目录

[1. Oracle函数方法等 1](#_Toc54961192)

[2. Case 的其他用法 4](#_Toc54961193)

[3. 多列in 和exists 4](#_Toc54961194)

[4. 用其他表中的字段更新本表字段 4](#_Toc54961195)

[5. 删除重复数据列，利用row\_number() over (partition by 分组列 order by 排序列) 5](#_Toc54961196)

[6. 用PIVOT行转列，用UNPIVOT列转行 5](#_Toc54961197)

[7. 树型查询 6](#_Toc54961198)

1. Oracle函数方法等

相对于nvl来说，coalesce支持多个参数，能很方便地返回第一个不为空的值。

SELECT COALESCE(C1,C2，C3，C4,C5，C6) AS c FROM V;

如果上面的语句改用nvl，就要嵌套很多层。

SELECT nvl(nvl(nvl(nvl(nvl(cl，c2)，c3)，c4)，c5)，c6) AS C FROM V;

拼接列：

SELECT 'TRUNCATE TABLE'|| owner ||'.'|| table\_name ||';' AS 清空表 FROM all\_tables

限制返回的行数：

select t.\*, t.rowid from USERINFO t where rownum<2;

随机数与系统时间：

select dbms\_random.value(0,70000), sysdate from dual;

select t.\* from USERINFO t order by 1 ： 1代表第一列

Translate 替换：

简单说来，就是将from中的字符转换为to中与之位置对应的字符，  
        若to中找不到与之对应的字符，返回值中的该字符将会被删除。

去除空格

update PIVOT\_TBL set subject=replace(subject,' ','')

oracle中 ‘’等于null,反过来不成立

按照正则表达式查询与按照正则表达式替换的函数

select t.\*,regexp\_replace(name,'.\*8.\*','发') from USERINFO t where regexp\_like (name,'.\*8.\*')

1，REGEXP\_LIKE ：与LIKE的功能相似

2，REGEXP\_INSTR ：与INSTR的功能相似

3，REGEXP\_SUBSTR ：与SUBSTR的功能相似

4，REGEXP\_REPLACE ：与REPLACE的功能相似

select regexp\_count('a,sd,sdfs,fsdfdf,sd,sd,s',',') from dual

select MOD(65,50) from dual     --取余  
select  trunc( 65/33) from dual  -- 取整 trunc (1.9) = 1 用于截取时间或者数值，返回指定的值。  
select ceil(65/60) from dual          -- 取整 ceil(1.1) = 2

两者等价，当字符串中包含特殊字符可用第二种

select 'asd]' from dual;

select q'[asd']]' from dual;

listagg连接字符串

select t.gradeid,count(1),listagg(name,',') within group(order by name) as aaa from USERINFO t

group by gradeid

在std:

select t.custname,count(1),listagg(cust\_code,',') within group(order by custname) as aaa from CUSTOMERINFO t

group by custname

注意days1相同时对 row\_number() rank() dense\_rank()的影响

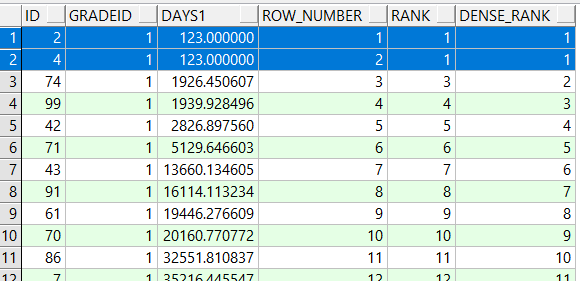
select t.id,t.gradeid,t.days1,

row\_number() over (partition by gradeid order by days1) row\_number,

rank() over (partition by gradeid order by days1) rank,

dense\_rank() over (partition by gradeid order by days1) dense\_rank

from userinfo t



时间函数

select date'2020-01-12'+1 from dual

select add\_months(sysdate,2\*12),*--增加2年*

add\_months(sysdate,1),*--增加1个月*

sysdate+1,*--增加1天*

sysdate+3/24,*--增加3个小时*

sysdate+5/24/60,*--增加5分钟*

sysdate+6/24/60/60 *--增加6秒*

from dual

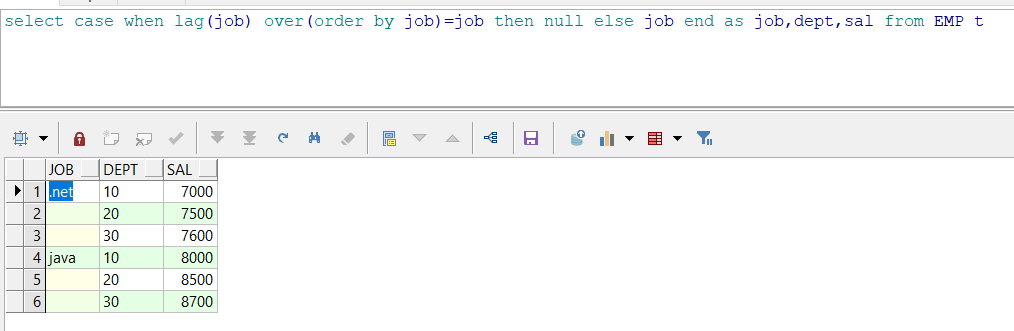
months\_between(,) 和天数相减 计算时间差

lead获得下一行的days0 lag获得上一行

select t.id,t.code,t.name,t.days0,lead(days0) over (order by id), t.rowid from USERINFO t order by id ;

某一列相同合并显示

select case when lag(job) over(order by job)=job then null else job end,dept,sal from EMP t



SELECT hiredate,

to\_number(to\_char(hiredate, 'hh24'））时，

to\_number(to\_char (hiredate, 'mi')）分，

to\_number(to\_char (hiredate, 'ss')）秒，

to\_number(to\_char(hiredate,'dd'）日，

to number(to char (hiredate,'mm')）月，

to\_number(to\_char(hiredate,'yyyy')）年，

to\_number(to\_char(hiredate, 'ddd')）年内第几天，

trunc(hiredate, 'dd')一天之始，

trunc (hiredate,'day'）周初，

trunc (hiredate, 'mm'）月初，

last\_day(hiredate)月未，

add months (trunc(hiredate,'mm'),1）下月初，

trunc(hiredate, 'yy'）年初，

to\_char(hiredate,'day'）周几，

to char(hiredate, 'month'）月份

FROM(SELECT hiredate+ 30/24/60/60+20/24/60 + 5/24 AS hiredate FROM

emp WHERE ROWNUM<=1）;

判断Ｂ表和Ａ表是否满足ON中条件，如果满足则用B表去更新A表，

merge into LOG a

using(select rowid as rid,id,row\_number() over(partition by name order by rowid) as n from log) b

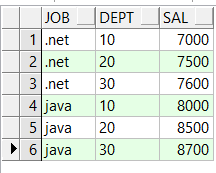
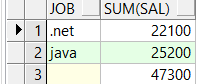
on (a.id=b.id)

when matched then

update set a.name=a.name||b.n

小计功能：

select job,sum(sal) from EMP group by rollup(job);

原始数据： 查询后的数据：

*--decode 相当于if else 或if else if else*

declare a number(4);

b varchar2(8);

begin

a:=3;

select decode(a, 1, '男生',2,'女生','其他') into b from dual;

dbms\_output.put\_line(b);

end;

1. Case 的其他用法

select \* from USERINFO t order by case when id<10 then 1 when id<20 and id>=10 then 4 else 3 end,id

update USERINFO set gender=case when mod(id,2)=1 then 'F' else 'M' end

1. 多列in 和exists

select \* from USERINFO where (gender,gradeid) in (select 'F',id from gradeinfo where rownum<3) order by id;

select \* from USERINFO a where exists (select 1 from (select \* from gradeinfo where rownum<3 ) b where a.gradeid=b.id and a.gender='F' ) order by id;

select t.\*, t.rowid from GRADEINFO t where exists (select 1 from USERINFO where USERINFO.GRADEID=t.id)

1. 用其他表中的字段更新本表字段

方法1：update userinfo set teacherid\_ext=(select teacherid from gradeinfo where userinfo.gradeid=gradeinfo.id and gradeinfo.g\_no in ('三班','二班') )

where exists (select teacherid from gradeinfo where userinfo.gradeid=gradeinfo.id and gradeinfo.g\_no in ('三班','二班'))

如果没有where中的条件则会把其他数据更新为null;

方法2：update (select a.teacherid\_ext,b.teacherid from userinfo a left join gradeinfo b on a.gradeid=b.id where g\_no in ('三班','二班'))

set teacherid\_ext=teacherid

gradeinfo中需要唯一索引或者主键

方发3:建议用此，只访问了一次gradeinfo

merge into userinfo ui

using gradeinfo g on (ui.gradeid=g.id)

when matched then update

set ui.teacherid\_ext=g.teacherid

1. 删除重复数据列，利用row\_number() over (partition by 分组列 order by 排序列)

删除userinfo中name重复的数据并且只保留一条：

delete from USERINFO t where t.rowid in (

select rowid from

(

select rowid,row\_number() over (partition by name order by id) as seq,id from USERINFO

) a where seq>1

)

重复数据列应该是分组列

1. 用PIVOT行转列，用UNPIVOT列转行

select \* from (

select name--该列没在pivot中相当于分组条件

,subject,val from PIVOT\_tbl)

pivot (sum(val) --设和不设别名都可以。设置后使用的是别名和in中的组合，否则只为in中的别名

for subject in ('语文' ,'数学' ,'外语' ,'物理' ,'化学' )

--相当于sum( case when subject=' 语文 ' then val else 0 end ) as ' 语文 '

)

select \* --或者 '星期一','星期二','星期三','星期四','星期五','星期六','星期日'

from WEEK\_INCOME pivot (

sum(income) for week in ('星期一','星期二','星期三','星期四','星期五','星期六','星期日')

)

unpivot是将列名做为列值，因此，会新增两个column：一个column用于存储列名，一个column用于存储列值

select \* from UNPIVOT\_tbl unpivot(val for 科目 in (语文,数学,外语,物理,化学 ))



1. 树型查询

select id,name,code,(prior name),(prior code),sys\_connect\_by\_path(name,' ') 完整路径 from SYS\_REGION

--where r\_level=3

start with id='0001' --为起点向下递归查询

connect by (prior id)=pid --connect by 是递归条件， prior 上一级

order siblings by id *--专有给树排序的方式*