

Communities

[Intelligent
Performance](#)

[IntelligentCRM](#)

[Intelligent
Applications](#)

[Intelligent Integration](#)

Sponsored by **Sunopsis**

[Intelligent Portals](#)

Information Centers

[Analytic Applications](#)

[Business Intelligence](#)

[Database](#)

[Data Integration](#)

[Data Warehousing](#)

[Enterprise
Development](#)

[Storage](#)

[Supply Chain](#)

Celko

October 20, 2000



Email Article



Print Article

Trees in SQL

Some answers to some common questions about SQL trees and hierarchies.

By Joe Celko

This theme is an old one of mine, but it is worth repeating. I have been seeing too many questions about SQL trees and hierarchies in the newsgroup discussions. The usual example of a tree structure in SQL books is called an adjacency list model and it looks like this:

```
CREATE TABLE Personnel
(emp CHAR(10) NOT NULL PRIMARY KEY,
boss CHAR(10) DEFAULT NULL REFERENCES Personnel(emp),
salary DECIMAL(6,2) NOT NULL DEFAULT 100.00);
```

Personnel		
emp	boss	salary
'Albert'	'NULL'	1,000.00
'Bert'	'Albert'	900.00
'Chuck'	'Albert'	900.00
'Donna'	'Chuck'	800.00
'Eddie'	'Chuck'	700.00
'Fred'	'Chuck'	600.00

Table 1: Adjacency List Mode



Other Resources

- [IE Imperatives](#)
- [Archives](#)
- [Realware Awards](#)
- [Career Center](#)
- [Research Library](#)
- [Event Calendar](#)
- [Editorial Calendar](#)
- [Advanced Search](#)
- [Translate](#)

Another way of representing trees is to show them as nested sets. Because SQL is a set-oriented language, this model is a better than the usual adjacency list approach you see in most textbooks. Let us define a simple Personnel table like this, ignoring the left (lft) and right (rgt) columns for now:

```
CREATE TABLE Personnel
(emp CHAR(10) NOT NULL PRIMARY KEY,
 lft INTEGER NOT NULL UNIQUE CHECK (lft > 0),
 rgt INTEGER NOT NULL UNIQUE CHECK (rgt > 1),
 CONSTRAINT order_okay CHECK (lft < rgt) );
```

This problem is always given with a column for the employee and one for his boss in the textbooks. This table (Table 2) -- without the lft and rgt columns -- is called the adjacency list model, after the graph theory technique of the same name; the pairs of nodes are adjacent to each other.

Personnel emp	lft	rgt
'Albert'	1	12
'Bert'	2	3
'Chuck'	4	11
'Donna'	5	6
'Eddie'	7	8
'Fred'	9	10

Table 2: Nested Set Model

The organizational chart would look like Figure 1 as a directed graph:

Figure 1

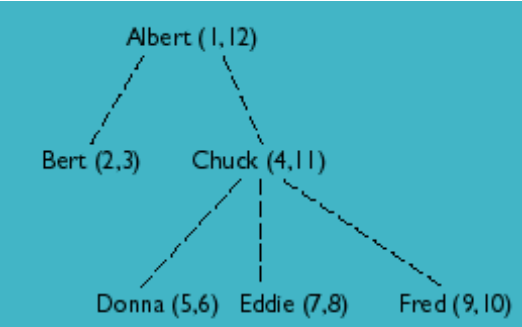


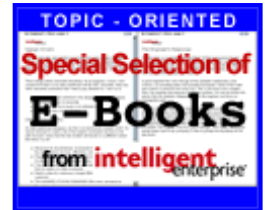
Figure 1: Directed graph.

Table 1 is denormalized in several ways. We are modeling both the personnel and the organizational chart in one table. But for the sake of saving space, pretend that the names are job titles and that we have another table that describes the personnel that hold those positions.

Another problem with the adjacency list model is that the boss and employee columns are the same kind (names of personnel) and therefore should be shown in only one column in a normalized table. To prove that this is not normalized, assume that "Chuck" changes his name to "Charles;" you have to change his name in both columns and several places. The defining characteristic of a normalized table is that you have one fact in one place at one time.

The final problem is that the adjacency list model does not model subordination. Authority flows downhill in a hierarchy, but if I fire Chuck, I disconnect all of his subordinates from Albert. There are times (such as water pipes) where this is true, but that situation is not expected in this case.

To show a tree as nested sets, replace the nodes with ovals, then nest subordinate ovals inside each other. The root will be the largest oval and will contain every other node. The leaf nodes will be the innermost ovals with nothing else inside them and the nesting will show the hierarchical relationship. The rgt and lft columns (I cannot use the reserved words LEFT and RIGHT in SQL) are what show the nesting.



FOCUS ON... [Security](#)

- ▶ [The latest security news](#)
- ▶ [Desktop Security Information](#)

Brought to you by

**security
pipeline**

[Copyright © 2004 CMP Media LLC](#)
[Privacy Statement](#) · [Terms Of Service](#)
a United Business Media Company
ALL RIGHTS RESERVED
No reproduction without permission



TRANSFORM



[Home](#) | [Privacy Policy](#) | [Feedback](#) | [Subscribe](#) | [CMP Media LLC](#)