

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的员工管理的数据库设计 | | |
| 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 | 董鑫 | 学号 | 201710414103 |
| 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

## 系统概述

越来越多的网络用户希望能够在网络平台上更多地展现自己的个性，更方便地与人互动交流，在传统的WEB1.0时代，无论是论坛、社区还是公司管理，都试图在这些方面进行努力，随着WEB2. 0时代的到来，我们可以用一个新型的系统来管理公司员工.员工信息管理是一个企业必不可少的部分，随着计算机和计算机知识的普及，员工信息管理系统得到了更大的发展空间，通过对员工信息管理系统的开发，可以提高管理人员和公司职员的工作效率。  
    员工信息管理是一个企业不可缺少的部分，一个良好的员工信息管理系统应该能够为公司职员提供充足的信息和快捷的查询手段。员工信息管理系统对企业加强人员管理和资金管理有着极其重要的作用，由于公司员工数量的增多，对管理人才需求的越来越高，对于如何管理制作如此庞大的员工数量以及人员的绩效显得更为复杂，传统的手工成绩管理不仅工作量大，而且容易出现问题，如：效率低、保密性差，另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。已不能适应时代的发展。  
    随着科学技术的不断提高，计算机科学日渐成熟，其强大的功能已为人们深刻认识，它已进入人类社会的各个领域并发挥着越来越重要的作用。  
    作为计算机应用的一部分，使用计算机对员工信息信息进行管理，具有手工管理所无法比拟的优点。例：检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高管理者管理的效率，也是企业走向科学化、正规化管理与性界物的重变条件。  
因此，开发设计这样一套员工信息管理的数据库系统成为很有必要的事情。

## 实体模型

## 根据应用场景分析，共有5个实体，他们是部门，职位，员工，工资，绩效奖励部门( department)，包括部门ID(department\_id），部门成员（department\_content），员工（staff），员工职位（staff\_post），员工工资（staff\_salary），员工奖金（staff\_bonus），还应该包括管理人员，表示是由某位管理人员发布的，员工ID表示是某一个员工，该属性可以为空，表示

为终极管理员.部门表的实体如图 2-1

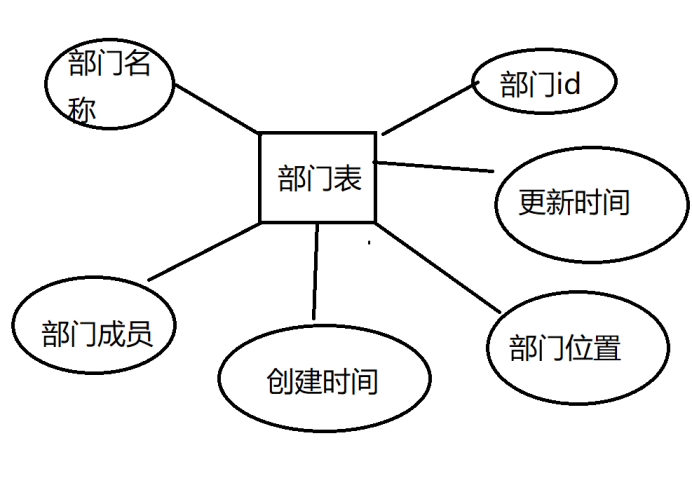


图2-1

职位（post），包括，职位ID（post\_id），职位等级（post\_grade），升职候选（post\_candidate），职位的属性中还应包括部门的ID，员工ID，表示职位属于某个部门，是由某个员工的,职位的ID,表示是某员工的职位，职位的实体如图2-2

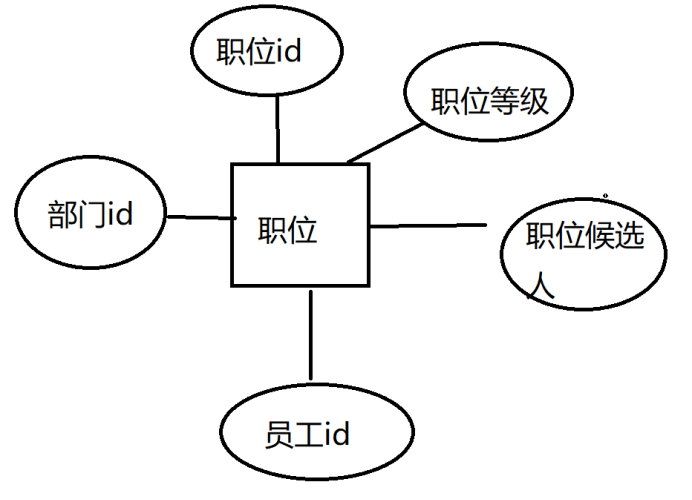
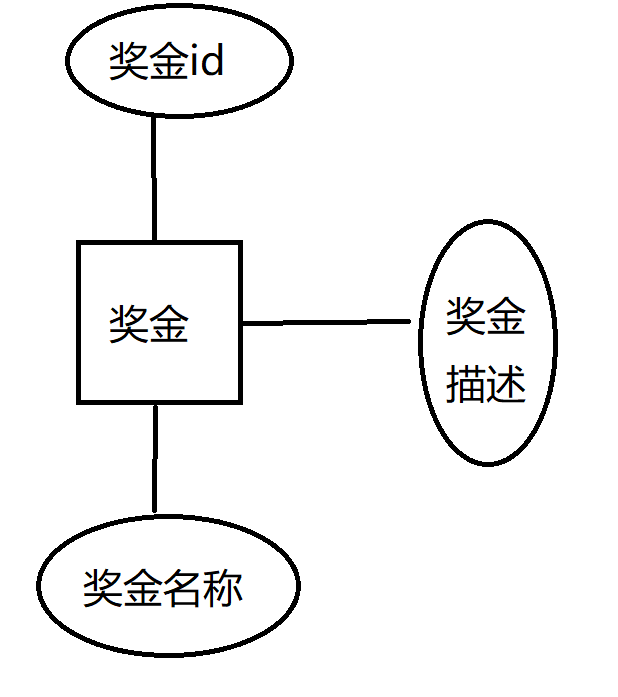


图2-2

奖金（bonus），包括奖金ID（bonus\_id），奖金描述（bonus\_describe），奖金名称（bonus\_name）奖金的实体如图2-3

图2-3

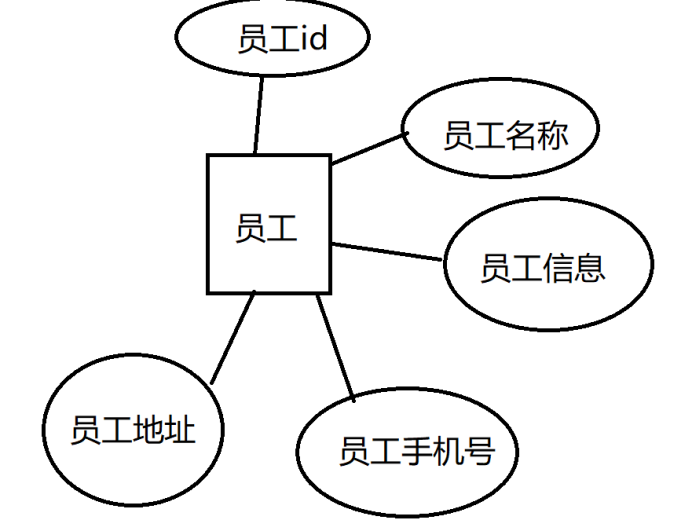
员工（staff），包括员工ID（staff\_id），员工名称（staff\_name），员工信息（staff\_content），用户手机号（staff\_tel），用户头像（staff\_location）用户的实体如图2-4

图2-4

工资（salary）。包括工资数（salary\_number），工资等级（salary\_grade），工资描述（salary\_description）工资的实体如图2-5

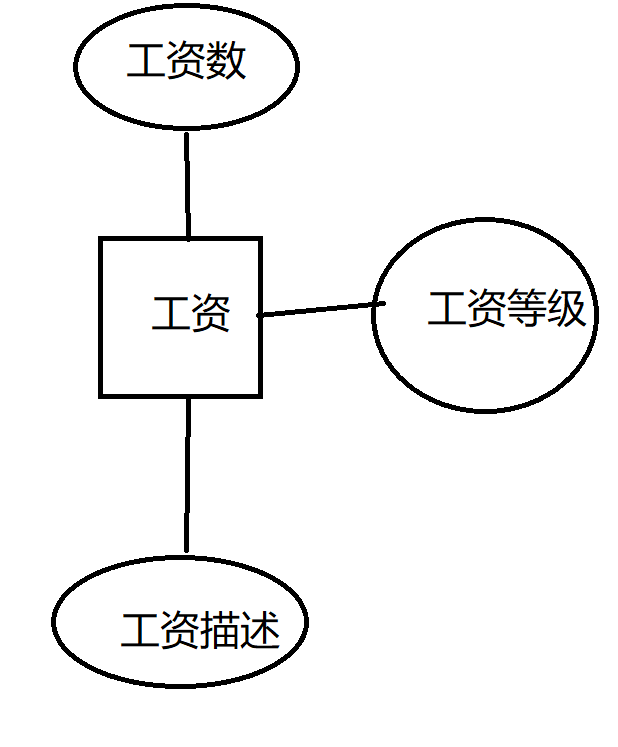
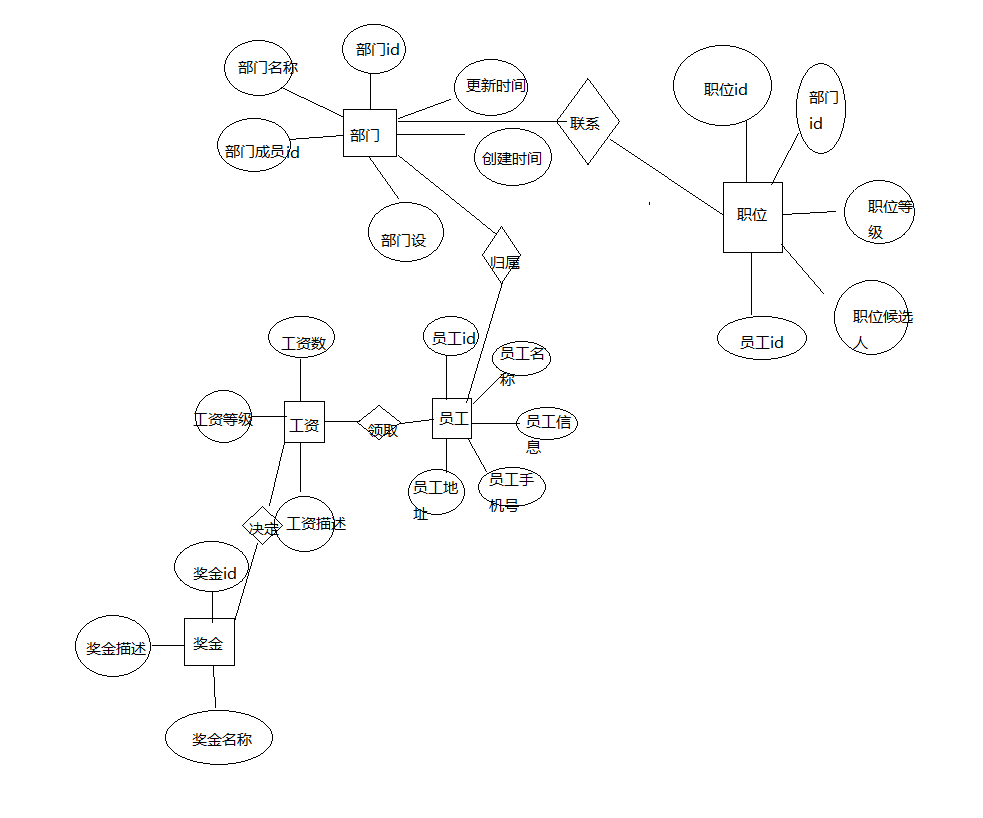


图2-5

## 实体联系模型

每个管理员可以删改员工信息，因此管理员和员工之间就有一个“crud”的联系。在每个员工描述下面都可以发布对该员工的评价，评价会和绩效奖金挂钩.工资数又取决于员工的部门和员工的职位等,工资等级决定了工资的多少,而工资等级又由部门和员工职位决定.所以员工和工资之间就有一个“领取”的关系，在发工资的时候可以为员工设置工资等级，所以工资和员工就有一个“分发”的联系.



## 数据表的设计

E-R模型建立好以后，就可以设计Oracle的关系表了，在独立实体中找出主要的属性设置为主键，在用户表中员工ID（staff\_id）就是主键，在部门中部门ID（department\_id）就是主键，工资表中的主键是工资ID（salary\_id），在职位表中分类ID（psot\_id）就是主键，在系统配置表中奖金号（bonus\_id）是主键。右关系派生出的实体中要加入外键关系，部门对员工就是1对多，即在部门表中添加用户ID属性（staff\_id），职位对员工也是1对多，在职员表中添加员工ID（staff\_id），员工对奖金就是多对多，在员工表中添加奖金ID（salary\_id）

职员（staff），包括员工ID（staff\_id），员工名（staff\_name），员工信息（staff\_message），员工手机号（staff\_tel），员工地址（staff\_location）见表4-1

表4-1 员工表 staff

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| staff\_id | int | No | 员工ID |
| staff\_name | Char（10） | No | 员工名称 |
| staff\_message | Char（64） | No | 员工信息 |
| staff\_tel | Char（11） | No | 员工手机号 |
| staff\_location | Char(32) | No | 员工地址 |

部门(department)，包括部门ID（department\_id），部门成员id（department\_content\_id），创建日期（department\_data），更新日期（department\_update）,部门名称（department\_title），部门设备（goods），见表4-2，注意部门表中的分类成员id可以为空，一个部门可以没有员工.

表4-2部门表department

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| department\_id | Int | No | 部门ID,主键 |
| department\_content\_id | text | No | 部门成员id |
| department\_data | Data | No | 创建日期 |
| department\_title | Char（20） | No | 部门名称 |
| goods | int | Yes | 部门设备 |
| department\_update | Data | yes | 部门更新日期 |
| Department\_staff | Int | yes | 分类ID，是职员表的外键 |
| Staff\_id | Char(64) | No | 员工ID，是staff员工表的外键 |

职位（post），包括，评论ID（post\_id），评论日期（post\_candidate），职员id（staff\_id）,部门id(department\_id)见表4-3

表4-3 职位表 post

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| Post\_id | Int | No | 职位ID，主键 |
| Post\_candidate | Char（100） | No | 职位候选人 |
| Post\_grade | Char(64) | No | 职位级别 |
| Apartment\_id | Int | No | 部门ID,是职位表的外键 |
| Staffid | int | No | 职员ID，是职员表的外键 |

工资（salary），包括工资ID（salary\_id），工资描述（salary\_describe），工资级别（salary\_grade），见表4-4

表4-4 工资表salary

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| Salary\_id | Int | No | 工资ID，主键 |
| Salary\_describe | Char（100） | No | 工资描述 |
| Salary\_grade | Int | No | 工资级别 |

奖金id（bonus\_id）。包括奖金描述（bonus\_description），奖金级别（bonus\_grade），见表4-5

表4-5 绩效表bonus

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 可以为空 | 注释 |
| bonus\_id | Int | No | 奖金id,主键 |
| bonus\_description | Char（10） | No | 奖金描述 |
| bonus\_grade | Char（10） | No | 奖金级别 |

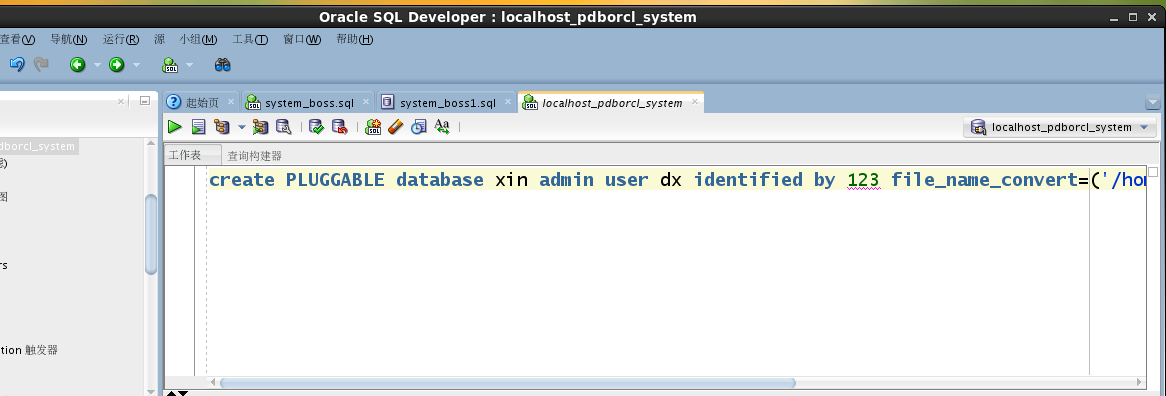
当所有表创建成功后，通过SQL-Developer的Data Modeler工具生成数据库关系图，见图4-6.

## 用户创建与空间分配

表的解构设计好之后，还要考虑用户的空间分配用户

创建了一个名叫xin的数据库，指定了它的存储位置，以及创建了数据库dx的管理员，创建角色system\_xin和用户system\_boss，并授权和分配空间

|  |
| --- |
| create p1uggable database xin admin user dx identified by 123 file\_name\_convert=('/home/orac1e/xin/myscott/',' /home/orac1e/xin/myscott2');|  $ sqlplus system/123@pdborcl  SQL> CREATE ROLE system\_xin;  Role created.  SQL> GRANT connect,resource,CREATE VIEW TO system\_xin;Grant succeeded.  SQL> CREATE USER system\_boss IDENTIFIED BY 123 DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;  User created.  SQL> ALTER USER new\_user QUOTA 50M ON users;  User altered.  SQL> GRANT system\_xin TO system\_boss;Grant succeeded.  SQL> exit |

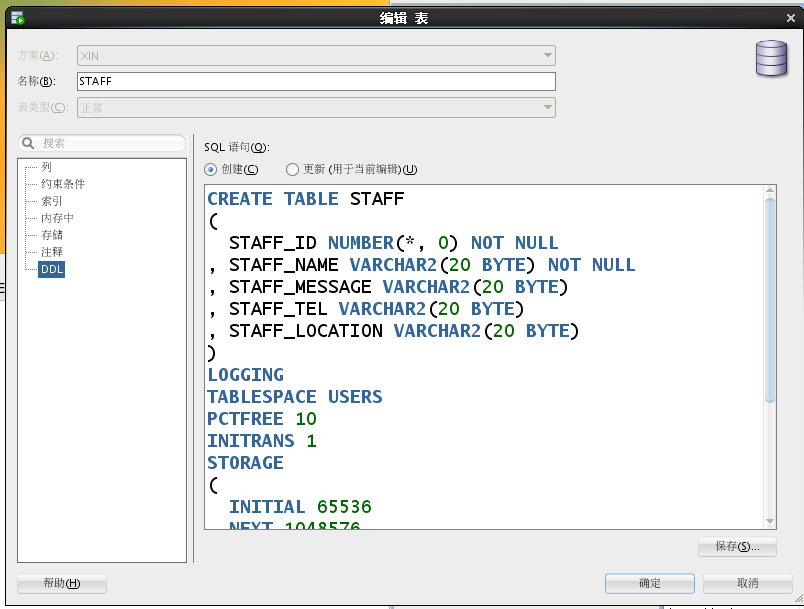


然后对新创建的角色进行select,delete,insert,updata等权限.

此时使用创建好的用户system\_xin登录pdborcl,创建上述的实体表.

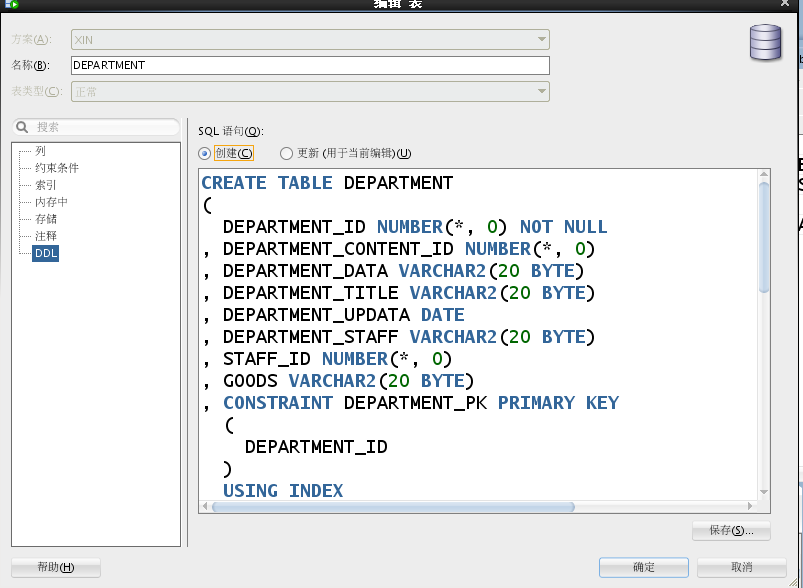
创建员工表

|  |
| --- |
| CREATE TABLE STAFF  (  STAFF\_ID NUMBER(\*, 0) NOT NULL  , STAFF\_NAME VARCHAR2(20 BYTE) NOT NULL  , STAFF\_MESSAGE VARCHAR2(20 BYTE)  , STAFF\_TEL VARCHAR2(20 BYTE)  , STAFF\_LOCATION VARCHAR2(20 BYTE)  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |



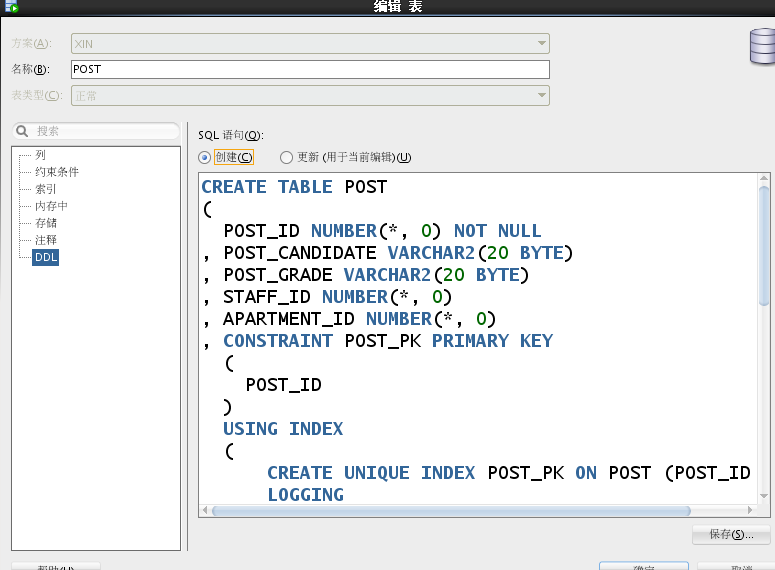
创建部门表

|  |
| --- |
| CREATE TABLE DEPARTMENT  (  DEPARTMENT\_ID NUMBER(\*, 0) NOT NULL  , DEPARTMENT\_CONTENT\_ID NUMBER(\*, 0)  , DEPARTMENT\_DATA VARCHAR2(20 BYTE)  , DEPARTMENT\_TITLE VARCHAR2(20 BYTE)  , DEPARTMENT\_UPDATA DATE  , DEPARTMENT\_STAFF VARCHAR2(20 BYTE)  , STAFF\_ID NUMBER(\*, 0)  , GOODS VARCHAR2(20 BYTE)  , CONSTRAINT DEPARTMENT\_PK PRIMARY KEY  (  DEPARTMENT\_ID  )  USING INDEX  (  CREATE UNIQUE INDEX DEPARTMENT\_PK ON DEPARTMENT (DEPARTMENT\_ID ASC)  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 2  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOPARALLEL  )  ENABLE  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |



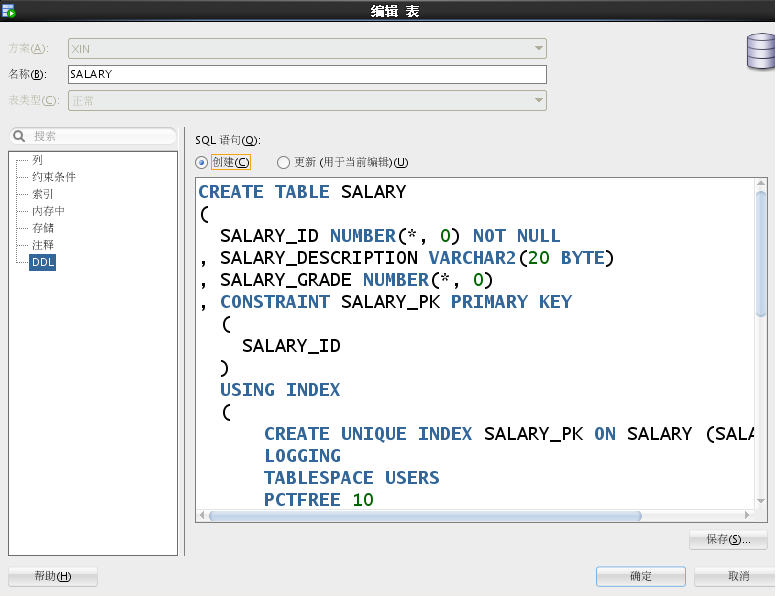
创建职位表

|  |
| --- |
| CREATE TABLE POST  (  POST\_ID NUMBER(\*, 0) NOT NULL  , POST\_CANDIDATE VARCHAR2(20 BYTE)  , POST\_GRADE VARCHAR2(20 BYTE)  , STAFF\_ID NUMBER(\*, 0)  , APARTMENT\_ID NUMBER(\*, 0)  , CONSTRAINT POST\_PK PRIMARY KEY  (  POST\_ID  )  USING INDEX  (  CREATE UNIQUE INDEX POST\_PK ON POST (POST\_ID ASC)  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 2  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOPARALLEL  )  ENABLE  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |



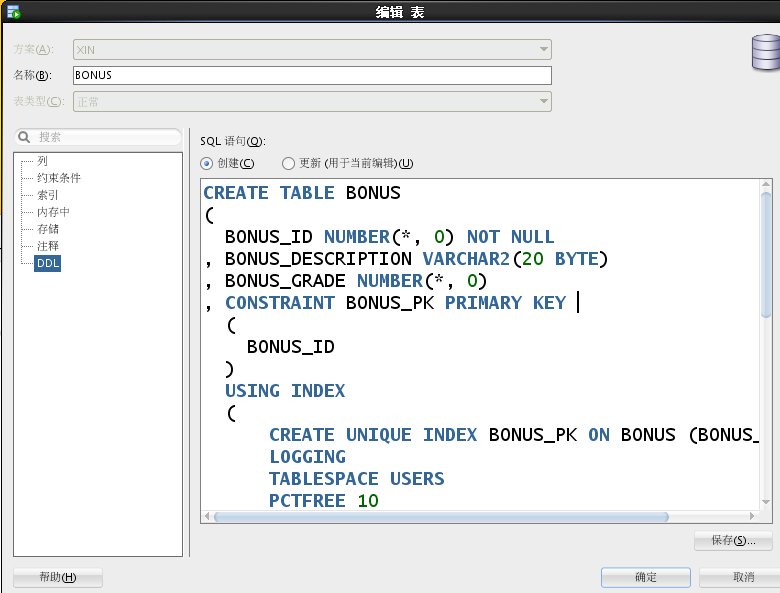
创建工资表

|  |
| --- |
| CREATE TABLE SALARY  (  SALARY\_ID NUMBER(\*, 0) NOT NULL  , SALARY\_DESCRIPTION VARCHAR2(20 BYTE)  , SALARY\_GRADE NUMBER(\*, 0)  , CONSTRAINT SALARY\_PK PRIMARY KEY  (  SALARY\_ID  )  USING INDEX  (  CREATE UNIQUE INDEX SALARY\_PK ON SALARY (SALARY\_ID ASC)  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 2  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOPARALLEL  )  ENABLE  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |



创建绩效表

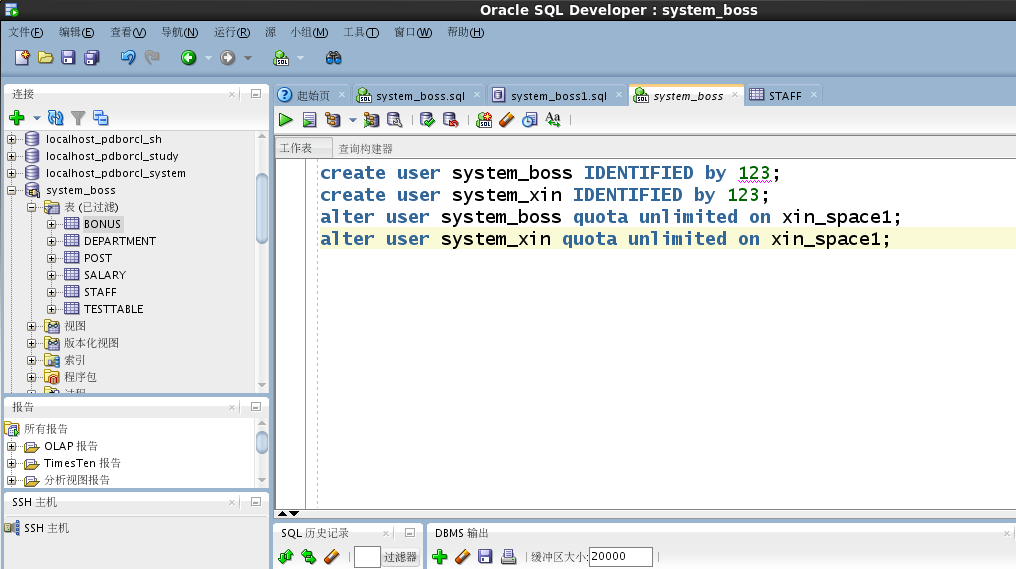
|  |
| --- |
| CREATE TABLE BONUS  (  BONUS\_ID NUMBER(\*, 0) NOT NULL  , BONUS\_DESCRIPTION VARCHAR2(20 BYTE)  , BONUS\_GRADE NUMBER(\*, 0)  , CONSTRAINT BONUS\_PK PRIMARY KEY  (  BONUS\_ID  )  USING INDEX  (  CREATE UNIQUE INDEX BONUS\_PK ON BONUS (BONUS\_ID ASC)  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 2  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOPARALLEL  )  ENABLE  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |



然后分别对员工表,职位表,工资表,绩效表,部门表进行数据插入.其中职位表设置有:小白,实习生,程序员,项目经理,架构师,技术总监,总经理.工资表中设置了

### 创建用户,授权,插入数据，以及对分区，授权的验证

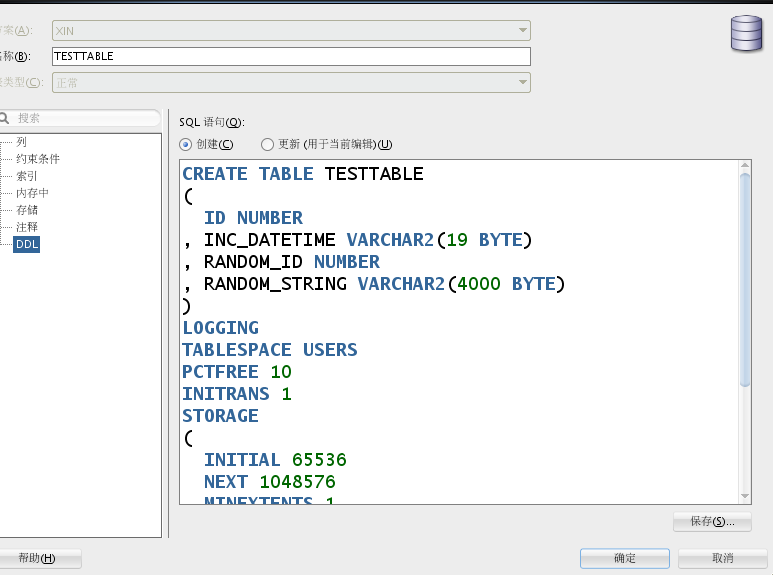
|  |
| --- |
| create user system\_boss IDENTIFIED by 123;  create user system\_xin IDENTIFIED by 123;  alter user system\_boss quota unlimited on xin\_space1;  alter user system\_xin quota unlimited on xin\_space1; |



由于要插入至少五万条数据,因此先创建一个测试表进行,进行测试数据的插入.

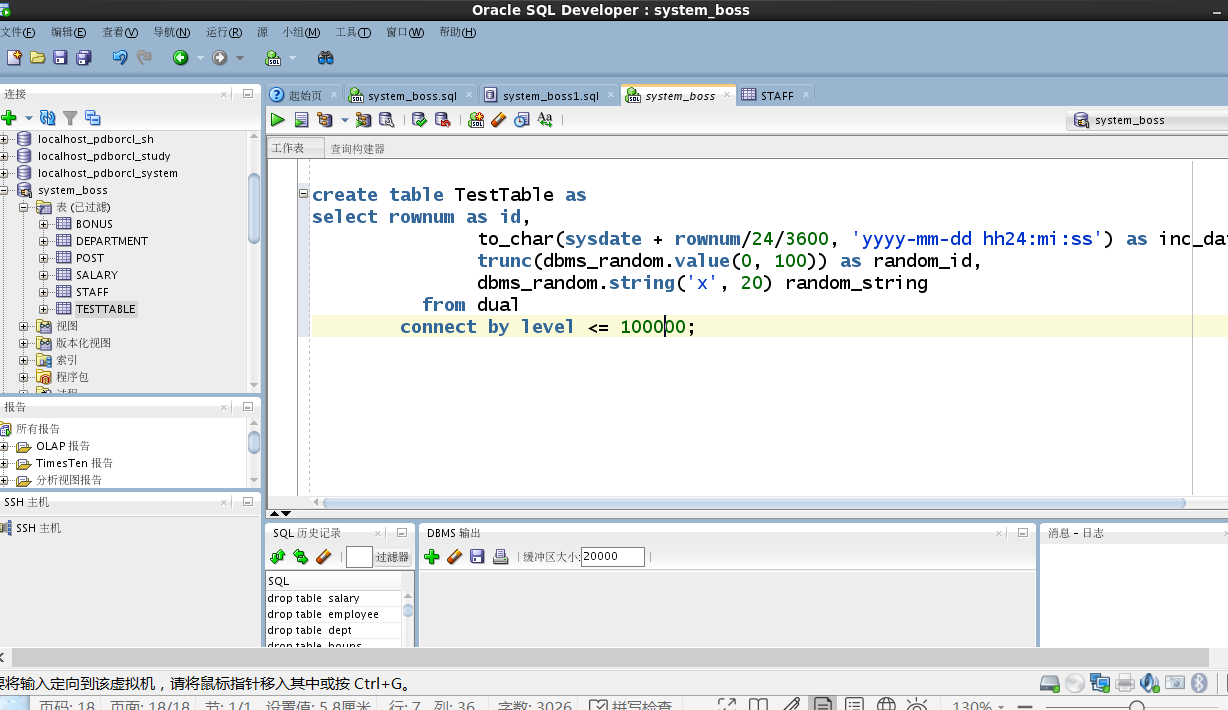
创建测试表

|  |
| --- |
| CREATE TABLE TESTTABLE  (  ID NUMBER  , INC\_DATETIME VARCHAR2(19 BYTE)  , RANDOM\_ID NUMBER  , RANDOM\_STRING VARCHAR2(4000 BYTE)  )  LOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 65536  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS  NO INMEMORY  NOPARALLEL; |

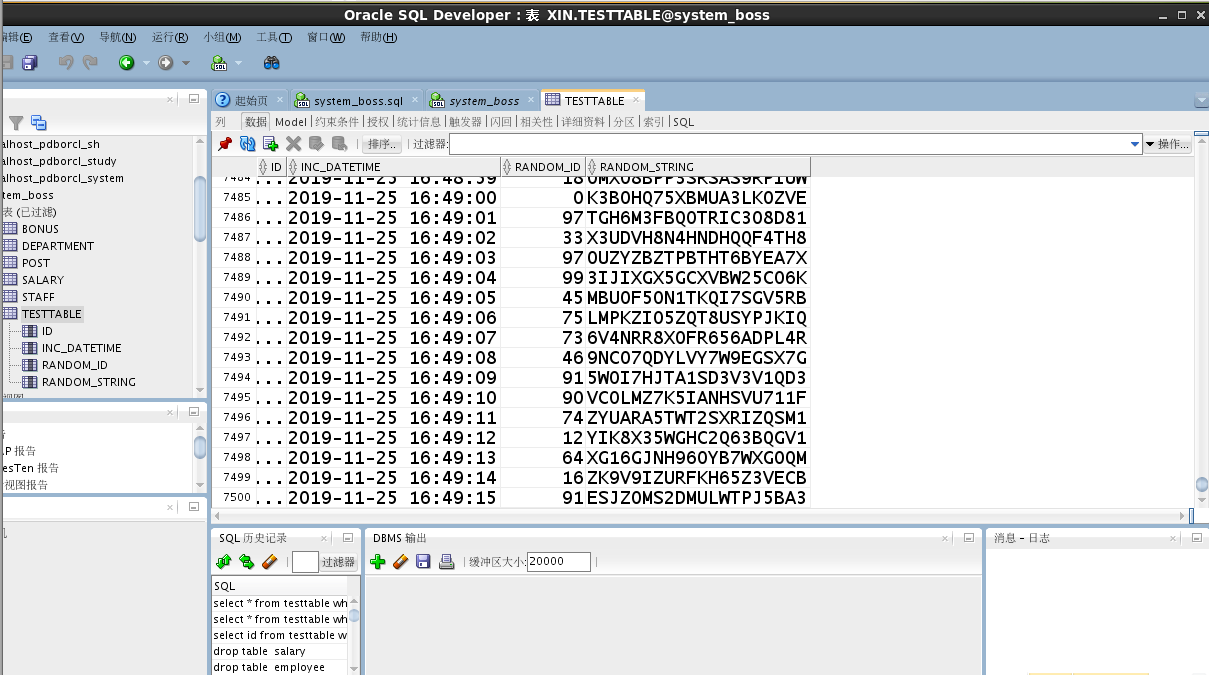


然后向测试表TESTTABLE中插入10万数据进行测试.

|  |
| --- |
| create table TestTable as  select rownum as id,  to\_char(sysdate + rownum/24/3600, 'yyyy-mm-dd hh24:mi:ss') as inc\_datetime,  trunc(dbms\_random.value(0, 100)) as random\_id,  dbms\_random.string('x', 20) random\_string  from dual  connect by level <= 100000; |

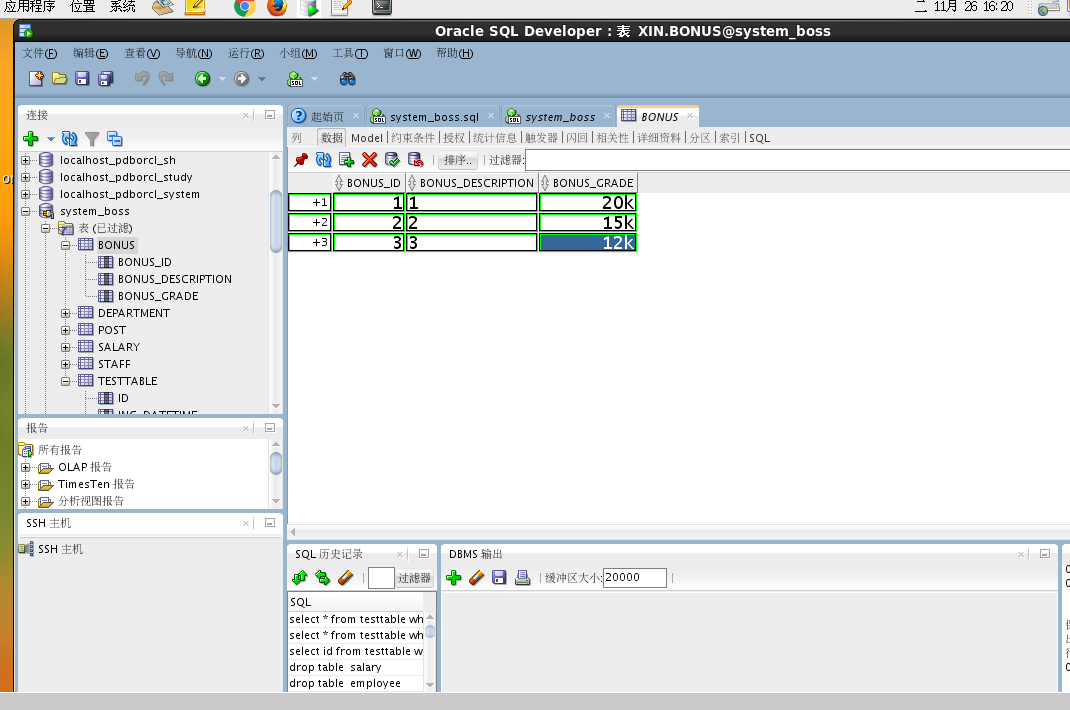


此时TESTTABLE表中已经插入随机测试数据

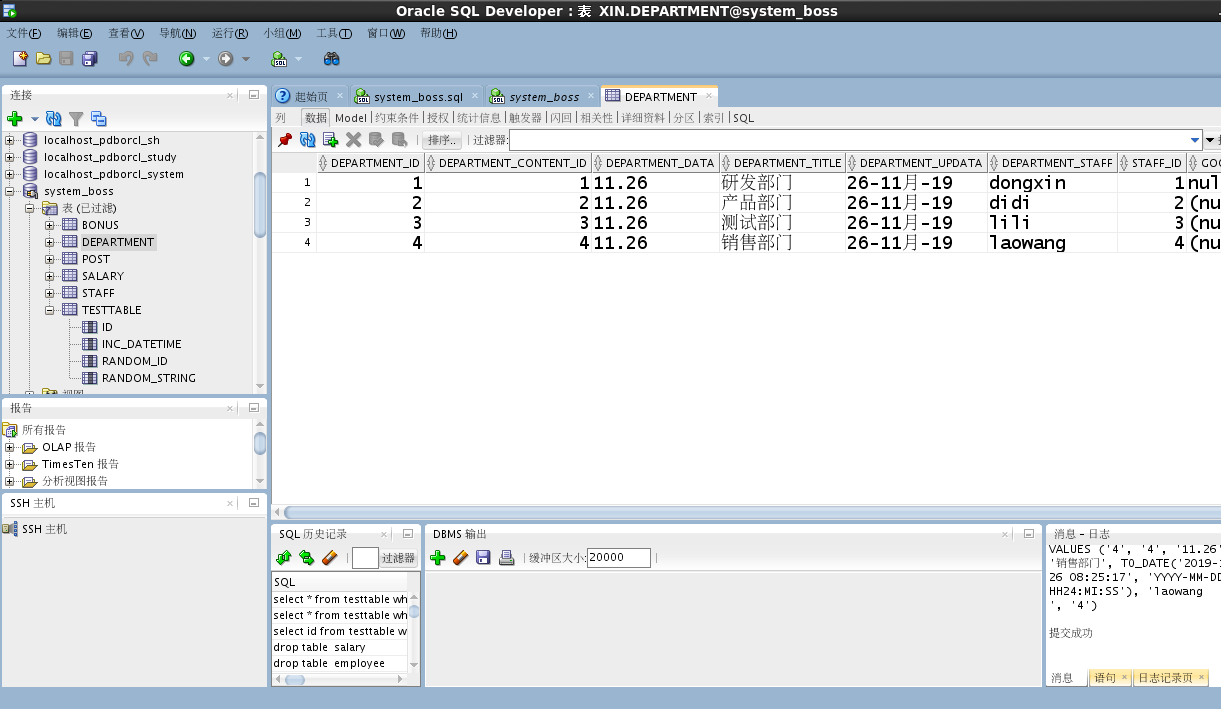


之后开始对五张实体表进行数据插入

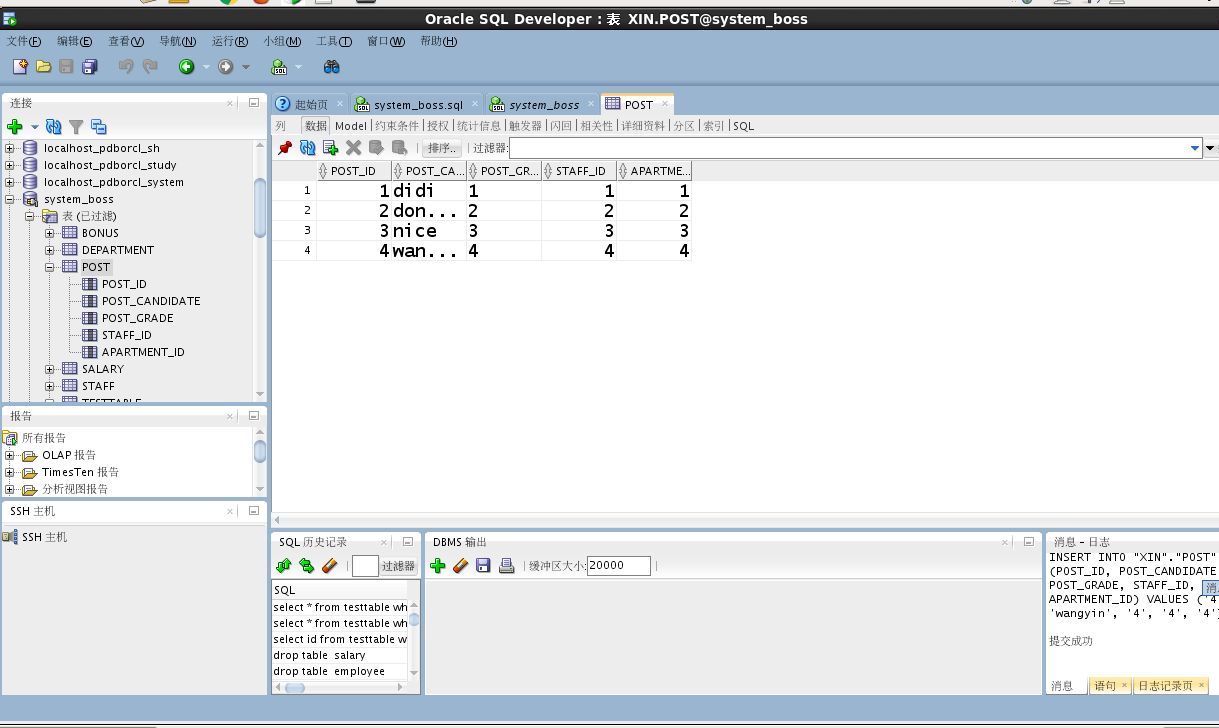
奖金表数据分为三个奖项,一等奖金20k,二等奖15k,三等奖金12k.



然后是对部门表进行数据的插入,总共有测试部门,研发部门,产品部门,销售部门



然后对职位表post 进行数据插入



## 程序包的建立

建立一个程序包里面有一个函数和一个存储过程,用来统计员工的工资和奖金.

|  |
| --- |
| create or replace PACKAGE xinPack IS  FUNCTION Get\_TOTAL(dt1 char,dt2 char) RETURN NUMBER;  PROCEDURE get\_people(dt1 char,dt2 char);  END xinPack;  /  create or replace PACKAGE BODY xinPack IS  FUNCTION Get\_TOTAL(dt1 char,dt2 char) RETURN NUMBER  AS  N NUMBER;  BEGIN  select sum(price) into N from REGISTRATION\_FORM where REGISTRATION\_DATE >= to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')  and REGISTRATION\_DATE <= to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');  RETURN N;  END;  -- 得到每个部门的工资总额  PROCEDURE get\_apartment\_total(dt1 char,dt2 char)  AS  a1 NUMBER;  b1 NUMBER;  c1 NUMBER;  d1 NUMBER;  e1 NUMBER;  f1 NUMBER;  cursor cur is  select \* from REGISTRATION\_FORM where REGISTRATION\_DATE >= to\_date(dt1,'yyyy-mm-dd hh24:mi:ss')  and REGISTRATION\_DATE <= to\_date(dt2,'yyyy-mm-dd hh24:mi:ss');  begin  a1 := 0;  b1 := 0;  c1 := 0;  d1 := 0;  --使用游标  for v in cur  LOOP  if v.REGISTRATION\_CLASS = '研发部门'  then a1 := a1 + 1;  elsif v.REGISTRATION\_CLASS = '产品部门'  then b1 := b1 + 1;  elsif v.REGISTRATION\_CLASS = '测试部门'  then c1 := c1 + 1;  elsif v.REGISTRATION\_CLASS = ‘销售部门'  then d1 := d1 + 1;  end if;  end;  END xinPack; |

## 系统的定时备份

对人员管理系统的数据库做定时备份:

编写rman增量备份脚本

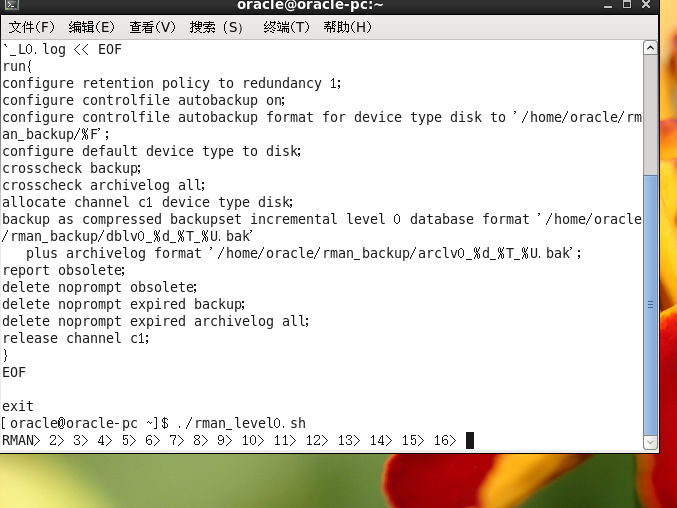
|  |
| --- |
| #rman\_level1.sh  #!/bin/sh  export NLS\_LANG='SIMPLIFIED CHINESE\_CHINA.AL32UTF8'  export ORACLE\_HOME=/home/oracle/xin/dbhome\_1  export ORACLE\_SID=orcl  export PATH=$ORACLE\_HOME/bin:$PATH  rman target / nocatalog msglog=/home/oracle/xin/rman\_backup/lv1\_`date +%Y%m%d-%H%M%S`\_L0.log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

进行一遍测试

开始全备份

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

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

备份之后进行模拟数据库损坏,删除数据库文件.

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

然后进行数据库的恢复.

sqlplus / as sysdba

shutdown immediate

shutdown abort

startup mount

rman target /

之后可以删除备份集,也可以保留.

## 容灭方案设计dataGuard

先搭建一套oracle rac环境,

设置本地hosts

|  |
| --- |
| vi /etc/hosts  127.0.0.1 localhost    # Public Network - (eth0)    192.168.186.11 rac1    192.168.186.12 rac2    # Private Interconnect - (eth1)    192.168.40.11 rac1-priv    192.168.40.12 rac2-priv    # Public Virtual IP (VIP) addresses - (eth0:1)    192.168.186.13 rac1-vip    192.168.186.14 rac2-vip    # Single Client Access Name (SCAN)    192.168.186.10 racscan    # Private Storage Network    192.168.186.15 storage    192.168.40.15 storage |

安装补丁包:

|  |
| --- |
| yum -y install gcc elfutils-libelf-devel gcc-c++ libaio-devel |

创建一个用户和表空间:数据文件存放在/home/oracle/xin/data中

然后进行数据盘的挂载,关闭oracle rac集群服务,停止读写数据,搭载另一套容灾oracle rac环境,安装集群软件Grid Infrastructure,数据库软件Database,注意不要创建数据库实例.

复制生产环境下的/home/xin/data 中的两个挂载lun,把备份的两个lun挂载到容灾环境的系统下.

在容灾环境的节点上修改权限

|  |
| --- |
| chown grid:oinstall /dev/raw/raw4  chown grid:oinstall /dev/raw/raw5 |

在容灾环境的两个节点上分别创建好目录和授权

|  |
| --- |
| mkdir -p /home/oracle/xin/pratice/admin/  chown -R oracle:oinstall /home/oracle/xin/pratice/admin/ |

然后分别配置两个节点

|  |
| --- |
| vi/home/oracle/xin/pratice/admin/initracdb1.ora  添加以下内容：  -------------------------------------------------------------  SPFILE='+DATA/racdb/spfileracdb.ora'  -------------------------------------------------------------  vi /home/oracle/xin/pratice/admin/initracdb2.ora  添加以下内容：  -------------------------------------------------------------  SPFILE='+DATA/racdb/spfileracdb.ora'  ------------------------------------------------------------- |

在两个节点下进行grid执行组挂载命令

|  |
| --- |
| su - grid  sqlplus / as sysasmalter diskgroup data mount; |

然后启动实例,把实例添加到集群配置中:

|  |
| --- |
| su - oracle  srvctl status database -d racdb  srvctl add database -d racdb -o dbhome\_1  srvctl config database -d racdb  srvctl add instance -d racdb -i racdb1 -n rac1  srvctl add instance -d racdb -i racdb2 -n rac2  srvctl config database -d racdb  srvctl start database -d racdb  srvctl status database -d racdb |