

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的图书售卖系统数据库设计 | | |
| 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 | 刘勇 | 学号 | 201810414316 |
| 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

目 录

[系统场景 3](#_Toc74490246)

[数据库设计 3](#_Toc74490247)

[管理员表(adminstrator) 3](#_Toc74490248)

[用户表(bookuser) 3](#_Toc74490249)

[商品表(commodity) 4](#_Toc74490250)

[购物车表(cart) 4](#_Toc74490251)

[评论表(forum) 4](#_Toc74490252)

[实验步骤 5](#_Toc74490253)

[1.创建表空间 5](#_Toc74490254)

[2.创建角色及用户 6](#_Toc74490255)

[3.创建表 7](#_Toc74490256)

[4.创建视图计算每个用户购车中单个商品的总价 15](#_Toc74490257)

[5. 用户ly\_1空间不足，修改ly\_1空间大小 15](#_Toc74490258)

[6. 插入用户、商品、购物车数据 16](#_Toc74490259)

[7. 创建程序包、存储过程、函数执行分析计划 17](#_Toc74490260)

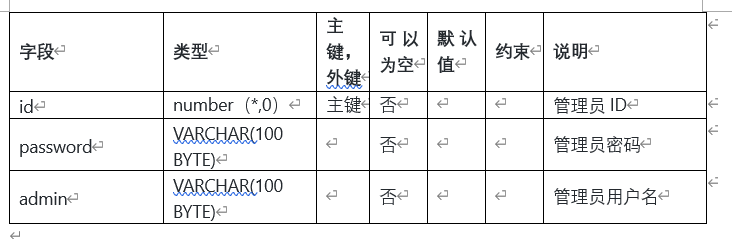
[8. 采用Rman备份 20](#_Toc74490261)

# 系统场景

有一个售卖图书的交易系统，需要对图书、买卖双方进行管理。涉及到的表有：管理员表、用户表、商品表、购物车表、评论表。

# ****数据库设计****

## 管理员表(adminstrator)



## 用户表(bookuser)



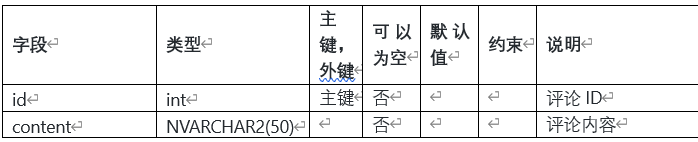
## 商品表(commodity)



## 购物车表(cart)



## 评论表(forum)



# ****实验步骤****

## 1.创建表空间

space\_ly001

Create Tablespace space\_ly001

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_ly001\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_ly001\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

space\_ly002

Create Tablespace space\_ly002

datafile

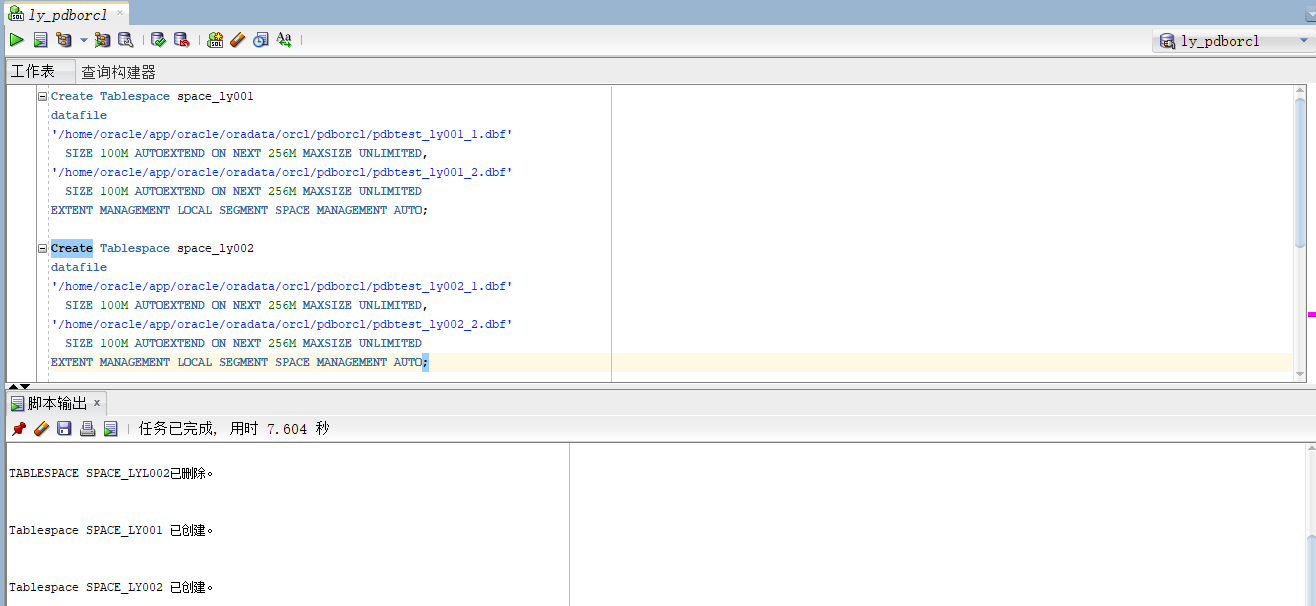
'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_ly002\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_ly002\_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\_ly001

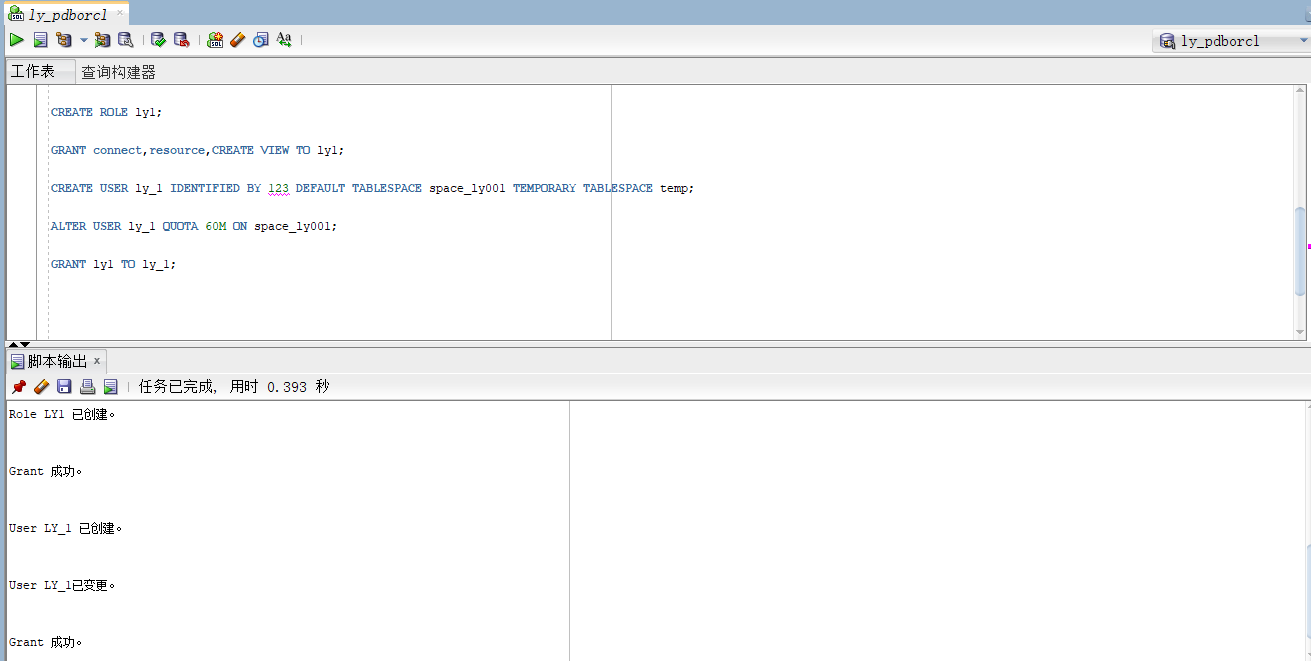
--创建角色ly1及用户ly\_1:

CREATE ROLE ly1;

GRANT connect,resource,CREATE VIEW TO ly1;

CREATE USER ly\_1 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_ly001 TEMPORARY TABLESPACE temp;

ALTER USER ly\_1 QUOTA 60M ON space\_ly001;

 GRANT ly1 TO ly\_1;

--创建角色ly2及用户ly\_2

``` oracle

CREATE ROLE ly2;

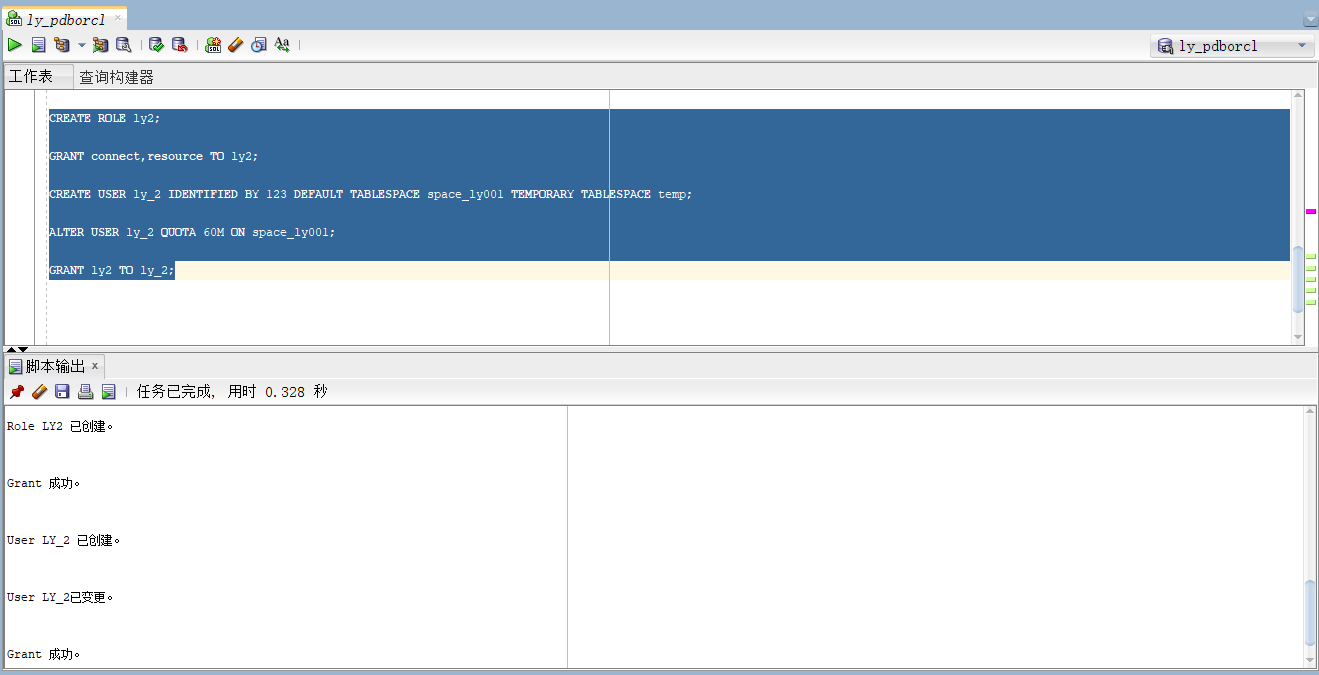
GRANT connect,resource TO ly2;

CREATE USER ly\_2 IDENTIFIED BY 123 DEFAULT TABLESPACE space\_ly001 TEMPORARY TABLESPACE temp;

ALTER USER ly\_2 QUOTA 60M ON space\_ly001;

GRANT ly2 TO ly\_2;

```



## 3.创建表

**管理员表**

``` oracle

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\_LY001

PCTFREE 10

INITRANS 2

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NO INMEMORY

NOPARALLEL;

```

**用户表**

``` oracle

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\_LY001

PCTFREE 10

INITRANS 2

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

TABLESPACE SPACE\_LY001

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\_LY001

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\_LY001

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

)

);

```

**商品表**

``` oracle

CREATE TABLE COMMODITY

(

ID NUMBER(\*, 0) NOT NULL

, PID NUMBER(\*, 0) NOT NULL

, BOOKSNAME VARCHAR2(50 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\_LY001

PCTFREE 10

INITRANS 2

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NO INMEMORY

NOPARALLEL;

```

**购物车表**

``` oracle

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\_LY001

PCTFREE 10

INITRANS 2

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

, CONSTRAINT CART\_BOOKUSER FOREIGN KEY

(

BOOKUSER\_ID

)

REFERENCES BOOKUSER

(

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\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY,

PARTITION DATE2018\_6

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2018\_9

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2018\_12

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2019\_3

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2019\_6

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2019\_9

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION DATE2019\_12

LOGGING

TABLESPACE SPACE\_LY001

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

);

```

**评论表**

``` oracle

CREATE TABLE FORUM

(

ID INT NOT NULL

, CONTENT NVARCHAR2(50) NOT NULL

, CONSTRAINT TABLE1\_PK PRIMARY KEY

(

ID

)

ENABLE

);

```

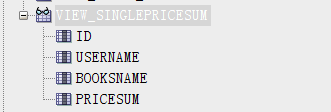
## 4.创建视图计算每个用户购车中单个商品的总价

``` oracle

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;

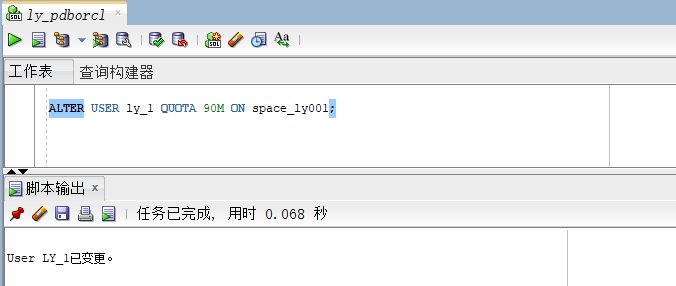
```

## 5. 用户ly\_1空间不足，修改ly\_1空间大小

``` oracle

ALTER USER ly\_1 QUOTA 90M ON space\_ly001;

```



## 6. 插入用户、商品、购物车数据

``` oracle

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..50000

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 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

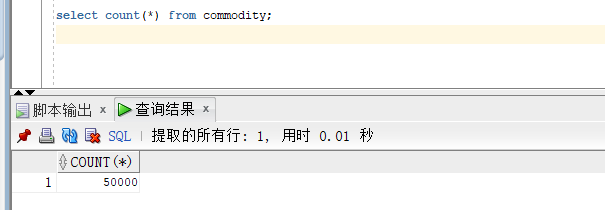
commit; --每次提交会加快插入数据的速度

END IF;

end loop;

end;

```



## 7. 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

``` oracle

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

``` oracle

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;

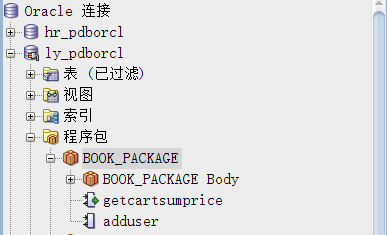
insert into bookuser values(maxId+1,password,username,phone,address,to\_date(registerdate,'yyyy-mm-dd'),maxId+1);

commit;

end;

end adduser;

end book\_package;

```

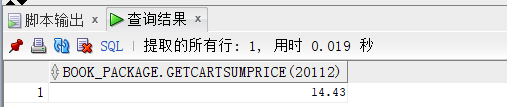
存储过程、函数执行分析

使用自定义函数getcartsumprice（）查询id号为20112的用户购物车商品总价

``` oracle

select BOOK\_PACKAGE.getcartsumprice(20112) from dual;

```



使用存储过程adduser插入用户数据

``` oracle

set serveroutput on

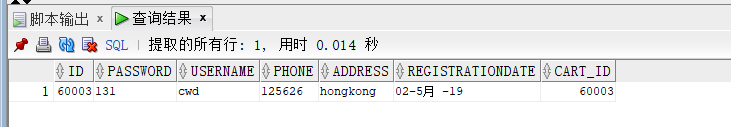
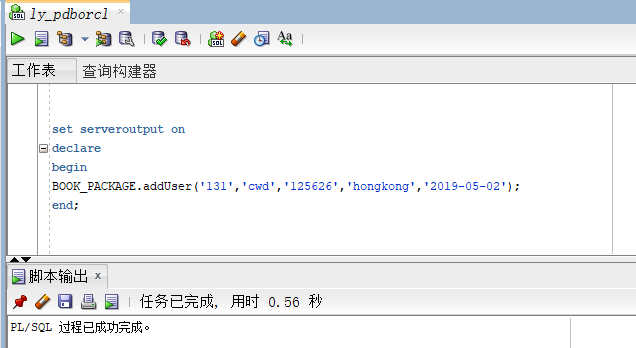
declare

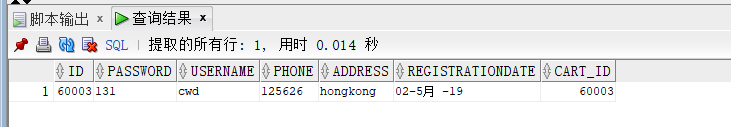
begin

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

end;

```





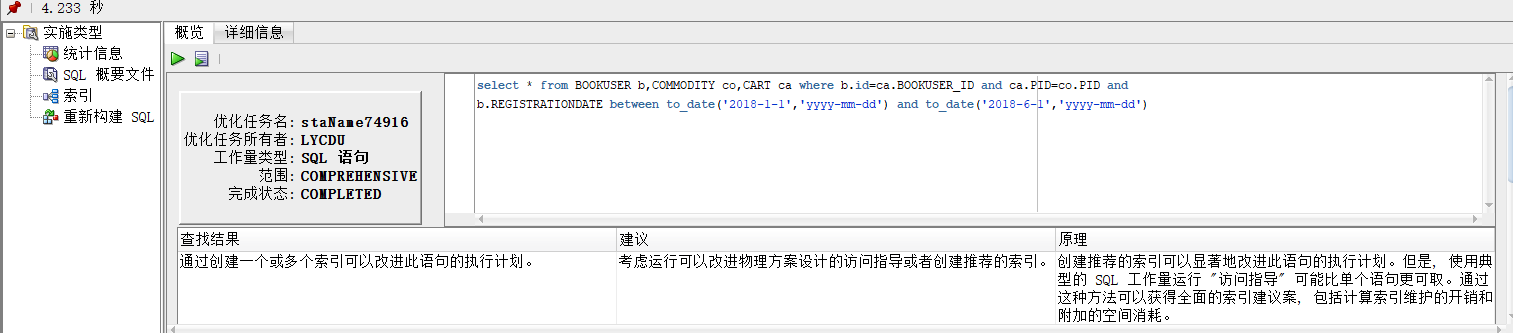
执行计划分析

``` oracle

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');

```



## 8. 采用Rman备份

**8.1创建恢复目录**

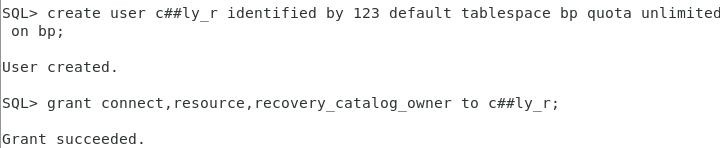
-- 创建恢复目录：用来存储RMAN资料库的

create tablespace bp datafile '/opt/oracle/oradata/bp.dbf' size 20m autoextend on next 5m maxsize unlimited;

-- 在恢复目录数据库中创建RMAN用户并授权

create user c##ly\_r identified by 123 default tablespace bp quota unlimited on bp;

grant connect,resource,recovery\_catalog\_owner to c##ly\_r;



**8.2连接RMAN恢复目录数据库**

- 连接RMAN恢复目录数据库

rman catalog c##ly\_r/123

-- 创建恢复目录

create catalog tablespace bp;

-- 退出

exit

-- 确认环境信息

echo $ORACLE\_SID

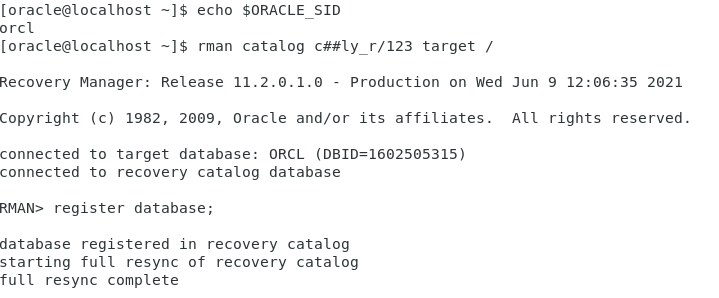
-- 连接到目标数据库、连接到恢复目录数据库

rman catalog c##ly\_r/123 target /

-- 向恢复目录注册数据库ORCL——此时就可以使用RMAN的恢复目录对目标数据库进行备份和恢复操作

register database;





**8.3通道分配**

-- 手动通道配置

run

{

allocate channel ch1 device type disk;

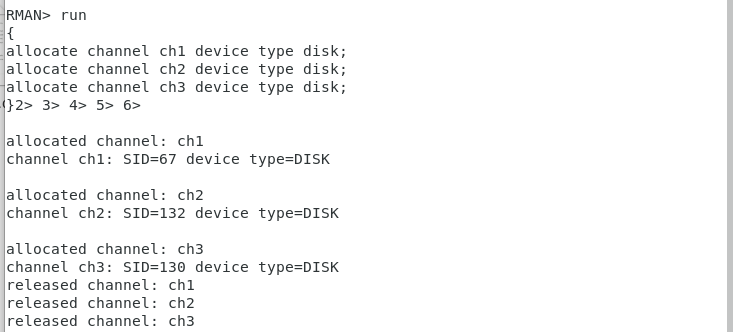
allocate channel ch2 device type disk;

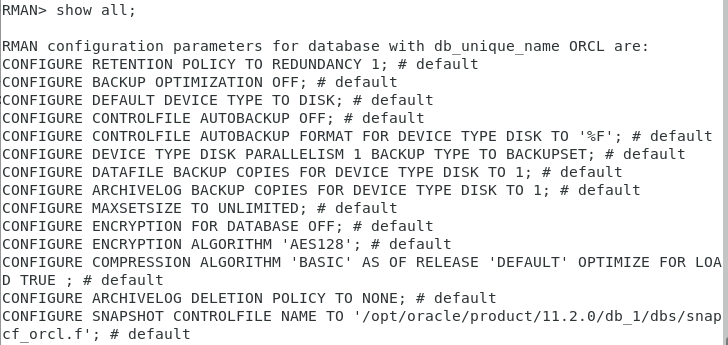
allocate channel ch3 device type disk;

}

-- 显示已经配置过的有默认值的参数，其中包括通道参数

show all;





**8.4归档模式下备份与恢复**

-- 查看数据库是否处于归档模式下

archive log list;

-- 关闭数据库

shutdown immediate

-- 重启并设置成归档模式

startup mount;

alter database archivelog;

archive log list;

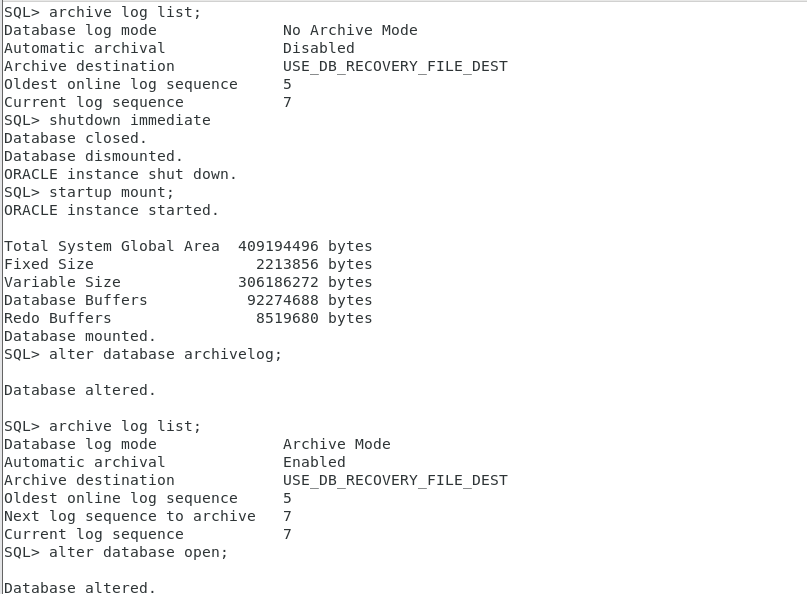
alter database open;

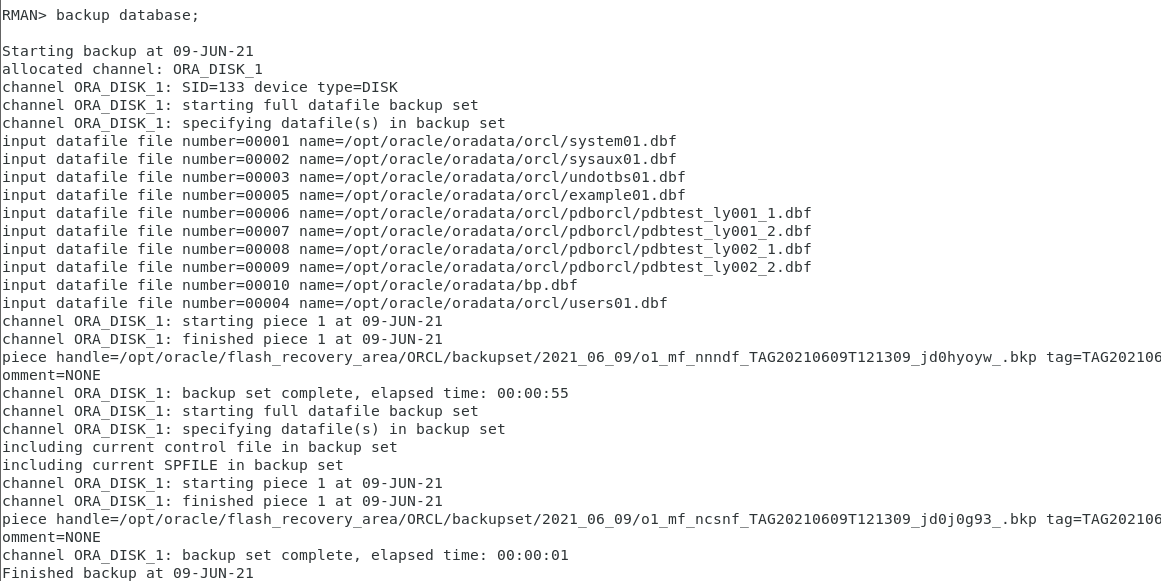
-- 连接到目标数据库、连接到恢复目录数据库

rman catalog c##ly\_r/123 target /

-- 备份和恢复整个数据库

backup database;





**8.5测试备份情况**

-- 切换到保存路径

cd /opt/oracle/flash\_recovery\_area/ORCL/backupset/

-- 查看文件

ls



**8.6测试恢复功能**

-- 关闭数据库

shutdown immediate;

-- 退出数据库

exit

-- 切换到数据文件存储路径

cd /home/oracle/app/oracle/oradata/orcl/pdborcl

-- 查看数据文件

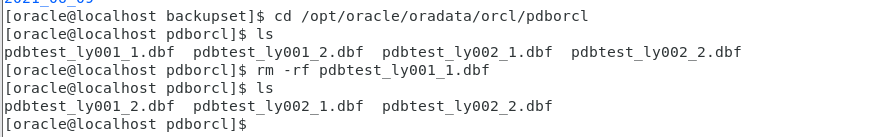
ls

-- 删除train\_data01.dbf

rm -rf train\_data01.dbf

-- 再次确认

ls



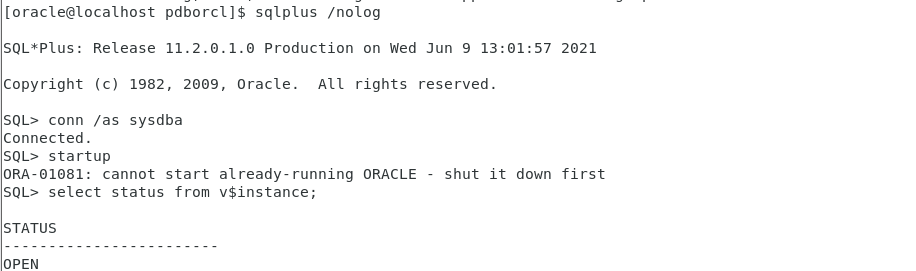
-- 启动数据库

sqlplus /nolog

conn /as sysdba

startup

select status from v$instance;



- 连接RMAN

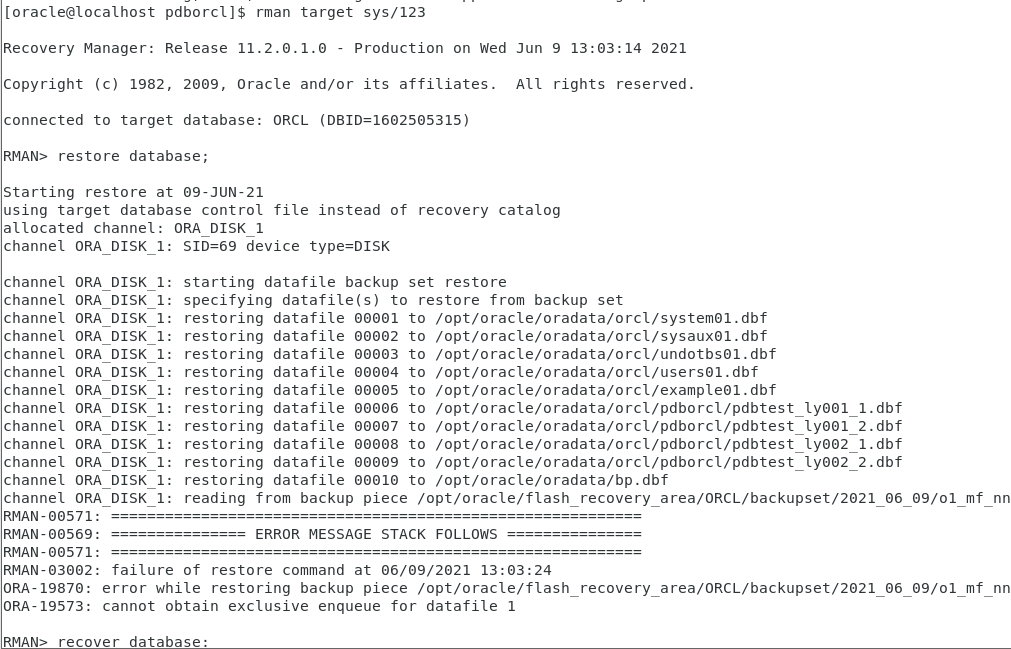
rman target sys/123

-- 恢复数据库

restore database;

-- 同步恢复

recover database;



-- 打开数据库

alter database open resetlogs;

-- 再次启动数据库，启动成功，检查此时数据库状态，此时状态已经打开

startup

select status from v$instance;

-- 再次切换到数据文件存储路径

cd /opt/oracle/oradata/orcl/pdborcl

-- 查看数据文件，删除的已经还原

Ls

