

**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 | 朱金龙 | 学号 | 201810414430 |
| 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 15 日

**目录**

[一、 概述 3](#_Toc14129)

[二、 需求与功能分析 3](#_Toc11097)

[三、 数据库设计 3](#_Toc18312)

[1. 创建用户及权限管理 3](#_Toc15859)

[创建表空间： 3](#_Toc6909)

[创建用户并指定表空间： 4](#_Toc8043)

[新用户授权： 5](#_Toc27184)

[2. 表设计 6](#_Toc16802)

[用户表： 6](#_Toc5174)

[管理员表： 7](#_Toc20782)

[图书表： 8](#_Toc32218)

[借书记录表： 9](#_Toc2064)

[还书记录表： 10](#_Toc18414)

[3. 索引设计 10](#_Toc32575)

[创建普通用户id序列： 10](#_Toc32062)

[创建管理员id序列： 10](#_Toc15710)

[创建图书id序列： 11](#_Toc7382)

[创建图书ISBN序列： 11](#_Toc29785)

[创建借书记录id序列： 11](#_Toc19387)

[创建还书记录id序列： 11](#_Toc11459)

[4. 存储过程 11](#_Toc4435)

[users表插入数据： 11](#_Toc21788)

[admin表插入数据： 12](#_Toc23556)

[book表插入数据： 13](#_Toc7286)

[borrow\_record表插入数据： 14](#_Toc14292)

[return\_record表插入数据： 15](#_Toc31501)

[5. 函数设计 17](#_Toc12054)

[随机获得用户id： 17](#_Toc8398)

[随机获得ISBN： 17](#_Toc19810)

[随机获得一个借书id： 18](#_Toc223)

[6. 备份 19](#_Toc16237)

[数据库备份与恢复： 19](#_Toc10919)

[热备份： 19](#_Toc7801)

[冷备份： 19](#_Toc25585)

[全备份： 20](#_Toc11383)

[7. 容灾方案 21](#_Toc21769)

基于Oracle的图书管理系统的数据库设计

## 概述

随着计算机技术的不断发展，计算机作为知识经济时代的产物，已被广泛应用于社会各个行业和领域。目前，我国的科技水平高速发展，计算机作为今天使用最广的现代化工具已深入到各个领域，并且正在成为未来社会——信息社会的重要支柱。在这样的大背景下，现代图书馆的管理方式，资源建设等方面都发生了重大变化，这种变化表现在图书馆工作，管理和服务平台发生的变化，图书馆不再是传统的手工操作，人工管理，而是全面实行计算机管理。

图书馆的正常运营中总是面对大量的读者信息，图书信息以及两者相互作用产生的借书信息，因此要对读者资源，图书资源，借书信息进行管理。本系统的开发就是在于提高图书管理的工作效率，加强图书馆的管理，全面实行计算机管理。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。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。

## 需求与功能分析

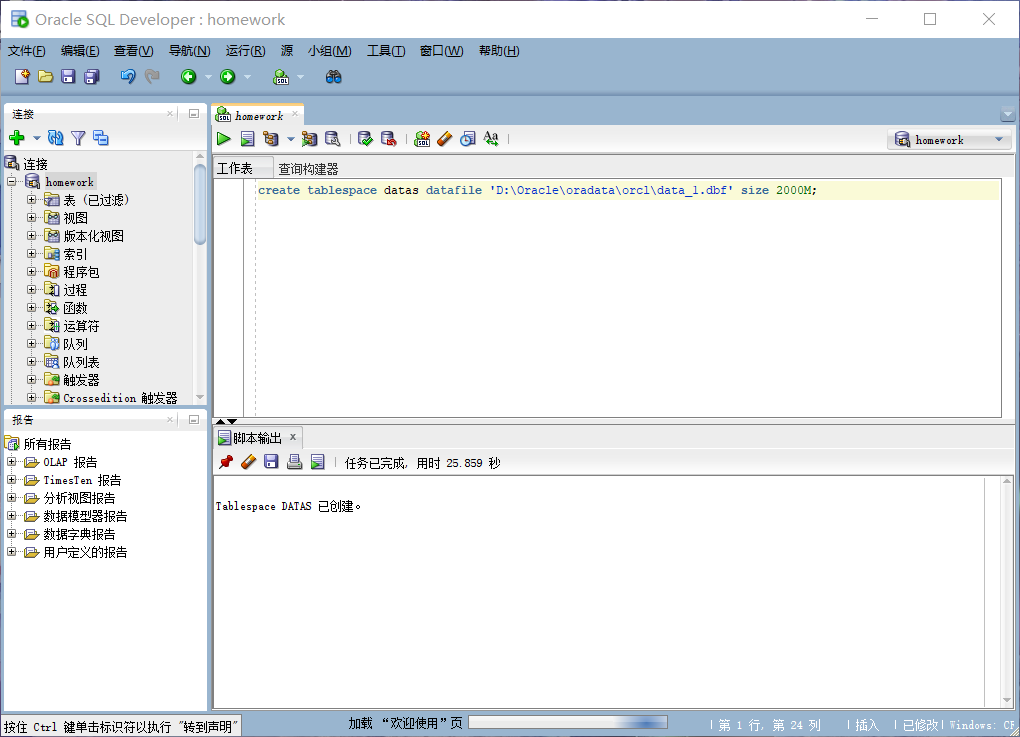
图书信息管理系统，可用于学校、图书馆等机构进行图书信息管理，查询，更新与维护，使用方便，易用性强。该系统实现的大致功能；用户的登录、注册、注销；用户信息的查询，添加，修改，删除；图书信息的录入，修改，删除，查询；用户对图书的查询、借阅、归还、赔偿等功能。管理员管理拥有最高的权限，允许对图书信息和用户信息进行修改操作。该系统提供了简单方便的操作。

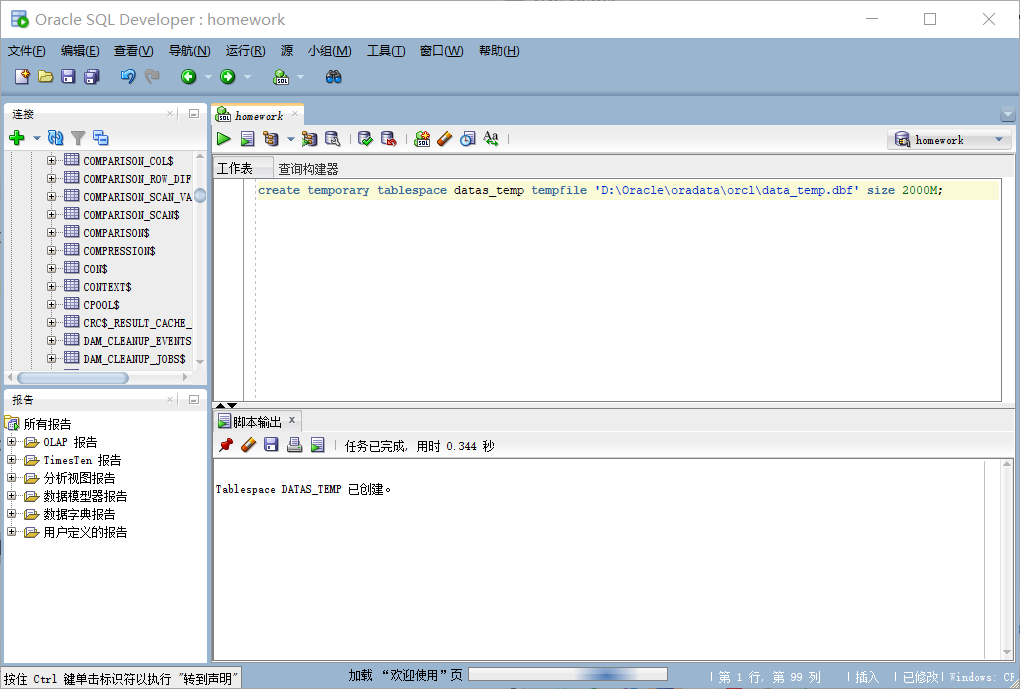
1. 数据库设计
2. 创建用户及权限管理

创建表空间：

create tablespace datas datafile 'D:\Oracle\oradata\orcl\data\_1.dbf' size 2000M;

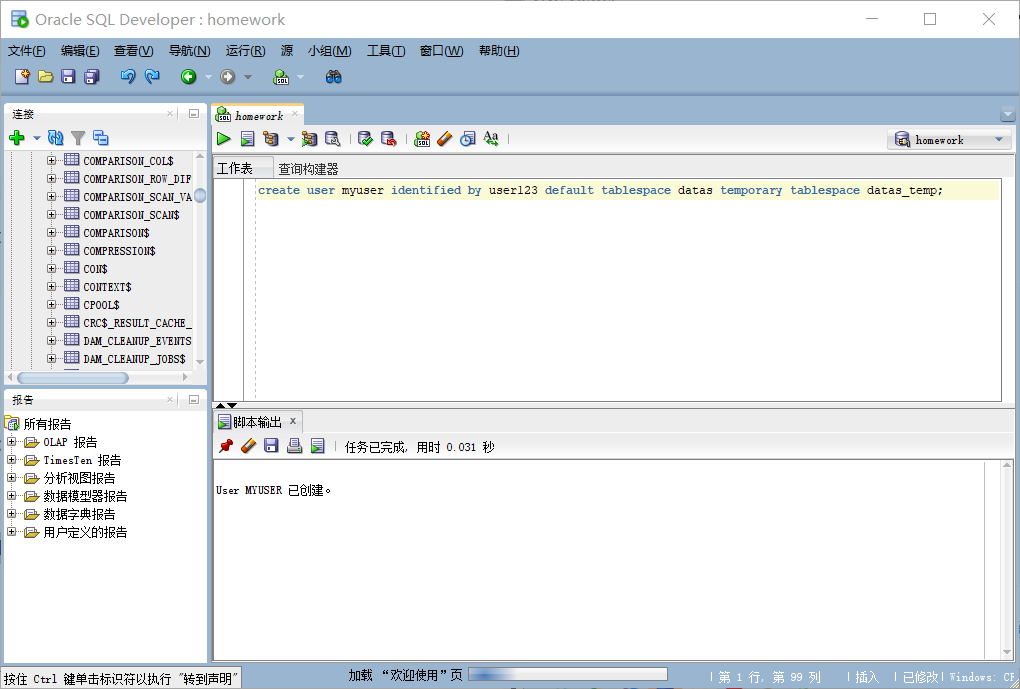
create tablespace datas\_temp datafile 'D:\Oracle\oradata\orcl\data\_temp.dbf' size 2000M;





创建用户并指定表空间：

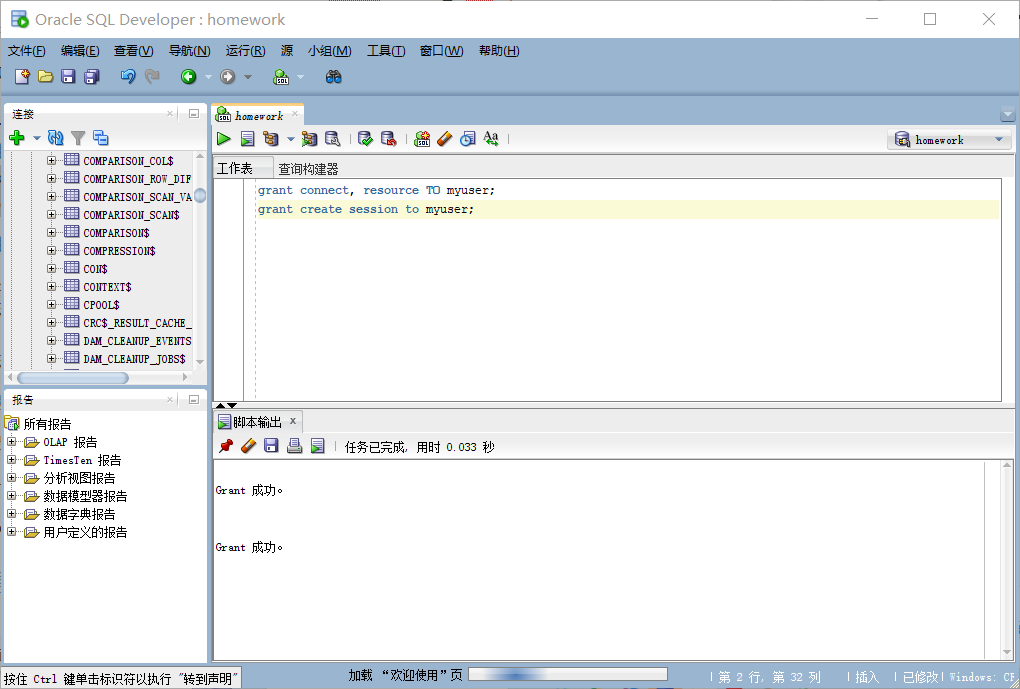
create user user identified by user default tablespace datas temporary tablespace datas\_temp;



新用户授权：

grant connect, resource TO myuser;

grant create session to myuser;



1. 表设计

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 |
| password | varchar | not null | 密码 |
| name | varchar | not null | 姓名 |
| sex | varchar | not null | 性别 |
| quota | number | not null | 借书限额 |
| overdue | number | not null | 逾期次数 |

CREATE TABLE users(

user\_id number NOT NULL,

username varchar(20) NOT NULL,

password varchar(20) NOT NULL,

name varchar(20) NOT NULL,

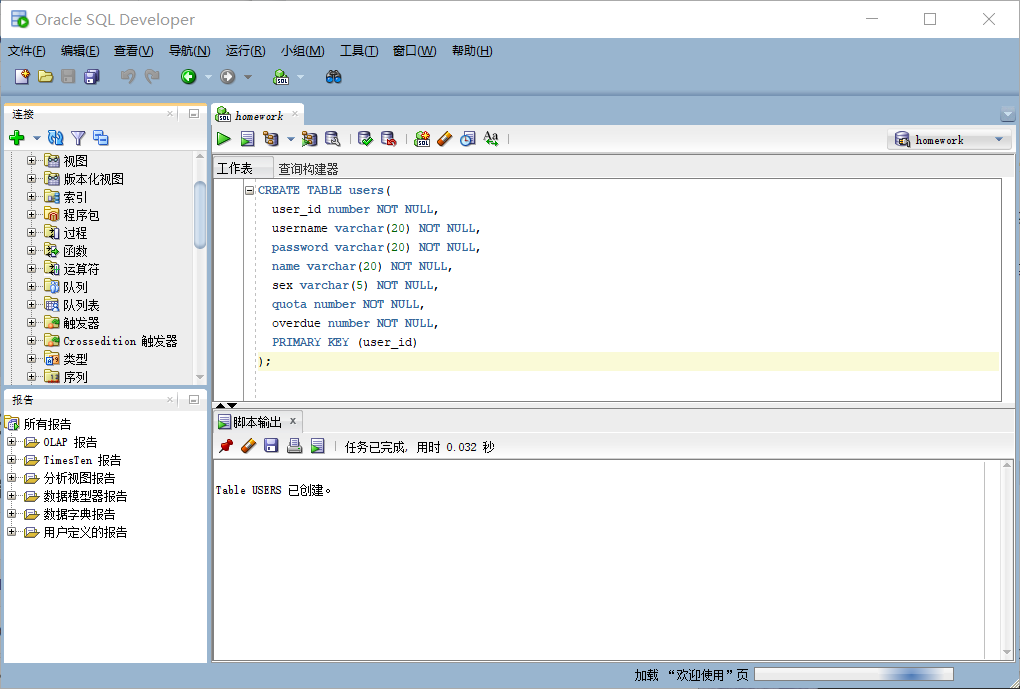
sex varchar(5) NOT NULL,

quota number NOT NULL,

overdue number NOT NULL,

PRIMARY KEY (user\_id)

);



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 |
| password | varchar | not null | 密码 |
| name | varchar | not null | 姓名 |
| sex | varchar | not null | 性别 |
| power | varchar | not null | 权限 |

CREATE TABLE admin (

admin\_id number NOT NULL,

username varchar(16) NOT NULL,

password varchar(16) NOT NULL,

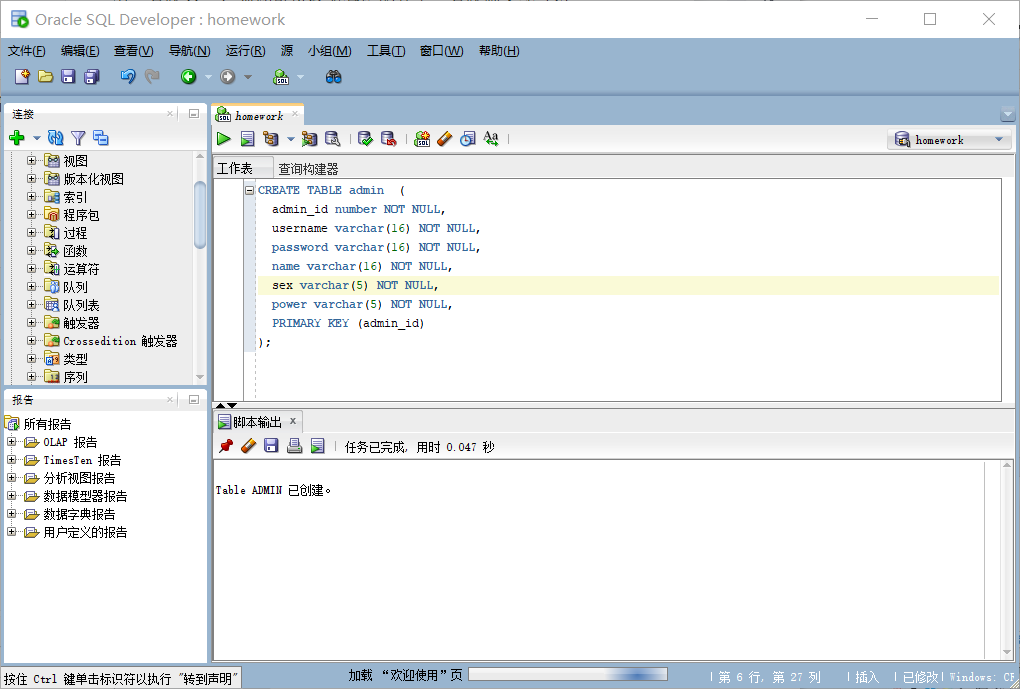
name varchar(16) NOT NULL,

sex varchar(5) NOT NULL,

power varchar(5) NOT NULL,

PRIMARY KEY (admin\_id)

);



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 |
| bookname | varchar | not null | 书名 |
| author | varchar | not null | 作者 |
| press | varchar | not null | 出版社 |
| Inventory | number | not null | 库存量 |

CREATE TABLE book(

book\_id number NOT NULL,

ISBN varchar(20) UNIQUE NOT NULL,

bookname varchar(50) NOT NULL,

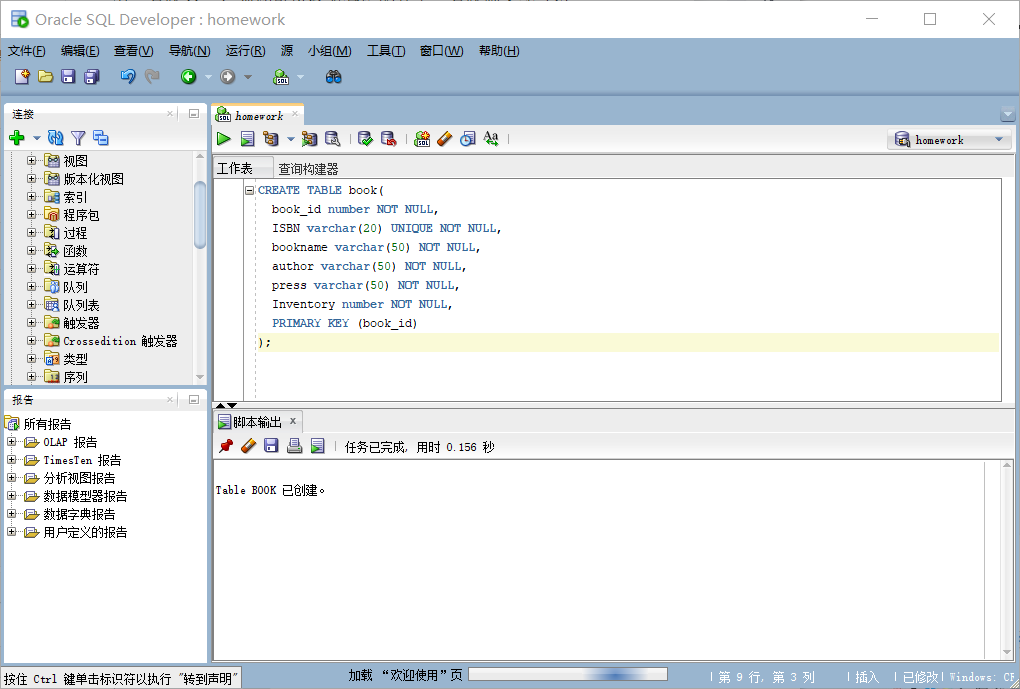
author varchar(50) NOT NULL,

press varchar(50) NOT NULL,

Inventory number NOT NULL,

PRIMARY KEY (book\_id)

);



借书记录表：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **是否允许NULL** | **备注** |
| borrow\_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） |
| borrow\_date | date | not null | 借出日期 |
| term | number | not null | 借出天数 |

CREATE TABLE borrow\_record(

borrow\_id number NOT NULL,

user\_id number NOT NULL,

ISBN varchar(50) NOT NULL,

borrow\_date date NOT NULL,

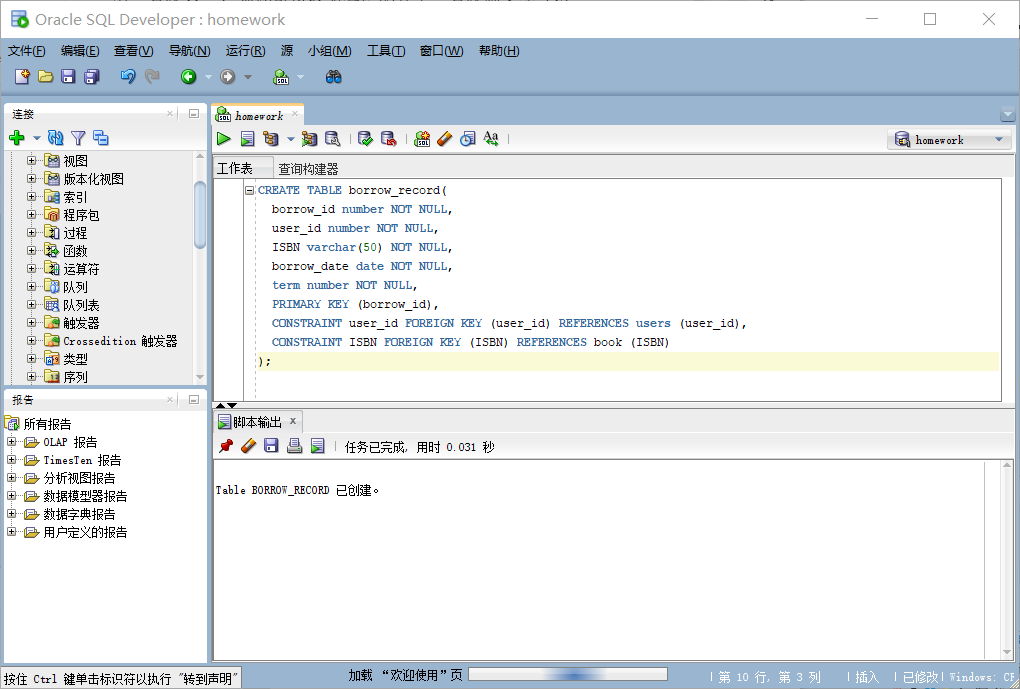
term number NOT NULL,

PRIMARY KEY (borrow\_id),

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

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

);



还书记录表：

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

CREATE TABLE return\_record(

return\_id number NOT NULL,

borrow\_id number NOT NULL,

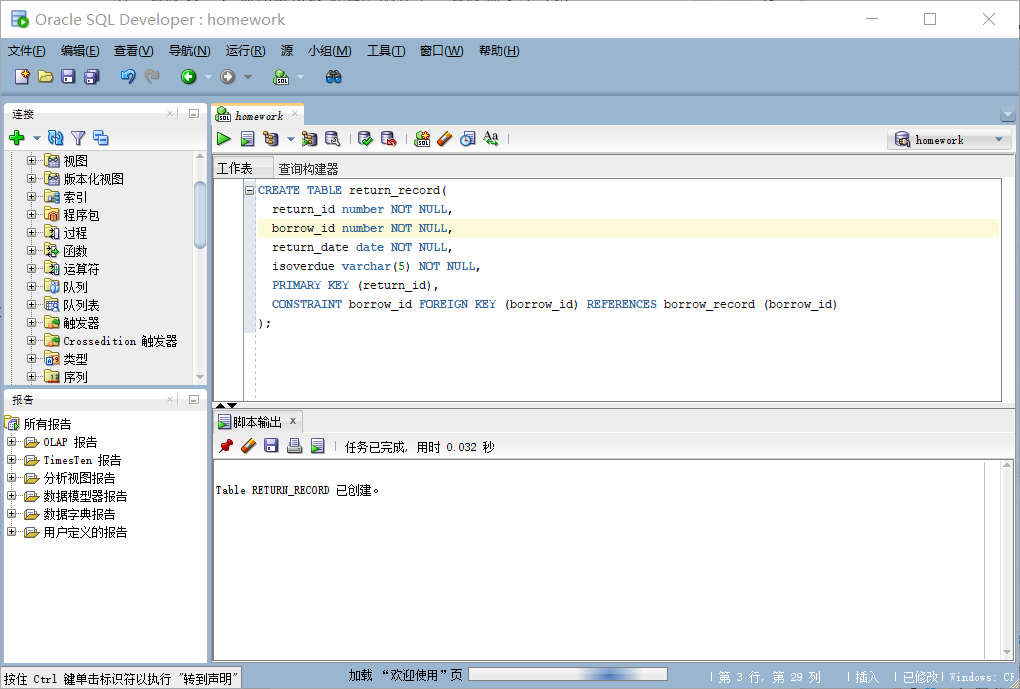
return\_date date NOT NULL,

isoverdue varchar(5) NOT NULL,

PRIMARY KEY (return\_id),

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

);



1. 索引设计

创建普通用户id序列：

create sequence new\_userid increment by 1 start with 1 maxvalue 999999999;

创建管理员id序列：

create sequence new\_adminid increment by 1 start with 1 maxvalue 999999999;

创建图书id序列：

create sequence new\_bookid increment by 1 start with 1 maxvalue 999999999;

创建图书ISBN序列：

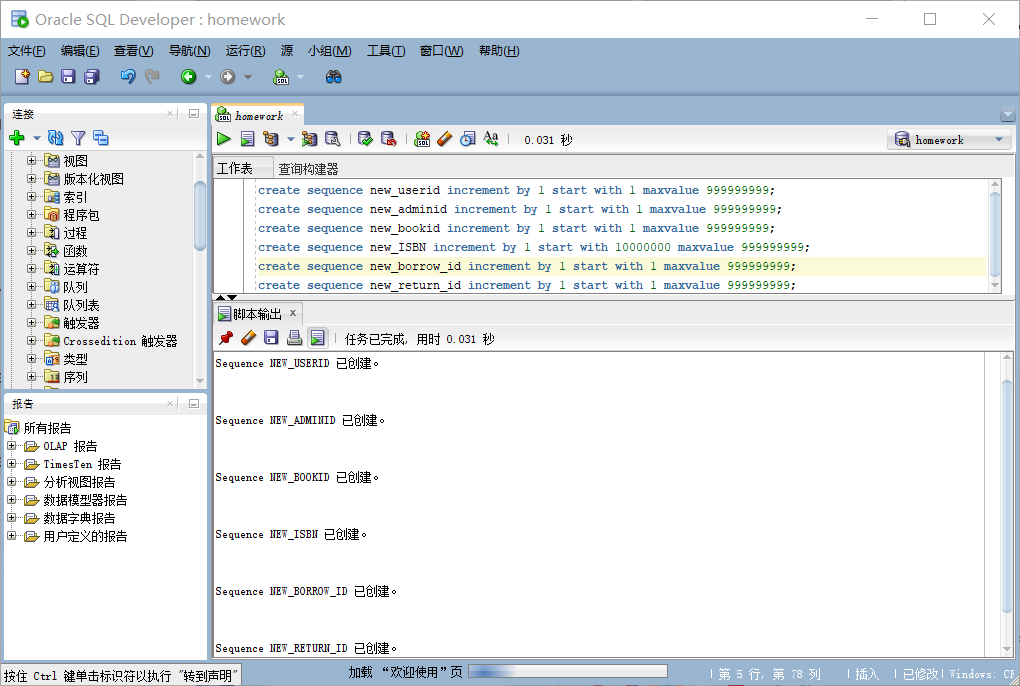
create sequence new\_ISBN increment by 1 start with 10000000 maxvalue 999999999;

创建借书记录id序列：

create sequence new\_borrow\_id increment by 1 start with 1 maxvalue 999999999;

创建还书记录id序列：

create sequence new\_return\_id increment by 1 start with 1 maxvalue 999999999;



1. 存储过程

users表插入数据：

create or replace

PROCEDURE insertdata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into users(user\_id,username,password,name,sex,quota,overdue)

values(new\_userid.nextval,'testuser','000','jzsj','男',3,0);

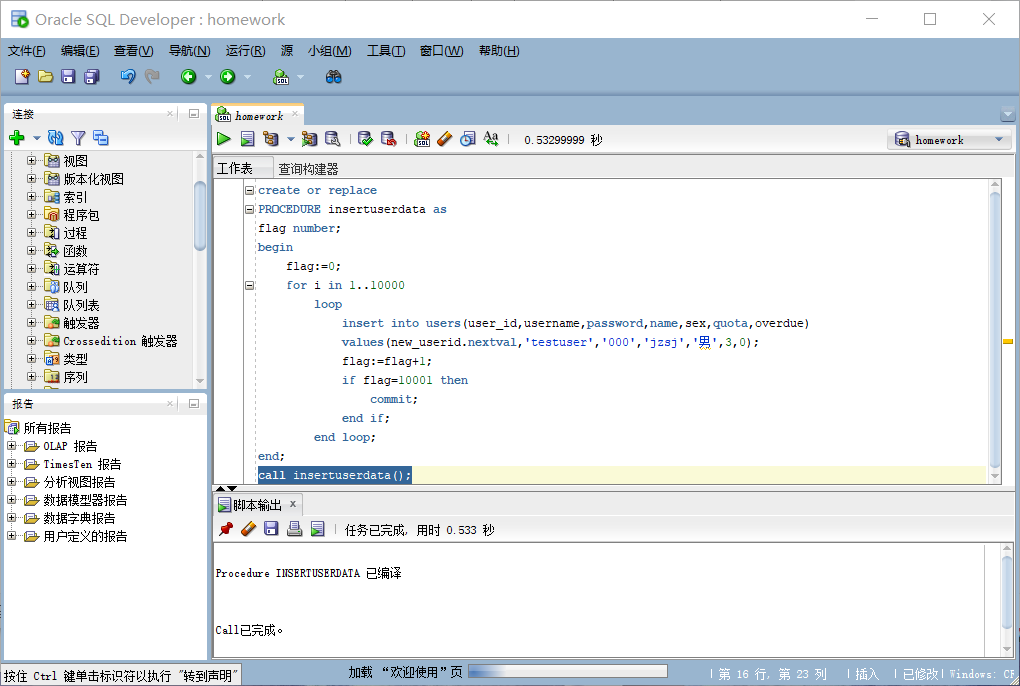
flag:=flag+1;

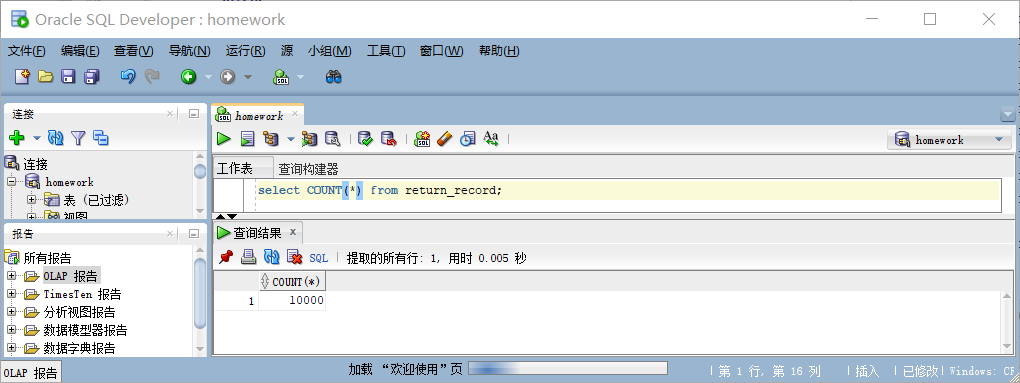
commit;

end loop;

end;

call insertuserdata();





admin表插入数据：

create or replace

PROCEDURE insertadmindata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into admin(admin\_id,username,password,name,sex,quota,overdue)

values(new\_adminid.nextval,'testadmin','001','zjl','女','sys');

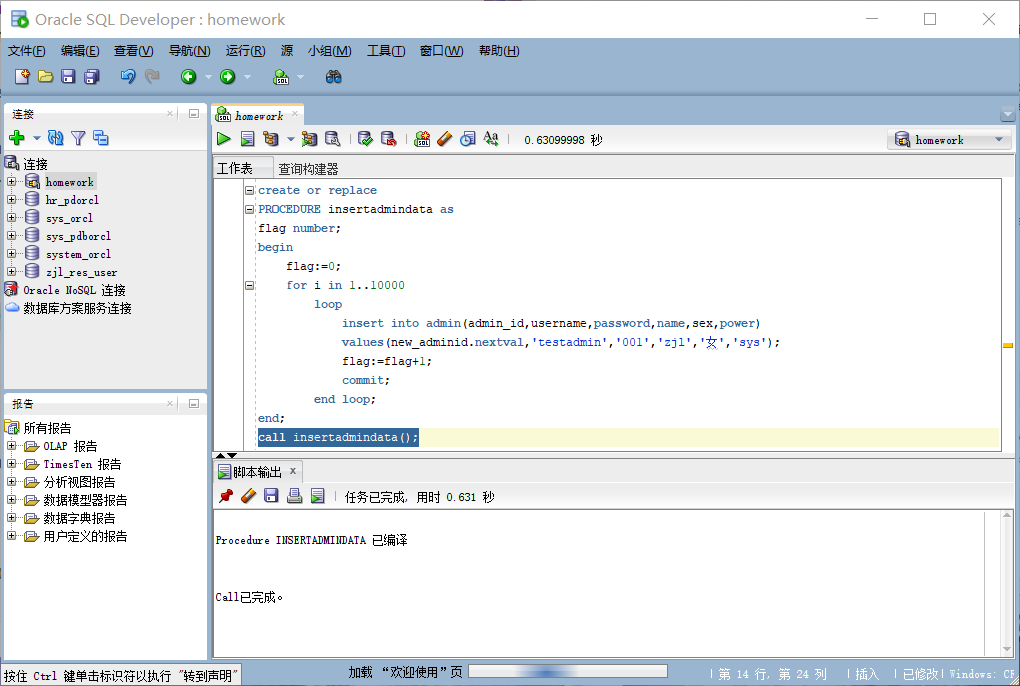
flag:=flag+1;

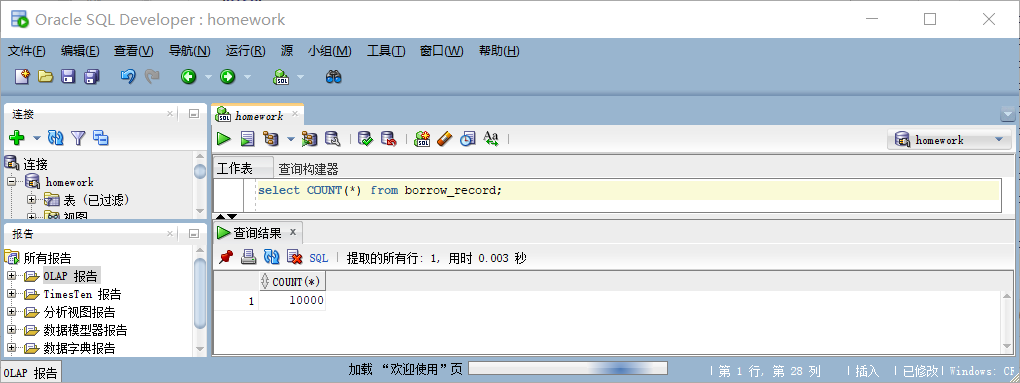
commit;

end loop;

end;

call insertadmindata();





book表插入数据：

create or replace

PROCEDURE insertbookdata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into book(book\_id,isbn,bookname,author,press,Inventory)

values(new\_bookid.nextval,to\_char(new\_ISBN.nextval),'红楼梦','曹雪芹','新华出版社','5');

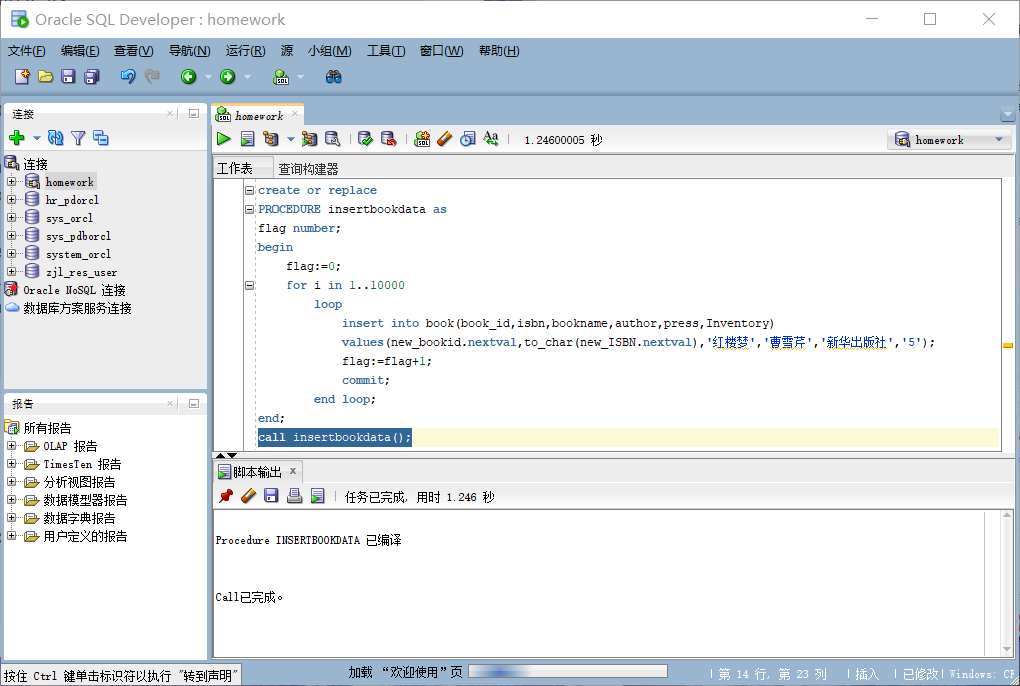
flag:=flag+1;

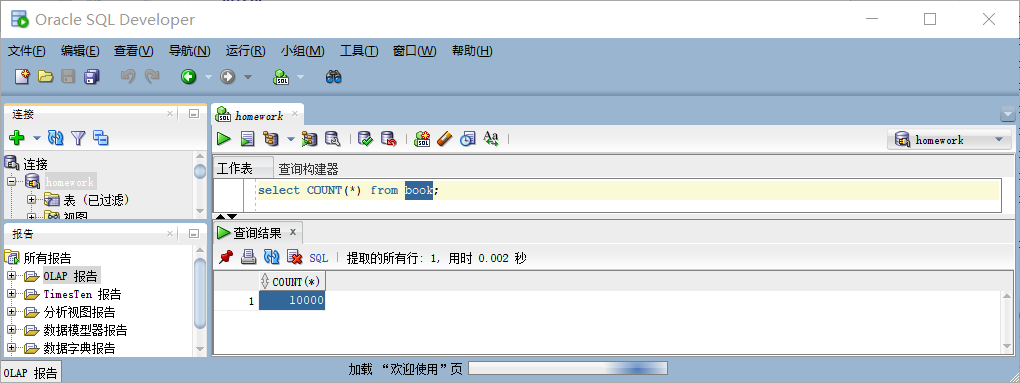
commit;

end loop;

end;

call insertbookdata();





borrow\_record表插入数据：

create or replace

PROCEDURE insertborrowbookdata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into borrow\_record(borrow\_id,user\_id,isbn,borrow\_date,term)

values(new\_borrow\_id.nextval,getuserid(),getisbn(),SYSDATE(),5);

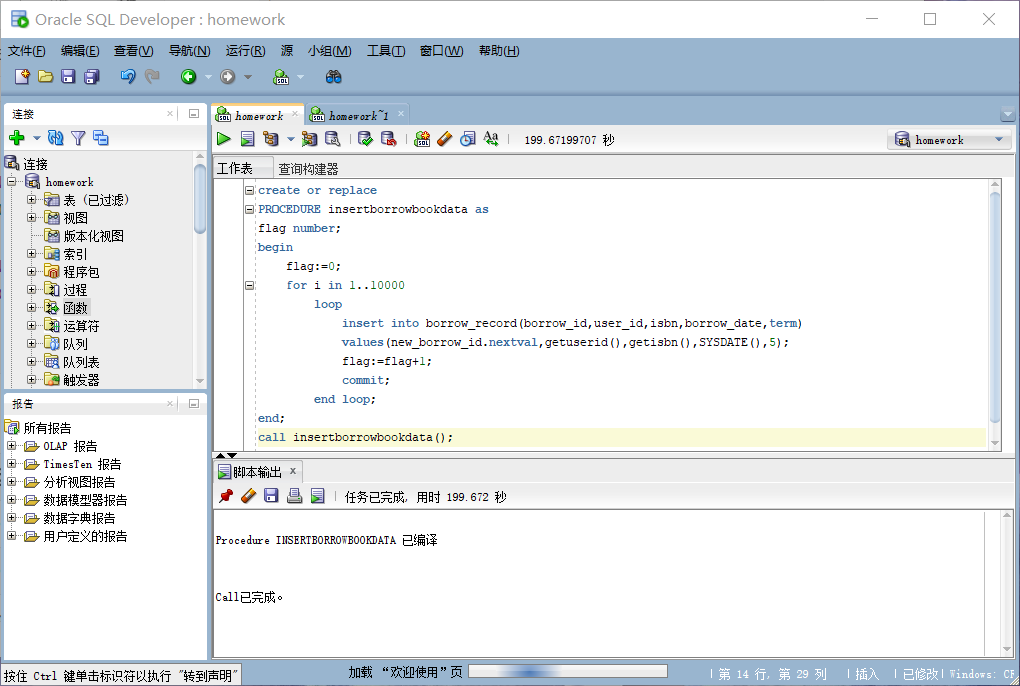
flag:=flag+1;

commit;

end loop;

end;

call insertborrowbookdata();





return\_record表插入数据：

create or replace

PROCEDURE insertreturnbookdata as

flag number;

begin

flag:=0;

for i in 1..10000

loop

insert into return\_record(return\_id,borrow\_id,return\_date,isoverdue)

values(new\_return\_id.nextval,getborrowid(),SYSDATE(),'false');

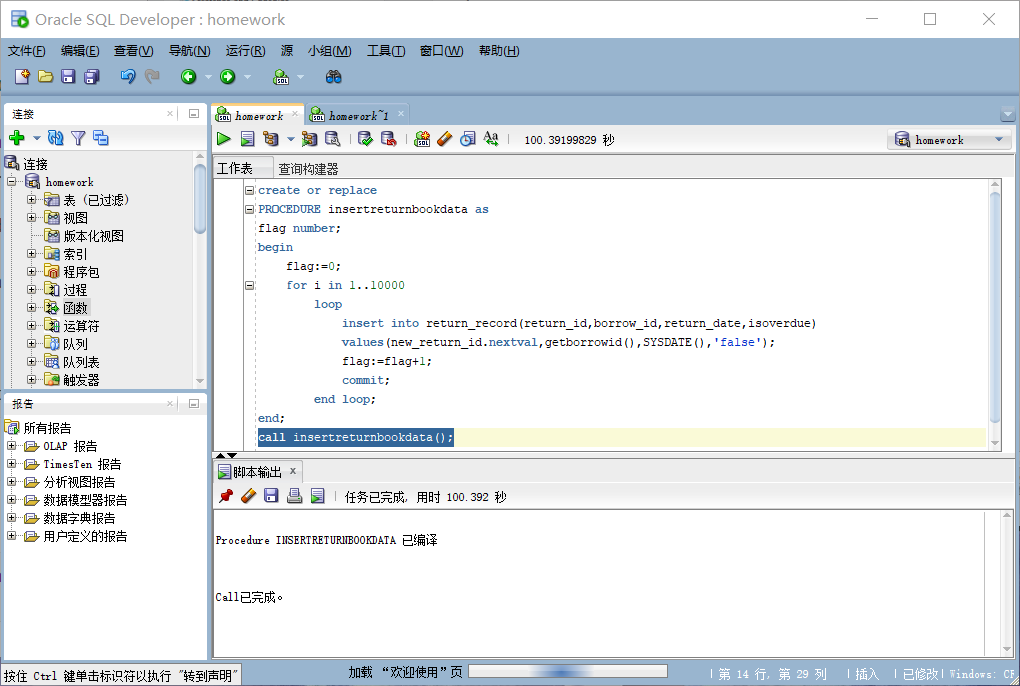
flag:=flag+1;

commit;

end loop;

end;

call insertreturnbookdata();





1. 函数设计

随机获得用户id：

create or replace function getUserid

return number

is

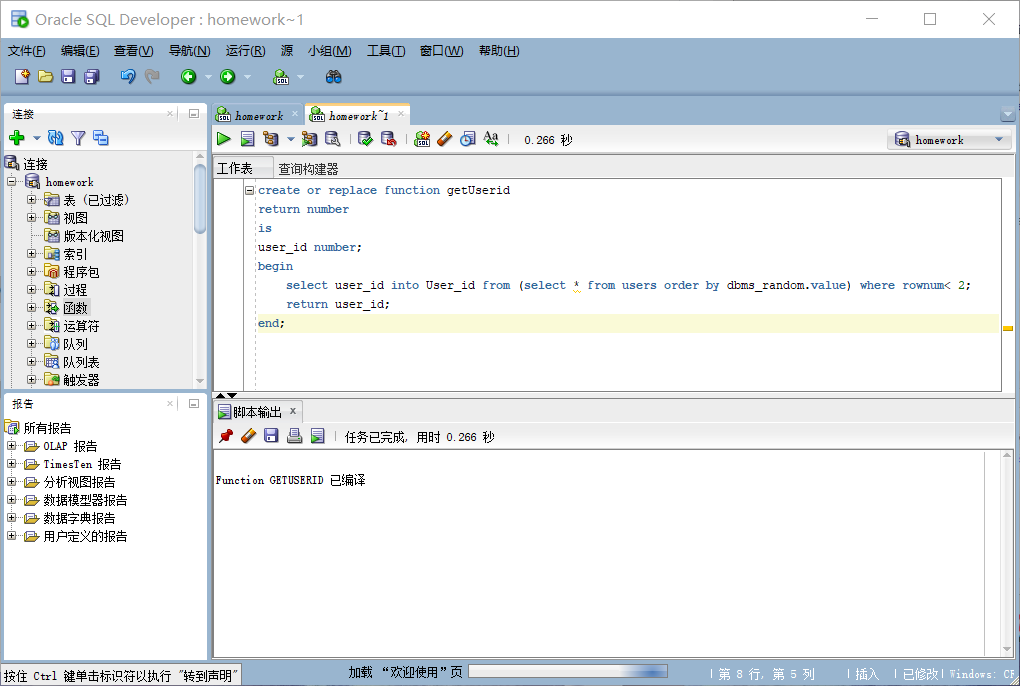
user\_id number;

begin

select user\_id into User\_id from (select \* from users order by dbms\_random.value) where rownum< 2;

return user\_id;

end;



随机获得ISBN：

create or replace function getIsbn

return varchar2

as

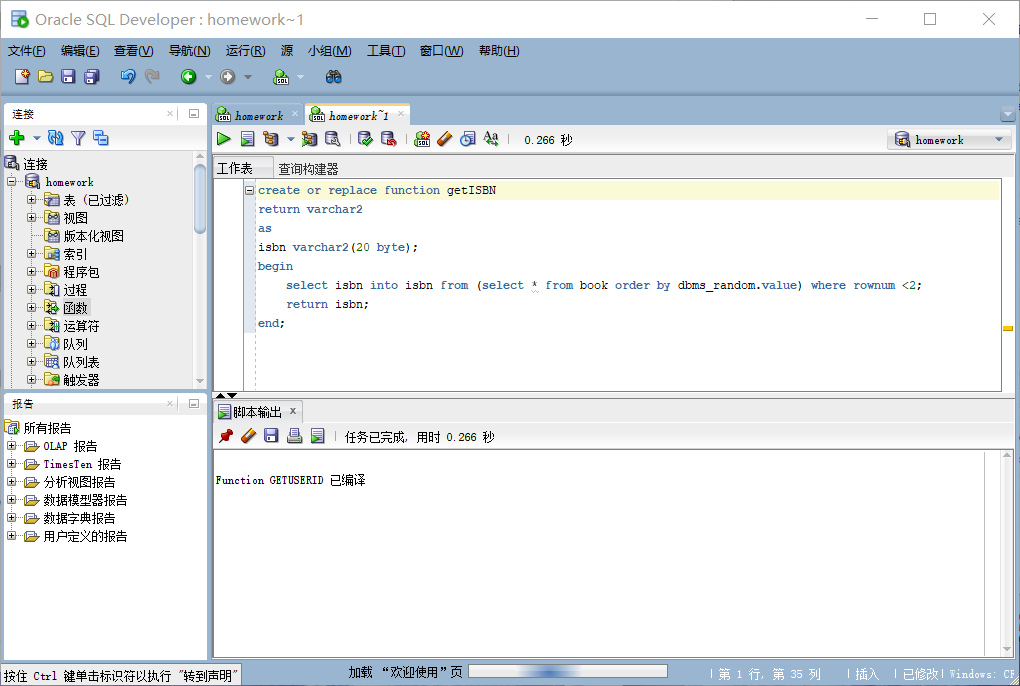
isbn varchar2(20 byte);

begin

select isbn into isbn from (select \* from book order by dbms\_random.value) where rownum <2;

return isbn;

end;



随机获得一个借书id：

create or replace function getborrowid

return number

as

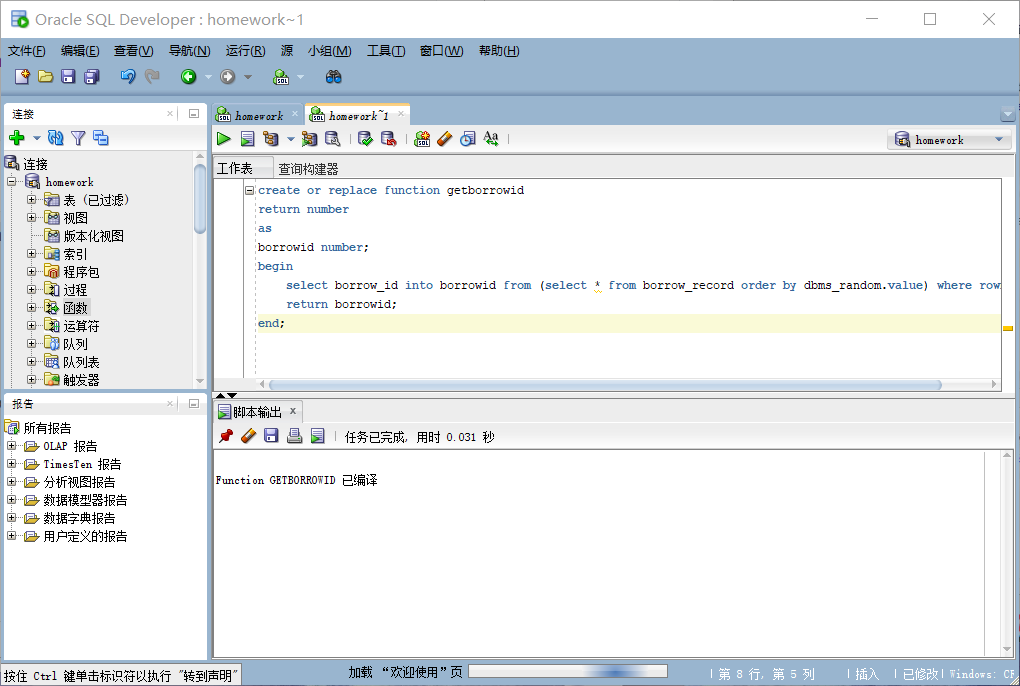
borrow\_id number;

begin

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

return borrow\_id;

end;



1. 备份

数据库备份与恢复：

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：

优点：

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。

不足：

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：

优点：

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。

不足：

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。

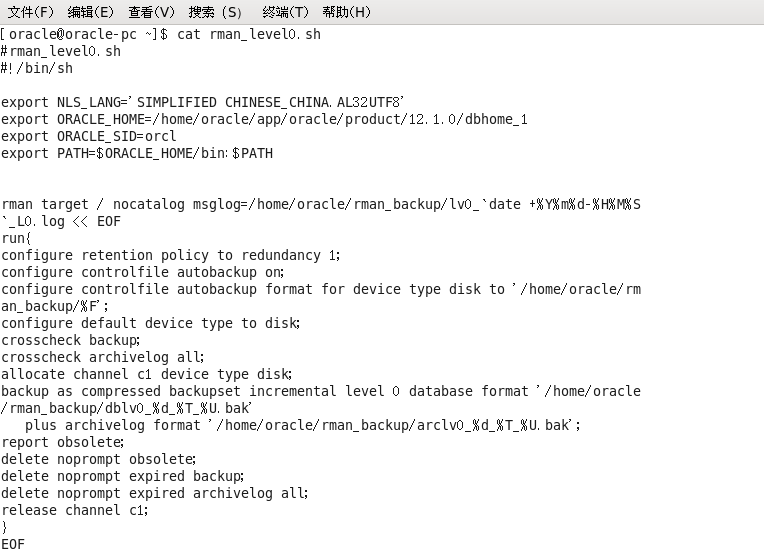
4．不能按表或按用户恢复。

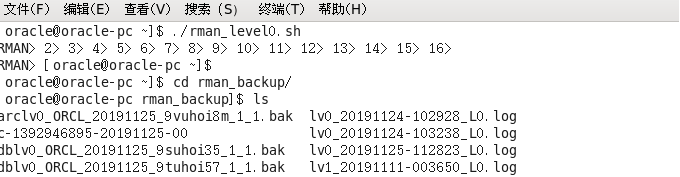
全备份：

通过rman\_level0.sh和rman\_level1.sh脚本对数据库进行全备份和全恢复，在数据库出现异常时候，不损失任何数据！

[oracle@oracle-pc ~]$ cat rman\_level0.sh

[oracle@oracle-pc ~]$ ./rman\_level0.sh





1. 容灾方案

Dataguard是ORACLE 提供的一种高可用性(HIGH AVAILABLE)的数据库方案，它是在主节点与备用节点间通过日志同步来保证数据的同步，可以实现快速切换与灾难性恢复。中软公司自主研发的基于 Dataguard同步引擎的Oracle数据库异地同步解决方案RS5，能够对安全、高效的实现数据库远程实时备份，最大限度保证用户的数据安全。

Data Guard作为Oracle免费推出的一个灾难恢复技术，利用Oracle 12g Data Guard而实现备用数据库操作的好处可以简单地总结为:它提供了灾难保护并防止数据丢失、维护主数据库的几个事务一致的副本、 防止灾难、数据损坏和用户错误、无需昂贵且复杂的HW/SW镜像。但是Data Guard也有以下缺点:不支持异构、不可跨平台、数据库版本需一致， 目标数据库出入rcovery状态不可用等。

步骤：

1. 将主数据库设置成归档状态和自动归档模式
2. 在主数据库上创建备用数据库控制文件、初始化参数文件:
3. 把将主数据库备份文件、控制文件和初始化参数文件复制到备用数据库上;
4. 设置 备用数据库控制文件、初始化参数文件和备用数据库侦听;
5. 设置备用数据库连接主数据库的服务名与主数据库到备用数据库的服务名;
6. 启动备用数据库实例，初始化日志应用服务并且将备用数据库设置成standby;
7. 启动归档到备用数据库。