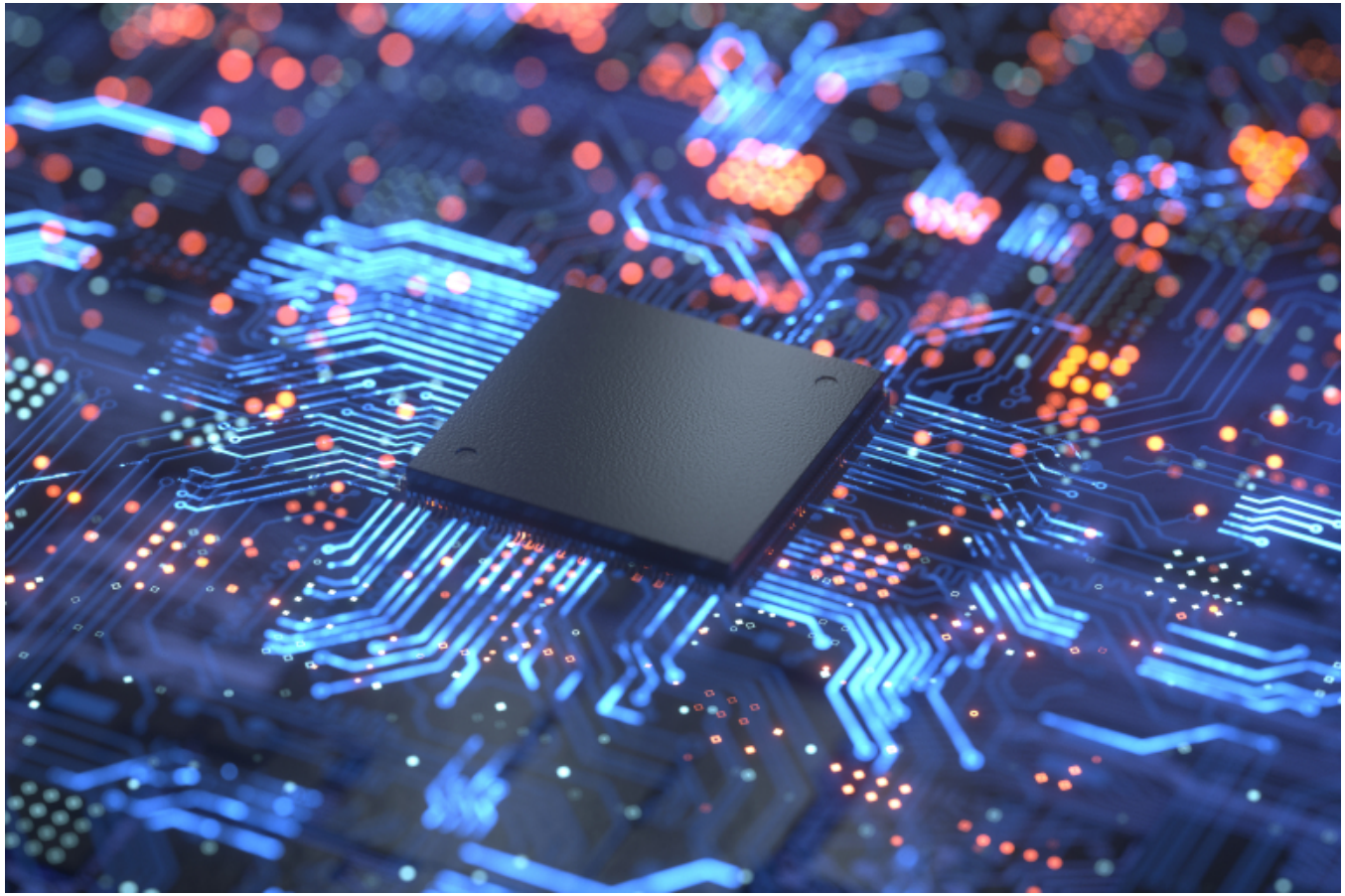
中山大學
SUN YAT-SEN UNIVERSITY



2021级《数据库原理与应用》第12周

2023.5.15

中山大学数学学院 黄志洪

知识点：%type变量类型

中山大學 SUN YAT-SEN UNIVERSITY

Syntax:

```
identifier      Table.column_name%TYPE;
```

Examples:

```
...
  v_name              employees.last_name%TYPE;
  v_balance           NUMBER(7,2);
  v_min_balance       v_balance%TYPE := 10;
...
```

知识点：%rowtype变量类型

中山大學 SUN YAT-SEN UNIVERSITY

Examples:

Declare a variable to store the information about a department from the DEPARTMENTS table.

```
dept_record    departments%ROWTYPE;
```

Declare a variable to store the information about an employee from the EMPLOYEES table.

```
emp_record     employees%ROWTYPE;
```

Record类型

```
declare
type abcdef is record
(c1 char(4),
c2 number(3),
c3 date);
x abcdef;
begin
x.c1:='WXYZ';
x.c2:=123;
x.c3:=sysdate;
dbms_output.enable;
dbms_output.put_line(x.c1);
dbms_output.put_line(x.c2);
dbms_output.put_line(x.c3);
end;
```

测试

```
SQL> set serveroutput on
SQL> declare
  2  type abcdef is record
  3  (c1 char(4),
  4  c2 number(3),
  5  c3 date);
  6  x abcdef;
  7  begin
  8  x.c1:='WXYZ';
  9  x.c2:=123;
 10  x.c3:=sysdate;
 11  dbms_output.enable;
 12  dbms_output.put_line(x.c1);
 13  dbms_output.put_line(x.c2);
 14  dbms_output.put_line(x.c3);
 15  end;
 16  /
WXYZ
123
17-5月 -22
WXYZ
```

更复杂的类型

- Oracle支持create type语句创建更复杂类型（object，array等）
- 参考：
  https://blog.csdn.net/qiuzhi__ke/article/details/104073748

游标（cursor）

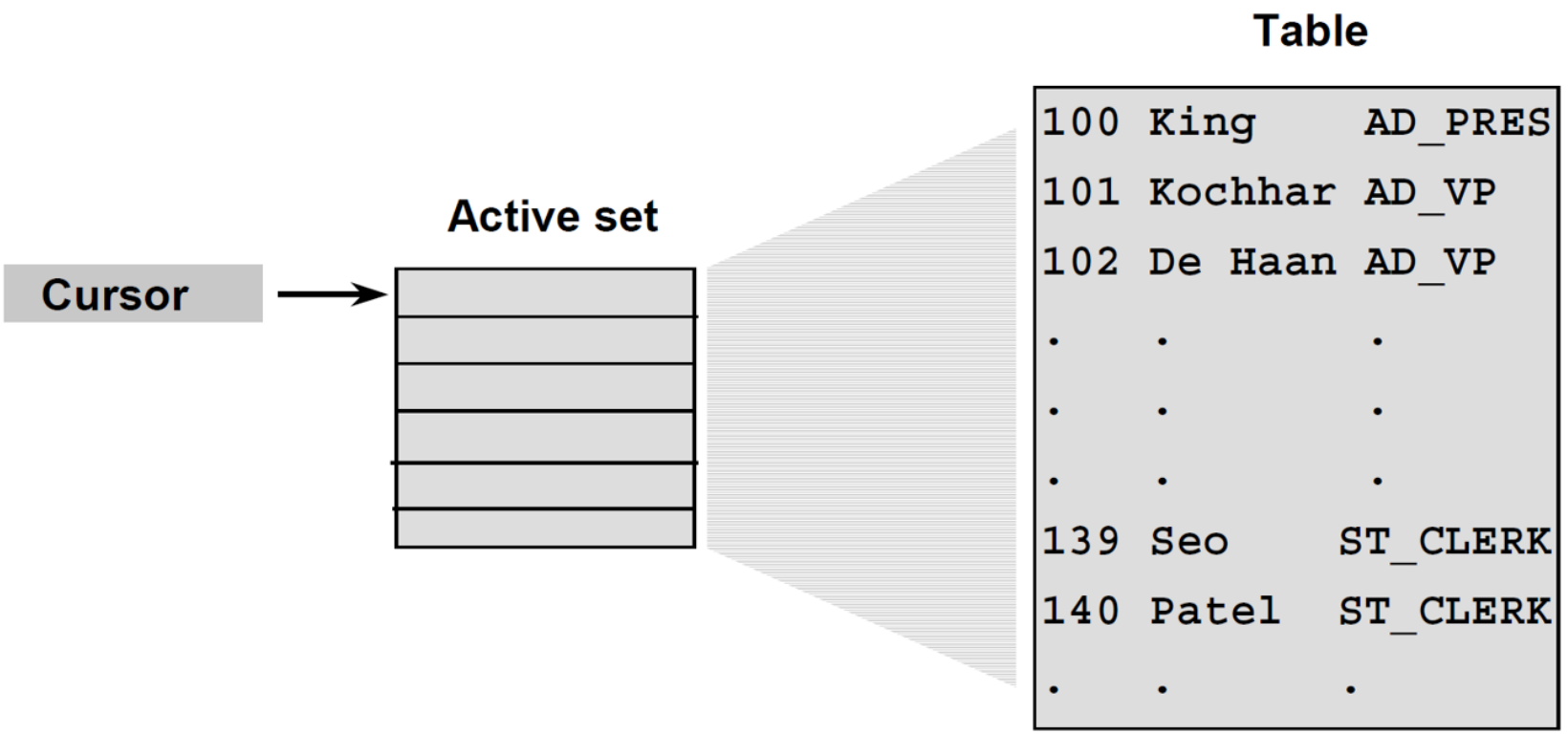
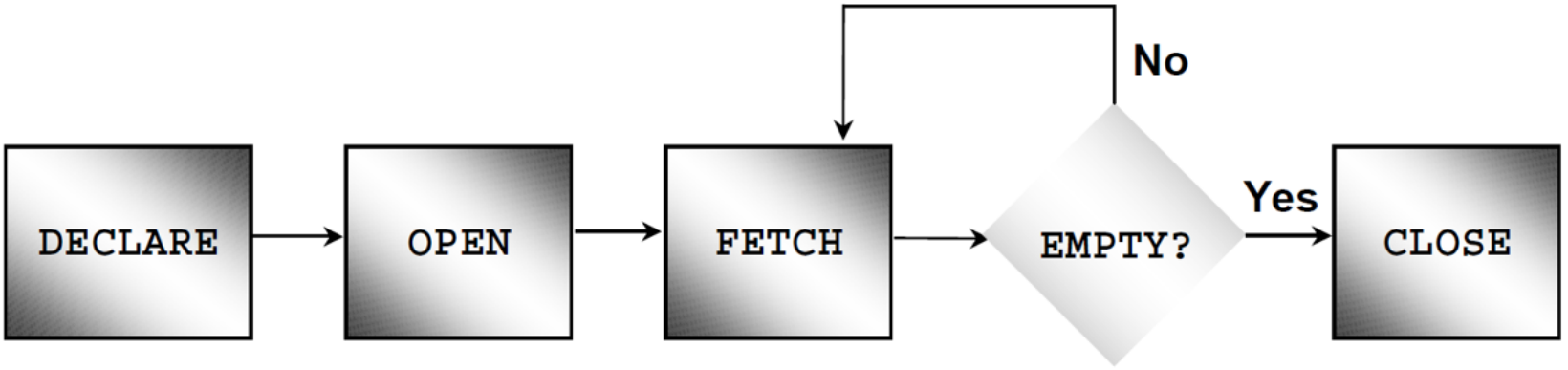


Table
100 King AD_PRES
101 Kochhar AD_VP
102 De Haan AD_VP
. . .
. . .
. . .
139 Seo ST_CLERK
140 Patel ST_CLERK
. . .
Active set
Cursor

游标使用步骤



- Create a named SQL area
- Identify the active set
- Load the current row into variables
- Test for existing rows
- Return to FETCH if rows are found
- Release the active set

样例

- 利用游标实现“查询公司中工资最高的三位员工”

样例

```
declare
cursor emp_cur is select empno,sal from emp order by sal desc;
empno_t emp.empno%type;
sal_t emp.sal%type;
i int:=0;
begin
open emp_cur;
loop
fetch emp_cur into empno_t,sal_t;
i:=i+1;
exit when i>3 or emp_cur%notfound;
dbms_output.put_line(empno_t||' '||sal_t);
end loop;
close emp_cur;
end;
```

结果

```
SQL> set serveroutput on
SQL> declare
  2  cursor emp_cur is select empno,sal from emp order by sal desc;
  3  empno_t emp.empno%type;
  4  sal_t emp.sal%type;
  5  i int:=0;
  6  begin
  7  open emp_cur;
  8  loop
  9  fetch emp_cur into empno_t,sal_t;
 10  i:=i+1;
 11  exit when i>3 or emp_cur%notfound;
 12  dbms_output.put_line(empno_t||' '||sal_t);
 13  end loop;
 14  close emp_cur;
 15  end;
 16
 17  /
7839 5000
4566 3500
4695 3050

PL/SQL 过程已成功完成。
```

查看编译错误

```
SQL> create or replace function zh(n number)
  2  return char as
  3  T varchar(20):='零壹贰叁肆伍陆柒捌玖';
  4  begin
  5  return substk(T,n+1,1);
  6  end;
  7
  8  /

警告: 创建的函数带有编译错误。

SQL> show errors
FUNCTION ZH 出现错误:

LINE/COL ERROR
-------- -----------------------------------------------------------------
5/1      PL/SQL: Statement ignored
5/8      PLS-00201: 必须声明标识符 'SUBSTK'
SQL>
```

例外处理

- 系统预定义例外
- 非预定义的系统例外
- 用户自定义例外

系统预定义例外

```
declare
a emp.ename%type;
begin
select ename into a from emp where sal>4900;
exception
when too_many_rows
then dbms_output.put_line('too much!!!!');
end;
```

测试

```
SQL> set serveroutput on
SQL>
SQL> declare
  2  a emp.ename%type;
  3  begin
  4  select ename into a from emp where sal>4900;
  5  exception
  6  when too_many_rows
  7  then dbms_output.put_line('too much!!!!');
  8  end;
  9  /

PL/SQL 过程已成功完成。
```

测试

```
SQL> update emp set sal=6000 where ename='JONES';

已更新 1 行。

已用时间:  00: 00: 00.00
SQL>
SQL>
SQL> declare
  2  a emp.ename%type;
  3  begin
  4  select ename into a from emp where sal>4900;
  5  exception
  6  when too_many_rows
  7  then dbms_output.put_line('too much!!!!');
  8  end;
  9  /
too much!!!!

PL/SQL 过程已成功完成。

已用时间:  00: 00: 00.00
SQL> rollback;

回退已完成。
```

非预定义的系统例外

```
declare
abc exception;
pragma exception_init(abc,-0001);
begin
insert into emp (empno) values (7900);
exception
when abc
then dbms_output.put_line('error');
end;
```

测试

```
SQL> declare
  2  abc exception;
  3  pragma exception_init(abc,-0001);
  4  begin
  5  insert into emp (empno) values (7900);
  6  exception
  7  when abc
  8  then dbms_output.put_line('error');
  9  end;
 10  /
error

PL/SQL 过程已成功完成。
```

用户自定义例外

```
declare
n number;
xyz exception;
begin
select count(*) into n from emp where sal>1500;
if n>10 then raise xyz;
 else dbms_output.put_line('under 10');
end if;
exception
when xyz then
dbms_output.put_line('>10');
end;
```

测试

```
declare
n number;
xyz exception;
begin
select count(*) into n from emp where sal>1500;
if n>10 then raise xyz;
 else dbms_output.put_line('under 10');
end if;
exception
when xyz then
dbms_output.put_line('>10');
end;
```

测试

```
SQL> declare
  2  n number;
  3  xyz exception;
  4  begin
  5  select count(*) into n from emp where sal>1500;
  6  if n>10 then raise xyz;
  7   else dbms_output.put_line('under 10');
  8  end if;
  9  exception
 10  when xyz then
 11  dbms_output.put_line('>10');
 12  end;
 13  /
>10

PL/SQL 过程已成功完成。
```

触发器

- 特殊的存储过程。当满足一定条件时会被触发运行（fire）
- 常见触发条件有before，after（某种DML语句），深入的还有各种系统事件的触发器，例如登录，登出数据库，启动或关闭数据库（系统级触发器）
- 表级触发器与行级触发器
- 问题：当一个表绑定了很多触发器时，它们的点火次序?

样例

- 写一个触发器，使emp表只有在周一到周五8:00-18:00这个时间段才可以被修改

代码

```
create or replace trigger s_emp
 before insert on emp
 begin
 if (to_char(sysdate,'DY') in ('星期六','星期日')
  or (to_char(sysdate,'HH24:MI') not between
  '08:00' and '18:00'))
 then raise_application_error(-20500,'有人入侵');
 end if;
 end;
```

测试

```
SQL> connect scott/tiger
已连接。
SQL> create or replace trigger s_emp
  2   before insert on emp
  3   begin
  4   if (to_char(sysdate,'DY') in ('星期六','星期日')
  5    or (to_char(sysdate,'HH24:MI') not between
  6    '08:00' and '18:00'))
  7   then raise_application_error(-20500,'有人入侵');
  8   end if;
  9   end;
 10  /

触发器已创建
```

测试

```
SQL> select sysdate from dual;

SYSDATE
-------------
24-5月 -22

SQL> insert into emp (empno) values (5555);
insert into emp (empno) values (5555)
            *
第 1 行出现错误:
ORA-20500: 有人入侵
ORA-06512: 在 "SCOTT.S_EMP", line 5
ORA-04088: 触发器 'SCOTT.S_EMP' 执行过程中出错


SQL>
```

触发器用于审计

```
create table rec (name varchar2(40),time date);


create or replace trigger rec_update
after update on emp
begin
insert into rec values(user,sysdate);
end;
```

测试

```
SQL> create table rec (name varchar2(40),time date);

表已创建。

SQL> create or replace trigger rec_update
  2  after update on emp
  3  begin
  4  insert into rec values(user,sysdate);
  5  end;
  6  /

触发器已创建
```

测试

```
SQL> select * from rec;

未选定行

SQL> update emp set sal=9000 where ename='KING';

已更新 1 行。

SQL> select * from rec;

NAME                                               TIME
-------------------------------------------------- ---------------
SCOTT                                              24-5月 -22

SQL>
```

行级触发器

```
create or replace  trigger r_sal
before insert or update of sal on emp for each row
begin
if not (:new.ename='KING') and :new.sal>=5000
then raise_application_error(-20202,'非法工资');
end if;
end;
```

测试

```
SQL>
SQL> create or replace  trigger r_sal
  2  before insert or update of sal on emp for each row
  3  begin
  4  if not (:new.ename='KING') and :new.sal>=5000
  5  then raise_application_error(-20202,'非法工资');
  6  end if;
  7  end;
  8  /

触发器已创建
```

测试

```
SQL> rollback;

回退已完成。

SQL> update emp set sal=9000 where ename='KING';

已更新 1 行。

SQL> update emp set sal=9000 where ename='SCOTT';

已更新0行。

SQL> update emp set sal=9000 where ename='JONES';
update emp set sal=9000 where ename='JONES'
       *
第 1 行出现错误:
ORA-20202: 非法工资
ORA-06512: 在 "SCOTT.R_SAL", line 3
ORA-04088: 触发器 'SCOTT.R_SAL' 执行过程中出错
```

点火优先的测试

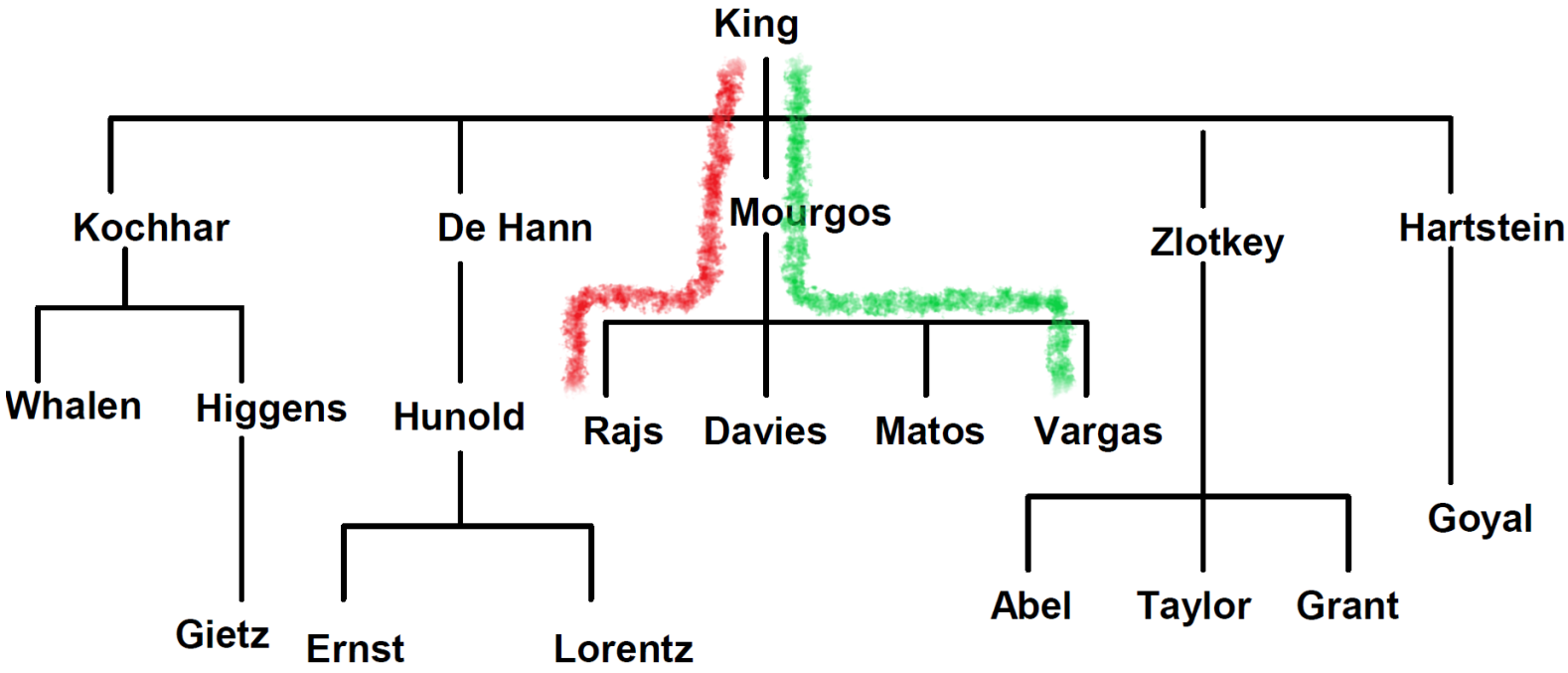
```
SQL> rollback;

回退已完成。

SQL> insert into emp (empno,ename,sal) values (6666,'Ai',8000);
insert into emp (empno,ename,sal) values (6666,'Ai',8000)
            *
第 1 行出现错误:
ORA-20500: 有人入侵
ORA-06512: 在 "SCOTT.S_EMP", line 5
ORA-04088: 触发器 'SCOTT.S_EMP' 执行过程中出错
```

禁止和删除触发器

- 禁用触发器 alter trigger tri_uname(触发器名字) disable;
- 激活触发器 alter trigger tri_uname(触发器名字) enable;
- 重新编译 alter trigger tri_uname(触发器名字) complie;
- 禁用某个表上的触发器 alter table table_name(表名) diable all triggers;
- 删除触发器 DROP TRIGGER tri_uname(触发器名字);

作业答案

- 假设每个人都只能和直接上司或直接下属交流，求任意两人间交流信息需要经过的最小中间节点数



作业答案

- 假设每个人都只能和直接上司或直接下属交流，求任意两人间交流信息需要经过的最小中间节点数。
- 能否用宽表展示结果？ （略）

求指定两个人之间的传递路径节点数

- 求SMITH与MARTIN之间的传递节点数

```
with
aaa as
(select empno from emp
start with ename='SMITH'
connect by prior mgr=empno),
bbb as
(select empno from emp
start with ename='MARTIN'
connect by prior mgr=empno)
select count(*) from
(
((select * from aaa) union (select * from bbb))
minus
((select * from aaa) intersect (select * from bbb))
);
```

测试

```
SQL> with
  2  aaa as
  3  (select empno from emp
  4  start with ename='SMITH'
  5  connect by prior mgr=empno),
  6  bbb as
  7  (select empno from emp
  8  start with ename='MARTIN'
  9  connect by prior mgr=empno)
 10  select count(*) from
 11  (
 12  ((select * from aaa) union (select * from bbb))
 13  minus
 14  ((select * from aaa) intersect (select * from bbb))
 15  );

  COUNT(*)
----------
         5
```

不用集合运算的解法

```
with
aaa as
(select empno from emp
start with ename='SMITH'
connect by prior mgr=empno),
bbb as
(select empno from emp
start with ename='MARTIN'
connect by prior mgr=empno)
select
(select count(*) from aaa)+(select count(*) from bbb)-2*(select count(*) from aaa,bbb where
aaa.empno=bbb.empno)
from dual;
```

测试

```
SQL> with
  2  aaa as
  3  (select empno from emp
  4  start with ename='SMITH'
  5  connect by prior mgr=empno),
  6  bbb as
  7  (select empno from emp
  8  start with ename='MARTIN'
  9  connect by prior mgr=empno)
 10  select
 11  (select count(*) from aaa)+(select count(*) from bbb)-2*(select count(*) from aaa,bbb where aaa.empno=bbb.em
pno)
 12  from dual;

(SELECTCOUNT(*)FROMAAA)+(SELECTCOUNT(*)FROMBBB)-2*(SELECTCOUNT(*)FROMAAA,BBBWHEREAAA.EMPNO=BBB.EMPNO)
-----------------------------------------------------------------------------------------------------
                                                                                                    5
```

任意两人的路径节点数计算

```
select x.ename,y.ename,
(select count(*) from (select empno from emp
start with ename=x.ename
connect by prior mgr=empno))
+
(select count(*) from (select empno from emp
start with ename=y.ename
connect by prior mgr=empno))
-
2*(select count(*) from
(select empno from emp
start with ename=x.ename
connect by prior mgr=empno) aaa,
(select empno from emp
start with ename=y.ename
connect by prior mgr=empno) bbb
where aaa.empno=bbb.empno)
from emp x,emp y;
```

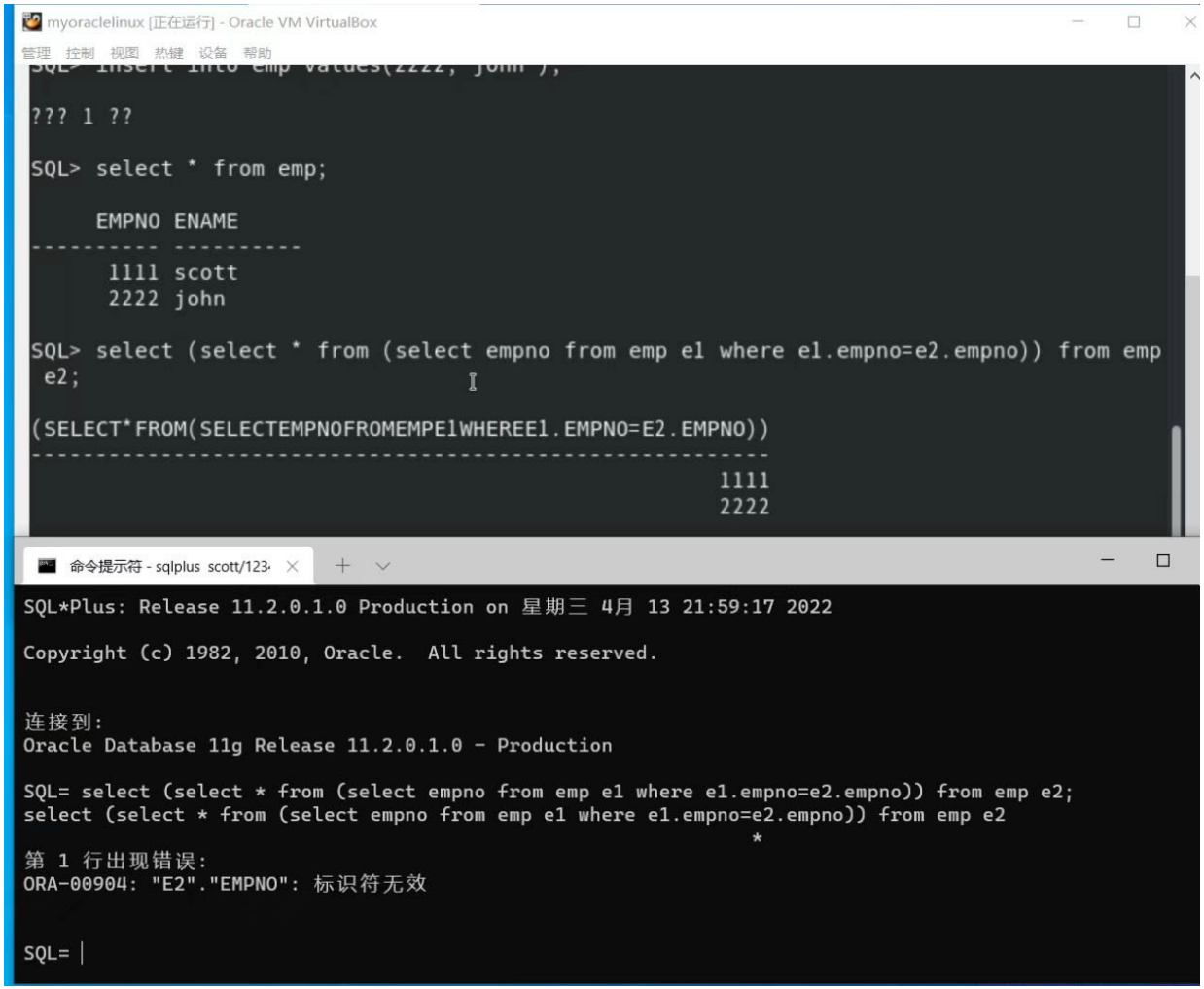
诡异错误

```
SQL> select x.ename,y.ename,
  2  (select count(*) from (select empno from emp
  3  start with ename=x.ename
  4  connect by prior mgr=empno))
  5  +
  6  (select count(*) from (select empno from emp
  7  start with ename=y.ename
  8  connect by prior mgr=empno))
  9  -
 10  2*(select count(*) from
 11  (select empno from emp
 12  start with ename=x.ename
 13  connect by prior mgr=empno) aaa,
 14  (select empno from emp
 15  start with ename=y.ename
 16  connect by prior mgr=empno) bbb
 17  where aaa.empno=bbb.empno)
 18  from emp x,emp y;
start with ename=x.ename
                  *
第 3 行出现错误:
ORA-00904: "X"."ENAME": 标识符无效
```

测试

```
SQL> select count(*) from
  2  (select empno from emp
  3  start with ename='SMITH'
  4  connect by prior mgr=empno) aaa,
  5  (select empno from emp
  6  start with ename='MARTIN'
  7  connect by prior mgr=empno) bbb
  8  where aaa.empno=bbb.empno;

  COUNT(*)
----------
         1
```

ORA-00904是Oracle的BUG



2023.5.15

ORA-00904是Oracle的BUG

- https://asktom.oracle.com/pls/apex/f?p=100:11:0::::P11_QUESTION_ID:9531834700346297391

> Hi Team,
>
> I have a correlated query, trying to execute it on oracle 11g. The same query executes perfectly fine on 12c but it gives ORA-00904 Invalid Identifier u.col_1 on 11g.
>
> Can someone please explain what's causing the issue in 11g?
>
> ```
> SELECT *
> FROM table_1 u
> WHERE u.col_1 LIKE '%XYZ%'
> AND EXISTS (SELECT 1
> FROM table_2 c,
> (SELECT s.col_1 FROM table_2 s WHERE s.col_1 = u.col_1) dd
> WHERE c.col_1 = u.col_1
> AND c.col_2 = dd.col_2);
> ```
>
> Thanks!!

ORA-00904是Oracle的BUG

Chris said...

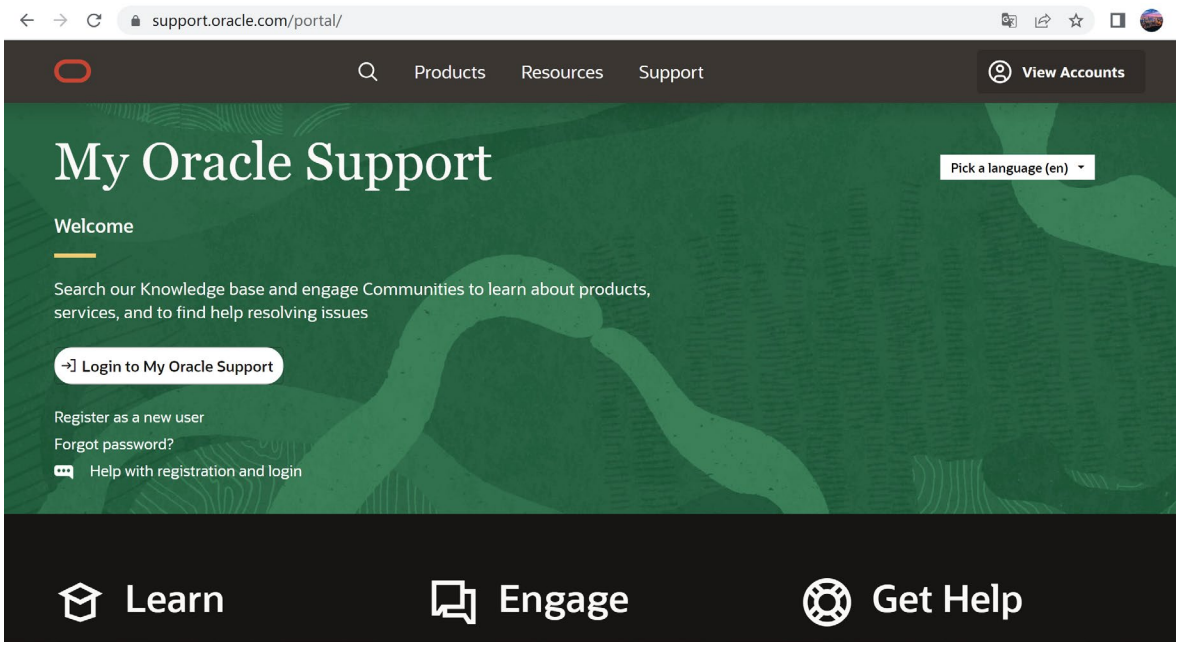
You've gone **"too deep"** with your join to table_2. **In 11g you can only reference tables in the query immediately inside/outside your current subquery. 12c lifted this restriction.**

关于Tom Kyte与Asktom网站

- Tom Kyte (thomas.kyte@oracle.com) 从1993年起一直在Oracle工作。Kyte是负责Oracle 治理、教育和保健集团的副总裁,也是"Effective Oracle by Design"(Oracle 出版社出版)和"EXPert One-on-One: Oracle"(Apress出版)两书的作者。
- Thomas Kyte号称“知晓Oracle的一切”。从Oracle 7.0.9版本开始就一直任职于Oracle公司，不过，其实他从5.1.5c版本就开始使用Oracle了。 在Oracle公司，Kyte的任务是帮助使用Oracle数据库的客户，并与他们共同设计和构建系统，或者对系统进行重构和调优。Thomas Kyte就是主持**Oracle Magazine Ask Tom专栏和Oracle公司同名在线论坛**的那个Tom，他通过这一方式热心地回答困扰着Oracle开发人员和DBA的各种问题。

可以光顾的英文网站

- AskTom
- Stackoverflow
- Leetcode（刷题）
- Metalink.oracle.com（现在叫Oracle support）



解法：自上而下的计算

- 19级夏威龙同学提供的思路

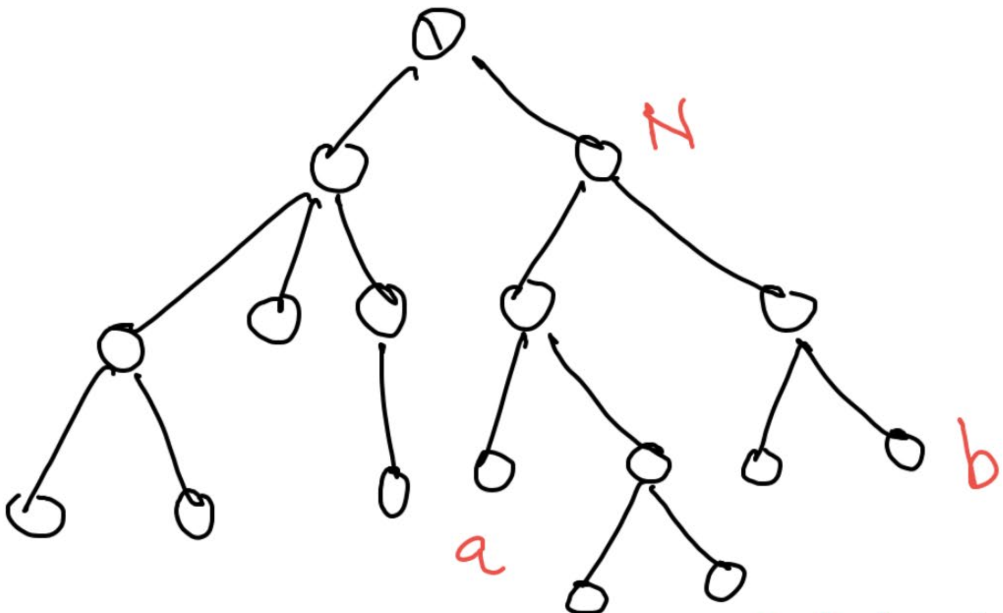
解法：自上而下的计算

```
SELECT a.ename as employee1, b.ename  as employee2,
(SELECT level FROM emp
WHERE empno = a.empno
START WITH ename = 'KING'
CONNECT BY PRIOR empno = mgr AND PRIOR empno != a.empno)
--数字1，意义：a到根上司KING的层数
+(SELECT level FROM emp
WHERE empno = b.empno
START WITH ename = 'KING'
CONNECT BY PRIOR empno = mgr AND PRIOR empno ! =b.empno)
--数字2，意义：b到根上司KING的层数
```

解法：接上页

```
-(SELECT MAX(level) FROM emp N
WHERE a.empno IN(
SELECT empno FROM emp START WITH empno = N.empno
CONNECT BY PRIOR empno = mgr)
AND b.empno IN(
SELECT empno FROM emp START WITH empno = N.empno
CONNECT BY PRIOR empno = mgr)
START WITH ename = 'KING'
CONNECT BY PRIOR empno = mgr)*2
--数字3，减去两倍的：根上司KING到两人公共上司距离的最大值
as le from emp a,emp b;
```

思路

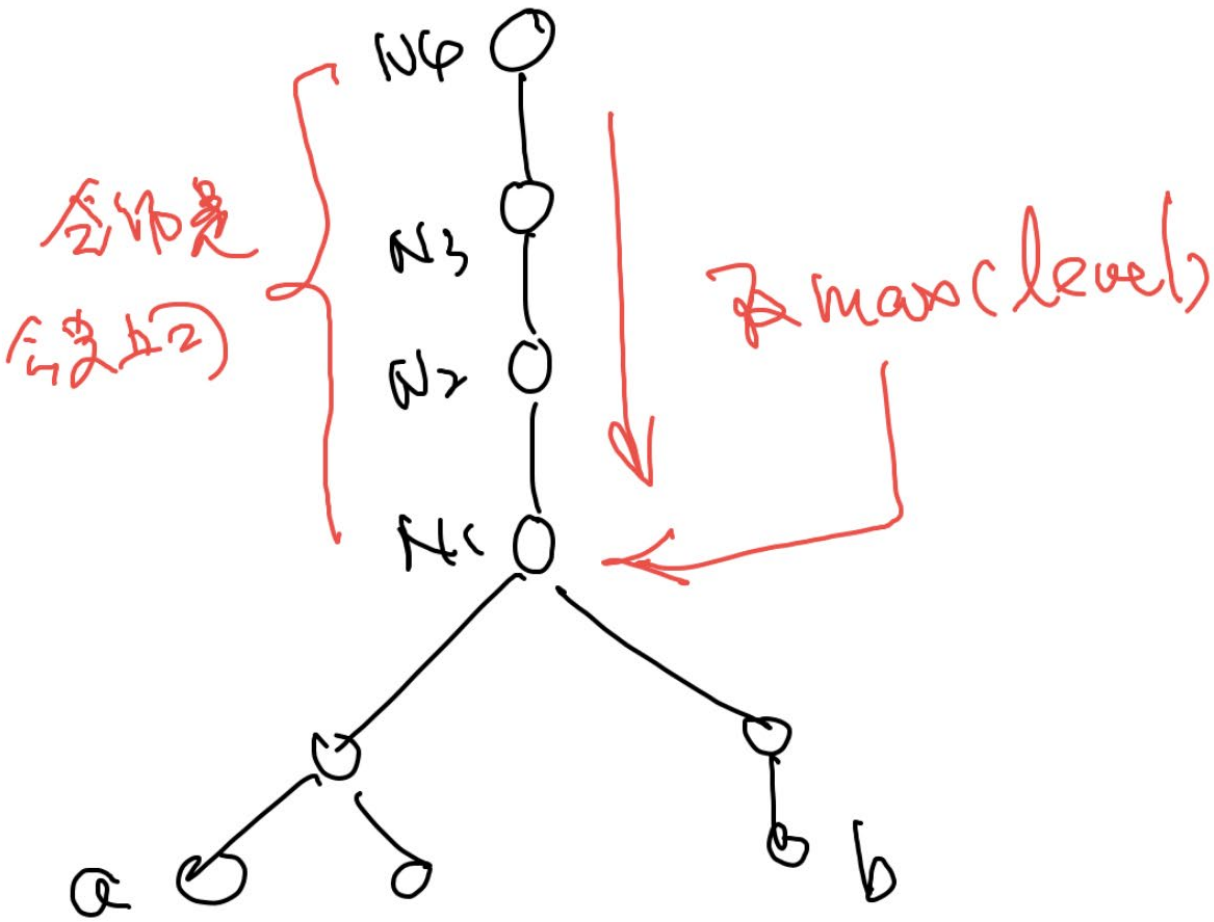


1) a 的祖先必在N开始的子树. N是a上司

2) b 的祖先必在N ——— b ——

3) 由1). 2). N是a. b公共上司.

思路



测试

FORD	FORD	0
FORD	MILLER	4
MILLER	SMITH	5
MILLER	ALLEN	4
MILLER	WARD	4
MILLER	JONES	3
MILLER	MARTIN	4
MILLER	BLAKE	3
MILLER	CLARK	1
MILLER	KING	2
MILLER	TURNER	4
MILLER	JAMES	4
MILLER	FORD	4
MILLER	MILLER	0

已选择144行。

更简洁解法

SELECT e1.ename,e2.ename,

(SELECT MIN(SUM(level)-2) FROM emp

START WITH ename in (e1.ename,e2.ename) CONNECT by PRIOR mgr=empno GROUP by ename HAVING count(*)=2) L

FROM emp e1,emp e2;

看点：自下而上，但又不会出现不会在11g环境下Ora-00904错误的解法。列表括号<>子查询括号?

19级秦威同学提供的答案

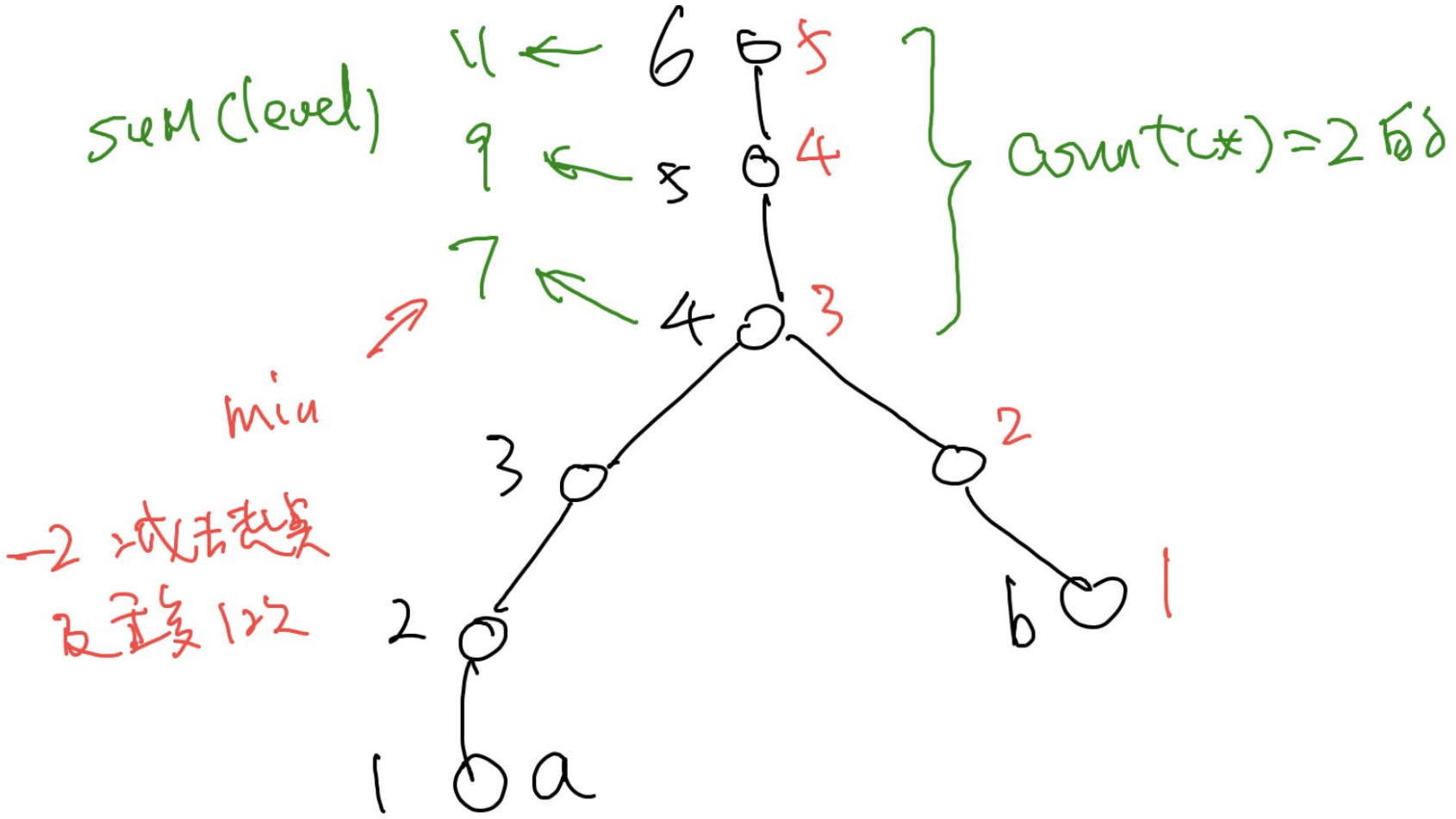
思路解释

```
SQL>
SQL> SELECT level,ename FROM emp
  2  START WITH ename in ('SMITH','MARTIN') CONNECT by PRIOR mgr=empno  ;

     LEVEL ENAME
---------- ----------
         1 SMITH
         2 FORD
         3 JONES
         4 KING
         1 MARTIN
         2 BLAKE
         3 KING

已选择7行。
```

解释

```
SQL> SELECT ename,(SUM(level)-2) , count(*) FROM emp
  2  START WITH ename in ('SMITH','MARTIN') CONNECT by PRIOR mgr=empno GROUP by ename  ;

ENAME      (SUM(LEVEL)-2)   COUNT(*)
---------- -------------- ----------
JONES                   1          1
FORD                    0          1
SMITH                  -1          1
MARTIN                 -1          1
KING                    5          2
BLAKE                   0          1

已选择6行。

SQL> SELECT ename,(SUM(level)-2) , count(*) FROM emp
  2  START WITH ename in ('SMITH','MARTIN') CONNECT by PRIOR mgr=empno GROUP by ename  having count(*)=2;

ENAME      (SUM(LEVEL)-2)   COUNT(*)
---------- -------------- ----------
KING                    5          2
```

思路



运行结果

ENAME	ENAME	L
SMITH	SMITH	
SMITH	ALLEN	5
SMITH	WARD	5
SMITH	JONES	2
SMITH	MARTIN	5
SMITH	BLAKE	4
SMITH	CLARK	4
SMITH	KING	3
SMITH	TURNER	5
SMITH	JAMES	5
SMITH	FORD	1

用这个方法解决上周习题

```
with xxx as
(SELECT e1.ename n1,e2.ename n2,
(SELECT MIN(SUM(level)-2) FROM emp
START WITH ename in (e1.ename,e2.ename) CONNECT by PRIOR mgr=empno
GROUP by ename HAVING count(*)=2) L
FROM emp e1,emp e2)
select n2 from xxx
where n1='MARTIN'
and L=(select max(L) from xxx where n1='MARTIN');
```

解释：with定义的CTE内容

N1	N2	L
SMITH	MILLER	5
ALLEN	SMITH	5
ALLEN	ALLEN	
ALLEN	WARD	2
ALLEN	JONES	3
ALLEN	MARTIN	2
ALLEN	BLAKE	1
ALLEN	CLARK	3
ALLEN	KING	2
ALLEN	TURNER	2
ALLEN	JAMES	2

运行结果

```
SQL> with xxx as
  2  (SELECT e1.ename n1,e2.ename n2,
  3  (SELECT MIN(SUM(level)-2) FROM emp
  4  START WITH ename in (e1.ename,e2.ename) CONNECT by PRIOR mgr=empno GROUP by ename HAVING count(*)=2) L
  5  FROM emp e1,emp e2)
  6  select n2 from xxx
  7  where n1='MARTIN'
  8  and L=(select max(L) from xxx where n1='MARTIN');

N2
----------
SMITH
```

另一种答案

- 由19级张景轩同学提供
- 看点：使用了NOCYCLE关键词

另一种答案

```
WITH TMP AS ( --以所有人为起点，生成一个关系树
                    SELECT LEVEL - 2        AS LV
                                , ENAME
                                , EMPNO
                                , MGR
                                , CEIL(ROWNUM / 14) AS GN --每14人来自同一个起点，所以按14人进行分组
                    FROM EMP
                    CONNECT BY NOCYCLE PRIOR MGR = EMPNO
                        OR PRIOR EMPNO = MGR --既可向上查询也可向下查询，但是不重复查询
        )
SELECT T1.ENAME AS NAME1
                , T2.ENAME AS NAME2
                , T1.LV    AS MID_PEOPLE
FROM TMP            T1
                    INNER JOIN TMP T2 ON
                                            T1.GN = T2.GN
                                AND T2.LV = -1 --同一组的取中间人个数为-1的人作为起点
;
```

代码解释

- NOCYCLE的作用是保证不会生成环路，一定是树状结构，相当于对于每个人生成了一棵“关系树”

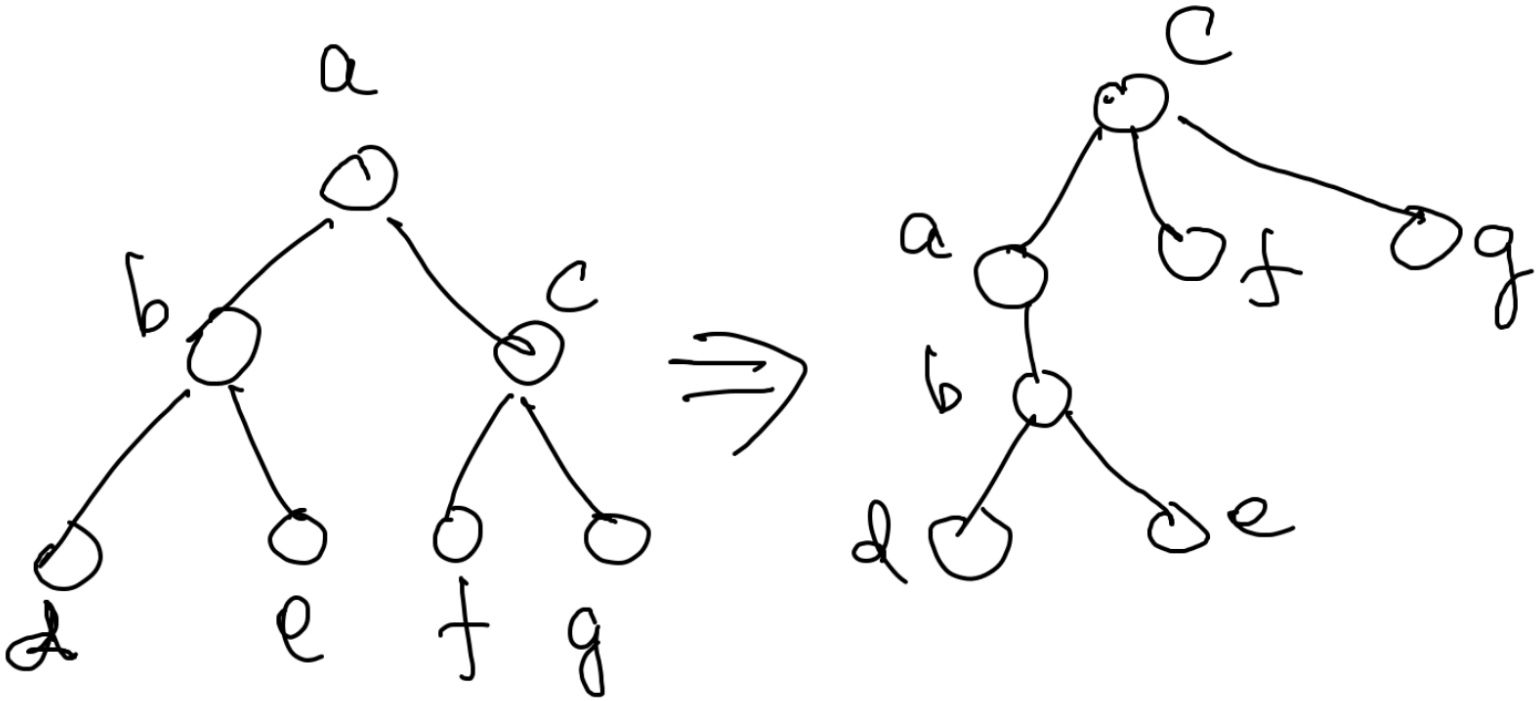
```
WITH TMP AS ( --以所有人为起点，生成一个关系树
                    SELECT LEVEL - 2        AS LV
                                , ENAME
                                , EMPNO
                                , MGR
                                , CEIL(ROWNUM / 14) AS GN --每14人来自同一个起点，所以按14人进行分组
                    FROM EMP
                    CONNECT BY NOCYCLE PRIOR MGR = EMPNO
                          OR PRIOR EMPNO = MGR --既可向上查询也可向下查询，但是不重复查询
        )
SELECT T1.ENAME AS NAME1
            , T2.ENAME AS NAME2
            , T1.LV    AS MID_PEOPLE
FROM TMP              T1
                INNER JOIN TMP T2 ON
                                    T1.GN = T2.GN
                    AND T2.LV = -1 --同一组的取中间人个数为-1的人作为起点
```

With定义的CTE内容

- 想象在KING为起点的“下属关系树”中，捏住每个员工抖一下，形成一棵新的“直属关系树”

LV	ENAME	EMPNO	MGR	GN
-1	SMITH	7369	7902	1
0	FORD	7902	7566	1
1	JONES	7566	7839	1
2	KING	7839		1
3	BLAKE	7698	7839	1
4	ALLEN	7499	7698	1
4	WARD	7521	7698	1
4	MARTIN	7654	7698	1
4	TURNER	7844	7698	1
4	JAMES	7900	7698	1
3	CLARK	7782	7839	1
4	MILLER	7934	7782	1
-1	ALLEN	7499	7698	1
0	BLAKE	7698	7839	1
1	WARD	7521	7698	2
1	MARTIN	7654	7698	2
1	KING	7839		2
2	JONES	7566	7839	2
3	FORD	7902	7566	2

思路



代码解释：怎样得到最终结果输出

```
SELECT T1.ENAME AS NAME1
              , T2.ENAME AS NAME2
              , T1.LV    AS MID_PEOPLE
FROM TMP              T1
                  INNER JOIN TMP T2 ON
                                    T1.GN = T2.GN
                               AND T2.LV = -1 --同一组的取中间人个数为-1的人作为起点
```

简化版本

```
WITH T AS
(SELECT FLOOR((ROWNUM-1)/14)+1 R,LEVEL-1 DISTANCE,EMPNO FROM
EMP
START WITH EMPNO IN (SELECT EMPNO FROM EMP)
CONNECT BY NOCYCLE PRIOR EMPNO=MGR OR PRIOR MGR=EMPNO)
SELECT A.EMPNO,B.EMPNO,B.DISTANCE FROM T A,T B
WHERE A.R=B.R AND A.DISTANCE=0;
```

中山大學
SUN YAT-SEN UNIVERSITY

Thanks

FAQ时间