

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的校园论坛的数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 计算机学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 谢芳煊 | 学号 | 201810424304 |
| 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** | | |  |

2021 年 6 月 1 日

目录

[一.背景 3](#_Toc9219)

[二． 创建表空间 3](#_Toc28444)

[三． 创建角色 4](#_Toc8422)

[四． 创建用户 6](#_Toc14839)

[五．创建表及插入数据 10](#_Toc27755)

[六．PL/SQL语言设计存储过程和函数 20](#_Toc329)

[七．备份方案 22](#_Toc19727)

一.背景

在当今网络技术快速发展的时代，已经不是“日新月异”就能形容的信息化世界里，动态性、交互性与时速性的网络论坛成为其中最快速、最丰富和最自由的网络信息资源，也是最受广大的现代人类欢迎的信息交流方式之一。校园论坛网给各个大学的大学生提供了一个良好的交流学习平台，同学们可以在论坛上发表或评论帖子实现相互之间的学习交流。

1. 创建表空间

1.创建space01

CREATE TABLESPACE space01 DATAFILE

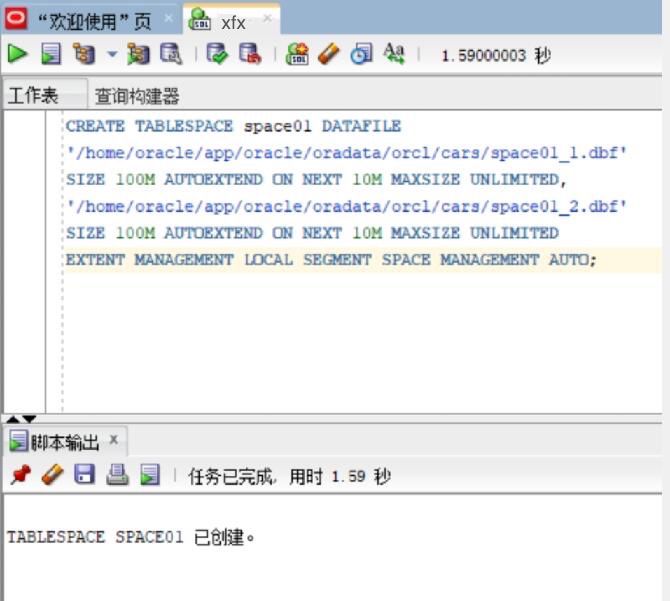
'/home/oracle/app/oracle/oradata/orcl/cams/space01\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/cams/space01\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



1. 创建表空间sapce02

CREATE TABLESPACE space02 DATAFILE

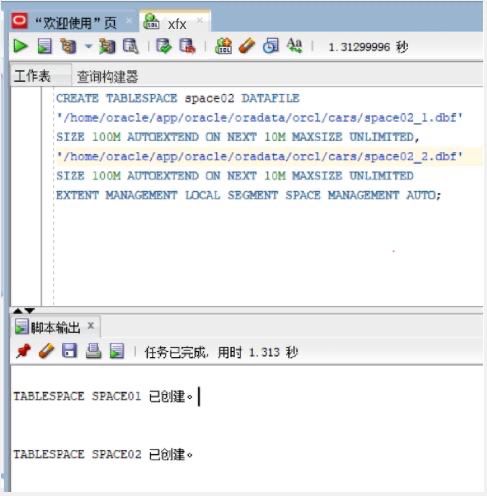
'/home/oracle/app/oracle/oradata/orcl/cams/space02\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/cams/space02\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



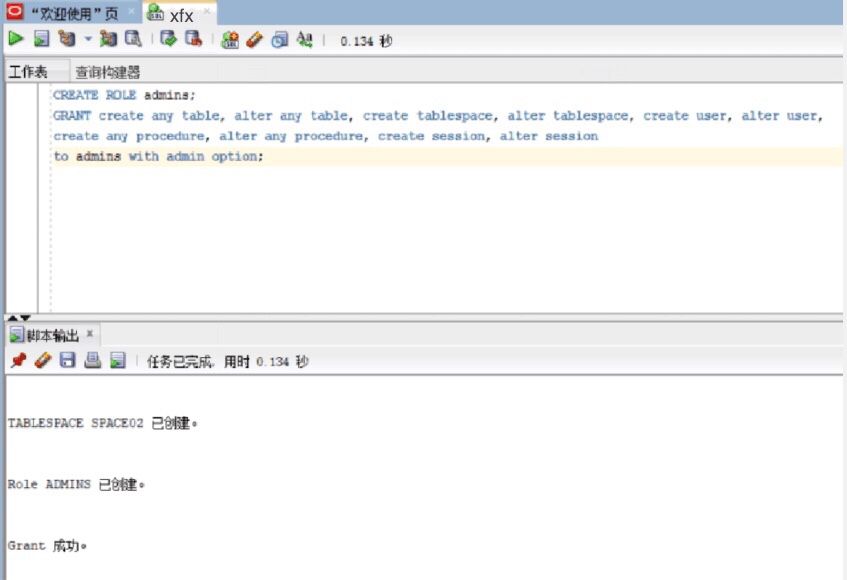
1. 创建角色
2. 创建管理员角色

CREATE ROLE admins;

GRANT create any table, alter any table, create tablespace, alter tablespace, create user, alter user,

create any procedure, alter any procedure, create session, alter session

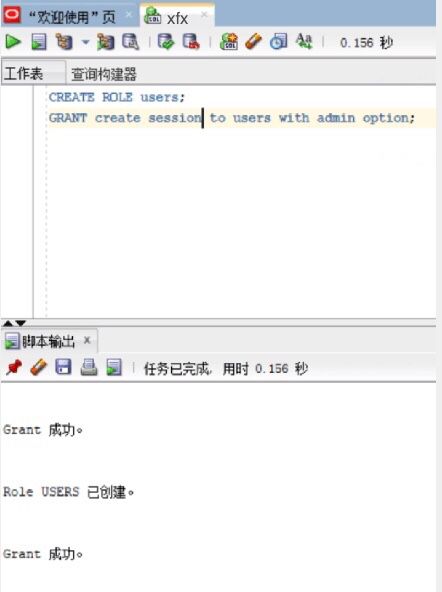
to admins with admin option;



1. 创建用户角色

CREATE ROLE users;

GRANT create session to users with admin option;

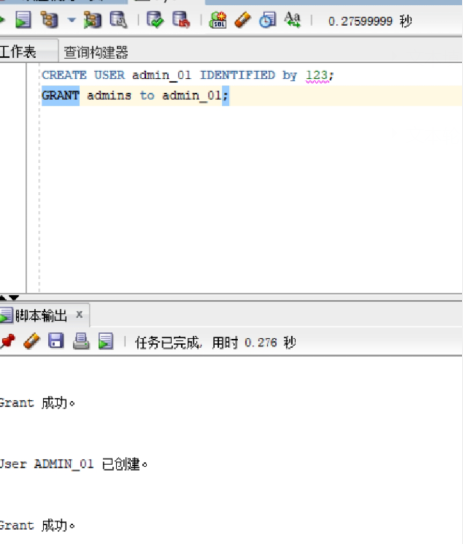


1. 创建用户
2. 创建管理员用户

创建管理员用户admin\_01,并赋予它所有权限

CREATE USER user\_01 IDENTIFIED by 123;

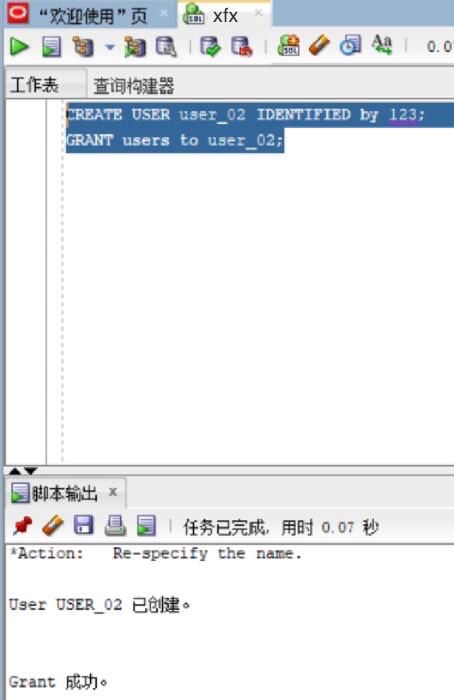
GRANT users to user\_01;



1. 创建普通用户的用户

CREATE USER user\_02 IDENTIFIED by 123;

GRANT users to user\_02;



五．创建表及插入数据

1.模块表 module

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段编号** | **物理字段** | **逻辑字段** | **字段类型** | **是否主键** | **是否非空** | **备注** |
| 1 | 模块名 | name | VARCHAR2 | Y | 非空 |  |
| 2 | 模块描述 | describe | VARCHAR2 |  | 非空 |  |

DROP TABLE Module;

CREATE TABLE Module(

"name" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"describe" VARCHAR2(255 BYTE) VISIBLE NOT NULL

)

TABLESPACE "SPACE01"

LOGGING

NOCOMPRESS

PCTFREE 10

INITRANS 1

STORAGE (

BUFFER\_POOL DEFAULT

)

PARALLEL 1

NOCACHE

DISABLE ROW MOVEMENT

;

COMMENT ON COLUMN Module."name" IS '模块名';

COMMENT ON COLUMN Module."describe" IS '模块的描述';

ALTER TABLE Module ADD CONSTRAINT "SYS\_C0038651" PRIMARY KEY ("name");

-- ----------------------------

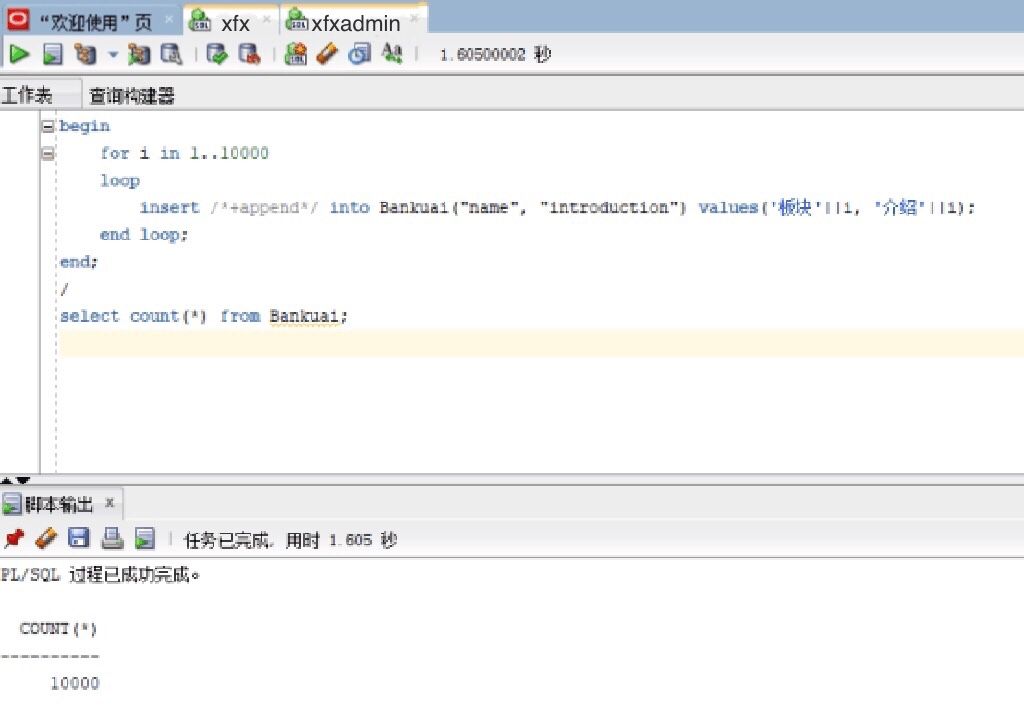
-- Checks structure for table Module

-- ----------------------------

ALTER TABLE Module ADD CONSTRAINT "SYS\_C0038649" CHECK ("name" IS NOT NULL) NOT DEFERRABLE INITIAL xfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE Module ADD CONSTRAINT "SYS\_C0038650" CHECK ("describe" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

插入数据



1. 帖子表 topic

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段编号** | **物理字段** | **逻辑字段** | **字段类型** | **是否主键** | **是否非空** | **备注** |
| 1 | 话题id | id | NUMBER | Y | 非空 |  |
| 2 | 所属模块名 | name | VARCHAR2 |  | 非空 |  |
| 3 | 发帖人id | user\_id | NUMBER |  | 非空 |  |
| 4 | 话题标题 | title | VARCHAR2 |  | 非空 |  |
| 5 | 话题内容 | details | VARCHAR2 |  | 非空 |  |
| 6 | 发起时间 | time | DATE |  | 非空 |  |

DROP TABLE topic;

CREATE TABLE topic(

"id" NUMBER VISIBLE NOT NULL ,

"name" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"user\_id" NUMBER VISIBLE NOT NULL ,

"title" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"details" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"time" DATE VISIBLE NOT NULL

)

TABLESPACE "SPACE01"

LOGGING

NOCOMPRESS

PCTFREE 10

INITRANS 1

STORAGE (

BUFFER\_POOL DEFAULT

)

PARALLEL 1

NOCACHE

DISABLE ROW MOVEMENT

;

COMMENT ON COLUMN post."id" IS '话题id';

COMMENT ON COLUMN post."name" IS '所属模块名';

COMMENT ON COLUMN post."user\_id" IS '发帖人id';

COMMENT ON COLUMN post."title" IS '话题标题';

COMMENT ON COLUMN post."details" IS '话题内容';

COMMENT ON COLUMN post."time" IS '发起时间';

ALTER TABLE post ADD CONSTRAINT "SYS\_C0038667" PRIMARY KEY ("id");

-- ----------------------------

-- Checks structure for table topic

-- ----------------------------

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038661" CHECK ("id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038662" CHECK ("name" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038663" CHECK ("user\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038664" CHECK ("title" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038665" CHECK ("details" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE topicADD CONSTRAINT "SYS\_C0038666" CHECK ("time" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

插入数据



1. 回帖表 reply1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段编号** | **物理字段** | **逻辑字段** | **字段类型** | **是否主键** | **是否非空** | **备注** |
| 1 | 回帖id | id | NUMBER | Y | 非空 |  |
| 2 | 话题id | post\_id | NUMBER | Y | 非空 |  |
| 3 | 回复用户id | user\_id | NUMBER |  | 非空 |  |
| 4 | 回复内容 | details | VARCHAR2 |  | 非空 |  |
| 5 | 回复时间 | time | DATE |  | 非空 |  |

DROP TABLE reply1;

CREATE TABLE reply1 (

"id" NUMBER VISIBLE NOT NULL ,

"post\_id" NUMBER VISIBLE NOT NULL ,

"user\_id" NUMBER VISIBLE NOT NULL ,

"details" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"time" DATE VISIBLE NOT NULL

)

TABLESPACE "SPACE01"

LOGGING

NOCOMPRESS

PCTFREE 10

INITRANS 1

STORAGE (

BUFFER\_POOL DEFAULT

)

PARALLEL 1

NOCACHE

DISABLE ROW MOVEMENT

;

COMMENT ON COLUMN reply."id" IS '回帖id';

COMMENT ON COLUMN reply."post\_id" IS '话题id';

COMMENT ON COLUMN reply."user\_id" IS '回复用户id';

COMMENT ON COLUMN reply."details" IS '回复内容';

COMMENT ON COLUMN reply."time" IS '回复时间';

ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038673" PRIMARY KEY ("id", "post\_id");

-- ----------------------------

-- Checks structure for table reply1

-- ----------------------------

ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038668" CHECK ("id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

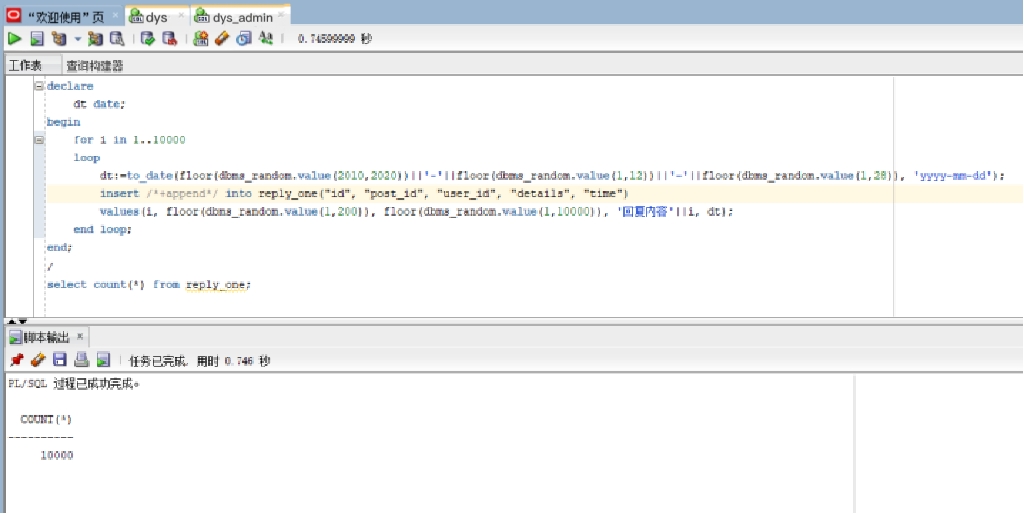
ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038669" CHECK ("post\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038670" CHECK ("user\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038671" CHECK ("details" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply ADD CONSTRAINT "SYS\_C0038672" CHECK ("time" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

插入数据



1. 回贴表 reply2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **字段编号** | **物理字段** | **逻辑字段** | **字段类型** | **是否主键** | **是否非空** | **备注** |
| 1 | 再次回复的id | id | NUMBER | Y | 非空 |  |
| 2 | 话题id | post\_id | NUMBER | Y | 非空 |  |
| 3 | 回复的id | reply\_id | NUMBER | Y | 非空 |  |
| 4 | 回复用户id | user\_id | NUMBER |  | 非空 |  |
| 5 | 回复内容 | details | VARCHAR2 |  | 非空 |  |
| 6 | 回复时间 | time | DATE |  |  |  |

DROP TABLE reply2;

CREATE TABLE reply2 (

"id" NUMBER VISIBLE NOT NULL ,

"post\_id" NUMBER VISIBLE NOT NULL ,

"reply\_id" NUMBER VISIBLE NOT NULL ,

"user\_id" NUMBER VISIBLE NOT NULL ,

"details" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"time" DATE VISIBLE NOT NULL

)

TABLESPACE "SPACE01"

LOGGING

NOCOMPRESS

PCTFREE 10

INITRANS 1

STORAGE (

BUFFER\_POOL DEFAULT

)

PARALLEL 1

NOCACHE

DISABLE ROW MOVEMENT

;

COMMENT ON COLUMN reply2."id" IS '再次回复的id';

COMMENT ON COLUMN reply2."post\_id" IS '话题id';

COMMENT ON COLUMN reply2."reply\_id" IS '回复的id';

COMMENT ON COLUMN reply2."user\_id" IS '用户id';

COMMENT ON COLUMN reply2."details" IS '回复内容';

COMMENT ON COLUMN reply2."time" IS '回复时间';

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038680" PRIMARY KEY ("id", "post\_id", "reply\_id");

-- ----------------------------

-- Checks structure for table reply2

-- ----------------------------

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038674" CHECK ("id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038675" CHECK ("post\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

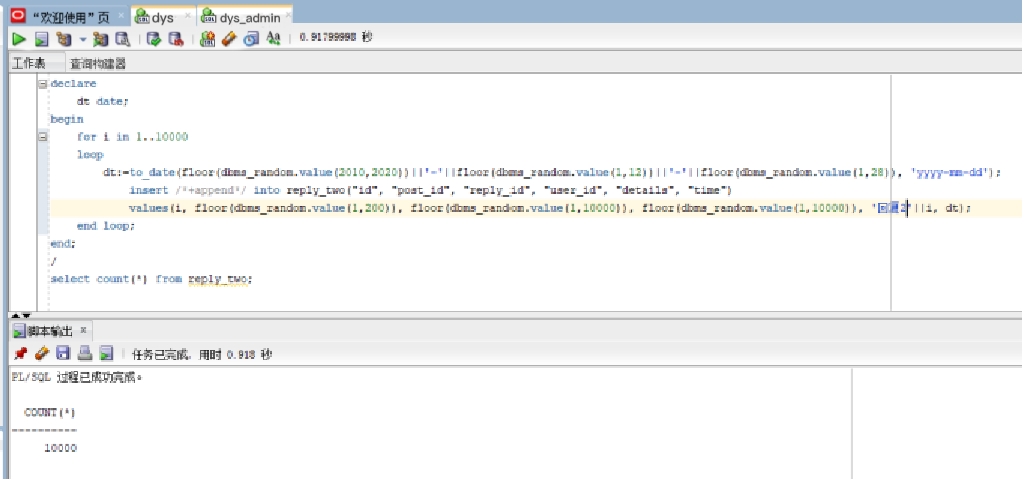
ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038676" CHECK ("reply\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038677" CHECK ("user\_id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038678" CHECK ("details" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE reply2 ADD CONSTRAINT "SYS\_C0038679" CHECK ("time" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

插入数据



1. 用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | user\_table | | | | | |
| 字段编号 | 物理字段 | 逻辑字段 | 字段类型 | 是否主键 | 是否非空 | 备注 |
| 1 | 用户id | id | NUMBER | Y | 非空 |  |
| 2 | 用户名 | user\_name | VARCHAR2 |  | 非空 |  |
| 3 | 用户性别 | sex | VARCHAR2 |  |  |  |

DROP TABLE user\_table;

CREATE TABLE user\_table (

"id" NUMBER VISIBLE NOT NULL ,

"user\_name" VARCHAR2(255 BYTE) VISIBLE NOT NULL ,

"sex" VARCHAR2(255 BYTE) VISIBLE

)

TABLESPACE "SPACE01"

LOGGING

NOCOMPRESS

PCTFREE 10

INITRANS 1

STORAGE (

BUFFER\_POOL DEFAULT

)

PARALLEL 1

NOCACHE

DISABLE ROW MOVEMENT

;

COMMENT ON COLUMN user\_table."id" IS '用户id';

COMMENT ON COLUMN user\_table."user\_name" IS '用户名';

COMMENT ON COLUMN user\_table."sex" IS '用户性别';

ALTER TABLE user\_table ADD CONSTRAINT "SYS\_C0038687" PRIMARY KEY ("id");

-- ----------------------------

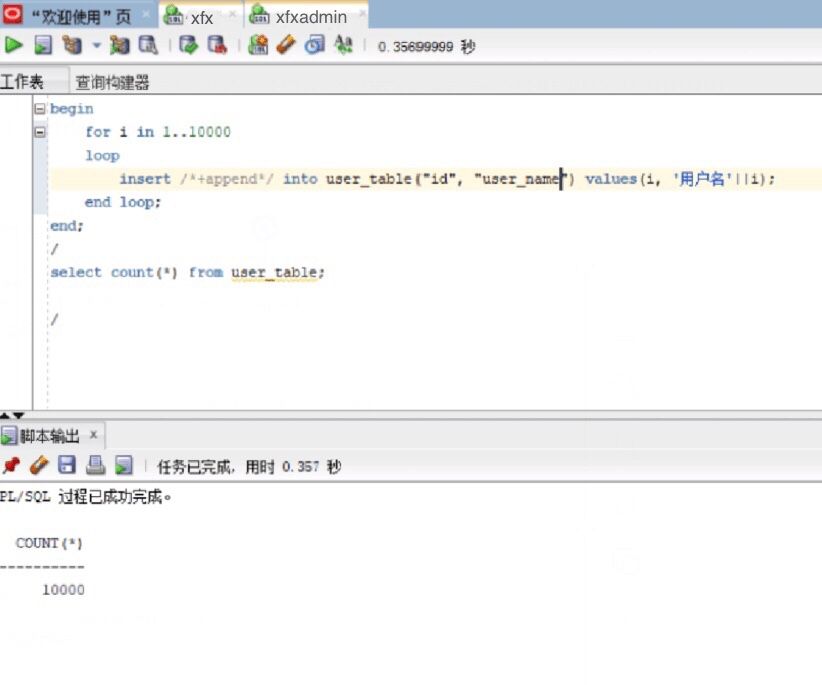
-- Checks structure for table user\_table

-- ----------------------------

ALTER TABLE user\_table ADD CONSTRAINT "SYS\_C0038685" CHECK ("id" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

ALTER TABLE user\_table ADD CONSTRAINT "SYS\_C0038686" CHECK ("user\_name" IS NOT NULL) NOT DEFERRABLE INITIALxfx IMMEDIATE NORExfx VALIDATE;

插入数据



六．PL/SQL语言设计存储过程和函数

1. 函数

create or replace PACKAGE MyPack IS

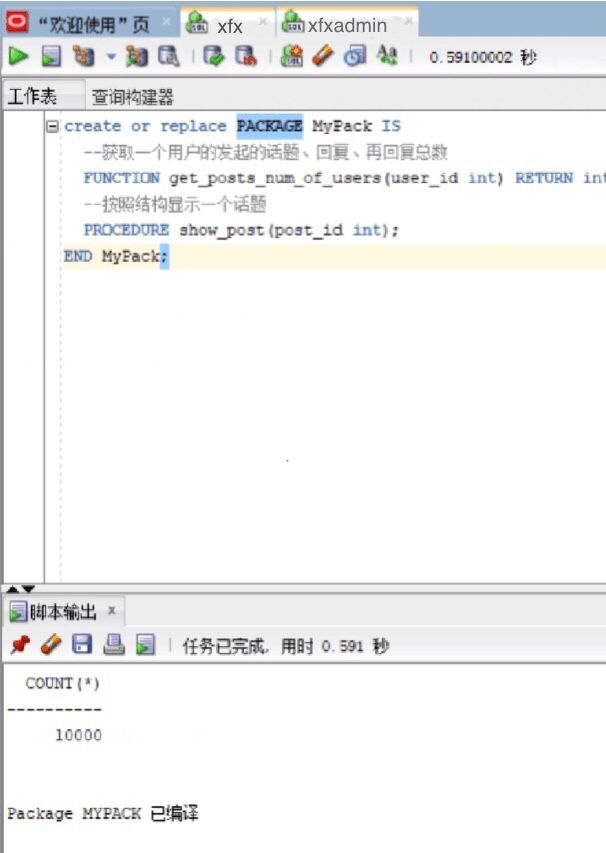
--获取一个用户的发起的话题、回复、再回复总数

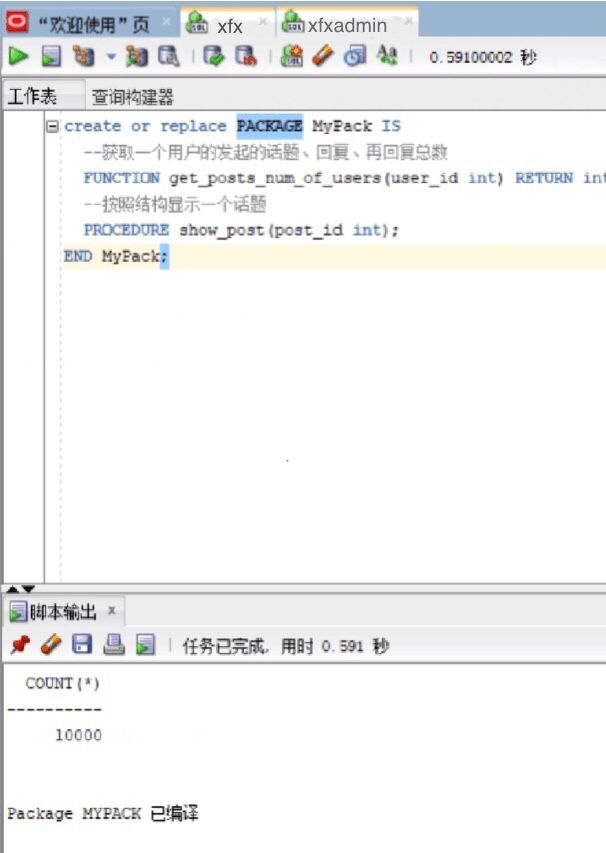
FUNCTION get\_posts\_num\_of\_users(user\_id int) RETURN int;

--按照结构显示一个话题

PROCEDURE show\_post(post\_id int);

END MyPack;





1. 存储过程

create or replace PACKAGE BODY MyPack IS

FUNCTION get\_posts\_num\_of\_users(user\_id int) RETURN int

AS

posts int;

reply1 int;

reply2 int;

begin

select count(\*) into posts from post where "user\_id" = user\_id;

select count(\*) into reply1 from post where "user\_id" = user\_id;

select count(\*) into reply2 from post where "user\_id" = user\_id;

return posts+reply1+reply2;

end;

PROCEDURE show\_post(post\_id int)

AS

temp varchar2(255);

leftspace varchar2(255);

begin

leftspace:=' ';

select "title" into temp from post where "id"=post\_id;

DBMS\_OUTPUT.PUT\_LINE('标题：'||temp);

select "details" into temp from post where "id"=post\_id;

DBMS\_OUTPUT.PUT\_LINE('细节：'||temp);

for v in (select \* from reply1 where "post\_id"=post\_id)

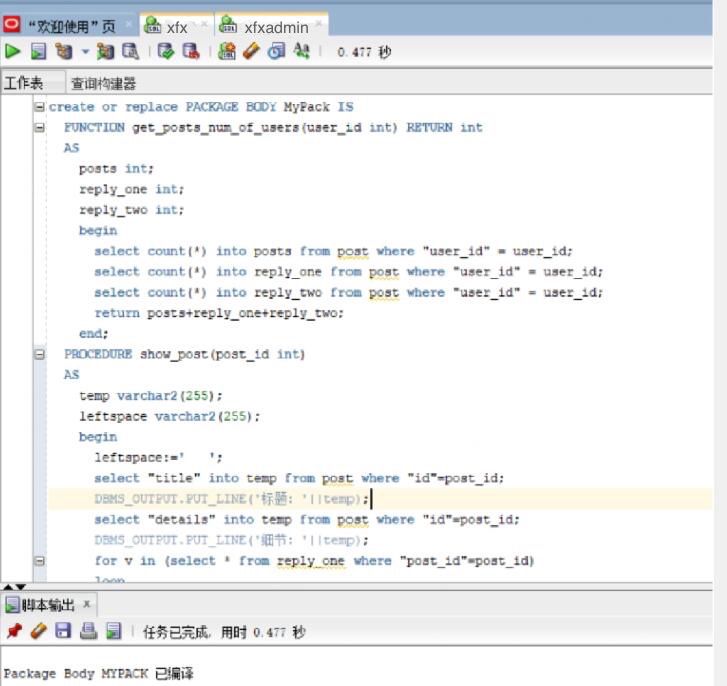
loop

DBMS\_OUTPUT.PUT\_LINE(leftspace||'用户'||v."user\_id"||'回复：'||v."details");

end loop;

END;

END MyPack;



七．备份方案

数据库备份采取每月一次进行数据库全备份，并且在每日凌晨十二点自动开始增量备份的方式进行备份。

1. 开始全备份

全库0级备份

[oracle@deep02~]$ sqlplus study/123@pdborcl

run{

configure retention policy to redundancy 1;

configure controlfile autobackup on;

configure controlfile autobackup format for device type disk to '/home/student/rman\_backup/%F';

configure default device type to disk;

crosscheck backup;

crosscheck archivelog all;

allocate channel c1 device type disk;

allocate channel c2 device type disk;

allocate channel c3 device type disk;

backup incremental level 0 database format '/home/student/rman\_backup/level0\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

release channel c1;

release channel c2;

release channel c3;

}

## 全库1级增量备份

run{

configure retention policy to redundancy 1;

configure controlfile autobackup on;

configure controlfile autobackup format for device type disk to '/home/student/rman\_backup/%F';

configure default device type to disk;

crosscheck backup;

crosscheck archivelog all;

allocate channel c1 device type disk;

allocate channel c2 device type disk;

allocate channel c3 device type disk;

backup incremental level 1 database format '/home/student/rman\_backup/level1\_%d\_%T\_%U.bak';

report obsolete;

delete noprompt obsolete;

delete noprompt expired backup;

delete noprompt expired archivelog all;

release channel c1;

release channel c2;

release channel c3;

}

2. 备份后修改数据

[oracle@deep02~]$ sqlplus xfx/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 hr.mytable;

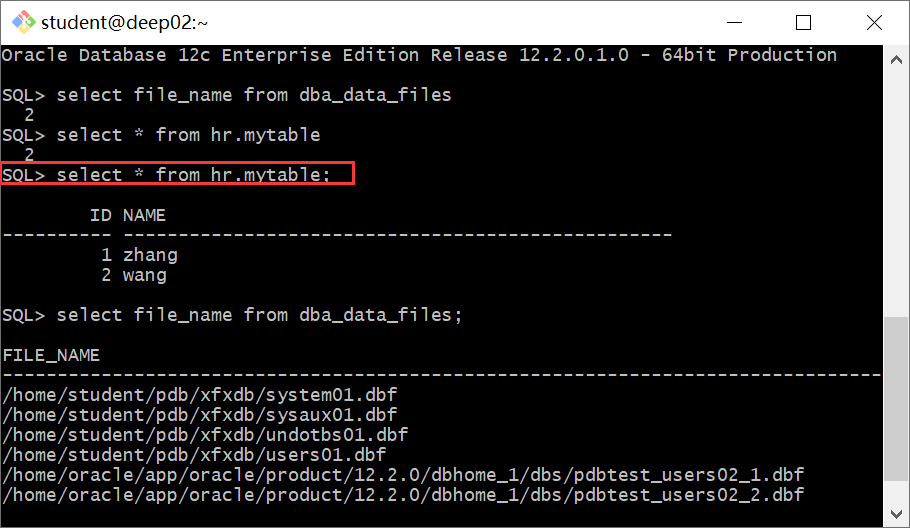
ID NAME

---------- --------------------------------------------------

1 zhang

2 wang

SQL> exit



3. 数据库完全恢复

oracle登录linux,不是student用户,dedicated专用连接模式

需要全库停机，需要oracle用户

sys登录到orcl，查看全库的数据文件

$ sqlplus / as sysdba

SQL> select file\_name from dba\_data\_files;

- 全库停机

$rman target /

RMAN> shutdown immediate; 或者 shutdown abort;

RMAN> exit

- 数据文件改名，模拟文件损失

$mv /home/student/pdb\_xfx/pdbtest\_users02\_1.dbf /home/student/pdb\_xfx/pdbtest\_users02\_1.dbf2

- 全库恢复

$rman target /

RMAN> startup mount;

RMAN> restore database;

RMAN> recover database;

RMAN> alter database open;

## 4.单库完全恢复/不完全恢复

$ sqlplus system/123@202.115.82.8/xfx

SQL> select file\_name from dba\_data\_files;

/home/student/pdb/ xfx /system01.dbf

/home/student/pdb/ xfx /sysaux01.dbf

/home/student/pdb/ xfx /undotbs01.dbf

/home/student/pdb/ xfx /users01.dbf

/home/student/pdb/pdbtest\_users02\_1.dbf

/home/student/pdb/pdbtest\_users02\_2.dbf

/home/student/pdb\_xfx/pdbtest\_users02\_1.dbf

/home/student/pdb\_xfx/pdbtest\_users02\_2.dbf

SQL> select \* from xfx.mytable;

ID NAME

---------- --------------------------------------------------

3 zhang

4 wang

5 xie

SQL> select to\_char(sysdate,'yyyy-mm-dd hh24:mi:ss') as currentdate from dual;

CURRENTDATE

-------------------

2021-06-14 08:02:24

SQL> update xfx.mytable set id=id+1;

SQL> commit;

SQL> select \* from xfx.mytable;

ID NAME

---------- --------------------------------------------------

4 zhang

5 wang

6 xie

SQL> select to\_char(sysdate,'yyyy-mm-dd hh24:mi:ss') as currentdate from dual;

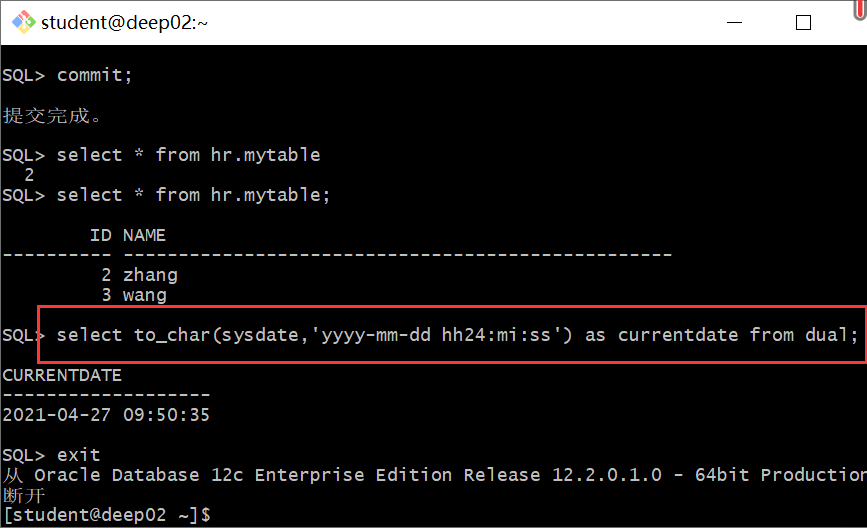
CURRENTDATE

-------------------

2021-06-14 08:03:01

SQL> exit;

- 关闭xfx数据库



$rman target sys/123@202.115.82.8/orcl:dedicated

RMAN> alter pluggable database xfx close;

RMAN> exit;

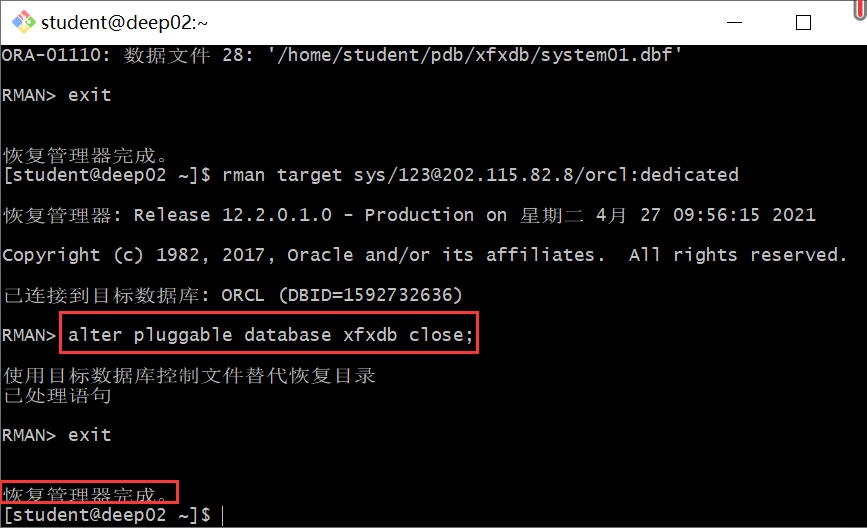
或者

$sqlplus sys/123@202.115.82.8/orcl:dedicated as sysdba;

SQL>SELECT server FROM v$session WHERE SID=(SELECT DISTINCT SID FROM v$mystat);

SQL>alter session set container=xfx;

SQL>shutdown immediate; 或者 shutdown abort;



- 数据文件改名，模拟文件损失

$mv -f /home/student/pdb\_xfx/pdbtest\_users02\_1.dbf /home/student/pdb\_xfx/pdbtest\_users02\_1.dbf2

- 选项1：单库完全恢复

$rman target sys/123@202.115.82.8/orcl:dedicated

RMAN> restore pluggable database xfx;

RMAN> recover pluggable database xfx;

RMAN> alter pluggable database xfx open;

RMAN> exit;

- 完全恢复成功后，xfx用户登录xfx，

$ sqlplus xfx/123@202.115.82.8/xfx

SQL> select \* from mytable;

ID NAME

---------- --------------------------------------------------

4 zhang

5 wang

6 xie

可见，完全恢复成功，数据是最新的（即2021-06-14 09:23:01），无损失。

## 选项2：单库不完全恢复,恢复到update语句之前的状态，即恢复到2021-06-14 08:02:24时刻的数据

$rman target sys/123@202.115.82.8/orcl:dedicated

RMAN> restore pluggable database xfx;

RMAN> recover pluggable database xfx until time "to\_date('2021-06-14 08:02:24','yyyy-mm-dd hh24:mi:ss')" AUXILIARY DESTINATION '/home/student/zwd';

正在开始介质的恢复

线程 1 序列 1624 的归档日志已作为文件 /home/oracle/app/oracle/product/12.2.0/dbhome\_1/dbs/arch1\_1624\_1064951903.dbf 存在于磁盘上

线程 1 序列 1625 的归档日志已作为文件 /home/oracle/app/oracle/product/12.2.0/dbhome\_1/dbs/arch1\_1625\_1064951903.dbf 存在于磁盘上

RMAN> alter pluggable database xfx open resetlogs;

已处理语句

RMAN>exit

- 不完全恢复成功后，xfx用户登录xfx，

$ sqlplus xfx/123@202.115.82.8/xfx

SQL> select \* from mytable;

ID NAME

---------- --------------------------------------------------

3 zhang

4 wang

5 xie

可见，不完全恢复成功，数据回到了修改前的状态.