

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的书籍交易系统的数据库设计 | | |
| 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 | 吴梦佳 | 学号 | 201810414406 |
| 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

**实验6（期末考核）基于Oracle的书籍交易系统数据库设计**

**基于Oracle的书籍交易系统数据库设计**

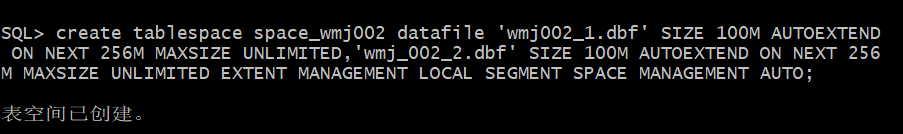
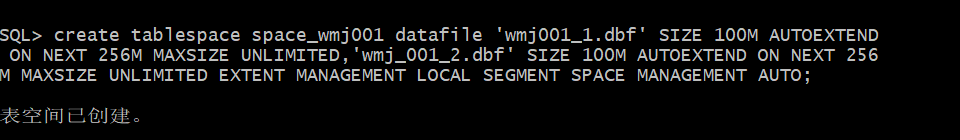
**1.创建表空间**

* space\_wmj001

create tablespace space\_wmj001 datafile 'wmj001\_1.dbf' SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,'wmj\_001\_2.dbf' SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

* space\_wmj002

 create tablespace space\_wmj002 datafile 'wmj002\_1.dbf' SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,'wmj\_002\_2.dbf' SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

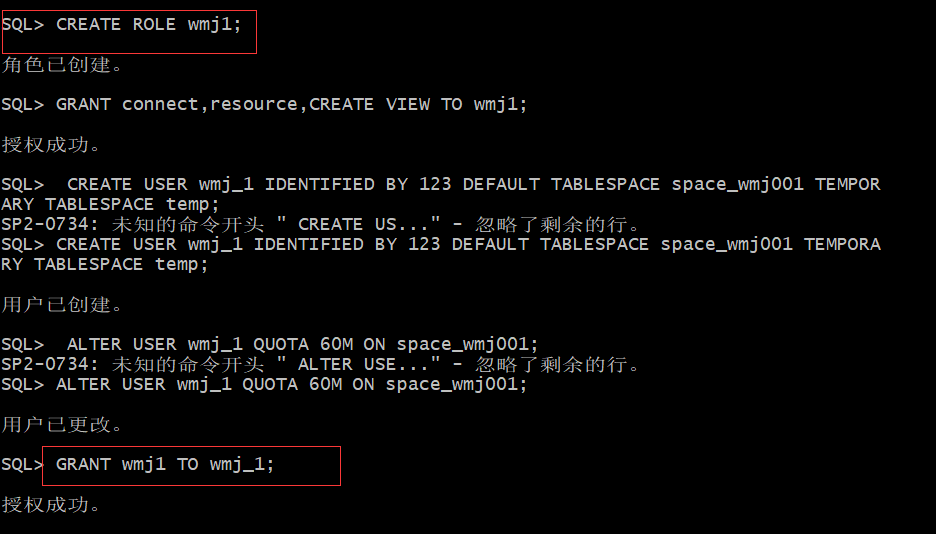


**2. 创建角色及用户**

**用户默认使用表空间space\_wmj001** **创建第一个角色和用户**

* 创建角色wmj1将connect,resource,create view授权给wmj1
* 创建用户wmj\_1
* 分配60M空间给wmj\_1并将角色wmj1授权给用户wmj\_1

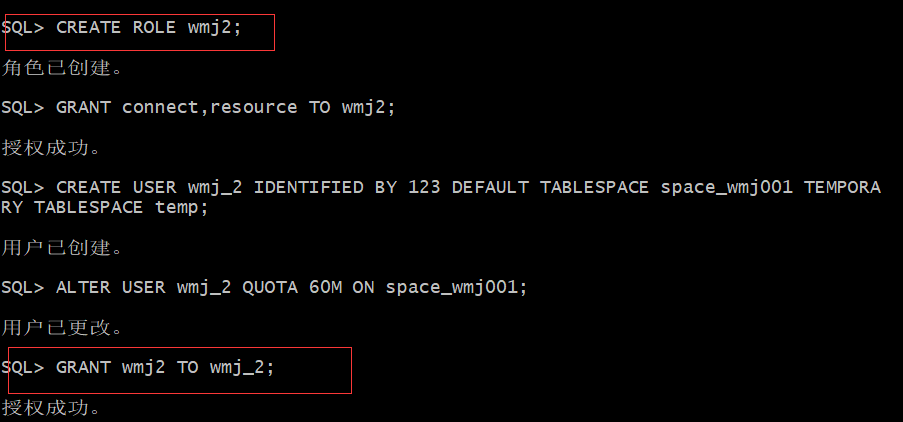
 CREATE ROLE wmj1;  
 ​  
 GRANT connect,resource,CREATE VIEW TO wmj1;  
 ​  
 CREATE USER wmj\_1 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_wmj001 TEMPORARY TABLESPACE temp;  
 ​  
 ALTER USER wmj\_1 QUOTA 60M ON space\_wmj001;  
 ​  
 GRANT wmj1 TO wmj\_1;



**创建第二个角色和用户**

* 创建角色wmj2，将connect,resource权限给wmj2
* 创建用户wmj\_2
* 分配60M空间给wmj\_2并将角色wmj2授权给用户wmj\_2

 CREATE ROLE wmj2;  
 ​  
 GRANT connect,resource TO wmj2;  
 ​  
 CREATE USER wmj\_2 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_wmj001 TEMPORARY TABLESPACE temp;  
 ​  
 ALTER USER wmj\_2 QUOTA 60M ON space\_wmj001;  
 ​  
 GRANT wmj2 TO wmj\_2;

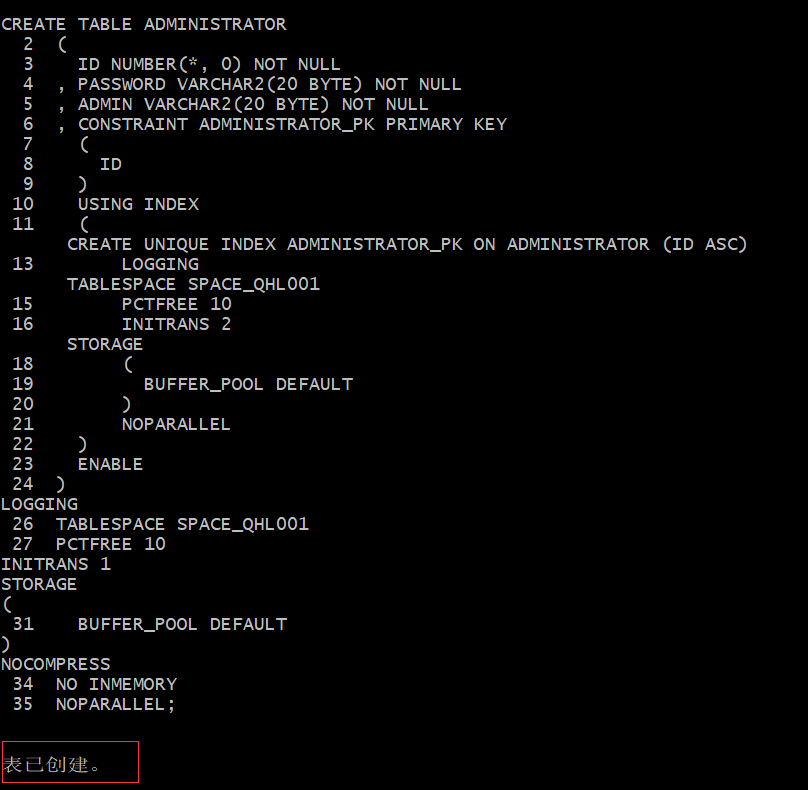


**3. 在用户wmj\_1下创建表**

**创建管理员表**

* id为主键

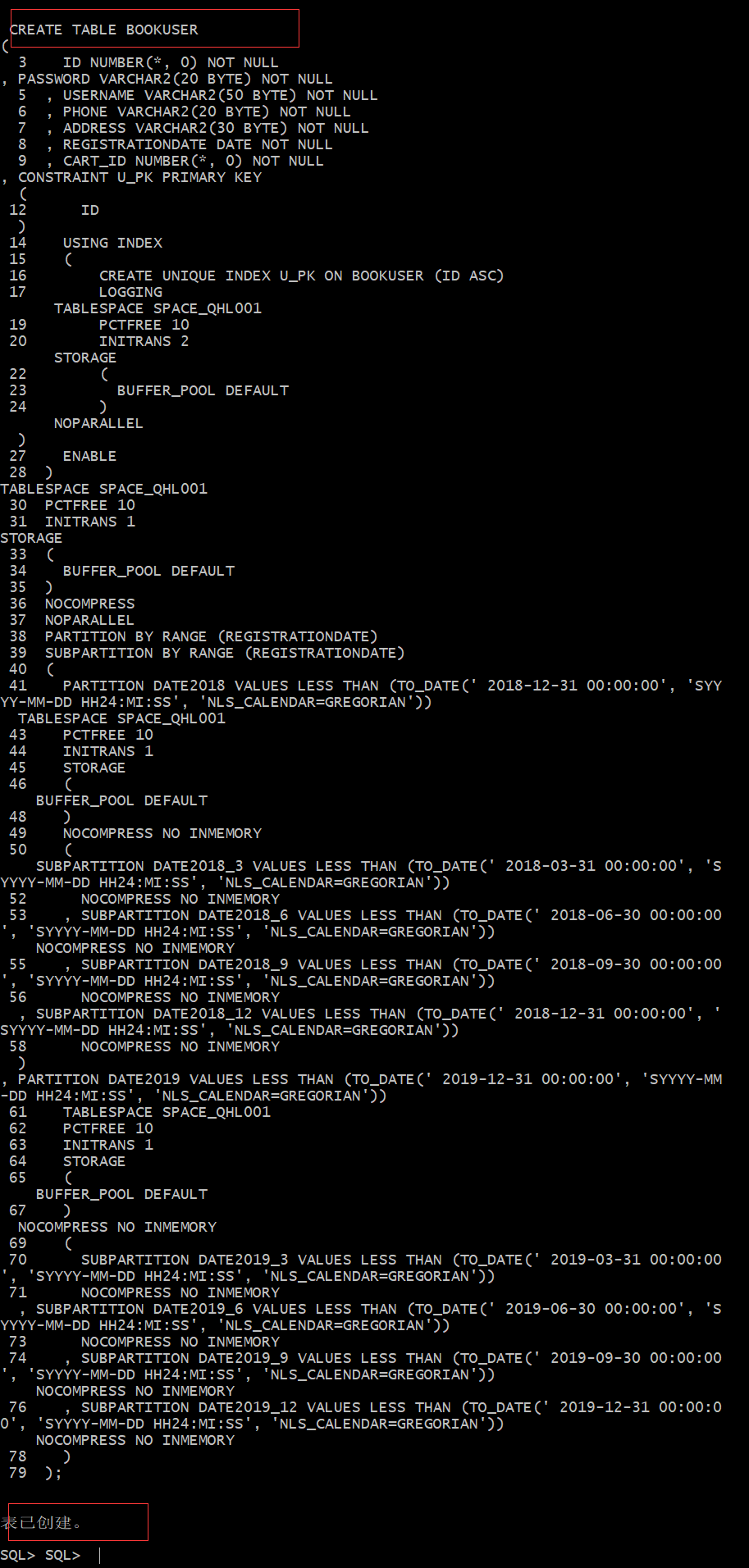
 CREATE TABLE ADMINISTRATOR   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PASSWORD VARCHAR2(20 BYTE) NOT NULL   
 , ADMIN VARCHAR2(20 BYTE) NOT NULL   
 , CONSTRAINT ADMINISTRATOR\_PK PRIMARY KEY   
  (  
    ID   
  )  
  USING INDEX   
  (  
       CREATE UNIQUE INDEX ADMINISTRATOR\_PK ON ADMINISTRATOR (ID ASC)   
      LOGGING   
      TABLESPACE SPACE\_wmj001   
      PCTFREE 10   
      INITRANS 2   
      STORAGE   
      (   
        BUFFER\_POOL DEFAULT   
      )   
      NOPARALLEL   
  )  
  ENABLE   
 )   
 LOGGING   
 TABLESPACE SPACE\_wmj001   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NO INMEMORY   
 NOPARALLEL;

****

**创建用户表**

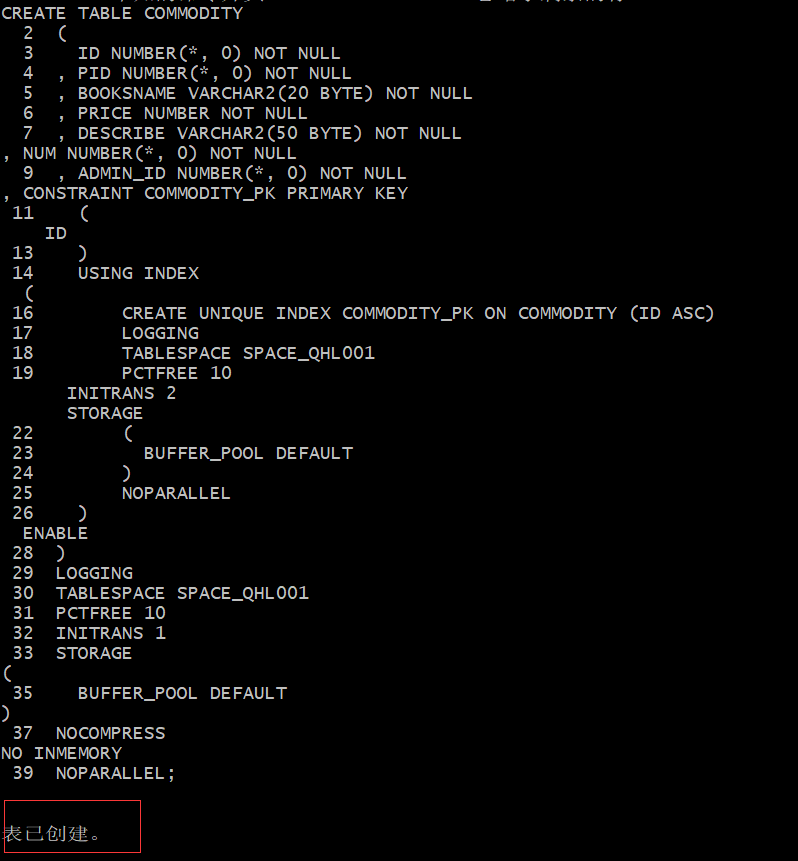
* id为主键
* 根据注册日期按范围分区
* 分为2018和2019年两个分区，每年按季度划4个子分区

 CREATE TABLE BOOKUSER   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PASSWORD VARCHAR2(20 BYTE) NOT NULL   
 , USERNAME VARCHAR2(50 BYTE) NOT NULL   
 , PHONE VARCHAR2(20 BYTE) NOT NULL   
 , ADDRESS VARCHAR2(30 BYTE) NOT NULL   
 , REGISTRATIONDATE DATE NOT NULL   
 , CART\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT U\_PK PRIMARY KEY   
  (  
    ID   
  )  
  USING INDEX   
  (  
       CREATE UNIQUE INDEX U\_PK ON BOOKUSER (ID ASC)   
      LOGGING   
      TABLESPACE SPACE\_wmj001   
      PCTFREE 10   
      INITRANS 2   
      STORAGE   
      (   
        BUFFER\_POOL DEFAULT   
      )   
      NOPARALLEL   
  )  
  ENABLE   
 )   
 TABLESPACE SPACE\_wmj001   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NOPARALLEL   
 PARTITION BY RANGE (REGISTRATIONDATE)   
 SUBPARTITION BY RANGE (REGISTRATIONDATE)   
 (  
  PARTITION DATE2018 VALUES LESS THAN (TO\_DATE(' 2018-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY   
  (  
    SUBPARTITION DATE2018\_3 VALUES LESS THAN (TO\_DATE(' 2018-03-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2018\_6 VALUES LESS THAN (TO\_DATE(' 2018-06-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2018\_9 VALUES LESS THAN (TO\_DATE(' 2018-09-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2018\_12 VALUES LESS THAN (TO\_DATE(' 2018-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  )    
 , PARTITION DATE2019 VALUES LESS THAN (TO\_DATE(' 2019-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY   
  (  
    SUBPARTITION DATE2019\_3 VALUES LESS THAN (TO\_DATE(' 2019-03-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2019\_6 VALUES LESS THAN (TO\_DATE(' 2019-06-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2019\_9 VALUES LESS THAN (TO\_DATE(' 2019-09-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  , SUBPARTITION DATE2019\_12 VALUES LESS THAN (TO\_DATE(' 2019-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
  )    
 );

****

**创建商品表**

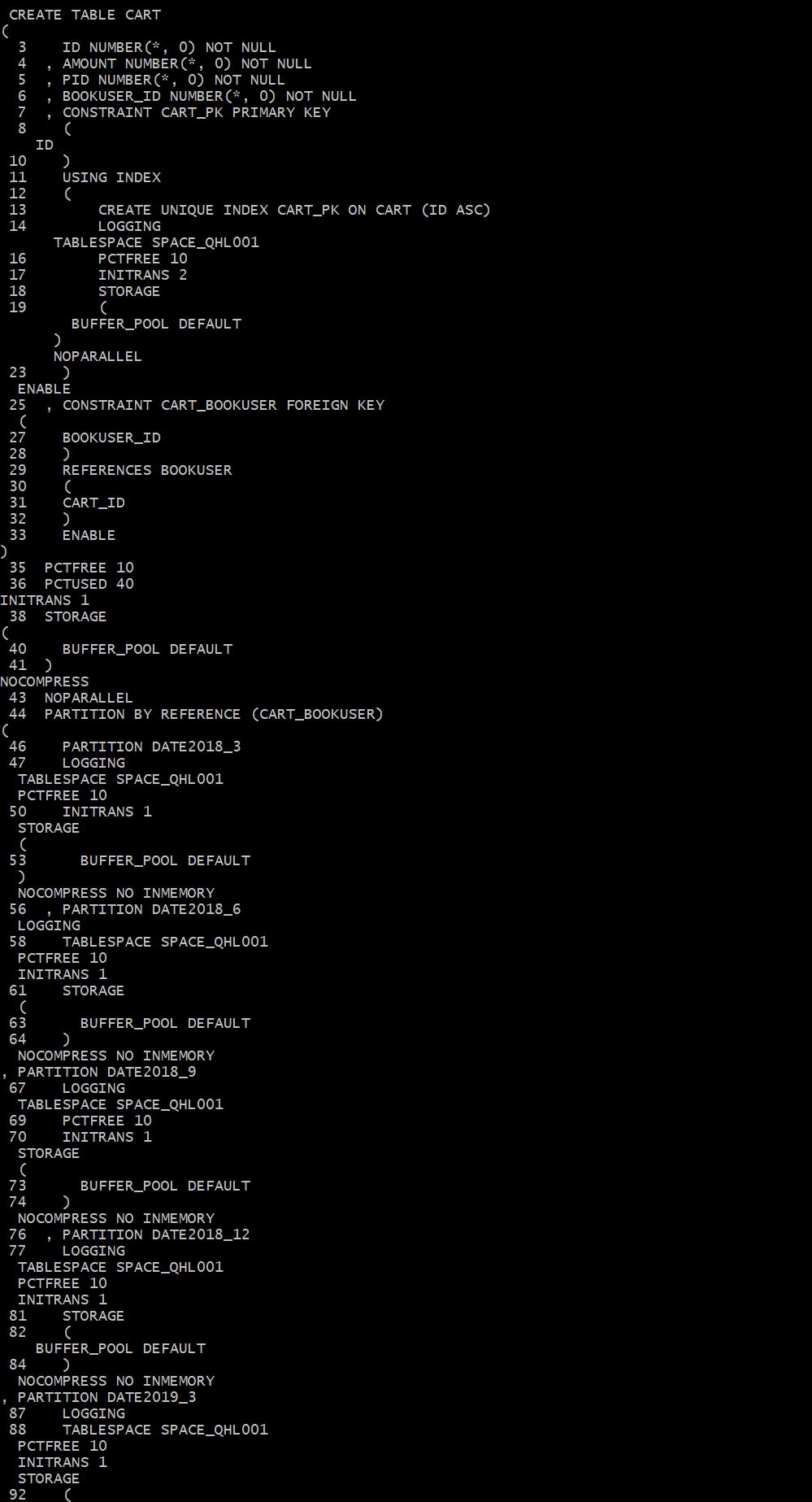
 CREATE TABLE COMMODITY   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PID NUMBER(\*, 0) NOT NULL   
 , BOOKSNAME VARCHAR2(20 BYTE) NOT NULL   
 , PRICE NUMBER NOT NULL   
 , DESCRIBE VARCHAR2(50 BYTE) NOT NULL   
 , NUM NUMBER(\*, 0) NOT NULL   
 , ADMIN\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT COMMODITY\_PK PRIMARY KEY   
  (  
    ID   
  )  
  USING INDEX   
  (  
       CREATE UNIQUE INDEX COMMODITY\_PK ON COMMODITY (ID ASC)   
      LOGGING   
      TABLESPACE SPACE\_wmj001   
      PCTFREE 10   
      INITRANS 2   
      STORAGE   
      (   
        BUFFER\_POOL DEFAULT   
      )   
      NOPARALLEL   
  )  
  ENABLE   
 )   
 LOGGING   
 TABLESPACE SPACE\_wmj001   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NO INMEMORY   
 NOPARALLEL;

****

**创建购物车表**

* 用户表字段BOOKUSER\_ID为购物车表的外键
* 购物车采用引用分区

 CREATE TABLE CART   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , AMOUNT NUMBER(\*, 0) NOT NULL   
 , PID NUMBER(\*, 0) NOT NULL   
 , BOOKUSER\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT CART\_PK PRIMARY KEY   
  (  
    ID   
  )  
  USING INDEX   
  (  
       CREATE UNIQUE INDEX CART\_PK ON CART (ID ASC)   
      LOGGING   
      TABLESPACE SPACE\_wmj001   
      PCTFREE 10   
      INITRANS 2   
      STORAGE   
      (   
        BUFFER\_POOL DEFAULT   
      )   
      NOPARALLEL   
  )  
  ENABLE   
 , CONSTRAINT CART\_BOOKUSER FOREIGN KEY  
  (  
  BOOKUSER\_ID   
  )  
  REFERENCES BOOKUSER  
  (  
  CART\_ID   
  )  
  ENABLE   
 )   
 PCTFREE 10   
 PCTUSED 40   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NOPARALLEL   
 PARTITION BY REFERENCE (CART\_BOOKUSER)   
 (  
  PARTITION DATE2018\_3   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_6   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_9   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_12   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_3   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_6   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_9   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_12   
  LOGGING   
  TABLESPACE SPACE\_wmj001   
  PCTFREE 10   
  INITRANS 1   
  STORAGE   
  (   
    BUFFER\_POOL DEFAULT   
  )   
  NOCOMPRESS NO INMEMORY    
 );

****

**论坛表**

CREATE TABLE TABLE1

(

ID INT NOT NULL

, CONTENT NVARCHAR2(50) NOT NULL

, CONSTRAINT TABLE1\_PK PRIMARY KEY

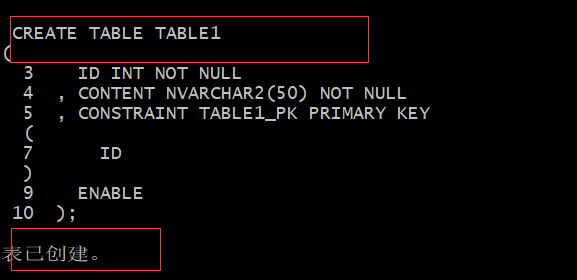
(

ID

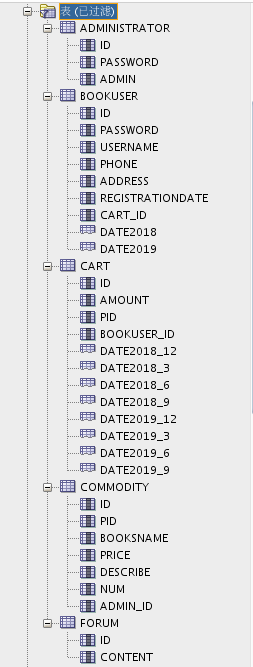
)

ENABLE

);

****

**创建完成后表结构如下**



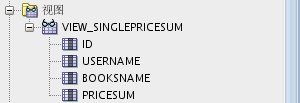
**创建视图计算每个用户购车中单个商品的总价**

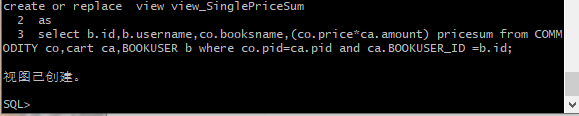
create or replace view view\_SinglePriceSum

as

select b.id,b.username,co.booksname,(co.price\*ca.amount) pricesum from COMMODITY co,cart ca,BOOKUSER b where co.pid=ca.pid and ca.BOOKUSER\_ID =b.id;

select \* from view\_SinglePriceSum;





11

**用户wmj\_1空间不足，修改wmj\_1空间大小**

ALTER USER wmj\_1 QUOTA 90M ON space\_wmj001;

**4. 插入用户、商品、购物车数据**

declare

id number(38,0);

username varchar2(50);

phone varchar2(20);

address varchar2(30);

REGISTRATIONDATE date;

booksname varchar2(50);

price number(5,2);

num number(38,0);

amount number(38,0);

begin

for i in 1..20000

loop

if i mod 2 =0 then

REGISTRATIONDATE:=to\_date('2018-5-6','yyyy-mm-dd')+(i mod 60);

else

REGISTRATIONDATE:=to\_date('2019-5-6','yyyy-mm-dd')+(i mod 60);

end if;

--插入用户

id:=SEQ\_ORDER\_ID.nextval; --应该将SEQ\_ORDER\_ID.nextval保存到变量中。

username := 'aa'|| 'aa';

username := 'wang' || i;

phone := '131785693' || i;

booksname := '唐诗三百首版本号' || i;

address :='成都'|| '四川';

price :=(dbms\_random.value() \* 100);

num :=(i mod 5);

insert /\*+append\*/ into bookuser (id,password,username,phone,address,REGISTRATIONDATE,cart\_id)

values (id,username,username,phone,address,REGISTRATIONDATE,id);

--插入货品

insert into commodity(id,pid,booksname,price,describe,num,admin\_id)

values (id,id,booksname,price,'good',num,1);

--插入购物车

amount :=(id mod 3 ) + 1;

insert into cart(id,amount,pid,bookuser\_id)

values (id,amount,id,id);

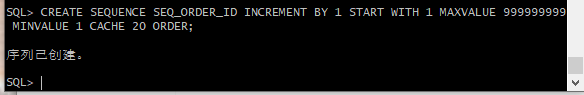
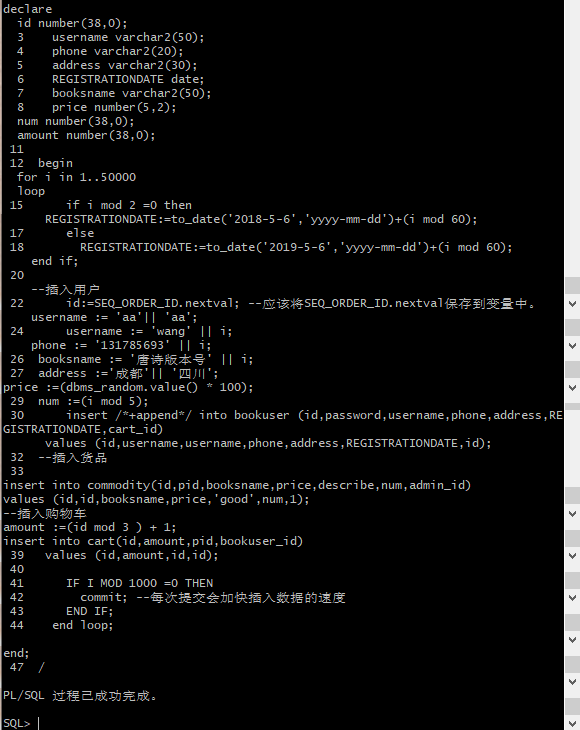
IF I MOD 1000 =0 THEN

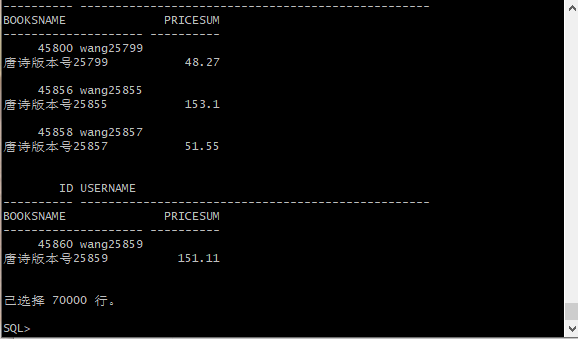
commit; --每次提交会加快插入数据的速度

END IF;

end loop;

end;





**6.创建程序包、存储过程、函数执行分析计划**

**创建程序包**

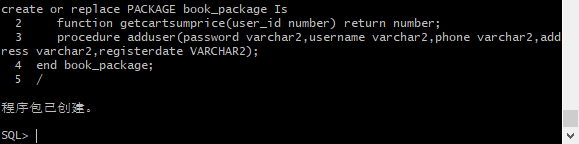
* 函数getcartsumprice计算每个用户的购物车商品总金额
* 存储过程adduser插入用户信息

create or replace PACKAGE book\_package Is

function getcartsumprice(user\_id number) return number;

procedure adduser(password varchar2,username varchar2,phone varchar2,address varchar2,registerdate VARCHAR2);

end book\_package;



**创建函数、存储过程**

create or replace PACKAGE body book\_package Is

function getcartsumprice(user\_id number) return number as

begin

declare cart\_sum number;

query\_sql varchar2(200);

begin

query\_sql:='select sum(pricesum) from view\_SinglePriceSum where ID=' || user\_id;

execute immediate query\_sql into cart\_sum;

return cart\_sum;

end;

end getcartsumprice;

procedure addUser(password varchar2,username varchar2,phone varchar2,address varchar2,registerdate varchar2) as

begin

declare maxId number;

begin

select max(id) into maxId from bookuser;

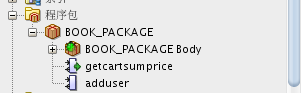
insert into bookuser values(maxId+1,password,username,phone,address,to\_date(registerdate,'yyyy-mm-dd'),maxId+1);

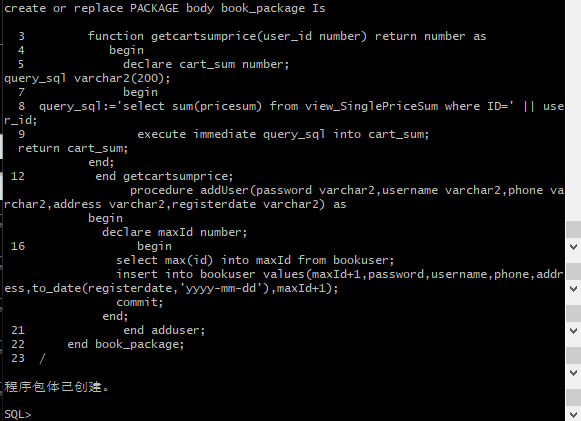
commit;

end;

end adduser;

end book\_package;

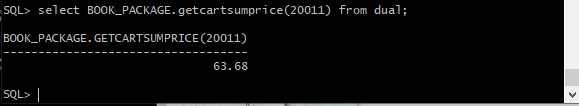




**存储过程、函数执行分析**

使用自定义函数getcartsumprice（）查询id号为20011的用户购物车商品总价

select BOOK\_PACKAGE.getcartsumprice(20011) from dual;



使用存储过程adduser插入用户数据

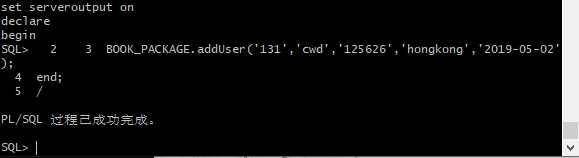
set serveroutput on

declare

begin

BOOK\_PACKAGE.addUser('131','cwd','125626','hongkong','2019-05-02');

end;

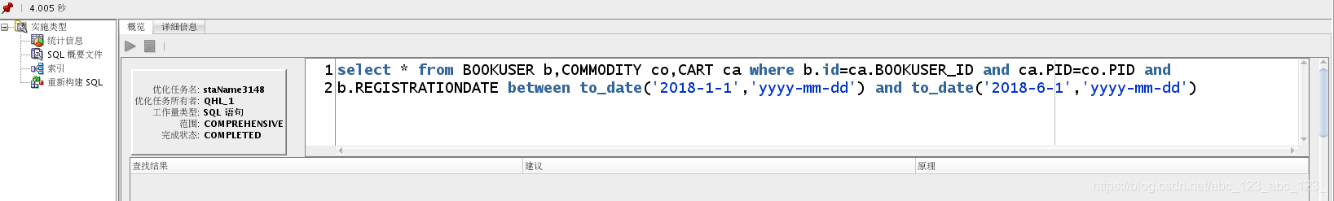


E:\GitWorkSpace\Oracle\test06\pic\9.png

**执行计划分析**

select \* from BOOKUSER b,COMMODITY co,CART ca where b.id=ca.BOOKUSER\_ID and ca.PID=co.PID and

b.REGISTRATIONDATE between to\_date('2018-1-1','yyyy-mm-dd') and to\_date('2018-6-1','yyyy-mm-dd');





**表空间使用状况**

SELECT a.tablespace\_name "表空间名",

total "表空间大小",

free "表空间剩余大小",

(total - free) "表空间使用大小",

total / (1024 \* 1024 \* 1024) "表空间大小(G)",

free / (1024 \* 1024 \* 1024) "表空间剩余大小(G)",

(total - free) / (1024 \* 1024 \* 1024) "表空间使用大小(G)",

round((total - free) / total, 4) \* 100 "使用率 %"

FROM (SELECT tablespace\_name, SUM(bytes) free

FROM dba\_free\_space

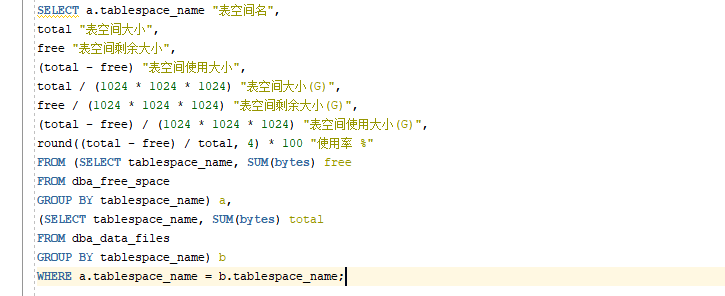
GROUP BY tablespace\_name) a,

(SELECT tablespace\_name, SUM(bytes) total

FROM dba\_data\_files

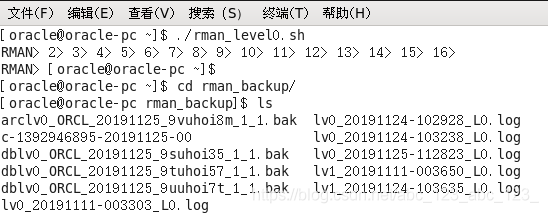
GROUP BY tablespace\_name) b

WHERE a.tablespace\_name = b.tablespace\_name

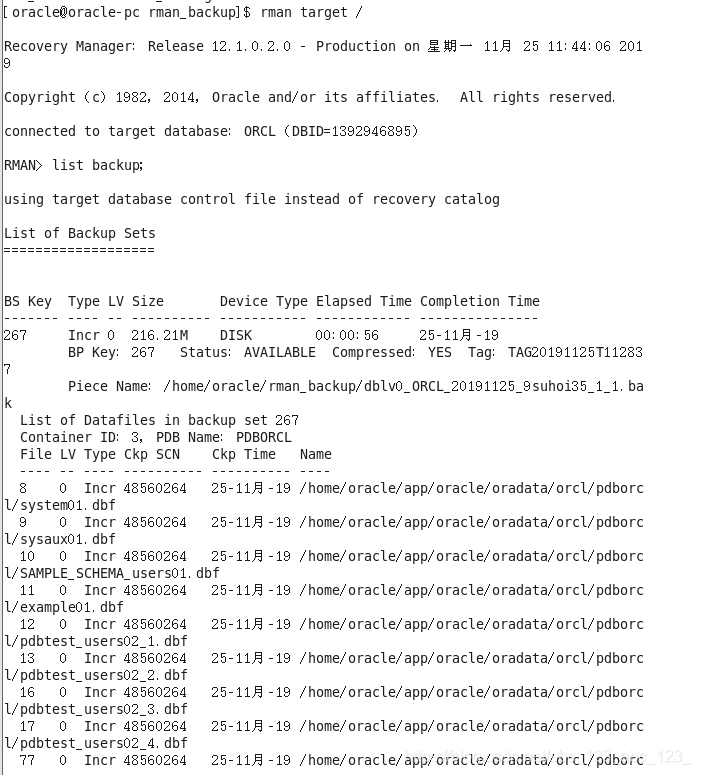


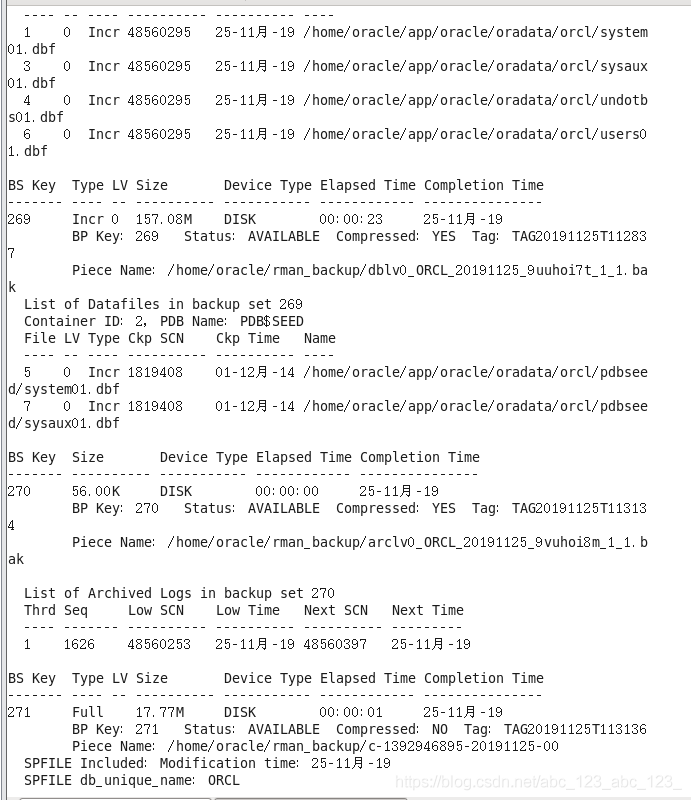
**7.备份恢复**

* 备份./rman\_level0.sh



* 查看备份内容

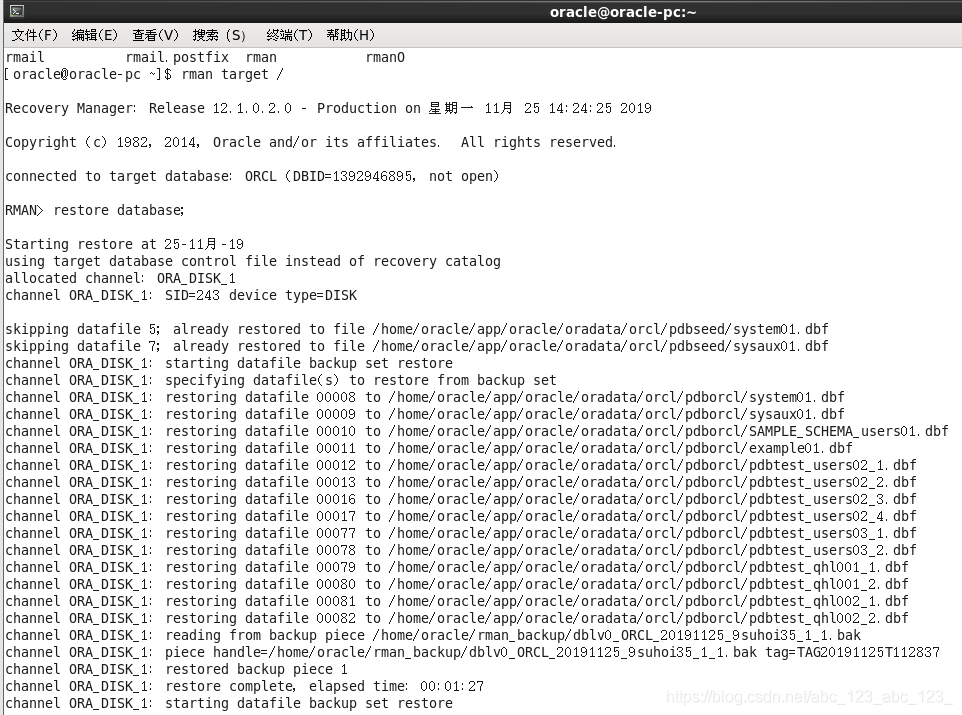


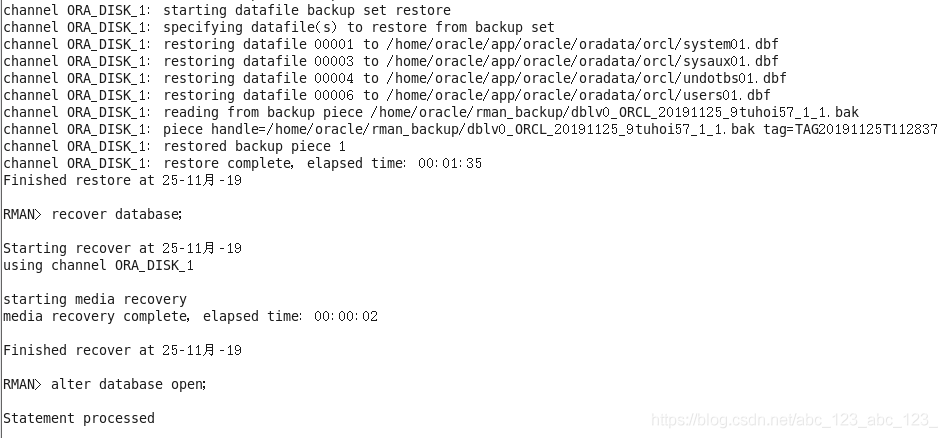


* 删除数据

E:\GitWorkSpace\Oracle\test06\pic\15.png

* 恢复备份





* 数据已恢复

