

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的学生信息系统的数据库设计 | | |
| 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 | 冯苗 | 学号 | 201810414409 |
| 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

# orcale数据库实验六-成都大学学生信息系统

[orcale数据库实验六-成都大学信息系统](#header-n0)  
 [一. 概述](#header-n4)  
 [二. 需求与功能分析](#header-n7)   
 [三. 数据库设计](#header-n9)  
 [1. 添加用户及权限管理](#header-n10)  
 [创建student角色，并创建student\_lft用户，并且给用户分配角色空间](#header-n24)  
 [2. 通过新创建的用户student\_lft连接到 pdborcl](#header-n28)   
 [3. 创建teacher角色并创建teacher\_lft用户](#header-n30)  
 [4. 利用新创建的用户student\_lft创建了五个表](#header-n34)  
 [6. 创建视图](#header-n138)  
 [7. 将五个表的视图的SELECT对象权限授予teacher用户](#header-n151)   
 [8. 向数据库中写入数据共计50000多个数据](#header-n153)  
 [8.1 班级表](#header-n154)  
 [8.2 教师表](#header-n157)  
 [8.3 学科表](#header-n160)  
 [8.4 课程表](#header-n163)  
 [8.5 学生表](#header-n166)  
 [9. PL/SQL设计](#header-n169)  
 [9.1 存储过程：](#header-n190)  
 [9.1.1 查询班级人数](#header-n191)  
 [9.1.2. 查询学生分数](#header-n194)  
 [9.2. 创建函数](#header-n197)  
 [9.2.1. 查询所有科目总学分](#header-n198)  
 [10. 数据库备份](#header-n202)  
 [热备份和冷备份优缺点](#header-n205)  
 [热备份的优点是：](#header-n206)  
 [热备份的不足是：](#header-n212)  
 [冷备份的优点是：](#header-n216)  
 [冷备份不足是：](#header-n222)  
 [物理备份之冷备份：](#header-n227)  
 [10.1. 全数据库备份](#header-n236)  
 [10.2. 查询备份文件](#header-n241)  
 [10.3. 查看备份文件的内容](#header-n245)  
 [由上面的"list backup;" 输出可以看出，备份集中的文件内容是：](#header-n248)  
 [10.4. 备份后修改数据库](#header-n255)  
 [10.5. 删除数据库文件模拟数据损坏](#header-n259)  
 [10.6. 数据库完全恢复](#header-n268)  
 [10.6.1. 重启损坏的数据库到mount状态](#header-n269)  
 [10.6.2. 开始恢复数据库](#header-n274)  
 [10.6.3. 查询数据是否恢复](#header-n278)  
 [11. DataGuard实现数据库整体的异地备份](#header-n284)

## 一. 概述

学生信息管理系统是学校管理的重要工具，是学校不可或缺的一部分。随着在校人数的不断增加，教务系统的数量也不断的上涨。学校工作繁杂，资料众多，人工管理信息的难度也越来越大，显然是不能满足实际的需要，效率也是很低的。并且这种传统的方式存在着众多的弊端，如：保密性差.查询不便.效率低，很难维护和更新等，然而，本系统针对以上的缺点能够极大的提高学生信息管理的效率，也是科学化.正规化的管理，与世界接轨的重要条件。所以如何自动高效地管理信息是这些年来许多人所研究的。

随着这些年电脑计算机的速度质的提高，成本的下降，IT互联网大众趋势的发展。我们使用电脑的高效率才处理数据信息成为可能。学生学籍管理系统的出现，正是管理人员与信息数据，计算机的进入互动时代的体现。友好的人机交互模式，清晰简明的图形界面，高效安全的操作使得我们对成千上万的的信息的管理得心入手。通过这个系统，可以做到信息的规范处理，科学统计和快速的查询，从而减少管理方面的工作量。毋庸置疑，切实有效的把计算机管理引入学校教务管理中，对于促进学校管理制度，提高学校教学质量与办学水平有着显著意义。

## 二. 需求与功能分析

学生信息管理系统，可用于学校等机构的学生信息管理，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；用户登陆。提供了学生学籍信息的查询，添加，修改，删除；学生成绩的录入，修改，删除，查询班级排名，修改密码等功能。管理员管理拥有最高的权限。允许添加教师信息和课程信息等。其提供了简单.方便的操作。

## 三. 数据库设计

## 1. 添加用户及权限管理

oracle中的表就是一张存储数据的表。表空间是逻辑上的划分。方便管理的。

数据表空间 (Tablespace)

存放数据总是需要空间， Oracle把一个数据库按功能划分若干空间来保存数据。当然数据存放在磁盘最终是以文件形式，所以一盘一个数据表空间包含一个以上的物理文件  
数据表。

在仓库，我们可能有多间房子，每个房子又有多个货架，每架又有多层。 我们在数据库中存放数据，最终是数据表的单元来存储与管理的。  
数据文件。

以上几个概念都是逻辑上的， 而数据文件则是物理上的。就是说，数据文件是真正“看得着的东西”，它在磁盘上以一个真实的文件体现。

创建表空间：

格式: create tablespace 表间名 datafile '数据文件名' size 表空间大小  
 create tablespace data\_test datafile 'e:\oracle\oradata\test\data\_1.dbf' size 2000M;  
 create tablespace idx\_test datafile 'e:\oracle\oradata\test\idx\_1.dbf' size 2000M;  
 (\*数据文件名 包含全路径, 表空间大小 2000M 表是 2000兆)

建好tablespace, 就可以建用户

格式: create user 用户名 identified by 密码 default tablespace 表空间表;  
 create user study identified by study default tablespace data\_test;  
 (\*我们创建一个用户名为 study,密码为 study, 缺少表空间为 data\_test -这是在第二步建好的.)  
 (\*缺省表空间表示 用户study今后的数据如果没有专门指出，其数据就保存在 data\_test中, 也就是保存在对应的物理文件 e:\oracle\oradata\test\data\_1.dbf中)

创建用户并指定表空间

CREATE USER cici IDENTIFIED BY cici PROFILE DEFAULT DEFAULT TABLESPACE CICI ACCOUNT UNLOCK;  
create user jykl identified by jykl default tablespace jykl\_data temporary tablespace jykl\_temp;  
授权给新用户  
GRANT connect, resource TO cici;  
grant create session to cici;

授权给新用户

grant connect,resource to study;   
 --表示把 connect,resource权限授予study用户  
 grant dba to study;  
 --表示把 dba权限授予给 study

创建数据表

在上面，我们已建好了用户 study 我们现在进入该用户

sqlplusw study/study@test 然后就可以在用户study中创建数据表了

格式: create table 数据表名

### 创建student角色，并创建student\_lft用户，并且给用户分配角色空间

CREATE ROLE student;  
GRANT connect,resource,CREATE VIEW TO student;  
CREATE USER user\_lft IDENTIFIED BY 123 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;  
ALTER USER user\_lft QUOTA 50M ON users;  
GRANT student TO user\_lft;  
exit

创建结果



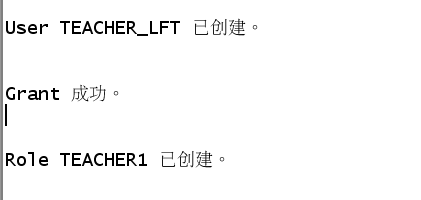
## 2. 通过新创建的用户student\_lft连接到 pdborcl



## 3. 创建teacher角色并创建teacher\_lft用户

CREATE ROLE teacher;  
CREATE USER teacher\_lft IDENTIFIED BY 123 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;  
GRANT teacher TO teacher\_lft;

创建结果



## 4. 利用新创建的用户student\_lft创建了五个表

CREATE TABLE "STUDENT\_LFT"."CLASS"   
 ( "ID" NUMBER,   
 "NAME" VARCHAR2(50 BYTE)  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;  
  
  
 CREATE TABLE "STUDENT\_LFT"."CURRICULUM"   
 ( "ID" NUMBER,   
 "SUBJECT\_ID" NUMBER,   
 "CLASS\_ID" NUMBER,   
 "TEACHER\_ID" NUMBER,   
 "TIME" DATE  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;  
   
 CREATE TABLE "STUDENT\_LFT"."MYCLASS"   
 ( "ID" NUMBER,   
 "NAME" VARCHAR2(50 BYTE)  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;  
   
 CREATE TABLE "STUDENT\_LFT"."STUDENT"   
 ( "ID" NUMBER,   
 "NAME" VARCHAR2(50 BYTE),   
 "CLASS\_ID" NUMBER,   
 "AGE" NUMBER(\*,0),   
 "SEX" VARCHAR2(50 BYTE)  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;  
   
 CREATE TABLE "STUDENT\_LFT"."SUBJECT"   
 ( "ID" NUMBER,   
 "NAME" VARCHAR2(50 BYTE),   
 "CREDIT" NUMBER(\*,0)  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;  
  
 CREATE TABLE "STUDENT\_LFT"."TEACHER"   
 ( "ID" NUMBER,   
 "NAME" VARCHAR2(50 BYTE),   
 "INFORMATION" VARCHAR2(50 BYTE)  
 ) SEGMENT CREATION DEFERRED   
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255   
 NOCOMPRESS LOGGING  
 TABLESPACE "USERS" ;

学生信息表

|  |  |  |
| --- | --- | --- |
| 属性 | 字段 | 注解 |
| 编号 | id | 学生编号 |
| 姓名 | name | 学生姓名 |
| 班级编号 | class\_id | 班级编号 |
| 年龄 | age | 学生年龄 |
| 性别 | sex | 学生性别 |

课程信息表

|  |  |  |
| --- | --- | --- |
| 属性 | 字段 | 注解 |
| 编号 | id | 课程编号 |
| 名称 | name | 课程姓名 |
| 学分 | credit | 课程学分 |

课表信息表

|  |  |  |
| --- | --- | --- |
| 属性 | 字段 | 注解 |
| 编号 | id | 课表编号 |
| 课程编号 | subject\_id | 课程编号 |
| 班级编号 | class\_id | 班级编号 |
| 教师编号 | teacher\_id | 教师编号 |
| 时间 | time | 上课时间 |

班级信息表

|  |  |  |
| --- | --- | --- |
| 属性 | 字段 | 注解 |
| 编号 | id | 班级编号 |
| 名称 | name | 班级名称 |

教师信息表

|  |  |  |
| --- | --- | --- |
| 属性 | 字段 | 注解 |
| 编号 | id | 教师编号 |
| 名称 | name | 教师姓名 |
| 信息 | information | 教师信息介绍 |

## 6. 创建视图

* 视图(view)，也称虚表, 不占用物理空间，这个也是相对概念，因为视图本身的定义语句还是要存储在数据字典里的。视图只有逻辑定义。每次使用的时候,只是重新执行SQL。
* 视图是从一个或多个实际表中获得的，这些表的数据存放在数据库中。那些用于产生视图的表叫做该视图的基表。一个视图也可以从另一个视图中产生。
* 视图的定义存在数据库中，与此定义相关的数据并没有再存一份于数据库中。通过视图看到的数据存放在基表中。
* 视图看上去非常象数据库的物理表，对它的操作同任何其它的表一样。当通过视图修改数据时，实际上是在改变基表中的数据；相反地，基表数据的改变也会自动反映在由基表产生的视图中。由于逻辑上的原因，有些Oracle视图可以修改对应的基表，有些则不能（仅仅能查询）。
* 还有一种视图：物化视图（MATERIALIZED VIEW ），也称实体化视图，快照 （8i 以前的说法） ，它是含有数据的，占用存储空间。

CREATE OR REPLACE FORCE EDITIONABLE VIEW "STUDENT\_LFT"."MYVIEW\_CURRICULUM" ("ID") AS   
 SELECT id FROM curriculum;  
   
CREATE OR REPLACE FORCE EDITIONABLE VIEW "STUDENT\_LFT"."MYVIEW\_MYCLASS" ("ID") AS   
 SELECT name FROM myclass;  
   
CREATE OR REPLACE FORCE EDITIONABLE VIEW "STUDENT\_LFT"."MYVIEW\_STUDENT" ("ID") AS   
 SELECT name FROM student;  
   
 CREATE OR REPLACE FORCE EDITIONABLE VIEW "STUDENT\_LFT"."MYVIEW\_SUBJECT" ("ID") AS   
 SELECT name FROM subject;  
   
 CREATE OR REPLACE FORCE EDITIONABLE VIEW "STUDENT\_LFT"."MYVIEW\_TEACHER" ("ID") AS   
 SELECT name FROM teacher;

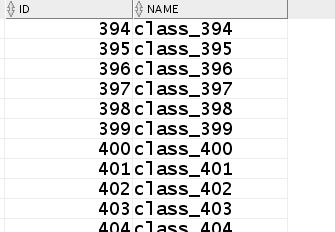
## 7. 将五个表的视图的SELECT对象权限授予teacher用户

GRANT SELECT ON myview\_MYCLASS TO teacher\_lft;  
GRANT SELECT ON myview\_SURRICULUM TO teacher\_lft;  
GRANT SELECT ON myview\_STUDENT TO teacher\_lft;  
GRANT SELECT ON myview\_SUBJECT TO teacher\_lft;  
GRANT SELECT ON myview\_TEACHER TO teacher\_lft;

## 8. 向数据库中写入数据共计50000多个数据

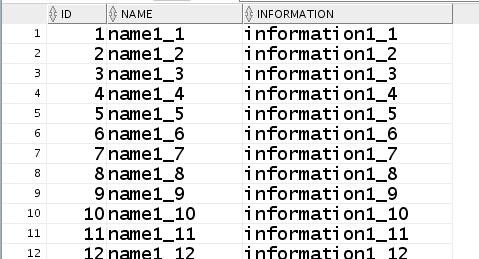
### 8.1 班级表

-- 班级表  
DECLARE  
 maxnumber CONSTANT NUMBER := 100;  
 i NUMBER := 1;  
BEGIN  
 FOR i IN 1..maxnumber LOOP  
 INSERT INTO myclass ( id,name ) VALUES (  
 i,  
 concat('class\_',i)  
 ); --CONCAT('test',i)是将test与i进行拼接  
 END LOOP;  
  
 dbms\_output.put\_line(' 成功录入数据！ ');  
 COMMIT;  
END;



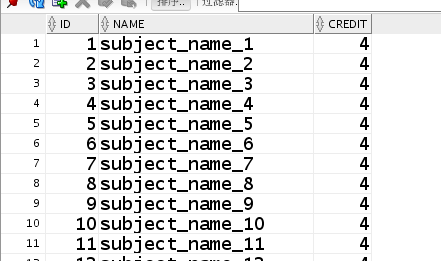
### 8.2 教师表

--教师表  
DECLARE  
 maxnumber CONSTANT NUMBER := 100;  
 i NUMBER := 1;  
BEGIN  
 FOR i IN 1..maxnumber LOOP  
 INSERT INTO teacher ( id,name,information ) VALUES (  
 i,  
 concat('name1\_',i),  
 concat('information1\_',i)  
 ); --CONCAT('test',i)是将test与i进行拼接  
  
 END LOOP;  
  
 dbms\_output.put\_line(' 成功录入数据！ ');  
 COMMIT;  
END;



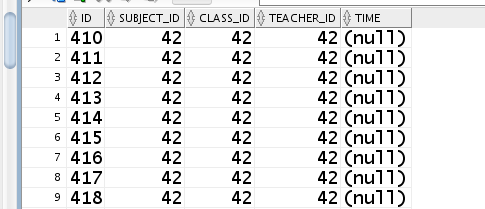
### 8.3 学科表

-- 学科表  
DECLARE  
 maxnumber CONSTANT NUMBER := 100;  
 i NUMBER := 1;  
BEGIN  
 FOR i IN 1..maxnumber LOOP  
 INSERT INTO subject ( id,name,credit ) VALUES (  
 i,  
 concat('subject\_name\_',i),  
 4  
 ); --CONCAT('test',i)是将test与i进行拼接  
  
 END LOOP;  
  
 dbms\_output.put\_line(' 成功录入数据！ ');  
 COMMIT;  
END;



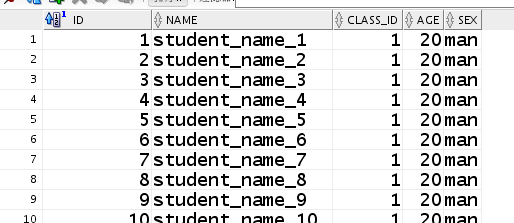
### 8.4 课程表

--课程表  
DECLARE  
 maxnumber CONSTANT NUMBER := 1000;  
 i NUMBER := 1;  
 my\_subject\_id NUMBER := 1;  
 my\_class\_id NUMBER := 1;  
 my\_teacher\_id NUMBER := 1;  
BEGIN  
 FOR i IN 1..maxnumber LOOP  
 if mod(i,10)=0 then my\_class\_id:=my\_class\_id+1;  
 end if;  
 if mod(i,10)=0 then my\_subject\_id:=my\_subject\_id+1;  
 end if;  
 if mod(i,10)=0 then my\_teacher\_id:=my\_teacher\_id+1;  
 end if;  
 INSERT INTO curriculum ( id,subject\_id,class\_id,teacher\_id ) VALUES (  
 i,  
 my\_subject\_id,  
 my\_class\_id,  
 my\_teacher\_id  
 ); --CONCAT('test',i)是将test与i进行拼接  
  
 END LOOP;  
   
 dbms\_output.put\_line(' 成功录入数据！ ');  
 COMMIT;  
END;



### 8.5 学生表

--学生表  
DECLARE  
 maxnumber CONSTANT NUMBER := 50000;  
 i NUMBER := 1;  
 my\_class\_id NUMBER := 1;  
BEGIN  
 FOR i IN 1..maxnumber LOOP  
 if mod(i,500)=0 then my\_class\_id:=my\_class\_id+1;  
 end if;  
 INSERT INTO student ( id,name,class\_id,age,sex ) VALUES (  
 i,  
 concat('student\_name\_',i),  
 my\_class\_id,  
 20,  
 'man'  
 ); --CONCAT('test',i)是将test与i进行拼接  
  
 END LOOP;  
  
 dbms\_output.put\_line(' 成功录入数据！ ');  
 COMMIT;  
END;



## 9. PL/SQL设计

过程和函数由以下4部分：

* 签名或头
* 关键字IS或AS
* 局部声明（可选）
* BEGIN和END之间的过程体（包括异常处理程序）

简单示例：

create or replace procedure show\_line(ip\_line\_length in number, ip\_separator in varchar2)  
is   
actual\_line varchar2(150);  
begin  
 insert into t\_user(id,name,sex)values(ip\_line\_length,ip\_separator,ip\_line\_length);  
 for idx in 1..ip\_line\_length loop  
 actual\_line := actual\_line||ip\_separator;  
 end loop;  
 dbms\_output.put\_line(actual\_line);  
exception when others then  
 dbms\_output.put\_line(SQLERRM);  
end;

如下调用：

begin show\_line(50,'=');end;/

在SQLPLUS里面调用：

SQL> BEGIN2 show\_line(50,'=');3 END;

几点说明：

1、参数没有指定长度，当有实际数据传递进来的时候，参数的长度才被确定。

2、局部声明为：actual\_line varchar2(150);

3、使用命令SQL> show errors在SQLPLUS里面查看错误。

### 9.1 存储过程：

#### 9.1.1 查询班级人数

create or replace PACKAGE MyPack IS  
  
 PROCEDURE numberOfQueries(my\_class\_id NUMBER);  
 END MyPack;  
 /  
 create or replace PACKAGE BODY MyPack IS  
 PROCEDURE numberOfQueries(my\_class\_id NUMBER)  
 AS  
 lft integer;   
 BEGIN  
 select id into lft  
 from student  
 where class\_id=my\_class\_id;  
 COMMIT;  
 END numberOfQueries;  
 END MyPack;  
  
 set serveroutput on  
 DECLARE  
 my\_class\_id NUMBER;   
 BEGIN  
 my\_class\_id := 1;  
 MYPACK.numberOfQueries (my\_class\_id) ;   
 END;

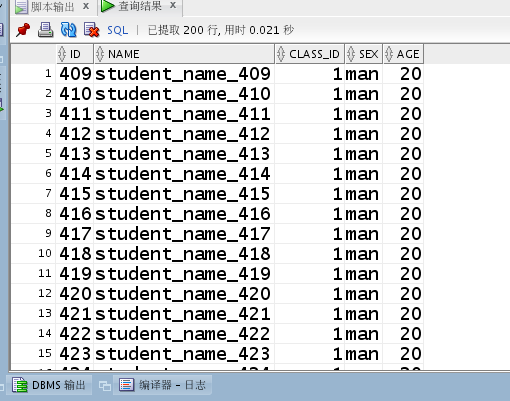


image-20191126201711814

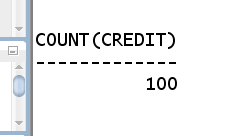
#### 9.1.2. 查询学生分数

create or replace procedure p\_contract\_purchase\_import(  
 V\_IN\_SUBCOMPANYID in VARCHAR2,  
 V\_IN\_PURCONTRACTMONEY in NUMBER,   
 V\_IN\_PARTYBNAME in VARCHAR2,   
 v\_o\_ret out number )  
   
 as  
 V\_SUPPLIERID INTEGER;   
 V\_PARTYBACCOUNT VARCHAR2(100);  
 V\_SQLERRM VARCHAR2(4000);  
   
 begin  
   
 v\_o\_ret := 1;   
 V\_SUPPLIERID := '';  
 V\_PARTYBACCOUNT := '';  
   
 if(V\_IN\_PARTYBNAME is not null) then  
 begin  
 select t.SUPPLIERID,t.PARTYBACCOUNT,t.PARTYBBANK ,t.PARTYBNAME   
 into V\_SUPPLIERID,V\_PARTYBACCOUNT,V\_PARTYBBANK,V\_PARTYBNAME  
 from T\_SUPPLIER t where t.PARTYBNAME=trim(V\_IN\_PARTYBNAME) and t.SUBCOMPANYID=trim(V\_IN\_SUBCOMPANYID);  
   
 exception  
 when others then  
 v\_o\_ret := 4 ;   
 V\_PARTYBNAME := V\_IN\_PARTYBNAME;  
   
 V\_SQLERRM := SQLERRM;  
 INSERT INTO T\_LOG\_DBERR  
 (ERRTIME, ERRMODEL, ERRDESC)  
 VALUES  
 (SYSDATE,  
 'PROCEDURES',  
 'p\_contract\_purchase\_import:ret=' || v\_o\_ret ||','||  
 V\_SQLERRM);  
 COMMIT;  
 end ;  
 end if;  
   
 end ;   
 commit;  
 v\_o\_ret :=0 ;  
 return;  
   
 EXCEPTION  
 WHEN OTHERS THEN  
 ROLLBACK;  
 V\_SQLERRM := SQLERRM;  
 INSERT INTO T\_LOG\_DBERR  
 (ERRTIME, ERRMODEL, ERRDESC)  
 VALUES  
 (SYSDATE,  
 'PROCEDURES',  
 'p\_contract\_purchase\_import:ret=' || v\_o\_ret ||','||  
 V\_SQLERRM);  
 COMMIT;  
   
 end p\_contract\_purchase\_import;

### 9.2. 创建函数

#### 9.2.1. 查询所有科目总学分

create or replace PACKAGE MyPack IS  
   
 FUNCTION Get\_AllCredit(V\_DEPARTMENT\_ID NUMBER) RETURN NUMBER;  
   
 END MyPack;  
   
 FUNCTION Get\_AllCredit(V\_DEPARTMENT\_ID NUMBER) RETURN NUMBER  
 AS  
 N NUMBER(20,2);  
 BEGIN  
 SELECT SUM(subject.credit) into N FROM subject;  
 RETURN N;  
 END;  
 END MyPack;



## 10. 数据库备份

**ORACLE数据库备份与恢复详解**

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。

### 热备份和冷备份优缺点

#### 热备份的优点是：

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。

#### 热备份的不足是：

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”。

#### 冷备份的优点是：

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。

#### 冷备份不足是：

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。

### 物理备份之冷备份：

当数据库可以暂时处于关闭状态时，我们需要将它在这一稳定时刻的数据相关文件转移到安全的区域，当数据库遭到破坏，再从安全区域将备份的数据库相关文件拷 贝回原来的位置，这样，就完成了一次快捷安全等数据转移。由于是在数据库不提供服务的关闭状态，所以称为冷备份。冷备份具有很多优良特性，比如上面图中我 们提到的，快速，方便，以及高效。一次完整的冷备份步骤应该是：

1，首先关闭数据库（shutdown normal）

2，拷贝相关文件到安全区域（利用操作系统命令拷贝数据库的所有的数据文件、日志文件、控制文件、参数文件、口令文件等（包括路径））

3，重新启动数据库（startup）

以上的步骤我们可以用一个脚本来完成操作：

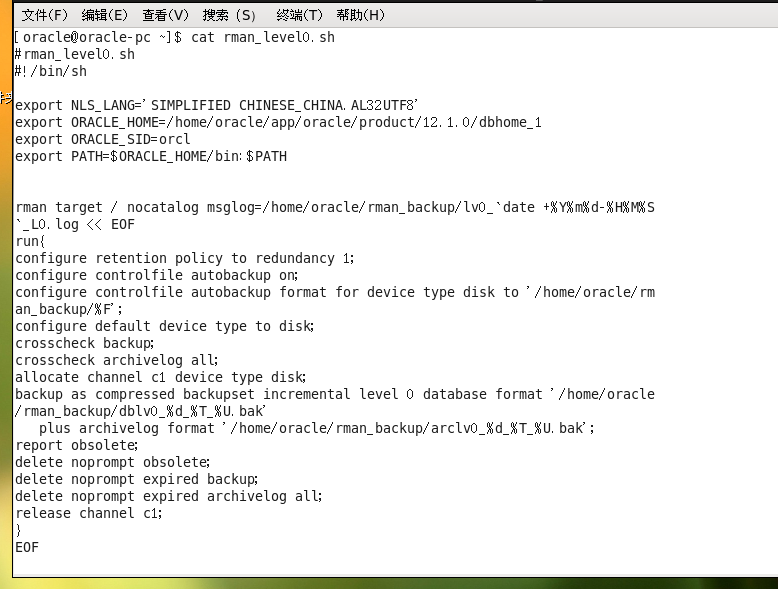
su – oracle < sqlplus /nolog   
 connect / as sysdba  
 shutdown immediate;  
 !cp 文件 备份位置（所有的日志、数据、控制及参数文件）;  
 startup;  
 exit;

这样，我们就完成了一次冷备份，请确定你对这些相应的目录（包括写入的目标文件夹）有相应的权限。

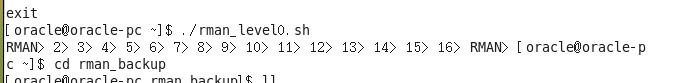
恢复的时候，相对比较简单了，我们停掉数据库，将文件拷贝回相应位置，重启数据库就可以了，当然也可以用脚本来完成。

### 10.1. 全数据库备份

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



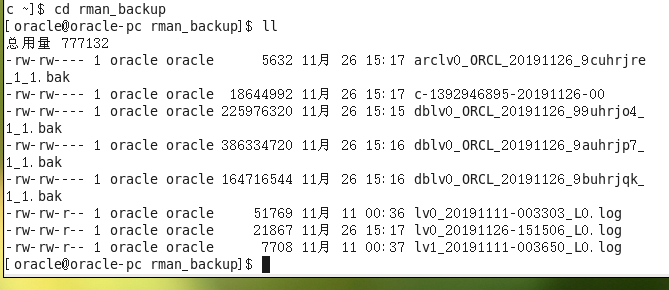
img



img

### 10.2. 查询备份文件

[oracle@oracle-pc ~]$ cd rman\_backup  
   
 c ~]$ cd rman\_backup  
 [oracle@oracle-pc rman\_backup]$ ll  
 总用量 777132  
 -rw-rw---- 1 oracle oracle 5632 11月 26 15:17 arclv0\_ORCL\_20191126\_9cuhrjre\_1\_1.bak  
 -rw-rw---- 1 oracle oracle 18644992 11月 26 15:17 c-1392946895-20191126-00  
 -rw-rw---- 1 oracle oracle 225976320 11月 26 15:15 dblv0\_ORCL\_20191126\_99uhrjo4\_1\_1.bak  
 -rw-rw---- 1 oracle oracle 386334720 11月 26 15:16 dblv0\_ORCL\_20191126\_9auhrjp7\_1\_1.bak  
 -rw-rw---- 1 oracle oracle 164716544 11月 26 15:16 dblv0\_ORCL\_20191126\_9buhrjqk\_1\_1.bak  
 -rw-rw-r-- 1 oracle oracle 51769 11月 11 00:36 lv0\_20191111-003303\_L0.log  
 -rw-rw-r-- 1 oracle oracle 21867 11月 26 15:17 lv0\_20191126-151506\_L0.log  
 -rw-rw-r-- 1 oracle oracle 7708 11月 11 00:37 lv1\_20191111-003650\_L0.log



img

### 10.3. 查看备份文件的内容

[oracle@oracle-pc rman\_backup]$ rman target /   
   
 Recovery Manager: Release 12.1.0.2.0 - Production on 星期二 11月 26 15:39:01 2019   
   
 Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.   
   
 connected to target database: ORCL (DBID=1392946895)   
   
 RMAN> list backup;   
   
 using target database control file instead of recovery catalog   
   
 List of Backup Sets   
 ===================   
   
   
 BS Key Type LV Size Device Type Elapsed Time Completion Time   
 ------- ---- -- ---------- ----------- ------------ ---------------   
 249 Incr 0 215.50M DISK 00:00:32 26-11月-19   
 BP Key: 249 Status: AVAILABLE Compressed: YES Tag: TAG20191126T151516   
 Piece Name: /home/oracle/rman\_backup/dblv0\_ORCL\_20191126\_99uhrjo4\_1\_1.bak   
 List of Datafiles in backup set 249   
 Container ID: 3, PDB Name: PDBORCL   
 File LV Type Ckp SCN Ckp Time Name   
 ---- -- ---- ---------- ---------- ----   
 8 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/system01.dbf   
 9 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/sysaux01.dbf   
 10 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/SAMPLE\_SCHEMA\_users01.dbf   
 11 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/example01.dbf   
 12 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_1.dbf   
 13 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_2.dbf   
 16 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_3.dbf   
 17 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_4.dbf   
 77 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users03\_1.dbf   
 78 0 Incr 47696807 26-11月-19 /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users03\_2.dbf   
   
 BS Key Type LV Size Device Type Elapsed Time Completion Time   
 ------- ---- -- ---------- ----------- ------------ ---------------   
 250 Incr 0 368.43M DISK 00:00:43 26-11月-19   
 BP Key: 250 Status: AVAILABLE Compressed: YES Tag: TAG20191126T151516   
 Piece Name: /home/oracle/rman\_backup/dblv0\_ORCL\_20191126\_9auhrjp7\_1\_1.bak   
 List of Datafiles in backup set 250   
 File LV Type Ckp SCN Ckp Time Name   
 ---- -- ---- ---------- ---------- ----   
 1 0 Incr 47696822 26-11月-19 /home/oracle/app/oracle/oradata/orcl/system01.dbf   
 3 0 Incr 47696822 26-11月-19 /home/oracle/app/oracle/oradata/orcl/sysaux01.dbf   
 4 0 Incr 47696822 26-11月-19 /home/oracle/app/oracle/oradata/orcl/undotbs01.dbf   
 6 0 Incr 47696822 26-11月-19 /home/oracle/app/oracle/oradata/orcl/users01.dbf   
   
 BS Key Type LV Size Device Type Elapsed Time Completion Time   
 ------- ---- -- ---------- ----------- ------------ ---------------   
 251 Incr 0 157.08M DISK 00:00:18 26-11月-19   
 BP Key: 251 Status: AVAILABLE Compressed: YES Tag: TAG20191126T151516   
 Piece Name: /home/oracle/rman\_backup/dblv0\_ORCL\_20191126\_9buhrjqk\_1\_1.bak   
 List of Datafiles in backup set 251   
 Container ID: 2, PDB Name: PDB$SEED   
 File LV Type Ckp SCN Ckp Time Name   
 ---- -- ---- ---------- ---------- ----   
 5 0 Incr 1819408 01-12月-14 /home/oracle/app/oracle/oradata/orcl/pdbseed/system01.dbf   
 7 0 Incr 1819408 01-12月-14 /home/oracle/app/oracle/oradata/orcl/pdbseed/sysaux01.dbf   
   
 BS Key Size Device Type Elapsed Time Completion Time   
 ------- ---------- ----------- ------------ ---------------   
 252 5.00K DISK 00:00:00 26-11月-19   
 BP Key: 252 Status: AVAILABLE Compressed: YES Tag: TAG20191126T151702   
 Piece Name: /home/oracle/rman\_backup/arclv0\_ORCL\_20191126\_9cuhrjre\_1\_1.bak   
   
 List of Archived Logs in backup set 252   
 Thrd Seq Low SCN Low Time Next SCN Next Time   
 ---- ------- ---------- ---------- ---------- ---------   
 1 1614 47696796 26-11月-19 47696850 26-11月-19   
   
 BS Key Type LV Size Device Type Elapsed Time Completion Time   
 ------- ---- -- ---------- ----------- ------------ ---------------   
 253 Full 17.77M DISK 00:00:00 26-11月-19   
 BP Key: 253 Status: AVAILABLE Compressed: NO Tag: TAG20191126T151703   
 Piece Name: /home/oracle/rman\_backup/c-1392946895-20191126-00   
 SPFILE Included: Modification time: 26-11月-19   
 SPFILE db\_uniqe\_name: ORCL   
 Control File Included: Ckp SCN: 47696860 Ckp time: 26-11月-19

### 由上面的"list backup;" 输出可以看出，备份集中的文件内容是：

• /home/oracle/rman*backup/dblv0*ORCL*20191120*d7uhb2ap*1*1.bak 是插接数据库PDBORCL的备份集

• /home/oracle/rman*backup/dblv0*ORCL*20191120*d8uhb2c6*1*1.bak 是ORCL的备份集

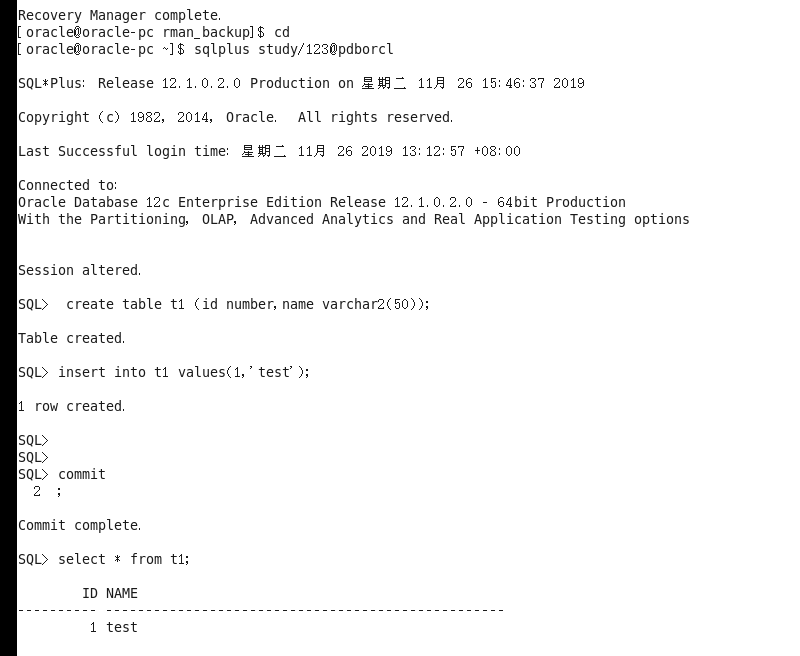
• /home/oracle/rman*backup/dblv0*ORCL*20191120*d9uhb2ei*1*1.bak是PDB$SEED的备份集

• /home/oracle/rman*backup/arclv0*ORCL*20191120*dauhb2fm*1*1.bak是归档文件的备份集

• /home/oracle/rman\_backup/c-1392946895-20191120-01是参数文件(SPFILE)和控制文件(Control File)的备份集

### 10.4. 备份后修改数据库

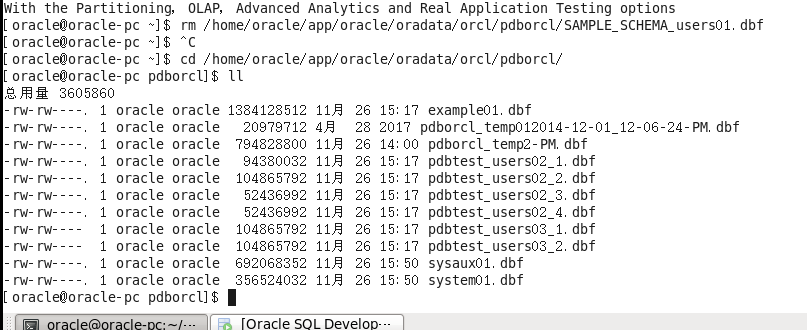
[oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  
 SQL> create table t1 (id number,name varchar2(50));  
 Table created.  
 SQL> insert into t1 values(1,'zhang');  
 1 row created.  
 SQL> commit;  
 Commit complete.  
 SQL> select \* from t1;  
   
 ID NAME  
 ---------- --------------------------------------------------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



img

### 10.5. 删除数据库文件模拟数据损坏

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



img

删除数据库文件后修改数据

删除数据文件后，仍然可以增加一条数据。这是因为增加的数据并没有写入数据文件，而是写到了日志文件中。如果增加的数据较多的时候，就会出问题了。

[oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  
 SQL> insert into t1 values(2,'wang');  
 1 row created.  
 SQL> commit;  
 Commit complete.  
 SQL> select \* from t1;  
 ID NAME  
 ---------- --------------------------------------------------  
 1 zhang  
 2 wang  
 SQL>



img

### 10.6. 数据库完全恢复

#### 10.6.1. 重启损坏的数据库到mount状态

通过shutdown immediate无法正常关闭数据库，只能通过shutdown abort强制关闭。然后将数据库启动到mount状态。

[oracle@oracle-pc ~]$ sqlplus / as sysdba  
 SQL> shutdown immediate  
 ORA-01116: 打开数据库文件 10 时出错  
 ORA-01110: 数据文件 10: '/home/oracle/app/oracle/oradata/orcl/pdborcl/SAMPLE\_SCHEMA\_users01.dbf'  
 ORA-27041: 无法打开文件  
 Linux-x86\_64 Error: 2: No such file or directory  
 Additional information: 3  
 SQL> shutdown abort  
 ORACLE instance shut down.  
 SQL> startup mount  
 ORACLE instance started.  
   
 Total System Global Area 1577058304 bytes  
 Fixed Size 2924832 bytes  
 Variable Size 738201312 bytes  
 Database Buffers 654311424 bytes  
 Redo Buffers 13848576 bytes  
 In-Memory Area 167772160 bytes  
 Database mounted.



img

#### 10.6.2. 开始恢复数据库

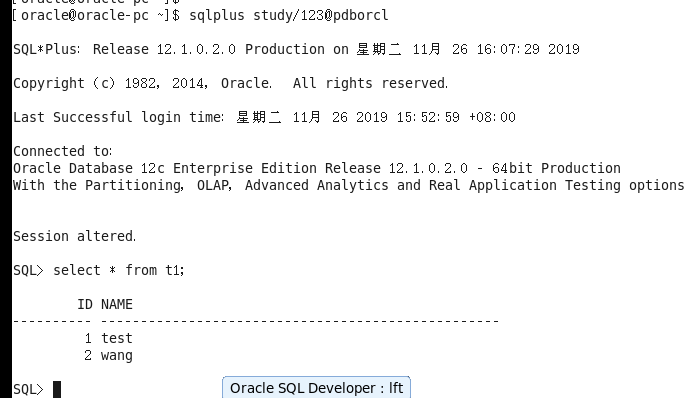
[oracle@oracle-pc ~]$ rman target /  
 RMAN> restore database ;  
 Starting restore at 20-11月-19  
 using target database control file instead of recovery catalog  
 allocated channel: ORA\_DISK\_1  
 channel ORA\_DISK\_1: SID=243 device type=DISK  
   
 skipping datafile 5; already restored to file /home/oracle/app/oracle/oradata/orcl/pdbseed/system01.dbf  
 skipping datafile 7; already restored to file /home/oracle/app/oracle/oradata/orcl/pdbseed/sysaux01.dbf  
 channel ORA\_DISK\_1: starting datafile backup set restore  
 channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set  
 channel ORA\_DISK\_1: restoring datafile 00008 to /home/oracle/app/oracle/oradata/orcl/pdborcl/system01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00009 to /home/oracle/app/oracle/oradata/orcl/pdborcl/sysaux01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00010 to /home/oracle/app/oracle/oradata/orcl/pdborcl/SAMPLE\_SCHEMA\_users01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00011 to /home/oracle/app/oracle/oradata/orcl/pdborcl/example01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00012 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_1.dbf  
 channel ORA\_DISK\_1: restoring datafile 00013 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_2.dbf  
 channel ORA\_DISK\_1: restoring datafile 00016 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_3.dbf  
 channel ORA\_DISK\_1: restoring datafile 00017 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_4.dbf  
 channel ORA\_DISK\_1: restoring datafile 00077 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users03\_1.dbf  
 channel ORA\_DISK\_1: restoring datafile 00078 to /home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users03\_2.dbf  
 channel ORA\_DISK\_1: reading from backup piece /home/oracle/rman\_backup/dblv0\_ORCL\_20191120\_d7uhb2ap\_1\_1.bak  
 channel ORA\_DISK\_1: piece handle=/home/oracle/rman\_backup/dblv0\_ORCL\_20191120\_d7uhb2ap\_1\_1.bak tag=TAG20191120T083953  
 channel ORA\_DISK\_1: restored backup piece 1  
 channel ORA\_DISK\_1: restore complete, elapsed time: 00:01:41  
 channel ORA\_DISK\_1: starting datafile backup set restore  
 channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set  
 channel ORA\_DISK\_1: restoring datafile 00001 to /home/oracle/app/oracle/oradata/orcl/system01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00003 to /home/oracle/app/oracle/oradata/orcl/sysaux01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00004 to /home/oracle/app/oracle/oradata/orcl/undotbs01.dbf  
 channel ORA\_DISK\_1: restoring datafile 00006 to /home/oracle/app/oracle/oradata/orcl/users01.dbf  
 channel ORA\_DISK\_1: reading from backup piece /home/oracle/rman\_backup/dblv0\_ORCL\_20191120\_d8uhb2c6\_1\_1.bak  
 channel ORA\_DISK\_1: piece handle=/home/oracle/rman\_backup/dblv0\_ORCL\_20191120\_d8uhb2c6\_1\_1.bak tag=TAG20191120T083953  
 channel ORA\_DISK\_1: restored backup piece 1  
 channel ORA\_DISK\_1: restore complete, elapsed time: 00:01:55  
 Finished restore at 20-11月-19  
   
 RMAN> recover database;  
 RMAN> alter database open;  
 Statement processed  
 RMAN> exit



img

#### 10.6.3. 查询数据是否恢复

[oracle@oracle-pc ~]$ sqlplus study/123@pdborcl  
 SQL> select \* from t1;  
 ID NAME  
 ---------- --------------------------------------------------  
 1 zhang  
 2 wang  
 SQL>



img

由以上查询结果可见，数据100%恢复了!

## 11. DataGuard实现数据库整体的异地备份

Data Guard 是Oracle的集成化灾难恢复解决方案，该技术可以维护生产数据库一个或多个同步备份，由一个主数据库和多个备用数据库组成，并形成一个独立的、易于管理的数据保护方案。  
Data Guard 备用数据库可以与主系统位于相同的数据中心，也可以是在地理位置上分布较远的远程灾难备份中心。

Data Guard 的基本原理是，当某次事务处理对生产数据库中的数据做出更改时，Oracle 数据库将在一个联机重做日志 文件中记录此次更改。在DataGuard中可以配置写日志的这个过程，除了把日志记录到本地的联机日志文件和归档日志文件中，还可以通过网络，把日志信息发送的远程的备用数据库服务器上。这个备用日志文件写入过程可以是实时，同步的，以实现零数据丢失(最大保护模式);也可以是异步的，以减少对网络带宽的压力(最大可用性模式);或者是通过归档日志文件、一个日志文件的批量传输模式，以减少对生产系统的性能影响(最大性能模式)。当备份数据库接收到日志信息后，Data Guard 可以自动利用日志信息实现数据的同步。当主数据库打开并处于活动状态时，备用数据库可以执行恢复操作，如果主数据库出现了故障，备用数据库即可以被激活并接管生产数据库的工作。

安装配置Data Guard的步骤如下:

(1)将主数据库设置成归档状态和自动归档模式

(3)在主数据库 上创建备用数据库控制文件、初始化参数文件:

(4)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;

(5)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;

(6) 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;

(7)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;

(8)启动归档到备用数据库。

四、Data Guard在我院HIS中的应用

(1)容灾备份: 我院在住院大楼的机房新建了-个异地备份机房，硬件采用一台IBM X3650服务器， 服务器上配置5块146G硬盘并做了RAID5,安装配置了Oracle 10g 和Data Guard. 保证了无论是硬盘或网络出现故障的情况下全院信息系统能正常运行。同时对于火灾、盗窃等非技术情况下的异地备份，提高了整体系统完全性。经过多次测试，主数据库和备份数据库之间能够在5分钟之类切换，基本上能保证系统的连续运行。

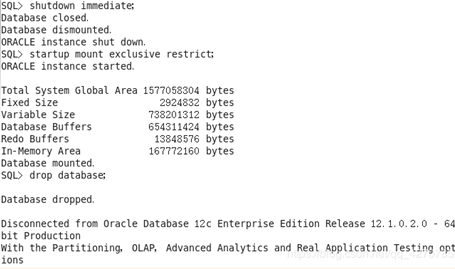
Data Guard作为Oracle免费推出的一个灾难恢复技术，利用Oracle 10g Data Guard而实现备用数据库操作的好处可以简单地总结为:它提供了灾难保护并防止数据丢失、维护主数据库的几个事务一致的副本、 防止灾难、数据损坏和用户错误、无需昂贵且复杂的HW/SW镜像。但是Data Guard也有以下缺点:不支持异构、不可跨平台、数据库版本需一致， 目标数据库出入rcovery状态不可用等，如果用户需要更高性能及可靠性可以购买第三方软件，例如DSG Real Synce等产品。

第一步：备库

mkdir -p /home/oracle/app/oracle/diag/orcl   
mkdir -p /home/oracle/app/oracle/oradata/stdorcl/   
mkdir -p /home/oracle/app/oracle/oradata/stdorcl/pdborcl   
mkdir -p /home/oracle/arch   
mkdir -p /home/oracle/rman   
mkdir -p /home/oracle/app/oracle/oradata/stdorcl/pdbseed/   
mkdir -p /home/oracle/app/oracle/oradata/stdorcl/pdb/   
   
//linux 命令，一行一行在终端运行   
//删除原有数据库:   
$sqlplus / as sysdba //连数据库   
shutdown immediate; //sql>   
startup mount exclusive restrict; //sql>   
drop database; //sql>   
exit; //sql>   
启动到nomount   
$sqlplus / as sysdba //连数据库   
startup nomount //sql>   
sql>   
//换到primary



image-20191126204925927



第二步：主库:

$sqlplus / sysdba //sql>  
select group#,thread#,members,status from v$log; //sql>  
  
alter database add standby logfile group 5 '/home/oracle/app/oracle/oradata/orcl/stdredo1.log' size 50m; //sql>  
alter database add standby logfile group 6 '/home/oracle/app/oracle/oradata/orcl/stdredo2.log' size 50m; //sql>  
alter database add standby logfile group 7 '/home/oracle/app/oracle/oradata/orcl/stdredo3.log' size 50m; //sql>   
alter database add standby logfile group 8 '/home/oracle/app/oracle/oradata/orcl/stdredo4.log' size 50m; //sql>

主库环境开启强制归档

ALTER DATABASE FORCE LOGGING; //sql>  
  
alter system set LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(orcl,stdorcl)' scope=both sid='\*'; //sql>  
alter system set log\_archive\_dest\_1='LOCATION=/home/oracle/arch VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=orcl' scope=spfile;  
alter system set LOG\_ARCHIVE\_DEST\_2='SERVICE=stdorcl LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=stdorcl' scope=both sid='\*';  
alter system set fal\_client='orcl' scope=both sid='\*';   
alter system set FAL\_SERVER='stdorcl' scope=both sid='\*';   
alter system set standby\_file\_management=AUTO scope=both sid='\*';  
alter system set DB\_FILE\_NAME\_CONVERT='/home/oracle/app/oracle/oradata/stdorcl/','/home/oracle/app/oracle/oradata/orcl/' scope=spfile sid='\*';   
alter system set LOG\_FILE\_NAME\_CONVERT='/home/oracle/app/oracle/oradata/stdorcl/','/home/oracle/app/oracle/oradata/orcl/' scope=spfile sid='\*';  
alter system set log\_archive\_format='%t\_%s\_%r.arc' scope=spfile sid='\*';  
alter system set remote\_login\_passwordfile='EXCLUSIVE' scope=spfile;  
alter system set PARALLEL\_EXECUTION\_MESSAGE\_SIZE=8192 scope=spfile;  
  
exit;

编辑主库以及备库的/home/oracle/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora文件

$gedit /home/oracle/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora //terminal  
  
ORCL =  
 (DESCRIPTION =  
 (ADDRESS\_LIST =  
 (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.206.131)(PORT = 1521)) //\*\*  
 )  
 (CONNECT\_DATA =  
 (SERVER = DEDICATED)  
 (SERVICE\_NAME = orcl)  
 )  
 )  
  
stdorcl =  
 (DESCRIPTION =  
 (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.206.132)(PORT = 1521)) //\*\*  
 (CONNECT\_DATA =  
 (SERVER = DEDICATED)  
 (SID = orcl)  
 )  
 )  
  
 //ifconfig 查询ip后 测试两能否平ping通 只改ip地址 最后一个块被替换为以上两个块 记得保存（主+备）

在主库上生成备库的参数文件:

$sqlplus / as sysdba //如果没有出现sql>  
SQL>create pfile from spfile;  
生成/home/oracle/app/oracle/product/12.1.0/dbhome\_1/dbs/initorcl.ora  
exit;

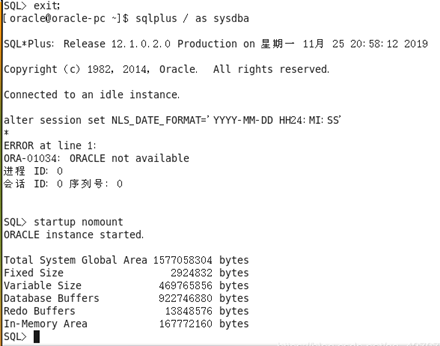


image-20191126205058301

将主库的参数文件，密码文件拷贝到备库:

scp /home/oracle/app/oracle/product/12.1.0/dbhome*1/dbs/initorcl.ora 192.168.206.132:/home/oracle/app/oracle/product/12.1.0/dbhome*1/dbs/  
scp /home/oracle/app/oracle/product/12.1.0/dbhome*1/dbs/orapworcl 192.168.206.132:/home/oracle/app/oracle/product/12.1.0/dbhome*1/dbs/

将主库复制到备库

$rman target sys/123@orcl auxiliary sys/123@stdorcl //terminal运行  
rman>  
执行duplicate:  
run{   
allocate channel c1 type disk;  
allocate channel c2 type disk;  
allocate channel c3 type disk;  
allocate AUXILIARY channel c4 type disk;  
allocate AUXILIARY channel c5 type disk;  
allocate AUXILIARY channel c6 type disk;  
DUPLICATE TARGET DATABASE  
 FOR STANDBY  
 FROM ACTIVE DATABASE  
 DORECOVER  
 NOFILENAMECHECK;  
release channel c1;  
release channel c2;  
release channel c3;  
release channel c4;  
release channel c5;  
release channel c6;  
}

//等待直至出现rman>

第三步：备库

在备库上更改参数文件

$gedit /home/oracle/app/oracle/product/12.1.0/dbhome\_1/dbs/initorcl.ora //运行

文件内容是：

orcl.\_\_data\_transfer\_cache\_size=0  
orcl.\_\_db\_cache\_size=671088640  
orcl.\_\_java\_pool\_size=16777216  
orcl.\_\_large\_pool\_size=33554432  
orcl.\_\_oracle\_base='/home/oracle/app/oracle'#ORACLE\_BASE set from environment  
orcl.\_\_pga\_aggregate\_target=536870912  
orcl.\_\_sga\_target=1258291200  
orcl.\_\_shared\_io\_pool\_size=50331648  
orcl.\_\_shared\_pool\_size=301989888  
orcl.\_\_streams\_pool\_size=0  
\*.\_allow\_resetlogs\_corruption=TRUE  
\*.\_catalog\_foreign\_restore=FALSE  
\*.audit\_file\_dest='/home/oracle/app/oracle/admin/orcl/adump'  
\*.audit\_trail='db'  
\*.compatible='12.1.0.2.0'  
\*.control\_files='/home/oracle/app/oracle/oradata/orcl/control01.ctl','/home/oracle/app/oracle/fast\_recovery\_area/orcl/control02.ctl','/home/oracle/app/oracle/fast\_recovery\_area/orcl/control03.ctl'  
\*.db\_block\_size=8192  
\*.db\_domain=''  
\*.db\_file\_name\_convert='/home/oracle/app/oracle/oradata/orcl/','/home/oracle/app/oracle/oradata/stdorcl/'  
\*.db\_name='orcl'  
\*.db\_unique\_name='stdorcl'  
\*.db\_recovery\_file\_dest='/home/oracle/app/oracle/fast\_recovery\_area'  
\*.db\_recovery\_file\_dest\_size=4823449600  
\*.diagnostic\_dest='/home/oracle/app/oracle'  
\*.dispatchers='(PROTOCOL=TCP)(dispatchers=1)(pool=on)(ticks=1)(connections=500)(sessions=1000)'  
\*.enable\_pluggable\_database=true  
\*.fal\_client='stdorcl'  
\*.fal\_server='orcl'  
\*.inmemory\_max\_populate\_servers=2  
\*.inmemory\_size=157286400  
\*.local\_listener=''  
\*.log\_archive\_config='DG\_CONFIG=(stdorcl,orcl)'  
\*.log\_archive\_dest\_1='LOCATION=/home/oracle/arch VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=stdorcl'  
\*.log\_archive\_dest\_2='SERVICE=orcl LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=orcl'  
\*.log\_archive\_format='%t\_%s\_%r.arc'  
\*.log\_file\_name\_convert='/home/oracle/app/oracle/oradata/orcl/','/home/oracle/app/oracle/oradata/stdorcl/'  
\*.max\_dispatchers=5  
\*.max\_shared\_servers=20  
\*.open\_cursors=400  
\*.parallel\_execution\_message\_size=8192  
\*.pga\_aggregate\_target=511m  
\*.processes=300  
\*.recovery\_parallelism=0  
\*.remote\_login\_passwordfile='EXCLUSIVE'  
\*.service\_names='ORCL'  
\*.sga\_max\_size=1572864000  
\*.sga\_target=1258291200  
\*.shared\_server\_sessions=200  
\*.standby\_file\_management='AUTO'  
\*.undo\_tablespace='UNDOTBS1'

//以上全部替换弹出的文件内容 记得保存

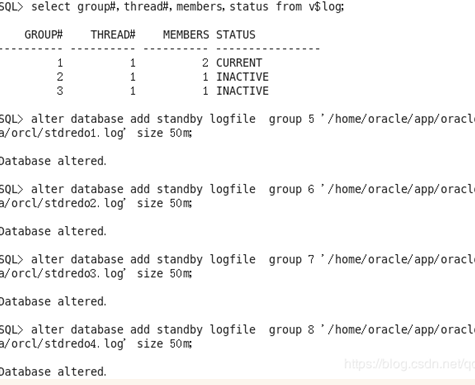


image-20191126205030287

在备库增加静态监听

$gedit /home/oracle/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora //运行   
   
SID\_LIST\_LISTENER =   
 (SID\_LIST =   
 (SID\_DESC =   
 (ORACLE\_HOME = /home/oracle/app/oracle/product/12.1.0/db\_1)   
 (SID\_NAME = orcl)   
 )   
 )   
 //添加到此文件最后   
//重新启动,备库开启实时应用模式:：   
$sqlplus / as sysdba //再次连数据库   
shutdown immediate //sql>   
startup //sql>   
alter database recover managed standby database disconnect; //sql>

测试：

在主库连数据库

$sqlplus / as sysdba //连数据库   
create table lft(sex varchar);

进入备库：

$sqlplus / as sysdba //连数据库   
select \* from lft;