

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的图书信息管理系统的数据库设计 | | |
| 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 | 袁海洋 | 学号 | 201810414427 |
| 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

# orcale数据库实验六-图书信息管理系统

[orcale数据库实验六-成都大学信息系统](#header-n0)  
 [一. 概述](#header-n4)  
 [二. 需求与功能分析](#header-n7)   
 [三. 数据库设计](#header-n9)  
 [1. 添加用户及权限管理](#header-n10)  
 [创建student角色，并创建student\_lft用户，并且给用户分配角色空间](#header-n24)  
 [2. 通过新创建的用户student\_lft连接到 pdborcl](#header-n28)   
 [3. 创建testuser角色并创建testuser\_lft用户](#header-n30)  
 [4. 利用新创建的用户student\_lft创建了五个表](#header-n34)  
 [6. 创建视图](#header-n138)  
 [7. 将五个表的视图的SELECT对象权限授予testuser用户](#header-n151)   
 [8. 向数据库中写入数据共计50000多个数据](#header-n153)  
 [8.1 用户表](#header-n154)  
 [8.2 管理员表](#header-n157)  
 [8.3 图书表](#header-n160)  
 [8.4 借书记录表](#header-n163)  
 [8.5 还书记录表](#header-n166)  
 [9. PL/SQL设计](#header-n169)  
 [9.1 存储过程：](#header-n190)  
 [9.1.1](#header-n191) 插入数据  
 [9.2. 创建函数](#header-n197)  
 [9.2.1. 查询随机图书id](#header-n198)  
 [10. 数据库备份](#header-n202)  
 [热备份和冷备份优缺点](#header-n205)  
 [热备份的优点是：](#header-n206)  
 [热备份的不足是：](#header-n212)  
 [冷备份的优点是：](#header-n216)  
 [冷备份不足是：](#header-n222)  
 *数据导出*

*数据导入*

## 一. 概述

随着电脑、手机等信息工具的普及，人们能更加完善、快捷地处理信息数据。在图书馆的日常运作中，由于图书借阅和用户查询次数繁多，需要进行繁杂的信息数据管理。

书籍是人类不可缺少的精神食粮,尤其重要。所以图书馆借阅信息管理系统应该能够为用户提供充足的信息和快捷的查询手段。但一直以来人们使用传统人工的方式管理图书、期刊、试卷合订本等查询信息,这种查询管理方式存在着许多缺点,如:效率低、保密性差,另外时间一长,将产生大量的文件和数据,这对于查找、更新和维护都带来了不少的困难。随着科学技术的不断提高,计算机科学日渐成熟,其强大的功能已为人们深刻认识。作为计算机应用的一部分,使用计算机对图书资源信息进行查询管理,具有着手工管理所无法比拟的优点.例如:检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高图书查询管理的效率,也是企业、学校的科学化、正规化管理,与世界接轨的重要条件。为了能更高效、快捷、稳定地管理图书馆的数据信息，本文设计并实现了一套图书信息管理系统。

## 二. 需求与功能分析

图书信息管理系统，可用于学校等机构图书信息管理，查询，更新与维护，使用方便，易用性强。该系统实现的大致功能；用户登陆。提供了用户信息的查询，添加，修改，删除；图书信息的录入，修改，删除，查询，修改密码等功能。管理员管理拥有最高的权限。允许添加图书信息和用户信息等。其提供了简单.方便的操作。

## 三. 数据库设计

## 1. 添加用户及权限管理

oracle中的表就是一张存储数据的表。表空间是逻辑上的划分。方便管理的。

数据表空间 (Tablespace)

存放数据总是需要空间， Oracle把一个数据库按功能划分若干空间来保存数据。当然数据存放在磁盘最终是以文件形式，所以一盘一个数据表空间包含一个以上的物理文件  
数据表。

在仓库，我们可能有多间房子，每个房子又有多个货架，每架又有多层。 我们在数据库中存放数据，最终是数据表的单元来存储与管理的。  
数据文件。

以上几个概念都是逻辑上的， 而数据文件则是物理上的。就是说，数据文件是真正“看得着的东西”，它在磁盘上以一个真实的文件体现。

创建表空间：

格式: create tablespace 表间名 datafile '数据文件名' size 表空间大小  
 create tablespace data\_test datafile 'e:\oracle\oradata\test\data\_1.dbf' size 2000M;  
 create tablespace idx\_test datafile 'e:\oracle\oradata\test\idx\_1.dbf' size 2000M;  
 (\*数据文件名 包含全路径, 表空间大小 2000M 表是 2000兆)

建好tablespace, 就可以建用户

格式: create user 用户名 identified by 密码 default tablespace 表空间表;  
 create user study identified by study default tablespace data\_test;  
 (\*我们创建一个用户名为 study,密码为 study, 缺少表空间为 data\_test -这是在第二步建好的.)  
 (\*缺省表空间表示 用户study今后的数据如果没有专门指出，其数据就保存在 data\_test中, 也就是保存在对应的物理文件 e:\oracle\oradata\test\data\_1.dbf中)

创建用户并指定表空间

CREATE USER admin IDENTIFIED BY admin PROFILE DEFAULT DEFAULT TABLESPACE ADMIN ACCOUNT UNLOCK;  
create user user identified by user default tablespace user temporary tablespace user\_temp;  
授权给新用户  
GRANT connect, resource TO user;  
grant create session to user;

授权给新用户

grant connect,resource to user01;   
 --表示把 connect,resource权限授予user01用户  
 grant dba to user01;  
 --表示把 dba权限授予给 user01

创建数据表

在上面，我们已建好了用户 study 我们现在进入该用户

sqlplusw study/user01@test 然后就可以在用户user中创建数据表了

格式: create table 数据表名

## 4. 创建五个表

CREATE TABLE users(

user\_id number NOT NULL,

username varchar(20) NOT NULL,

passwd varchar(20) NOT NULL,

uname varchar(20) NOT NULL,

sex varchar(5) NOT NULL,

book\_quota number NOT NULL,

overdue\_num number NOT NULL,

PRIMARY KEY (user\_id)

);

CREATE TABLE admin (

admin\_id number NOT NULL,

username varchar(16) NOT NULL,

passwd varchar(16) NOT NULL,

uname varchar(16) NOT NULL,

sex varchar(5) NOT NULL,

book\_quota number NOT NULL,

overdue\_num number NOT NULL,

PRIMARY KEY (admin\_id)

);

CREATE TABLE book(

book\_id number NOT NULL,

ISBN varchar(20) UNIQUE NOT NULL,

book\_name varchar(50) NOT NULL,

author varchar(50) NOT NULL,

publishing\_house varchar(50) NOT NULL,

surplus number NOT NULL,

PRIMARY KEY (book\_id)

);

CREATE TABLE borrow\_record(

borrow\_books\_record\_id number NOT NULL,

user\_id number NOT NULL,

ISBN varchar(50) NOT NULL,

lend\_time date NOT NULL,

lend\_days number NOT NULL,

PRIMARY KEY (borrow\_books\_record\_id),

CONSTRAINT user\_id FOREIGN KEY (user\_id) REFERENCES users (user\_id),

CONSTRAINT ISBN FOREIGN KEY (ISBN) REFERENCES book (ISBN)

);

CREATE TABLE return\_record(

id number NOT NULL,

borrow\_books\_record\_id number NOT NULL,

return\_time date NOT NULL,

is\_overdue varchar(5) NOT NULL,

PRIMARY KEY (id),

CONSTRAINT borrow\_books\_record\_id FOREIGN KEY (borrow\_books\_record\_id) REFERENCES borrow\_record (borrow\_books\_record\_id)

)

用户表：

| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| --- | --- | --- | --- |
| user\_id | number | not null | 主键 |
| username | varchar | not null | 用户名 |
| passwd | varchar | not null | 密码 |
| uname | varchar | not null | 姓名 |
| sex | varchar | not null | 性别 |
| book\_quota | number | not null | 借书限额 |
| overdue\_num | number | not null | 逾期次数 |

#### 管理员表：

| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| --- | --- | --- | --- |
| admin\_id | number | not null | 主键 |
| username | varchar | not null | 用户名 |
| passwd | varchar | not null | 密码 |
| uname | varchar | not null | 姓名 |
| sex | varchar | not null | 性别 |
| book\_quota | number | not null | 借书限额 |
| overdue\_num | number | not null | 逾期次数 |

#### 图书表：

| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| --- | --- | --- | --- |
| book\_id | number | not null | 主键 |
| ISBN | varchar | not null | 书籍ISBN号 |
| book\_name | varchar | not null | 书名 |
| author | varchar | not null | 作者 |
| publishing\_house | varchar | not null | 出版社 |
| surplus | number | not null | 可借余量 |

#### 借书记录表：

| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| --- | --- | --- | --- |
| borrow\_books\_record\_id | number | not null | 借书记录id（主键） |
| user\_id | number | not null | 用户id（外键） |
| ISBN | varchar | not null | 书籍ISBN号（外键） |
| lend\_time | date | not null | 借出日期 |
| lend\_days | number | not null | 借出天数 |

#### 还书记录表：

| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| --- | --- | --- | --- |
| id | number | not null | 主键 |
| borrow\_books\_record\_id | number | not null | 借书记录id（外键） |
| return\_time | varchar | not null | 归还日期 |
| is\_overdue | varchar | not null | 是否逾期 |

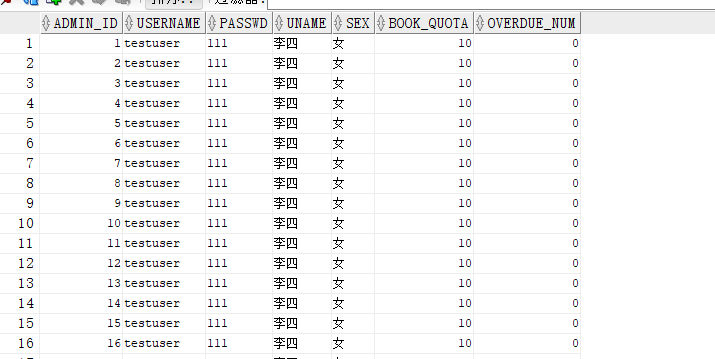
6. 创建视图

* 视图(view)，也称虚表, 不占用物理空间，这个也是相对概念，因为视图本身的定义语句还是要存储在数据字典里的。视图只有逻辑定义。每次使用的时候,只是重新执行SQL。
* 视图是从一个或多个实际表中获得的，这些表的数据存放在数据库中。那些用于产生视图的表叫做该视图的基表。一个视图也可以从另一个视图中产生。
* 视图的定义存在数据库中，与此定义相关的数据并没有再存一份于数据库中。通过视图看到的数据存放在基表中。
* 视图看上去非常象数据库的物理表，对它的操作同任何其它的表一样。当通过视图修改数据时，实际上是在改变基表中的数据；相反地，基表数据的改变也会自动反映在由基表产生的视图中。由于逻辑上的原因，有些Oracle视图可以修改对应的基表，有些则不能（仅仅能查询）。
* 还有一种视图：物化视图（MATERIALIZED VIEW ），也称实体化视图，快照 （8i 以前的说法） ，它是含有数据的，占用存储空间。  
  8. 向数据库中写入数据共计50000多个数据
* 创建序列：
* -- 创建普通用户id序列
* create sequence seq\_newUserids increment by 1 start with 1 maxvalue 999999999;
* -- 创建管理员id序列
* create sequence seq\_newAdminids increment by 1 start with 1 maxvalue 999999999;
* -- 创建图书id序列
* create sequence seq\_newBookids increment by 1 start with 1 maxvalue 999999999;
* -- 创建图书ISBN序列
* create sequence seq\_newISBNs increment by 1 start with 10000000 maxvalue 999999999;
* -- 创建借书记录id序列
* create sequence seq\_newBorrow\_books\_recordids increment by 1 start with 1 maxvalue 999999999;
* -- 创建还书记录id序列
* create sequence seq\_newids increment by 1 start with 1 maxvalue 999999999;

调用存储过程插入数据：

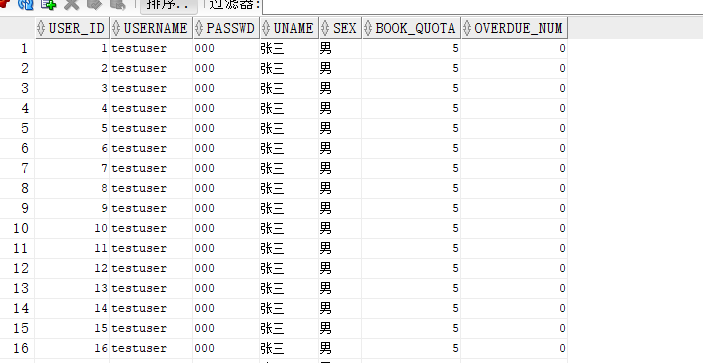
管理员数据

call insertdata();



用户数据

call insertadmindata();



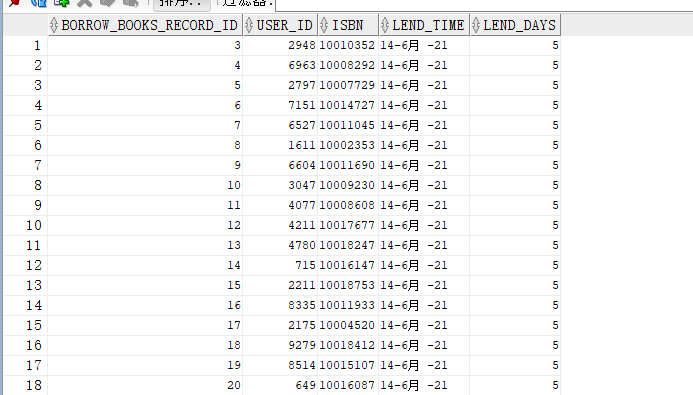
图书数据

call insertbookdata();



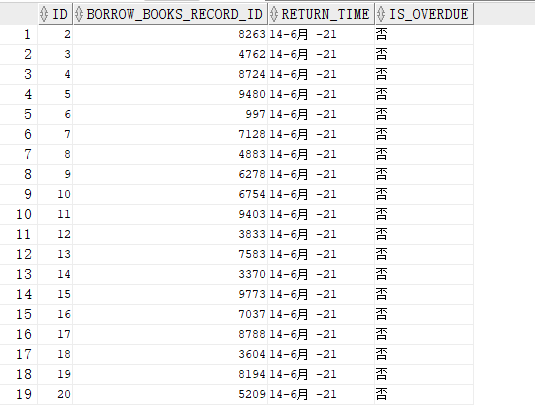
借书记录数据

call insertlendbookdata();



还书记录数据

call insertreturnbookdata();



## 9. PL/SQL设计

过程和函数由以下4部分：

* 签名或头
* 关键字IS或AS
* 局部声明（可选）
* BEGIN和END之间的过程体（包括异常处理程序）

简单示例：

create or replace procedure show\_line(ip\_line\_length in number, ip\_separator in varchar2)  
is   
actual\_line varchar2(150);  
begin  
 insert into t\_user(id,name,sex)values(ip\_line\_length,ip\_separator,ip\_line\_length);  
 for idx in 1..ip\_line\_length loop  
 actual\_line := actual\_line||ip\_separator;  
 end loop;  
 dbms\_output.put\_line(actual\_line);  
exception when others then  
 dbms\_output.put\_line(SQLERRM);  
end;

如下调用：

begin show\_line(50,'=');end;/

在SQLPLUS里面调用：

SQL> BEGIN2 show\_line(50,'=');3 END;

几点说明：

1、参数没有指定长度，当有实际数据传递进来的时候，参数的长度才被确定。

2、局部声明为：actual\_line varchar2(150);

3、使用命令SQL> show errors在SQLPLUS里面查看错误。

### 9.1 存储过程：

#### 9.1.1 插入数据

create or replace

PROCEDURE insertreturnbookdata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into return\_record(id,borrow\_books\_record\_id,return\_time,is\_overdue)

values(SEQ\_NEWIDS.nextval,getborrowid(),SYSDATE(),'否');

flag:=flag+1;

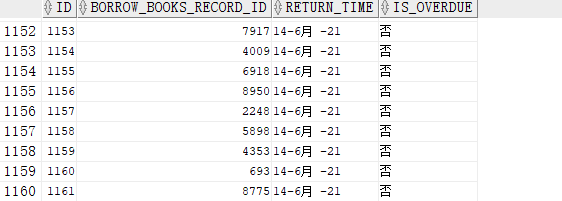
if flag=10001 then

commit;

end if;

end loop;

end;



### 9.2. 创建函数

#### 9.2.1. 随机获得一个图书id

create or replace function getborrowid

return number

as

borrowid number;

begin

select borrow\_books\_record\_id into borrowid from (select \* from borrow\_record order by dbms\_random.value) where rownum <2;

return borrowid;

end;



## 

## 10. 数据库备份

**ORACLE数据库备份与恢复详解**

Oracle的备份与恢复有三种标准的模式，大致分为两 大类，备份恢复(物理上的)以及导入导出(逻辑上的)，而备份恢复又可以根据数据库的工作模式分为非归档模式(Nonarchivelog-style) 和归档模式(Archivelog-style),通常，我们把非归档模式称为冷备份，而相应的把归档模式称为热备份。

### 热备份和冷备份优缺点

#### 热备份的优点是：

1．可在表空间或数据文件级备份，备份时间短。

2．备份时数据库仍可使用。

3．可达到秒级恢复（恢复到某一时间点上）。

4．可对几乎所有数据库实体作恢复。

5．恢复是快速的，在大多数情况下在数据库仍工作时恢复。

#### 热备份的不足是：

1．不能出错，否则后果严重。

2．若热备份不成功，所得结果不可用于时间点的恢复。

3．因难维护，所以要特别仔细小心，不允许“以失败而告终”。

#### 冷备份的优点是：

1．是非常快速的备份方法（只需拷贝文件）

2．容易归档（简单拷贝即可）

3．容易恢复到某个时间点上（只需将文件再拷贝回去）

4．能与归档方法相结合，作数据库“最新状态”的恢复。

5．低度维护，高度安全。

#### 冷备份不足是：

1．单独使用时，只能提供到“某一时间点上”的恢复。

2．在实施备份的全过程中，数据库必须要作备份而不能作其它工作。也就是说，数据库必须是关闭状态。

3．若磁盘空间有限，只能拷贝到磁带等其它外部存储设备上，速度会很慢。

4．不能按表或按用户恢复。

数据导出：

  1 将数据库orcl完全导出,用户名system 密码manager 导出到D:\daochu.dmp中

exp pdborcl/123@orcl file=d:\daochu.dmp full=y

  2 将数据库中pdborcl用户与sys用户的表导出

exp pdborcl/123@orcl file=d:\daochu.dmp owner=(system,sys)

  3 将数据库中的表table1 、table2导出

exp pdborcl/123@orcl file=d:\daochu.dmp tables=(table1,table2)

数据的导入

  1 将D:\daochu.dmp 中的数据导入 orcl数据库中。

imp pdborcl/123@orcl file=d:\daochu.dmp

  上面可能有点问题，因为有的表已经存在，然后它就报错，对该表就不进行导入。

  在后面加上 ignore=y 就可以了。

  2 将d:\daochu.dmp中的表table1 导入

imp pdborcl /123@orcl file=d:\daochu.dmp tables=(table1)