个程序包里面主要包括了一个函数和一个存储过程，其功能是：统计一段时间内商品销售的总收入；统计一段时间内各个商品的售卖数量。（程序包——表示嵌入或链接信息的图标。此信息可以包含整个文件（如[位图文件](https://baike.baidu.com/item/%E4%BD%8D%E5%9B%BE%E6%96%87%E4%BB%B6/8504338" \t "https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%8C%85/_blank)）或部分文件（如[电子表格](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E8%A1%A8%E6%A0%BC/8488847" \t "https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%8C%85/_blank)中的某个单元）。当选择此数据包时，用于创建对象的应用程序将播放对象（如声音文件）或者打开并显示对象。如果更改原始信息，链接的信息将自动更新。）

create or replace PACKAGE mypack is

FUNCTION get\_total(dt1 char,dt2 char)RETURN number;

PROCEDURE get\_number(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(price) into n from orderby where order\_date >=to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')

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

a4 number;

a5 number;

cursor cur is

select \* from orderby 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

a1 :=0;

a2 :=0;

a3 :=0;

a4 :=0;

a5 :=0;

for v in cur

loop

if v.business\_no='1001'

then a1 :=a1+1;

elsif v.business\_no='1002'

then a2 :=a2+1;

elsif v.business\_no='1003'

then a3 :=a3+1;

elsif v.business\_no='1004'

then a4 :=a4+1;

elsif v.business\_no='1005'

then a5 :=a5+1;

end if;

end loop;

dbms\_output.put\_line('business为1001的人数为:'|| a1);

dbms\_output.put\_line('business为1002的人数为:'|| a2);

dbms\_output.put\_line('business为1003的人数为:'|| a3);

dbms\_output.put\_line('business为1004的人数为:'|| a4);

dbms\_output.put\_line('business为1005的人数为:'|| a5);

end;

end mypack;

return n;

end;

--得到每段时间的销售数量

PROCEDURE get\_number(dt1 char,dt2 char)

AS

a1 number;

a2 number;

a3 number;

a4 number;

a5 number;

cursor cur is

select \* from orderby 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

a1 :=0;

a2 :=0;

a3 :=0;

a4 :=0;

a5 :=0;

for v in cur

loop

if v.business\_no='1001'

then a1 :=a1+1;

elsif v.business\_no='1002'

then a2 :=a2+1;

elsif v.business\_no='1003'

then a3 :=a3+1;

elsif v.business\_no='1004'

then a4 :=a4+1;

elsif v.business\_no='1005'

then a5 :=a5+1;

end if;

end loop;

dbms\_output.put\_line('business为1001的人数为:'|| a1);

dbms\_output.put\_line('business为1002的人数为:'|| a2);

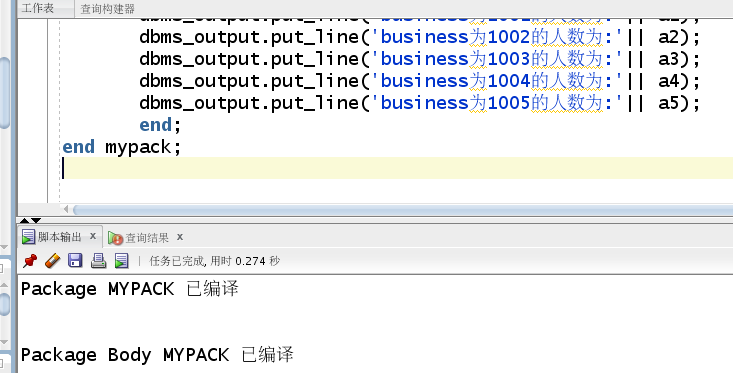
dbms\_output.put\_line('business为1003的人数为:'|| a3);

dbms\_output.put\_line('business为1004的人数为:'|| a4);

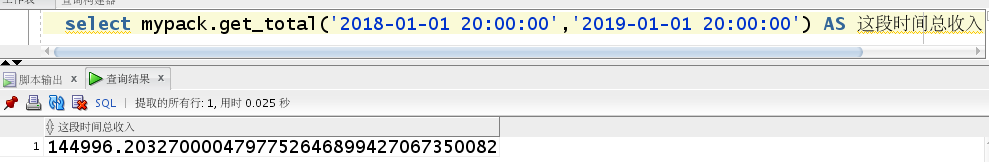
dbms\_output.put\_line('business为1005的人数为:'|| a5);

end;

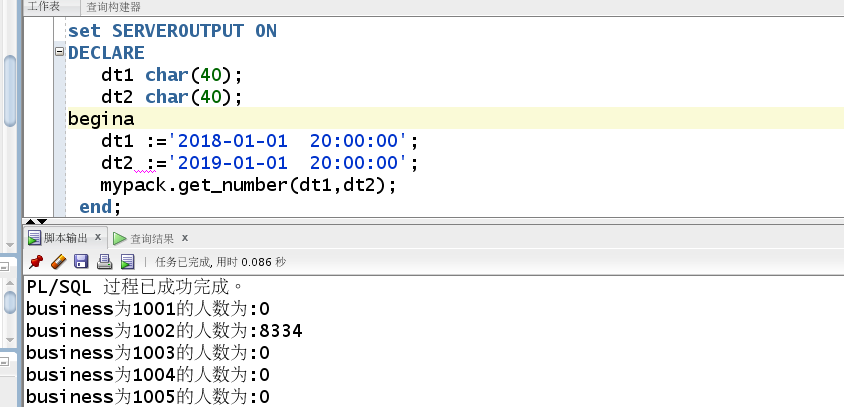
end mypack;



1.函数效果展示，如图所示



2.存储过程效果展示，如图所示



**第七章 系统备份**

1.开始全备份

[oracle@oracle-pc ~]$ cat rman\_level0.sh

#rman\_level0.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/lv0\_`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 0 database format '/home/oracle/rman\_backup/dblv0\_%d\_%T\_%U.bak'

plus archivelog format '/home/oracle/rman\_backup/arclv0\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

release channel c1;

}

EOF

exit

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 0 database format '/home/oracle/rman\_backup/dblv0\_%d\_%T\_%U.bak'

plus archivelog format '/home/oracle/rman\_backup/arclv0\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

release channel c1;

}

EOF

exit

2.每天定时开启备份



3.执行 ./rman\_level1.sh脚本对数据库进行备份



4.删除数据库文件



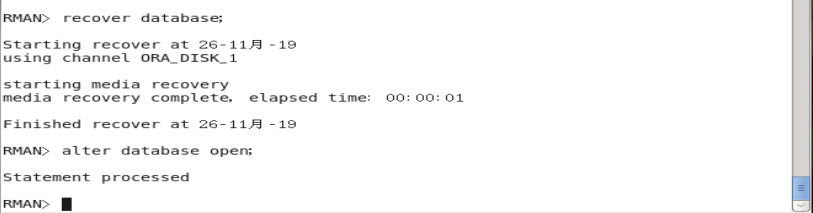


* 1. 重启损坏的数据库到mount状态

通过shutdown immediate无法正常关闭数据库，只能通过shutdown abort强制关闭。然后将数据库启动到mount状态。



* 1. 开始恢复数据库



* 1. 查看是否已经恢复



6.删除备份集

