

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的图书购买系统的数据库设计 | | |
| 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 | 2017级 |
| 学生姓名 | 许瑞峰 | 学号 | 201710414123 |
| 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 | 10 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 25 |  |
| 备份方案 | 备份方案设计正确 | 25 |  |
| 容灾方案 | DataGuard设计正确 | 10 |  |
| **得分合计** | | |  |

2019 年 11 月 18 日

## 第1章．系统概述

网上图书商城，顾名思义，网站式的书店。是一种高质量，更快捷，更方便的购书方式。其规模相比一般书店要大，网上图书商城不仅可用于图书的在线销售，也有音碟、影碟的在线销售。而且网站式的书城对图书的管理更加合理化，信息化。售书的同时还具有书籍类商品管理、购物车、订单管理等功能，非常灵活的网站内容和文章管理功能。但网上书城的真实性是消费者的最大顾虑，这也是网上书店还不被大多数消费者所接受的原因之一。

## 第2章．业务流程

业务或业务活动是对组织或企业的一切专业工作和活动的总称。业务流程图是根据系统详细调查过程中所得的资料和问卷调查的结果，按业务实际处理过程绘制成的图。

1. 用户获取到所有的图书信息，可以将其放入购物车。
2. 在购物车中可以修改图书数量，并可以提交订单，清空购物车

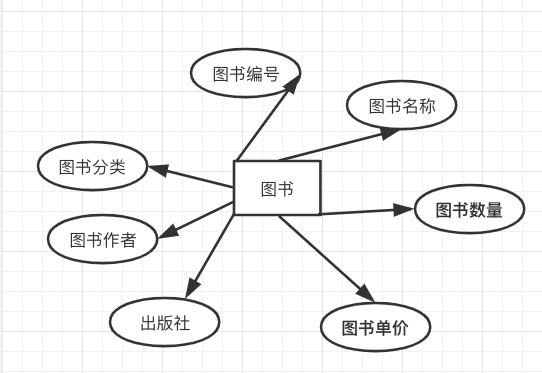
(3)在用户购买书籍后，相应的书籍的库存里对减少图书的数量，并且生成对应的订单。

**第3章. 实体分析**

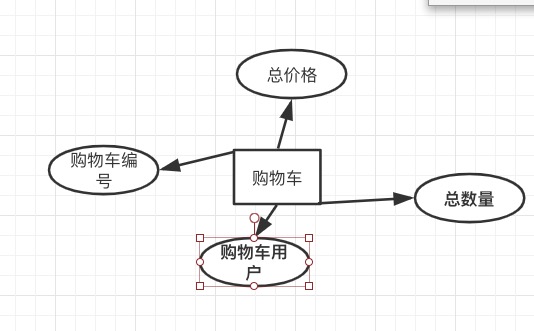
3.1**实体模型**

根据业务流程分析，一共有6个实体，分别为：图书，购物车，订单，用户，图书-订单，图书-购物车。

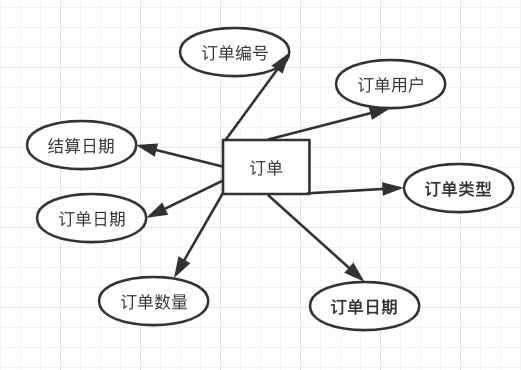
1)图书（book）：图书编号(book\_no)、图书名称(book\_name)、图书分类(book\_class)、图书作者(book\_author)、出版舍(book\_out)、图书单价(book\_price)，图书数量（book\_num）



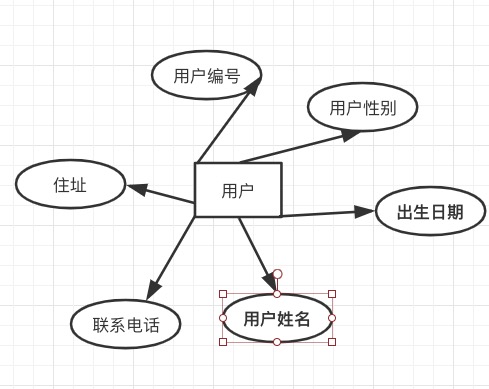
1. 购物车（cart）:购物车编号(cart\_no)，购物车价格(cart\_price)，购物车数量(section\_num)，购物车用户(cart\_user)



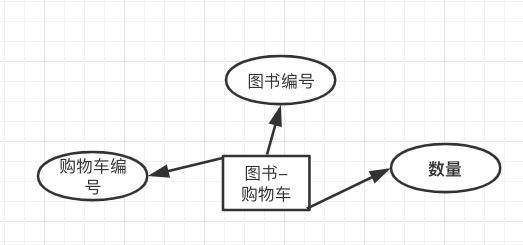
1. 订单（order）:订单编号(order\_no)，订单用户(order\_user)，订单类型(order\_type)，订单日期(order\_date)，订单数量(order\_num)，订单价格(order\_price)，结算时间(order\_pay\_data)

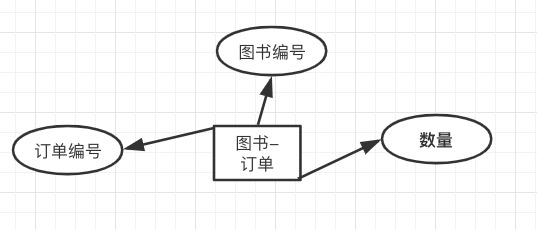


4）用户（user）:用户编号(user\_no)，用户姓名(user\_name)，性别(sex)，出生日期(birthday)，联系电话(tel)，住址(address)。



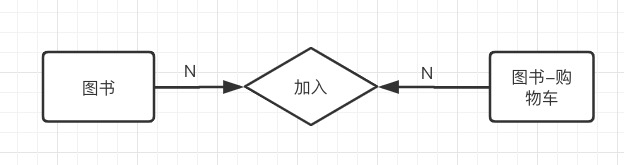
1. 图书-订单（book\_order）,图书编号（book\_no），订单编号（order\_no），数量（num）



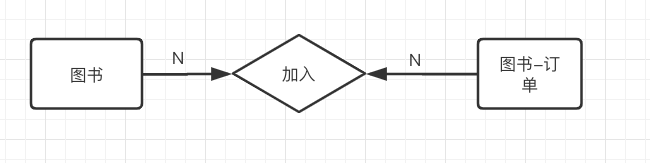
1. 图书-购物车（book\_cart），图书编号（book\_no），购物车编号（cart\_no），数量（num） 

3.2 实体联系模型

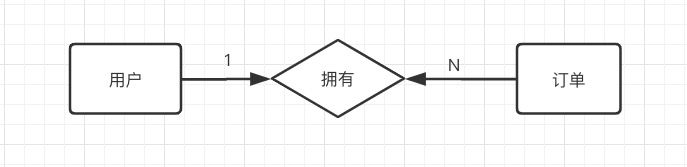
1）一本图书可在多个购物车里，一个购物车里可有多个图书。



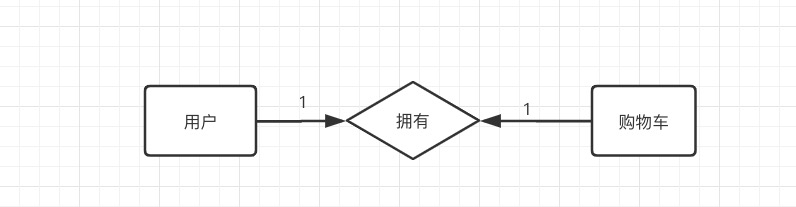
1. 同理，图书和订单的关系



1. 一个人可以有多个订单



1. 一个用户只有一个购物车



# 第4章 数据表的设计

1）图书表(book)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| book\_no | char(5) | 否 | 主键 |  |
| book\_name | char(20) | 否 |  |  |
| book\_class | char(8) |  |  |  |
| book\_author | char(10) |  |  |  |
| book\_out | char(10) |  |  |  |
| book\_num | int | 否 |  | 大于0 |
| book\_price | float | 否 |  | 大于0 |

1. 购物车表(cart)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| cart\_no | char(5) | 否 |  |  |
| cart\_price | char(5) | 否 |  | 大于0 |
| cart\_num | char(10) |  |  | 大于0 |
| cart\_user | char(20) | 否 | 外键 |  |

1. 订单表(order)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| order\_no | char(5) | 否 | 主键 |  |
| order\_user | char(10) | 否 | 外键 |  |
| order\_class | char(5) | 否 |  |  |
| order\_date | char(11) |  |  |  |
| order\_price | char(5) |  |  | 大于0 |
| order\_pay\_date | char(11) |  |  |  |
| order\_num | float |  |  | 大于0 |

4）用户表(user)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| user\_no | char(5) | 否 | 主键 |  |
| user\_name | char(10) | 否 |  |  |
| sex | char(2) | 否 |  | 男，女 |
| phone | char(5) | 否 |  |  |
| address | char(10) | 否 |  |  |
| birthday | char(10) | 否 |  |  |

1. 图书-订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| book\_no | char(5) | 否 | 主键 |  |
| order\_no | char(5) | 否 | 主键 |  |
| num | float | 否 |  | 大于0 |

6）图书-购物车表

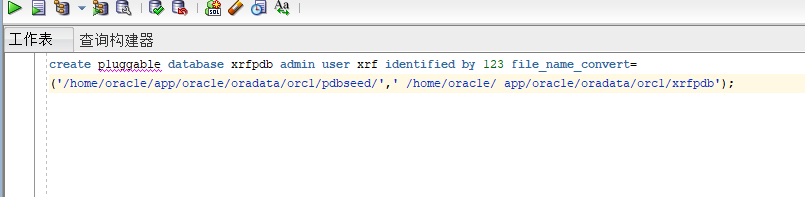
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 是否为空 | 主键/外键 | 取值范围 |
| book\_no | char(5) | 否 | 主键 |  |
| cart\_no | char(5) | 否 | 主键 |  |
| num | float | 否 |  | 大于0 |

## 创建pdb数据库，表空间，表，以及用户

5.1 创建pdb数据库以及管理员

创建了一个名叫xrfpdb的数据库，指定了它的存储位置，以及创建了数据库xrfpdb的管理员，具体sql如下

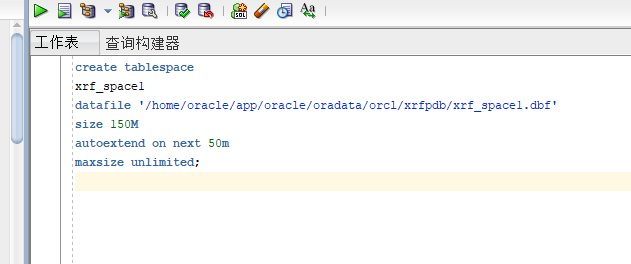
|  |
| --- |
| create p1uggable database xrfpdb admin user xrf identified by 123 file\_name\_convert=('/home/orac1e/app/orac1e/oradata/orc1/pdbseed/','/home/orac1e/app/orac1e/oradata/orc1/xrfpdb'); |



5.2 创建表空间

在xrfpdb数据库中创建了表空间xrf\_space1，表空间xrf\_space2,表空间xrf\_space3,表空间xrf\_space4

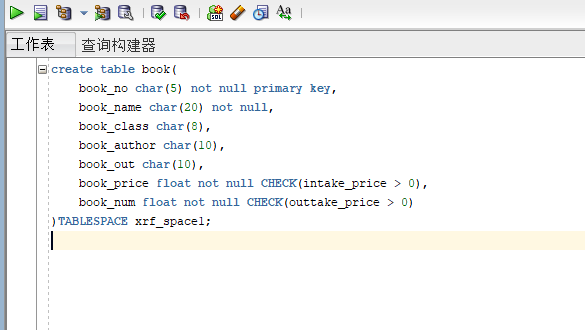
|  |
| --- |
| Create tablespace xrf\_space1datafile '/home/oracle/app/oracle/oradata/orcl/xrfpdb/xrf\_space1.dbf' size 150M autoextend on next 50m maxsize unlimited;  create tablespace xrf\_space2 DATAFILE '/home/oracle/app/oracle/oradata/orcl/xrfpdb/xrf\_space2.bdf' SIZE 50M AUTOEXTEND ON NEXT 5M MAXSIZE 100M;  create tablespace xrf\_space3 DATAFILE '/home/oracle/app/oracle/oradata/orcl/xrfpdb/xrf\_space3.bdf' SIZE 50M AUTOEXTEND ON NEXT 5M MAXSIZE 100M;  create tablespace xrf\_space4 DATAFILE '/home/oracle/app/oracle/oradata/orcl/xrfpdb/xrf\_space4.bdf' SIZE 50M AUTOEXTEND ON NEXT 5M MAXSIZE 100M; |



### 5.3 创建表以及将registration\_form表进行分区

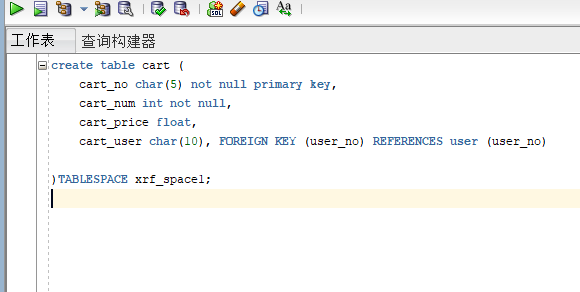
1）创建图书表

|  |
| --- |
| create table book(  book\_no char(5) not null primary key,  book\_name char(20) not null,  book\_class char(8),  book\_author char(10),  book\_out char(10),  book\_price float not null CHECK(intake\_price > 0),  book\_num float not null CHECK(outtake\_price > 0)  )TABLESPACE xrf\_space1; |



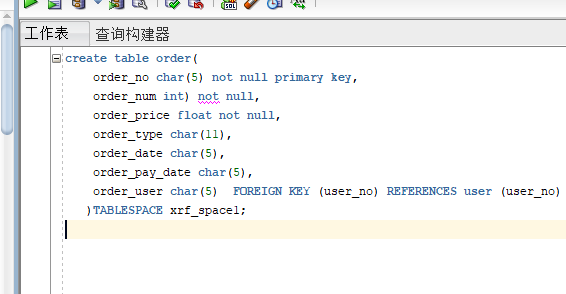
2）创建购物车表（cart）

|  |
| --- |
| create table cart (  cart\_no char(5) not null primary key,  cart\_num int not null,  cart\_price float,  cart\_user char(10) FOREIGN KEY (user\_no) REFERENCES user (user\_no)  )TABLESPACE xrf\_space1; |



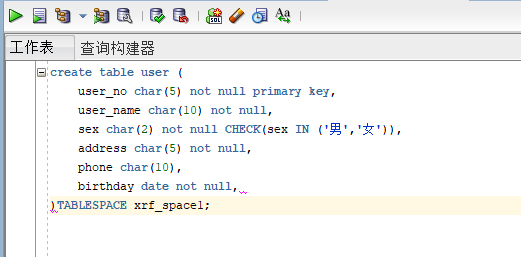
3）创建订单表（order）

|  |
| --- |
| create table order(  order\_no char(5) not null primary key,  order\_num int) not null,  order\_price float not null,  order\_type char(11),  order\_date char(5),  order\_pay\_date char(5),  order\_user char(5) FOREIGN KEY (user\_no) REFERENCES user (user\_no)  )TABLESPACE xrf\_space1; |



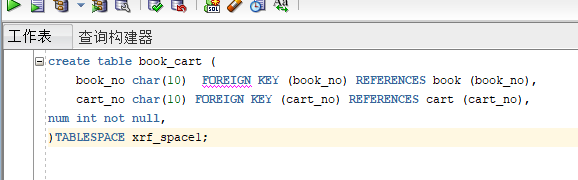
4）创建用户表（user）

|  |
| --- |
| create table user (  user\_no char(5) not null primary key,  user\_name char(10) not null,  sex char(2) not null CHECK(sex IN ('男','女')),  address char(5) not null,  phone char(10),  birthday date not null,  )TABLESPACE xrf\_space1; |



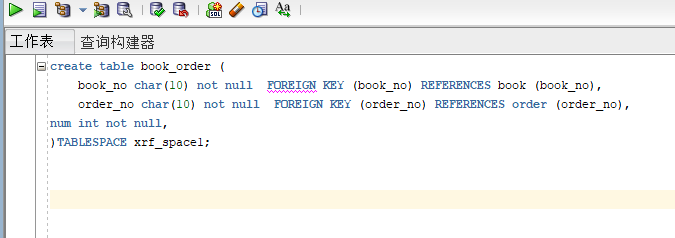
5）创建图书-购物车(book\_cart)

|  |
| --- |
| create table book\_cart (  book\_no char(5) not null FOREIGN KEY (book\_no) REFERENCES book (book\_no),  cart\_no char(10) not null FOREIGN KEY (cart\_no) REFERENCES cart (cart\_no),  num int not null,  )TABLESPACE xrf\_space1; |



6）创建图书-订单（book\_order）

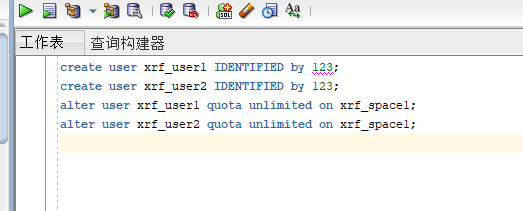
|  |
| --- |
| create table book\_order (  book\_no char(10) not null FOREIGN KEY (book\_no) REFERENCES book (book\_no),  order\_no char(10) not null FOREIGN KEY (order\_no) REFERENCES order (order\_no),  num int not null,  )TABLESPACE xrf\_space1; |



### 5.3 创建用户,授权,插入数据

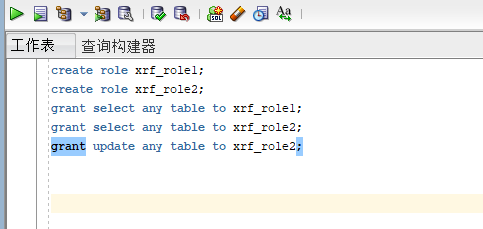
1）分别创建两个用户xrf\_user1和xrf\_user2,并允许他们是使用xrf\_space1表空间

|  |
| --- |
| create user xrf\_user1 IDENTIFIED by 123;  create user xrf\_user2 IDENTIFIED by 123;  alter user xrf\_user1 quota unlimited on xrf\_space1;  alter user xrf\_user2 quota unlimited on xrf\_space1; |



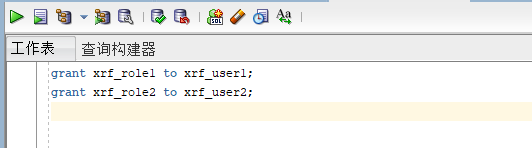
2）创建两个角色xrf\_role1，xrf\_role2，并分别赋予xrf\_role1，xrf\_role2读任何表和读、修改任何表的权利；

|  |
| --- |
| create role xrf\_role1;  create role xrf\_role2;  grant select any table to xrf\_role1;  grant select any table to xrf\_role2;  grant update any table to xrf\_role2; |



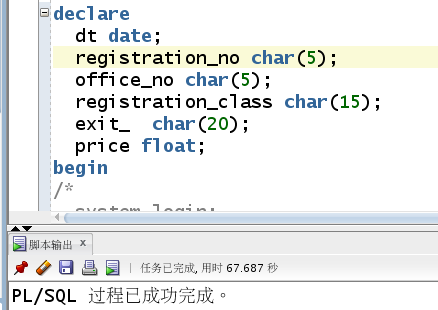
3）将xrf\_role1，xrf\_role2分别赋予给用户xrf\_user1，xrf\_user2

|  |
| --- |
| grant xrf\_role1 to xrf\_user1;  grant xrf\_role2 to xrf\_user2; |



1. 向表中插入数据：

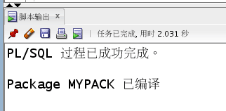
|  |
| --- |
| declare   dt date;   registration\_no char(5);   registration\_class char(15);   exit\_  char(20);   price float; begin   insert into book(book\_no,book\_name,book\_class,book\_author,book\_num,book\_price,book\_out) values ('1','java从入门到精通','计算机','xrf',100,12.5,'人民出版社');   insert into book(book\_no,book\_name,book\_class,book\_author,book\_num,book\_price,book\_out) values ('2',Python从入门到精通','计算机','xrf',200,22.5,'人民出版社');   insert into book(book\_no,book\_name,book\_class,book\_author,book\_num,book\_price,book\_out) values ('3','哈利波特','科幻','xrf',400,50.5,'国外出版社');   insert into book(book\_no,book\_name,book\_class,book\_author,book\_num,book\_price,book\_out) values ('4',C/C++从入门到精通','计算机','xrf',120,12.5,'人民出版社');   insert into book(book\_no,book\_name,book\_class,book\_author,book\_num,book\_price,book\_out) values ('5','红高粱','爱情','莫言',100,10.5,'人民出版社');      for i in 1..50000   loop     if i mod 3 =0 then       dt:=to\_date('2018-01-01','yyyy-mm-dd')+(i mod 60);  --PARTITION\_2018     elsif i mod 6 =1 then       dt:=to\_date('2019-01-01','yyyy-mm-dd')+(i mod 60);  --PARTITION\_2019     elsif i mod 6 =2 then       dt:=to\_date('2020-01-01','yyyy-mm-dd')+(i mod 60);  --PARTITION\_2020     end if;        --插入订单     registration\_no := i;     office\_no := case i mod 6 when 0 then '1001' when 1 then '1002' when 2 then '1003' when 4 then '1004' else '1005'end;    registration\_class := CASE i MOD 4 WHEN 0 THEN '科幻' WHEN 1 THEN '计算机' WHEN 2 THEN'爱情'WHEN 3 THEN '生活' END;      exit\_ := CASE i MOD 2 WHEN 0 THEN '是' ELSE '否' end;      price := dbms\_random.value(5,30);     insert /\*+append\*/ into REGISTRATION\_FORM (cart\_no,book\_no,book\_class,exit\_,price,registration\_date) values (registration\_no,cart\_no,registration\_class,exit\_,price,dt);      end loop; end; |



## 建立程序包

统计一段时间内的卖书总收入；统计一段时间内，各个种类的图书的销售数量：

|  |
| --- |
| create or replace PACKAGE MyPack  FUNCTION Get\_TOTAL(dt1 char,dt2 char) RETURN NUMBER;  PROCEDURE get\_people(dt1 char,dt2 char);  END MyPack;  /  create or replace PACKAGE BODY MyPack  FUNCTION Get\_TOTAL(dt1 char,dt2 char) RETURN NUMBER  AS  N NUMBER;  BEGIN  select sum(price) into N from REGISTRATION\_FORM where REGISTRATION\_DATE >= to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')  and REGISTRATION\_DATE <= to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');  RETURN N;  END;  -- 各个种类的图书的销售数量  PROCEDURE get\_people(dt1 char,dt2 char)  AS  a1 NUMBER;  b1 NUMBER;  c1 NUMBER;  d1 NUMBER;  cursor cur is  select \* from REGISTRATION\_FORM where REGISTRATION\_DATE >= to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')  and REGISTRATION\_DATE <= to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');  begin  a1 := 0;  b1 := 0;  c1 := 0;  d1 := 0;  for v in cur  LOOP  if v.REGISTRATION\_CLASS = '科幻'  then a1 := a1 + 1;  elsif v.REGISTRATION\_CLASS = '计算机'  then b1 := b1 + 1;  elsif v.REGISTRATION\_CLASS = '爱情'  then c1 := c1 + 1;  elsif v.REGISTRATION\_CLASS = '生活'  then d1 := d1 + 1;  end if;  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('科幻类图书数量为：' || a1);  DBMS\_OUTPUT.PUT\_LINE('计算机类图书数量为：' || b1);  DBMS\_OUTPUT.PUT\_LINE('爱情类图书数量为：' || c1);  DBMS\_OUTPUT.PUT\_LINE('生活类图书数量为：' || d1);  end;  END MyPack; |



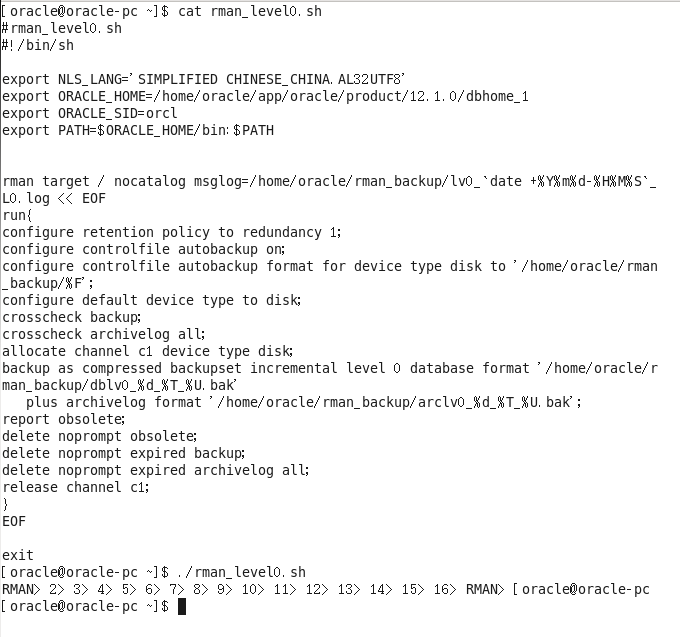
## 恢复备份

1. 编写rman增量备份脚本

|  |
| --- |
| #rman\_level1.sh  #!/bin/sh  export NLS\_LANG='SIMPLIFIED CHINESE\_CHINA.AL32UTF8'  export ORACLE\_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome\_1  export ORACLE\_SID=orcl  export PATH=$ORACLE\_HOME/bin:$PATH  rman target / nocatalog msglog=/home/oracle/rman\_backup/lv1\_`date +%Y%m%d-%H%M%S`\_L0.log << EOF  run{  configure retention policy to redundancy 1;  configure controlfile autobackup on;  configure controlfile autobackup format for device type disk to '/home/oracle/rman\_backup/%F';  configure default device type to disk;  crosscheck backup;  crosscheck archivelog all;  allocate channel c1 device type disk;  backup as compressed backupset incremental level 1 database format '/home/oracle/rman\_backup/dblv1\_%d\_%T\_%U.bak'  plus archivelog format '/home/oracle/rman\_backup/arclv1\_%d\_%T\_%U.bak';  report obsolete;  delete noprompt obsolete;  delete noprompt expired backup;  delete noprompt expired archivelog all;  release channel c1;  }  EOF  exit |

1. 开启全备份

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



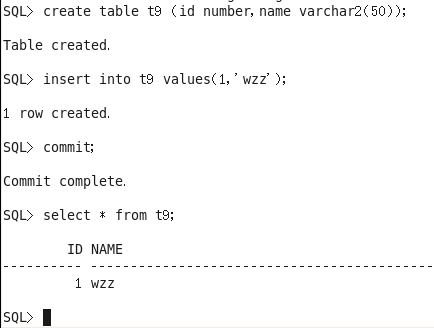
1. 每天定时开始增量备份

|  |
| --- |
| [oracle@oracle-pc ~]$ cat rman\_level1.sh  [oracle@oracle-pc ~]$ ./rman\_level1.sh |



4) 备份后修改数据

|  |
| --- |
| oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  SQL> create table t1 (id number,name varchar2(50));  SQL> insert into t1 values(1,'zhang');1 row created.  SQL> commit;  SQL> select \* from t1;  SQL> exit |

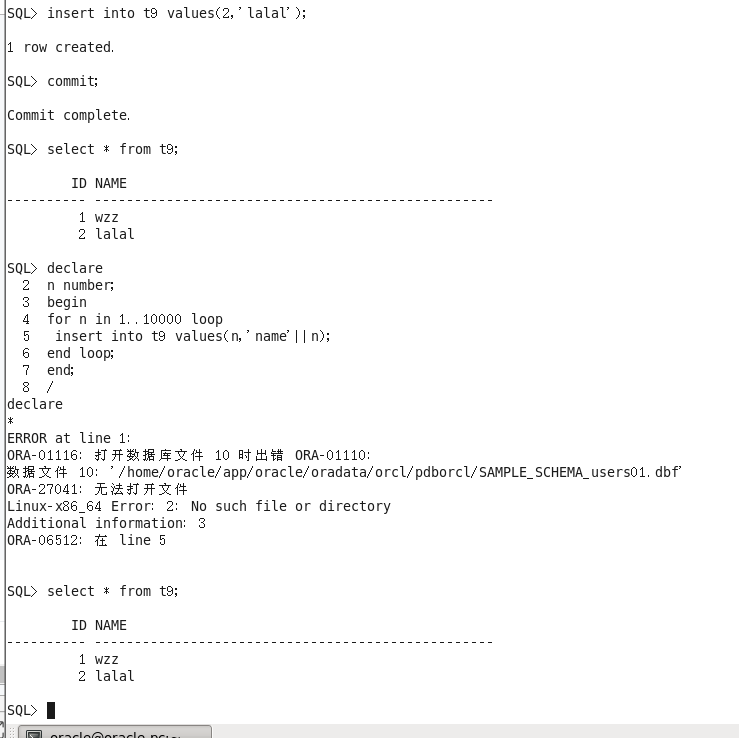


1. 删除数据库，模仿数据库被损坏

|  |
| --- |
| [oracle@oracle-pc~]$rm /home/oracle/app/oracle/oradata/orcl/pdborcl/SAMPLE\_SCHEMA\_users01.dbf |

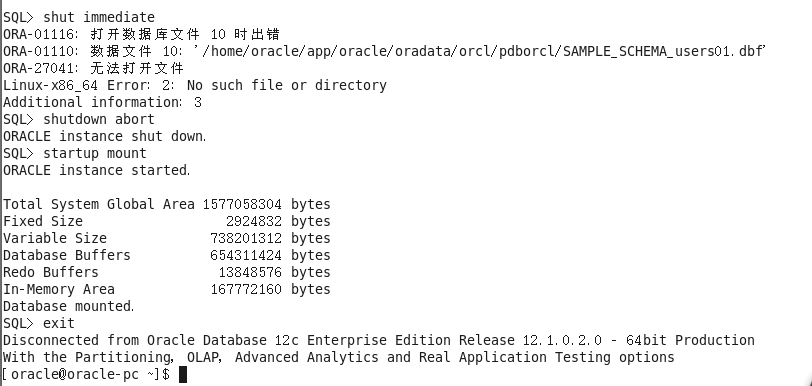


|  |
| --- |
| oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  SQL> insert into t1 values(2,'wang');  SQL> commit;  SQL> select \* from t1;  SQL> declare  2 n number;  3 begin  4 for n in 1..10000 loop  5 insert into t1 values(n,'name'||n);  6 end loop;  7 end;  SQL> select \* from t1;  SQL> exit |



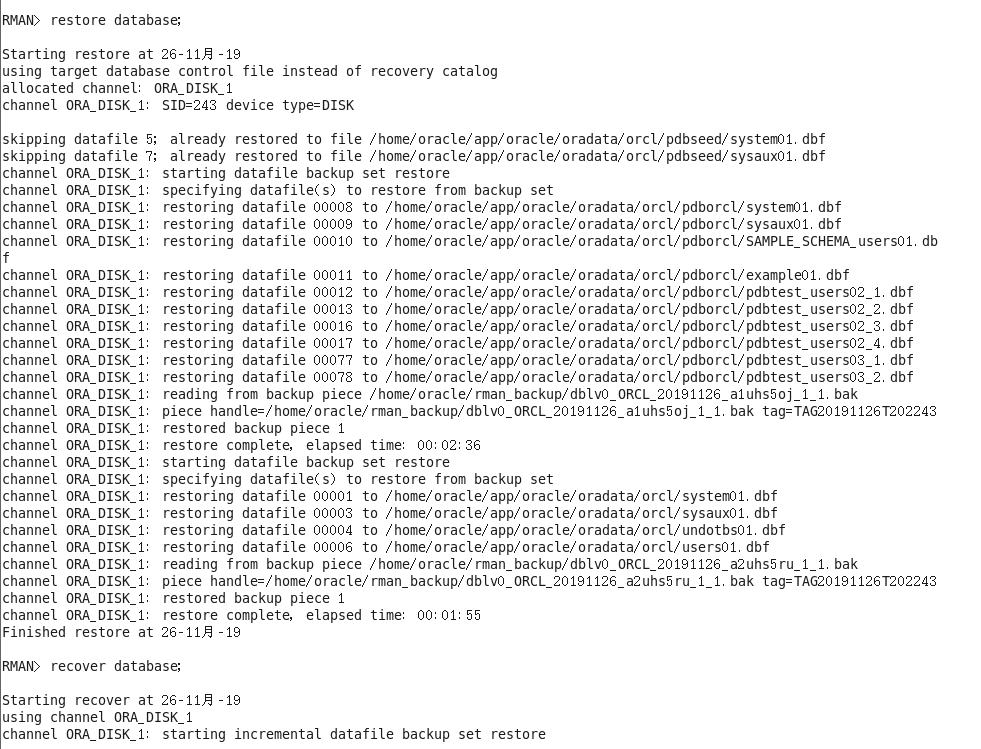
6) 数据库启动到mount状态

|  |
| --- |
| oracle@oracle-pc ~]$ sqlplus / as sysdba  SQL> shutdown immediate  SQL> shutdown abort  SQL> startup mount  SQL> exit |



1. 开始恢复数据库

|  |
| --- |
| oracle@oracle-pc ~]$ rman target /  RMAN> restore database ;  RMAN> recover database;  RMAN> alter database open;  Statement processed  RMAN> exit |



1. 查看数据库是否恢复

|  |
| --- |
| oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  SQL> select \* from t9; |

