

**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 | 计算机学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 陈耕 | 学号 | 201810414107 |
| 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

### 购物系统数据库表结构设计

#### USERS表（用户表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| ID | NUMBER(10, 0) | 主键 | 否 |  |  | 用户ID |
| USERNAME | VARCHAR2(50 BYTE) |  | 否 |  |  | 用户名 |
| PASSWORD | VARCHAR2(50 BYTE) |  | 否 |  |  | 密码 |
| ADDRESS | VARCHAR2(50 BYTE) |  | 否 |  |  | 收货地址 |
| TEL | VARCHAR2(20 BYTE) |  | 否 |  |  | 收货人电话 |
| CREATE\_TIME | DATE |  | 否 |  |  | 用户注册时间 |

#### GOODS表（商品表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| ID | NUMBER(10, 0) | 主键 | 否 |  |  | 商品ID |
| NAME | VARCHAR2(50 BYTE) |  | 否 |  |  | 商品名称 |
| PRICE | NUMBER(8, 2) |  | 否 |  |  | 商品单价 |
| PROFILE | VARCHAR2(100 BYTE) |  | 否 |  |  | 商品简介 |
| STOCK | NUMBER(10, 0) |  | 否 |  |  | 商品库存 |
| CATEGORY\_ID | NUMBER(10, 0) | 外键 | 否 |  |  | 类别ID |

#### CATEGORY表（商品类别表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| ID | NUMBER(10, 0) | 主键 | 否 |  |  | 类别ID |
| NAME | VARCHAR2(20 BYTE) |  | 否 |  |  | 类别名称 |

#### ORDERS表（订单表）

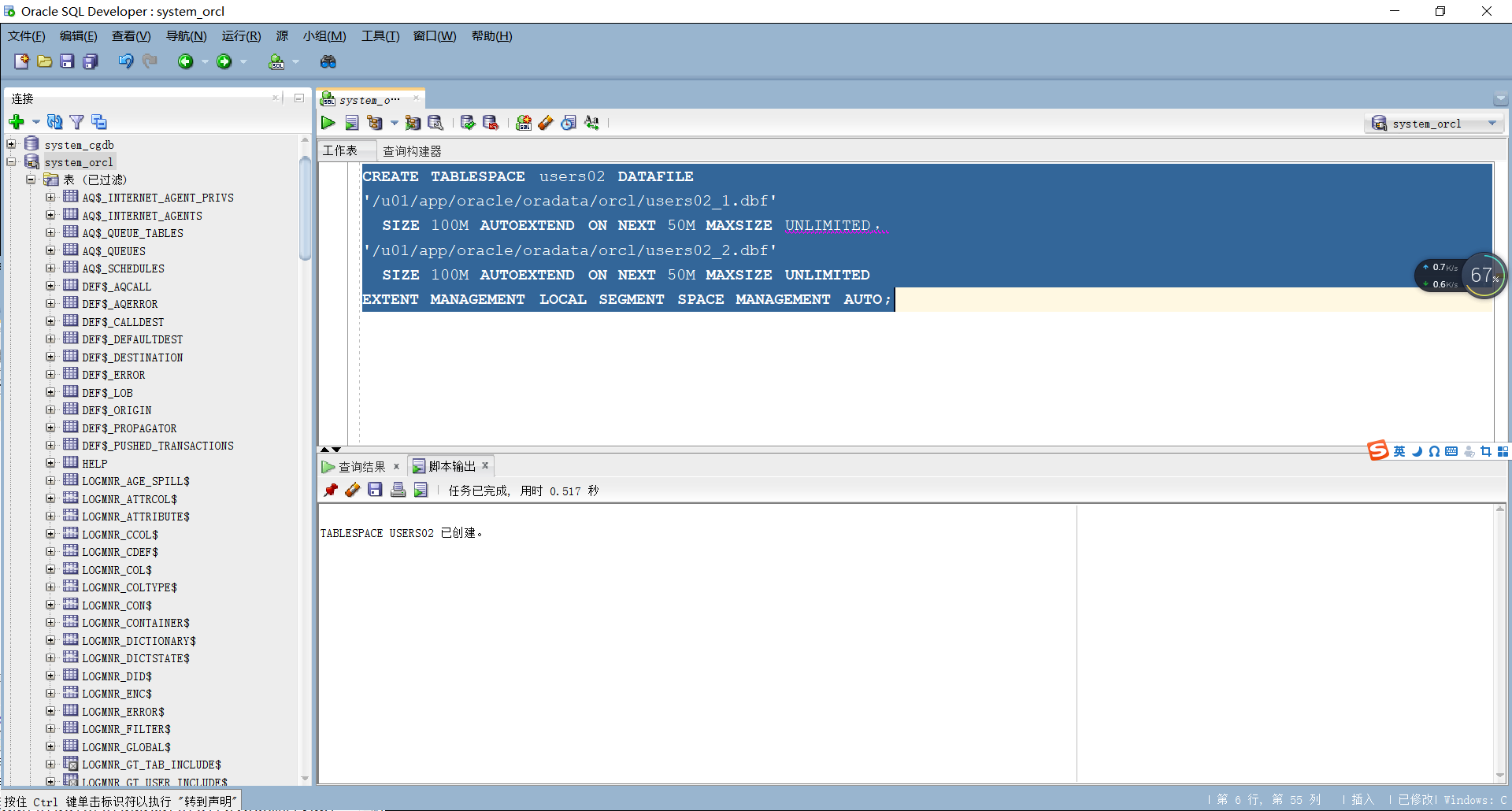
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| ID | NUMBER(10, 0) | 主键 | 否 |  |  | 订单ID |
| USER\_ID | NUMBER(10, 0) | 外键 | 否 |  |  | 用户ID |
| TOTAL\_PRICE | NUMBER(8, 2) |  | 否 |  |  | 订单总金额 |
| CREATE\_TIME | DATE |  | 否 |  |  | 订单创建时间 |
| STATUS | VARCHAR2(20 BYTE) |  | 否 | 未支付 |  | 订单状态 |

#### ORDER\_GOODS表（订单商品关系表）

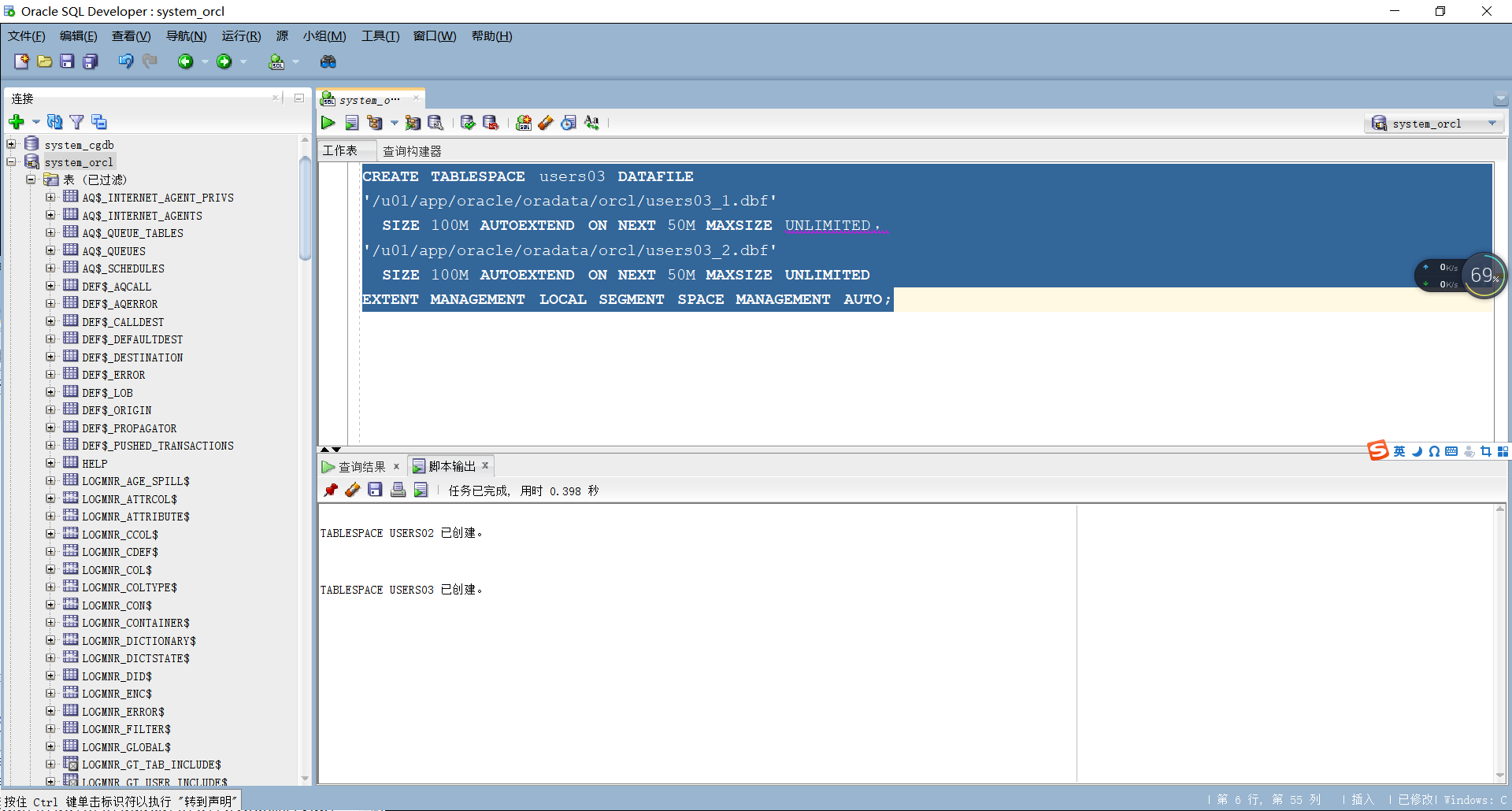
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| ID | NUMBER(10, 0) | 主键 | 否 |  |  | 唯一标识 |
| ORDER\_ID | NUMBER(10, 0) | 外键 | 否 |  |  | 订单ID |
| GOODS\_ID | NUMBER(10, 0) | 外键 | 否 |  |  | 商品ID |
| GOODS\_NUM | NUMBER(10, 0) |  | 否 |  |  | 商品数量 |

### 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

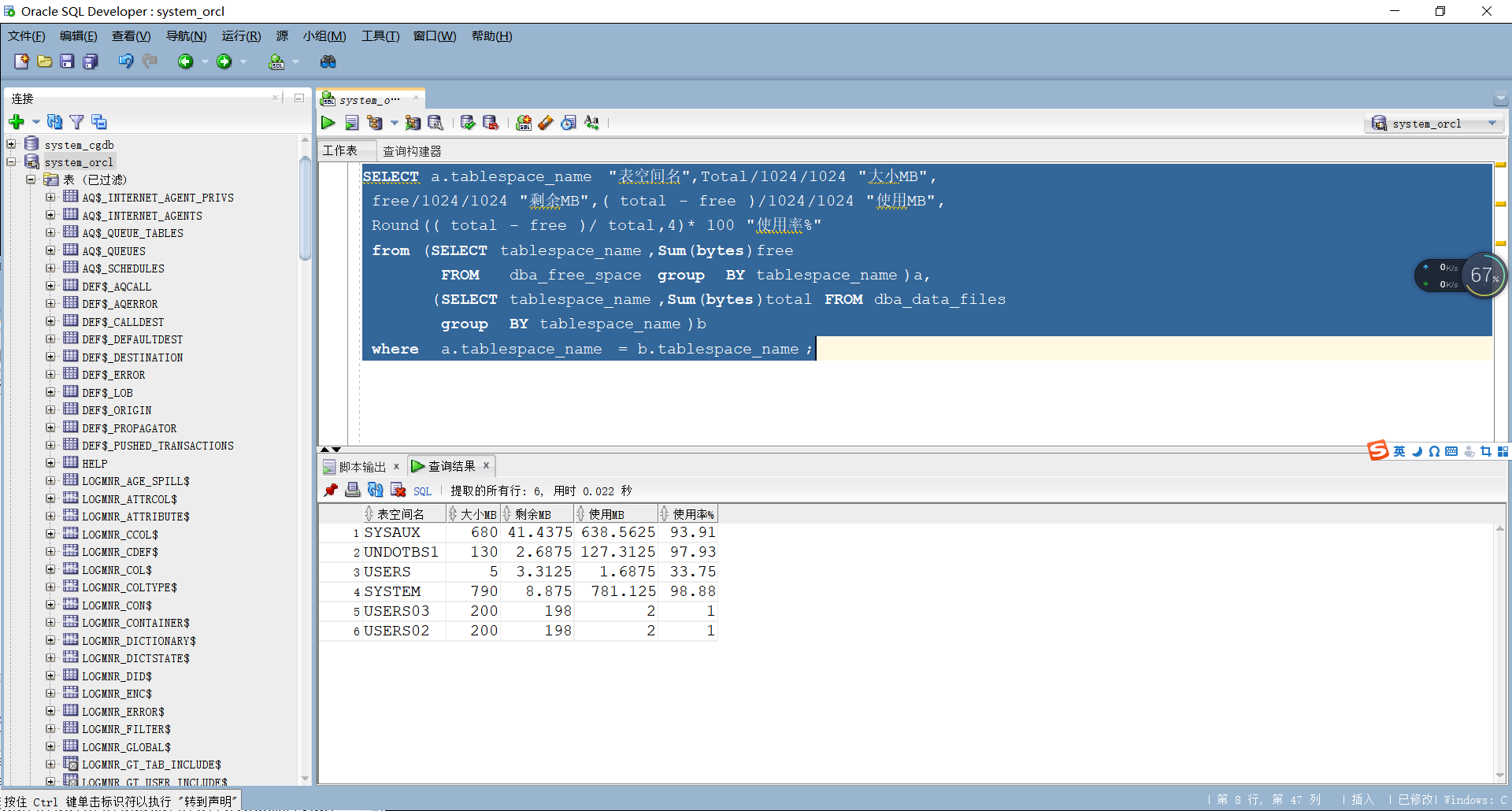
#### 创建表空间users02



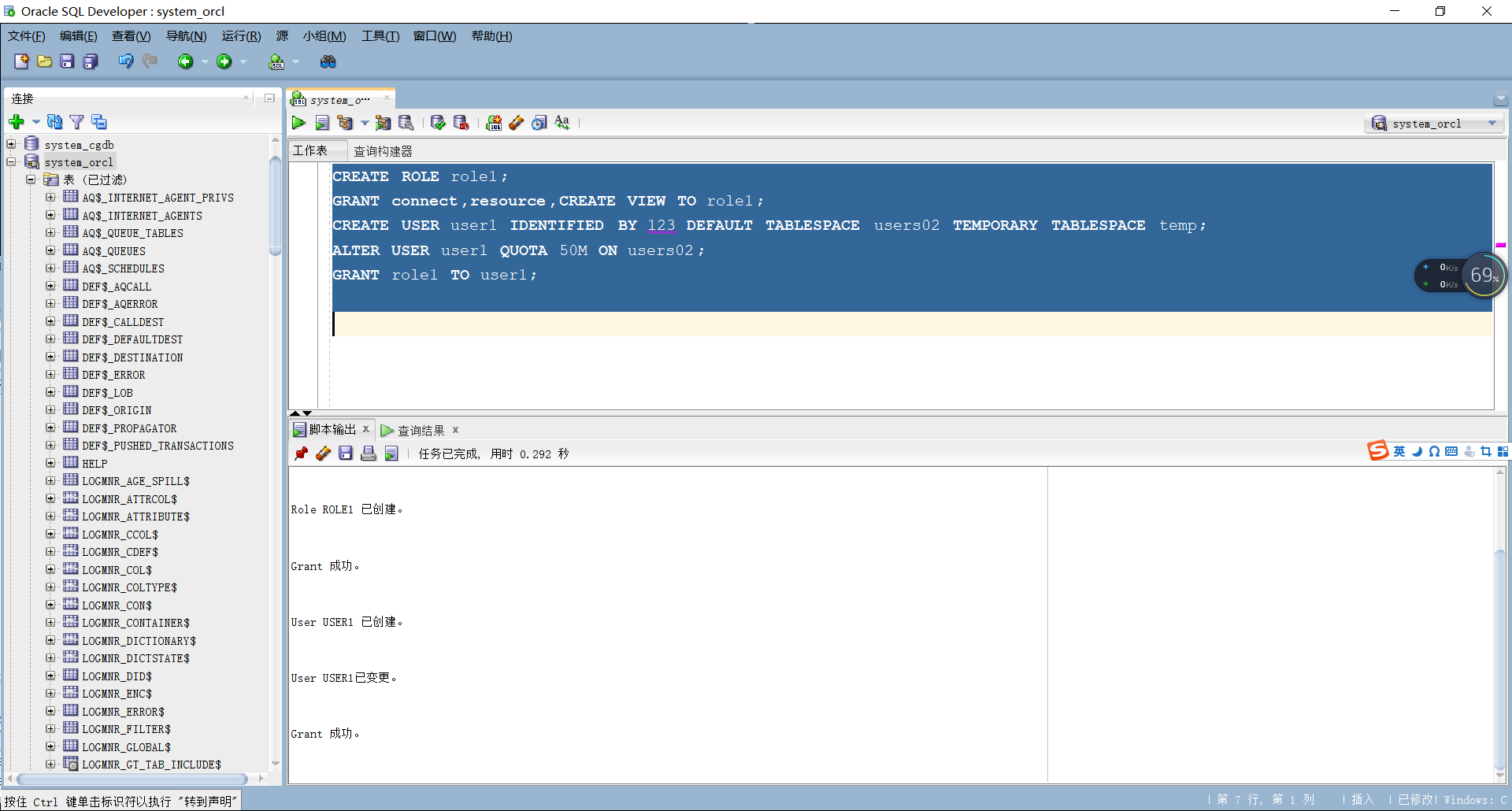
#### 创建表空间users03

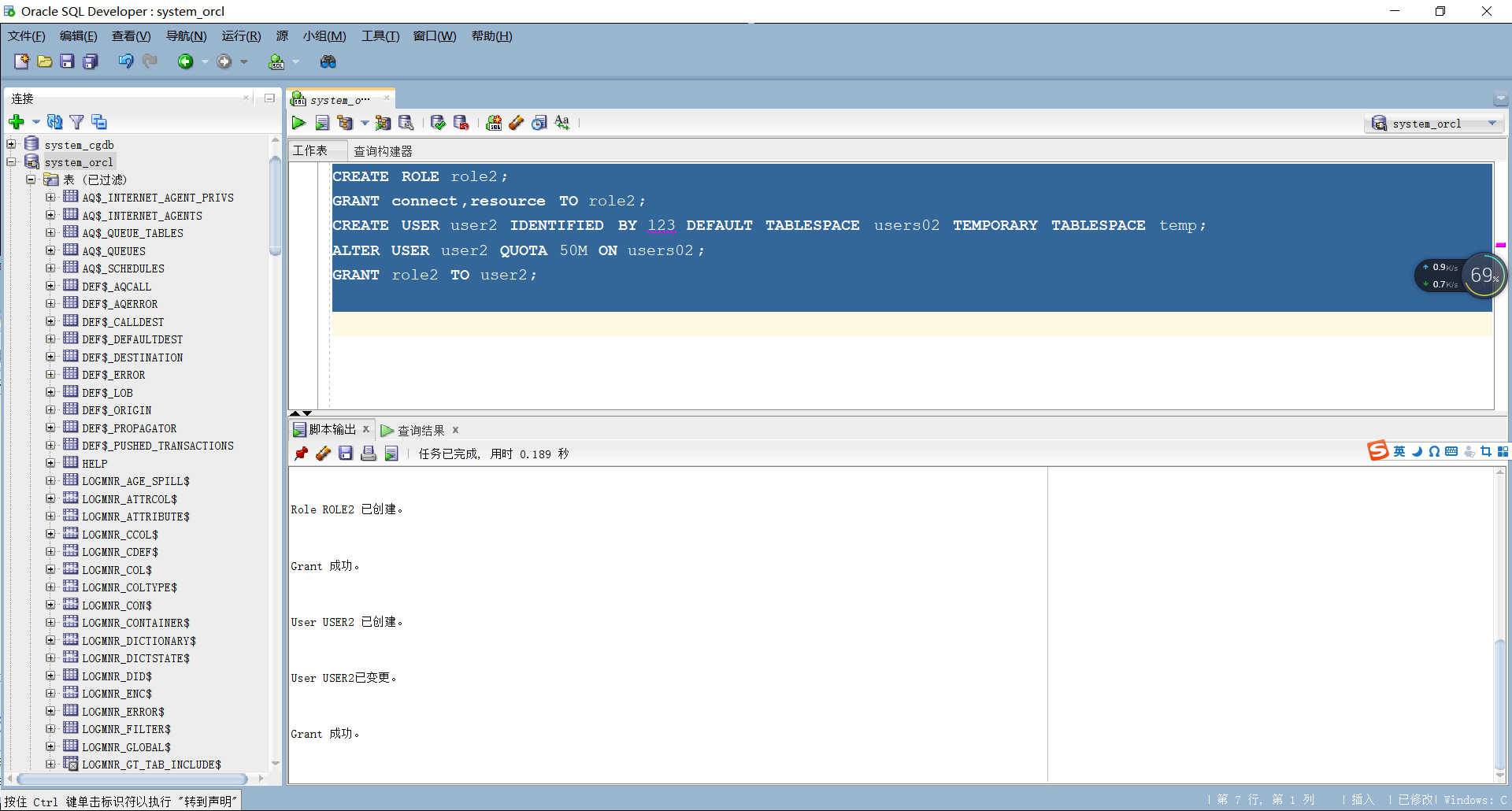


#### 查看表空间的数据库文件，以及每个文件的磁盘占用情况



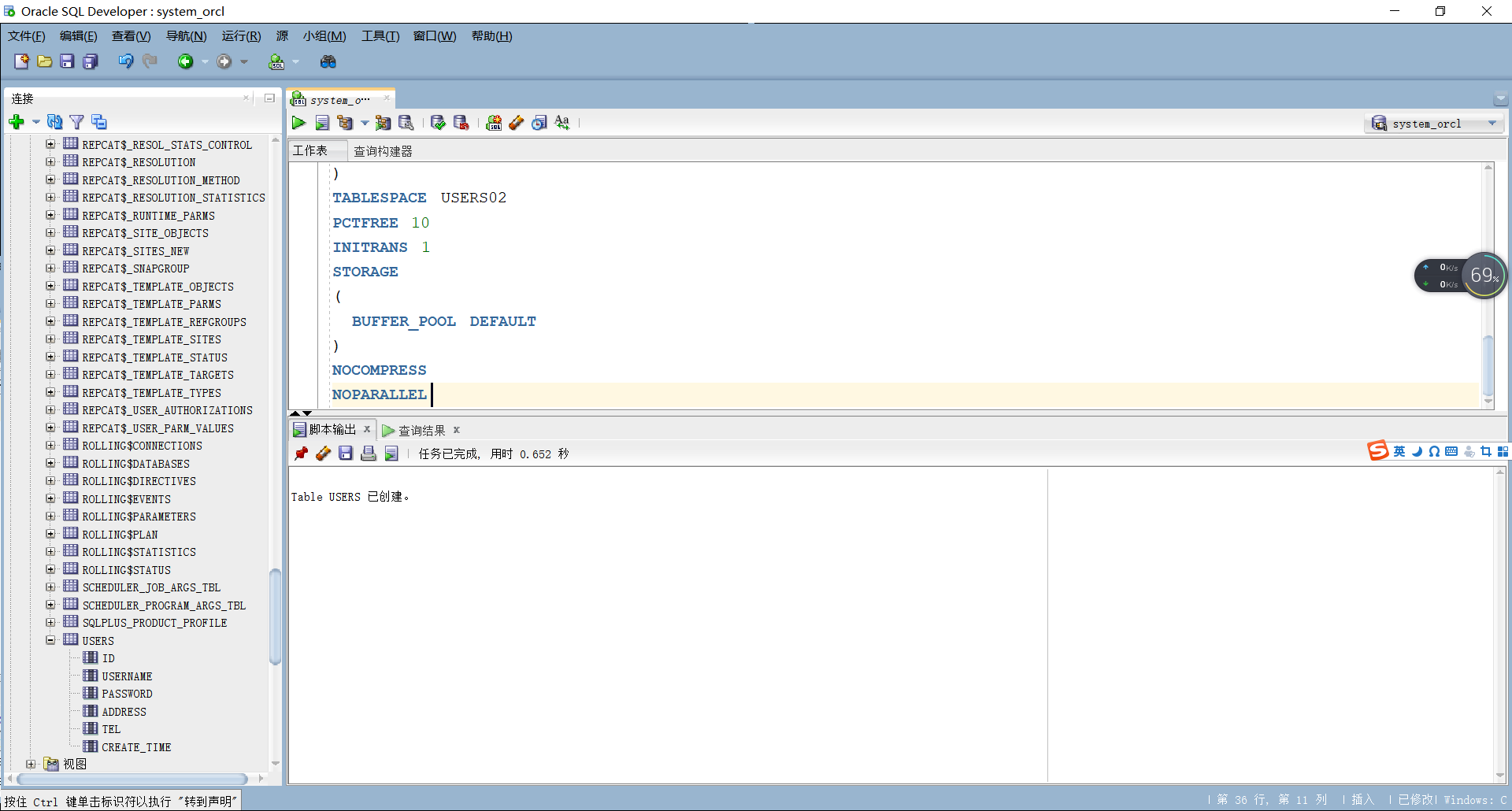
#### 设计权限及用户分配方案。至少两类角色，两个用户

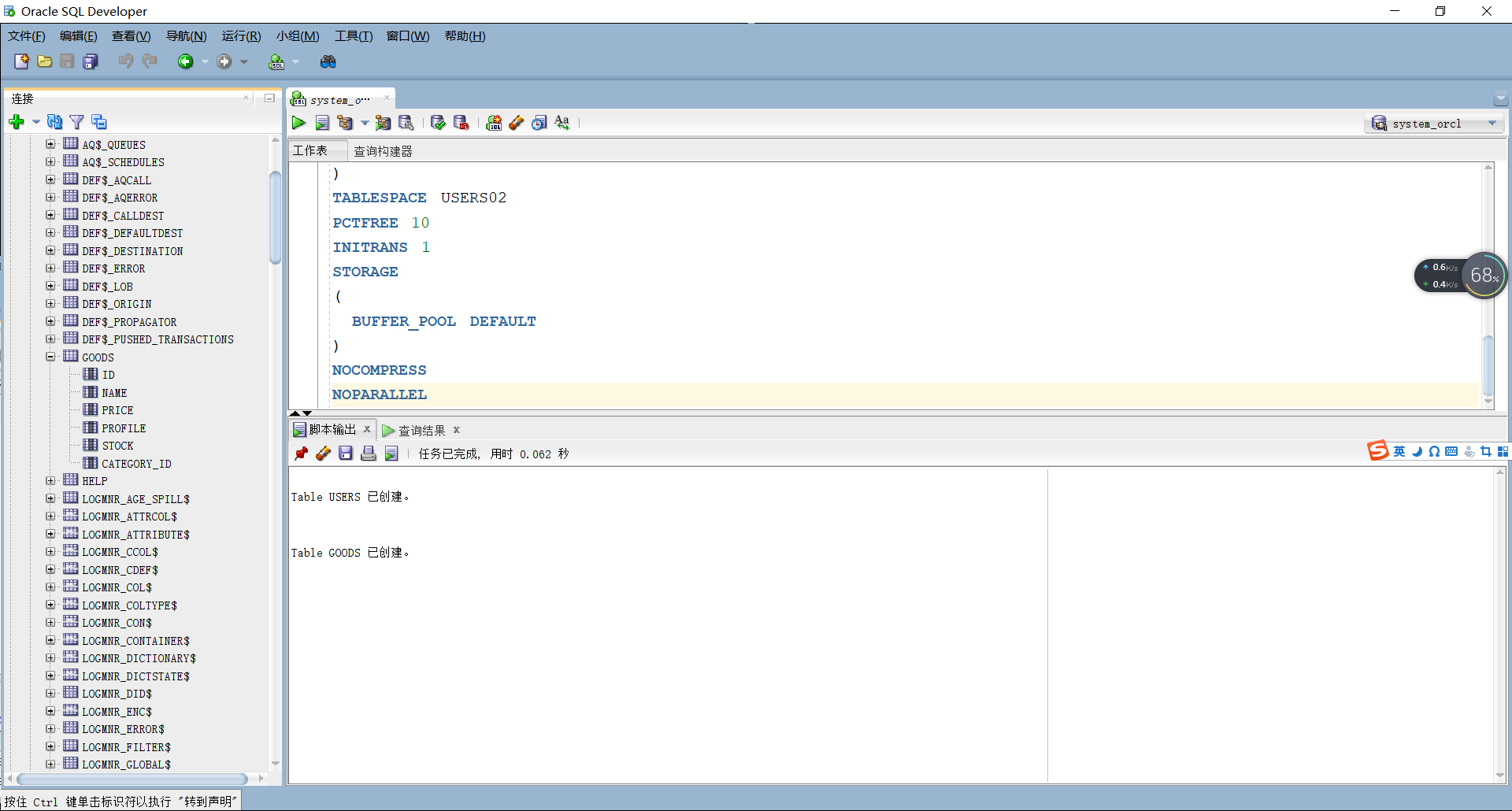
以system用户登录到orcl，创建角色role1和用户user1，并授权和分配表空间  


以system用户登录到orcl，创建角色role2和用户user2，并授权和分配表空间  


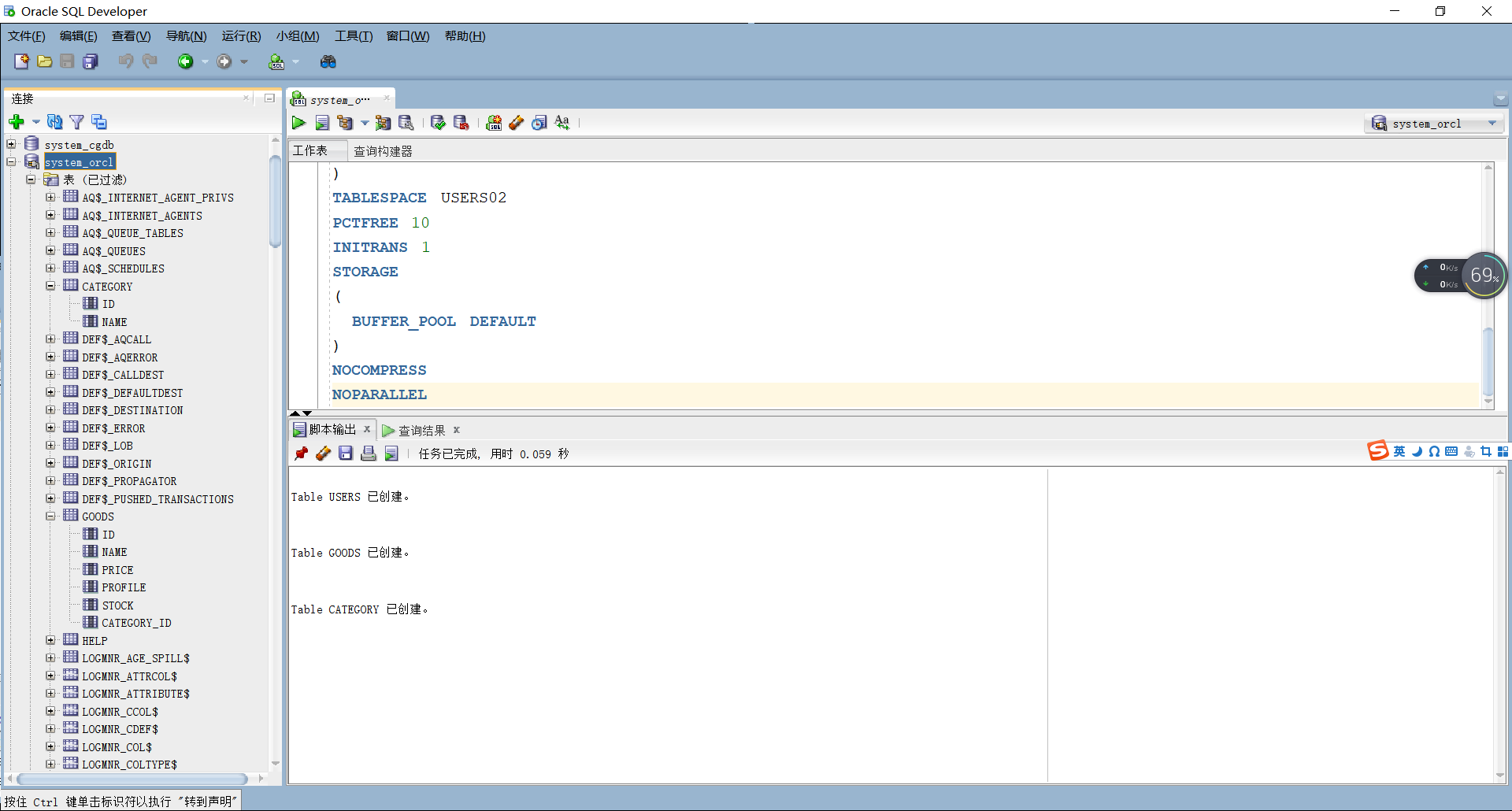
#### 创建表

创建USERS表

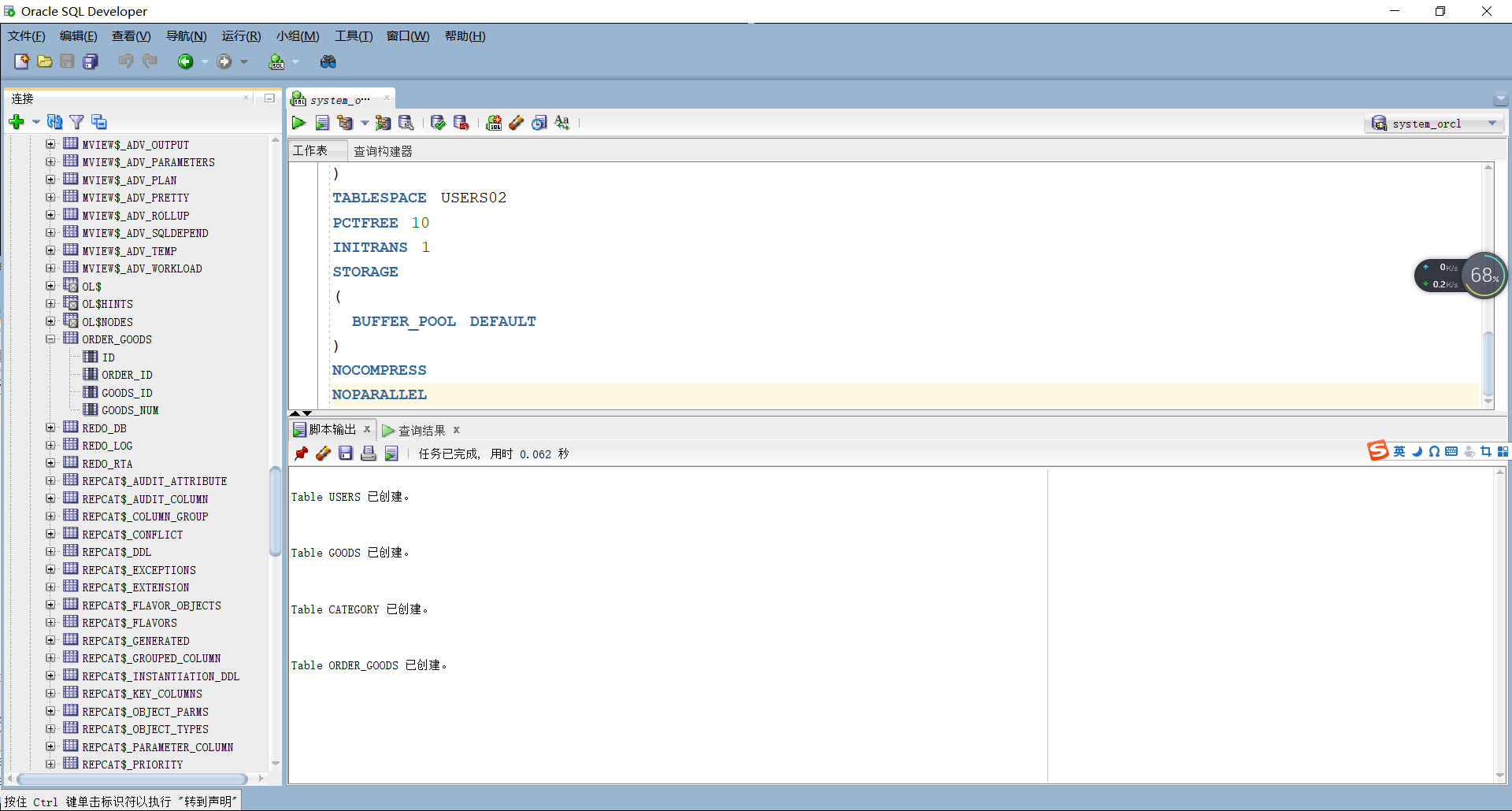
CREATE TABLE USERS  
 (  
 ID NUMBER(10, 0) NOT NULL  
 , USERNAME VARCHAR2(50 BYTE) NOT NULL  
 , PASSWORD VARCHAR2(50 BYTE) NOT NULL  
 , ADDRESS VARCHAR2(50 BYTE) NOT NULL  
 , TEL VARCHAR2(20 BYTE) NOT NULL  
 , CREATE\_TIME DATE NOT NULL  
 , CONSTRAINT USER\_PK PRIMARY KEY  
 (  
 ID  
 )  
 USING INDEX  
 (  
 CREATE UNIQUE INDEX USER\_PK ON USERS (ID ASC)  
 LOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
 )  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 1  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOCOMPRESS  
 NOPARALLEL  


创建GOODS表  
CREATE TABLE GOODS  
(  
 ID NUMBER(10, 0) NOT NULL  
, NAME VARCHAR2(50 BYTE) NOT NULL  
, PRICE NUMBER(8, 2) NOT NULL  
, PROFILE VARCHAR2(100 BYTE) NOT NULL  
, STOCK NUMBER(10, 0) NOT NULL  
, CATEGORY\_ID NUMBER(10, 0) NOT NULL  
, CONSTRAINT GOODS\_PK PRIMARY KEY  
 (  
 ID  
 )  
 USING INDEX  
 (  
 CREATE UNIQUE INDEX GOODS\_PK ON GOODS (ID ASC)  
 LOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
TABLESPACE USERS02  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NOPARALLEL  


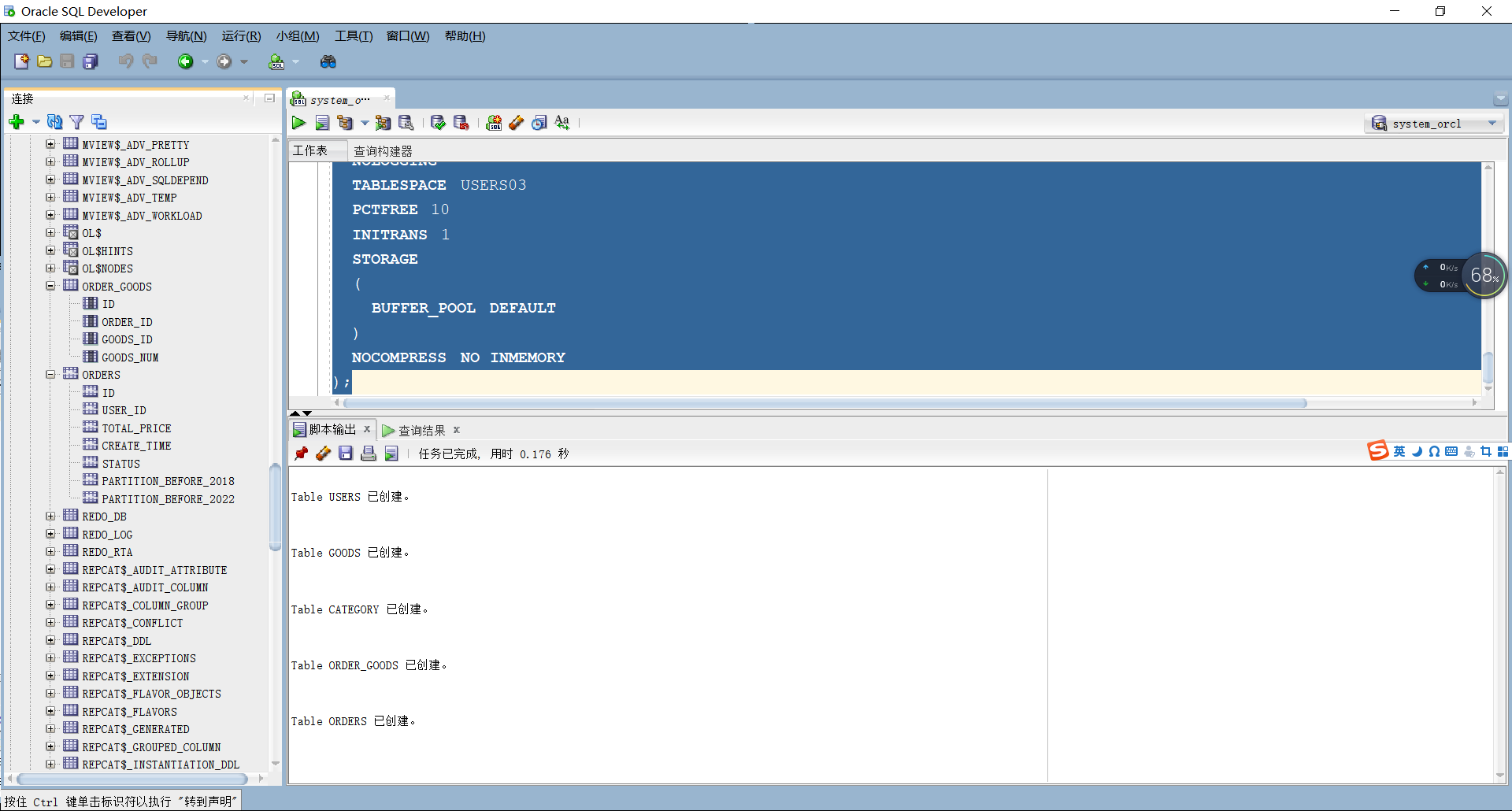
创建CATEGORY表

CREATE TABLE CATEGORY  
(  
 ID NUMBER(10, 0) NOT NULL  
, NAME VARCHAR2(20 BYTE) NOT NULL  
, CONSTRAINT CATEGORY\_PK PRIMARY KEY  
 (  
 ID  
 )  
 USING INDEX  
 (  
 CREATE UNIQUE INDEX CATEGORY\_PK ON CATEGORY (ID ASC)  
 LOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
TABLESPACE USERS02  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NOPARALLEL  


创建ORDER\_GOODS表

CREATE TABLE ORDER\_GOODS  
(  
 ID NUMBER(10, 0) NOT NULL  
, ORDER\_ID NUMBER(10, 0) NOT NULL  
, GOODS\_ID NUMBER(10, 0) NOT NULL  
, GOODS\_NUM NUMBER(10, 0) NOT NULL  
, CONSTRAINT ORDER\_GOODS\_PK PRIMARY KEY  
 (  
 ID  
 )  
 USING INDEX  
 (  
 CREATE UNIQUE INDEX ORDER\_GOODS\_PK ON ORDER\_GOODS (ID ASC)  
 LOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
TABLESPACE USERS02  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NOPARALLEL  


创建ORDERS表

CREATE TABLE ORDERS  
(  
 ID NUMBER(10, 0) NOT NULL  
, USER\_ID NUMBER(10, 0) NOT NULL  
, TOTAL\_PRICE NUMBER(8, 2) NOT NULL  
, CREATE\_TIME DATE NOT NULL  
, STATUS VARCHAR2(20 BYTE) NOT NULL  
, CONSTRAINT ORDERS\_PK PRIMARY KEY  
 (  
 ID  
 )  
 USING INDEX  
 (  
 CREATE UNIQUE INDEX ORDERS\_PK ON ORDERS (ID ASC)  
 LOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
TABLESPACE USERS02  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NOPARALLEL  
PARTITION BY RANGE (CREATE\_TIME)  
(  
 PARTITION PARTITION\_BEFORE\_2018 VALUES LESS THAN (TO\_DATE(' 2018-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))  
 NOLOGGING  
 TABLESPACE USERS02  
 PCTFREE 10  
 INITRANS 1  
 STORAGE  
 (  
 INITIAL 8388608  
 NEXT 1048576  
 MINEXTENTS 1  
 MAXEXTENTS UNLIMITED  
 BUFFER\_POOL DEFAULT  
 )  
 NOCOMPRESS NO INMEMORY  
, PARTITION PARTITION\_BEFORE\_2022 VALUES LESS THAN (TO\_DATE(' 2022-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))  
 NOLOGGING  
 TABLESPACE USERS03  
 PCTFREE 10  
 INITRANS 1  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOCOMPRESS NO INMEMORY  
);  


#### 添加模拟数据

添加商品模拟数据

declare  
 v\_id NUMBER(10,0);  
 v\_name VARCHAR2(50);  
 v\_price NUMBER(8,2);  
 v\_profile VARCHAR2(100);  
 v\_stock NUMBER(10,0);  
 v\_cate\_id NUMBER(10,0);  
begin  
 for i in 1..1000  
 loop  
 v\_id:=i;  
 v\_name:='商品'||i;  
 v\_price:=i mod 500 + 1;  
 v\_profile:='这是商品'||i||'的简介';  
 v\_stock:=i mod 200 + 100;  
 v\_cate\_id:=i mod 10 + 1;  
 insert into GOODS(ID,NAME,PRICE,PROFILE,STOCK,CATEGORY\_ID)  
 values(v\_id,v\_name,v\_price,v\_profile,v\_stock,v\_cate\_id);  
 end loop;  
end;

添加商品类别模拟数据

declare  
 v\_id NUMBER(10,0);  
 v\_name VARCHAR2(20);  
begin  
 for i in 1..10  
 loop  
 v\_id:=i;  
 v\_name:='类别'||i;  
 insert into CATEGORY(ID,NAME)  
 values(v\_id,v\_name);  
 end loop;  
end;

添加用户模拟数据

declare  
 v\_id NUMBER(10,0);  
 v\_uname VARCHAR2(50);  
 v\_pwd VARCHAR2(50);  
 v\_addr VARCHAR2(50);  
 v\_tel VARCHAR2(20);  
 v\_ct DATE;  
begin  
 for i in 1..10000  
 loop  
 v\_id:=i;  
 v\_uname:='用户'||i;  
 v\_pwd:='123';  
 v\_addr:='这是用户'||i||'的收货地址';  
 v\_tel:=concat('1875684',((i mod 9000)+1000));  
 v\_ct:=to\_date('2015-1-5','yyyy-mm-dd')+(i mod 60);  
 insert into USERS(ID,USERNAME,PASSWORD,ADDRESS,TEL,CREATE\_TIME)  
 values(v\_id,v\_uname,v\_pwd,v\_addr,v\_tel,v\_ct);  
 end loop;  
end;

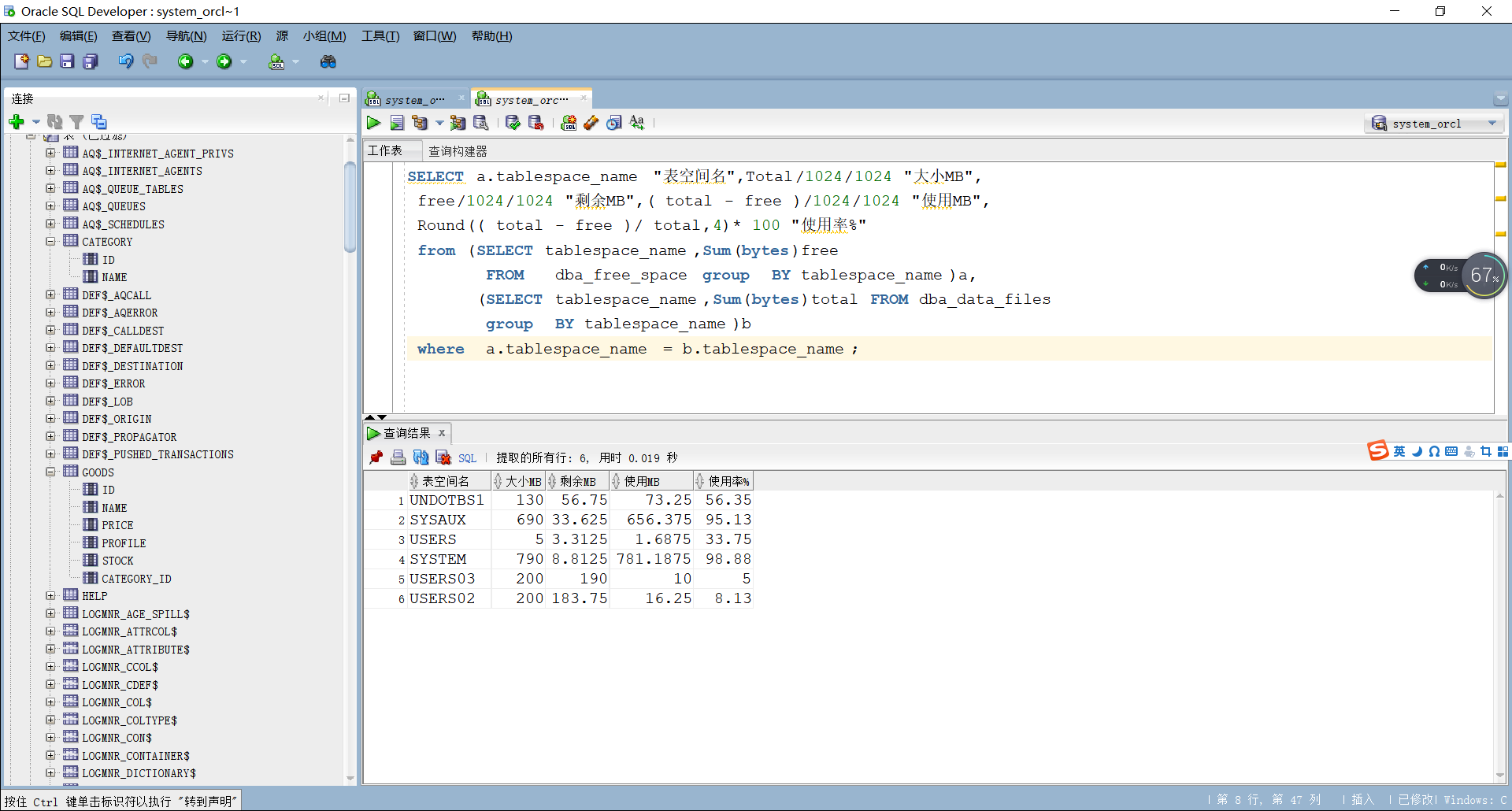
添加订单商品关系模拟数据

declare  
 v\_id NUMBER(10,0);  
 v\_oid NUMBER(10,0);  
 v\_gid NUMBER(10,0);  
 v\_gnum NUMBER(10,0);  
begin  
 for i in 1..50000  
 loop  
 v\_id:=i;  
 v\_oid:=i;  
 v\_gid:=(i mod 1000)+1;  
 select round(dbms\_random.value(1,5)) into v\_gnum from dual;  
 insert into ORDER\_GOODS(ID,ORDER\_ID,GOODS\_ID,GOODS\_NUM)  
 values(v\_id,v\_oid,v\_gid,v\_gnum);  
 end loop;  
end;

添加订单模拟数据

declare  
 v\_id NUMBER(10,0);  
 v\_uid NUMBER(10,0);  
 v\_tp NUMBER(8,2);  
 v\_ct DATE;  
 v\_status VARCHAR2(20);  
begin  
 for i in 1..50000  
 loop  
 v\_id:=i;  
 v\_uid:=(i mod 10000);  
 if (i mod 2)=0 then  
 v\_status:='已支付';  
 v\_ct:=to\_date('2017-5-8','yyyy-mm-dd') + (i mod 60);  
 else  
 v\_status:='未支付';  
 v\_ct:=to\_date('2021-5-8','yyyy-mm-dd') + (i mod 60);  
 end if;  
 select DISTINCT g.price\*og.goods\_num into v\_tp  
 from order\_goods og, goods g  
 where og.goods\_id=g.id and og.order\_id=i;  
 insert into ORDERS(ID,USER\_ID,TOTAL\_PRICE,CREATE\_TIME,STATUS)  
 values(v\_id,v\_uid,v\_tp,v\_ct,v\_status);  
 end loop;  
end;

#### 再次查看表空间使用情况



### 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

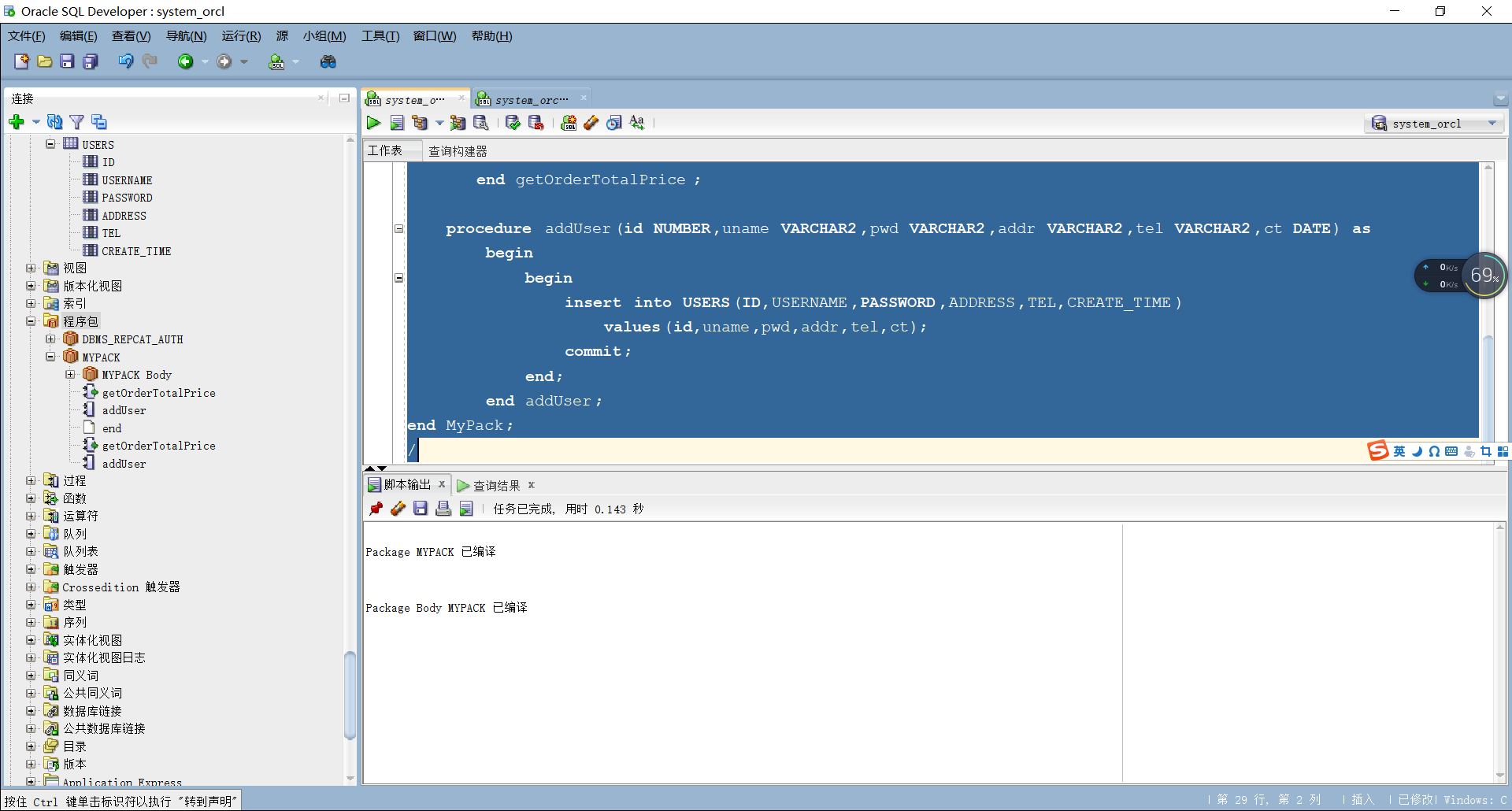
#### 创建程序包

create or replace PACKAGE MyPack Is  
 function getOrderTotalPrice(order\_id number) return number;  
 procedure addUser(id NUMBER,uname VARCHAR2,pwd VARCHAR2,addr VARCHAR2,tel VARCHAR2,ct DATE);  
end MyPack;  
/

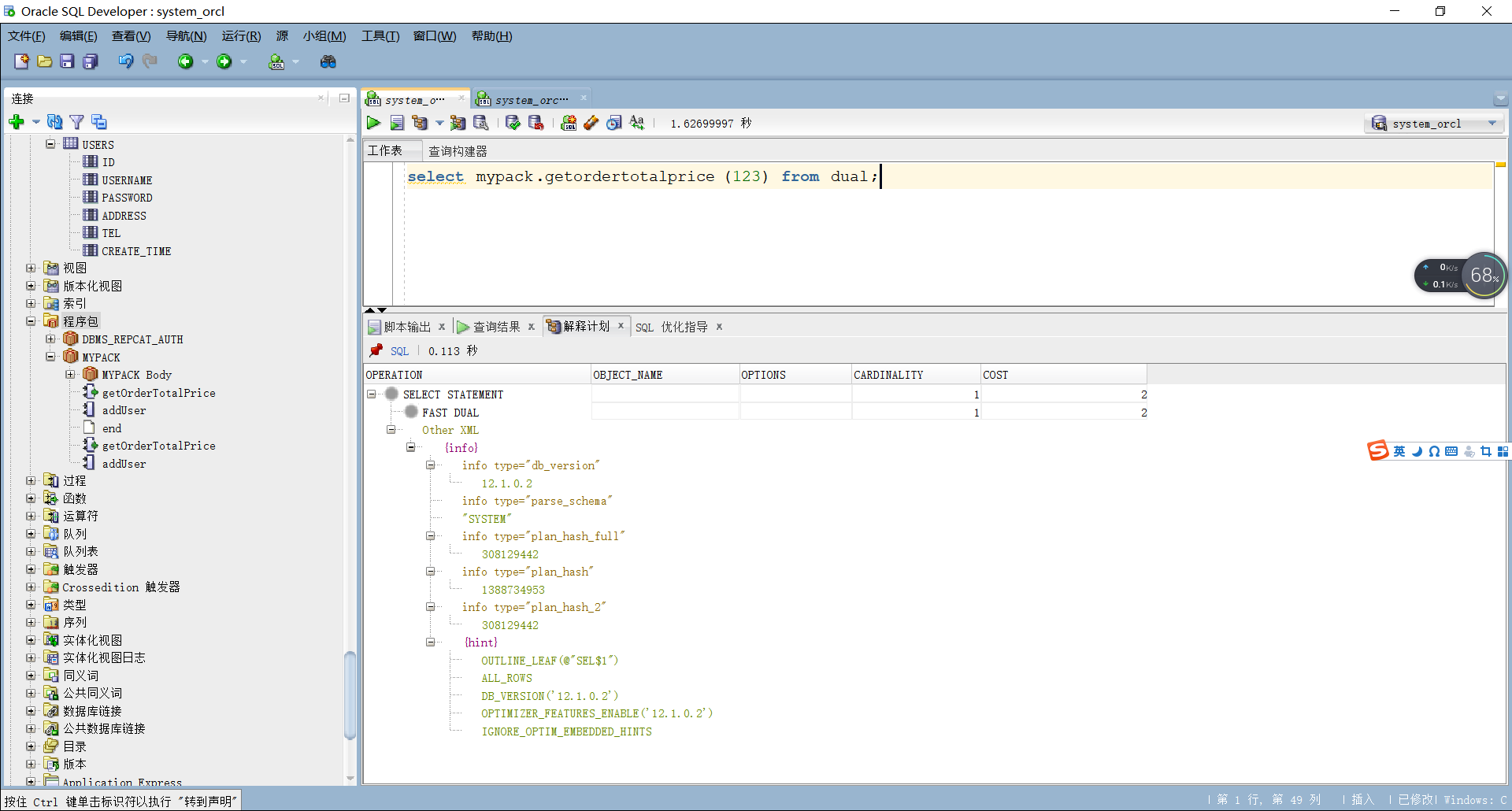
create or replace PACKAGE body MyPack Is  
 function getOrderTotalPrice(order\_id number) return number as  
 begin  
 declare v\_tp number;  
 query\_sql varchar2(200);  
 begin  
 query\_sql:='select DISTINCT g.price\*og.goods\_num  
 from order\_goods og, goods g  
 where og.goods\_id=g.id and og.order\_id='||order\_id;  
 execute immediate query\_sql into v\_tp;  
 return v\_tp;  
 end;  
 end getOrderTotalPrice;

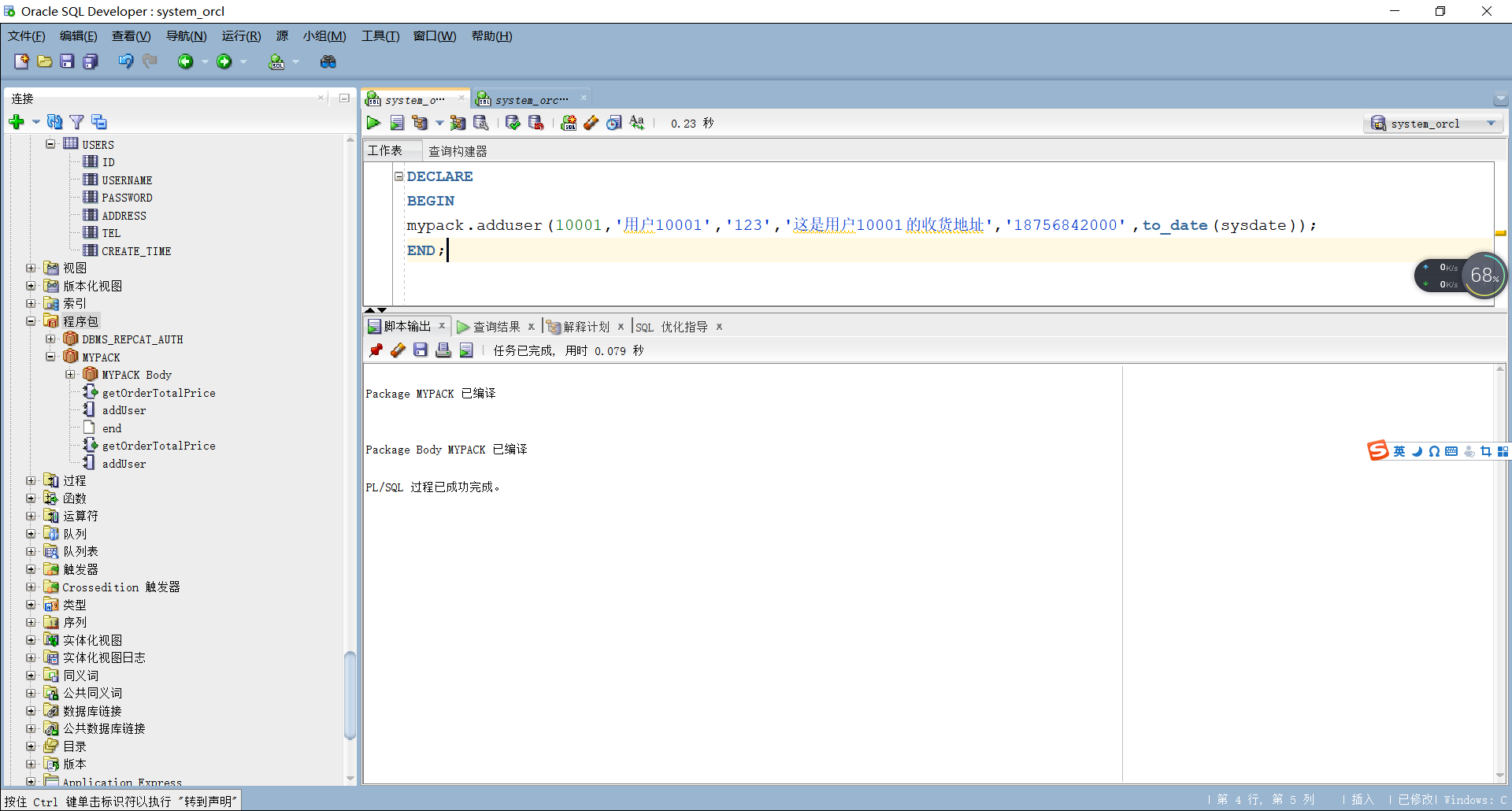
procedure addUser(id NUMBER,uname VARCHAR2,pwd VARCHAR2,addr VARCHAR2,tel VARCHAR2,ct DATE) as  
 begin  
 begin  
 insert into USERS(ID,USERNAME,PASSWORD,ADDRESS,TEL,CREATE\_TIME)  
 values(id,uname,pwd,addr,tel,ct);  
 commit;  
 end;  
 end addUser;

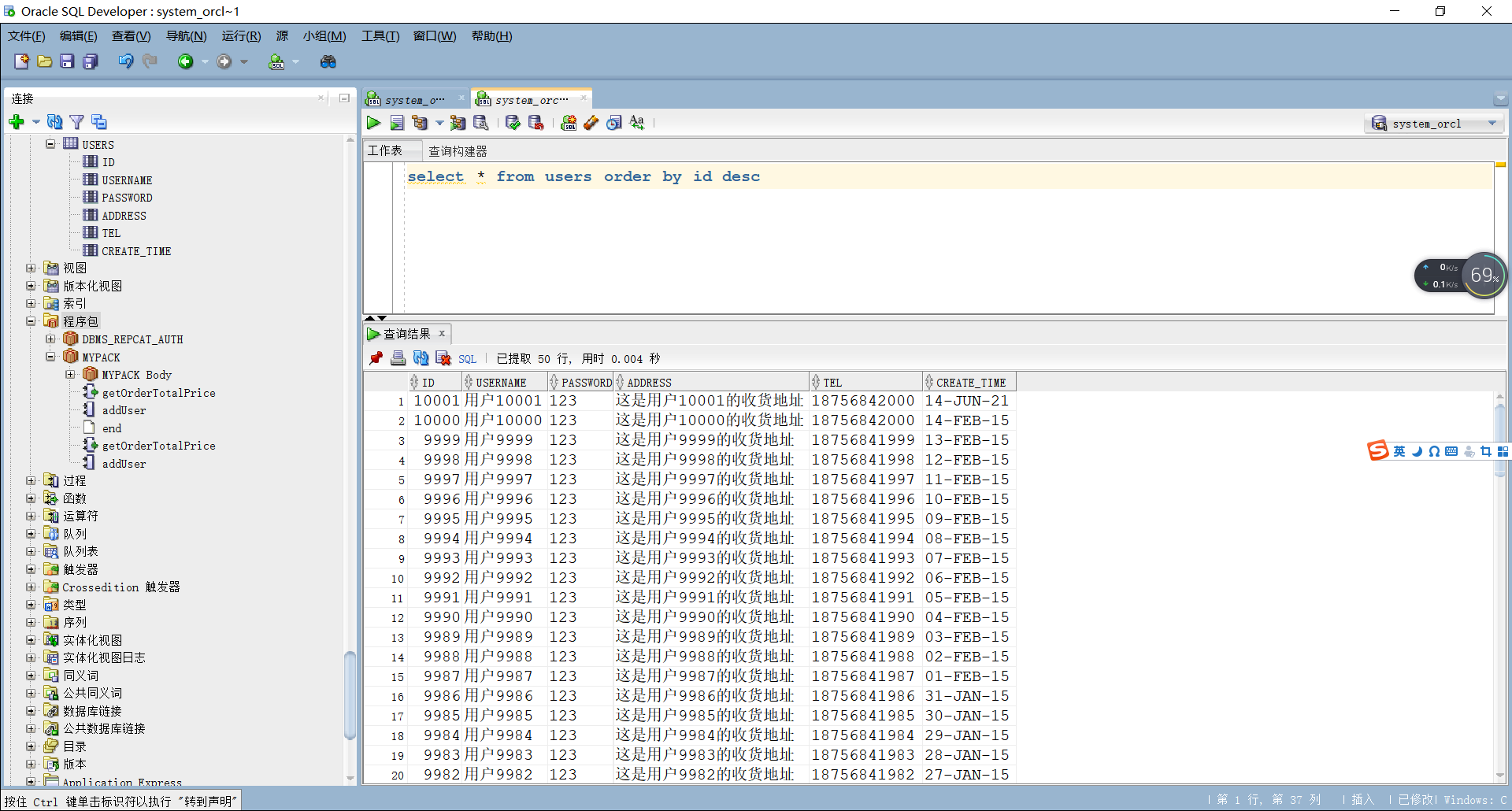
end MyPack;  
/



#### 执行计划分析

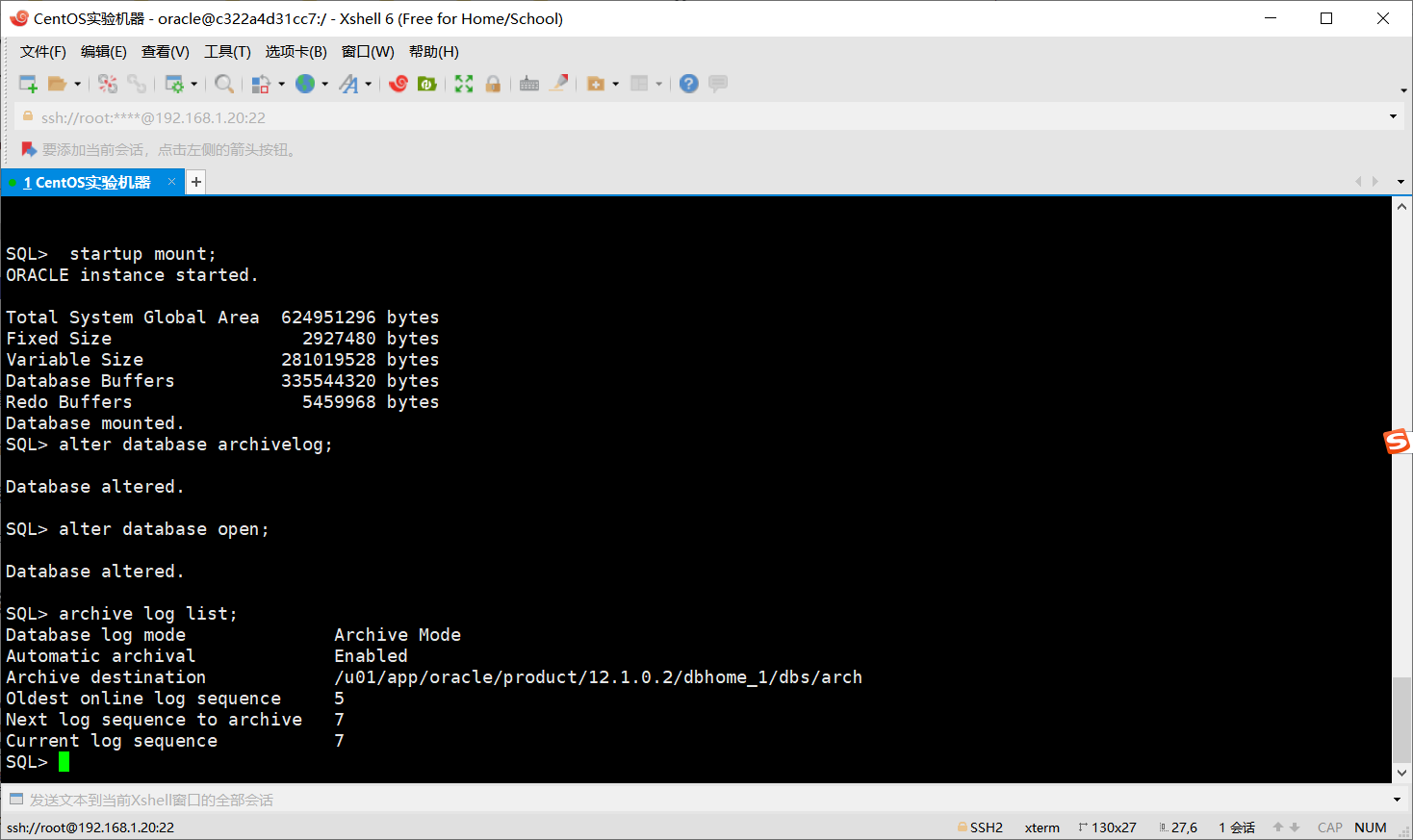




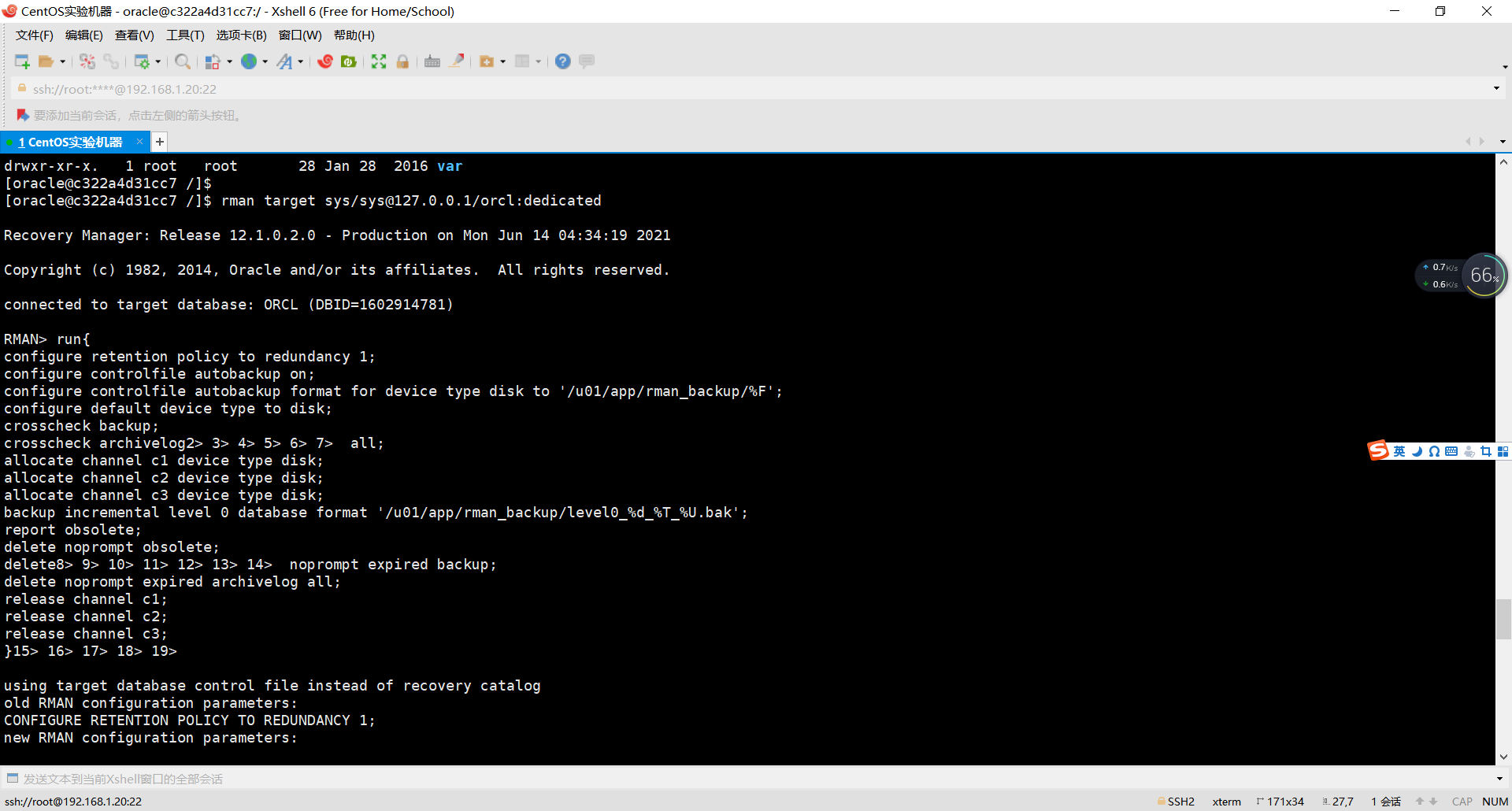


### 设计自动备份方案或则手工备份方案

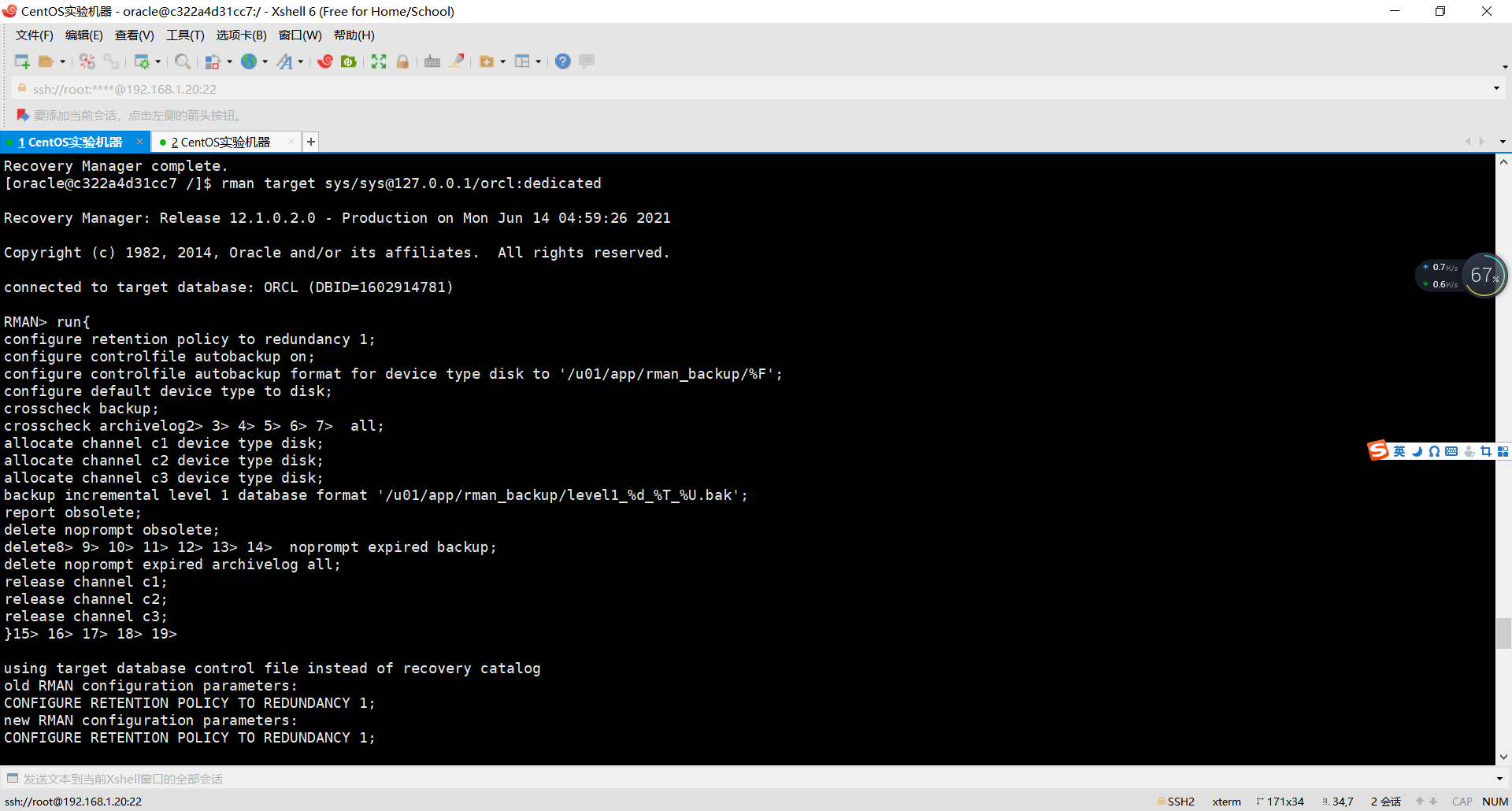
#### Oracle切换到归档模式



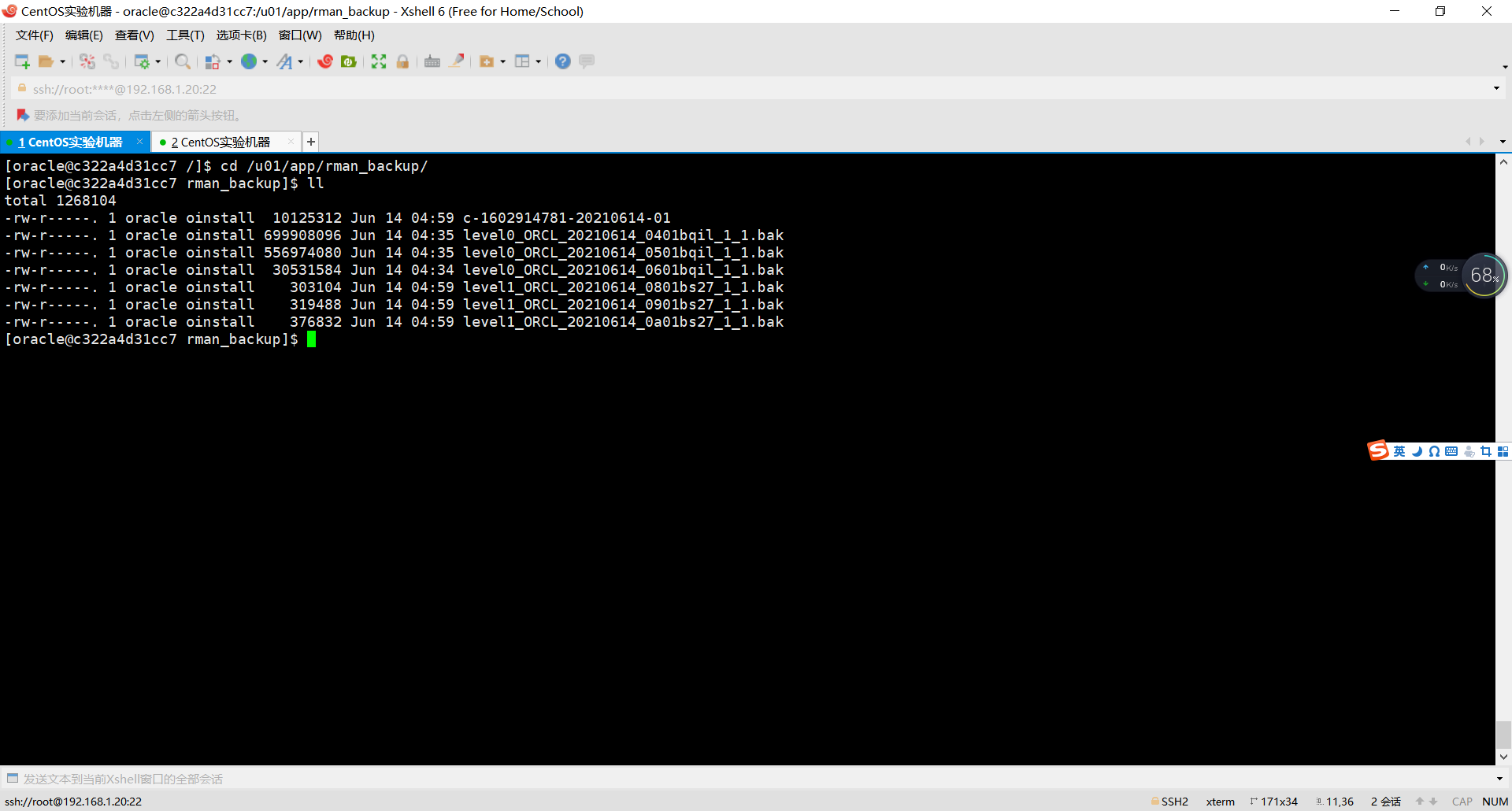
#### 全库0级备份(只作一次)



#### 全库1级增量备份



#### 查看备份文件



#### 全库完全恢复

全库停机  
rman target /  
RMAN> shutdown immediate;  
RMAN> exit

文件改名，模拟文件损失

mv /u01/app/oracle/oradata/orcl/users02\_1.dbf /u01/app/oracle/oradata/orcl/users02\_1.dbf2

全库恢复  
rman target /  
RMAN> startup mount;  
RMAN> restore database;  
RMAN> recover database;  
RMAN> alter database open;

