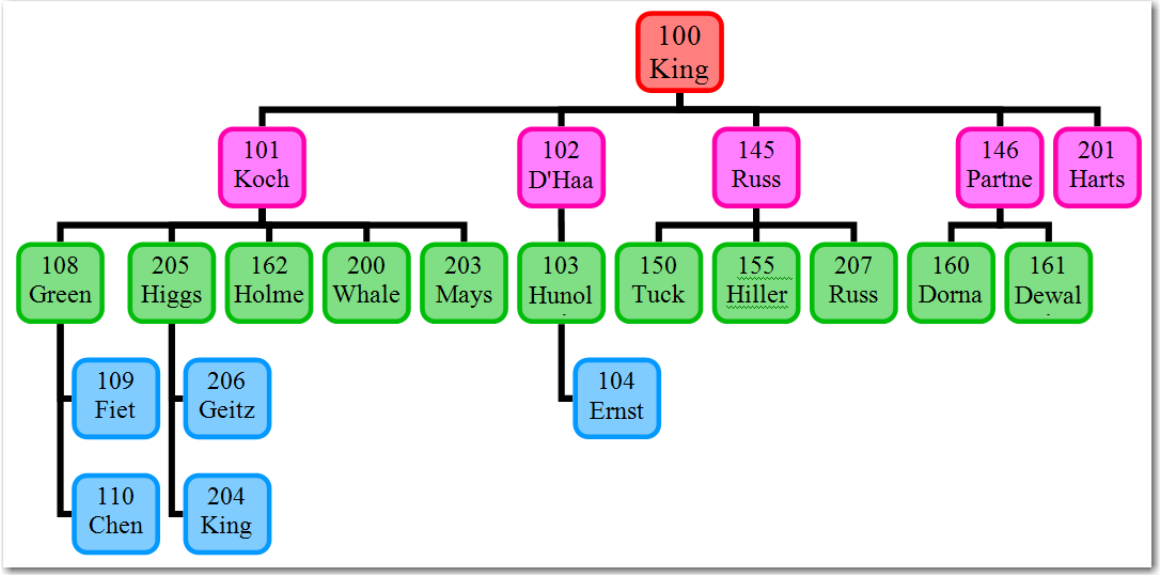
Table of Contents

This is one of those techniques that looks interesting, has a fairly complication syntax, and it quite powerful if you have data that that this type of a hierarchical organization. Remember this is optional.

1. Hierarchical Data

We often use hierarchical organization when think about or organizing data. The hierarchy chart here makes it pretty obvious which employee reports to whom. This is commonly called a tree structure and we use terms such as node, leaf, branch, and root; we also refer to this organization using the terms parent and child. The basic rules for a tree structure are that there is one root node and that each node other than the root has one parent. There are no circular branches.

This is a tree showing our employees and their managers. Employee 101 reports to employee 100 and employee 205 reports to employee 101 and employees 206 and 204 report to employee 205.



1.1. Flattening hierarchal data for a table

When we store this data in a relational table we have to flatten it; we do this by placing the parent id into the child row. The root row commonly has its parent left as a null. This is the view listing. The view definition is in the demo.

```
select  emp_id, name_last, mng, salary
from hier_emp;
```

```
    EMP_ID NAME_LAST                      MNG
---------- ------------------------ ----------
       100 King
       201 Harts                           100
       101 Koch                            100
       108 Green                           101
       205 Higgs                           101
```

```
       102 D'Haa                          100
       103 Hunol                          102
       104 Ernst                          103
       145 Russ                           100
       150 Tuck                           145
       155 Hiller                         145
       162 Holme                          101
       200 Whale                          101
       207 Russ                           145
       203 Mays                           101
       146 Partne                         100
       109 Fiet                           108
       160 Dorna                          146
       161 Dewal                          146
       110 Chen                           108
       206 Geitz                          205
       204 King                           205
```

There are a number of functions and expressions which Oracle supplies for working with this type of data.

2. Start With & Connect By Prior

For the basic query to follow the tree you can use the Start With clause to specify at what point in the tree you want to start. And you add the Connect by Prior to show the relationships- here we want to say that the prior element of the employee id is the manager id—i.e. that we travel up the tree from an employee to their manager.

Demo 01: Using start with and connect by prior

```
select emp_id, name_last, mng
from hier_emp
start with  mng is null
connect by  prior emp_id = mng
order by mng nulls first, emp_id
;
```

This just shows us the same table display as before.

Demo 02: Now add a level pseudo column

```
select level, emp_id, name_last, mng
from hier_emp
start with  mng is null
connect by  prior emp_id = mng
order by  mng nulls first, emp_id
;
```

```
     LEVEL     EMP_ID NAME_LAST                     MNG
---------- ---------- ------------------------- ----------
         1        100 King
         2        101 Koch                           100
         2        102 D'Haa                          100
         2        145 Russ                           100
         2        146 Partne                         100
         2        201 Harts                          100
         3        108 Green                          101
         3        162 Holme                          101
         3        200 Whale                          101
         3        203 Mays                           101
         3        205 Higgs                          101
```

```
         3        103 Hunol                          102
         4        104 Ernst                          103
         4        109 Fiet                           108
         4        110 Chen                           108
         3        150 Tuck                           145
         3        155 Hiller                         145
         3        207 Russ                           145
         3        160 Dorna                          146
         3        161 Dewal                          146
         4        204 King                           205
         4        206 Geitz                          205
```

Although this does not seem very interesting, the level column does tell us how far down the tree an employee is. Employee 103 Huno is on the third level while employee 206 Geitz is at the fourth level. That is new information.

Demo 03: Change the Start with column. Now our tree starts with manager 101 and employees who report to him directly or indirectly

```
select level, emp_id, name_last, mng
from hier_emp
START WITH  mng = 101
connect by  prior emp_id = mng
order by  mng nulls first, emp_id;
```

```
     LEVEL     EMP_ID NAME_LAST         MNG
---------- ---------- ---------- ----------
         1        108 Green              101
         1        162 Holme              101
         1        200 Whale              101
         1        203 Mays               101
         1        205 Higgs              101
         2        109 Fiet               108
         2        110 Chen               108
         2        204 King               205
         2        206 Gietz              205
```

Demo 04: If we start further down the tree we get fewer rows returned.

```
select level, emp_id, name_last, mng
from hier_emp
START WITH mng = 205
connect by prior emp_id = mng
order by mng nulls first, emp_id;
```

```
     LEVEL     EMP_ID NAME_LAST         MNG
---------- ---------- ---------- ----------
         1        204 King               205
         1        206 Gietz              205
```

Demo 05: How many levels do we have?

```
select count(distinct level)
from hier_emp
start with mng  is null
connect by prior emp_id = mng
;
```

```
COUNT(DISTINCTLEVEL)
--------------------
                   4
```

Demo 06: How many employees at each level?

```
select level
, count(emp_id)
from hier_emp
start with mng is null
connect by prior emp_id = mng
group by level
order by level
;
```

```
     LEVEL COUNT(EMP_ID)
---------- -------------
         1             1
         2             5
         4             5
         3            11
```

Demo 07: Variations- here we have two trees . One start with emp 101 and the other starts with emp 146

```
select level, emp_id, name_last, mng
from hier_emp
start with  mng in (101, 146)
connect by  prior emp_id = mng
;
```

```
     LEVEL     EMP_ID NAME_LAST          MNG
---------- ---------- ---------- ----------
         1        108 Green             101
         2        109 Fiet              108
         2        110 Chen              108
         1        162 Holme             101
         1        200 Whale             101
         1        203 Mays              101
         1        205 Higgs             101
         2        204 King              205
         2        206 Gietz             205
         1        160 Dorna             146
         1        161 Dewal             146
```

Demo 08: Is employee 204 on the branch that starts with employee 101?

```
Select name_last, emp_id
from hier_emp
where emp_id = 204
start with emp_id = 101
connect by mng = prior emp_id;
```

```
NAME_LAST                 EMP_ID
------------------------- -----------
King                              204
```

Demo 09: Is employee 207 on the branch that starts with employee 101?

```
Select name_last, emp_id
from hier_emp
where emp_id = 207
start with emp_id = 101
connect by mng = prior emp_id
;
```

```
no rows returned
```

Demo 10: How many people are on the branch starting at employee 146

```
select count(*)
from hier_emp E1
start with emp_id = 146
connect by mng = prior emp_id
;
```

```
  COUNT(*)
----------
         3
```

3. connect_by_isleaf

This functions returns a 1 if the row is a leaf and 0 if it is not.

Demo 11: Which employees are not managers?

```
select level, emp_id, name_last
from hier_emp
where connect_by_isleaf = 1
start with  mng is null
connect by  prior emp_id = mng
order by mng nulls first, emp_id;
```

```
     LEVEL     EMP_ID NAME_LAST
---------- ---------- ----------
         2        201 Harts
         3        162 Holme
         3        200 Whale
         3        203 Mays
         4        104 Ernst
         4        109 Fiet
         4        110 Chen
         3        150 Tuck
         3        155 Hiller
         3        207 Russ
         3        160 Dorna
         3        161 Dewal
         4        204 King
         4        206 Gietz
```

Demo 12: Which employees are managers?

```
select level, emp_id, name_last
from hier_emp
where connect_by_isleaf = 0
start with mng is null
connect by prior emp_id = mng
order by  mng nulls first, emp_id;
```

```
     LEVEL     EMP_ID NAME_LAST
---------- ---------- ----------
         1        100 King
         2        101 Koch
         2        102 D'Haa
         2        145 Russ
         2        146 Partne
         3        108 Green
         3        205 Higgs
         3        103 Huno
```

4. SYS_CONNECT_BY_PATH

SYS_CONNECT_BY_PATH returns the path of a column value from root to node, with column values separated by a string literal for each row returned by CONNECT BY condition.

Demo 13:

```
select SYS_CONNECT_BY_PATH(emp_id || ' ' || name_last, ' -- ')
           as  "Path"
from hier_emp
start with  emp_id = 101
connect by  prior emp_id = mng
;
```

```
Path
--------------------------------------------------
 -- 101 Koch
 -- 101 Koch -- 108 Green
 -- 101 Koch -- 108 Green -- 109 Fiet
 -- 101 Koch -- 108 Green -- 110 Chen
 -- 101 Koch -- 162 Holme
 -- 101 Koch -- 200 Whale
 -- 101 Koch -- 203 Mays
 -- 101 Koch -- 205 Higgs
 -- 101 Koch -- 205 Higgs -- 204 King
 -- 101 Koch -- 205 Higgs -- 206 Gietz

10 rows selected.
```

To get the full tree, use: `start with  mng is null`

Demo 14: Sometimes the nested levels are indented by using lpad

```
select lpad(' ', 6*level-1)||
            SYS_CONNECT_BY_PATH(emp_id || ' ' || name_last, '--')  AS  "Path"
from hier_emp
start with  emp_id = 101
connect by  prior emp_id = mng
;
```

```
Path
--------------------------------------------------
     --101 Koch
           --101 Koch--108 Green
                 --101 Koch--108 Green--109 Fiet
                 --101 Koch--108 Green--110 Chen
           --101 Koch--162 Holme
           --101 Koch--200 Whale
           --101 Koch--203 Mays
           --101 Koch--205 Higgs
                 --101 Koch--205 Higgs--204 King
                 --101 Koch--205 Higgs--206 Gietz

10 rows selected.
```

5. Order Siblings By

This option lets you have the sibling rows sorted. The siblings are rows of the same parent row.

Demo 15:

```
select lpad(' ', 6*level-1)||
            SYS_CONNECT_BY_PATH(emp_id || ' ' || name_last, '--')  AS  "Path"
from hier_emp
start with  emp_id = 101
connect by  prior emp_id = mng
order siblings by name_last
;
```

```
Path
----------------------------------------------------
     --101 Koch
           --101 Koch--108 Green
                 --101 Koch--108 Green--110 Chen
                 --101 Koch--108 Green--109 Fiet
           --101 Koch--205 Higgs
                 --101 Koch--205 Higgs--206 Gietz
                 --101 Koch--205 Higgs--204 King
           --101 Koch--162 Holme
           --101 Koch--203 Mays
           --101 Koch--200 Whale

10 rows selected.
```

Demo 16:

```
select lpad(' ', 6*level-1)|| emp_id || ' ' || name_last  AS  "Path"
,  mng
from hier_emp
start with  emp_id = 101
connect by prior emp_id = mng
order siblings by name_last
;
```

```
Path                              MNG
------------------------------ ----------
     101 Koch                          100
           108 Green                   101
                 110 Chen              108
                 109 Fiet              108
           205 Higgs                   101
                 206 Gietz             205
                 204 King              205
           162 Holme                   101
           203 Mays                    101
           200 Whale                   101

10 rows selected.
```