# **[oracle 用户管理一](http://www.cnblogs.com/linjiqin/archive/2012/01/31/2332616.html)**

一、创建用户  
概述：在oracle中要创建一个新的用户使用create user语句，一般是具有dba(数据库管理员)的权限才能使用。  
create user 用户名 identified by 密码;   
注意：oracle有个毛病，密码必须以字母开头，如果以数字开头，它不会创建用户  
eg、create user xiaoming identified by oracle;

二、给用户修改密码  
概述：如果给自己修改密码可以直接使用  
SQL> password 用户名或passw  
如果给别人修改密码则需要具有dba的权限，或是拥有alter user的系统权限  
SQL> alter user 用户名 identified by 新密码  
                    
三、删除用户  
概述：一般以dba的身份去删除某个用户，如果用其它用户去删除用户则需要具有drop user的权限。  
比如drop user 用户名 【cascade】  
注意：在删除用户时，如果要删除的用户，已经创建了表，那么就需要在删除的时候带一个参数cascade，即把该用户及表一同删除;

四、权限  
权限分为系统权限和对象权限。  
何为系统权限？  
用户对数据库的相关权限，connect、resource、dba等系统权限，如建库、建表、建索引、建存储过程、登陆数据库、修改密码等。  
何为对象权限？  
用户对其他用户的数据对象操作的权限，insert、delete、update、select、all等对象权限，数据对象有很多，比如表，索引，视图，触发器、存储过程、包等。  
执行SELECT \* FROM Dba\_Object\_Size;语句可得到oracle数据库对象。

五、角色  
角色分为预定义角色和自定义角色。

六、用户管理的综合案例  
概述：创建的新用户是没有任何权限的，甚至连登陆的数据库的权限都没有，需要为其指定相应的权限。给一个用户赋权限使用命令grant，回收权限使用命令revoke。  
为了讲清楚用户的管理，这里我给大家举一个案例。  
SQL> conn xiaoming/oracle  
ERROR:  
ORA-01045: user XIAOMING lacks CREATE SESSION privilege; logon denied  
警告: 您不再连接到 ORACLE。  
SQL> show user  
USER 为 ""  
SQL> conn system/oracle  
已连接。  
SQL> grant connect to xiaoming;  
授权成功。  
SQL> conn xiaoming/oracle  
已连接。  
SQL>  
注意：grant connect to xiaoming;在这里，准确的讲，connect不是权限，而是角色。  
                              
现在说下对象权限，现在要做这么件事情：  
\* 希望xiaoming用户可以去查询emp表  
\* 希望xiaoming用户可以去查询scott的emp表  
grant select on scott.emp to xiaoming  
\* 希望xiaoming用户可以去修改scott的emp表  
grant update on scott.emp to xiaoming  
\* 希望xiaoming 用户可以去修改/删除，查询，添加scott的emp表  
grant all on scott.emp to xiaoming  
\* scott希望收回xiaoming对emp表的查询权限  
revoke select on scott.emp from xiaoming

七、权限的传递  
//对权限的维护。  
\* 希望xiaoming用户可以去查询scott的emp表/还希望xiaoming可以把这个权限传递给别人。  
--如果是对象权限，就加入with grant option  
grant select on emp to xiaoming with grant option  
我的操作过程：  
SQL> conn scott/oracle;  
已连接。  
SQL> grant select on scott.emp to xiaoming with grant option;  
授权成功。  
SQL> conn system/oracle;  
已连接。  
SQL> create user xiaohong identified by oracle;  
用户已创建。  
SQL> grant connect to xiaohong;  
授权成功。  
SQL> conn xiaoming/oracle;  
已连接。  
SQL> grant select on scott.emp to xiaohong;  
授权成功。  
                               
--如果是系统权限。  
system给xiaoming权限时：grant connect to xiaoming with admin option  
问题：如果scott把xiaoming对emp表的查询权限回收，那么xiaohong会怎样？  
答案：被回收。  
下面是我的操作过程：  
SQL> conn scott/oracle;  
已连接。  
SQL> revoke select on emp from xiaoming;  
撤销成功。  
SQL> conn xiaohong/oracle;  
已连接。  
SQL> select \* from scott.emp;  
select \* from scott.emp  
\*  
第 1 行出现错误:  
ORA-00942: 表或视图不存在  
结果显示：小红受到诛连了。。

八、with admin option与with grant option区别  
1、with admin option用于系统权限授权，with grant option用于对象授权。

2、给一个用户授予系统权限带上with admin option时，此用户可把此系统权限授予其他用户或角色，但收回这个用户的系统权限时，这个用户已经授予其他用户或角色的此系统权限不会因传播无效，如授予A系统权限create session with admin option,然后A又把create session权限授予B,但管理员收回A的create session权限时，B依然拥有create session的权限，但管理员可以显式收回B create session的权限，即直接revoke create session from B.

而with grant option用于对象授权时，被授予的用户也可把此对象权限授予其他用户或角色，不同的是但管理员收回用with grant option授权的用户对象权限时，权限会因传播而失效，如grant select on table with grant option to A,A用户把此权限授予B，但管理员收回A的权限时，B的权限也会失效，但管理员不可以直接收回B的SELECT ON TABLE 权限。

## 一、使用profile管理用户口令

概述：profile是口令限制，资源限制的命令集合，当建立数据库时，oracle会自动建立名称为default的profile。当建立用户没有指定profile选项时，那么oracle就会将default分配给用户。

### 1.账户锁定

概述：指定该账户(用户)登陆时最多可以输入密码的次数，也可以指定用户锁定的时间(天)一般用dba的身份去执行该命令。

例子：指定scott这个用户最多只能尝试3次登陆，锁定时间为2天，让我们看看怎么实现。

创建profile文件

SQL> create profile lock\_account limit failed\_login\_attempts 3 password\_lock\_time 2;

SQL> alter user scott profile lock\_account;

### 2.给账户(用户)解锁

SQL> alter user scott account unlock;

### 3.终止口令

为了让用户定期修改密码可以使用终止口令的指令来完成，同样这个命令也需要dba的身份来操作。

例子：给前面创建的用户test创建一个profile文件，要求该用户每隔10天要修改自己的登陆密码，宽限期为2天。看看怎么做。

SQL> create profile myprofile limit password\_life\_time 10 password\_grace\_time 2;

SQL> alter user test profile myprofile;

## 二、口令历史

概述：如果希望用户在修改密码时，不能使用以前使用过的密码，可使用口令历史，这样oracle就会将口令修改的信息存放到数据字典中，这样当用户修改密码时，oracle就会对新旧密码进行比较，当发现新旧密码一样时，就提示用户重新输入密码。

例子：

1）建立profile

SQL>create profile password\_history limit password\_life\_time 10 password\_grace\_time 2

password\_reuse\_time 10 //password\_reuse\_time指定口令可重用时间即10天后就可以重用

2）分配给某个用户

SQL> alter user test profile password\_history;

## 三、删除profile

概述：当不需要某个profile文件时，可以删除该文件。

SQL> drop profile password\_history 【casade】

注意：文件删除后，用这个文件去约束的那些用户通通也都被释放了。。

加了casade，就会把级联的相关东西也给删除掉

# oracle 表的管理

## 一、表名和列名的命名规则

1)、必须以字母开头

2)、长度不能超过30个字符

3)、不能使用oracle的保留字

4)、只能使用如下字符 a-z，a-z，0-9，$,#等

## 二、数据类型

1)、字符类

char 长度固定，最多容纳2000个字符。

例子：char(10) ‘小韩’前四个字符放‘小韩’，后添6个空格补全，如‘小韩 ’

varchar2(20) 长度可变，最多容纳4000个字符。

例子：varchar2（10） ‘小韩’ oracle分配四个字符。这样可以节省空间。

clob(character large object) 字符型大对象，最多容纳4g

char 查询的速度极快浪费空间，适合查询比较频繁的数据字段。

varchar 节省空间

2)、数字型

number范围-10的38次方到10的38次方，可以表示整数，也可以表示小数

number(5,2)表示一位小数有5位有效数，2位小数；范围：-999.99 到999.99

number(5)表示一个5位整数；范围99999到-99999

3)、日期类型

date 包含年月日和时分秒 oracle默认格式1-1月-1999

timestamp 这是oracle9i对date数据类型的扩展。可以精确到毫秒。

4)、图片

blob 二进制数据，可以存放图片/声音4g；一般来讲，在真实项目中是不会把图片和声音真的往数据库里存放，一般存放图片、视频的路径，如果安全需要比较高的话，则放入数据库。

## 三、怎样创建表

### --创建表

### --学生表

create table student (

xh number(4), --学号

xm varchar2(20), --姓名

sex char(2), --性别

birthday date, --出生日期

sal number(7,2) --奖学金

);

### --班级表

create table class(

classid number(2),

cname varchar2(40)

);

### --修改表

--添加一个字段

sql>alter table student add (classid number(2));

--修改一个字段的长度

sql>alter table student modify (xm varchar2(30));

--修改字段的类型或是名字（不能有数据） 不建议做

sql>alter table student modify (xm char(30));

--删除一个字段 不建议做(删了之后，顺序就变了。加就没问题，应该是加在后面)

sql>alter table student drop column sal;

--修改表的名字 很少有这种需求

sql>rename student to stu;

### --删除表

sql>drop table student;

### --添加数据

--所有字段都插入数据

insert into student values ('a001', '张三', '男', '01-5 月-05', 10);

--oracle中默认的日期格式‘dd-mon-yy’ dd 天 mon 月份 yy 2位的年 ‘09-6 月-99’ 1999年6月9日

--修改日期的默认格式（临时修改，数据库重启后仍为默认；如要修改需要修改注册表）

alter session set nls\_date\_format ='yyyy-mm-dd';

--修改后，可以用我们熟悉的格式添加日期类型：

insert into student values ('a002', 'mike', '男', '1905-05-06', 10);

--插入部分字段

insert into student(xh, xm, sex) values ('a003', 'john', '女');

--插入空值

insert into student(xh, xm, sex, birthday) values ('a004', 'martin', '男', null);

--问题来了，如果你要查询student表里birthday为null的记录，怎么写sql呢？

--错误写法：select \* from student where birthday = null;

--正确写法：select \* from student where birthday is null;

--如果要查询birthday不为null,则应该这样写：

select \* from student where birthday is not null;

### --修改数据

--修改一个字段

update student set sex = '女' where xh = 'a001';

--修改多个字段

update student set sex = '男', birthday = '1984-04-01' where xh = 'a001';

--修改含有null值的数据

不要用 = null 而是用 is null；

select \* from student where birthday is null;

### --删除数据

delete from student; --删除所有记录，表结构还在，写日志，可以恢复的，速度慢。

--delete的数据可以恢复。

savepoint a; --创建保存点

delete from student;

rollback to a; --恢复到保存点

一个有经验的dba，在确保完成无误的情况下要定期创建还原点。

drop table student; --删除表的结构和数据；

delete from student where xh = 'a001'; --删除一条记录；

truncate table student; --删除表中的所有记录，表结构还在，不写日志，无法找回删除的记录，速度快。

# [六、表查询一](http://www.cnblogs.com/linjiqin/archive/2012/02/02/2335347.html)

通过scott用户下的表来演示如何使用select语句，接下来对emp、dept、salgrade表结构进行解说。

emp 雇员表  
字段名称   数据类型       是否为空   备注  
--------   -----------   --------   --------  
EMPNO    NUMBER(4)                 员工编号                  
ENAME    VARCHAR2(10) Y         员工名称          
JOB        VARCHAR2(9)   Y         职位                  
MGR       NUMBER(4)       Y         上级的编号              
HIREDATE DATE             Y         入职日期                 
SAL         NUMBER(7,2)   Y         月工资              
COMM      NUMBER(7,2)   Y         奖金                  
DEPTNO   NUMBER(2)      Y         所属部门  
-------------------------------------------  
job字段：  
clerk 普员工  
salesman 销售  
manager 经理  
analyst 分析师  
president 总裁

dept 部门表  
字段名称    数据类型          是否为空   备注  
--------    -----------      --------   --------  
DEPTNO   NUMBER(2)                    部门编号                 
DNAME    VARCHAR2(14)    Y          部门名称          
LOC         VARCHAR2(13)   Y          部门所在地点  
-------------------------------------------  
DNAME字段：  
accounting 财务部  
research 研发部  
operations 业务部  
    
salgrade 工资级别表  
字段名称  数据类型   是否为空  备注  
--------  ---------  --------  --------  
GRADE   NUMBER     Y         级别                  
LOSAL    NUMBER     Y         最低工资                 
HISAL     NUMBER     Y         最高工资

1、查看表结构  
desc emp;

2、查询所有列  
select \* from dept;  
备注:切忌动不动就用select \*，使用\*效率比较低，特别在大表中要注意。  
          
3、set timing on/off;  
打开显示操作时间的开关，在底部显示操作时间。  
eg、sql> insert into tb\_stu values('0001', 'zhangsan', 24);   
1 row inserted  
executed in 0.015 seconds

4、insert into...select...表复制语句  
语法：insert into table2(field1,field2,...) select value1,value2,... from table1

--创建tb\_dept表  
create table tb\_dept  
(  
  deptno number(4) not null,  
  dname  varchar2(14),  
  loc    varchar2(13)  
)  
--添加主键约束  
alter table tb\_dept add constraint tb\_dept primary key (deptno);

--insert into...select...用法  
insert into tb\_dept (deptno, dname, loc) select a.deptno, a.dname, a.loc from dept a;

5、统计  
select count (\*) from emp;

6、查询指定列  
select ename, sal, job, deptno from emp;

7、如何取消重复行distinct  
select distinct deptno, job from emp;

8、查询smith所在部门，工作，薪水  
select deptno, job, sal from emp where ename = 'smith';  
注意：oracle对内容的大小写是敏感的，所以ename='smith'和ename='smith'是不同的

9、nvl函数  
格式为：nvl(string1, replace\_with) 　　  
功能：如果string1为null，则nvl函数返回replace\_with的值，否则返回string1的值。 　  
注意事项：string1和replace\_with必须为同一数据类型，除非显示的使用to\_char函数。 　  
eg、如何显示每个雇员的年工资？  
select sal\*13+nvl(comm, 0)\*13 "年薪" , ename, comm from emp;

10、使用列的别名  
select ename "姓名", sal\*12 as "年收入" from emp;

11、如何处理null值  
使用nvl函数来处理

12、如何连接字符串(||)  
select ename || ' is a ' || job from emp;

13、使用where子句  
问题：如何显示工资高于3000的员工？  
select \* from emp where sal > 3000;  
问题：如何查找1982.1.1后入职的员工？  
select ename,hiredate from emp where hiredate >'1-1 月-1982';  
问题：如何显示工资在2000到3000的员工？  
select ename,sal from emp where sal>=2000 and sal<=3000;

14、如何使用like操作符  
%：表示0到多个字符 \_：表示任意单个字符  
问题：如何显示首字符为s的员工姓名和工资？  
select ename,sal from emp where ename like 's%';  
如何显示第三个字符为大写o的所有员工的姓名和工资？  
select ename,sal from emp where ename like '\_\_o%';

15、在where条件中使用in  
问题：如何显示empno为7844,7839,123,456的雇员情况？  
select \* from emp where empno in (7844, 7839, 123, 456);

1. 使用is null的操作符  
   问题：如何显示没有上级的雇员的情况？  
   错误写法：select \* from emp where mgr = '';  
   正确写法：select \* from emp where mgr is null;

# 七、oracle 表查询二

## 1、使用逻辑操作符号

问题：查询工资高于500或者是岗位为manager的雇员，同时还要满足他们的姓名首字母为大写的J？

select \* from emp where (sal > 500 or job = 'MANAGER') and ename like 'J%';

## 2、使用order by字句 默认asc

问题：如何按照工资从低到高的顺序显示雇员的信息？

select \* from emp order by sal;

问题：按照部门号升序而雇员的工资降序排列

select \* from emp order by deptno, sal desc;

## 3、使用列的别名排序

问题：按年薪排序

select ename, (sal+nvl(comm,0))\*12 "年薪" from emp order by "年薪" asc;

备注：别名需要使用“”号圈中,英文不需要“”号

## 4、聚合函数用法：max，min，avg，sum，count

问题：如何显示所有员工中最高工资和最低工资？

select max(sal),min(sal) from emp e;

最高工资那个人是谁？

错误写法：select ename, sal from emp where sal=max(sal);

正确写法：select ename, sal from emp where sal=(select max(sal) from emp);

注意：select ename, max(sal) from emp;这语句执行的时候会报错，说ora-00937：非单组分组函数。因为max是分组函数，而ename不是分组函数.......

但是select min(sal), max(sal) from emp;这句是可以执行的。因为min和max都是分组函数，就是说：如果列里面有一个分组函数，其它的都必须是分组函数，否则就出错。这是语法规定的

## 5、问题：如何显示所有员工的平均工资和工资总和？

select sum(e.sal), avg(e.sal) from emp e;

查询最高工资员工的名字，工作岗位

select ename, job, sal from emp e where sal = (select max(sal) from emp);

显示工资高于平均工资的员工信息

select \* from emp e where sal > (select avg(sal) from emp);

## 6、group by 和 having 子句

group by 用于对查询的结果分组统计，

having 子句用于限制分组显示结果。

问题：如何显示每个部门的平均工资和最高工资？

select avg(sal), max(sal), deptno from emp group by deptno;

（注意：这里暗藏了一点，如果你要分组查询的话，分组的字段deptno一定要出现在查询的列表里面，否则会报错。因为分组的字段都不出现的话，就没办法分组了）

问题：显示每个部门的每种岗位的平均工资和最低工资？

select min(sal), avg(sal), deptno, job from emp group by deptno, job;

问题：显示平均工资低于2000的部门号和它的平均工资？

select avg(sal), max(sal), deptno from emp group by deptno having avg(sal)< 2000;

## 7、对数据分组的总结

1 分组函数只能出现在选择列表、having、order by子句中(不能出现在where中)

2 如果在select语句中同时包含有group by, having, order by 那么它们的顺序是group by, having, order by

3 在选择列中如果有列、表达式和分组函数，那么这些列和表达式必须有一个出现在group by 子句中，否则就会出错。

如select deptno, avg(sal), max(sal) from emp group by deptno having avg(sal) < 2000;这里deptno就一定要出现在group by中

## 8、多表查询

多表查询是指基于两个和两个以上的表或是视图的查询。在实际应用中，查询单个表可能不能满足你的需求，如显示sales部门位置和其员工的姓名，这种情况下需要使用到dept表和emp表。

1)、问题：显示雇员名，雇员工资及所在部门的名字【笛卡尔集】？

SELECT e.ename, e.sal, d.dname FROM emp e, dept d WHERE e.deptno = d.deptno;

规定：多表查询的条件是至少不能少于表的个数N-1才能排除笛卡尔集（如果有N张表联合查询，必须得有N-1个条件，才能避免笛卡尔集合）

2)、问题：显示部门号为10的部门名、员工名和工资？

SELECT d.dname, e.ename, e.sal FROM emp e, dept d WHERE e.deptno = d.deptno and e.deptno = 10;

3)、问题：显示各个员工的姓名，工资及工资的级别？

SELECT e.ename, e.sal, s.grade FROM emp e, salgrade s WHERE e.sal BETWEEN s.losal AND s.hisal;

4)、问题：显示雇员名，雇员工资及所在部门的名字，并按部门排序？

SELECT e.ename, e.sal, d.dname FROM emp e, dept d WHERE e.deptno = d.deptno ORDER by e.deptno;

注意：如果用group by，一定要把e.deptno 放到查询列里面

5)、自连接

自连接是指在同一张表的连接查询

问题：显示某个员工的上级领导的姓名？

比如显示员工‘FORD’的上级

SELECT worker.ename, boss.ename FROM emp worker, emp boss WHERE worker.mgr = boss.empno AND worker.ename = 'FORD';

6)、子查询

什么是子查询？

子查询是指嵌入在其他sql语句中的select语句，也叫嵌套查询。

单行子查询？

单行子查询是指只返回一行数据的子查询语句

请思考：显示与SMITH同部门的所有员工？

思路：

1))、查询出SMITH的部门号(返回单行结果集)

select deptno from emp WHERE ename = 'SMITH';

2))、显示

SELECT \* FROM emp WHERE deptno = (select deptno from emp WHERE ename = 'SMITH');

数据库在执行sql是从左到右扫描的，如果有括号的话，括号里面的先被优先执行。

多行子查询

多行子查询指返回多行数据的子查询

请思考：如何查询和部门10的工作相同的雇员的名字、岗位、工资、部门号

1))、查询出部门10的所有工作(返回多行结果集)

SELECT DISTINCT job FROM emp WHERE deptno = 10;

2))、显示

SELECT \* FROM emp WHERE job IN (SELECT DISTINCT job FROM emp WHERE deptno = 10);

注意：不能用job=..，因为等号=是一对一的

在多行子查询中使用all操作符

问题：如何显示工资比部门30的所有员工的工资高的员工的姓名、工资和部门号？

--方法一

SELECT ename, sal, deptno FROM emp WHERE sal > all(SELECT sal FROM emp WHERE deptno = 30);

--方法二(执行效率最高，使用聚合函数)

SELECT ename, sal, deptno FROM emp WHERE sal > (SELECT max(sal) FROM emp WHERE deptno = 30);

在多行子查询中使用any操作符

问题：如何显示工资比部门30的任意一个员工的工资高的员工姓名、工资和部门号？

--方法一

SELECT ename, sal, deptno FROM emp WHERE sal > ANY (SELECT sal FROM emp WHERE deptno = 30);

--方法二(执行效率最高，使用聚合函数)

SELECT ename, sal, deptno FROM emp WHERE sal > (SELECT min(sal) FROM emp WHERE deptno = 30);

多列子查询

单行子查询是指子查询只返回单列、单行数据，多行子查询是指返回单列多行数据，都是针对单列而言的，而多列子查询是指查询返回多个列数据的子查询语句。

请思考如何查询与SMITH 的部门和岗位完全相同的所有雇员。

SELECT deptno, job FROM emp WHERE ename = 'SMITH';

SELECT \* FROM emp WHERE (deptno, job) = (SELECT deptno, job FROM emp WHERE ename = 'SMITH');

在from子句中使用子查询

请思考：如何显示高于自己部门平均工资的员工的信息

思路：

1. 查出各个部门的平均工资和部门号

SELECT deptno, AVG(sal) mysal FROM emp GROUP by deptno;

2. 把上面的查询结果看做是一张子表

SELECT e.ename, e.deptno, e.sal, ds.mysal

FROM emp e, (SELECT deptno, AVG(sal) mysal FROM emp GROUP by deptno) ds

WHERE e.deptno = ds.deptno AND e.sal > ds.mysal;

小总结：

在这里需要说明的当在from子句中使用子查询时，该子查询会被作为一个视图来对待，因此叫做内嵌视图，当在from 子句中使用子查询时，必须给子查询指定别名。

注意：别名不能用as，如：SELECT e.ename, e.deptno, e.sal, ds.mysal FROM emp e, (SELECT deptno, AVG(sal) mysal FROM emp GROUP by deptno) as ds WHERE e.deptno = ds.deptno AND e.sal > ds.mysal;

在ds前不能加as，否则会报错（给表取别名的时候，不能加as；但是给列取别名，是可以加as的）

如何衡量一个程序员的水平？

网络处理能力，数据库，程序代码的优化程序的效率要很高

7)、用查询结果创建新表，这个命令是一种快捷的建表方式

CREATE TABLE mytable (id, name, sal, job, deptno) as SELECT empno, ename, sal, job, deptno FROM emp;

8)、合并查询

有时在实际应用中，为了合并多个select语句的结果，可以使用集合操作符号union，union all，intersect，minus。

多用于数据量比较大的数据局库，运行速度快。

1). union

该操作符用于取得两个结果集的并集。当使用该操作符时，会自动去掉结果集中重复行。

SELECT ename, sal, job FROM emp WHERE sal >2500

UNION

SELECT ename, sal, job FROM emp WHERE job = 'MANAGER';

2).union all

该操作符与union相似，但是它不会取消重复行，而且不会排序。

SELECT ename, sal, job FROM emp WHERE sal >2500

UNION ALL

SELECT ename, sal, job FROM emp WHERE job = 'MANAGER';

该操作符用于取得两个结果集的并集。当使用该操作符时，不会自动去掉结果集中重复行。

3). intersect

使用该操作符用于取得两个结果集的交集。

SELECT ename, sal, job FROM emp WHERE sal >2500

INTERSECT

SELECT ename, sal, job FROM emp WHERE job = 'MANAGER';

4). minus

使用该操作符用于取得两个结果集的差集，他只会显示存在第一个集合中，而不存在第二个集合中的数据。

SELECT ename, sal, job FROM emp WHERE sal >2500

MINUS

SELECT ename, sal, job FROM emp WHERE job = 'MANAGER';

（MINUS就是减法的意思）

# [八、oracle 分页](http://www.cnblogs.com/linjiqin/archive/2012/02/03/2336731.html)

oracle的分页一共有三种方式

## 方法一 根据rowid来分

[IMG_256](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

SELECT \*  
 FROM EMP  
 WHERE ROWID IN  
 (SELECT RID  
 FROM (SELECT ROWNUM RN, RID  
 FROM (SELECT ROWID RID, EMPNO FROM EMP ORDER BY EMPNO DESC)  
 WHERE ROWNUM <= ( (currentPage-1) \* pageSize + pageSize )) --每页显示几条  
 WHERE RN > ((currentPage-1) \* pageSize) ) --当前页数  
 ORDER BY EMPNO DESC;  
   
eg、  
-- 5 = (currentPage-1) \* pageSize + pageSize 每页显示几条  
-- 0 = (currentPage-1) \* pageSize 当前页数  
SELECT \*  
 FROM EMP  
 WHERE ROWID IN  
 (SELECT RID  
 FROM (SELECT ROWNUM RN, RID  
 FROM (SELECT ROWID RID, EMPNO FROM EMP ORDER BY EMPNO DESC)  
 WHERE ROWNUM <= ( (1-1) \* 5 + 5 )) --每页显示几条  
 WHERE RN > ((1-1) \* 5) ) --当前页数  
 ORDER BY EMPNO DESC;

## 方法二 按分析函数来分

[IMG_258](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

SELECT \*  
FROM (SELECT T.\*, ROW\_NUMBER() OVER(ORDER BY empno DESC) RK FROM emp T)  
WHERE RK <= ( (currentPage-1) \* pageSize + pageSize ) --每页显示几条  
AND RK > ( (currentPage-1) \* pageSize ); --当前页数

eg、  
-- 5 = (currentPage-1) \* pageSize + pageSize 每页显示几条  
-- 0 = (currentPage-1) \* pageSize 当前页数  
SELECT \*  
FROM (SELECT T.\*, ROW\_NUMBER() OVER(ORDER BY empno DESC) RK FROM emp T)  
WHERE RK <= 5  
AND RK > 0;

[IMG_259](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

## 方法三 按rownum 来分

[IMG_260](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

SELECT \*  
 FROM (SELECT T.\*, ROWNUM RN  
 FROM (SELECT \* FROM EMP ORDER BY EMPNO DESC) T  
 WHERE ROWNUM <= ( (currentPage-1) \* pageSize + pageSize )) --每页显示几条  
 WHERE RN > ( (currentPage-1) \* pageSize ); --当前页数  
   
eg、  
-- 5 = (currentPage-1) \* pageSize + pageSize 每页显示几条  
-- 0 = (currentPage-1) \* pageSize 当前页数  
SELECT \*  
 FROM (SELECT T.\*, ROWNUM RN  
 FROM (SELECT \* FROM EMP ORDER BY EMPNO DESC) T  
 WHERE ROWNUM <= 5)  
 WHERE RN > 0;

[IMG_261](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

其中emp为表名称，empno 为表的主键id，获取按empno降序排序后的第1-5条记录，emp表有70000 多条记录。  
个人感觉方法一的效率最好，方法三 次之，方法二 最差。

下面通过方法三来分析oracle怎么通过rownum分页的

[IMG_262](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

1、SELECT \* FROM emp;2、显示rownum，由oracle分配的SELECT e.\*, ROWNUM rn FROM (SELECT \* FROM emp) e; --rn相当于Oracle分配的行的ID号 3、先查出1-10条记录

正确的: SELECT e.\*, ROWNUM rn FROM (SELECT \* FROM emp) e WHERE ROWNUM<=10;

错误的：SELECT e.\*, ROWNUM rn FROM (SELECT \* FROM emp) e WHERE rn<=10;4、然后查出6-10条记录SELECT \* FROM (SELECT e.\*, ROWNUM rn FROM (SELECT \* FROM emp) e WHERE ROWNUM<=10) WHERE rn>=6;

[IMG_263](http://www.cnblogs.com/linjiqin/archive/2012/02/03/javascript:void(0);)

# [十、oracle 常用函数](http://www.cnblogs.com/linjiqin/archive/2012/02/07/2341925.html)

一、字符函数  
字符函数是oracle中最常用的函数，我们来看看有哪些字符函数：  
lower(char)：将字符串转化为小写的格式。  
upper(char)：将字符串转化为大写的格式。  
length(char)：返回字符串的长度。  
substr(char, m, n)：截取字符串的子串，n代表取n个字符的意思，不是代表取到第n个  
replace(char1, search\_string, replace\_string)  
instr(C1,C2,I,J) -->判断某字符或字符串是否存在，存在返回出现的位置的索引，否则返回小于1;在一个字符串中搜索指定的字符,返回发现指定的字符的位置;  
C1 被搜索的字符串  
C2 希望搜索的字符串  
I 搜索的开始位置,默认为1  
J 出现的位置,默认为1

问题：将所有员工的名字按小写的方式显示  
SQL> select lower(ename) from emp;  
问题：将所有员工的名字按大写的方式显示。  
SQL> select upper(ename) from emp;  
问题：显示正好为5个字符的员工的姓名。  
SQL> select \* from emp where length(ename)=5;  
问题：显示所有员工姓名的前三个字符。  
SQL> select substr(ename, 1, 3) from emp;  
问题：以首字母大写,后面小写的方式显示所有员工的姓名。  
SQL> select upper(substr(ename,1,1)) || lower(substr(ename,2,length(ename)-1)) from emp;  
问题：以首字母小写,后面大写的方式显示所有员工的姓名。  
SQL> select lower(substr(ename,1,1)) || upper(substr(ename,2,length(ename)-1)) from emp;  
问题：显示所有员工的姓名，用“我是老虎”替换所有“A”  
SQL> select replace(ename,'A', '我是老虎') from emp;  
问题：instr(char1,char2,[,n[,m]])用法  
SQL> select instr('azhangsanbcd', 'zhangsan') from dual; --返回2  
SQL> select instr('oracle traning', 'ra', 1, 1) instring from dual; --返回2  
SQL> select instr('oracle traning', 'ra', 1, 2) instring from dual; --返回9  
SQL> select instr('oracle traning', 'ra', 1, 3) instring from dual; --返回0，根据条件，由于ra只出现二次，第四个参数3，就是说第3次出现ra的位置，显然第3次是没有再出现了，所以结果返回0。注意空格也算一个字符  
SQL> select instr('abc','d') from dual;  --返回0

二、数学函数  
数学函数的输入参数和返回值的数据类型都是数字类型的。数学函数包括cos，cosh，exp，ln, log，sin，sinh，sqrt，tan，tanh，acos，asin，atan，round等  
我们讲最常用的：  
round(n,[m]) 该函数用于执行四舍五入，  
如果省掉m，则四舍五入到整数。  
如果m是正数，则四舍五入到小数点的m位后。  
如果m是负数，则四舍五入到小数点的m位前。  
eg、SELECT round(23.75123) FROM dual; --返回24  
SELECT round(23.75123, -1) FROM dual; --返回20  
SELECT round(27.75123, -1) FROM dual; --返回30  
SELECT round(23.75123, -3) FROM dual; --返回0  
SELECT round(23.75123, 1) FROM dual; --返回23.8  
SELECT round(23.75123, 2) FROM dual; --返回23.75  
SELECT round(23.75123, 3) FROM dual; --返回23.751  
trunc(n,[m]) 该函数用于截取数字。  
如果省掉m，就截去小数部分，  
如果m是正数就截取到小数点的m位后，  
如果m是负数，则截取到小数点的前m位。  
eg、SELECT trunc(23.75123) FROM dual; --返回23  
SELECT trunc(23.75123, -1) FROM dual; --返回20  
SELECT trunc(27.75123, -1) FROM dual; --返回20  
SELECT trunc(23.75123, -3) FROM dual; --返回0  
SELECT trunc(23.75123, 1) FROM dual; --返回23.7  
SELECT trunc(23.75123, 2) FROM dual; --返回23.75  
SELECT trunc(23.75123, 3) FROM dual; --返回23.751  
mod(m,n)取余函数  
eg、select mod(10,2) from dual; --返回0  
SELECT MOD(10,3) FROM dual; --返回1  
floor(n) 返回小于或是等于n的最大整数  
ceil(n) 返回大于或是等于n的最小整数  
eg、SELECT ceil(24.56) from dual; --返回25  
SELECT floor(24.56) from dual; --返回24  
abs(n) 返回数字n的绝对值  
对数字的处理，在财务系统或银行系统中用的最多，不同的处理方法，对财务报表有不同的结果

三、日期函数  
日期函数用于处理date类型的数据。默认情况下日期格式是dd-mon-yy 即“12-7 月-12”  
(1)sysdate 返回系统时间  
eg、SQL> select sysdate from dual;  
(2)oracle add\_months函数  
oracle add\_months(time,months)函数可以得到某一时间之前或之后n个月的时间  
eg、select add\_months(sysdate,-6) from dual; --该查询的结果是当前时间半年前的时间  
select add\_months(sysdate,6) from dual; --该查询的结果是当前时间半年后的时间  
(3)last\_day(d)：返回指定日期所在月份的最后一天  
问题：查找已经入职8个月多的员工  
SQL> select \* from emp where sysdate>=add\_months(hiredate,8);  
问题：显示满10年服务年限的员工的姓名和受雇日期。  
SQL> select ename, hiredate from emp where sysdate>=add\_months(hiredate,12\*10);  
问题：对于每个员工，显示其加入公司的天数。  
SQL> select floor(sysdate-hiredate) "入职天数",ename from emp;  
或者  
SQL> select trunc(sysdate-hiredate) "入职天数",ename from emp;  
问题：找出各月倒数第3天受雇的所有员工。  
SQL> select hiredate,ename from emp where last\_day(hiredate)-2=hiredate;  
             
四、转换函数  
转换函数用于将数据类型从一种转为另外一种。在某些情况下，oracle server允许值的数据类型和实际的不一样，这时oracle server会隐含的转化数据类型  
比如：  
create table t1(id int);  
insert into t1 values('10');--这样oracle会自动的将'10' -->10  
create table t2 (id varchar2(10));  
insert into t2 values(1); --这样oracle就会自动的将1 -->'1'；  
我们要说的是尽管oracle可以进行隐含的数据类型的转换，但是它并不适应所有的情况，为了提高程序的可靠性，我们应该使用转换函数进行转换。

to\_char()函数  
你可以使用select ename, hiredate, sal from emp where deptno = 10;显示信息，可是，在某些情况下，这个并不能满足你的需求。  
问题：日期是否可以显示 时/分/秒  
SQL> select ename, to\_char(hiredate, 'yyyy-mm-dd hh24:mi:ss') from emp;  
问题：薪水是否可以显示指定的货币符号  
SQL>  
yy：两位数字的年份 2004-->04  
yyyy：四位数字的年份 2004年  
mm：两位数字的月份 8 月-->08  
dd：两位数字的天 30 号-->30  
hh24： 8点-->20  
hh12：8点-->08  
mi、ss-->显示分钟\秒  
9：显示数字，并忽略前面0  
0：显示数字，如位数不足，则用0补齐  
.：在指定位置显示小数点  
,：在指定位置显示逗号  
$：在数字前加美元  
L：在数字前面加本地货币符号  
C：在数字前面加国际货币符号  
G：在指定位置显示组分隔符、  
D：在指定位置显示小数点符号(.)

问题：显示薪水的时候，把本地货币单位加在前面  
SQL> select ename, to\_char(hiredate, 'yyyy-mm-dd hh24:mi:ss'), to\_char(sal,'L99999.99') from emp;  
问题：显示1980年入职的所有员工  
SQL> select \* from emp where to\_char(hiredate, 'yyyy')=1980;  
问题：显示所有12月份入职的员工  
SQL> select \* from emp where to\_char(hiredate, 'mm')=12;

to\_date()函数  
函数to\_date用于将字符串转换成date类型的数据。  
问题：能否按照中国人习惯的方式年—月—日添加日期。  
eg、SELECT to\_date('2012-02-18 09:25:30','yyyy-mm-dd hh24:mi:ss') FROM dual;

五、sys\_context()系统函数  
1)terminal：当前会话客户所对应的终端的标示符，如计算机名  
2)language: 语言  
3)db\_name： 当前数据库名称  
4)nls\_date\_format： 当前会话客户所对应的日期格式  
5)session\_user： 当前会话客户所对应的数据库用户名  
6)current\_schema： 当前会话客户所对应的默认方案名  
7)host： 返回数据库所在主机的名称  
通过该函数，可以查询一些重要信息，比如你正在使用哪个数据库？  
select sys\_context('USERENV','db\_name') from dual;  
注意：USERENV是固定的，不能改的，db\_name可以换成其它,  
eg、select sys\_context('USERENV','language') from dual;  
select sys\_context('USERENV','current\_schema') from dual;

# [十九、oracle pl/sql简介](http://www.cnblogs.com/linjiqin/archive/2012/02/22/2363781.html)

一、pl/sql 是什么  
pl/sql(procedural language/sql)是oracle在标准的sql语言上的扩展。  
pl/sql不仅允许嵌入sql语言，还可以定义变量和常量，允许使用条件语句和循环语句，允许使用例外处理各种错误，这样使得它的功能变得更加强大。

二、为什么要学pl/sql  
1.提高应用程序的运行性能  
2.模块化的设计思想(分页的过程，订单的过程，转账的过程。。)  
3.减少网络传输量  
4.提高安全性(sql会包括表名，有时还可能有密码，传输的时候会泄露。PL/SQL就不会)

三、Oracle为什么在PL/SQL developer执行很快,用c# oracleclient执行就慢  
因为PL/SQL这门语言是专门用于在各种环境下对Oracle数据库进行访问。由于该语言集成于数据库服务器中，所以PL/SQL代码可以对数据进行快速高效的处理。  
而c#语言是微软的产品，它在连接ORACLE的时候先存到“连接池”中，所以第一次会慢点，但是当你的Web程序没有重起的时候，以后的速度就不会慢了。

四、使用pl/sql的缺点  
移植性不好(换数据库就用不了)

五、pl/sql理解  
1)、存储过程、函数、触发器是pl/sql编写的  
2)、存储过程、函数、触发器是存在oracle中的  
3)、pl/sql是非常强大的数据库过程语言  
4)、存储过程、函数可以在java中调用

## 六、编写一个存储过程，该过程可以向某表中添加记录。

[IMG_256](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

1、创建一张简单的表CREATE TABLE mytest(

username VARCHAR2(30),

pwd VARCHAR2(30)

);

2、创建过程(replace:表示如果有insert\_proc，就替换)CREATE OR REPLACE PROCEDURE insert\_proc ISBEGIN

INSERT INTO mytest VALUES('林计钦', '123456');END;/

3、如何查看错误信息：show error;

注意要在命令窗口执行

4、如何调用该过程：exec 过程名(参数值1，参数值2...);

eg、exec insert\_proc;

注意要在命令窗口执行

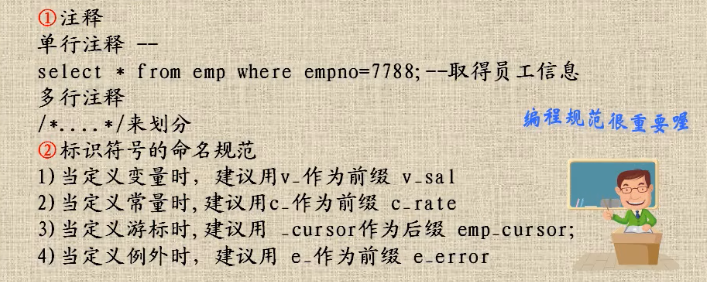
# [二十、oracle pl/sql基础](http://www.cnblogs.com/linjiqin/archive/2012/02/22/2363869.html)

一、pl/sql developer开发工具  
pl/sql developer是用于开发pl/sql块的集成开发环境(ide)，它是一个独立的产品，而不是oracle的一个附带品。

二、pl/sql介绍  
开发人员使用pl/sql编写应用模块时，不仅需要掌握sql语句的编写方法，还要掌握pl/sql语句及语法规则。pl/sql编程可以使用变量和逻辑控制语句，从而可以编写非常有用的功能模块。比如：分页存储过程模块、订单处理存储过程模块、转账存储过程模块。而且如果使用pl/sql编程，我们可以轻松地完成非常复杂的查询要求。

三、pl/sql可以做什么  
可以用来编写存储过程、函数、触发器、包等

## 四、编写规范



五、pl/sql块介绍  
块(block)是pl/sql的基本程序单元，编写pl/sql程序实际上就是编写pl/sql块，要完成相对简单的应用功能，可能只需要编写一个pl/sql块，但是如果想要实现复杂的功能，可能需要在一个pl/sql块中嵌套其它的pl/sql块。

六、块结构示意图  
pl/sql块由三个部分构成：定义部分，执行部分，例外处理部分。  
如下所示：  
declare  
/\*定义部分——定义常量、变量、游标、例外、复杂数据类型\*/  
begin  
/\*执行部分——要执行的pl/sql 语句和sql 语句\*/  
exception  
/\*例外处理部分——处理运行的各种错误\*/  
end;

说明：  
定义部分是从declare开始的，该部分是可选的；  
执行部分是从begin开始的，该部分是必须的；  
例外处理部分是从exception开始的，该部分是可选的。  
可以和java编程结构做一个简单的比较。

七、pl/sql块的实例一

实例一 只包括执行部分的pl/sql块

[IMG_257](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

set serveroutput on; --打开输出选项  
  
begin   
 dbms\_output.put\_line('hello world');  
end;  
/ --执行

[IMG_258](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

相关说明：  
dbms\_output是oracle所提供的包(类似java 的开发包)，该包包含一些过程，put\_line就是dbms\_output包的一个过程。

八、pl/sql块的实例二

实例二 包含定义部分和执行部分的pl/sql块

[IMG_259](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

set serveroutput on; --打开输出选项  
DECLARE  
 --定义字符串变量  
 v\_ename varchar2(10);   
BEGIN  
 --执行部分  
 select ename into v\_ename from emp where empno=&empno; --& 表示要接收从控制台输入的变量

--在控制台显示雇员名  
  
 dbms\_output.put\_line('雇员名：'||v\_ename);  
end;  
/

[IMG_260](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

九、pl/sql块的实例三  
实例三 包含定义部分，执行部分和例外处理部分  
为了避免pl/sql程序的运行错误，提高pl/sql的健壮性，应该对可能的错误进行处理，这个很有必要。  
1.比如在实例二中，如果输入了不存在的雇员号，应当做例外处理。  
2.有时出现异常，希望用另外的逻辑处理，我们看看如何完成1的要求。  
           
相关说明：oracle事先预定义了一些例外，no\_data\_found就是找不到数据的例外

[IMG_261](http://www.cnblogs.com/linjiqin/archive/2012/02/22/javascript:void(0);)

--打开输出选项  
set serveroutput on;   
DECLARE  
 --定义字符串变量  
 v\_ename varchar2(10);   
 v\_sal NUMBER(7,2);  
BEGIN  
 --执行部分  
 select ename, sal into v\_ename, v\_sal from emp where empno=&empno;   
 dbms\_output.put\_line('雇员名：'||v\_ename||'，薪水：'||v\_sal);  
EXCEPTION  
 --异常处理   
 WHEN no\_data\_found THEN dbms\_output.put\_line('朋友，您的编号输入有误！');  
end;  
/

# **[oracle系统包—-dbms\_output用法](http://www.cnblogs.com/linjiqin/p/3152647.html)**

## 例子一、put和new\_line

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | set serveroutput on;  begin     dbms\_output.put('a'); --写入buffer但不输出     dbms\_output.put('b'); --写入buffer但不输出     dbms\_output.new\_line; --回车(换行)，输出     dbms\_output.put\_line('hello world!'); --输出并换行     dbms\_output.put('d'); --写入buffer但不输出  end;  / |

执行运行结果：

|  |  |
| --- | --- |
| 1  2 | ab  hello world! |

## 例子二、put\_line

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | set serveroutput off;  create table t(a int, b int, c int);  insert into t values(111111,222222,333333);  insert into t values(444444,555555,666666);  insert into t values(777777,888888,999999);  commit;    create table tt(a int,b varchar2(100));    declare     msg varchar2(120);     cursor t\_cur is select \* from t order by a;     v\_line varchar2(100);     v\_status integer := 0;  begin     dbms\_output.enable;     for i in t\_cur loop         msg := i.a || ',' || i.b || ',' || i.c;         dbms\_output.put\_line(msg); --put     end loop;       dbms\_output.get\_line(v\_line, v\_status); --get     while v\_status = 0 loop         insert into tt values(v\_status, v\_line);         dbms\_output.get\_line(v\_line, v\_status);     end loop;  end;  /    select \* from tt; |

执行结果如下：

|  |  |
| --- | --- |
| 1  2  3  4  5 | a    b  --- -----------------------  0   111111,222222,333333  0   444444,555555,666666  0   777777,888888,999999 |

注：使用get\_line时不能用put\_line输出，因为put\_line之后会将buffer清空。(当然在serveroutput off的情况下put\_line是不影响buffer的)。

## 例子三：put\_lines

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | set serveroutput on;  declare     v\_data dbms\_output.chararr;     v\_numlines number;  begin     --enable the buffer first.     dbms\_output.enable(1000000);       dbms\_output.put\_line('line one');     dbms\_output.put\_line('line two');     dbms\_output.put\_line('line three');       v\_numlines := 3;     dbms\_output.get\_lines(v\_data, v\_numlines);  --array, index     for v\_counter in 1..v\_numlines loop         dbms\_output.put\_line(v\_data(v\_counter));     end loop;  end;  / |

执行结果如下：

|  |  |
| --- | --- |
| 1  2  3 | line one  line two  line three |

注意数组使用的格式。