

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的游戏账号的数据库设计 | | |
| 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 | 李东声 | 学号 | 201810414116 |
| 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** | | |  |

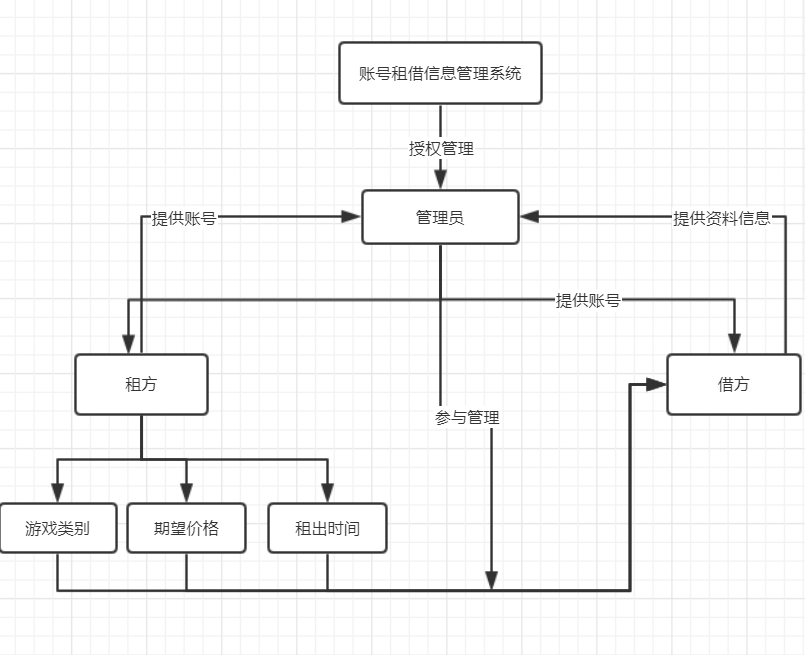
2021 年 6 月 1 日

## 一、游戏账号交易系统概述

随着Internet的发展和电子商务的日益普及，随着互联网游戏行业的发展，游戏交易需求的改变，租号号在传统游戏租号的基础上进行分类运营。在为买家提供账号租赁的同时，也为卖家闲置账号带来增值收益.这使得人们的网络资产,在闲余时间,可以给自己带来一定的收益,充分发挥剩余价值,让人们又更多的方式去获得一定的收入.

## 功能需求分析

### 1.功能需求模块和联系



### 功能描述

此游戏账号租借信息管理系统适对于如何玩家和号主都是很有用处的一个管理系统.它将号主和租号者通过管理员和系统联系在一起,实现了游戏账号的价值利用,也方便了租号者来享受游戏的乐趣,是现代方便娱乐,减少娱乐成本的重要途经.

**2.1、管理员信息管理**：

管理员是本系统的管理者，具有最高权限，对游戏和租借者添加，修改和删除等功能。但管理员只能修改自己的信息，不能添加或删除自己的信息。要想添加和删除管理员必须到数据库中去操作表结构。

**2.2、租借者信息管理**：

由系统管理员，给每个租借创建一个账户，账户应具有租借者的详细信息，并且借号者的信.

**2.3、号库管理：**

1）新增账号。管理员对新增的入库新账号进行分类,并且参考借号者提供的参考价格和实际市场价格来指定价格,并且订好价格后对号主进行通知和交流.

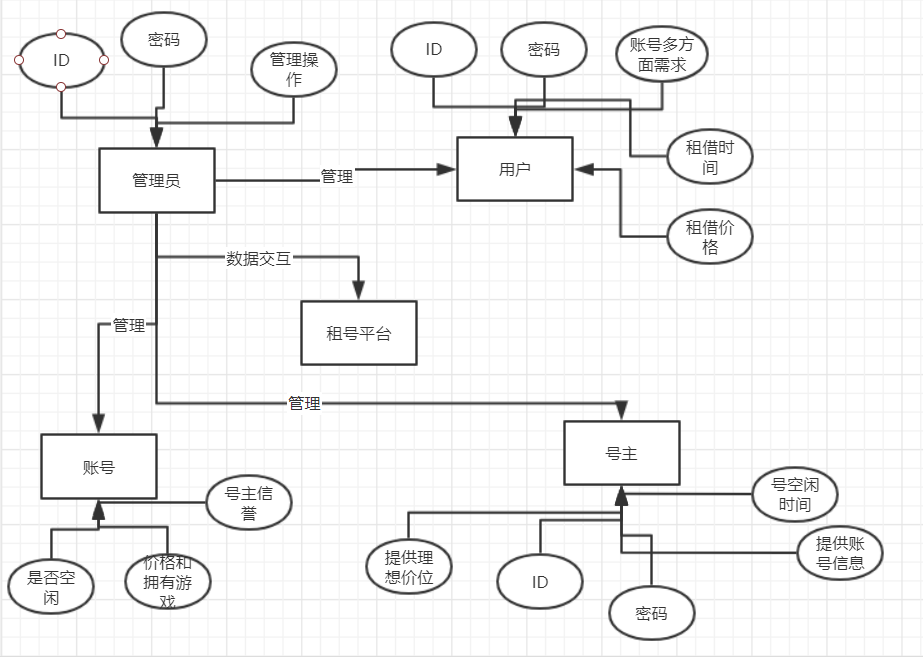
2.4、账号出租管理：

租借者用帐号进行认证后，可以在系统里查看游戏信息和自己的租借情况，可以进行账号的租借，在选完账号后系统会对这个账号的租借信息进行修改，将账号的租借信息改为出库，并反应在租借界面上。

**2.5、游戏归还管理：**

管理员可以根据租借的关键信息，查询到该租借者所有的租借信息，租借日期等，从中选择要归还的账号，显示有关该账号的所有租借信息，方便让管理员判断出是否过期，是否对租借者进行罚款，退还押金后并对数据库记录做出改变，，并把该账号状态信息改为在库。并且如果号主方强行违法条例,可以处罚号主方.

## 三 概念结构设计（E-R图）



上图是该游戏租界信息管理系统总E-R图，该系统涉及的实体集有：

账号实体集：是否空闲,拥有游戏,号主信誉

号主实体:ID,密码,理想价位,空闲时间,提供账号具体信息

管理员实体:ID,密码管理操作

用户实体:ID,密码,对账号的多方面需求,租借时间,租借价格.

## 四 系统结构设计

#### 3.1 管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为主键 | 约束 | 备注 |
| ID | int |  | 是 | NOT NULL | 自增 |
| password | varchar | 20 | 否 | NOT NULL | 管理员登录密码 |
| admin | varchar | 20 | 否 | NOT NULL |  |

#### 3.2用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为主键 | 约束 | 备注 |
| ID | int |  | 是 | NOT NULL | 自增 |
| password | varchar | 20 | 否 | NOT NULL | 用户登录密码 |
| username | varchar | 50 | 否 | NOT NULL | 用户名 |
| phone | varchar | 20 | 否 | NOT NULL | 用户手机号码 |
| Meassage | varchar | 30 | 否 | NOT NULL | 用户提供需求信息 |
| RegistrationDate | Date |  | 否 | NOT NULL |  |
| Use\_time | DateTime |  | 否 | NOT NULL | 租借时间 |

#### 3.3账号表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为主键 | 约束 | 备注 |
| ID | int |  | 是 | NOT NULL | 账号 |
| PID | int |  | 否 | NOT NULL | 流水号 |
| USER\_ID | varchar | 20 | 否 | NOT NULL | 用户id |
| price | int |  | 否 | NOT NULL | 单位时间价格 |
| Describe | varchar | 50 | 否 | NOT NULL | 账号描述 |
| Num | Date |  | 否 | NOT NULL |  |
| Admin\_ID | int |  | 否 | NOT NULL |  |

#### 3.4号主表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为主键 | 约束 | 备注 |
| ID | int |  | 是 | NOT NULL | 号主ID |
| password | varchar | 20 | 否 | NOT NULL | 号主登录密码 |
| username | varchar | 50 | 否 | NOT NULL | 号主用户名 |
| phone | varchar | 20 | 否 | NOT NULL | 号主手机号码 |
| Meassage | varchar | 30 | 否 | NOT NULL | 号主提供需求信息 |
| RegistrationDate | Date |  | 否 | NOT NULL |  |
| Use\_time | DateTime |  | 否 | NOT NULL | 号主租借时间 |
| Price | Int |  | 否 | NOT NULL | 号主提供的价位 |

## 五 表空间设计

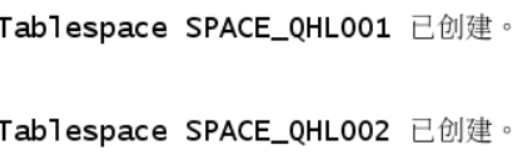
/\*第1步：创建表空间 \*\*/

#### space\_qhl001

Create Tablespace space\_qhl001  
datafile  
'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_qhl001\_1.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_qhl001\_2.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

#### space\_qhl002

Create Tablespace space\_qhl002  
datafile  
'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_qhl002\_1.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_qhl002\_2.dbf'  
 SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



/\*

创建用户gameleaseholder, gamemanger

\*/

create role borrow;

create role sale;

/\* 创建两个角色\*/

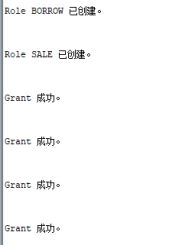
grant borrow to gameleaseholder;

grant sale to gamemanger;

/\*第4步：给用户授予权限 \*/

grant create session, create table to gameleaseholder;

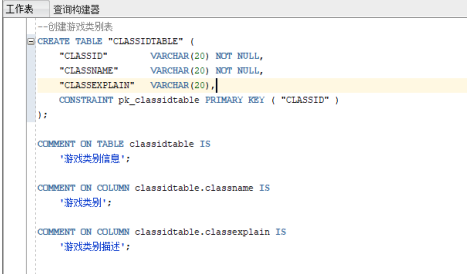
grant connect,resource to gamemanger;



#### 创建管理员表

* id 为主键

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\_QHL001  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
LOGGING  
TABLESPACE SPACE\_QHL001  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NO INMEMORY  
NOPARALLEL;



#### 创建用户表

* id 为主键
* 根据注册日期按范围分区
* 分为 2018 和 2019 年两个分区，每年按季度划 4 个子分区

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\_QHL001  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
TABLESPACE SPACE\_QHL001  
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\_QHL001  
 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\_QHL001  
 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  
 )  
);

#### 创建商品表

CREATE TABLE COMMODITY  
(  
 ID NUMBER(\*, 0) NOT NULL  
, PID NUMBER(\*, 0) NOT NULL  
, BOOKSNAME VARCHAR2(20 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\_QHL001  
 PCTFREE 10  
 INITRANS 2  
 STORAGE  
 (  
 BUFFER\_POOL DEFAULT  
 )  
 NOPARALLEL  
 )  
 ENABLE  
)  
LOGGING  
TABLESPACE SPACE\_QHL001  
PCTFREE 10  
INITRANS 1  
STORAGE  
(  
 BUFFER\_POOL DEFAULT  
)  
NOCOMPRESS  
NO INMEMORY  
NOPARALLEL;  
  
--创建管理员表

CREATE TABLE "MANGER" (

"ID" VARCHAR(20) NOT NULL,

"NAME" VARCHAR(20) NOT NULL,

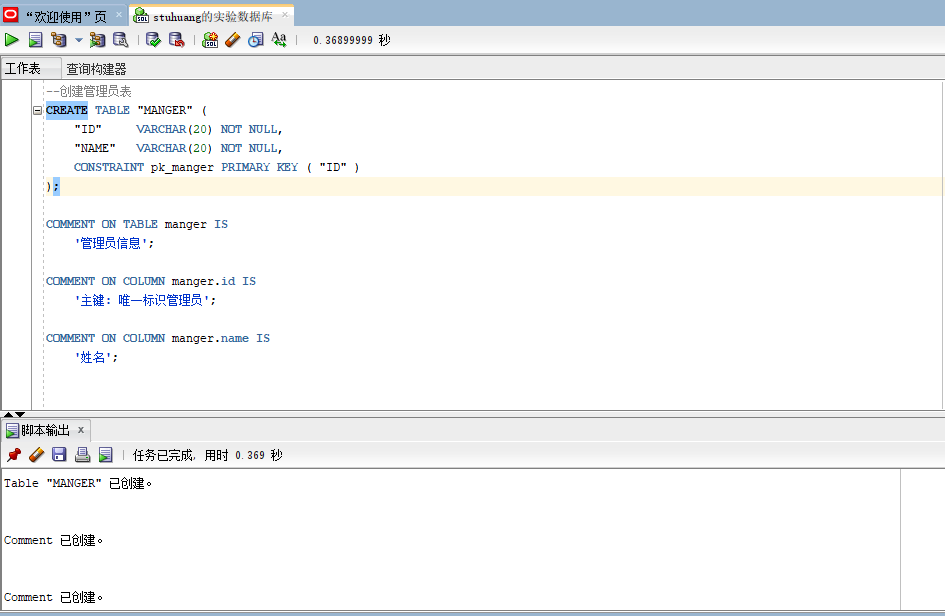
CONSTRAINT pk\_MANGER PRIMARY KEY ( "ID" )

);

comment on table Manger is '管理员信息';

comment on column Manger.ID is '主键: 唯一标识管理员';

comment on Column Manger.name is '姓名';



--创建租借者表

CREATE TABLE "LEASEHOLDER" (

"LEASEHOLDERID" VARCHAR(20) NOT NULL,

"LEASEHOLDERCLASS" VARCHAR(20) NOT NULL,

"LEASEHOLDERNAME" VARCHAR(20),

"TIME" VARCHAR(20),

CONSTRAINT pk\_LEASEHOLDER PRIMARY KEY ("LEASEHOLDERID")

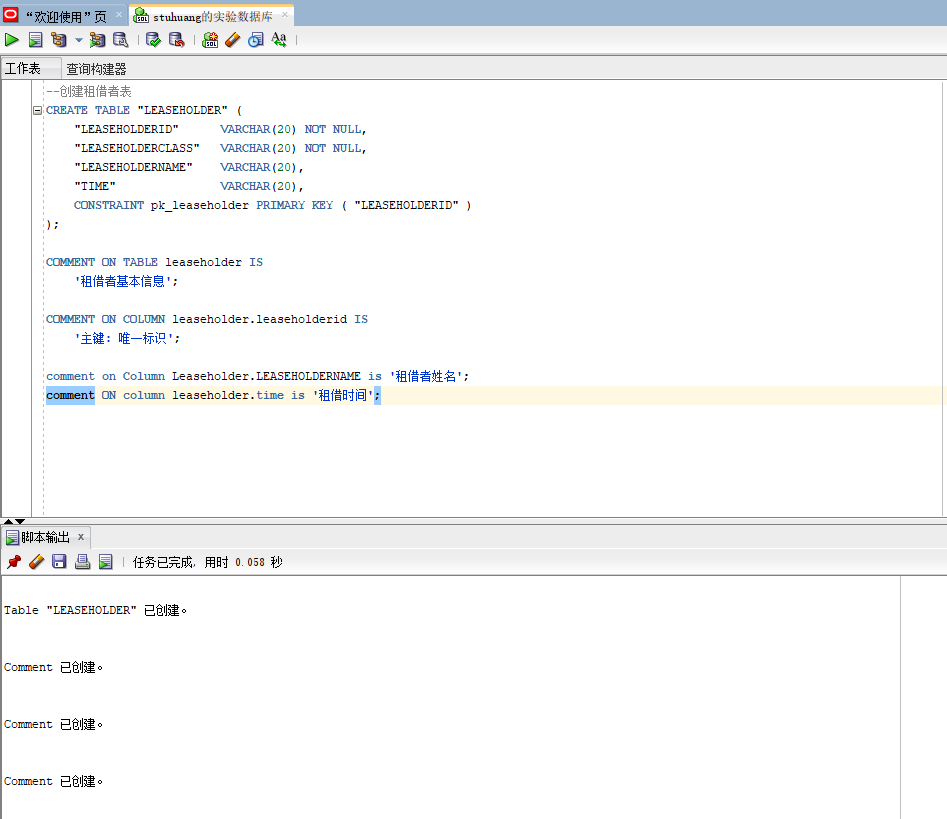
);

comment on table Leaseholder is '租借者基本信息';

comment on column Leaseholder.LEASEHOLDERID is '主键: 唯一标识';

commnent on Column Leaseholder.LEASEHOLDERNAME is '租借者姓名';

commnent on Column Leaseholder.TIME is '租借时间';



### /\*

### 插入用户、账号、号主数据

\*/

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 := '李东声|| 'aa';  
 username := '李东声' || i;  
 phone := '131785693' || i;  
 booksname := 'LOL账号' || i;  
 address :='成都'|| '四川';  
 price :=(dbms\_random.value() \* 100);  
 num :=(i mod 5);  
 insert /\*+append\*/ into user (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,name,price,describe,num,admin\_id)  
 values (id,id,booksname,price,'good',num,1);  
 --插入账号选择

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;

/\*查看数据\*/

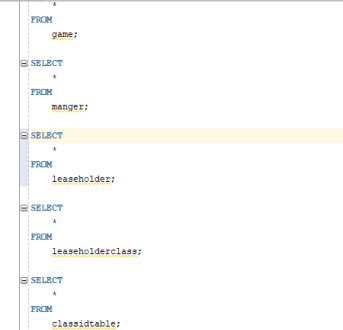
select \* from GAME;

select \* from MANGER

select \* from LEASEHOLDER;

select \* from LEASEHOLDERCLASS;

select \* from CLASSIDTABLE;



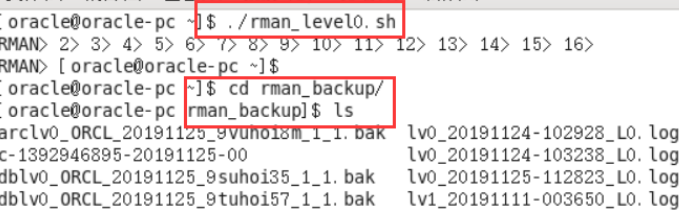
#### 使用存储过程 adduser 插入用户数据

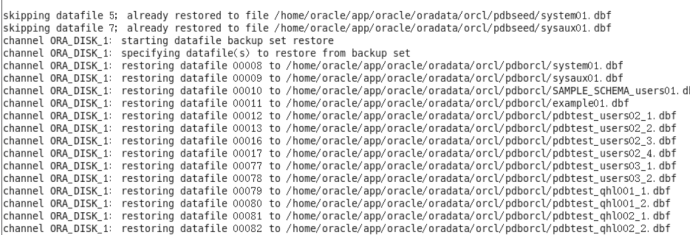
set serveroutput on  
declare  
begin  
BOOK\_PACKAGE.addUser('131','wwj','123456','hongkong','2021-06-12');  
end;

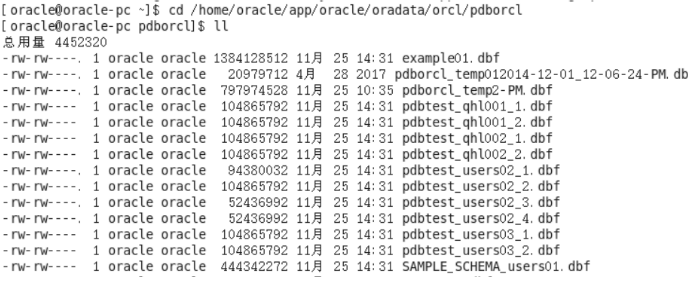


#### 表空间使用状况

SELECT a.tablespace\_name "表空间名",  
total "表空间大小",  
free "表空间剩余大小",  
(total - free) "表空间使用大小",  
total / (1024 \* 1024 \* 1024) "表空间大小(G)",  
free / (1024 \* 1024 \* 1024) "表空间剩余大小(G)",  
(total - free) / (1024 \* 1024 \* 1024) "表空间使用大小(G)",  
round((total - free) / total, 4) \* 100 "使用率 %"  
FROM (SELECT tablespace\_name, SUM(bytes) free  
FROM dba\_free\_space  
GROUP BY tablespace\_name) a,  
(SELECT tablespace\_name, SUM(bytes) total  
FROM dba\_data\_files

/\*备份\*/  






## 六 总结

这学期的Oracle数据库课程比较的快,其中又很多的东西不是太懂,只能跟着老师给的操作步骤来一步步的进行实验和作业.有些东西还是问了很多老师和身边的同学才大概知道操作流程和使用的办法.不过Oracle这项课程老师的热心讲解和上课时候的细致操作也让我收获到了很多.这次期末做的是关于游戏账号的租借系统,找个是因为现代社会中,青年少年对于游戏的看法发生了改变,它成为了一种消遣,乃至为赚钱的工具,而账号的租借系统正好可以将自己手中闲置下来的账号发挥它的剩余价值

这次实验中,也学习了如何自己去建立一个Oracle数据库,并且将它完善,各种功能和角色相互的互动和完善.自己也流程走下来也学会了功能分析和系统设计等.其中也碰了不少的麻烦,比如删除部分数据库内容时候发生错误等等.总之Oralce这门课程让我收获很多.