

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的农产品销售的数据库设计 | | |
| 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 | 喻丹 | 学号 | 201810414203 |
| 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** | | |  |

1. 年 6 月 1 日

基于Oracle的农产品销售数据库设计

实验要求

·自行设计一个信息系统的数据库项目，自拟 某项目 名称。

·设计项目涉及的表及表空间使用方案。至少5张表和5万条数据，两个表空间。

·设计权限及用户分配方案。至少两类角色，两个用户。

·在数据库中建立一个程序包，在包中用PL/SQL语言设计一些存储过程和函数，

·实现比较复杂的业务逻辑，用模拟数据进行执行计划分析。

·设计自动备份方案或则手工备份方案。

·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(可选)。

实验内容

目的：

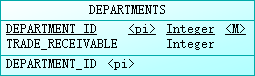
当今信息技术的全面发展和网络大环境的完善,以网上买卖为主的电子商务行业飞快发展。我国是闻名世界的农产品大国,而农业又是国家经济的支柱,网购的迅速发展极大地推动了农产品销售业的成长。线上农产品销售，利用网上购物商城，进行网上销售，与传统销售相比，有助于及时获取农产品供求信息，降低交易成本，提高效率。 这个农产品销售管理系统是将IT技术用于农产品销售信息的管理，它能收集和存储农产品销售的信息，提供更新与检索销售信息档案的接口;来提高农产品销售的工作效率。

实验步骤

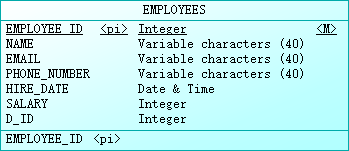
1. 农产品销售管理系统E-R设计

有3个实体：部门、员工、商品

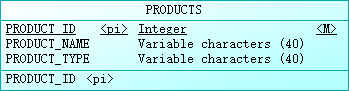
部门



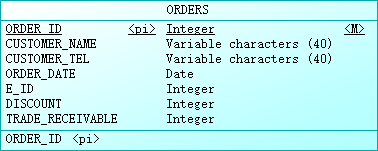
员工



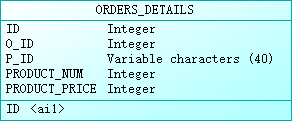
商品



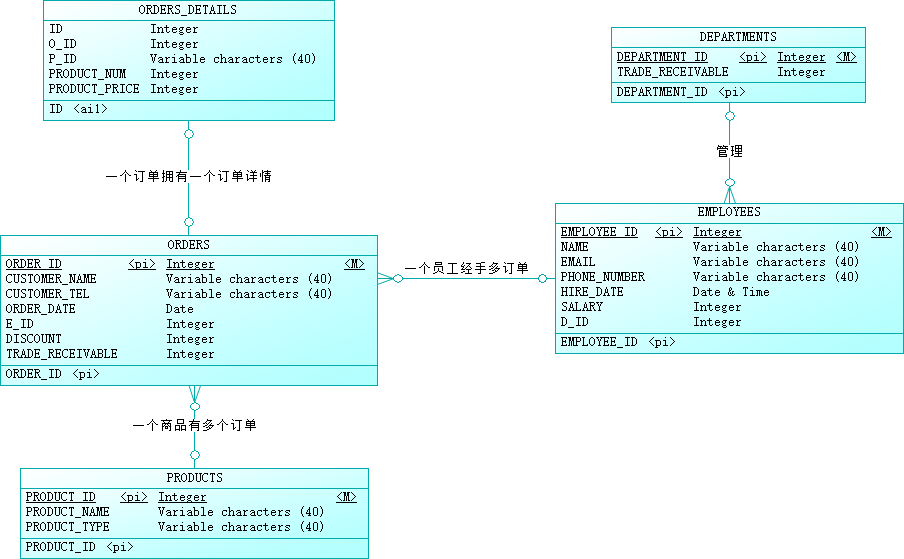
订单表



订单详表



E-R关系总设计



1. 表结构设计

部门表DEPARTMENTS

| **编号** | **字段名** | **数据类型** | **可以为空** | **注释** |
| --- | --- | --- | --- | --- |
| 1 | DEPARTMENT\_ID | NUMBER(6,0) | NO | 部门ID，主键 |
| 2 | DEPARTMENT\_NAME | VARCHAR2(40 BYTE) | NO | 部门名称，非空 |

员工表EMPLOYEES

| **编号** | **字段名** | **数据类型** | **可以为空** | **注释** |
| --- | --- | --- | --- | --- |
| 1 | EMPLOYEE\_ID | NUMBER(6,0) | NO | 员工ID，员工表的主键。 |
| 2 | NAME | VARCHAR2(40 BYTE) | NO | 员工姓名，不能为空 |
| 3 | EMAIL | VARCHAR2(40 BYTE) | YES | 电子信箱 |
| 4 | PHONE\_NUMBER | VARCHAR2(40 BYTE) | YES | 电话 |
| 5 | HIRE\_DATE | DATE | NO | 雇佣日期 |
| 6 | SALARY | NUMBER(8,2) | YES | 月薪，必须>0 |
| 7 | MANAGER\_ID | NUMBER(6,0) | YES | 员工的上司，是员工表EMPOLYEE\_ID的外键，MANAGER\_ID不能等于EMPLOYEE\_ID,即员工的领导不能是自己。主键删除时MANAGER\_ID设置为空值。 |
| 8 | DEPARTMENT\_ID | NUMBER(6,0) | YES | 员工所在部门，是部门表DEPARTMENTS的外键 |

产品表PRODUCTS

| **编号** | **字段名** | **数据类型** | **可以为空** | **注释** |
| --- | --- | --- | --- | --- |
| 1 | PRODUCT\_ID | NUMBER(6,0) | NO | 产品ID，产品表主键 |
| 2 | PRODUCT\_NAME | VARCHAR2(40 BYTE) | NO | 产品名称 |
| 3 | PRODUCT\_TYPE | VARCHAR2(40 BYTE) | NO | 产品类型，只能取值：农产品 |

订单表ORDERS

| **编号** | **字段名** | **数据类型** | **可以为空** | **注释** |  |
| --- | --- | --- | --- | --- | --- |
| 1 | ORDER\_ID | NUMBER(10,0) | NO | 订单编号，主键，值来自于序列：SEQ\_ORDER\_ID |  |
| 2 | CUSTOMER\_NAME | VARCHAR2(40 BYTE) | NO | 客户名称，B树索引 |  |
| 3 | CUSTOMER\_TEL | VARCHAR2(40 BYTE) | NO | 客户电话 |  |
| 4 | ORDER\_DATE | DATE | NO | 订单日期 |  |
| 5 | EMPLOYEE\_ID | NUMBER(6,0) | NO | 订单经手人，员工表EMPLOYEES的外键 |  |
| 6 | DISCOUNT | Number(8,2) | YES | 订单整体优惠金额。默认值为0 |  |
| 7 | TRADE\_RECEIVABLE | Number(8,2) | YES | 订单应收货款，默认为0，Trade\_Receivable= sum(订单详单表.Product\_Num\*订单详单表.Product\_Price)- Discount |  |

订单详单表ORDER\_DETAILS

| **编号** | **字段名** | **数据类型** | **可以为空** | **注释** |
| --- | --- | --- | --- | --- |
| 1 | ID | NUMBER(10,0) | NO | 本表的主键，值来自于序列：SEQ\_ORDER\_DETAILS\_ID |
| 2 | ORDER\_ID | NUMBER(10,0) | NO | 所属的订单号，订单表ORDERS的外键 |
| 4 | PRODUCT\_NAME | VARCHAR2(40 BYTE) | NO | 产品名称, 是产品表PRODUCTS的外键 |
| 5 | PRODUCT\_NUM | NUMBER(8,2) | NO | 产品销售数量，必须>0 |
| 6 | PRODUCT\_PRICE | NUMBER(8,2) | NO | 产品销售价格 |

3.用户权限设计

首先要创建新用户,然后创建表空间，再给其分配表空间，以及其他的权限。

创建表空间

Create Tablespace Users02

datafile

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_1.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,

'/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_users02\_2.dbf'

SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED

EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

--创建用户YUDAN

CREATE USER YUDAN IDENTIFIED BY 123

DEFAULT TABLESPACE "USERS"

TEMPORARY TABLESPACE "TEMP";

-- QUOTAS

ALTER USER YUDAN QUOTA UNLIMITED ON USERS;

ALTER USER YUDAN QUOTA UNLIMITED ON USERS02;

-- ROLES

GRANT "CONNECT" TO YUDAN WITH ADMIN OPTION;

GRANT "RESOURCE" TO YUDAN WITH ADMIN OPTION;

ALTER USER YUDAN DEFAULT ROLE "CONNECT","RESOURCE";

-- SYSTEM PRIVILEGES

GRANT CREATE VIEW TO YUDAN WITH ADMIN OPTION;

--创建用户DP

CREATE USER DP IDENTIFIED BY 123

DEFAULT TABLESPACE "USERS"

TEMPORARY TABLESPACE "TEMP";

-- QUOTAS

ALTER USER DP QUOTA UNLIMITED ON USERS;

ALTER USER DP QUOTA UNLIMITED ON USERS02;

-- ROLES

GRANT "CONNECT" TO DP WITH ADMIN OPTION;

GRANT "RESOURCE" TO DP WITH ADMIN OPTION;

ALTER USER DP DEFAULT ROLE "CONNECT","RESOURCE";

-- SYSTEM PRIVILEGES

GRANT CREATE VIEW TO DP WITH ADMIN OPTION;

4.数据表设计

删除数据表：

建立表之前要先对要建立确已有的表进行删除，先执行语句去匹配，如果匹配到就将那张表删除，如果不存在匹配的表之后就可以直接建立新的数据表。

declare

num number;

begin

select count(1) into num from user\_tables where TABLE\_NAME = 'DEPARTMENTS';

if num=1 then

execute immediate 'drop table DEPARTMENTS cascade constraints PURGE';

end if;

select count(1) into num from user\_tables where TABLE\_NAME = 'EMPLOYEES';

if num=1 then

execute immediate 'drop table EMPLOYEES cascade constraints PURGE';

end if;

select count(1) into num from user\_tables where TABLE\_NAME = 'ORDER\_ID\_TEMP';

if num=1 then

execute immediate 'drop table ORDER\_ID\_TEMP cascade constraints PURGE';

end if;

select count(1) into num from user\_tables where TABLE\_NAME = 'ORDER\_DETAILS';

if num=1 then

execute immediate 'drop table ORDER\_DETAILS cascade constraints PURGE';

end if;

select count(1) into num from user\_tables where TABLE\_NAME = 'ORDERS';

if num=1 then

execute immediate 'drop table ORDERS cascade constraints PURGE';

end if;

select count(1) into num from user\_tables where TABLE\_NAME = 'PRODUCTS';

if num=1 then

execute immediate 'drop table PRODUCTS cascade constraints PURGE';

end if;

select count(1) into num from user\_sequences where SEQUENCE\_NAME = 'SEQ\_ORDER\_DETAILS\_ID';

if num=1 then

execute immediate 'drop SEQUENCE SEQ\_ORDER\_DETAILS\_ID';

end if;

select count(1) into num from user\_sequences where SEQUENCE\_NAME = 'SEQ\_ORDER\_ID';

if num=1 then

execute immediate 'drop SEQUENCE SEQ\_ORDER\_ID';

end if;

select count(1) into num from user\_views where VIEW\_NAME = 'VIEW\_ORDER\_DETAILS';

if num=1 then

execute immediate 'drop VIEW VIEW\_ORDER\_DETAILS';

end if;

SELECT count(object\_name) into num FROM user\_objects\_ae WHERE object\_type = 'PACKAGE' and OBJECT\_NAME='MYPACK';

if num=1 then

execute immediate 'DROP PACKAGE MYPACK';

end if;

end;

/

建立数据表：

创建 DEPARTMENTS表

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 USERS

PCTFREE 10

INITRANS 2

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

创建 EMPLOYEES表

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 USERS

PCTFREE 10

INITRANS 2

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOPARALLEL

)

ENABLE

)

创建 PRODUCTS表

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

)

创建 ORDERS表

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

)

创建 ORDER\_DETAILS表

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

)

5.添加数据

--插入DEPARTMENTS，EMPLOYEES数据

INSERT INTO DEPARTMENTS(DEPARTMENT\_ID,DEPARTMENT\_NAME) values (1,'总经办');

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (1,'赵董事长',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,NULL,1);

INSERT INTO DEPARTMENTS(DEPARTMENT\_ID,DEPARTMENT\_NAME) values (11,'市场部');

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (11,'钱总',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,1,1);

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (111,'孙经理',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,11,11);

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (112,'李经理',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,11,11);

INSERT INTO DEPARTMENTS(DEPARTMENT\_ID,DEPARTMENT\_NAME) values (12,'后勤部');

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (12,'周总',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,1,1);

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (121,'吴经理',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,12,12);

INSERT INTO EMPLOYEES(EMPLOYEE\_ID,NAME,EMAIL,PHONE\_NUMBER,HIRE\_DATE,SALARY,MANAGER\_ID,DEPARTMENT\_ID)

VALUES (122,'郑经理',NULL,NULL,to\_date('2010-1-1','yyyy-mm-dd'),50000,12,12);

--插入products数据

insert into products (product\_name,product\_type) values ('eggs1','土鸡蛋');

insert into products (product\_name,product\_type) values ('eggs2','土鸡蛋');

insert into products (product\_name,product\_type) values ('eggs3','土鸡蛋');

insert into products (product\_name,product\_type) values ('vegetables1','有机蔬菜');

insert into products (product\_name,product\_type) values (' vegetables2','有机蔬菜');

insert into products (product\_name,product\_type) values (' vegetables3','有机蔬菜');

insert into products (product\_name,product\_type) values ('seeds1','种子种苗');

insert into products (product\_name,product\_type) values ('seeds2','种子种苗');

insert into products (product\_name,product\_type) values ('seeds3','种子种苗');

批量插入订单数据，注意ORDERS.TRADE\_RECEIVABLE（订单应收款）的自动计算,注意插入数据的速度

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;

--统计用户的所有表，所需时间很长：2千万行数据，需要1600秒，该语句可选

--dbms\_stats.gather\_schema\_stats(User,estimate\_percent=>100,cascade=> TRUE); --estimate\_percent采样行的百分比

end;

/

6.存储过程和函数设计

create or replace PACKAGE SPM\_Pack IS

/\*

包SPM\_Pack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 SPM\_Pack;

/

create or replace PACKAGE BODY SPM\_Pack IS

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

AS

N NUMBER(20,2);

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

LEFTSPACE:=' ';

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 SPM\_Pack;

/

7.备份方案设计

备份就采用数据库导出和导入的方式：

利用Export可将数据从数据库中提取出来，利用Import则可将提取出来的数据送回到Oracle数据库中去。

１、 简单导出数据（Export）和导入数据（Import）

Oracle支持三种方式类型的输出：

（１）表方式（T方式），将指定表的数据导出。

（２）用户方式（U方式），将指定用户的所有对象及数据导出。

（３）全库方式（Full方式），瘵数据库中的所有对象导出。

数据导入（Import）的过程是数据导出（Export）的逆过程，分别将数据文件导入数据库和将数据库数据导出到数据文件。

２、 增量导出／导入

增量导出是一种常用的数据备份方法，它只能对整个数据库来实施，并且必须作为SYSTEM来导出。在进行此种导出时，系统不要求回答任何问题。导出文件名缺省为export.dmp，如果不希望自己的输出文件定名为export.dmp，必须在命令行中指出要用的文件名。

增量导出包括三种类型：

（１）、“完全”增量导出（Complete）

即备份三个数据库，比如：

exp system/manager inctype=complete file=040731.dmp

（２）、“增量型”增量导出

备份上一次备份后改变的数据，比如：

exp system/manager inctype=incremental file=040731.dmp

（３）、“累积型”增量导出

累计型导出方式是导出自上次“完全”导出之后数据库中变化了的信息。比如：

exp system/manager inctype=cumulative file=040731.dmp

如果数据库遭到意外破坏，数据库管理员可按一下步骤来回复数据库：

第一步：用命令CREATE DATABASE重新生成数据库结构；

第二步：创建一个足够大的附加回滚。

第三步：完全增量导入A：

imp system/manager inctype=RESTORE FULL=y FILE=A

第四步：累计增量导入E：

imp system/manager inctype=RESTORE FULL=Y FILE=E

第五步：最近增量导入F：

imp system/manager inctype=RESTORE FULL=Y FILE=F

数据库导出expdp

step1: system登录，创建目录，授权给自己用户

[student@deep02 ~]$ sqlplus system/123@202.115.82.8/pdborcl

SQL>create or replace directory expdir as '/home/student/pdborcl\_expdir';

SQL> grant read,write on directory expdir to yudan;

授权成功。

SQL>exit

step2: 自己用户备份

[student@deep02 pdborcl\_expdir]$ expdp yudan/123@202.115.82.8/pdborcl directory=expdir dumpfile=yudan.dmp

数据库导入impdp

step1: 删除表

[student@deep02 ~]$ sqlplus yudan/123@pdborcl

SQL> drop table order\_details;

表已删除。

SQL> drop table orders;

表已删除。

SQL> exit

step2: 恢复数据

[student@deep02 ~]$ impdp yudan/123@202.115.82.8/pdborcl directory=expdir dumpfile=yudan.dmp

8.总结

这个实验是一个综合性的实验，让我能对一个管理系统的数据库进行从头到尾的设计，其中运用到了各种在课上和平时实验所学所做，让我对平时所学习到的知识有了很好的巩固作用，理解的也更清晰、透彻、扎实，让我能运用自如。我还对数据库备份进行了深入的学习，虽然备份方式有很多种，但是我觉得最稳妥的还是导出数据，将数据存储在外部，这种物理备份虽然有点麻烦，但是所面临的失去数据的风险更小。数据库的知识还有很多需要去学习，更多地还需要自己去实际的动手练习才能理解的更清楚。