# 1、修改数据库密码（不同数据库不一样）

## 1、postgresql

### ALTER USER postgres with password '123456';

## 2、MySQL

### set password for root@localhost=password('147094');

### 3、oracle

### alter usr healerjean identified by 147094

# 2、having

### 数据库中where与having区别

having 和where 都是用来筛选用的   
having 是筛选组 而where是筛选记录  
1》当分组筛选的时候 用having  
2》其它情况用where  
-----------------------------------------------------  
用having就一定要和group by连用，  
用group by不一有having （它只是一个筛选条件用的）

<http://blog.csdn.net/yangyi22/article/details/7483859>

### 举例

**一、给定一张学生成绩表tb\_grade：学号，姓名，科目号，科目名，分数。（如下表）**

（select \* from tb\_grade order by Sno,Cno;）  
+------+-------+-----+-------+-------+  
| Sno  | Sname | Cno | Cname | score |  
+------+-------+-----+-------+-------+  
| 1001 | 李菲  |   1 | 语文  |    86 |  
| 1001 | 李菲  |   2 | 数学  |    56 |  
| 1001 | 李菲  |   3 | 物理  |    48 |  
| 1001 | 李菲  |   4 | 化学  |    90 |  
| 1001 | 李菲  |   5 | 英语  |    57 |  
| 1002 | 王琪  |   1 | 语文  |    90 |  
| 1004 | 和树  |   3 | 物理  |    67 |  
| 1004 | 和树  |   4 | 化学  |    30 |  
| 1004 | 和树  |   5 | 英语  |    68 |  
+------+-------+-----+-------+-------+

 1、查询不及格科目数大于等于2的学生学号和不及格科目数量：

[**MySQL**](http://lib.csdn.net/base/mysql)> select Sno,count(score) as '不及格科目数' from tb\_grade where score<60 group by Sno having count(score)>=2;

+------+--------------+

# 3、distinct

如何用distinct消除重复记录的同时又能选取多个字段值？

需求是：我要消除name字段值重复的记录，同时又要得到id字段的值，其中id是自增字段。   
select distinct name from t1 能消除重复记录，但只能取一个字段，现在要同时取id,name这2个字段的值。   
select distinct id,name from t1 可以取多个字段，但只能消除这2个字段值全部相同的记录   
最后解决方法：   
SELECT *id*,name FROM t1 WHERE id IN(SELECT MAX(id) FROM t1 GROUP BY name) order by *id* desc

注意开头的 *id* 的 一定要，后面的order by 里有的字段一定要加进select 结果，要不然排序无效

# 4. SELECT嵌套

嵌套SELECT语句也叫子查询，一个 SELECT 语句的查询结果能够作为另一个语句的输入值。子查询不但能够出现在Where子句中，也能够出现在from子句中，作为一个临时表使用，也能够出现在select list中，作为一个字段值来返回。

## 1、单行子查询 ：单行子查询是指子查询的返回结果只有一行数据。当主查询语句的条件语句中引用子查询结果时可用单行比较符号（＝, >, <, >=, <=, <>）来进行比较。

例：  
select ename,deptno,sal  
from emp  
where deptno=(select deptno from dept where loc='NEW YORK')；  
  
2、多行子查询：多行子查询即是子查询的返回结果是多行数据。当主查询语句的条件语句中引用子查询结果时必须用多行比较符号（IN，ALL,ANY）来进行比较。其中，IN的含义是匹配子查询结果中的任一个值即可（"IN" 操作符，能够测试某个值是否在一个列表中），ALL则必须要符合子查询的所有值才可，ANY要符合子查询结果的任何一个值即可。而且须注意ALL 和ANY 操作符不能单独使用，而只能与单行比较符（=、>、< 、>= 、<= 、<>）结合使用。

例：

## 2、.多行子查询使用IN操作符号例子：查询选修了老师名叫Rona(假设唯一)的学生名字

sql> select stName

from Student

where stId in(selectdistinct stId from score where teId=(select teId from teacher where teName='Rona'));

   查询所有部门编号为A的资料：

SELECT ename,job,sal

FROM EMP

WHERE deptno in ( SELECT deptno FROM dept WHERE dname LIKE 'A%')；  
2）.多行子查询使用ALL操作符号例子：查询有一门以上的成绩高于Kaka的最高成绩的学生的名字:

sql> select stName

from Student

where stId in(select distinct stId from score where score >all(select score from score where stId=(select stId from Student where stName= 'Kaka') ));

## 4、FROM (SELECT内联视图子查询，里面的表相当于外面的

例：  
(1)SELECT ename,job,sal,rownum  
FROM (SELECT ename,job,sal FROM EMP ORDER BY sal)；  
(2)SELECT ename,job,sal,rownum  
FROM ( SELECT ename,job,sal FROM EMP ORDER BY sal)  
WHERE rownum<=5；  
SELECT \* FROM  ( SELECT \* FROM  id\_type  );    
SELECT \* FROM id\_type t;   相同结果

第一个查了两次 第二个一次  一个有子查询

## 5、在HAVING子句中使用子查询

例：  
SELECT deptno,job,AVG(sal) FROM EMP GROUP BY deptno,job HAVING AVG(sal)>(SELECT sal FROM EMP WHERE ename='MARTIN')；

让我们再看看一些具体的实例，

# 5、直接根据字段导入另一张表

INSERT INTO "public"."tb\_student"(id,created\_on,email,mobile,pwd,real\_name,source,updated\_on) SELECT tn\_account.uuid\_accid,create\_date,email

,mobile,pwd,real\_name,source,modify\_date from tn\_account;

# 6、[select into和 inseret into select 两种表复制语句](http://www.cnblogs.com/freshman0216/archive/2008/08/15/1268316.html)

<http://www.cnblogs.com/freshman0216/archive/2008/08/15/1268316.html>

## 1、Insert into

语句形式为：Insert into Table2(field1,field2,...) select value1,value2,... from Table1

      要求目标表Table2必须存在，由于目标表Table2已经存在，所以我们除了插入源表Table1的字段外，还可以插入常量。示例如下：

### 举例：Insert into Table2(a, c, d) select a,c,5 from Table1

## 2、select into

SELECT vale1, value2 into Table2 from Table1

      要求目标表Table2不存在，因为在插入时会自动创建表Table2，并将Table1中指定字段数据复制到Table2中。示例如下：

### 举例select a,c INTO Table2 from Table1

# 7. declare 定义数据类型 （MySQL有oracle没有）

declare v\_number number(10);  
 v\_char varchar2(20);

## 解释：这个一般是用在函数和存储过程中的.

# 8. loop 循环

create or replace procedure pro\_test\_loop is  
i number;  
begin  
i:=0;  
loop  
  i:=i+1;  
  dbms\_output.put\_line(i);  
  if i>5 then  
    exit;  
  end if;  
end loop;  
end pro\_test\_loop;

# 9、函数

## 解释:用法

### 1、in表示传入，out表示输出

### 2、变量带取值范围，后面接分号RESULT\_VALUE:=INFO;

### 3、下面as和is区别

在存储过程(PROCEDURE)和函数(FUNCTION)中没有区别；http://blog.itpub.net/kindeditor/plugins/emoticons/images/20.gif  
在视图(VIEW)中只能用AS不能用IS；http://blog.itpub.net/kindeditor/plugins/emoticons/images/13.gif  
在游标(CURSOR)中只能用IS不能用AS。http://blog.itpub.net/kindeditor/plugins/emoticons/images/12.gif

## 1、编写大概的存储过程

|  |
| --- |
| create or replace function FNC\_IDNO\_CONVERT(INFO IN VARCHAR2) RETURN VARCHAR2 IS  RESULT\_VALUE varchar2(18);  BEGIN  //这里将来要写数据  return RESULT\_VALUE;  END FNC\_IDNO\_CONVERT; |

## 2、定义要用到的变量

|  |
| --- |
| create or replace function FNC\_IDNO\_CONVERT(INFO IN VARCHAR2) RETURN VARCHAR2 IS  RESULT\_VALUE varchar2(18);  i number;  nosum number;  temp number;  oneNum number;  indexno number;  fin number;  idLength number;  BEGIN  /\*这里将来要写数据\*/  return RESULT\_VALUE;  END FNC\_IDNO\_CONVERT; |

## 3、取得传入的参数进行赋值

|  |
| --- |
| create or replace function FNC\_IDNO\_CONVERT(INFO IN VARCHAR2) RETURN VARCHAR2 IS  RESULT\_VALUE varchar2(18);  i number;  nosum number;  temp number;  oneNum number;  indexno number;  fin number;  idLength number;  BEGIN  idLength:=length(INFO); //取得上面传入的的参数值  RESULT\_VALUE:=INFO;  /\*这里将来要写数据\*/  return RESULT\_VALUE;  END FNC\_IDNO\_CONVERT; |

## 4、使用各种条件判断和循环

### 1、if then end if

|  |
| --- |
| BEGIN  idLength:=length(INFO); //取得上面传入的的参数值  RESULT\_VALUE:=INFO;  /\*这里将来要写数据\*/  if idLength=18 then  RESULT\_VALUE:=substr(INFO, 0, 6) || substr(INFO, 9, 9);  end if;  if fin<10 then  begin  RESULT\_VALUE:=RESULT\_VALUE || to\_char(fin);  end;  end if;    2. if intip<0 **then** 3. **temp**:=intip-min\_int; 4. flag:=0; 5. **else** 6. **temp**:=intip; 7. flag:=1; 8. **end** if;   return RESULT\_VALUE; |

### 2、while loop begin 内容end; end loop;

|  |
| --- |
| i:=1;  nosum:=0;  while i<18 loop  begin  oneNum:=to\_number(substr(RESULT\_VALUE,i,1));  temp:=power(2,18-i);  temp:=mod(temp,11);  nosum:=nosum+oneNum\*temp;  i:=i+1;  end;  end loop; |

### 3、for循环

#### 1、数值型

|  |
| --- |
| --数值型For循环  procedure loop\_num\_for  (  lowest in number,  highest in number  )  is  begin  FOR even\_number in lowest .. highest --升序  // for i in (select one from table) loop  类似于这样的  loop  --处理非平滑增长的索引  if mod(even\_number,**2**)=**0**  then  dbms\_output.put\_line('now number:' || even\_number);  end if;  end loop;  end loop\_num\_for; |

#### 2、游标型

|  |
| --- |
| procedure loop\_cursor\_for  is  begin  declare cursor userinfo\_cur is select \* from userinfo\_table;  begin  FOR userinfo\_rec in userinfo\_cur  loop  dbms\_output.put\_line('username is:' || userinfo\_rec.user\_name);  end loop;  end;  end loop\_cursor\_for; |

# 10、存储过程

## 1、查询员工信息

|  |
| --- |
| create or replace procedure queryempinfo  (  eno in number,  pname out varchar2,  psal out number,  pjob out varchar2  )  as  begin  select ename,sal,job into pname,psal,pjob from emp where empno = eno;  end; |