2016年11月17日,星期四

作者：石国文

名词

- 表空间：存放数据的逻辑单位，相当于链接
- 数据字典：提供一些视图显示系统信息
- **DDL**：数据定义语言
- **DML**：数据操作语言，需要commit提交
- **DCL**：数据控制语言
- **SQL**：一种语言，ANSI标准，关键字不能缩写，使用语句控制数据库中的表的定义信息和表中的数据，例如：select，insert，delete，update
- **SQL*PLUS**：一种环境，Oracle的特性之一，关键字可以缩写，命令不能改变数据库中的数据的值，集中运行。例如：describe，edit，change，format，column
- SQL的类型
    - 1. DML(Data Manipulation Language 数据操作语言): insert update delete select
    - 2. DDL(Data Definition Language 数据定义语言): create/alter/drop/truncate table,create/drop view,sequence,index,synonym(同义词)
    - 3. DCL(Data Control Language 数据控制语言）： grant(授权) revoke（撤销权限）

小功能

- spool录屏：spool c:/a.txt
    - 结束录屏：spool off
- 显示行宽：show linesize
    - 设置行宽：set linesize 300;
- 设置列宽，a8表示8位字符串，9999表示4位数字。
    - col 列名 for a8
    - col sal for 9999
- 设置一页显示的大小：set pagesize 20
- 执行上一条语句：/
- 断开连接disconn和连接conn
    - disconn

    - conn 用户名/密码
- 编辑上一条语句：ed
- 退出：exit
- 当sql语句写错了怎么去修改

```
SQL> select empno,ename,sal ,sal*12
  2  form emp;
form emp
     *
第 2 行出现错误:
ORA-00923: 未找到要求的 FROM 关键字
```

c命令来修改

```
SQL> --c命令 change
```

首先，敲 2 定位到错误的那行

```
SQL> 2
  2* form emp
```

c表示 change 修改，/错误的/正确的

```
SQL> c /form/from
  2* from emp
```

    - 然后敲/执行这条 sql

- 清屏：host cls或clear screen
- a命令：append 追加
    - 在上一条语句后面追加'desc'：a desc
- SQL执行时间的开关：set timing on，set timing off
- 开启控制台输出：set serveroutput on

启动服务

- 1、 Linux下Oracle的启动过程

```
lsnrctl start：启动监听
//sqlplus / as sysdba：登录
sqlplus /nolog
sqlplus sys/sys as sysdba
startup
```

- 2、windows下Oracle的启动过程

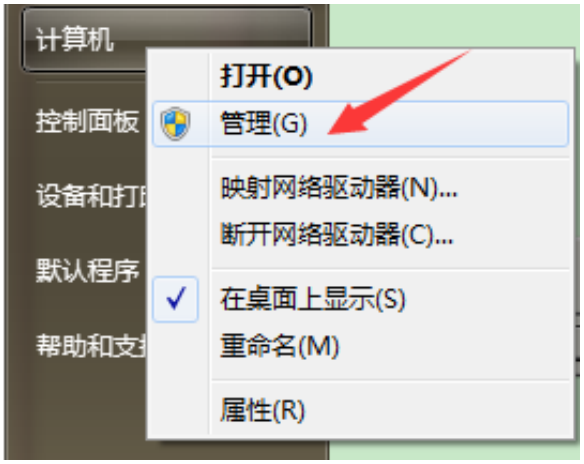
```
lsnrctl start
oradim -starup -sid 数据库实例名称
```

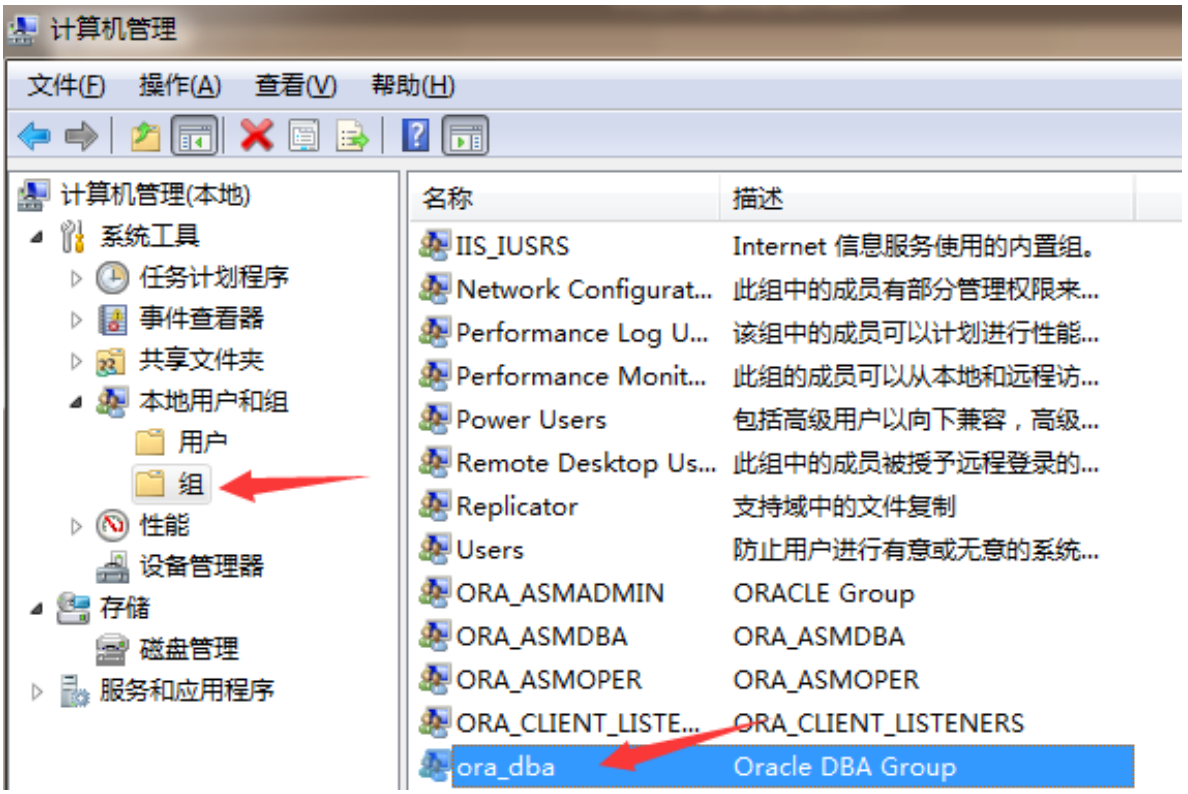
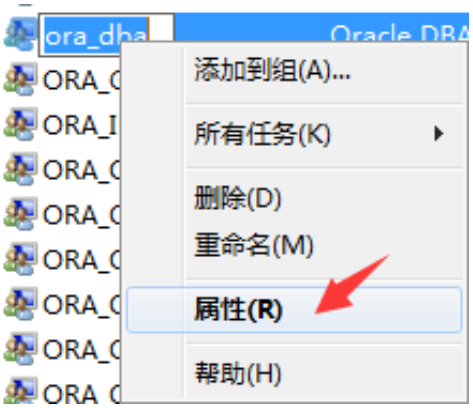
SQL优化原则

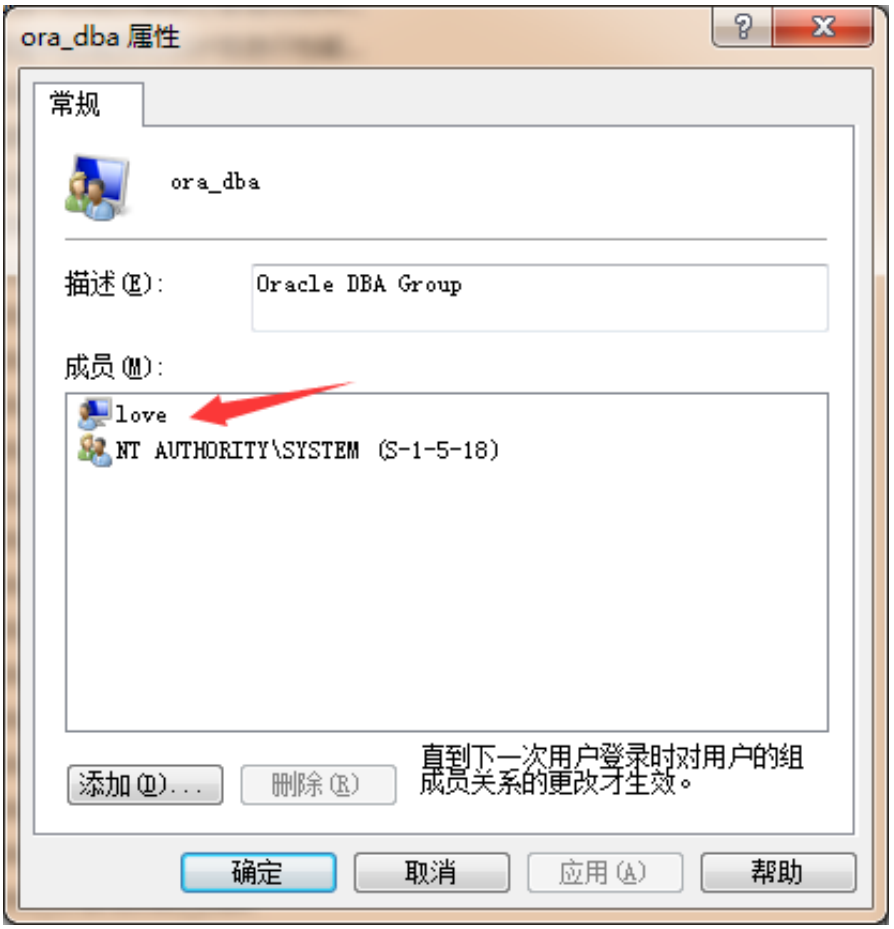
- 尽量使用列名代替*号
- where顺序： 右--》 左
- 再where和having都可以使用时，尽量使用where
- 理论上，尽量使用多表查询
- 尽量不要使用集合运算

登录

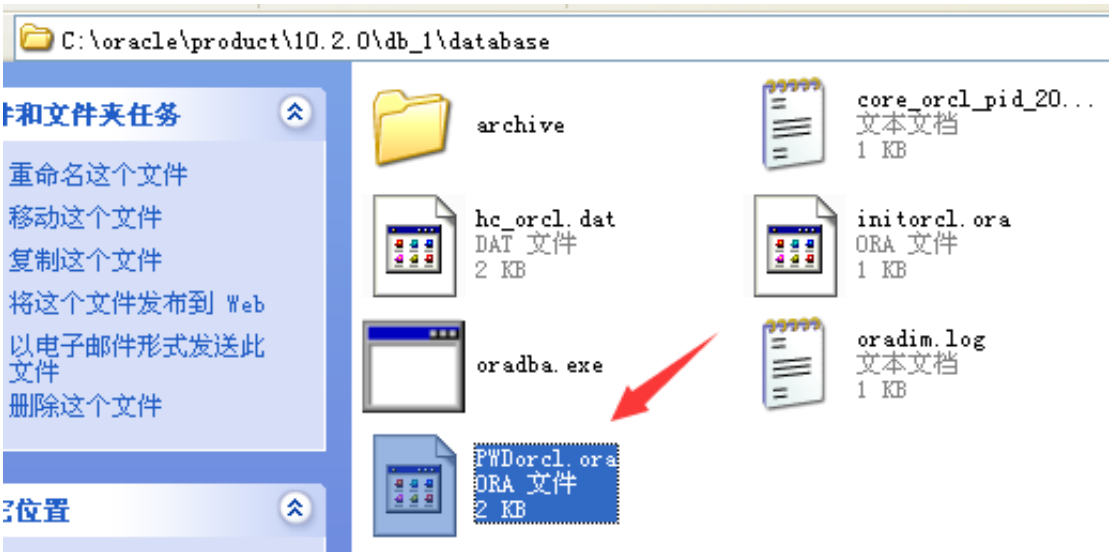
- 1、本地登录
  - 管理员登录：sqlplus / as sysdba
  - 普通用户登录：sqlplus scott/scott
- 2、远程登录
  - sqlplus scott/scott@192.168.199.192:1521/orcl
- 3、三种登录验证机制
  - 操作系统验证



- 
计算机管理
文件(F) 操作(A) 查看(V) 帮助(H)
计算机管理(本地)
系统工具
任务计划程序
事件查看器
共享文件夹
本地用户和组
用户
组
性能
设备管理器
存储
磁盘管理
服务和应用程序
名称
描述
IIS_IUSRS
Internet 信息服务使用的内置组。
Network Configurat...
此组中的成员有部分管理权限来...
Performance Log U...
该组中的成员可以计划进行性能...
Performance Monit...
此组的成员可以从本地和远程访...
Power Users
包括高级用户以向下兼容，高级...
Remote Desktop Us...
此组中的成员被授予远程登录的...
Replicator
支持域中的文件复制
Users
防止用户进行有意或无意的系统...
ORA_ASMADMIN
ORACLE Group
ORA_ASMDBA
ORA_ASMDBA
ORA_ASMOPER
ORA_ASMOPER
ORA_CLIENT_LISTE...
ORA_CLIENT_LISTENERS
ora_dba
Oracle DBA Group
- 
添加到组(A)...
所有任务(K)
删除(D)
重命名(M)
属性(R)
帮助(H)



- 密码文件验证
  - 密码文件位置



  - 重新生成密码文件

```
C:\Documents and Settings\Administrator>orapwd file=C:\oracle\product\
1\database\PWDorcl.ora password=sys entries=10
```

- 数据库验证
  - 登录时，系统会查询数据库验证用户名和密码是否正确

用户管理

- 默认用户
  - **sys**：超级管理员
  - **system**：普通管理员
  - **scott**：普通用户
  - **hr**：普通用户
- 显示当前用户：show user;
- 创建用户：create user lisi identified by lisi;
  - 创建用户的详细语句

```
create user zhangsan
identified by zhangsan
default tablespace users
temporary tablespace temp
quota 50M on users;
default tablespace：默认表空间
temporary tablespace：临时表空间，不能在临时表空间上使用限额
```

- 修改用户密码：alter user lisi identified by 密码;
- 删除用户：drop user 用户名;或者drop user 用户名 cascade;(这将会删除与用户相关联的表)
- 查询当前有多少个特权用户：select * from v$pwfile_users;
- 限制用户
  - 锁定用户：alter user lisi account lock;
  - 解锁用户：alter user lisi account unlock;
  - 使用户口令即刻失效：alter user lisi password expire;

权限管理

权限

系统权限：系统管理员才能授权

- 当前用户拥有哪些权限：**select * from user_*sys*_privs;**
- 授权
  - 给所有用户授予权限：grant create session to public

    - 授权 登录 权限：grant create session to lisi;
    - 授权 创建表 权限：grant create table to lisi;
    - 授权 表空间 权限：grant unlimited tablespace to lisi;
    - 授权 同时给用户授予多个权限：grant create table,unlimited tablespace to wangwu;
    - 授权 同时给多个用户授予权限：grant create table,unlimited tablespace to wangwu,lisi;
- 撤销权限
    - 撤销 用户创建表 权限：revoke create table from lisi;

对象权限：谁拥有谁授权

- 显示当前用户拥有的表权限：**select * from user_*tab*_privs;**
- 当前用户没有权限访问wangwu的表：select * from wangwu.mytab;
- 用户wangwu给用户lisi授权查询mytab表的权限：grant select on mytab to lisi;
- 用户wangwu给用户lisi授权操作mytab表的所有权限：grant all on mytab to lisi;
- 用户wangwu给lisi用户撤销操作mytab表的所有权限：revoke all on mytab from lisi;
- 用户wangwu给所有用户授予查询权限：grant select on mytab to public;
- 对象权限控制到列上：grant update on mytab to lisi;
- 查询当前用户操作列的权限：**select * from user_*col*_privs;**
    - 注意：查询和删除不能控制到列

权限传递

- 管理员赋予lisi用户alter any table权限，并且能够给其它用户授予该权限：grant alter any table to lisi with admin option;
- 用户lisi得到管理员的授权后可以继续给wangwu授权，前提是管理员给lisi授权时加上with admin option：grant alter any table to wangwu;
- lisi用户给wangwu授权select权限，并且wangwu可以继续给其它用户授权：grant select on mytable to wangwu with grant option;

角色管理

角色：权限的集合

- 创建角色：create role myrole;
- 给角色授权：grant create session to myrole;
- 把角色授权给wanger用户：grant myrole to wanger;

- 删除角色：drop role myrole;
- 有些权限比较特殊，不能授予给角色：grant unlimited tablespace to myrole;

基本查询

- 查询当前用户下有哪些表：select * from tab;
- 查看指定表的结构：desc emp;
- 使用别名：select empno as "员工号",ename "姓名",sal 月薪,sal*12 "年薪",comm "奖金" from emp;

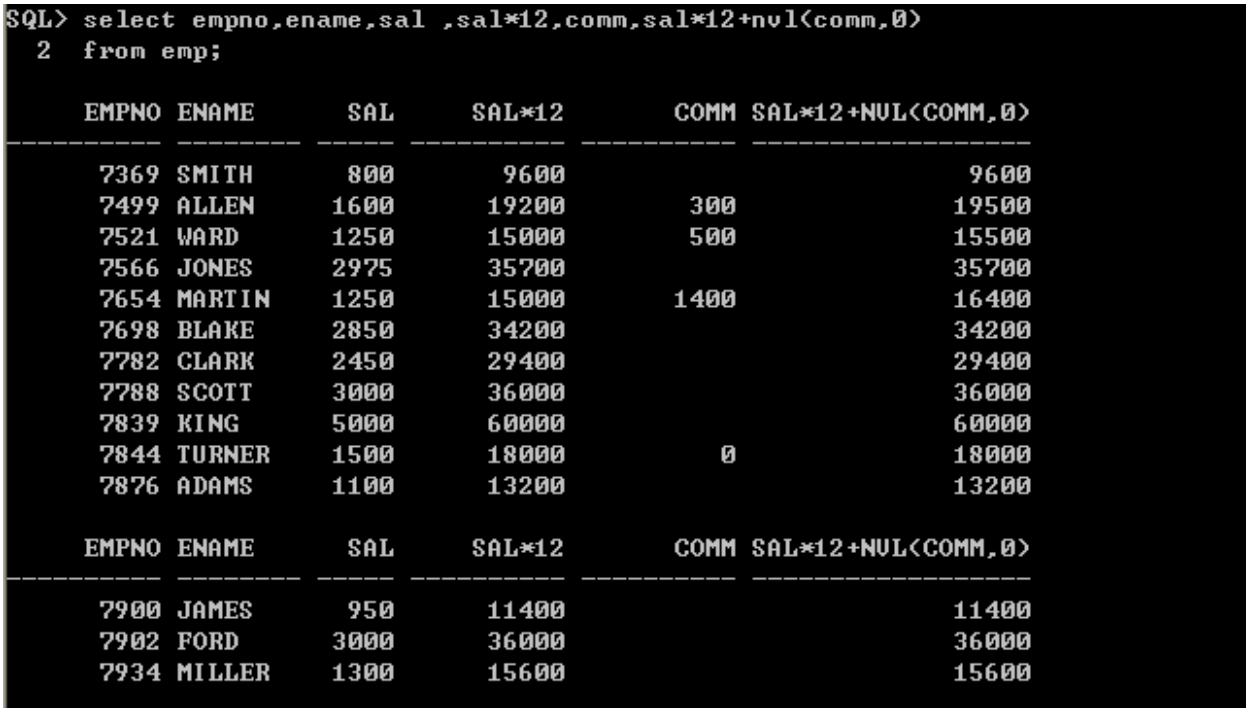
> 请问as ”员工号”，”姓名”，月薪三个别名有什么区别？
> as ”员工号”和”姓名”没有区别，”姓名”和月薪有区别，月薪中不能加入关键字、空格等特殊字符等

- distinct去掉重复的记录：select distinct deptno from emp;
- distinct作用于后面的所有列：select distinct deptno,job from emp;
- 连接符||：select 'Hello '||'World' from dual;
- dual表，伪表：仅仅满足语法的要求

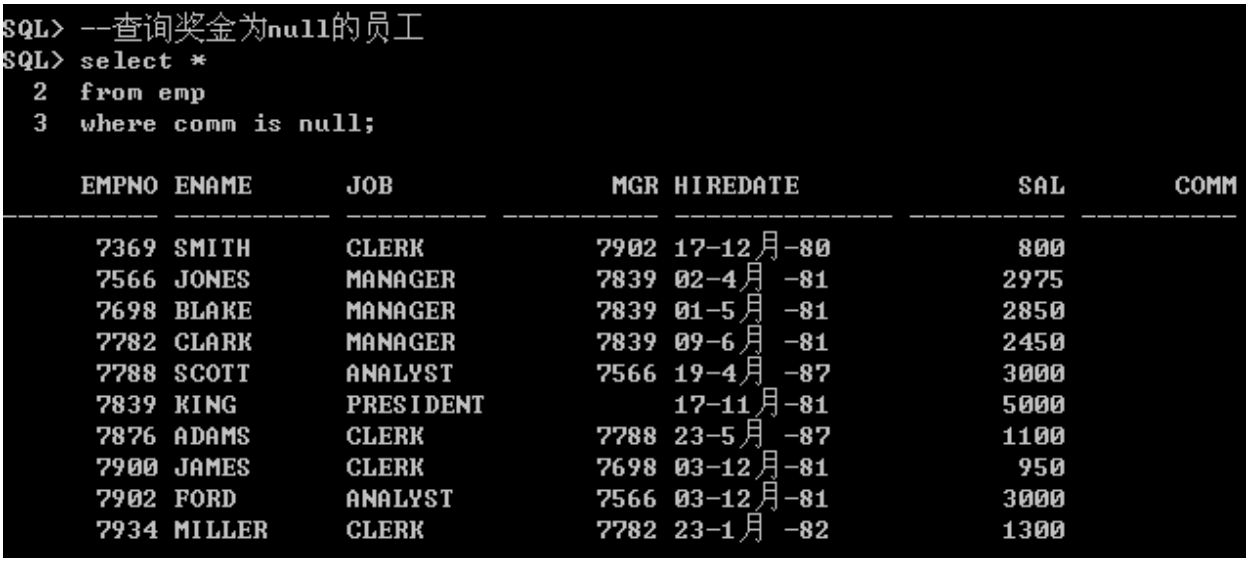
null值

- 包含**null**的表达式都为**null**
- **null**永远**!=null**
- 查询员工信息：员工号 姓名 月薪 年薪 奖金 年收入
  -

```
SQL> --查询员工信息：员工号 姓名 月薪 年薪 奖金 年收入
SQL> select empno,ename,sal ,sal*12,comm,sal*12+comm
  2  from emp;

     EMPNO ENAME      SAL     SAL*12       COMM SAL*12+COMM
---------- -------- ----- ---------- ---------- -----------
      7369 SMITH      800       9600
      7499 ALLEN     1600      19200        300       19500
      7521 WARD      1250      15000        500       15500
      7566 JONES     2975      35700
      7654 MARTIN    1250      15000       1400       16400
      7698 BLAKE     2850      34200
      7782 CLARK     2450      29400
      7788 SCOTT     3000      36000
      7839 KING      5000      60000
      7844 TURNER    1500      18000          0       18000
      7876 ADAMS     1100      13200

     EMPNO ENAME      SAL     SAL*12       COMM SAL*12+COMM
---------- -------- ----- ---------- ---------- -----------
      7900 JAMES      950      11400
      7902 FORD      3000      36000
      7934 MILLER    1300      15600
```

- 为什么没有奖金的没有年收入呢？奖金为null，则年收入为null，需要使用Oracle的滤空函数nvl
  - 

```
SQL> select empno,ename,sal ,sal*12,comm,sal*12+nvl(comm,0)
  2  from emp;

     EMPNO ENAME      SAL     SAL*12       COMM SAL*12+NVL(COMM,0)
---------- -------- ----- ---------- ---------- ------------------
      7369 SMITH      800       9600                          9600
      7499 ALLEN     1600      19200        300              19500
      7521 WARD      1250      15000        500              15500
      7566 JONES     2975      35700                         35700
      7654 MARTIN    1250      15000       1400              16400
      7698 BLAKE     2850      34200                         34200
      7782 CLARK     2450      29400                         29400
      7788 SCOTT     3000      36000                         36000
      7839 KING      5000      60000                         60000
      7844 TURNER    1500      18000          0              18000
      7876 ADAMS     1100      13200                         13200

     EMPNO ENAME      SAL     SAL*12       COMM SAL*12+NVL(COMM,0)
---------- -------- ----- ---------- ---------- ------------------
      7900 JAMES      950      11400                         11400
      7902 FORD      3000      36000                         36000
      7934 MILLER    1300      15600                         15600
```

- 查询奖金为null的员工，不能使用comm=null，需要comm is null
  - 

```
SQL> --查询奖金为null的员工
SQL> select *
  2  from emp
  3  where comm is null;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- -------------- ---------- ----------
      7369 SMITH      CLERK           7902 17-12月-80            800
      7566 JONES      MANAGER         7839 02-4月 -81           2975
      7698 BLAKE      MANAGER         7839 01-5月 -81           2850
      7782 CLARK      MANAGER         7839 09-6月 -81           2450
      7788 SCOTT      ANALYST         7566 19-4月 -87           3000
      7839 KING       PRESIDENT            17-11月-81           5000
      7876 ADAMS      CLERK           7788 23-5月 -87           1100
      7900 JAMES      CLERK           7698 03-12月-81            950
      7902 FORD       ANALYST         7566 03-12月-81           3000
      7934 MILLER     CLERK           7782 23-1月 -82           1300
```

过滤和排序

- 修改日期格式
  - select * from v$nls_parameters;
  - alter session set NLS*DATE*FORMAT='yyyy-mm-dd';
- between ... and 在。。。之间
  - select * from emp where sal between 1000 and 2000;
  - 1. 含有边界 2. 小值在前 大值在后
- in 在集合中

    - select * from emp where deptno in (10,20);
    - select * from emp where deptno not in (10,20)
    - **null** ，如果集合中，含有**null**，不能使用**not in**；但是可以使用**in**
- like 模糊查询
    - 查询名字以S打头的员工：select * from emp where ename like 'S%';
    - 查询名字是4个字的员工：select * from emp where ename like '____'
    - 查询名字中含有下划线的员工：select * from emp where ename like '%_%' escape '\'
- 查询员工信息，按照月薪排序：select * from emp order by sal;
- order by 后面 + 列，表达式，别名，序号
    - select empno,ename,sal,sal*12 from emp order by sal*12 desc;
    - select empno,ename,sal,sal*12 年薪 from emp order by 年薪 desc
    - select empno,ename,sal,sal*12 年薪 from emp order by 4 desc
- 多个列的排序:select * from emp order by deptno,sal;
    - order by 作用于后面所有的列，先按照第一个列排序，如果相同，再按照第二列排序；以此类推
    - select * from emp order by deptno,sal desc
- desc 只作用于离他最近的一列
    - select * from emp order by deptno desc,sal desc
- 含有**null**值的排序，使**null**值的列置后：select * from emp order by comm desc nulls last
    - oracle中，null最大

多表查询

- 等值连接
- 查询员工信息 ： 员工号 姓名 月薪 部门名称
    - select e.empno,e.ename,e.sal,d.dname from emp e,dept d where e.deptno=d.deptno;
- 不等值连接
- 查询员工信息 ： 员工号 姓名 月薪 工资级别
    - select e.empno,e.ename,e.sal,s.grade from emp e,salgrade s where e.sal between s.losal and s.hisal;
- 外连接
- 按部门统计员工人数： 部门号 部门名称 人数
    - select d.deptno 部门号,d.dname 部门名称,count(e.empno) 人数 from emp e,dept d where e.deptno=d.deptno group by d.deptno,d.dname;

    - select * from dept;
    - select * from emp where deptno=40;
    - 希望：对于某些不成立的记录，任然希望包含在最后的结果中
    - 左外连接：当where e.deptno=d.deptno不成立的时候，等号左边的表任然被包含
        - 写法： where e.deptno=d.deptno(+)
    - 右外连接：当where e.deptno=d.deptno不成立的时候，等号右边的表任然被包含
        - 写法： where e.deptno(+)=d.deptno
    - select d.deptno 部门号,d.dname 部门名称,count(e.empno) 人数 from emp e,dept d where e.deptno(+)=d.deptno group by d.deptno,d.dname;
- 自连接
- 查询员工信息： 员工的姓名 老板的姓名
- 自连接：通过表的别名，将同一张表视为多张表
    - select e.ename 员工的姓名,b.ename 老板姓名 from emp e,emp b where e.mgr=b.empno;
- 自连接不适合操作大表
- 层次查询
    - select level,empno,ename,mgr from emp connect by prior empno=mgr start with mgr is null order by 1;

单行函数

- 字符函数
    - select lower('Hello World') 转小写,upper('Hello World') 转大写, initcap('hello world') 首字母大写 from dual;
    - substr(a,b) 从a中，第b位开始取
        - select substr('Hello World',3) 子串 from dual;
    - substr(a,b,c) 从a中，第b位开始取,取c位
        - select substr('Hello World',3,4) 子串 from dual;
    - length 字符数 lengthb字节数
        - select length('Hello World') 字符,lengthb('Hello World') 字节 from dual;
        - select length('北京') 字符,lengthb('北京') 字节 from dual;
    - instr(a,b) 在a中，查找b
        - select instr('Hello World','ll') 位置 from dual;
    - lpad 左填充 rpad右填充
        - select lpad('abcd',10,'*') 左,*rpad('abcd',10,'*')* 右 from dual;

    - trim 去掉前后指定的字符
        - select trim('H' from 'Hello WorldH') from dual;
    - replace
        - select replace('Hello World','l','*') from dual;
- 数学函数
    - 四舍五入
        - select round(45.926,2) 一,round(45.926,1) 二,round(45.926,0) 三, round(45.926,-1) 四,round(45.926,-2) 五 from dual;
    - trunc
        - select trunc(45.926,2) 一,trunc(45.926,1) 二,trunc(45.926,0) 三, trunc(45.926,-1) 四,trunc(45.926,-2) 五 from dual
- 时间函数
    - 当前时间
        - select sysdate from dual;
    - 格式化显式一个时间
        - select to_char(sysdate,'yyyy-mm-dd hh24:mi:ss') from dual;
    - 昨天 今天 明天
        - select (sysdate-1) 昨天,sysdate 今天,(sysdate+1) 明天 from dual;
    - 计算员工的工龄：天 星期 月 年
        - select ename,hiredate,(sysdate-hiredate) 天,(sysdate-hiredate)/7 星期, (sysdate-hiredate)/30 月,(sysdate-hiredate)/365 年 from emp;
    - months_between
        - select ename,hiredate,(sysdate-hiredate)/30 一,months_between(sysdate,hiredate) 二 from emp;
    - add_months
        - select add_months(sysdate,73) from dual;
    - last_day
        - select last_day(sysdate) from dual;
    - 下一个星期五
        - select next_day(sysdate,'星期五') from dual;
        - select next_day(sysdate,'星期六') from dual;
        - next_day的应用：每个星期一自动备份数据
    - select round(sysdate,'month'),round(sysdate,'year') from dual;
    - 2014-12-26 15:19:12 今天是星期五

        - select to_char(sysdate,'yyyy-mm-dd hh24:mi:ss"今天是"day') from dual;
- 转换函数
    - 查询员工薪水：两位小数，千位符，货币代码
        - select to_char(sal,'L9,999.99') from emp;
    - nvl2(a,b,c) 当a=null时候，返回c；否则返回b
        - select sal*12+nvl2(comm,comm,0) from emp;
    - nullif(a,b) 当a=b时候，返回null，否则返回a
        - select nullif('abc','abc') 值 from dual;
        - select nullif('abc','abcd') 值 from dual;
    - coalesce 从左到右 找到第一个不为null的值
        - select comm,sal,coalesce(comm,sal) "第一个不为null的值" from emp;
    - 涨工资，总裁1000 经理800 其他400
        - case
            - select ename,job,sal 涨前,case job when 'PRESIDENT' then sal+1000 when 'MANAGER' then sal+800 else sal+400 end 涨后 from emp;
        - decode
            - select ename,job,sal 涨前,decode(job,'PRESIDENT',sal+1000,'MANAGER',sal+800,sal+400) 涨后 from emp;

多行函数

- 工资总额
    - select sum(sal) from emp;
- 人数
    - select count(*) from emp;
- 平均工资
    - select sum(sal)/count(*) 一,avg(sal) 二 from emp;
- 平均奖金
    - select sum(comm)/count(*) 一,sum(comm)/count(comm) 二,avg(comm) 三 from emp;
    - select count(*),count(comm) from emp;
    - select * from emp;
    - **null** 组函数自动滤空
        - select count(*),count(nvl(comm,0)) from emp;

    - null 组函数自动滤空;可以嵌套滤空函数来屏蔽他的滤空功能
- 求部门的平均工资
    - select deptno,avg(sal) from emp group by deptno;
    - 多个列的分组
        - select deptno,job,sum(sal) from emp group by deptno,job order by 1;
- 查询平均工资大于2000的部门
    - select deptno,avg(sal) from emp group by deptno having avg(sal) > 2000;
- where后面不能使用组函数
- 查询10号部门的员工
    - select deptno,avg(sal) from emp group by deptno having deptno=10;
    - select deptno,avg(sal) from emp where deptno=10 group by deptno
- group by的增强
    - select deptno,job,sum(sal) from emp group by rollup(deptno,job);
    - break on deptno skip 2
    - select deptno,job,sum(sal) from emp group by rollup(deptno,job);
    - break on null
    - select deptno,job,sum(sal) from emp group by rollup(deptno,job);

子查询

- 子查询所要解决的问题：不能一步求解
- 查询工资比SCOTT高的员工信息
    - select * from emp where sal > (select sal from emp where ename='SCOTT')

```
SQL> 注意的问题
    1. 括号
    2. 合理的书写风格
    3. 可以在主查询的where select having from后面放置子查询
    4. 不可以在group by后面放置子查询
    5. 强调from后面的子查询
    6. 主查询和子查询可以不是同一张表；只要子查询返回的结果主查询可以使用即可
    7. 一般不在子查询中排序；但在Top-N分析问题中，必须对子查询排序
    8. 一般先执行子查询，再执行主查询；但相关子查询例外
    9. 单行子查询只能使用单行操作符；多行子查询只能使用多行操作符
    10. 子查询中的null
```

- 可以在主查询的where select having from后面放置子查询

    - select empno,ename,sal,(select job from emp where empno=7839) 第四列 from emp;
    - 强调from后面的子查询
    - 查询员工信息：员工号 姓名 月薪
        - select * from (select empno,ename,sal from emp);
    - 查询员工信息：员工号 姓名 月薪 年薪
        - select * from (select empno,ename,sal,sal*12 annlsal from emp);
- 主查询和子查询可以不是同一张表；只要子查询返回的结果主查询可以使用即可
    - 查询部门名称是SALES的员工信息
        - select * from emp where deptno = (select deptno from dept where dname='SALES');
        - select e.* from emp e,dept d where e.deptno=d.deptno and d.dname='SALES';
- in 在集合中
    - select * from emp where deptno in (select deptno from dept where dname='SALES' or dname='ACCOUNTING');
    - select e.* from emp e,dept d where e.deptno=d.deptno and (d.dname='SALES' or d.dname='ACCOUNTING');
- any 和集合中任意一个值比较
    - 查询工资比30号部门任意一个员工高的员工信息
        - select * from emp where sal > any (select sal from emp where deptno=30);
        - select * from emp where sal > (select min(sal) from emp where deptno=30);
- all: 和集合中的所有值比较
    - 查询工资比30号部门所有员工高的员工信
        - select * from emp where sal > all (select sal from emp where deptno=30);
        - select * from emp where sal > (select max(sal) from emp where deptno=30)
- 多行子查询中的null
- not in (10,20,null)
- 查询不是老板的员工信息
    - select * from emp;
    - select * from emp where empno not in (select mgr from emp);
    - select * from emp where empno not in (select mgr from emp where mgr is not null);
- 查询是老板的员工信息
    - select * from emp where empno in (select mgr from emp);
- oracle分页
    - select * from (select rownum r,e1.* from (select * from emp order by sal) e1 where rownum

<=8) where r >=5;

集合运算

- 查询部门号是10和20的员工
  - select * from emp where deptno in (10,20);
  - select * from emp where deptno=10 or deptno=20;
  - select * from emp where deptno=10 union select * from emp where deptno=20;

```
注意的问题：
    1.  参与运算的各个集合必须列数相同  且类型一致
    2.  采用第一个集合表头作为最后表头
    3.   order by 永远在最后
    4.  括号
*   select deptno,job,sum(sal) from emp group by deptno,job union select deptno,to_char(null),su
```

伪列

- rownum 行号 伪列
  - select rownum,empno,ename,sal from emp;
  - select rownum,empno,ename,sal from emp where rownum<=3 order by sal desc;

```
关于行号
1. rownum 永远按照默认的顺序生成
2. rownum只能使用< <=; 不能使用> >=
*   select rownum,empno,ename,sal from emp order by sal desc;
*   select rownum,empno,ename,sal from (select * from emp order by sal desc) where rownum <= 3;
*   select rownum,empno,ename,sal from emp where rownum<=8;
*   select * from   (select rownum r,e1.* from (select * from emp order by sal) e1 where rownum
```

- 临时表：create global temporary table ******
- 特点：当事务或者会话结束的时候，表中的数据自动删除
- 第二题
  - select e.empno,e.ename,e.sal,d.avgsal from emp e,(select deptno,avg(sal) avgsal from emp group by deptno) d where e.deptno=d.deptno and e.sal>d.avgsal;
- 相关子查询：将主查询中的某些值作为参数传递给子查询
  - select empno,ename,sal,(select avg(sal) from emp where deptno=e.deptno) avgsal from emp e where sal > (select avg(sal) from emp where deptno=e.deptno);

- 行转列
  - wm_concat(varchar2) ---> 组函数
  - select deptno,wm_concat(ename) nameslist from emp group by deptno;

处理数据

- 统计每年入职的员工数

```
select count(*) Total,
    sum(decode(to_char(hiredate,'yyyy'),'1980',1,0)) "1980",
    sum(decode(to_char(hiredate,'yyyy'),'1981',1,0)) "1981",
    sum(decode(to_char(hiredate,'yyyy'),'1982',1,0)) "1982",
    sum(decode(to_char(hiredate,'yyyy'),'1987',1,0)) "1987"
from emp;
```

- 插入 insert

```
SQL> insert into emp(empno,ename,sal,deptno) values(1001,'Tom',3000,10);

已创建 1 行。

SQL> --PreparedStatement pst = "insert into emp(empno,ename,sal,deptno) values(?,?,?,?)";
SQL> --地址符 &
SQL> insert into emp(empno,ename,sal,deptno) values(&empno,&ename,&sal,&deptno);
输入 empno 的值:  1002
输入 ename 的值:  'Mary'
输入 sal 的值:  3000
输入 deptno 的值:  20
原值    1: insert into emp(empno,ename,sal,deptno) values(&empno,&ename,&sal,&deptno)
新值    1: insert into emp(empno,ename,sal,deptno) values(1002,'Mary',3000,20)

已创建 1 行。
--------------------------------------------------------------------------------

SQL> select empno,ename,&t
  2  from emp;
输入 t 的值:  sal
原值    1: select empno,ename,&t
新值    1: select empno,ename,sal

SQL> /
输入 t 的值:  job
原值    1: select empno,ename,&t
新值    1: select empno,ename,job
--------------------------------------------------------------------------------
```

```
SQL> select * from &t;
输入 t 的值:  dept
原值    1: select * from &t
新值    1: select * from dept
```

- 批处理
  - create table emp10 as select * from emp where 1=2;
  - 一次性将emp中所有10号部门的员工插入到emp10中
    - insert into emp10 select * from emp where deptno=10;
- 海量插入数据
  - 1. 数据泵
  - 2. SQL*Loader
  - 3. 外部表
- delete 和truncate区别
  - 1. delete 逐条删除，truncate先删除表，再重建
  - 2.****delete是DML(可以回滚) truncate是DDL（不可以回滚）
  - 3. delete 不会释放空间；truncate会
  - 4. delete会产生碎片 truncate不会
  - 5. delete可以闪回(flashback) truncate不可以
  - set feedback off
  - @d:\temp\testdelete.sql
  - delete 并不是真正删除；把数据换个地方(undo 表空间)存
- Oracle中的事务
- 1. 起始标志：事务中第一题DML语句
- a. 结束标志：

  b. 提交：
    - 显式 commit
    - 隐式 正常退出exit，DDL，DCL

  c. 回滚
    - 显式 rollback
    - 隐式 非正常退出，掉电，宕机
- set feedback on
- 定义保存点：savepoint a;

- 回滚到保存点：rollback to savepoint a;
- 设置事务为只读：set transaction read only;
- Oracle的回收站：show recyclebin;
    - 回收站已清空：purge recyclebin;

其它数据库对象

- 视图
    - create [or replace] view empinfoview as select e.empno,e.ename,e.sal,e.sal*12 annlsa,d.dname from emp e,dept d where e.deptno=d.deptno;
- 索引 index
    - create index myindex on emp(deptno);
    - 通过得到SQL的执行计划，可以确认是否查询了索引
- 序列 sequence
    - create sequence myseq;
        - myseq.currval
        - myseq.nextval
- 同义词 （别名）
    - 为hr.employees起别名 ---> 同义词
        - create synonym hremp for hr.employees;

PLSQL

- 结构

```
set serveroutput on

declare
   -- 说明部分
 begin
   --程序
   dbms_output.put_line('Hello World');
 end;
 /
```

- if语句

```
--判断用户从键盘上输入的数字
```

```
set serveroutput on

--接收键盘输入
--num 是一个地址值，在该地址上保存了输入的值
accept num prompt '请输入一个数字';

declare
  --定义变量保存输入的数字
  pnum number := &num;
begin
  if pnum = 0 then dbms_output.put_line('您输入的是0');
    elsif pnum = 1 then dbms_output.put_line('您输入的是1');
    elsif pnum = 2 then dbms_output.put_line('您输入的是2');
    else dbms_output.put_line('其他数字');
  end if;

end;
/
```

- 光标

```
--查询并打印员工的姓名和薪水
/*
1. 光标属性
   %isopen    %rowcount(影响的行数)
   %found     %notfound

*/
set serveroutput on
declare
  --定义光标
  cursor cemp is select ename,sal from emp;
  pename emp.ename%type;
  psal   emp.sal%type;
begin
  open cemp;
  loop
    --取一条记录
    fetch cemp into pename,psal;
    --退出条件
    --exit when 没有取到记录;
    exit when cemp%notfound;

    dbms_output.put_line(pename||'的薪水是'||psal);

  end loop;
  close cemp;
end;
```

```
/
```

- 带参数的光标

```
--查询某个部门的员工姓名
set serveroutput on

declare
  --定义光标保存某个部门的员工姓名
  cursor cemp(dno number) is select ename from emp where deptno=dno;
  pename emp.ename%type;
begin
  open cemp(20);
  loop
    fetch cemp into pename;
    exit when cemp%notfound;

    dbms_output.put_line(pename);

  end loop;
  close cemp;
end;
/
```

- 记录型变量

```
--查询7839的姓名和薪水
set serveroutput on

declare
  --定义记录型变量：代表一行
  emp_rec emp%rowtype;
begin
  select * into emp_rec from emp where empno=7839;

  dbms_output.put_line(emp_rec.ename||'的薪水是'||emp_rec.sal);
end;
/
```

- 实例一

```
/*
SQL语句
select to_char(hiredate,'yyyy') from emp;
--> 光标  --> 循环  --> 退出条件：notfound

变量：1. 初始值  2. 如何得到
```

```
count80 number := 0;
count81 number := 0;
count82 number := 0;
count87 number := 0;
*/
set serveroutput on
declare
  cursor cemp is select to_char(hiredate,'yyyy') from emp;
  phiredate varchar2(4);

  count80 number := 0;
  count81 number := 0;
  count82 number := 0;
  count87 number := 0;
begin
  open cemp;
  loop
    --取一个员工
    fetch cemp into phiredate;
    exit when cemp%notfound;

    --判断
    if phiredate = '1980' then count80:=count80+1;
      elsif phiredate = '1981' then count81:=count81+1;
      elsif phiredate = '1982' then count82:=count82+1;
      else count87:=count87+1;
    end if;

  end loop;
  close cemp;

  --输出
  dbms_output.put_line('Total:'||(count80+count81+count82+count87));
  dbms_output.put_line('1980:'||count80);
  dbms_output.put_line('1981:'||count81);
  dbms_output.put_line('1982:'||count82);
  dbms_output.put_line('1987:'||count87);
end;
/
```

- 实例二

```
/*
SQL语句
select empno,sal from emp order by sal;
-->  光标 --> 循环  --> 退出条件：1. 总额 > 5w  2. notfound

变量：1. 初始值  2. 如何得到
```

```
涨工资的人数：countEmp number := 0;
涨后的工资总额：  salTotal number;
1. select sum(sal) into salTotal from emp;
2. 涨后 = 涨前 + sal * 0.1

练习：
人数:8  工资总额：50205.325
*/
set serveroutput on
declare
  cursor cemp is select empno,sal from emp order by sal;
  pempno emp.empno%type;
  psal   emp.sal%type;
  --涨工资的人数：
  countEmp number := 0;
  --涨后的工资总额：
  salTotal number;
begin
  --得到工资总额的初始值
  select sum(sal) into salTotal from emp;

  open cemp;
  loop
    --1. 总额 > 5w
    exit when salTotal > 50000;
    --取一个员工
    fetch cemp into pempno,psal;
    --2. notfound
    exit when cemp%notfound;

    --涨工资
    update emp set sal=sal*1.1 where empno=pempno;
    --人数
    countEmp := countEmp + 1;
    --2. 涨后 = 涨前 + sal * 0.1
    salTotal := salTotal + psal * 0.1;

  end loop;
  close cemp;

  commit;
  dbms_output.put_line('人数:'||countEmp||'  工资总额：'||salTotal);
end;
/
```

- 实例三 # /* SQL语句 部门：select deptno from dept; --> 光标 部门中员工的薪水: select sal from emp where deptno=?? --> 带参数的光标

变量：1. 初始值 2. 如何得到 每个段的人数 count1 number; count2 number; count3 number;

部门的工资总额: salTotal number;

a. select sum(sal) into salTotal from emp where deptno=??

b. 累加 */ set serveroutput on declare --部门 cursor cdept is select deptno from dept; pdeptno dept.deptno%type;

--部门中员工的薪水 cursor cemp(dno number) is select sal from emp where deptno=dno; psal emp.sal%type; --每个段的人数 count1 number; count2 number; count3 number; --部门的工资总额: salTotal number; begin open cdept; loop --取一个部门 fetch cdept into pdeptno; exit when cdept%notfound;

```
--初始化
count1:=0;count2:=0;count3:=0;
--得到部门的工资总额
select sum(sal) into salTotal from emp where deptno=pdeptno;

--取部门中员工的薪水
open cemp(pdeptno);
loop
  --取一个员工的薪水
  fetch cemp into psal;
  exit when cemp%notfound;

  --判断薪水的范围
  if psal < 3000 then count1:=count1+1;
    elsif psal>=3000 and psal<6000 then count2:=count2+1;
    else count3:=count3+1;
  end if;

end loop;
close cemp;

--保存结果
insert into msg values(pdeptno,count1,count2,count3,nvl(salTotal,0));
```

end loop; close cdept;

commit; dbms*output.put*line('完成');

end; /

- 系统例外

```
--被0除
set serveroutput on
```

```
declare
  pnum number;
begin
  pnum := 1/0;

exception
  when zero_divide then dbms_output.put_line('1:0不能做被除数');
                        dbms_output.put_line('2:0不能做被除数');
  when value_error then dbms_output.put_line('算术或者转换错误');
  WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('其他例外');
end;
/
```

- 循环

```
--打印1~10
set serveroutput on

declare
  pnum number := 1;
begin
  loop
    --退出条件
    exit when pnum > 10;

    dbms_output.put_line(pnum);
    --加一
    pnum := pnum + 1;
  end loop;
end;
/
```

- 引用型变量

```
--查询7839的姓名和薪水
set serveroutput on

declare
  --定义变量保存姓名和薪水
  --pename varchar2(20);
  --psal   number;
  pename emp.ename%type;
  psal   emp.sal%type;
begin
  --得到姓名和薪水
  select ename,sal into pename,psal  from emp where empno=7839;
```

```
  dbms_output.put_line(pename||'的薪水是'||psal);
end;
/
```

- 涨工资

```
--涨工资，总裁1000 经理800 其他400
set serveroutput on

declare
  --alter table "SCOTT"."EMP" rename column "JOB" to empjob
  cursor cemp is select empno,empjob from emp;
  pempno emp.empno%type;
  pjob   emp.empjob%type;
begin
  rollback;
  open cemp;
  loop
    --取一条记录
    fetch cemp into pempno,pjob;
    exit when cemp%notfound;

    --判断职位
    if pjob = 'PRESIDENT' then update emp set sal=sal+1000 where empno=pempno;
      elsif pjob = 'MANAGER' then update emp set sal=sal+800 where empno=pempno;
      else update emp set sal=sal+400 where empno=pempno;
    end if;
  end loop;
  close cemp;

  --why?  --> ACID
  commit;
  dbms_output.put_line('完成');
end;
/
```

- 自定义例外

```
--查询50号部门的员工姓名
set serveroutput on

declare
  cursor cemp is select ename from emp where deptno=50;
  pename emp.ename%type;
  --自定义例外
  no_emp_found exception;
begin
```

```
  open cemp;
  --取第一条记录
  fetch cemp into pename;

  if cemp%notfound then
    --抛出例外
    raise no_emp_found;
  end if;

  --进程pmon (process monitor)
  close cemp;

exception
  when no_emp_found then dbms_output.put_line('没有找到员工');
  WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('其他例外');
end;
/
```

存储函数，存储过程，触发器

```
--查询某个员工的姓名  月薪   职位

/*
思考:
1. 查询某个员工的所有信息 ---> out参数太多
2. 查询某个部门中的所有员工信息 ---> 返回集合
*/
create or replace procedure queryEmpInfo(eno in number,
                                         pename out varchar2,
                                         psal   out number,
                                         pjob   out varchar2)
as
begin
  select ename,sal,empjob into pename,psal,pjob from emp where empno=eno;
end;
/
```

- 触发器例子一

```
/*
数据确认
涨后的薪水不能少于涨前的薪水
*/
create or replace trigger checksalary
before update
on emp
```

```
for each row
begin
  --if 涨后的薪水  < 涨前的薪水  then
  if :new.sal  < :old.sal then
    raise_application_error(-20002,'涨后的薪水不能少于涨前的薪水.涨前:'||:old.sal||'  涨后:'||:new.
  end if;
end;
/
```

- 触发器例子二

```
/*
复杂的安全性检查
禁止在非工作时间插入新员工

1. 周末：to_char(sysdate,'day') in ('星期六','星期日')
2. 上班前 下班后：to_number(to_char(sysdate,'hh24')) not between 9 and 18
*/
create or replace trigger securityemp
before insert
on emp
begin
  if to_char(sysdate,'day') in ('星期六','星期日') or
     to_number(to_char(sysdate,'hh24')) not between 9 and 18 then
     --禁止插入
    raise_application_error(-20001,'禁止在非工作时间插入新员工');
  end if;
end;
/
```

- 存储函数

```
--查询某个员工的年收入
create or replace function queryEmpIncome(eno in number)
return number
as
   --月薪和奖金
   psal emp.sal%type;
   pcomm emp.comm%type;
begin
   select sal,comm into psal,pcomm from emp where empno=eno;

   --返回年收入
   return psal*12+nvl(pcomm,0);
end;
/
```

- 存储过程

```
--给指定员工涨100，并且打印涨前和涨后的薪水
create or replace procedure raiseSalary(eno in number)
as
  --说明部分
  psal emp.sal%type;
begin
  --得到涨前的薪水
  select sal into psal from emp where empno=eno;

  update emp set sal=sal+100 where empno=eno;

  --要不要commit?

  dbms_output.put_line('涨前:'||psal||'   涨后:'||(psal+100));
end;
/
```

- 触发器

```
--每当成功插入新员工后，自动打印“成功插入新员工”

create trigger saynewemp
after insert
on emp
declare
begin
   dbms_output.put_line('成功插入新员工');
end;
/
```

- 第一个简单的存储过程

```
--打印Hello World
/*
调用存储过程
1. exec sayhelloworld();
2. begin
     sayhelloworld();
     sayhelloworld();
   end;
   /
*/
create or replace procedure sayhelloworld
as
   --说明部分
```

```
begin
   dbms_output.put_line('Hello World');
end;
/
```

- 包（package）和包体（package body）

```
2. 查询某个部门中的所有员工信息 ---> 返回集合

包头
CREATE OR REPLACE PACKAGE MYPACKAGE AS

  type empcursor is ref cursor;
  procedure queryEmpList(dno in number,empList out empcursor);

END MYPACKAGE;


包体
CREATE OR REPLACE PACKAGE BODY MYPACKAGE AS

  procedure queryEmpList(dno in number,empList out empcursor) AS
  BEGIN

    open empList for select * from emp where deptno=dno;

  END queryEmpList;

END MYPACKAGE;
```

附录

```
SCOTT用户下的表
SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE

SQL> desc dept;
 名称                      是否为空? 类型
 ----------------------- -------- --------------
 DEPTNO                  NOT NULL NUMBER(2)
```

```
 DNAME                                     VARCHAR2(14)
 LOC                                       VARCHAR2(13)

SQL> desc emp;
 名称                      是否为空? 类型
 ----------------------- -------- --------------
 EMPNO                     NOT NULL NUMBER(4)
 ENAME                              VARCHAR2(10)
 JOB                                VARCHAR2(9)
 MGR                                NUMBER(4)
 HIREDATE                           DATE
 SAL                                NUMBER(7,2)
 COMM                               NUMBER(7,2)
 DEPTNO                             NUMBER(2)

SQL> desc bonus;
 名称                      是否为空? 类型
 ----------------------- -------- -------------
 ENAME                              VARCHAR2(10)
 JOB                                VARCHAR2(9)
 SAL                                NUMBER
 COMM                               NUMBER

SQL> desc SALGRADE;
 名称                      是否为空? 类型
 ----------------------- -------- -------------
 GRADE                              NUMBER
 LOSAL                              NUMBER
 HISAL                              NUMBER

SQL> select * from DEPT;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

SQL> select * from EMP;

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-12月-80            800                    20
      7499 ALLEN      SALESMAN        7698 20-2月 -81           1600        300         30
      7521 WARD       SALESMAN        7698 22-2月 -81           1250        500         30
      7566 JONES      MANAGER         7839 02-4月 -81           2975                    20
      7654 MARTIN     SALESMAN        7698 28-9月 -81           1250       1400         30
      7698 BLAKE      MANAGER         7839 01-5月 -81           2850                    30
```

```
      7782 CLARK      MANAGER         7839 09-6月 -81           2450                    10
      7788 SCOTT      ANALYST         7566 19-4月 -87           3000                    20
      7839 KING       PRESIDENT            17-11月-81           5000                    10
      7844 TURNER     SALESMAN        7698 08-9月 -81           1500          0         30
      7876 ADAMS      CLERK           7788 23-5月 -87           1100                    20

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------------- ---------- ---------- ----------
      7900 JAMES      CLERK           7698 03-12月-81            950                    30
      7902 FORD       ANALYST         7566 03-12月-81           3000                    20
      7934 MILLER     CLERK           7782 23-1月 -82           1300                    10

已选择14行。

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
DEPT                           TABLE
EMP                            TABLE
BONUS                          TABLE
SALGRADE                       TABLE

SQL> select * from BONUS;

未选定行

SQL> select * from SALGRADE;

     GRADE      LOSAL      HISAL
---------- ---------- ----------
         1        700       1200
         2       1201       1400
         3       1401       2000
         4       2001       3000
         5       3001       9999

SQL> SPOOL OFF
```