

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的图书管理系统数据库设计 | | |
| 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 | 赵佳乐 | 学号 | 201810414230 |
| 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** | | |  |

2021 年 6 月 12 日

目录

[图书管理系统数据设计 2](#_Toc28776)

[一、引言 2](#_Toc18160)

[二、 概念模型设计 3](#_Toc27948)

[2.1 E-R图设计 3](#_Toc31085)

[2.2表空间设计 4](#_Toc8834)

[2.3数据表设计 4](#_Toc7597)

[三、数据库表 4](#_Toc26909)

[3.1书籍表设计 4](#_Toc12137)

[3.2书的种类表设计 5](#_Toc1119)

[3.3管理员表设计 5](#_Toc26846)

[3.4借阅信息表设计 5](#_Toc16492)

[3.5学生信息表设计 6](#_Toc23828)

[四.物理结构设计 6](#_Toc17876)

[创建表空间 6](#_Toc5468)

[创建表 7](#_Toc30626)

[五．数据库实施代码 10](#_Toc21411)

[向表中插入数据 10](#_Toc5844)

[触发器设计 12](#_Toc26775)

[储存函数包设计 13](#_Toc32381)

[角色分配以及权限赋予 15](#_Toc17323)

[建立备份方案 17](#_Toc9771)

[六． 项目总结及体会 18](#_Toc21629)

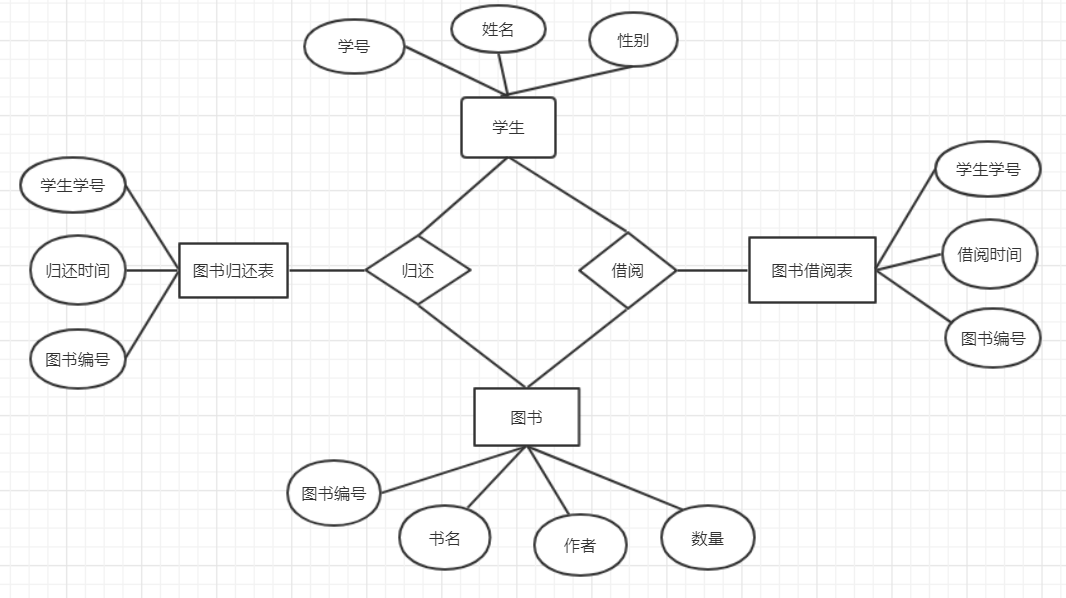
### 图书管理系统数据设计

## 一、引言

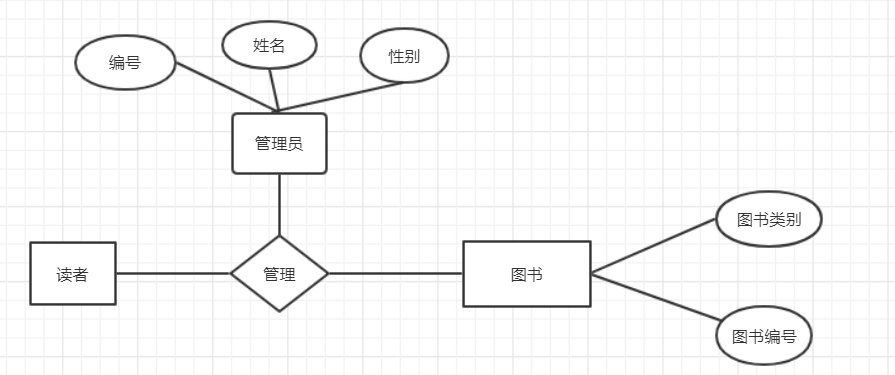
在图书馆的正常运行中，面临着大量的图书、读者信息和借阅、还书信息之间的互动产生。Oracle数据库系统是目前世界上流行的关系数据库管理系统，系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境。它是一种高效率、可靠性好的、适应高吞吐量的数据库方案。本系统是基于Oracle数据库的图书管理系统的数据库设计。

## 概念模型设计

### 2.1 E-R图设计



读者E-R模型



管理员E-R模型

### 2.2表空间设计

     创建一个表空间USERS02，用于保存书籍管理系统的各种表，首先需要定义数据文件的存储地址，并且定义该数据文件的大小为200M、的分配方式为自动扩展、表空间的管理方式为本地管理。

### 2.3数据表设计

创建五个表，存储在表空间users中。这五个表是学生表、图书表、借阅表和管理员表，书籍种类表。

学生表的作用是存放学生信息，学号，学生姓名，等等学生相关的信息。

图书表的作用是存放图书信息，编号，书名，作者，价格等图书的相关信息。

借阅表的作用是对学生表和图书表的一个链接，显示哪些人借的哪些书。借阅的日期和还书日期等信息。

书籍种类表的作用是将图书表的书进行分类，存放种类编号，种类名等信息。

管理员信息表的作用是用于存放管理员的账号，密码，姓名，加入日期等相关信息。

## 三、数据库表

### 3.1书籍表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| bookno | Number(10,0) | no | 书籍编号，主键 |
| bookname | Varchar2(20,BYTE) | no | 书名 |
| bookclass | Varchar2(20,BYTE) | no | 书的种类，书种类表外键 |
| writer | Varchar2(20,BYTE) | no | 作者 |
| price | Number(10,0) | no | 价格 |

### 3.2书的种类表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| classno | Number(10,0) | no | 书籍种类的编号，主键 |
| classname | Varchar2(20,BYTE) | no | 书籍的种类名 |

### 3.3管理员表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| Adminname | Varchar2(20,BYTE) | no | 管理员账号，主键 |
| pwd | Varchar2(20,BYTE) | no | 管理员密码 |
| mname | Varchar2(20,BYTE) | no | 管理员姓名 |
| Join\_date | DATE | no | 加入日期 |

### 3.4借阅信息表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| bookno | Varchar2(200,BYTE) | N0 | 书籍编号，书籍表外键 |
| studyno | Varchar2(200,BYTE) | N0 | 学生学号，学生表外键 |
| Borrow\_date | Varchar2(20,BYTE) | N0 | 借书日期 |
| Return\_date | Varchar2(20,BYTE) | N0 | 还书日期 |

### 3.5学生信息表设计

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| username | Varchar2(20,BYTE) | no | 学生账号 |
| studyno | Number(10,0) | no | 学生学号，主键 |
| name | Varchar2(10,BYTE) | no | 学生姓名 |
| password | Varchar2(20,BYTE) | no | 学生密码 |
| phone | Number(20,0) | no | 学生电话 |

## 四.物理结构设计

### 创建表空间

#### 第一个表空间

Create Tablespace space\_zjl001

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_zjl001\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_zjl001\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

#### 第二个表空间

Create Tablespace space\_zjl002

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_zjl002\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_zjl002\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

### 创建表

#### 创建读者表

CREATE TABLE "SCOTT"."READER"

(

  "STUDENT\_ID" INT PRIMARY KEY,

  "STUDENT\_NAME" VARCHAR(10) NOT NULL,

  "STUDENT\_GENDER" CHAR(2) NOT NULL,

  "STUDENT\_AGE" INT NOT NULL,

  " STUDENT\_PRO" VARCHAR(20) NOT NULL,

  " STUDENT\_GRADE" VARCHAR(20) NOT NULL,

  " STUDENT\_INTEGRITY" INT DEFAULT 1

)

TABLESPACE " zjl001";

ALTER TABLE "SCOTT"."READER" ADD CONSTRAINTS "Sex\_CK1" CHECK(Sex BETWEEN '男' AND '女');

#### 创建书籍表

CREATE TABLE "SCOTT"."BOOK"

(

  "BOOK\_ID" INT PRIMARY KEY,

  "BOOK\_NAME" VARCHAR(30) NOT NULL,

  "BOOK\_AUTHOR" VARCHAR(10) NOT NULL,

  "BOOK\_PUB" VARCHAR(40) NOT NULL,

  "BOOK\_NUM" INT NOT NULL,

  "BOOK\_SORT" VARCHAR(40) NOT NULL,

  " BOOK\_RECORD" DATETIME

)

TABLESPACE " zjl001";

#### 创建管理者表

CREATE TABLE "SCOTT"."MANAGER"

(

  "MANAGER\_ID" INT PRIMARY KEY,

  "MANAGER\_NAME" VARCHAR(10) NOT NULL,

  "MANAGER\_AGE" INT NOT NULL,

  " MANAGER\_PASSWORD" VARCHAR(20) NOT NULL,

  " MANAGER\_PHONE" VARCHAR(20) NOT NULL,)

TABLESPACE " zjl001";

#### 创建图书类别表

CREATE TABLE "SCOTT"."BOOK"

(

  "SORT\_ID" VARCHAR(10) PRIMARY KEY,

  "SORT\_NAME" VARCHAR(30) NOT NULLS

)

TABLESPACE " zjl001";

#### 创建借阅表

CREATE TABLE "SCOTT"."BORROW"

(

   "READER\_ID" INT REFERENCES "SCOTT"."READER"( READER\_ID),

   "BOOK\_ID" INT REFERENCES "SCOTT"."BOOK"( BOOK\_ID),

   "BORROW\_DATE" DATETIME,

   "EXPECT\_RETURN\_DATE" DATETIME

)

TABLESPACE " zjl002";

#### 创建归还表

CREATE TABLE "SCOTT"."RETURN"

(

   "READER\_ID" INT REFERENCES "SCOTT"."READER"( READER\_ID),

   "BOOK\_ID" INT REFERENCES "SCOTT"."BOOK"( BOOK\_ID),

   "BORROW\_DATE" DATETIME,

   "RETURN\_DATE" DATETIME

)

TABLESPACE " zjl002";

## 五．数据库实施代码

### 向表中插入数据

#### 图书信息表

declare

j NUMBER;

begin

j:=1;

for i in 1..1000

loop

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'ruin sience','sience','m.j','qinghua',to\_date('2015-09-18','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'go magazine','magazine','m.j','qinghua',to\_date('2005-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'push textbook','textbook','m.j','qinghua',to\_date('1903-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'ruin primer','primer','m.j','qinghua',to\_date('2014-09-18','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'go instruction','instruction','m.j','qinghua',to\_date('2003-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'push autobiography','autobiography','m.j','qinghua',to\_date('1906-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'ruin fable','fable','m.j','qinghua',to\_date('2019-09-18','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'go dictionary','dictionary','m.j','qinghua',to\_date('2008-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'push reference','reference','m.j','qinghua',to\_date('1908-3-2','yyyy-mm-dd'));j:=j+1;

insert into BOOK\_INFOR(BOOKNO,BOOKNAME,BOOKSTYLE,AUTHOR,PUBLISH,PUBLISHDATE) values(j,'push reference','reference','m.j','qinghua',to\_date('1988-3-2','yyyy-mm-dd'));j:=j+1;

end loop;

commit;

end;

#### 学生信息表

begin

for i in 1..2000

loop

insert into STUDENT(STUDENTNO,STUDENTNAME,STUDENTSEX,ISBORROWED) values('201510415354','ZHUMAOJIE','MAN','YES');

insert into STUDENT(STUDENTNO,STUDENTNAME,STUDENTSEX,ISBORROWED) values('201720418354','WULIUKA','WOMAN','NO');

insert into STUDENT(STUDENTNO,STUDENTNAME,STUDENTSEX,ISBORROWED) values('201612416354','LIUFUT','WOMAN','YES');

insert into STUDENT(STUDENTNO,STUDENTNAME,STUDENTSEX,ISBORROWED) values('201710415254','LIUCHEN','MAN','NO');

insert into STUDENT(STUDENTNO,STUDENTNAME,STUDENTSEX,ISBORROWED) values('201612415824','WANGWU','MAN','YES');

end loop;

commit;

end;

#### 图书种类表

begin

for i in 1..2000

loop

insert into book\_style(STYLENO,STYLENAME) values('1','magazine');

insert into book\_style(STYLENO,STYLENAME) values('2','textbook');

insert into book\_style(STYLENO,STYLENAME) values('3','primer');

insert into book\_style(STYLENO,STYLENAME) values('4','instruction');

insert into book\_style(STYLENO,STYLENAME) values('5','autobiography');

end loop;

commit;

end;

#### 借书还书表

begin

for i in 1..2000

loop

insert into BORROWRETURN\_BOOK(READERNO,BOOKNO,BORROWDATE,RETURNDATE) values('201510415354','8742',to\_date('2019-9-2','yyyy-mm-dd'),to\_date('2019-10-2','yyyy-mm-dd'));

insert into BORROWRETURN\_BOOK(READERNO,BOOKNO,BORROWDATE,RETURNDATE) values('201710415354','1585',to\_date('2019-6-2','yyyy-mm-dd'),to\_date('2019-7-2','yyyy-mm-dd'));

insert into BORROWRETURN\_BOOK(READERNO,BOOKNO,BORROWDATE,RETURNDATE) values('201810415354','3547',to\_date('2019-3-2','yyyy-mm-dd'),to\_date('2019-4-2','yyyy-mm-dd'));

insert into BORROWRETURN\_BOOK(READERNO,BOOKNO,BORROWDATE,RETURNDATE) values('201910415354','3585',to\_date('2019-7-2','yyyy-mm-dd'),to\_date('2019-8-2','yyyy-mm-dd'));

insert into BORROWRETURN\_BOOK(READERNO,BOOKNO,BORROWDATE,RETURNDATE) values('201710415354','5857',to\_date('2019-3-2','yyyy-mm-dd'),to\_date('2019-6-2','yyyy-mm-dd'));

end loop;

commit;

end;

### 触发器设计

1. **设计触发器borrow，当某学生借书成功后﹐图书表相应的图书不在架上,变为0∶**

create or replace trigger borrow

after insert on borrow

for each row

begin

update book set book\_num = book\_num -1

where book\_id = new.book\_id;

end borrow；

**⒉设计触发器trigger\_return，还书成功后﹐对应的书籍book\_num变为1:**

create or replace trigger trigger\_return

after insert on return\_table

for each row

begin

update book set book\_num = book\_num + 1

where book\_id = new.book\_id;

end trigger\_return;

**3.定义定时器（事件) eventJob ,每天自动触发一次﹐扫描视图stu\_borrow ，若发现当前有**

**预期归还时间小于当前时间,则判断为超期,生成处罚记录,这个定时器将每天定时触发存**

**储过程proc\_gen\_ticket :**

create event if not exists eventJob

on schedule every 1 DAYl\*每天触发\*\*

on completion PRESERVE

do call proc\_gen\_ticket(getdate());l\*调用存储过程\*/

set global event\_scheduler = 1;

alter event eventJob on completion preserve enable; \*开启定时器\*

### 储存函数包设计

#### 创建程序包

create or replace PACKAGE book\_package Is

function func\_get\_credit(stu\_id int) returns int;

function func\_get\_booknum(book\_id int) returns int;

procedure proc\_return(in stu\_id int, in book\_id int, in return\_date datetime);

procedure proc\_payoff(in stuid int, in bookid int);

procedure proc\_borrow(in stu\_id int,in book\_id int,in borrow\_date datetime);

procedure proc\_gen\_ticket(in currentdate datetime);

procedure stu\_register(in stu\_id int,in stu\_name varchar(20)，in stu\_sex varchar(20), in stu\_age int, in stu\_pro varchar(20), in stu\_grade varchar(20));

procedure ma\_register(in ma\_id int, in ma\_name varchar(20), in ma\_age int, in ma\_phone int);

end book\_package;

#### 创建存储过程

create or replace PACKAGE body book\_package Is

/\* 1．设计存储过程，产生罚单proc\_gen\_ticket :

当日期超过预定归还日期时﹐产生罚单，并将记录写入表ticket中,这个存储过程在定时器

eventJob中调用: \*/

procedure proc\_gen\_ticket(in currentdate datetime)

BEGIN

declare cur\_date datetime;

set cur\_date = currentdate;

replace into ticket(stu\_id, book\_id, over\_date, ticket\_fee)

select

stu\_id,

book\_id,

datediff(cur\_date,stu\_borrow.expect\_return\_date),0.1\*datediff(cur\_date,stu\_borrow.exp

ect\_return\_date)

from stu\_borrow

where cur\_date>stu\_borrow.expect\_return\_date;

end proc\_gen\_ticket;

/\* ⒉设计学生注册信息存储过程∶学生注册信息stu\_register \*/

procedure stu\_register(in stu\_id int,in stu\_name varchar(20)，in stu\_sex

varchar(20), in stu\_age int, in stu\_pro varchar(20), in stu\_grade varchar(20))

begin

insert into student(stu\_id, stu\_name, stu\_sex,stu\_age, stu\_pro, stu\_grade)

values(stu\_id,stu\_name, stu\_sex,stu\_age,stu\_pro,stu\_grade);

end stu\_register;

/\* 3.设计管理员注册信息存储过程:ma\_register \*/

procedure ma\_register(in ma\_id int, in ma\_name varchar(20), in ma\_age int, in

ma\_phone int)

BEGIN

insert into manager

values(ma\_id, ma\_name, ma\_age, ma\_phone);

END ma\_register;

/\* 4．借书过程的实现:

1)设计存储函数, func\_get\_credit ,返回学生的诚信级︰\*/

function func\_get\_credit(stu\_id int) returns int

begin

return(select stu\_integrity from student where student.stu\_id = stu\_id);

end

/\* 2)设计存储函数，func\_get\_booknum ,返回书籍是否在架上:\*/

function func\_get\_booknum(book\_id int) returns int

begin

return(select book\_num from book where book.book\_id = book\_id);

end

/\*3)设计存储过程proc\_borrow ,调用 func\_get\_booknum ,判断这个书籍是否在架上,若为真,则借书成功,在borrrow表中插入纪录;否则提示失败︰\*/

procedure proc\_borrow(in stu\_id int,in book\_id int,in borrow\_date datetime)

begin

if func\_get\_credit(stu\_id)= 1 and func\_get\_booknum(book\_id)= 1 then

insert into borrow

values(stu\_id, book\_id, borrow\_date);

else

select 'failed to borrow';

end if;

end proc\_borrow;

/\* 5.还书存储过程proc\_return :

当还书时,查看是否书是否超期,即查询ticket表项,当发现超期,提示交罚单后再次还书 ，

如没有超期，则纪录归还项目到return\_table中，并且删除借书纪录(以免还书后定时器仍

然扫描这个纪录): \*/

procedure proc\_return(in stu\_id int, in book\_id int, in return\_date datetime)

begin

DECLARE borrowdate datetime;

if(select payoff from ticket where ticket.stu\_id =stu\_idand

ticket.book\_id=book\_id)= 1 then l\*判断是否交了罚单﹐1表示没有交\*

select 'please pay off the ticket';

elsel\*纪录归还项目到return\_table 中，并且删除借书纪录\*/

set borrowdate = (select borrow\_date from borrow where borrow.stu\_id =

stu\_id and borrow.book\_id = book\_id);

insert into return\_table

values(stu\_id, book\_id, borrowdate, return\_date);

delete from borrow

where borrow.stu\_id = stu\_id and borrow.book\_id = book\_id;

end if;

end proc\_return;

### 角色分配以及权限赋予

**创建学生用户：**

create user student1

identified by 123

default tablespace users

quota 50M on users

password expire;

create user student2

identified by 123

default tablespace users

quota 50M on users

password expire;

**创建图书管理员用户：**

create user manager1

identified by 123

default tablespace users

quota 50M on users

password expire;

 create user manager2

identified by 123

default tablespace users

quota 50M on users

password expire;

**创建角色：**

create role student

not identified;

create role manager

not identified;

**将角色赋予用户：**

GRANT CONNECT,student TO student1;

GRANT CONNECT,student TO student2;

GRANT CONNECT,manager TO manager1;

GRANT CONNECT,manager TO manager2

**为角色赋予权限：**

grant select on "scott"."v\_reader\_c1" to student;

grant select on "scott"."v\_reader\_c2" to student;

grant select on "scott"."stu\_borrow” to student;

grant select on "scott"." stu\_borrow\_return” to student;

grant connect,resource,execute to manager;

grant select,insert,delete,update on "scott"."v\_manager\_c1" to manager;

grant select,insert,delete,update on "scott"." stu\_borrow " to manager;

### 建立备份方案

备份脚本如下：

#rman\_level1.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

rmantarget / nocatalog msglog=/home/oracle/rman\_backup/lv1\_`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 1 database format '/home/oracle/rman\_backup/dblv1\_%d\_%T\_%U.bak'

plus archivelog format '/home/oracle/rman\_backup/arclv1\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

release channel c1;

}

EOF

Exit

为了测试备份功能是否能够使用，现在模拟进行数据损坏备份，首先删除数据库文件：

[oracle@oracle-pc~]$rm /home/oracle/app/oracle/oradata/orcl/pdborcl/SAMPLE\_SCHEMA\_users01.dbf

然后通过备份脚本进行数据恢复：

sqlplus / as sysdba

SQL>shutdown immediate

SQL>shutdown abort

SQL>startup mount

rman target /

## 项目总结及体会

Oracle数据库系统是目前世界上流行的关系数据库管理系统，系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境。它是一种高效率、可靠性好的、适应高吞吐量的数据库方案。通过自己这个基于Oracle的书籍管理系统，体会到Oracle数据库与之前接触过的数据库SQL语句有些不一样，除了常用的增删查改与其他主关系型数据库相似，其他的语句还是有一些区别的。我发现自己的数据库设计能力还稍有不足，对图书管理系统的几个表的设计不是很合理，表中有些字段的类型和长度可能与实际应用不相符。在进行数据库设计时应该考虑实际项目中的应用，这样才能设计出满足实际需求的数据库管理系统。这次期末项目加深理解了之前课上做的各个实验，同时也巩固了课上学的知识，提高了自己的能力。