

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的游戏客户端的数据库设计 | | |
| 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 | 王炜 | 学号 | 201810414320 |
| 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

# Oracle期末项目

(基于Oracle的Zero-征途游戏客户端系统数据库设计)

## 期末考核要求

* 自行设计一个信息系统的数据库项目，自拟系统名称。
* 设计项目涉及的表及表空间使用方案。至少5张表和5万条数据，两个表空间。
* 设计权限及用户分配方案。至少两类角色，两个用户。
* 在数据库中建立一个程序包，在包中用PL/SQL语言设计一些存储过程和函数，实现比较复杂的业务逻辑，用模拟数据进行执行计划分析。
* 设计自动备份方案或则手工备份方案。

## 1.项目简介

项目名称：Zero-征途游戏客户端系统

本项目是基于Oracle的Zero-征途游戏客户端系统数据库设计。

涉及角色/用户：

涉及表：User表、Hero表、Attribute表、Shop表、Game\_Order表、Warehose表

## 2. 数据表说明

### 2.1 User表：存储用户数据

* user\_id：INT 用户id
* name：VARCHAR 用户名
* password：VARCHAR 密码
* gold：INT 金币

### 2. 2 Hero表：游戏英雄表

* hero\_Id：INT 英雄id
* attribute\_Id：INT 属性id
* name：VARCHAR 用户名

### 2.3 Attribute表：英雄属性表

* attribute\_Id：INT 属性id
* hp：INT 生命值
* magic：INT 魔力值
* attack：INT 攻击力
* defense：INT 防御力
* speed：INT 速度

### 2.4 Shop表：商店表

* hero\_id：INT 英雄id
* price：INT 售卖价格
* introduce：VARCHAR 介绍

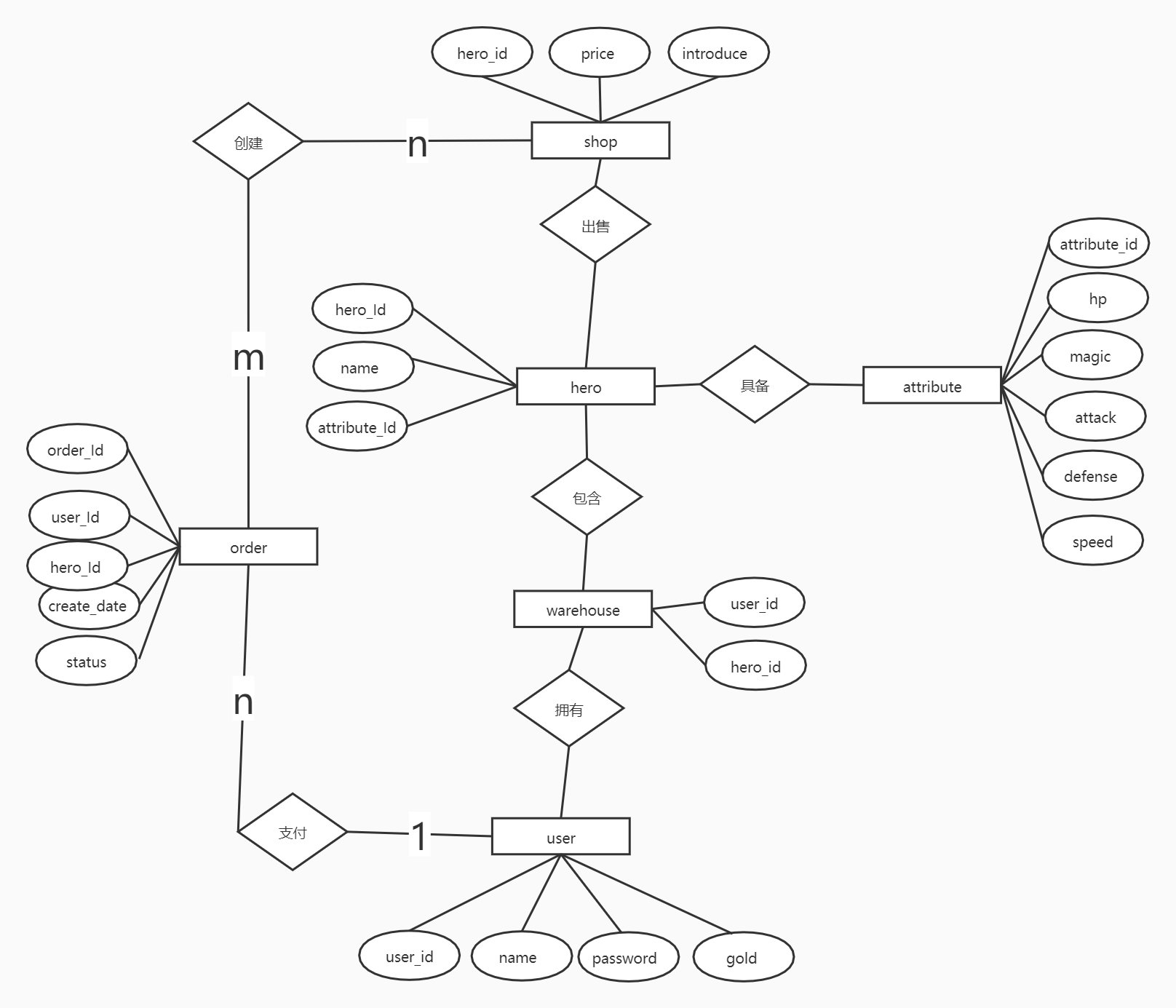
### 2.5 Game\_Order表：订单表

* order\_Id：INT 订单id
* user\_id：INT 用户id
* hero\_Id：INT 英雄id
* create\_date：DataTime 创建时间
* status：VARCHAR 订单状态

### 2.6 Warehose表：用户拥有的英雄仓库表

* user\_id：INT 用户id
* hero\_Id：INT 英雄id

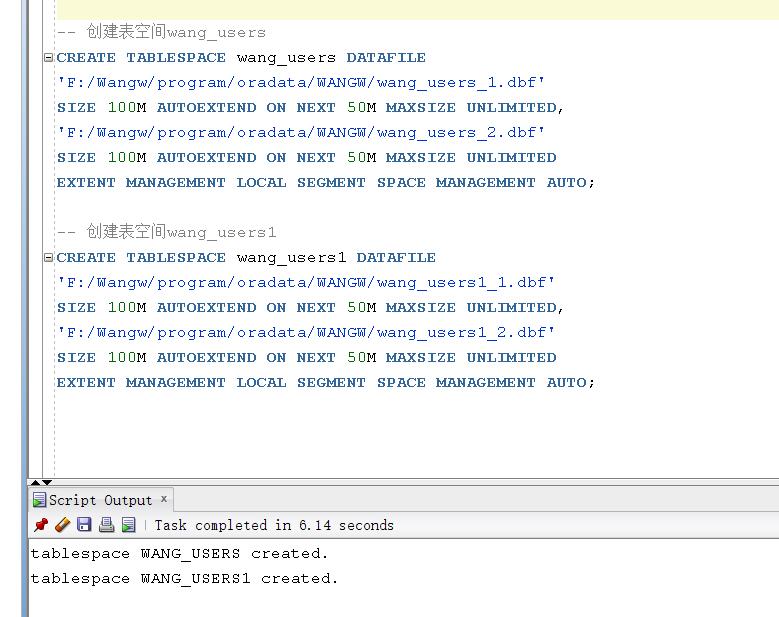
## 3. 数据库逻辑结构



## 4 设计简介

### 4.1 创建表空间

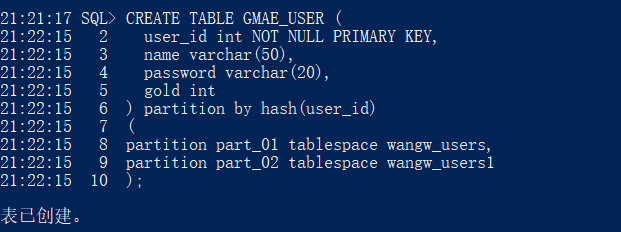
-- 创建表空间wang\_users  
CREATE TABLESPACE wang\_users DATAFILE  
'F:/Wangw/program/oradata/WANGW/wang\_users\_1.dbf'  
SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED,  
'F:/Wangw/program/oradata/WANGW/wang\_users\_2.dbf'  
SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;  
  
-- 创建表空间wang\_users1  
CREATE TABLESPACE wang\_users1 DATAFILE  
'F:/Wangw/program/oradata/WANGW/wang\_users1\_1.dbf'  
SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED,  
'F:/Wangw/program/oradata/WANGW/wang\_users1\_2.dbf'  
SIZE 100M AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



### 4.2 创建数据表

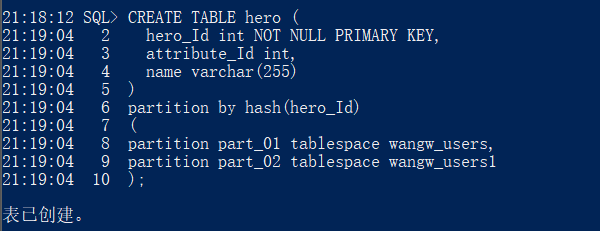
#### 4.2.1 User表

CREATE TABLE GMAE\_USER (  
user\_id int NOT NULL PRIMARY KEY,  
name varchar(50),  
password varchar(20),  
gold int)   
partition by hash(user\_id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



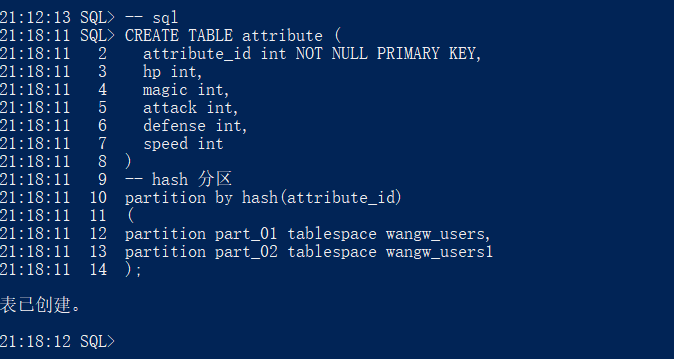
#### 4.2. 2 Hero表

CREATE TABLE hero (  
hero\_Id int NOT NULL PRIMARY KEY,  
attribute\_Id int,  
name varchar(255))  
partition by hash(hero\_Id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



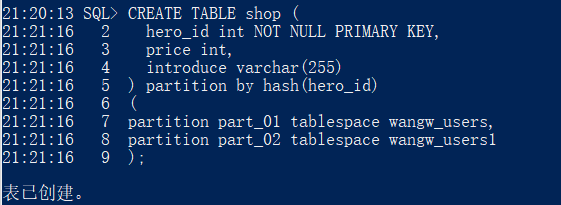
#### 4.2.3 Attribute表

CREATE TABLE attribute (  
attribute\_id int NOT NULL PRIMARY KEY,  
hp int,  
magic int,  
attack int,  
defense int,  
speed int)  
-- hash 分区  
partition by hash(attribute\_id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



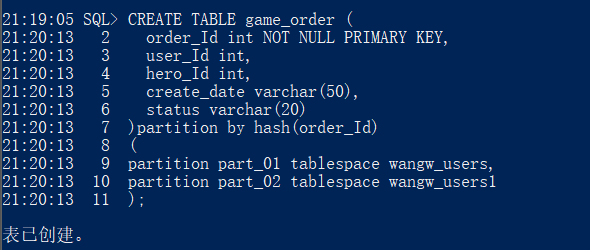
#### 4.2.4 Shop表

CREATE TABLE shop (  
hero\_id int NOT NULL PRIMARY KEY,  
price int,  
introduce varchar(255))   
partition by hash(hero\_id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



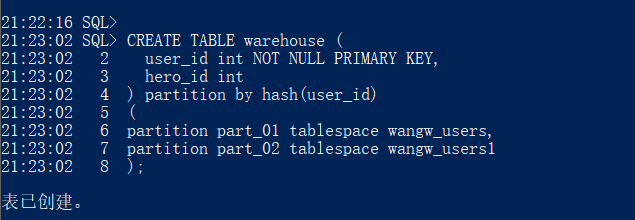
#### 4.2.5 Game\_Order表

CREATE TABLE game\_order (  
order\_Id int NOT NULL PRIMARY KEY,  
user\_Id int,  
hero\_Id int,  
create\_date varchar(50),  
status varchar(20))  
partition by hash(order\_Id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



#### 4.2.6 Warehose表

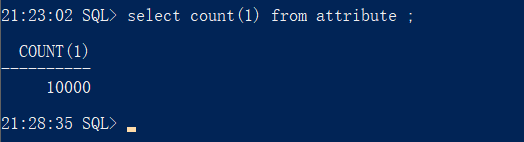
CREATE TABLE warehouse (  
user\_id int NOT NULL PRIMARY KEY,  
hero\_id int)   
partition by hash(user\_id)(  
partition part\_01 tablespace wangw\_users,  
partition part\_02 tablespace wangw\_users1);



## 4.3 向表中插入数据

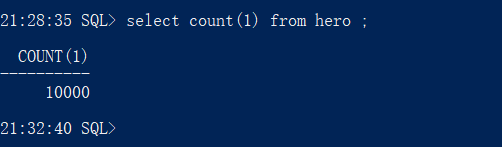
#### 4.3.1 Attribute表

CREATE OR REPLACE PROCEDURE ADD\_ATTRIBUTE AS   
BEGIN  
 for i in 1..10000  
 loop  
 insert into attribute (ATTRIBUTE\_ID,HP,MAGIC,ATTACK,DEFENSE,SPEED)   
 VALUES (i,100,100,15,25,25);  
 end loop;  
END ADD\_ATTRIBUTE;



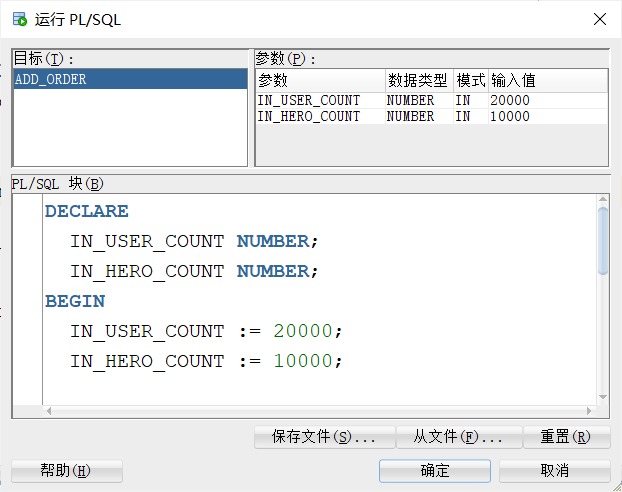
#### 4.3.2 Hero表

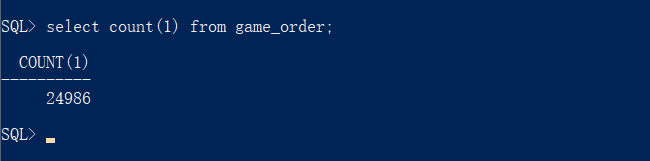
CREATE OR REPLACE PROCEDURE ADD\_HERO AS   
v\_name varchar(50);   
BEGIN   
 for i in 1..10000   
 loop   
 v\_name := '张三'||i||'号';   
 insert into hero (HERO\_ID,ATTRIBUTE\_ID,NAME) VALUES   
 (i,i,v\_name);   
 end loop;   
END ADD\_HERO;



#### 4.3.3 Game\_order表

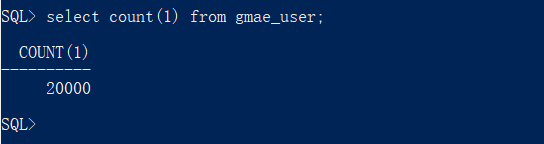
CREATE OR REPLACE PROCEDURE ADD\_ORDER(   
 -- 键入USER 数量   
 in\_user\_count in number,   
 -- 键入英雄 数量   
 in\_hero\_count in number   
) AS   
 -- 显示声明 hero 游标   
 cursor c\_hero is select HERO\_ID from HERO;   
 -- 定义hero\_id   
 v\_hero\_id HERO.HERO\_ID%type;   
 -- 随机订单数   
 v\_order\_random\_count number;   
 -- 随机用户ID   
 v\_user\_id game\_order.user\_id%type;   
 -- 日期   
 v\_date date;   
 -- 全局订单index;   
 v\_index game\_order.order\_id%type := 1;   
BEGIN   
 -- 打开游标   
 open c\_hero;   
 -- 遍历游标   
 loop   
 FETCH c\_hero into v\_hero\_id;   
 -- 退出条件   
 EXIT when c\_hero%NOTFOUND;   
 -- 执行操作   
 -- 改用户随机订单数   
 -- 1. 获取随机数   
 SELECT trunc(dbms\_random.value(1,5))into v\_order\_random\_count FROM DUAL;   
 for k in 1..v\_order\_random\_count   
 loop   
 -- 2.获得随机用户id   
 SELECT trunc(dbms\_random.value(1,in\_user\_count))into v\_user\_id FROM DUAL;   
 -- 4.获取随机日期   
 v\_date := TO\_DATE(sysdate);   
-- DBMS\_OUTPUT.PUT\_LINE(v\_hero\_id||'&'||v\_user\_id||'&'||v\_hero\_id||'&'||v\_date||'死');   
 insert into GAME\_ORDER (ORDER\_ID,USER\_ID,HERO\_ID,CREATE\_DATE,STATUS)   
 VALUES (v\_index,v\_user\_id,v\_hero\_id,v\_date,'死');   
 v\_index := v\_index+1;   
 end loop;   
 end loop;   
 -- 关闭游标   
 close c\_hero;   
END ADD\_ORDER;





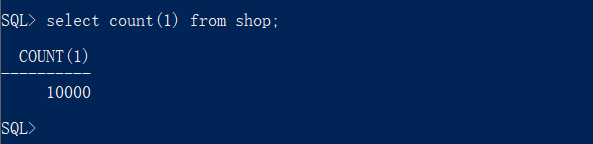
#### 4.3.4 User表

-- user 表   
CREATE OR REPLACE PROCEDURE ADD\_USER AS   
v\_name varchar(50);   
v\_password varchar(50);   
v\_gole number;   
BEGIN   
 -- 20000名召唤师   
 for i in 1..20000   
 loop   
 v\_name := '召唤师'||i||'号';   
 v\_password := 'password'||i||'.';   
 -- 1. 获取随机金币   
 SELECT trunc(dbms\_random.value(1,200000))into v\_gole FROM DUAL;   
 insert into gmae\_user (USER\_ID,NAME,PASSWORD,GOLD) VALUES (i,v\_name,v\_password,v\_gole);   
 end loop;  
END ADD\_USER;



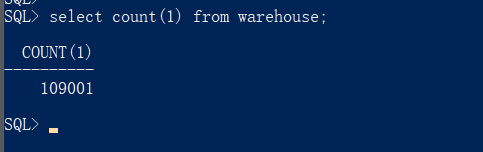
#### 4.3.5 Shop表

CREATE OR REPLACE PROCEDURE ADD\_SHOP AS   
 cursor c\_hero is select HERO\_ID from HERO;   
 v\_hero\_id HERO.HERO\_ID%type;   
 -- shop介绍   
 v\_shop\_introduce varchar(80);   
 v\_price shop.price%type;   
 v\_index number := 1;   
BEGIN   
 open c\_hero;   
 loop   
 fetch c\_hero into v\_hero\_id;   
 exit when c\_hero%NOTFOUND;   
 SELECT trunc(dbms\_random.value(1,3200))into v\_price FROM DUAL;   
 v\_shop\_introduce := '这是英雄'||v\_hero\_id||'号';  
 insert into shop (HERO\_ID,PRICE,INTRODUCE)   
 VALUES (v\_hero\_id,v\_price,v\_shop\_introduce);   
 end loop;   
 close c\_hero;   
END ADD\_SHOP;



#### 4.3.6 Warehose表

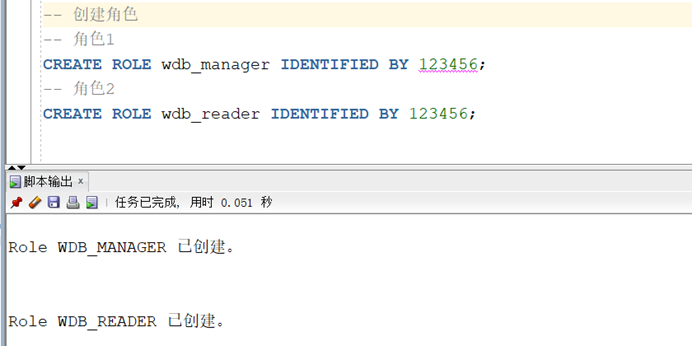
CREATE OR REPLACE PROCEDURE ADD\_WAREHOUSE(   
 in\_hero\_count in number   
) AS   
 cursor c\_user is select USER\_ID from GMAE\_USER;   
 v\_user\_id warehouse.user\_id%type;   
 v\_hero\_count number;   
 v\_hero\_id warehouse.hero\_id%type;   
BEGIN   
 DBMS\_OUTPUT.ENABLE(buffer\_size => null);   
 open c\_user;   
 loop   
 fetch c\_user into v\_user\_id;   
 exit when c\_user%NOTFOUND;   
 SELECT trunc(dbms\_random.value(0,10))into v\_hero\_count FROM DUAL;   
 DBMS\_OUTPUT.PUT\_LINE('该轮循环走'||v\_hero\_count||'次');   
 for i in 0..v\_hero\_count   
 loop   
 SELECT trunc(dbms\_random.value(0,in\_hero\_count))into v\_hero\_id FROM DUAL;   
 insert into WAREHOUSE(USER\_ID,HERO\_ID)   
 VALUES (v\_user\_id,v\_hero\_id);   
 DBMS\_OUTPUT.PUT\_LINE('插入 v\_user\_id = '||v\_user\_id||' v\_hero\_id = '|| v\_hero\_id);   
 end loop;   
 end loop;   
 close c\_user;   
END ADD\_WAREHOUSE;



## 5.设计权限及用户分配方案

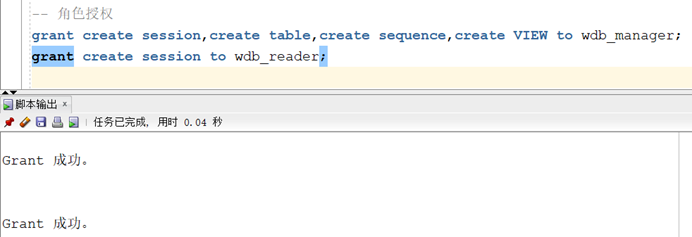
### 5.1 创建角色

-- 角色1  
CREATE ROLE wdb\_manager IDENTIFIED BY 123456;  
  
-- 角色2  
CREATE ROLE wdb\_reader IDENTIFIED BY 123456;

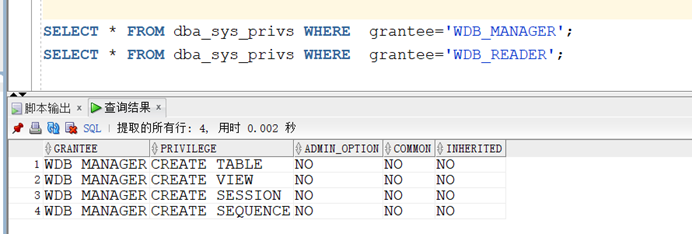


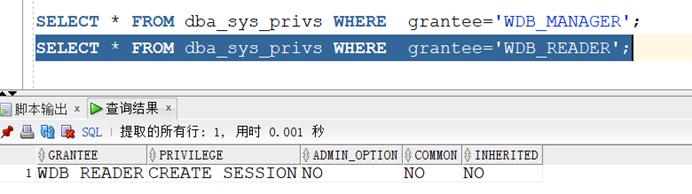
### 5.2 角色授权

-- 角色授权  
grant create session,create table,create sequence,create VIEW to wdb\_manager;  
grant create session to wdb\_reader;



### 5.3 查看结果

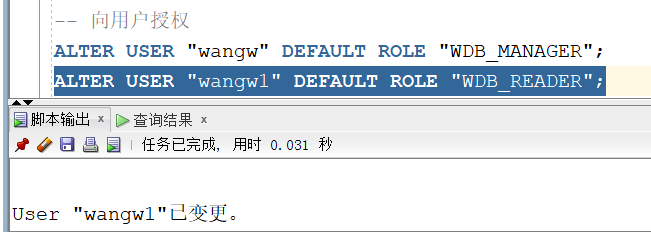


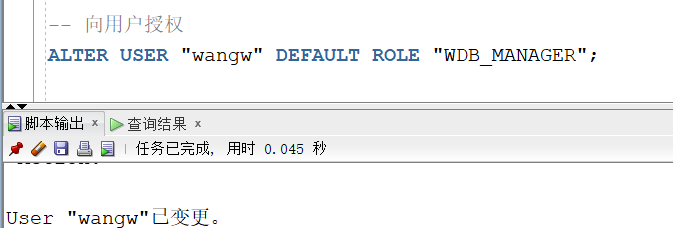


### 5.4 创建用户

-- USER SQL  
CREATE USER "wangw" IDENTIFIED BY "123456"  
DEFAULT TABLESPACE "WANGW\_USERS"  
TEMPORARY TABLESPACE "TEMP";  
CREATE USER "wangw1" IDENTIFIED BY "123456"  
DEFAULT TABLESPACE "WANGW\_USERS1"  
TEMPORARY TABLESPACE "TEMP";  
-- QUOTAS  
ALTER USER "wangw" QUOTA UNLIMITED ON "WANGW\_USERS";  
ALTER USER "wangw1" QUOTA UNLIMITED ON "WANGW\_USERS1";  
-- ROLES  
GRANT "CONNECT" TO "wangw" ;  
GRANT "CONNECT" TO "wangw1" ;

### 5.5 授权





## 6. 综合设计

**在数据库中建立一个程序包，在包中用PL/SQL语言设计一些存储过程和函数，实现比较复杂的业务逻辑，用模拟数据进行执行计划分析**

设计简介

包名：WW\_WORK

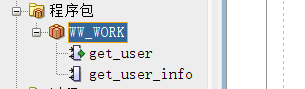
函数名：get\_user(user\_id number)

功能：输入用户id，根据用户仓库查询用户具有的英雄的属性

过程名：Get\_user\_info(user\_id number)

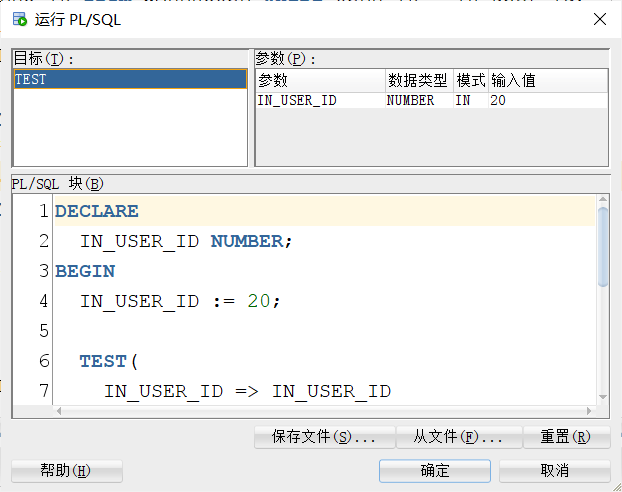
### 6.1 创建包

-- 创建包   
CREATE OR REPLACE  
PACKAGE WW\_WORK AS  
FUNCTION get\_user(order\_id\_t NUMBER) RETURN VARCHAR2;  
PROCEDURE get\_user\_info(train\_id\_t VARCHAR2);  
END WW\_WORK;

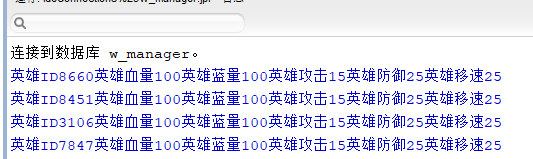


### 6.2 创建函数和过程

create or replace PACKAGE BODY WW\_WORK IS  
FUNCTION get\_user(user\_id NUMBER) RETURN VARCHAR2  
AS  
M VARCHAR2(100);  
BEGIN  
select \* into N from WAREHOUSE where USER\_ID=user\_id;  
RETURN N;  
END;  
PROCEDURE get\_user\_info(user\_id number)  
AS  
cursor c\_hero is select HERO\_ID from WAREHOUSE where USER\_ID = in\_user\_id;  
v\_HERO\_ID warehouse.hero\_id%type;  
v\_ATTRIBUTE\_ID hero.attribute\_id%type;  
v\_HP attribute.hp%type;  
v\_MAGIC attribute.magic%type;  
v\_ATTACK attribute.attack%type;  
v\_DEFENSE attribute.defense%type;  
v\_SPEED attribute.speed%type;  
BEGIN  
DBMS\_OUTPUT.ENABLE(buffer\_size => null);  
open c\_hero;  
loop  
fetch c\_hero into v\_HERO\_ID;  
exit when c\_hero%NOTFOUND;  
select ATTRIBUTE\_ID into v\_ATTRIBUTE\_ID from HERO where HERO\_ID = v\_HERO\_ID;  
select HP,MAGIC,ATTACK,DEFENSE,SPEED into  
v\_HP,v\_MAGIC,v\_ATTACK,v\_DEFENSE,v\_SPEED from ATTRIBUTE  
where ATTRIBUTE\_ID = v\_ATTRIBUTE\_ID;  
DBMS\_OUTPUT.PUT\_LINE('英雄ID'||v\_HERO\_ID||'英雄血量'||v\_HP||'英雄蓝量'||v\_MAGIC||  
'英雄攻击'||v\_ATTACK||'英雄防御'||v\_DEFENSE||'英雄移速'||v\_SPEED);  
end loop;  
close c\_hero;  
end;  
END WW\_WORK;



### 6.3 测试结果



## 7.设计手动备份方案

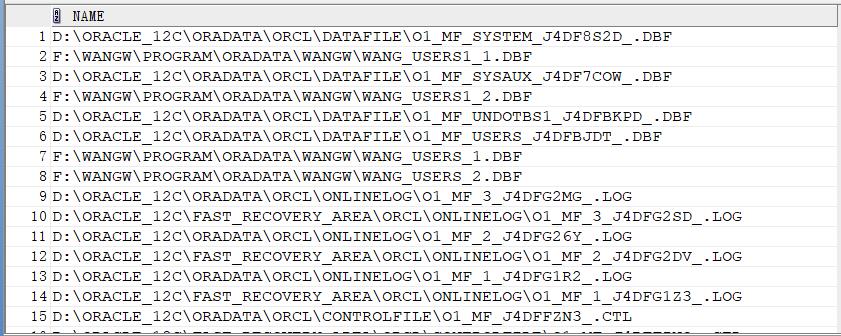
**手动备份方案——脱机备份**

**自动备份方案——用户管理备份**

### 7.1 手动备份

1. 查询所有的数据文件、控制文件和联机重做日志文件

SELECT NAME FROM v$datafile  
 UNION ALL  
 SELECT MEMBER as NAME FROM v$logfile  
 UNION ALL  
SELECT NAME FROM v$controlfile;

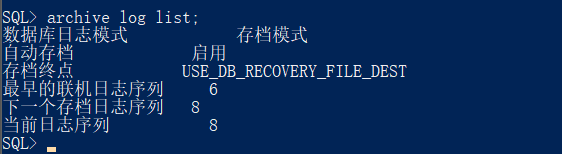


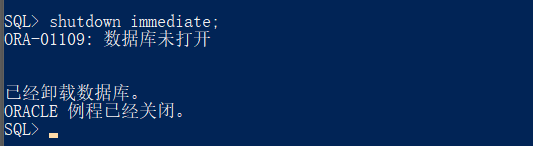
1. 手动备份这些文件
2. 数据更改后只需将原数据复制到原来的文件即可

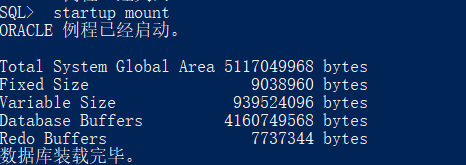
### 7.2 自动备份

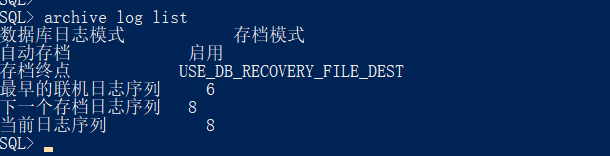
-- 让表空间进入备份模式   
ALTER TABLESPACE \*\*\* BEGIN BACKUP   
-- 复制表空间文件   
-- 让表空间结束备份模式   
ALTER TABLESPACE \*\*\* END BACKUP   
-- 复制控制文件和初始化文件   
-- 临时停止归档   
ALTER SYSTEM SWITCH LOGFILE;   
ALTER SYSTEM ARCHIVE LOG STOP;   
-- 复制归档日志文件   
-- 重新开启归档   
ALTER SYSTEM ARCHIVE LOG START;

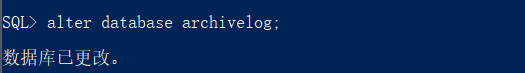
--1.查看归档模式   
archive log list;   
2.以archive模式启动数据库   
 2.2 shutdown immediate   
 2.3 startup mount   
 2.4 alter database archivelog;   
3.archive log list   
4.alter database open;   
5.ALTER tablespace users begin backup;





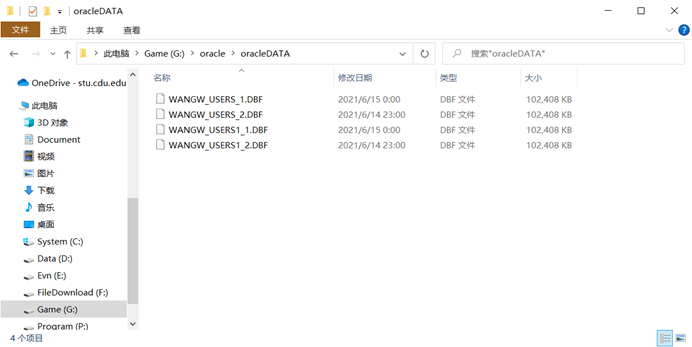




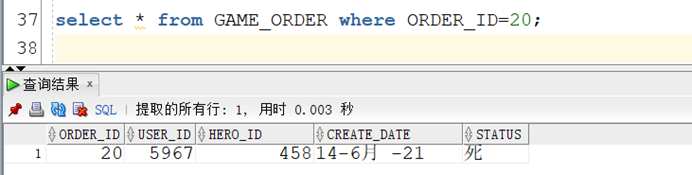


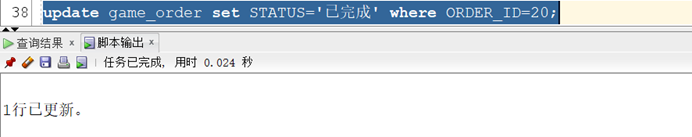


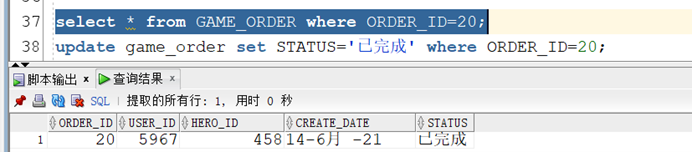
复制数据库文件



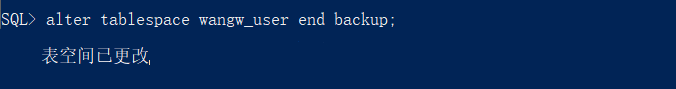
修改数据







结束备份



## 8.总结

通过本次期末项目，让我对Oracle的基本概念如表和视图理解的更加深刻了，Oracle 数据库数据对象中最基本的是表和视图，其他还有约束、序列、函数、存储过程、包、触发器等。对数据库的操作可以基本归结为对数据对象的操作,理解和掌握 Oracle数据库对象是学习Oracle的捷径。同时让我对以前就比较迷惑的数据库、实例、表空间、用户、角色之间的关系豁然开朗：

* 数据库是物理数据、内存、操作系统进程的组合体
* 实例是用来访问和使用数据库的一块进程，它只存在于内存中
* Oracle数据库是通过表空间来存储物理表的，一个数据库实例可以有N个表空间，一个表空间下可以有N张表

本次实验让我对创建表空间的语法，条件语句、分支语句、循环语句的使用方法和常用的PL/SQL函数掌握的更加熟练了，掌握了使用SQL语句创建视图，并且学会了部分存储过程和触发器的使用。

通过本次期末项目，让我收获很大，对自己的能力与不足有了更加清晰的认识，并且对Oracle也更加的充满激情了