卡通人物

中度可信度描述已自动生成

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的某项目的数据库设计 | | |
| 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 | 吴俊泠 | 学号 | 201810414322 |
| 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

目录

[**实验6（期末考核） 基于Oracle的火车售票系统数据库设计** 2](#_Toc74607413)

[**期末考核要求** 2](#_Toc74607414)

[**1.项目简介** 2](#_Toc74607415)

[**2.功能分析** 2](#_Toc74607416)

[**车次管理** 2](#_Toc74607417)

[**车站管理** 3](#_Toc74607418)

[**用户管理** 3](#_Toc74607419)

[**订单管理** 3](#_Toc74607420)

[**票务查询** 3](#_Toc74607421)

[**登录注册** 3](#_Toc74607422)

[**3.数据库设计** 3](#_Toc74607423)

[**3.1admin表（管理员表）** 3](#_Toc74607424)

[**3.2consumer表（普通用户表）** 4](#_Toc74607425)

[**3.3station表（车站表）** 4](#_Toc74607426)

[**3.4orders表（订单表）** 5](#_Toc74607427)

[**3.5numbers表（车次表）** 5](#_Toc74607428)

[**3.6seat表（座位表）** 5](#_Toc74607429)

[**3.7ticket表（车票信息表）** 6](#_Toc74607430)

[**4.创建表空间并插入数据** 6](#_Toc74607431)

[**5.设计权限及用户分配方案,两类角色，两个用户** 26](#_Toc74607432)

[**6.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** 28](#_Toc74607433)

[**7.设计手动备份方案** 31](#_Toc74607434)

[**7.1创建恢复目录** 31](#_Toc74607435)

[**7.2连接RMAN恢复目录数据库** 32](#_Toc74607436)

[**7.3通道分配** 33](#_Toc74607437)

[**7.4归档模式下备份与恢复** 34](#_Toc74607438)

[**7.5测试备份情况** 35](#_Toc74607439)

[**7.6测试恢复功能** 35](#_Toc74607440)

[**8.项目总结** 36](#_Toc74607441)

**实验6（期末考核） 基于Oracle的火车售票系统数据库设计**

**期末考核要求**

* 自行设计一个信息系统的数据库项目，自拟火车车售票系统名称。
* 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：火车售票系统

本项目是基于Oracle的火车售票系统数据库设计。

涉及角色/用户：管理员、普通用户

涉及表：管理员表、普通用户表、订单表、车次表、座位表、车站表、车票信息表

**2.功能分析**

**车次管理**

由管理员进行的，包含车次查询、车次增加、车次修改、车次删除，通过操作数据库来实现这些操作。

**车站管理**

由管理员进行的，包含车站查询、车站增加、车站修改、车站删除，通过操作数据库来实现这些操作。

**用户管理**

用户管理功能指两个方面，一个是指管理员和普通用户都可以对自己的用户信息进行管理，包括查看、修改等操作，另一个是指管理员对普通用户的用户管理，包括查询用户、删除用户等。

**订单管理**

由管理员进行的，包含查看订单、退订、改签，通过操作数据库来实现这些操作。

**票务查询**

由普通用户进行的，包含车次查询、车站查询、余票查询，通过从数据库匹配关键信息来实现这些操作。

**登录注册**

登录注册功能，管理员和普通用户可以进行登录和注册的功能。

**3.数据库设计**

**3.1admin表（管理员表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| admin\_id | Integer（20） | 主键 | 否 |  |  |
| admin\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| admin\_password | VARCHAR(100 BYTE) |  | 否 |  |  |
| admin\_phone | Integer（20） |  | 否 |  |  |
| admin\_real\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| admin\_identity | Integer（20） |  | 否 |  |  |
| admin\_comment | VARCHAR(100 BYTE) |  | 是 | 空 |  |

​

​

**3.2consumer表（普通用户表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| user\_id | Integer（20） | 主键 | 否 |  |  |
| user\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| user\_password | VARCHAR(100 BYTE) |  | 否 |  |  |
| user\_real\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| user\_phone\_num | Integer（20） |  | 否 |  |  |
| user\_identity | Integer（20） |  | 否 |  |  |
| user\_level | Integer（20） |  | 是 | 空 |  |
| user\_point | Integer（20） |  | 是 | 空 |  |
| user\_comment | VARCHAR(100 BYTE) |  | 是 | 空 |  |

**3.3station表（车站表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| station\_id | Integer（20） | 主键 | 否 |  |  |
| station\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| station\_addr | VARCHAR(100 BYTE) |  | 否 |  |  |

​

​

**3.4orders表（订单表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| order\_id | Integer（20） | 主键 | 否 |  |  |
| order\_time | Date |  | 否 |  |  |
| pay\_info | VARCHAR(100 BYTE) |  | 否 |  |  |
| order\_name | VARCHAR(100 BYTE) |  | 否 |  |  |
| order\_train\_info | VARCHAR(100 BYTE) |  | 否 |  |  |
| order\_seat | VARCHAR(100 BYTE) |  | 否 |  |  |
| order\_identity | Integer（20） |  | 否 |  |  |
| order\_phone | Integer（20） |  | 否 |  |  |

​

​

**3.5numbers表（车次表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| train\_id | VARCHAR(100 BYTE) | 主键 | 否 |  |  |
| start\_time | Date |  | 否 |  |  |
| arrive\_time | Date |  | 否 |  |  |
| start\_station | VARCHAR(100 BYTE) |  | 否 |  |  |
| arrive\_station | VARCHAR(100 BYTE) |  | 否 |  |  |
| whole\_time | VARCHAR(100 BYTE) |  | 否 |  |  |
| whole\_course | VARCHAR(100 BYTE) |  | 否 |  |  |

**3.6seat表（座位表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| train\_id | VARCHAR(100 BYTE) | 外键 | 否 |  |  |
| spare\_num | Integer（20） |  | 否 |  |  |
| total\_num | Integer（20） |  | 否 |  |  |

**3.7ticket表（车票信息表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** |
| --- | --- | --- | --- | --- | --- |
| ticket\_id | Integer（20） | 主键 | 否 |  |  |
| train\_id | VARCHAR(100 BYTE) | 外键 | 否 |  |  |
| ticket\_price | Float（20） |  | 否 |  |  |

**4.创建表空间并插入数据**

创建表空间：

user\_2、train

每个表插入数据：

admin：20000条

consumer：20000条

station：20000条

orders：10000条

numbers：10000条

seat：10000条

ticket：10000条

-- 创建表空间  
CREATE TABLESPACE user\_2 logging DATAFILE 'D:\orcale19c\oradata\ORCL\user\_2.dbf' size 64m autoextend on next 65m maxsize 10240m extent management local;  
CREATE TABLESPACE train logging DATAFILE 'D:\orcale19c\oradata\ORCL\train.dbf' size 64m autoextend on next 65m maxsize 10240m extent management local;

文本

描述已自动生成

创建表：

-- admin表的创建  
create table admin  
(  
admin\_id number(20) primary key,  
admin\_name VARCHAR2(100 BYTE） not null,  
admin\_password VARCHAR2(100 BYTE） not null,  
admin\_phone number(20) not null,  
admin\_real\_name VARCHAR2(100 BYTE） not null,  
admin\_identity number(20) not null,  
admin\_comment VARCHAR2(100 BYTE）  
)  
tablespace user\_2   
pctfree 10   
initrans 1   
maxtrans 255   
storage   
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

电脑的截图

描述已自动生成

-- consumer表的创建  
create table consumer  
(  
user\_id number(20) primary key,  
user\_name VARCHAR2(100 BYTE） not null,  
user\_password VARCHAR2(100 BYTE） not null,  
user\_real\_name VARCHAR2(100 BYTE） not null,  
user\_phone\_num number(20) not null,                            
user\_identity number(20) not null,  
user\_level number(20),  
user\_point number(20),                          
user\_comment VARCHAR2(100 BYTE）  
)  
tablespace user\_2   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

文本

描述已自动生成

-- station表的创建  
create table station  
(  
station\_id number(20) primary key,  
station\_name VARCHAR2(100 BYTE） not null,  
station\_addr VARCHAR2(100 BYTE） not null  
)  
tablespace train   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

文本

描述已自动生成

-- orders表的创建  
create table orders  
(  
order\_id number(20) primary key,  
order\_time Date not null,  
pay\_info VARCHAR2(100 BYTE） not null,  
order\_name VARCHAR2(100 BYTE） not null,  
order\_train\_info VARCHAR2(100 BYTE) not null,                      
order\_seat VARCHAR2(100 BYTE) not null,  
order\_identity number(20) not null,  
order\_phone number(20) not null                          
)  
tablespace train   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

电脑截图

描述已自动生成

-- numbers表的创建  
create table numbers  
(  
train\_id VARCHAR2(100 BYTE） primary key,  
start\_time Date not null,  
arrive\_time Date not null,  
start\_station VARCHAR2(100 BYTE） not null,  
arrive\_station VARCHAR2(100 BYTE） not null,  
whole\_time VARCHAR2(100 BYTE） not null,                         
whole\_course VARCHAR2(100 BYTE） not null     
)  
tablespace train   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

文本

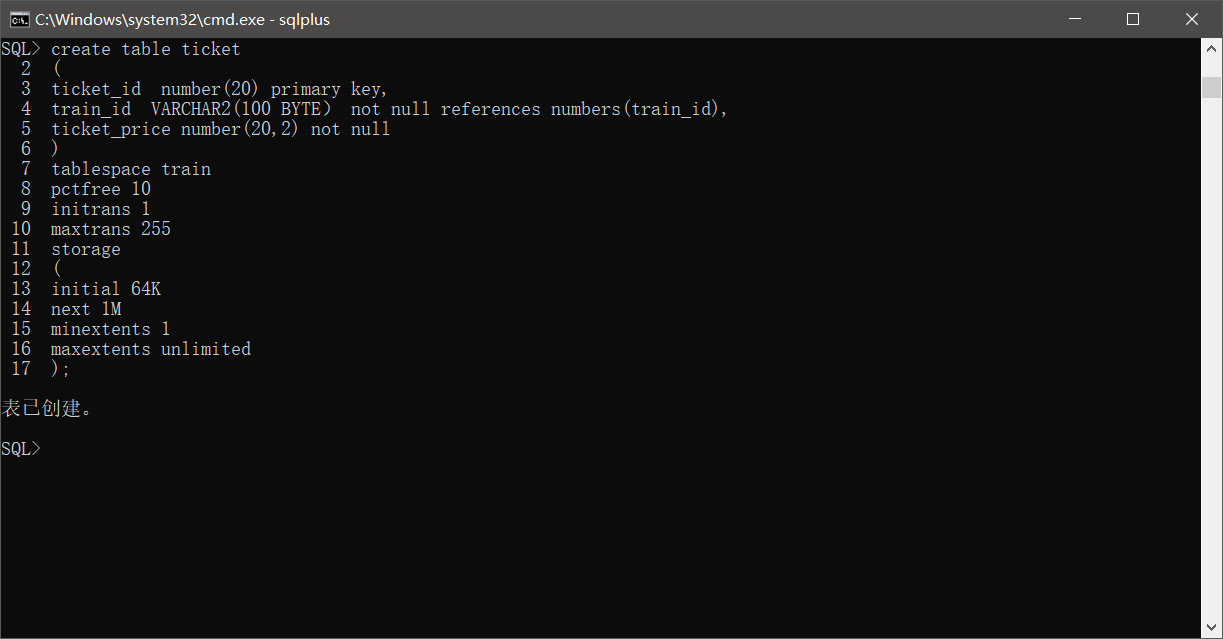
描述已自动生成

-- seat表的创建  
create table seat  
(  
train\_id VARCHAR2(100 BYTE） not null references numbers(train\_id),  
spare\_num number(20) not null,  
total\_num number(20) not null       
)  
tablespace train   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);

文本

描述已自动生成

-- ticket表的创建  
create table ticket  
(  
ticket\_id number(20) primary key,  
train\_id VARCHAR2(100 BYTE） not null references numbers(train\_id),  
ticket\_price number(20,2) not null      
)  
tablespace train   
pctfree 10   
initrans 1   
maxtrans 255   
storage                             
(  
initial 64K   
next 1M  
minextents 1   
maxextents unlimited   
);



插入数据：

总共插入数据：100000条

admin：20000条

consumer：20000条

station：20000条

orders：10000条

numbers：10000条

seat：10000条

ticket：10000条

-- admin表的数据插入20000

declare

i int;

admin\_id number(20);

admin\_name VARCHAR2(100);

admin\_password VARCHAR2(100);

admin\_phone number(20);

admin\_real\_name VARCHAR2(100);

admin\_identity number(20);

admin\_comment VARCHAR2(100);

begin

i:=1;

while i<=20000

loop

admin\_id:=i;

admin\_name:= 'admin'|| i;

admin\_password := '520'|| i;

admin\_phone := '19822'|| i;

admin\_real\_name := 'wjl'||i;

admin\_identity := '511622'||i;

admin\_comment := '就离谱'||i;

insert into admin(admin\_id,admin\_name,admin\_password,admin\_phone,admin\_real\_name,admin\_identity,admin\_comment) values (admin\_id,admin\_name,admin\_password,admin\_phone,admin\_real\_name,admin\_identity,admin\_comment);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- consumer表数据插入20000

declare

i int;

user\_id number(20);

user\_name VARCHAR2(100);

user\_password VARCHAR2(100);

user\_phone\_num number(20);

user\_real\_name VARCHAR2(100);

user\_identity number(20);

user\_level number(20);

user\_point number(20);

user\_comment VARCHAR2(100);

begin

i:=1;

while i<=20000

loop

user\_id:=i;

user\_name:= 'wcy'|| i;

user\_password := '123'|| i;

user\_phone\_num := '19822'||i;

user\_real\_name := 'wjl'||i;

user\_identity := '511622'||i;

user\_level := '5'||i;

user\_point := '5'||i;

user\_comment := '就离谱'||i;

insert into consumer(user\_id,user\_name,user\_password,user\_real\_name,user\_phone\_num,user\_identity,user\_level,user\_point,user\_comment) values (user\_id,user\_name,user\_password,user\_real\_name,user\_phone\_num,user\_identity,user\_level,user\_point,user\_comment);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- station表的数据插入20000

declare

i int;

station\_id number(20);

station\_name VARCHAR2(100);

station\_addr VARCHAR2(100);

begin

i:=1;

while i<=20000

loop

station\_id:=i;

station\_name:= 'xx站'|| i;

station\_addr := 'xx市'|| i;

insert into station(station\_id,station\_name,station\_addr) values (station\_id,station\_name,station\_addr);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- orders表的数据插入10000

declare

i int;

order\_id number(20);

order\_time Date ;

pay\_info VARCHAR2(100);

order\_name VARCHAR2(100);

order\_train\_info VARCHAR2(100) ;

order\_identity number(20);

order\_seat VARCHAR2(100);

order\_phone number(20);

begin

i:=1;

while i<=10000

loop

order\_id:=i;

if i mod 6 =0 then

order\_time:=to\_date('2000-1-5','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =1 then

order\_time:=to\_date('2002-5-12','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =2 then

order\_time:=to\_date('2003-5-25','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =3 then

order\_time:=to\_date('2005-9-6','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =4 then

order\_time:=to\_date('2006-6-22','yyyy-mm-dd')+(i mod 60);

else

order\_time:=to\_date('2007-7-11','yyyy-mm-dd')+(i mod 60);

end if;

pay\_info:= '未支付';

order\_name := 'wjl'||i;

order\_train\_info :=i;

order\_identity := '511622'||i;

order\_phone := '19822'||i;

order\_seat :='A'||i mod 60;

insert into orders(order\_id,order\_time,pay\_info,order\_name,order\_train\_info,order\_identity,order\_seat,order\_phone) values (order\_id,order\_time,pay\_info,order\_name,order\_train\_info,order\_identity,order\_seat,order\_phone);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- numbers表的数据插入10000

declare

i int;

train\_id VARCHAR2(100);

start\_time Date;

arrive\_time Date;

start\_station VARCHAR2(100);

arrive\_station VARCHAR2(100);

whole\_time VARCHAR2(100);

whole\_course VARCHAR2(100);

begin

i:=1;

while i<=10000

loop

train\_id:=i;

if i mod 6 =0 then

start\_time:=to\_date('2000-1-5','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =1 then

start\_time:=to\_date('2002-5-12','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =2 then

start\_time:=to\_date('2003-5-25','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =3 then

start\_time:=to\_date('2005-9-6','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =4 then

start\_time:=to\_date('2006-6-22','yyyy-mm-dd')+(i mod 60);

else

start\_time:=to\_date('2007-7-11','yyyy-mm-dd')+(i mod 60);

end if;

if i mod 6 =0 then

arrive\_time:=to\_date('2000-1-5','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =1 then

arrive\_time:=to\_date('2002-5-12','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =2 then

arrive\_time:=to\_date('2005-9-6','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =3 then

arrive\_time:=to\_date('2006-6-22','yyyy-mm-dd')+(i mod 60);

elsif i mod 6 =4 then

arrive\_time:=to\_date('2007-7-11','yyyy-mm-dd')+(i mod 60);

else

arrive\_time:=to\_date('2007-7-11','yyyy-mm-dd')+(i mod 60);

end if;

start\_station:= '起点'||i;

arrive\_station := '终点'|| i;

whole\_time := (TO\_NUMBER(arrive\_time - start\_time));

whole\_course := '120'||i mod 6;

insert into numbers(train\_id,start\_time,arrive\_time,start\_station,arrive\_station,whole\_time,whole\_course) values (train\_id,start\_time,arrive\_time,start\_station,arrive\_station,whole\_time,whole\_course);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- seat表的数据的插入10000条

declare

i int;

train\_id VARCHAR2(100);

spare\_num number(20);

total\_num number(20);

begin

i:=1;

while i<=10000

loop

train\_id:=i;

spare\_num:= i mod 60;

total\_num := 60;

insert into seat(train\_id,spare\_num,total\_num) values (train\_id,spare\_num,total\_num);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- ticket表的数据插入10000条

declare

i int;

ticket\_id number(20);

train\_id VARCHAR2(100);

ticket\_price number(20,2);

begin

i:=1;

while i<=10000

loop

ticket\_id:=i;

train\_id:=i;

ticket\_price:=dbms\_random.value(400,100);

insert into ticket(ticket\_id,train\_id,ticket\_price) values (ticket\_id,train\_id,ticket\_price);

i:=i+1;

end loop;

commit;

end;

/

文本

描述已自动生成

-- 查询是否插入成功

select count(\*) from admin;

select count(\*) from consumer;

select count(\*) from station;

select count(\*) from orders;

select count(\*) from numbers;

select count(\*) from seat;

select count(\*) from ticket;

文本

描述已自动生成

**5.设计权限及用户分配方案,两类角色，两个用户**

用户：

admin

拥有角色：t\_admin

分配表空间：train

c##consumer

拥有角色：t\_user

分配表空间：user

角色：

t\_admin

拥有角色：connect、resource、dba

t\_user

拥有角色：connect、resource

-- 创建角色

CREATE ROLE t\_admin;

CREATE ROLE t\_user;

-- 授权角色

GRANT connect,resource,dba, create table,create view,create trigger, create procedure,create sequence TO t\_admin;

GRANT connect,resource, create table,create view,create trigger, create procedure,create sequence TO t\_user;

-- 创建用户

CREATE USER admin IDENTIFIED BY 520 DEFAULT TABLESPACE train;

-- 指定用户额外表空间

ALTER USER admin QUOTA UNLIMITED ON user\_2;

-- 创建用户

CREATE USER consumer IDENTIFIED BY 520 DEFAULT TABLESPACE user\_2;

-- 指定用户额外表空间

ALTER USER consumer QUOTA UNLIMITED ON train ;

-- 分配角色给用户

GRANT t\_admin TO admin;

GRANT t\_user TO consumer;

成功截图如下：

文本

描述已自动生成

文本

描述已自动生成

**6.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**

建立的包：

包名： pack

函数：

函数名：get\_count(o\_id num)

功能：

通过输入订单id，来查询订单表和车次表，同时返回与该订单相关的车次对应的起始车站

过程：

过程名：get\_orders(t\_id var)

功能：

通过输入订单id，来获取与该订单相关的车次的起始车站，并且统计同一时间段的车次数量并输出与订单相关的车票价格

-- 创建包

create or replace PACKAGE pack IS

FUNCTION get\_count(o\_id num) RETURN var;

PROCEDURE get\_orders(t\_id var);

END TrainPack;

-- 创建函数和过程

create or replace PACKAGE BODY pack IS

FUNCTION get\_count(o\_id num) RETURN var

AS

x var(100);

y var(100);

BEGIN

select order\_train\_info into y from orders where order\_id=o\_id;

select start\_station into x from numbers where train\_id=y;

RETURN x;

END;

PROCEDURE get\_orders(t\_id var)

AS

N NUMBER(20);

L date;

S var(20);

R var(100);

begin

DBMS\_OUTPUT.PUT\_LINE('出发站：');

select start\_station into R from train\_num where train\_id = t\_id;

DBMS\_OUTPUT.PUT\_LINE(R);

select start\_time into L from train\_num where train\_id = t\_id;

select count(\*) into N from train\_num where start\_time = L;

DBMS\_OUTPUT.PUT\_LINE('车次数量：');

DBMS\_OUTPUT.PUT\_LINE(N);

select ticket\_price into S from ticket where train\_id = t\_id;

DBMS\_OUTPUT.PUT\_LINE('票价：');

DBMS\_OUTPUT.PUT\_LINE(S);

END;

END TrainPack;

/

测试：

-- 测试函数

select TrainPack.Get\_count(11) AS 订单11起始车站,TrainPack.Get\_count(12) AS 订单12起始车站 from dual;

-- 测试过程

DECLARE

t\_id var(100);

BEGIN

t\_id := 1;

TrainPack.Get\_orders (t\_id) ;

t\_id := 2;

TrainPack.Get\_orders (t\_id) ;

END;

**7.设计手动备份方案**

本项目设置手动备份方案——即采用Rman备份

**7.1创建恢复目录**

-- 创建恢复目录：用来存储RMAN资料库的

create tablespace bp datafile 'D:\orcale19c\oradata\ORCL\storage.dbf' size 20m autoextend on next 5m maxsize unlimited;

-- 在恢复目录数据库中创建RMAN用户并授权

create user wjl identified by 520 default tablespace bp quota unlimited on bp;

grant connect,resource,recovery\_catalog\_owner to wjl;

文本

描述已自动生成

**7.2连接RMAN恢复目录数据库**

-- 连接RMAN恢复目录数据库

rman catalog wjl/520

-- 创建恢复目录

create catalog tablespace bp;

-- 退出

quit

-- 确认环境信息

echo $ORACLE\_SID

-- 连接到目标数据库、连接到恢复目录数据库

rman catalog wjl/520 target /

-- 向恢复目录注册数据库ORCL——此时就可以使用RMAN的恢复目录对目标数据库进行备份和恢复操作

register database;

文本

描述已自动生成

**7.3通道分配**

-- 手动通道配置

run

{

allocate channel td1 device type disk;

allocate channel td2 device type disk;

allocate channel td3 device type disk;

}

-- 显示已经配置过的有默认值的参数，其中包括通道参数

show all;

文本

描述已自动生成

**7.4归档模式下备份与恢复**

-- 查看数据库是否处于归档模式下

archive log list;

-- 关闭数据库

shutdown immediate

-- 重启并设置成归档模式

startup mount;

alter database archivelog;

archive log list;

alter database open;

-- 连接到目标数据库、连接到恢复目录数据库

rman catalog c##zrf\_r/123 target /

-- 备份和恢复整个数据库

backup database;

**7.5测试备份情况**

-- 切换到保存路径

cd /home/oracle/app/oracle/recovery\_area/ORCL/backupset/

-- 查看文件

ls

**7.6测试恢复功能**

-- 关闭数据库

shutdown immediate;

-- 退出数据库

exit

-- 切换到数据文件存储路径

cd /home/oracle/app/oracle/oradata/orcl/pdborcl

-- 查看数据文件

ls

-- 删除train\_data01.dbf

rm -rf train\_data01.dbf

-- 再次确认

ls

-- 启动数据库

sqlplus /nolog

conn /as sysdba

startup

-- 此时因为缺少数据文件无法启动

-- 检查此时数据库状态

select status from v$instance;

-- 连接RMAN

rman target sys/123

-- 恢复数据库

restore database;

-- 同步恢复

recover database;

-- 打开数据库

alter database open resetlogs;

-- 再次启动数据库，启动成功，检查此时数据库状态，此时状态已经打开

startup

select status from v$instance;

-- 再次切换到数据文件存储路径

cd /home/oracle/app/oracle/oradata/orcl/pdborcl

-- 查看数据文件，删除的已经还原

ls

**8.项目总结**

本次项目，是对我们本学期所有Oracle学习的总结，在本次的期末项目中，我所做的是火车售票系统的一个数据库，因为第一次接触oracle这个大型的数据库，所以在很多方面还有很多不会，于是就上网查资料并且在一些方面去询问同学来解决自己所遇到的问题，最后也算成功的完成了这个项目，不过在最后备份时出现了一些问题，在我打算删除数据库来进行后面的备份恢复时发现了无论是自己的system登录还是自己所创建的用户登录，都无法删除数据库，都提示权限不足，上网也查了资料，但仍无法解决，这也算是这个项目的一个小缺点吧。

最后通过这个项目，让我对Oracle有了更深的理解，也学到了许多新的知识。