

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的图书管理系统的数据库设ji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 | 夏德成 | 学号 | 201810414323 |
| 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

[一：搭建本地Oracle环境后，登录Oracle并设置密码 3](#_Toc21802)

[二：创建两类角色和两个用户 3](#_Toc12068)

[三：创建表空间并授权给用户 5](#_Toc30151)

[四：创建数据表 7](#_Toc3608)

[五：向表中插入数据 15](#_Toc677)

[六：创建程序包 22](#_Toc9711)

[七：设计手动备份方案 23](#_Toc19695)

[八.项目总结 26](#_Toc10823)

一：搭建本地Oracle环境后，登录Oracle并设置密码

1.进入cmd命令窗口，不使用用户名和密码登录

sqlplus/nolog 或者 sqlplus/ as sysdba

2.更改system用户的密码

alter user system identified by system;

3.测试登录

sqlplus

输入用户system和密码system->登录成功

二：创建两类角色和两个用户

1.创建两类角色

--1角色

--创建 C##XDC\_Role1 角色

CREATE ROLE C##XDC\_Role1;

--给C##XDC\_Role1授权

GRANT connect,resource,CREATE VIEW TO C##XDC\_Role1;

--2角色

--创建 C##XDC\_Role2 角色

CREATE ROLE C##XDC\_Role2;

--给C##XDC\_Role2授权

GRANT connect,resource,CREATE VIEW TO C##XDC\_Role2;

2.创建两个用户

--1用户

--创建用户C##XDC\_user1

CREATE USER C##XDC\_user1 IDENTIFIED BY zytzyt1 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;

--指授权C##XDC\_user用户访问users表空间，空间限额是50M。

ALTER USER C##XDC\_user1 QUOTA 50M ON users;

--指定用户的角色

GRANT C##XDC\_Role1 TO C##XDC\_user1;

--2用户

--创建用户C##XDC\_user2

CREATE USER C##XDC\_user2 IDENTIFIED BY zytzyt2 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;

3.给用户赋予角色和指定表空间限额

--指授权C##XDC\_user用户访问users表空间，空间限额是50M。

ALTER USER C##XDC\_user2 QUOTA 50M ON users;

--指定用户的角色

GRANT C##XDC\_Role2 TO C##XDC\_user2;

运行结果：

SQL> CREATE ROLE C##XDC\_Role1;

角色已创建。

SQL> GRANT connect,resource,CREATE VIEW TO C##XDC\_Role1;

授权成功。

SQL> CREATE ROLE C##XDC\_Role2;

角色已创建。

SQL> GRANT connect,resource,CREATE VIEW TO C##XDC\_Role2;

授权成功。

SQL> CREATE USER C##XDC\_user1 IDENTIFIED BY zytzyt1 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;

用户已创建。

SQL> ALTER USER C##XDC\_user1 QUOTA 50M ON users;

用户已更改。

SQL> GRANT C##XDC\_Role1 TO C##XDC\_user1;

授权成功。

SQL> CREATE USER C##XDC\_user2 IDENTIFIED BY zytzyt2 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;

用户已创建。

SQL> ALTER USER C##XDC\_user2 QUOTA 50M ON users;

用户已更改。

SQL> GRANT C##XDC\_Role2 TO C##XDC\_user2;

授权成功。

三：创建表空间并授权给用户

1.创建表空间

--查看表空间

 select \* from v$tablespace;

-- 创建表空间1

CREATE TABLESPACE tablespace1 DATAFILE

'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace1\_1.dbf'

  SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE unlimited,

'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace1\_2.dbf'

  SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

-- 创建表空间2

CREATE TABLESPACE tablespace2 DATAFILE

'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace2\_1.dbf'

  SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE unlimited,

'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace2\_2.dbf'

  SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

运行结果：

SQL> CREATE TABLESPACE tablespace1 DATAFILE

  2  'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace1\_1.dbf'

  3    SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE unlimited,

  4  'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace1\_2.dbf'

  5    SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED

  6  EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

表空间已创建。

SQL> CREATE TABLESPACE tablespace2 DATAFILE

  2  'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace2\_1.dbf'

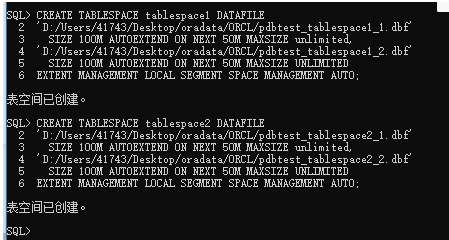
  3    SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE unlimited,

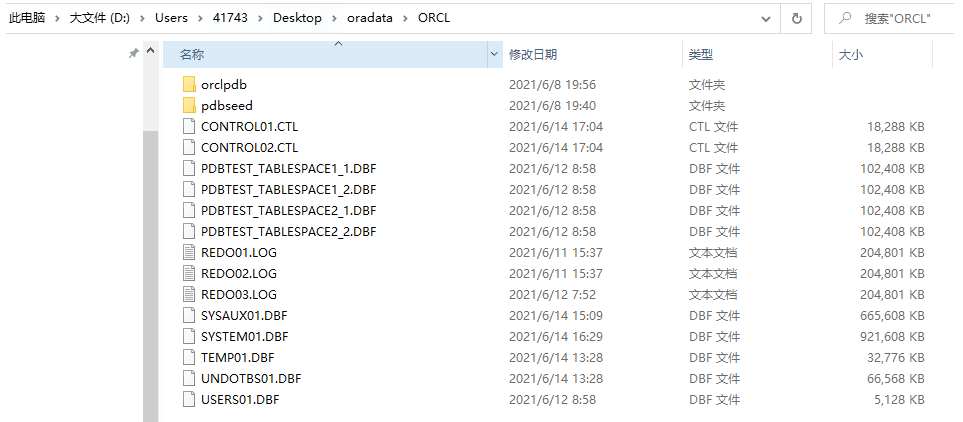
  4  'D:/Users/41743/Desktop/oradata/ORCL/pdbtest\_tablespace2\_2.dbf'

  5    SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED

  6  EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

表空间已创建。





2.登录账号并授予权限

--首先创建自己的账号your\_user，然后以system身份登录:

ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON USERS;

ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON tablespace1;

ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON tablespace2;

运行结果：

SQL> ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON USERS;

用户已更改。

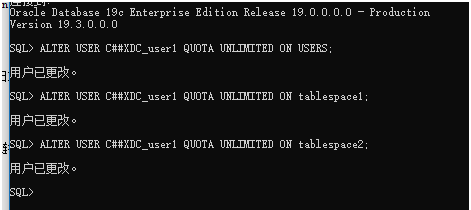
SQL> ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON tablespace1;

用户已更改

SQL> ALTER USER C##XDC\_user1 QUOTA UNLIMITED ON tablespace2;

用户已更改。

截图：



四：创建数据表

数据表分析：

1.用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| user\_id | number(20) | 否 | 主键 |  |  |
| user\_name | VARCHAR2(100 BYTE) | 否 |  |  |  |
| user\_password | VARCHAR2(100 BYTE) | 否 |  |  |  |

2.书种类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| type\_id | number(20) | 否 | 主键 |  |  |
| type\_name | VARCHAR2(100 BYTE) | 否 |  |  |  |
| type\_comment | VARCHAR2(100 BYTE) | 否 |  |  |  |

3.出版社表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| PUB\_id | number(20) | 否 | 主键 |  |  |
| PUB\_name | VARCHAR2(100 BYTE) | 否 |  |  |  |
| PUB\_tel | VARCHAR2(100 BYTE) | 否 |  |  |  |
| PUB\_intro | VARCHAR2(100 BYTE) | 否 |  |  |  |

4.书籍表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| book\_id | number(20) | 否 | 主键 |  |  |
| book\_name | VARCHAR2(100 BYTE) | 否 |  |  |  |
| book\_info | VARCHAR2(100 BYTE) | 否 |  |  |  |
| book\_price | VARCHAR2(100 BYTE) | 否 |  |  |  |
| TYPE\_ID\_FK | number(20,2) | 否 | 外键 |  |  |
| PUB\_ID\_FK | number(20) | 否 | 外键 |  |  |
| book\_IMAGE\_URL | number(20) | 否 |  |  |  |
| book\_AUTHOR | VARCHAR2(100 BYTE) | 否 |  |  |  |
| book\_REPERTORY\_SIZE | int | 否 |  |  |  |

5.交易记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| id | int | 否 | 主键 |  |  |
| RECORD\_DATE | Date | 否 |  |  |  |

6.书的销售记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| id | number(20) | 否 | 主键 |  |  |
| BOOK\_ID\_FK | number(20) | 否 | 外键 |  |  |
| T\_SALE\_RECORD\_ID\_FK | int | 否 | 外键 |  |  |
| TRADE\_SUM | int | 否 |  |  |  |

7.入库记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| id | number(20) | 否 | 主键 |  |  |
| RECORD\_DATE | Date | 否 |  |  |  |

8.书的入库记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **是否为空** | **主/外键** | **默认值** | **约束** |
| id | number(20) | 否 | 主键 |  |  |
| BOOK\_ID\_FK | number(20) | 否 | 外键 |  |  |
| T\_IN\_RECORD\_ID\_FK | number(20) | 否 | 外键 |  |  |
| IN\_SUM | int | 否 |  |  |  |

1. 用户表

create table T\_USER

(

user\_id  number(20) primary key,

user\_name  VARCHAR2(100 BYTE) not null,

user\_password VARCHAR2(100 BYTE) not null

)

tablespace tablespace1

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

2.书种类表：

create table T\_BOOK\_TYPE

(

type\_id  number(20) primary key,

type\_name VARCHAR2(100 BYTE) not null,

type\_comment VARCHAR2(100 BYTE)

)

tablespace tablespace1

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

3.出版社表：

create table T\_PUBLISHER

(

PUB\_id  number(20) primary key,

PUB\_name VARCHAR2(100 BYTE) not null,

PUB\_tel VARCHAR2(100 BYTE) not null,

PUB\_intro VARCHAR2(100 BYTE) not null

)

tablespace tablespace1

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

4.书籍表：

create table T\_BOOK

(

book\_id  number(20) primary key,

book\_name VARCHAR2(100 BYTE) not null,

book\_info VARCHAR2(100 BYTE) not null,

book\_price number(20,2) not null,

TYPE\_ID\_FK number(20) not null,

PUB\_ID\_FK number(20) not null,

book\_IMAGE\_URL VARCHAR2(100 BYTE) not null,

book\_AUTHOR VARCHAR2(100 BYTE) not null,

book\_REPERTORY\_SIZE int not null

)

tablespace tablespace2

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

5.交易记录表

create table T\_SALE\_RECORD

(

id  int primary key,

RECORD\_DATE Date not null

)

tablespace tablespace2

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

6.书的销售记录

create table T\_BOOK\_SALE\_RECORD

(

id  number(20) not null ,

BOOK\_ID\_FK number(20) not null references T\_BOOK(book\_id),

T\_SALE\_RECORD\_ID\_FK int not null references T\_SALE\_RECORD(id),

TRADE\_SUM int not null

)

tablespace tablespace1

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

7.入库记录表

create table T\_IN\_RECORD

(

id  number(20) primary key,

RECORD\_DATE Date not null

)

tablespace tablespace2

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

8.书的入库记录

create table T\_BOOK\_IN\_RECORD

(

id  number(20) primary key,

BOOK\_ID\_FK  number(20) not null references T\_BOOK(book\_id),

T\_IN\_RECORD\_ID\_FK number(20) not null references T\_IN\_RECORD(id),

IN\_SUM int not null

)

tablespace tablespace2

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

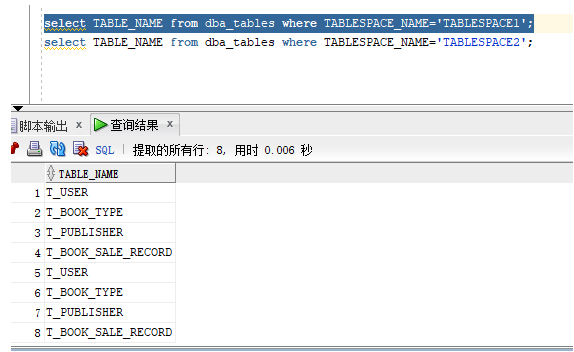
maxextents unlimited

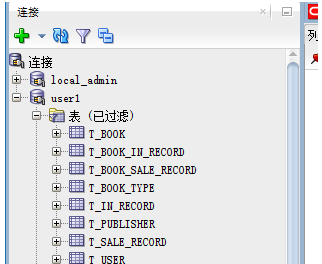
);

查询表空间的数据库

select TABLE\_NAME from dba\_tables where TABLESPACE\_NAME='TABLESPACE1';

select TABLE\_NAME from dba\_tables where TABLESPACE\_NAME='TABLESPACE2';





五：向表中插入数据

1.模拟数据插入：

总共插入数据：52200条

T\_USER：10000条

T\_BOOK\_TYPE：100条

T\_PUBLISHER：100条

T\_BOOK：20000条

T\_SALE\_RECORD：10000条

T\_BOOK\_SALE\_RECORD：10000条

T\_IN\_RECORD：1000条

T\_BOOK\_IN\_RECORD：1000条

2.T\_USER表的数据插入10000条

declare

i int;

user\_id  number(20) ;

user\_name  VARCHAR2(100 BYTE);

user\_password VARCHAR2(100 BYTE);

begin

i:=1;

while i<=10000

loop

user\_id:=i;

user\_name:= 'name'|| i;

user\_password := '123'|| i;

insert into T\_USER(user\_id,user\_name,user\_password) values (user\_id,user\_name,user\_password);

i:=i+1;

end loop;

commit;

end;

/

3.T\_BOOK\_TYPE表数据插入1000

declare

i int;

type\_id  number(20) ;

type\_name VARCHAR2(100 BYTE) ;

type\_comment VARCHAR2(100 BYTE);

begin

i:=1;

while i<=1000

loop

type\_id:=i;

type\_name:= 'type'|| i;

type\_comment := '介绍'||i;

insert into T\_BOOK\_TYPE(type\_id,type\_name,type\_comment) values (type\_id,type\_name,type\_comment);

i:=i+1;

end loop;

commit;

end;

/

4.T\_PUBLISHER表的数据插入100条

declare

i int;

PUB\_id  number(20);

PUB\_name VARCHAR2(100 BYTE) ;

PUB\_tel VARCHAR2(100 BYTE) ;

PUB\_intro VARCHAR2(100 BYTE) ;

begin

i:=1;

while i<=100

loop

PUB\_id  :=i;

PUB\_name := 'name'|| i;

PUB\_tel := '电话'|| i;

PUB\_intro := '介绍'|| i;

insert into T\_PUBLISHER(PUB\_id  ,PUB\_name ,PUB\_tel ,PUB\_intro ) values (PUB\_id  ,PUB\_name ,PUB\_tel ,PUB\_intro);

i:=i+1;

end loop;

commit;

end;

/

5.T\_BOOK表的数据插入20000条

declare

i int;

book\_id  number(20) ;

book\_name VARCHAR2(100 BYTE);

book\_info VARCHAR2(100 BYTE) ;

book\_price number(20,2) ;

TYPE\_ID\_FK number(20);

PUB\_ID\_FK number(20) ;

book\_IMAGE\_URL VARCHAR2(100 BYTE) ;

book\_AUTHOR VARCHAR2(100 BYTE) ;

book\_REPERTORY\_SIZE int ;

begin

i:=1;

while i<=20000

loop

book\_id:=i;

book\_name := 'name'||i;

book\_info := 'info'||i;

book\_price :=dbms\_random.value(100,1);

TYPE\_ID\_FK := i;

PUB\_ID\_FK := i;

book\_IMAGE\_URL:='Bnnnnn'||i;

book\_AUTHOR:='Bnnnnn'||i;

book\_REPERTORY\_SIZE :=i;

insert into T\_BOOK(book\_id,book\_name ,book\_info ,book\_price ,TYPE\_ID\_FK ,PUB\_ID\_FK ,book\_IMAGE\_URL,book\_AUTHOR,book\_REPERTORY\_SIZE ) values (book\_id,book\_name ,book\_info ,book\_price ,TYPE\_ID\_FK ,PUB\_ID\_FK ,book\_IMAGE\_URL,book\_AUTHOR,book\_REPERTORY\_SIZE);

i:=i+1;

end loop;

commit;

end;

/

6.T\_SALE\_RECORD表的数据插入10000条

declare

i int;

id  int;

RECORD\_DATE Date ;

begin

i:=1;

while i<=10000

loop

id  :=i;

if i mod 6 =0 then

  RECORD\_DATE :=to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =1 then

  RECORD\_DATE :=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =2 then

  RECORD\_DATE :=to\_date('2017-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =3 then

  RECORD\_DATE :=to\_date('2018-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =4 then

  RECORD\_DATE :=to\_date('2019-3-2','yyyy-mm-dd')+(i mod 60);

else

  RECORD\_DATE :=to\_date('2020-3-2','yyyy-mm-dd')+(i mod 60);

end if;

insert into T\_SALE\_RECORD(id,RECORD\_DATE) values (id,RECORD\_DATE);

i:=i+1;

end loop;

commit;

end;

/

7.T\_IN\_RECORD表的数据的插入10000条

declare

i int;

id  number(20);

RECORD\_DATE Date;

begin

i:=1;

while i<=10000

loop

id:=i;

if i mod 6 =0 then

RECORD\_DATE :=to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =1 then

RECORD\_DATE :=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =2 then

RECORD\_DATE :=to\_date('2017-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =3 then

RECORD\_DATE :=to\_date('2018-3-2','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =4 then

RECORD\_DATE :=to\_date('2019-3-2','yyyy-mm-dd')+(i mod 60);

else

RECORD\_DATE :=to\_date('2020-3-2','yyyy-mm-dd')+(i mod 60);

end if;

insert into T\_IN\_RECORD(id,RECORD\_DATE) values (id,RECORD\_DATE);

i:=i+1;

end loop;

end;

/

8.T\_BOOK\_IN\_RECORD表的数据插入10000条

declare

i int;

id number(20);

BOOK\_ID\_FK  number(20);

T\_IN\_RECORD\_ID\_FK number(20);

IN\_SUM int;

begin

i:=1;

while i<=10000

loop

id:=i;

BOOK\_ID\_FK:=i;

T\_IN\_RECORD\_ID\_FK:=i;

 IN\_SUM :=i;

insert into T\_BOOK\_IN\_RECORD(id,BOOK\_ID\_FK,T\_IN\_RECORD\_ID\_FK,IN\_SUM ) values (id,BOOK\_ID\_FK,T\_IN\_RECORD\_ID\_FK,IN\_SUM);

i:=i+1;

end loop;

commit;

end;

/

9.查询是否插入成功

select count(\*) from T\_USER;

select count(\*) from T\_BOOK\_TYPE;

select count(\*) from T\_PUBLISHER;

select count(\*) from T\_BOOK;

select count(\*) from T\_SALE\_RECORD;

select count(\*) from T\_BOOK\_SALE\_RECORD;

select count(\*) from T\_IN\_RECORD;

select count(\*) from T\_BOOK\_IN\_RECORD



六：创建程序包

包名：BookPack

函数名：Get\_OutBook(id NUMBER) 功能： 输入销售记录编号，获取该次销售记录所关联的图书

过程名：Get\_InBook(id NUMBER) 功能: 输入入库记录编号，获取该次入库所关联的图书

-- 创建包

create or replace PACKAGE BookPack IS

FUNCTION Get\_OutBook(id NUMBER);

PROCEDURE Get\_InBook(id NUMBER);

END TrainPack;

-- 创建函数和过程

create or replace PACKAGE BODY BookPack IS

FUNCTION Get\_OutBook(id NUMBER)

AS

M VARCHAR2(100);

N VARCHAR2(100);

BEGIN

select \* into N from T\_BOOK\_SALE\_RECORD a,T\_SALE\_RECORD,T\_book c b where a.id=b.id and a.BOOK\_ID\_FK=c.book\_id;

END;

PROCEDURE Get\_orders(train\_id\_t VARCHAR2)

AS

N NUMBER(20);

L date;

S VARCHAR2(20);

R VARCHAR2(100);

begin

select \* into M from T\_IN\_RECORD a,T\_BOOK\_IN\_RECORD b,T\_book c b where a.id=b.id and a.BOOK\_ID\_FK=c.book\_id;

END;

END TrainPack;

/

七：设计手动备份方案

1.创建恢复目录

-- 创建恢复目录

create tablespace bp datafile 'D:/Users/41743/Desktop/oradata/ORCLbp.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##\_XDC\_user1 identified by zytzyt1 default tablespace1 bp quota unlimited on bp;

grant connect,resource,recovery\_catalog\_owner to c##\_XDC\_user1;

2.连接RMAN恢复目录数据库

-- 连接RMAN恢复目录数据库

rman catalog c##\_XDC\_user1/zytzyt1

-- 创建恢复目录

create catalog tablespace1 bp;

-- 退出

quit

-- 确认环境信息

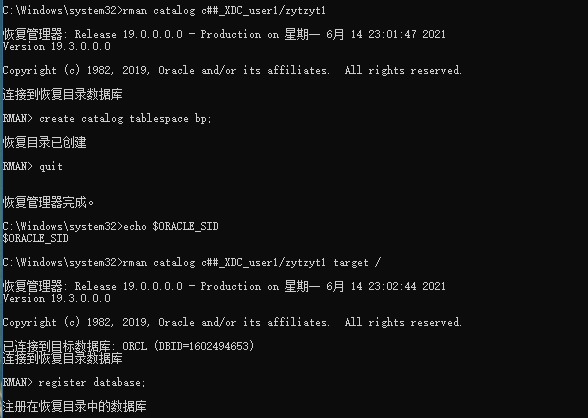
echo $ORACLE\_SID

-- 连接到目标数据库、连接到恢复目录数据库

rman catalog c##\_XDC\_user1/zytzyt1 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;

3.通道分配

-- 手动通道配置

run

{

allocate channel ch1 device type disk;

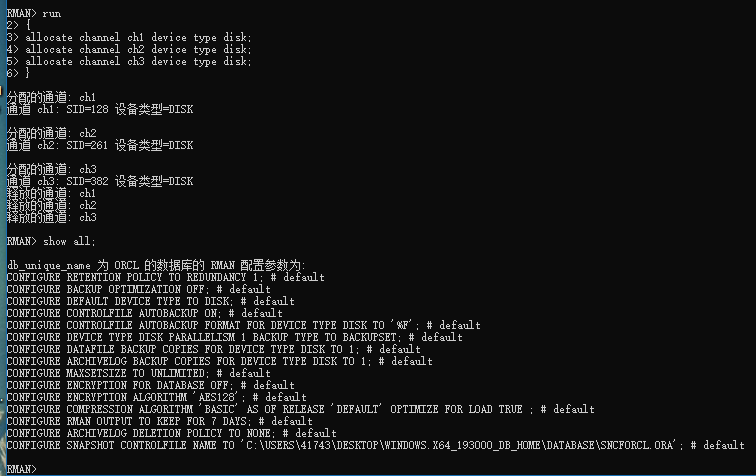
allocate channel ch2 device type disk;

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;

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##\_XDC\_user1/zytzyt1 target /

-- 备份和恢复整个数据库

backup database;

5.测试备份情况

-- 切换到保存路径

cd D:/Users/41743/Desktop/oradata/ORCL/backupset/

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

-- 此时因为缺少数据文件无法启动

-- 检查此时数据库状态

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

八.项目总结

​ 本次项目，是对我们本学期所有Oracle学习的总结，在本次期末项目中，我自行设计了一个信息系统的数据库项目，自拟题目为图书管理系统。并且设计了项目涉及的表及表空间使用方案，总共设计了两个表空间、八张表，共插入了52200条数据。在设计权限及用户分配方案阶段，我设计了两类角色，两个用户。并且设计了手动备份方案，是基于RMAN设计的备份。

​ 本次项目每个阶段，都是和学习课程过程中的实验息息相关，我是根据实验来完成该作业的。在完成作业的过程中，经常会碰见一些奇奇怪怪的问题，每次都要在CSDN论坛社区上去搜索问题的解决办法，并对相关的解决方案进行归类总结和完善，说不定有助于以后的对Oracle的学习。

​ 此次项目，我对Oracle数据库操作有了更进一层的理解。