

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的商城订单的数据库设计 | | |
| 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 | 刘肖 | 学号 | 201710414210 |
| 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

**一、引言**

**1.1编写目的**

本数据库设计说明书是对商城订单数据库设计定义，包括本系统数据库设计、数据库的实现代码、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，以及设计数据库备份方案和设计容灾方案。

**1.2背景**

数据库(Database) 是按照数据结构来组织、存储和管理数据的仓库，它产生于距今五十年前,随着信息技术和市场的发展,特别是二十世纪九十年代以后，数据管理不再仅仅是存储和管理数据,而转变成用户所需要的各种数据管理的方式。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

数据库的种类很多，常用的有DB2, Oracle, Sybase, SQL Server, mySQL, Access等，各种数据库各有优劣。其中Oracle以其使用方便，安全性高，功能强大和完善的企业服务系统赢得了众多用户的青睐。

在人类社会已经迈入21世纪的今天，人们深刻的感受到了Oracle在生活和工作中的作用越来越重要,越来越多的职业需要具有Oracle的应用技能。今天，计算机技术不但广泛地应用在办公自动化中，还全面渗透到各行各业。所有与相关的职业都要求工作者有很强的计算机操作技能，做到运用自如，熟练而且深入的掌握软件的应用。随着计算机技术的飞速发展，今后，计算机作为一种崭新的生产力，将在信息社会即新技术革命中发挥越来越重要的作用，并进一步推动人类社会更快的向前发展。其中Oracle数据库起着重要的作用。

如今企业面临着越来越激烈的外部竞争环境,考传统的经营方式已很难适应这种变化，企业现在需要的是先进的经营管理理念和高效的资源管理方式。这就需要引进高效，合理，针对性强的数据库管理系统，而oracle则是世界上大多数企业的首选，可以看出oracle在企业信息化建设中起到的巨大作用。

**二、数据库设计**

**2.1创建表空间**

（1）创建表空间user10

运行代码如下：

Create Tablespace user10

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_user10\_1.dbf'

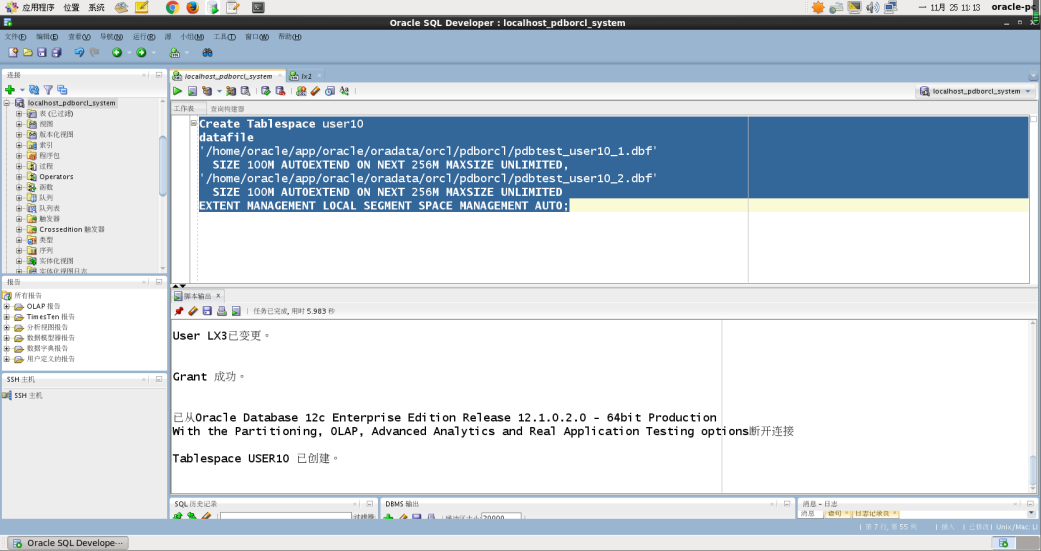
SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_user10\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

运行结果截图：



（2）创建表空间user22

运行代码如下：

Create Tablespace user22

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_user22\_1.dbf'

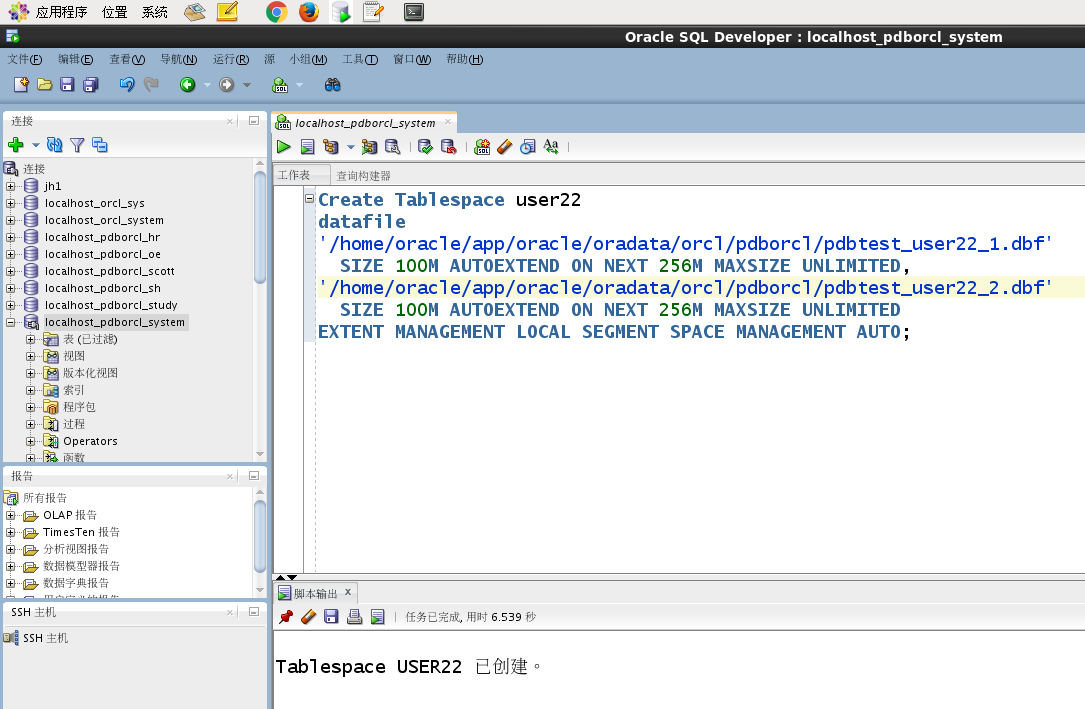
SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_user22\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

运行结果截图：



**2.2创建角色和用户，并授权和分配空间**

(1)创建角色jh1和用户jh

运行代码如下：

SQL> CREATE ROLE jh1;

SQL> GRANT connect,resource,CREATE VIEW TO jh1;

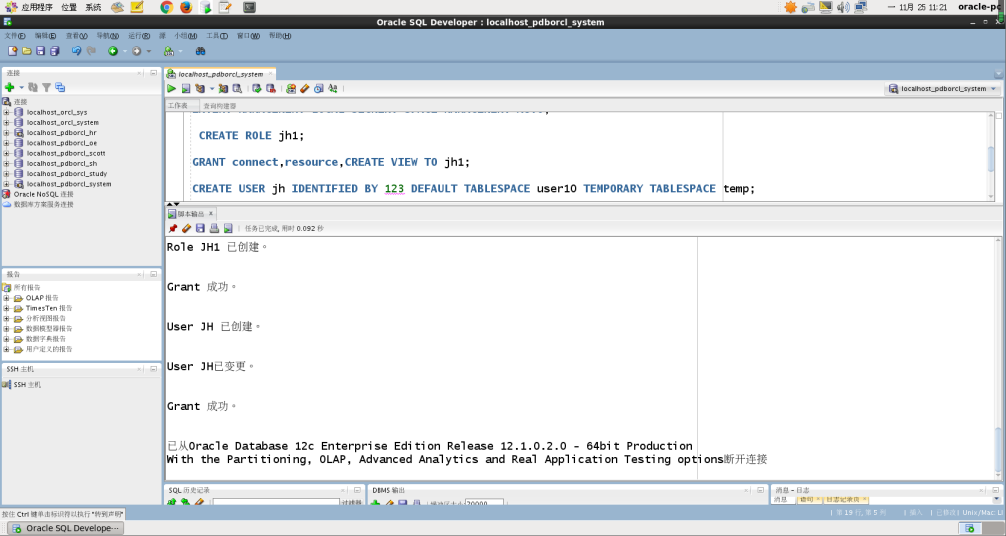
SQL> CREATE USER jh IDENTIFIED BY 123 DEFAULT TABLESPACE user10 TEMPORARY TABLESPACE temp;

SQL> ALTER USER jh QUOTA 50M ON users;

SQL> GRANT jh1 TO jh;

SQL> exit

运行结果截图：



(2)创建角色jh2和用户jh3

运行代码如下：

SQL> CREATE ROLE jh2;

SQL> GRANT connect,resource,CREATE VIEW TO jh2;

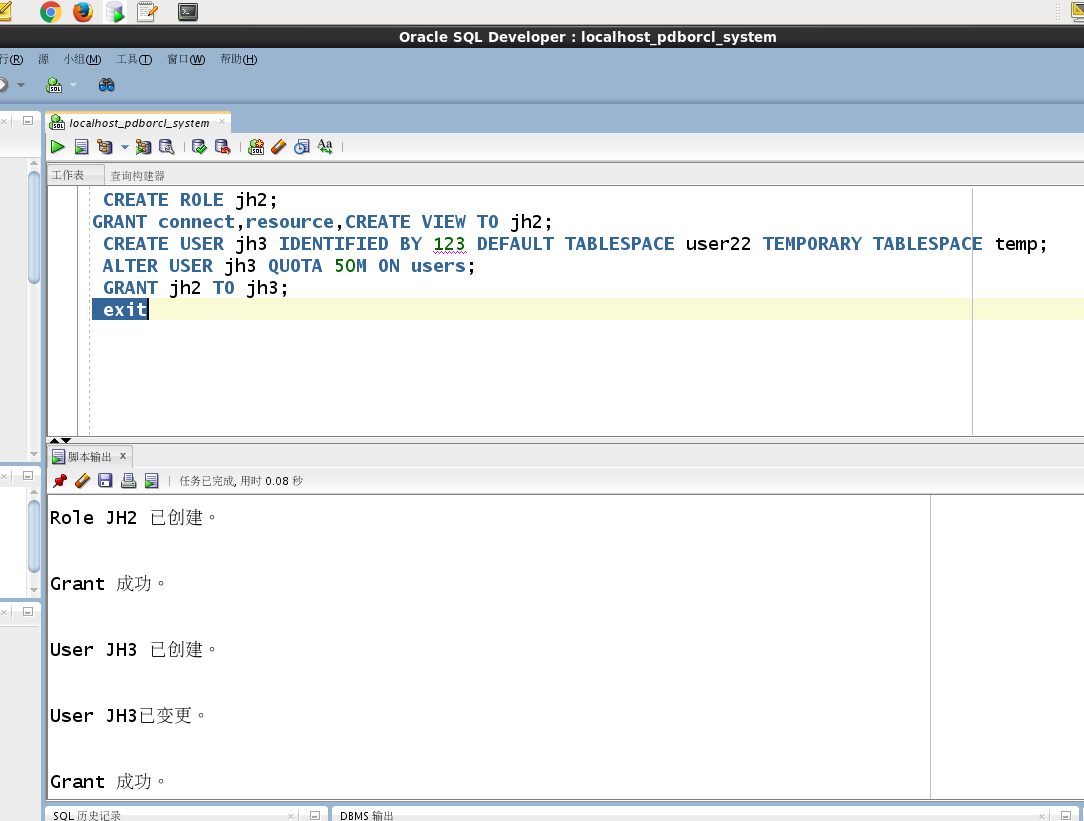
SQL> CREATE USER jh3 IDENTIFIED BY 123 DEFAULT TABLESPACE user22 TEMPORARY TABLESPACE temp;

SQL> ALTER USER jh3 QUOTA 50M ON users;

SQL> GRANT jh2 TO jh3;

SQL> exit

运行结果截图：



**2.3创建5张表**

（1）创建DEPARTMENTS表

运行代码如下：

CREATE TABLE DEPARTMENTS

(

DEPARTMENT\_ID NUMBER(6, 0) NOT NULL

, DEPARTMENT\_NAME VARCHAR2(40 BYTE) NOT NULL

, CONSTRAINT DEPARTMENTS\_PK PRIMARY KEY

(

DEPARTMENT\_ID

)

USING INDEX

(

CREATE UNIQUE INDEX DEPARTMENTS\_PK ON DEPARTMENTS (DEPARTMENT\_ID ASC)

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 2

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

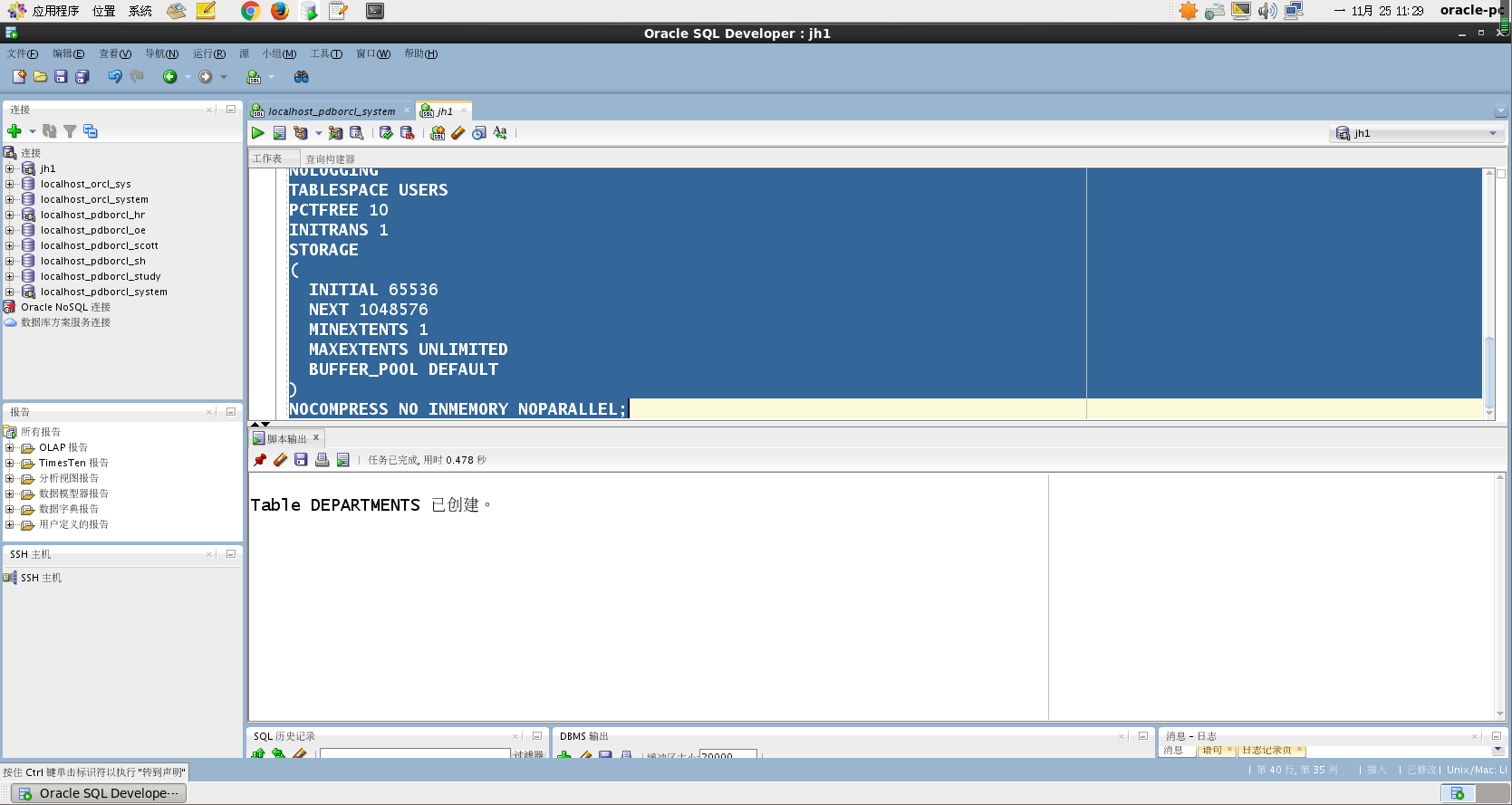
MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY NOPARALLEL;

运行结果截图：



1. 创建EMPLOYEES表

运行代码如下：

CREATE TABLE EMPLOYEES

(

EMPLOYEE\_ID NUMBER(6, 0) NOT NULL

, NAME VARCHAR2(40 BYTE) NOT NULL

, EMAIL VARCHAR2(40 BYTE)

, PHONE\_NUMBER VARCHAR2(40 BYTE)

, HIRE\_DATE DATE NOT NULL

, SALARY NUMBER(8, 2)

, MANAGER\_ID NUMBER(6, 0)

, DEPARTMENT\_ID NUMBER(6, 0)

, PHOTO BLOB

, CONSTRAINT EMPLOYEES\_PK PRIMARY KEY

(

EMPLOYEE\_ID

)

USING INDEX

(

CREATE UNIQUE INDEX EMPLOYEES\_PK ON EMPLOYEES (EMPLOYEE\_ID ASC)

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 2

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NO INMEMORY

NOPARALLEL

LOB (PHOTO) STORE AS SYS\_LOB0000092017C00009$$

(

ENABLE STORAGE IN ROW

CHUNK 8192

NOCACHE

NOLOGGING

TABLESPACE USER10

STORAGE

(

INITIAL 106496

NEXT 1048576

MINEXTENTS 1

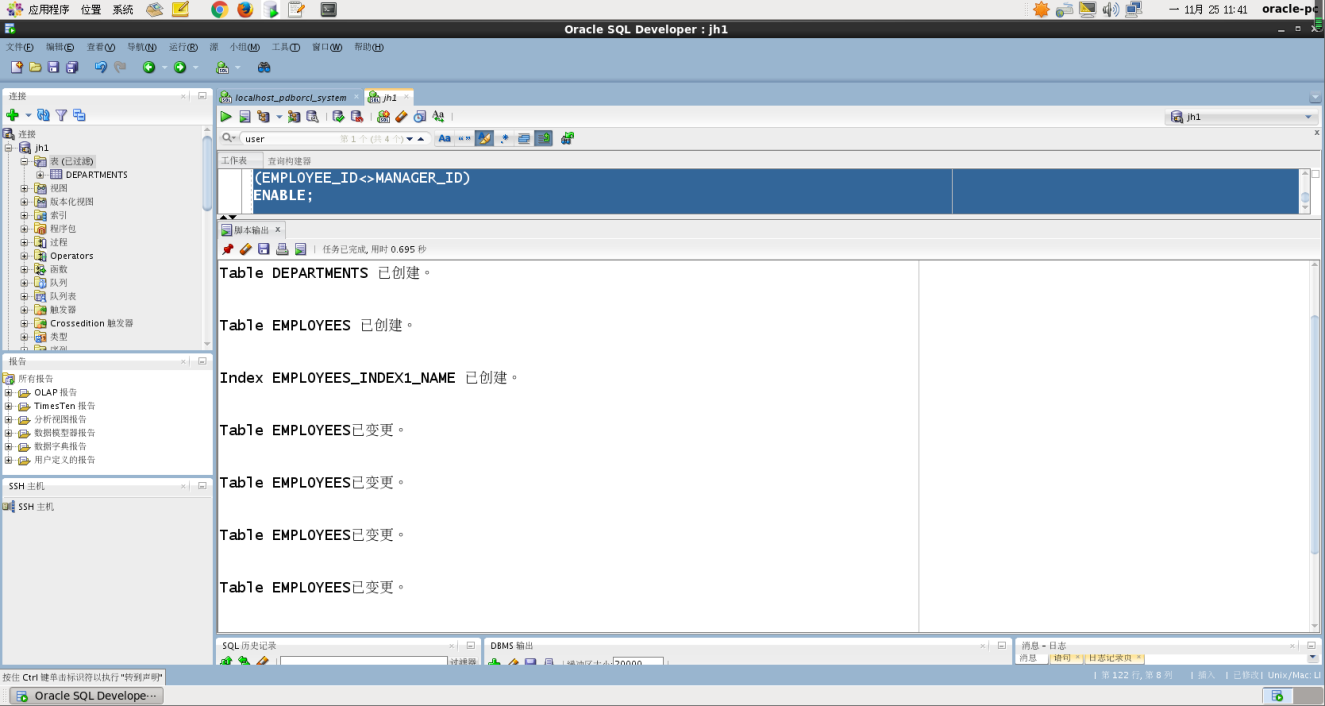
MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

);

运行结果截图：



1. 创建PRODUCTS表

运行代码如下：

CREATE TABLE PRODUCTS

(

PRODUCT\_NAME VARCHAR2(40 BYTE) NOT NULL

, PRODUCT\_TYPE VARCHAR2(40 BYTE) NOT NULL

, CONSTRAINT PRODUCTS\_PK PRIMARY KEY

(

PRODUCT\_NAME

)

ENABLE

)

LOGGING

TABLESPACE "USER10"

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

BUFFER\_POOL DEFAULT

);

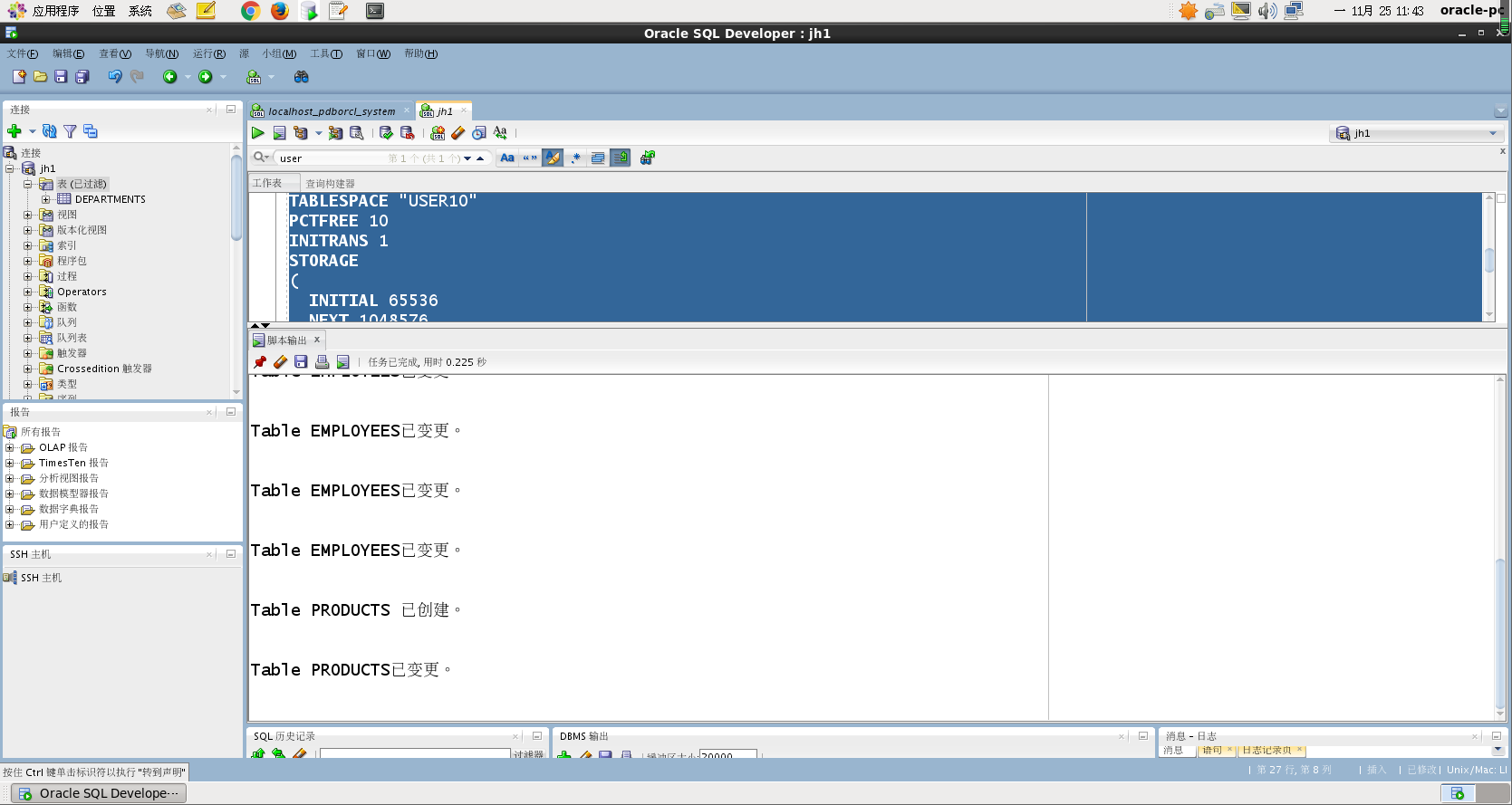
ALTER TABLE PRODUCTS

ADD CONSTRAINT PRODUCTS\_CHK1 CHECK

(PRODUCT\_TYPE IN ('耗材', '手机', '电脑'))

ENABLE;

运行结果截图：



1. 创建ORDERS表

运行代码如下：

CREATE TABLE ORDERS

(

ORDER\_ID NUMBER(10, 0) NOT NULL

, CUSTOMER\_NAME VARCHAR2(40 BYTE) NOT NULL

, CUSTOMER\_TEL VARCHAR2(40 BYTE) NOT NULL

, ORDER\_DATE DATE NOT NULL

, EMPLOYEE\_ID NUMBER(6, 0) NOT NULL

, DISCOUNT NUMBER(8, 2) DEFAULT 0

, TRADE\_RECEIVABLE NUMBER(8, 2) DEFAULT 0

)

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NOPARALLEL

PARTITION BY RANGE (ORDER\_DATE)

(

PARTITION PARTITION\_BEFORE\_2016 VALUES LESS THAN (TO\_DATE(' 2016-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 8388608

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_BEFORE\_2017 VALUES LESS THAN (TO\_DATE(' 2017-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 8388608

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

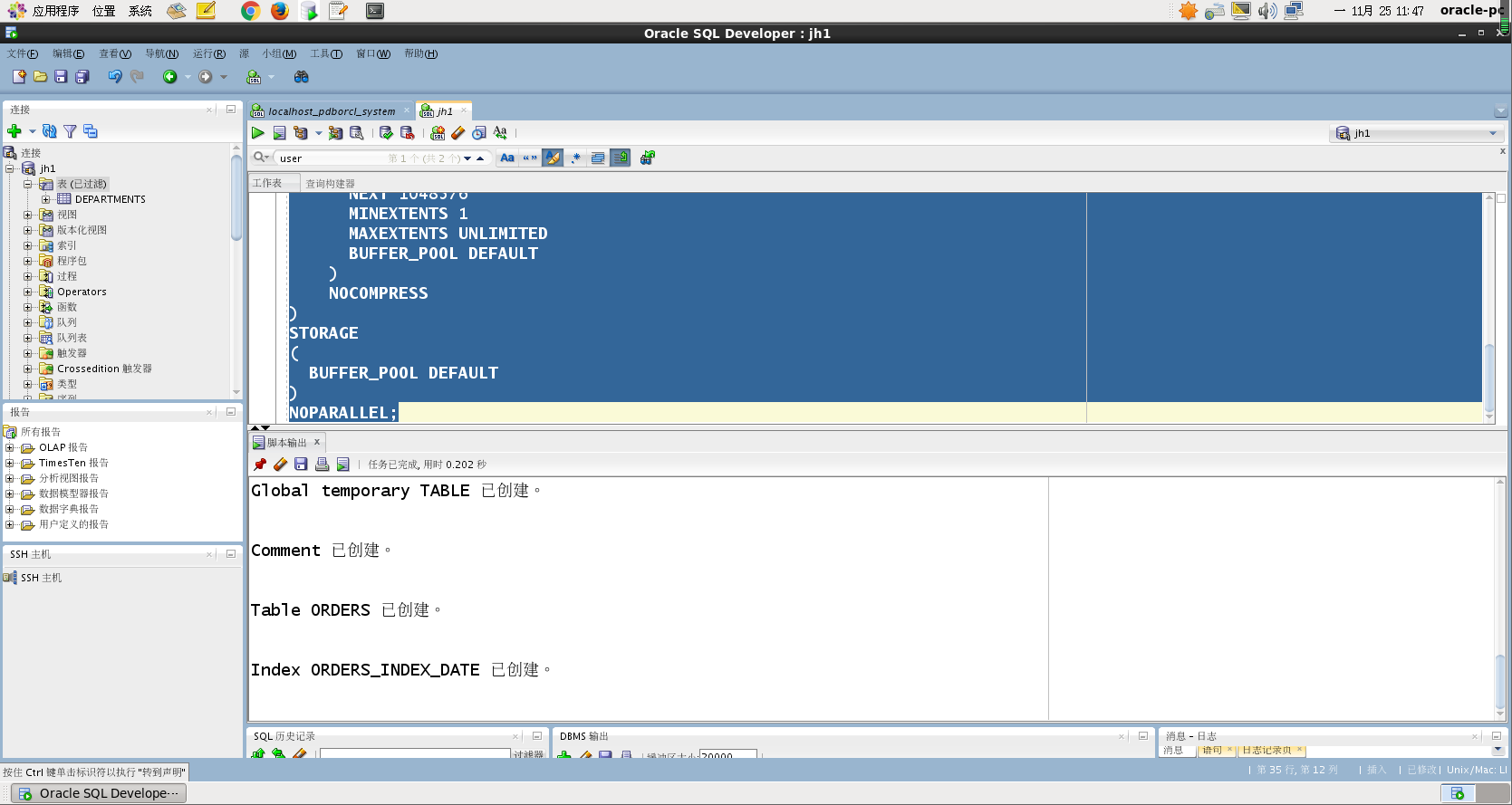
BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

);

运行结果截图：



（5）创建ORDER\_DETAILS表

运行代码如下：

CREATE TABLE ORDER\_DETAILS

(

ID NUMBER(10, 0) NOT NULL

, ORDER\_ID NUMBER(10, 0) NOT NULL

, PRODUCT\_NAME VARCHAR2(40 BYTE) NOT NULL

, PRODUCT\_NUM NUMBER(8, 2) NOT NULL

, PRODUCT\_PRICE NUMBER(8, 2) NOT NULL

, CONSTRAINT ORDER\_DETAILS\_FK1 FOREIGN KEY

(

ORDER\_ID

)

REFERENCES ORDERS

(

ORDER\_ID

)

ENABLE

)

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NOPARALLEL

PARTITION BY REFERENCE (ORDER\_DETAILS\_FK1)

(

PARTITION PARTITION\_BEFORE\_2016

NOLOGGING

TABLESPACE USERS --必须指定表空间，否则会将分区存储在用户的默认表空间中

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 8388608

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY,

PARTITION PARTITION\_BEFORE\_2017

NOLOGGING

TABLESPACE USER10

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 8388608

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

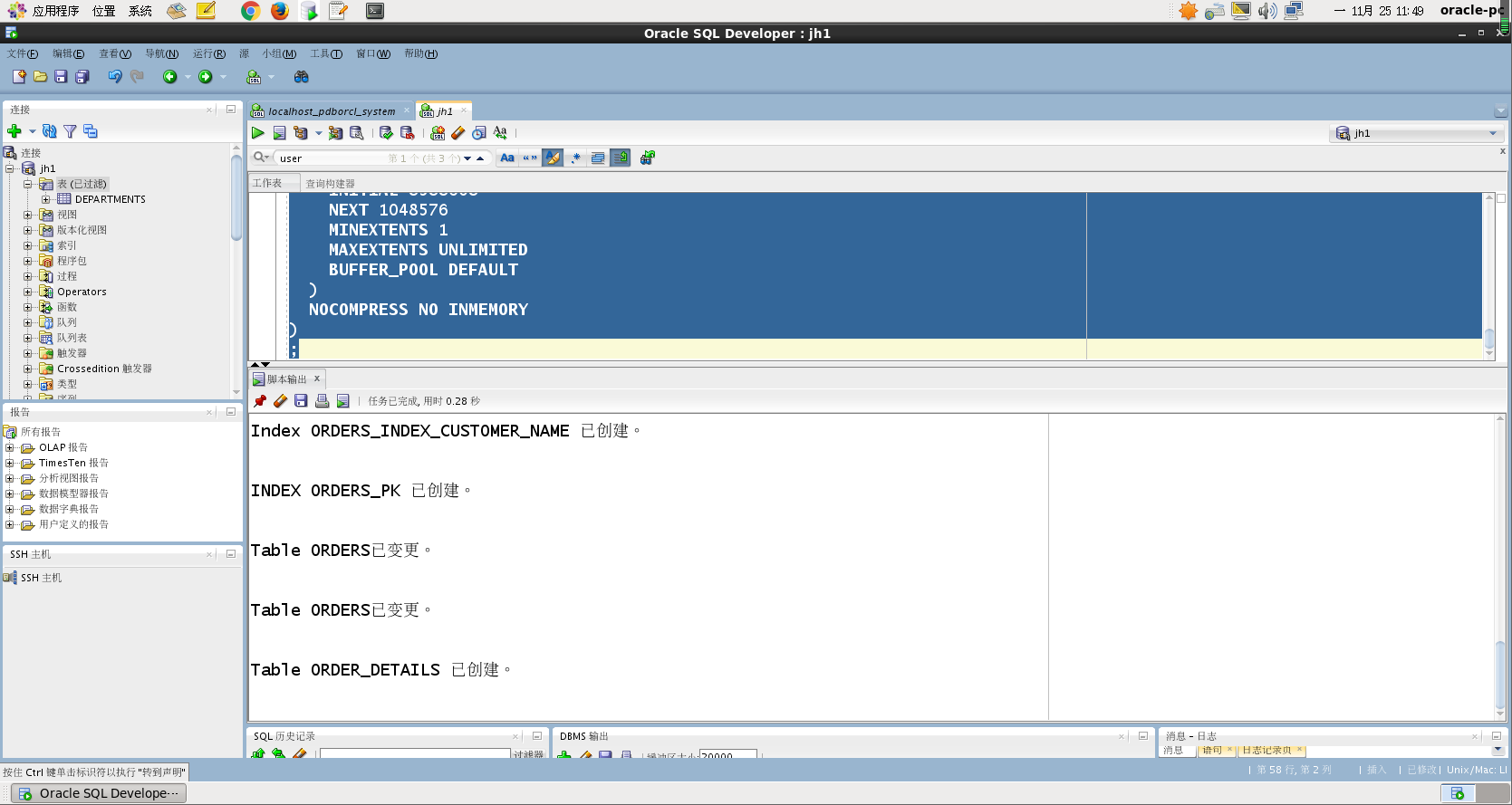
)

NOCOMPRESS NO INMEMORY

)

;

运行结果截图：



**2.4插入5万条数据**

运行代码如下：

declare

dt date;

m number(8,2);

V\_EMPLOYEE\_ID NUMBER(6);

v\_order\_id number(10);

v\_name varchar2(100);

v\_tel varchar2(100);

v number(10,2);

begin

for i in 1..10000

loop

if i mod 2 =0 then

dt:=to\_date('2015-3-2','yyyy-mm-dd')+(i mod 60);

else

dt:=to\_date('2016-3-2','yyyy-mm-dd')+(i mod 60);

end if;

V\_EMPLOYEE\_ID:=CASE I MOD 6 WHEN 0 THEN 11 WHEN 1 THEN 111 WHEN 2 THEN 112

WHEN 3 THEN 12 WHEN 4 THEN 121 ELSE 122 END;

--插入订单

v\_order\_id:=SEQ\_ORDER\_ID.nextval; --应该将SEQ\_ORDER\_ID.nextval保存到变量中。

v\_name := 'aa'|| 'aa';

v\_name := 'zhang' || i;

v\_tel := '139888883' || i;

insert /\*+append\*/ into ORDERS (ORDER\_ID,CUSTOMER\_NAME,CUSTOMER\_TEL,ORDER\_DATE,EMPLOYEE\_ID,DISCOUNT)

values (v\_order\_id,v\_name,v\_tel,dt,V\_EMPLOYEE\_ID,dbms\_random.value(100,0));

--插入订单y一个订单包括3个产品

v:=dbms\_random.value(10000,4000);

v\_name:='computer'|| (i mod 3 + 1);

insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)

values (SEQ\_ORDER\_DETAILS\_ID.NEXTVAL,v\_order\_id,v\_name,2,v);

v:=dbms\_random.value(1000,50);

v\_name:='paper'|| (i mod 3 + 1);

insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)

values (SEQ\_ORDER\_DETAILS\_ID.NEXTVAL,v\_order\_id,v\_name,3,v);

v:=dbms\_random.value(9000,2000);

v\_name:='phone'|| (i mod 3 + 1);

insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)

values (SEQ\_ORDER\_DETAILS\_ID.NEXTVAL,v\_order\_id,v\_name,1,v);

--在触发器关闭的情况下，需要手工计算每个订单的应收金额：

select sum(PRODUCT\_NUM\*PRODUCT\_PRICE) into m from ORDER\_DETAILS where ORDER\_ID=v\_order\_id;

if m is null then

m:=0;

end if;

UPDATE ORDERS SET TRADE\_RECEIVABLE = m - discount WHERE ORDER\_ID=v\_order\_id;

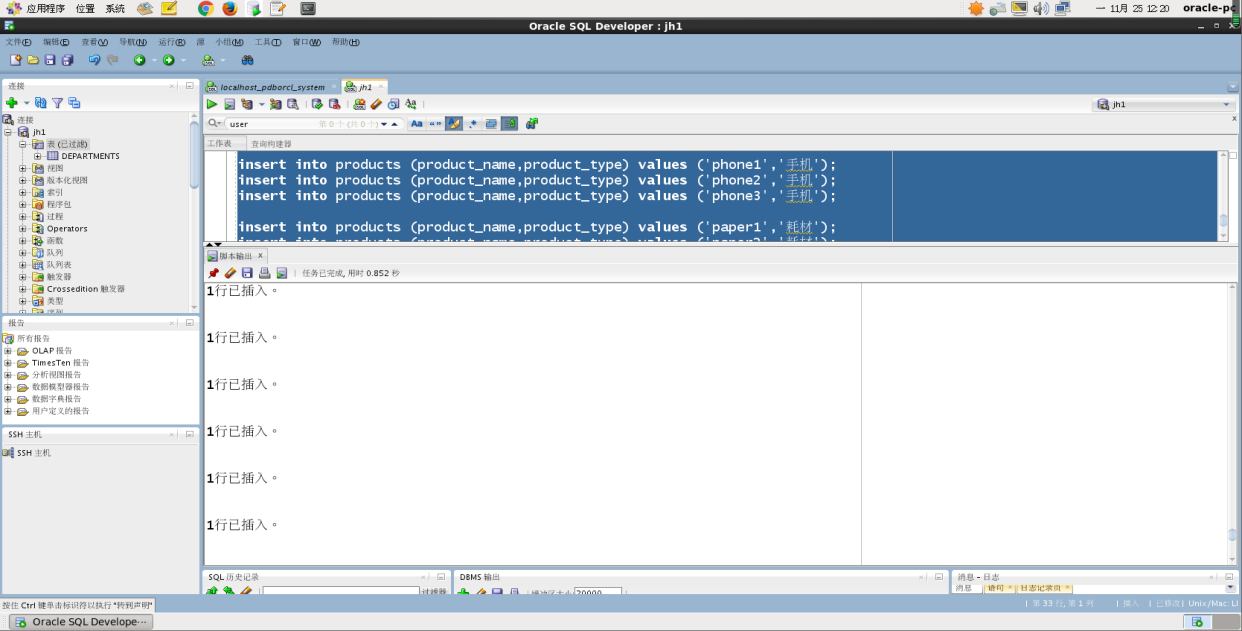
IF I MOD 1000 =0 THEN

commit; --每次提交会加快插入数据的速度

END IF;

end loop;

运行结果截图：



**2.5查询数据量**

运行代码如下：

SELECT count(\*) from ORDERS\_DETAILS

运行结果截图：



**2.6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。**

（1）建立程序包

运行代码如下：

create or replace PACKAGE MyPack IS

/\*

包MyPack中有：

一个函数:Get\_SaleAmount(V\_DEPARTMENT\_ID NUMBER)，

一个过程:Get\_Employees(V\_EMPLOYEE\_ID NUMBER)

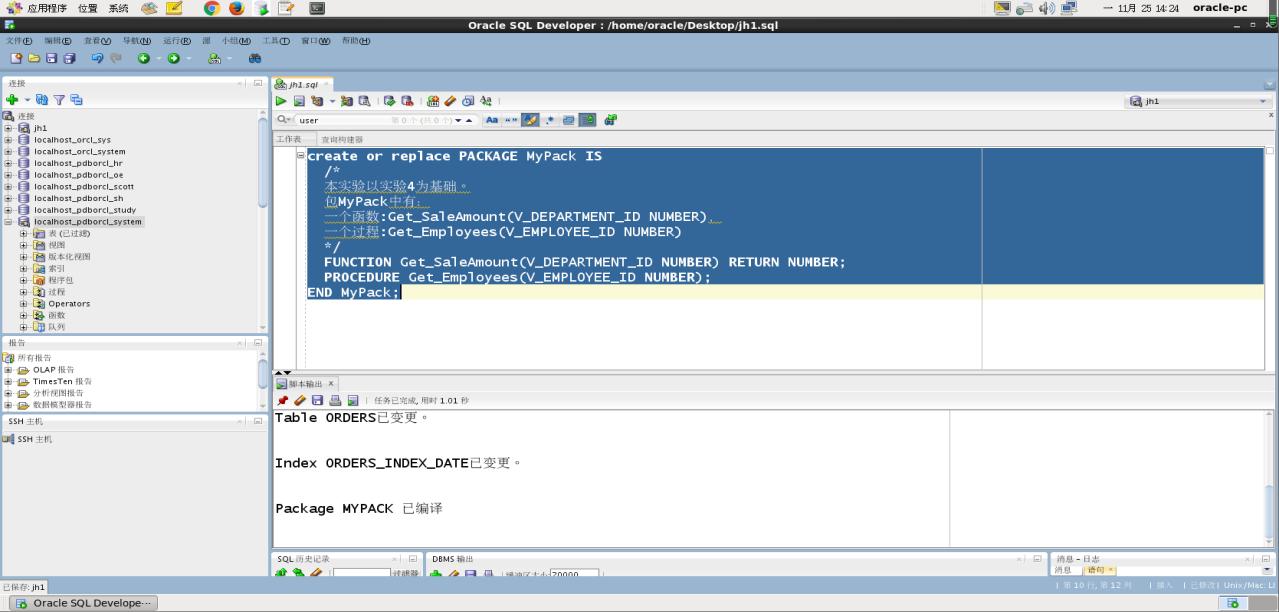
\*/

FUNCTION Get\_SaleAmount(V\_DEPARTMENT\_ID NUMBER) RETURN NUMBER;

PROCEDURE Get\_Employees(V\_EMPLOYEE\_ID NUMBER);

END MyPack;

运行结果截图：



运行代码如下：

create or replace PACKAGE BODY MyPack IS

FUNCTION Get\_SaleAmount(V\_DEPARTMENT\_ID NUMBER) RETURN NUMBER

AS

N NUMBER(20,2); --注意，订单ORDERS.TRADE\_RECEIVABLE的类型是NUMBER(8,2),汇总之后，数据要大得多。

BEGIN

SELECT SUM(O.TRADE\_RECEIVABLE) into N FROM ORDERS O,EMPLOYEES E

WHERE O.EMPLOYEE\_ID=E.EMPLOYEE\_ID AND E.DEPARTMENT\_ID =V\_DEPARTMENT\_ID;

RETURN N;

END;

PROCEDURE GET\_EMPLOYEES(V\_EMPLOYEE\_ID NUMBER)

AS

LEFTSPACE VARCHAR(2000);

begin

--通过LEVEL判断递归的级别

LEFTSPACE:=' ';

--使用游标

for v in

(SELECT LEVEL,EMPLOYEE\_ID,NAME,MANAGER\_ID FROM employees

START WITH EMPLOYEE\_ID = V\_EMPLOYEE\_ID

CONNECT BY PRIOR EMPLOYEE\_ID = MANAGER\_ID)

LOOP

DBMS\_OUTPUT.PUT\_LINE(LPAD(LEFTSPACE,(V.LEVEL-1)\*4,' ')||

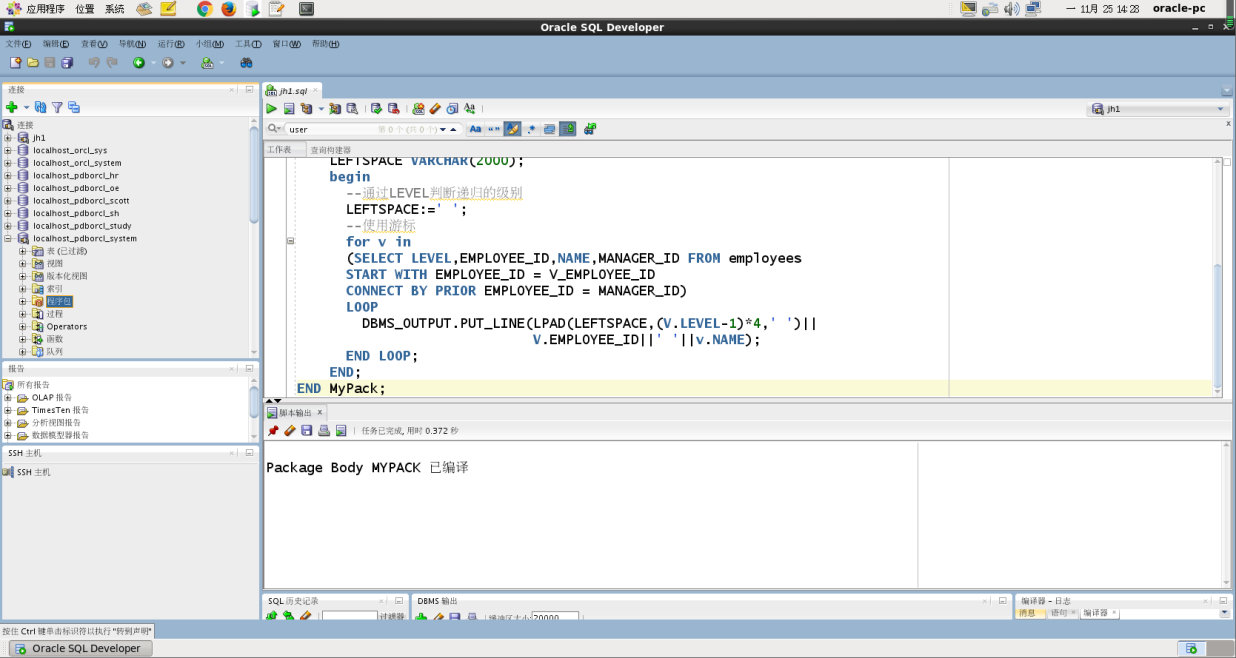
V.EMPLOYEE\_ID||' '||v.NAME);

END LOOP;

END;

END MyPack;

运行结果截图：



1. 用模拟数据进行执行计划分析

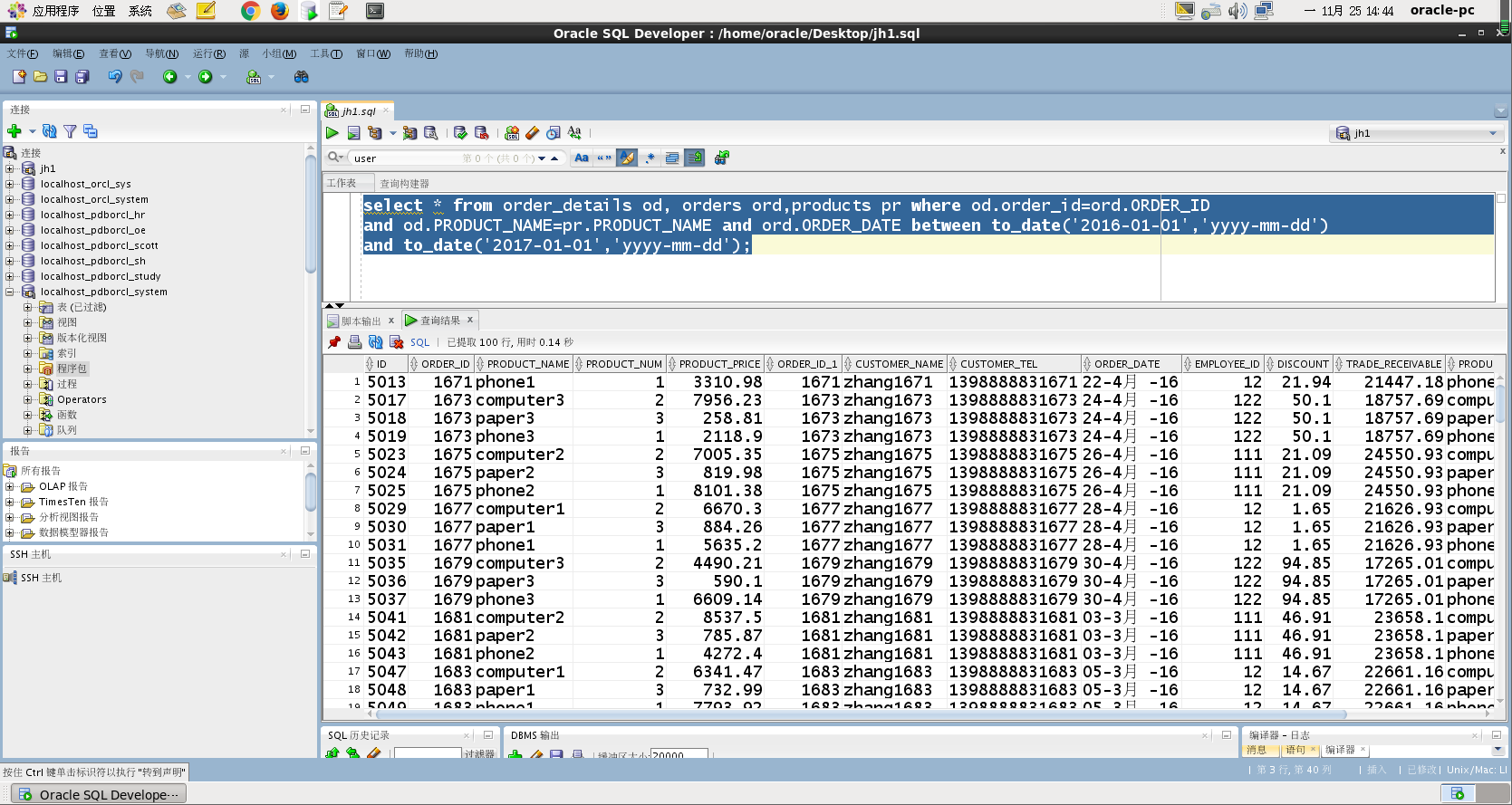
运行代码如下：

select \* from order\_details od, orders ord,products pr where od.order\_id=ord.ORDER\_ID

and od.PRODUCT\_NAME=pr.PRODUCT\_NAME and ord.ORDER\_DATE between to\_date('2016-01-01','yyyy-mm-dd')

and to\_date('2017-01-01','yyyy-mm-dd');

运行结果截图：



**2.7设计自动备份方案或则手工备份方案**

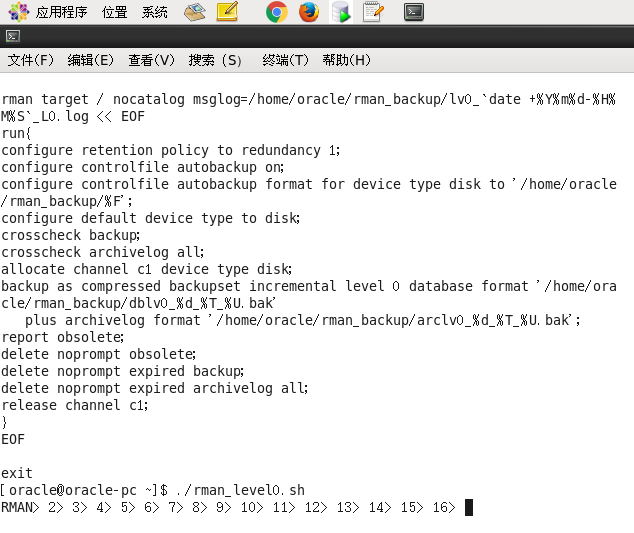
（1）开始全备份

运行代码如下：

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

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

运行结果截图：



（2）查看备份文件

\*.log是日志文件

dblv0\*.bak是数据库的备份文件

arclv0\*.bak是归档日期的备份文件

c-1392946895-20191120-01是控制文件和参数的备份

[oracle@oracle-pc ~]$ cd rman\_backup

[oracle@oracle-pc rman\_backup]$ ls

arclv0\_ORCL\_20191120\_dauhb2fm\_1\_1.bak

c-1392946895-20191120-01

dblv0\_ORCL\_20191120\_d7uhb2ap\_1\_1.bak

dblv0\_ORCL\_20191120\_d8uhb2c6\_1\_1.bak

dblv0\_ORCL\_20191120\_d9uhb2ei\_1\_1.bak

lv0\_20191120-083949\_L0.log

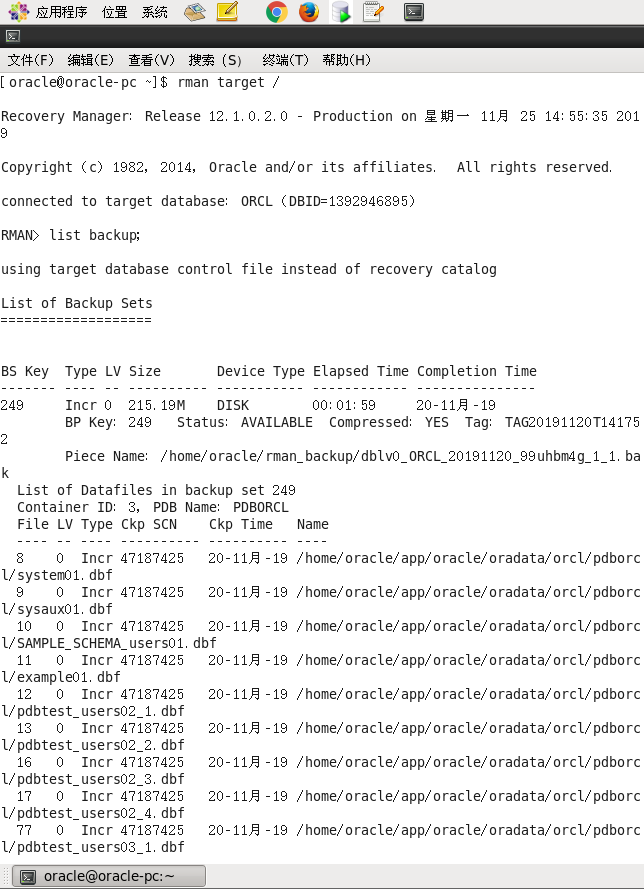
（3）查看备份文件的内容

运行代码如下：

[oracle@oracle-pc ~]$ rman target /

RMAN> list backup;

运行结果截图：



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

（4）备份后修改数据

运行代码如下：

[oracle@oracle-pc ~]$ sqlplus jh/123@pdborcl

SQL> create table t1 (id number,name varchar2(50));

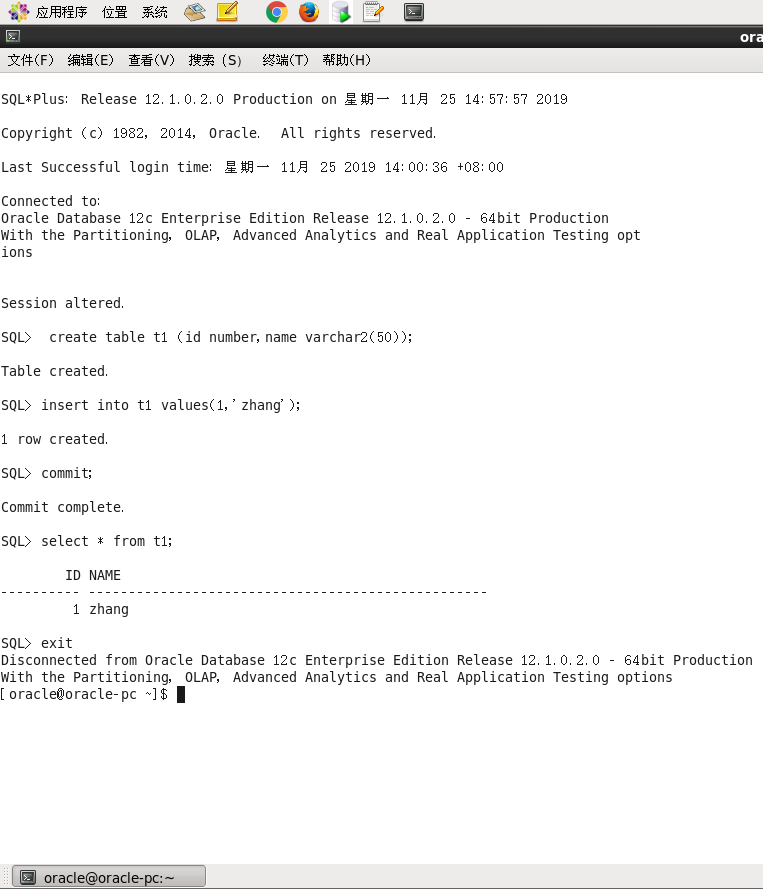
SQL> insert into t1 values(1,'zhang');

SQL> commit;

SQL> select \* from t1;

SQL> exit

运行结果截图：



（5）删除数据库文件，模拟数据库文件损坏

[oracle@oracle-pc~]$ rm/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_user10\_1.pdf

（6）删除数据库文件后修改数据

运行代码如下：

[oracle@oracle-pc ~]$ sqlplus jh/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;

8 /

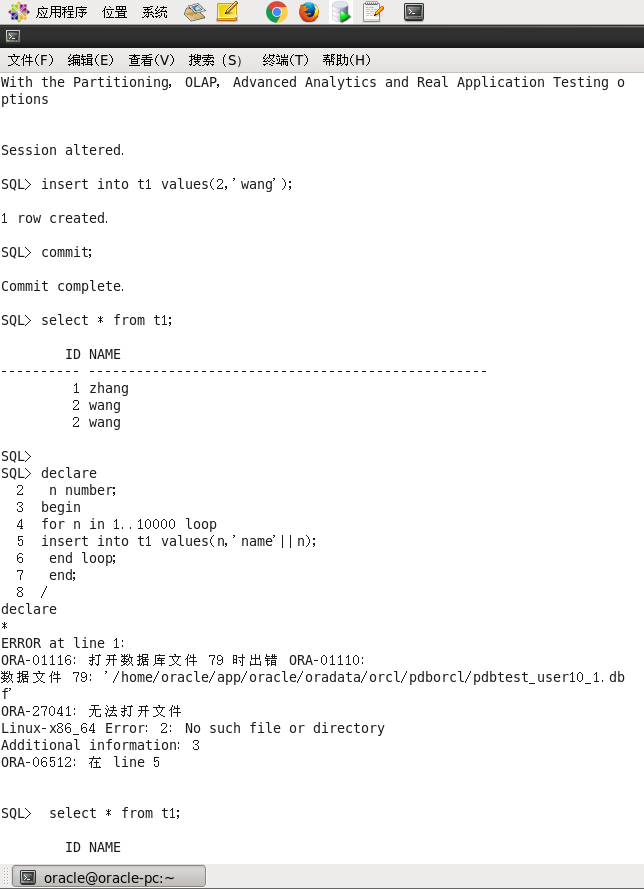
declare

\*

SQL> select \* from t1;

SQL> exit

运行结果截图：



（7）重启损坏的数据库到mount状态

运行代码如下：

[oracle@oracle-pc ~]$ sqlplus / as sysdba

SQL> shutdown immediate

SQL> shutdown abort

SQL> startup mount

SQL> exit

运行结果截图：



（8）开始恢复数据库

运行代码如下：

[oracle@oracle-pc ~]$ rman target /

RMAN> restore database ;

RMAN> recover database;

RMAN> alter database open;

RMAN> exit

运行结果截图：





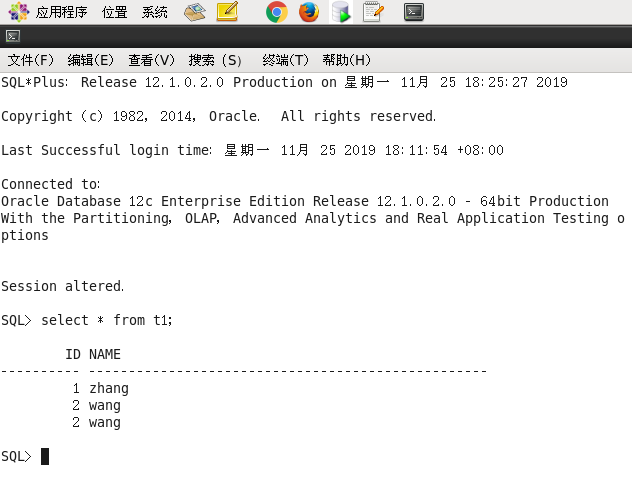
（9）查询数据是否恢复

运行代码如下：

[oracle@oracle-pc ~]$ sqlplus jh/123@pdborcl

SQL> select \* from t1;

运行结果截图：



由以上查询结果可见，数据100%恢复了。

**2.8设计容灾方案。使用两台主机，通过DataGuard实现数据库整体的异地备份。**

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

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;

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

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 hugu(sex varchar);

进入备库：

$sqlplus / as jhdba //连数据库

select \* from jh;

**结束语**

终于把基于oracle的商城订单系统做好了，做得很简单,算是把基本模型给做出来了吧。在做的过程中，不但是在复习之前所学过的知识，同时也是学习新知识的过程。做的时候才发现自己不懂的东西很多，需要从网上和教材上找资料，边做边学。这里简单总结一下做这个系统的流程。首先是对系统需求的分析，罗列出系统涉及的子模块，再做出相应的流程图和数据字典。其次是进行模块的概念设计和逻辑设计，画出E-R图和关系模型。最后是对系统的具体设计，包括数据库的规划，表空间，安全性，表，索引和存储过程的设计，基本的设计流程就是这样。通过完成oracle课程的课程报告，发现和之前学习的sql server 在代码的编写上非常相似，所以写代码花费的时间较少，同时也对sql有了更好地理解。