

# Different Approaches to Modeling the Trees Data in Relational Database

Abdelali Elloub  
Computer science  
Faculty of Science Dhar El  
Mahraz,  
University Sidi Mohammed  
Ben Abdellah, Fez, Morocco

Ouçamah Mohammed  
Cherkaoui Malki  
Computer science  
Faculty of Science Dhar El  
Mahraz,  
University Sidi Mohammed  
Ben Abdellah, Fez, Morocco

Sanae Mazouz  
Computer science  
Faculty of Science Dhar El  
Mahraz,  
University Sidi Mohammed  
Ben Abdellah, Fez, Morocco

## ABSTRACT

This paper presents a detailed study of the work to modeling the hierarchies and data trees in relational database area. Hierarchical structures are used in diverse domains, in engineering and in research. We give a comparison between the various models cited in literature. This study allowed us to synthesize the most interesting of these models to represents the data trees.

## General Terms

Relational database, tree data.

## Keywords

Trees data, hierarchies, SQL, relational database & models

## 1. INTRODUCTION

The hierarchical model database is the first to appear, the main problem of this system is the lack of independence of the programs with respect to data storage. However, the relational model [1] remains more used.

The relational databases (RDB) provide an excellent support for the representation and storage of the data within

information systems. This model is based to relational Algebra.

The hierarchies and trees data usually used in different domains for example: healthcare area. The tree structures facilitate the modeling of the System of information. They reduce complexity of connections between the entities.

The managing of the data in the information systems is a convivial functionality. This function is complex for the managing the graphs data. There are several SQL techniques to query tree structures [2][3] [4][5] [6]. They can be classified into two major categories: tree modeling approach and SQL recursive approach.

This article is organized as follows. Firstly in section introduction, we present the paramount role of the relational model to managing the data, and the advantage the hierarchies for modeling the complex systems. Then in section 2, we describe a variety of approach to modeling the relational database. Finally we conclude by a comparison between the various models study in section 2.

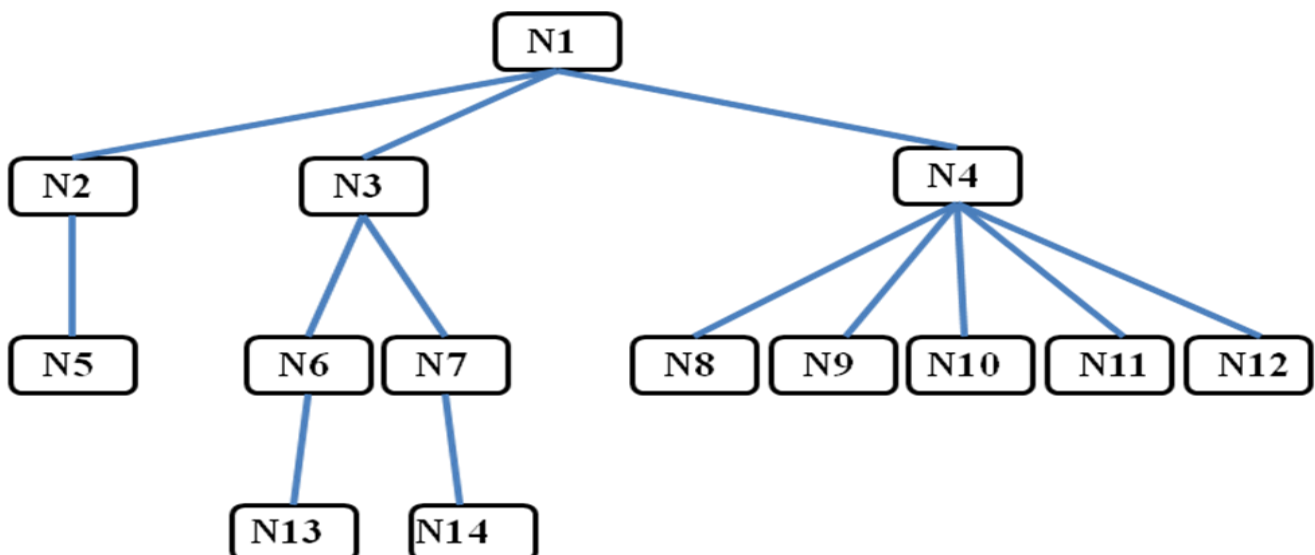


Fig 1: An abstract data tree

## 2. Models to representation the trees data

In this section, we give a study a different approach to storing and retrieving of the trees data into relational database.

### 2.1 Trees Modeling Approaches

This category is very efficiency to launch finding queries in database. However, the updating queries Requires a lot time.

#### 2.1.1 Adjacency List Model

The Adjacency List Model is a simplest representation of efficiently managing hierarchical data in a relational database. it consists in creating the doublets (key of the node, key of the parent). Table I is a representation of the abstract model noted in Figure 1.

We appreciate using integer data types rather than of string, so the creating query handling the trees table is as follows, either the table AdjacencyTree representing this model (see Table 1):

```
CREATE TABLE AdjacencyTree (
    Node INT NOT NULL,
    Parent INT,    -- null is root
    PRIMARY KEY (Node, Parent)
);
```

- The field “Node”: the child node is the reference value to the node stored in the table data.
- The field “Parent”: the parent node is the reference value to the node parent stored in the table data.

**Table 1. Adjacency List Model**

Node	Parent
N1	NULL
N2	N1
N3	N1
N4	N1
N5	N2
N6	N3
N7	N3
N8	N4
N9	N4
N10	N4
N11	N4
N12	N4
N13	N6
N14	N7

This representation constitutes the tree data stored in the database. For a better use of this model it is necessary to separate the data of the metadata from the structure to be stored. The self joins is an efficiency method for doing a tree traversal, but is limited to a known depth of traversal, which is

not always possible.

This model is having problems [2][5]:

1. This model requires complex constraints to maintain any data integrity.
2. This model needs to traverse through all nodes for any query.

In add this problems, this model is to be proscribed when:

- ✓ The tree is deep (more than 5 levels).
- ✓ The tree is broad (more than 100 elements on the same level).
- ✓ The tree contains many values (from 200 to 300 elements).

#### 2.1.2 Nested Set Model

The nested set model is an easy representation of efficiently managing hierarchical data in a relational database. In this model each nodes is indexed by two information (left bound and right bound). Either the table NestedTree representing this model (see Table 2):

**Table 2. Nested Set Model**

Node	LeftBound	RightBound
1	1	28
2	2	5
3	6	15
4	16	27
5	3	4
6	7	10
7	11	14
8	17	18
9	19	20
10	21	22
11	23	24
12	25	26
13	8	9
14	12	13

```
CREATE TABLE NestedTree (
```

```
    Node INT NOT NULL PRIMARY KEY,
```

```
    LeftBound INT NOT NULL, -- left bound
```

```
    RightBound INT NOT NULL -- right bound
```

```
);
```

- The field “Node”: Is the reference value to the node stored in the table data.
- The field “LeftBound”: is the left bound value of the node.
- The field “RightBound”: is the right bound value of the node.

The assignment of the keys to the tree nodes is carried out by traversing the tree of the nodes on the left towards the nodes on the right [4][5].

The majority of research operations in this model require only one simple request of SQL.

For a leaf, one always has: bound right - bound left = 1.

For a node, one always has: bound right - bound left > 1.

Those which easy make it possible to recover a subtree with only one SQL query, see an example in Figure 2.

Either the subtree N3= {N3, N6, N7, N13 and N14}.

- The set {N3, N6, N7} are Nodes.
- The set {N13, N14}: are leaves.

SQL code for the selection the subtree N3: “Select **Node** from NestedTree where **Leftbound** >= 6 and **RightBound** <= 15”.

The Nested Set Model, updating queries are expansive.

### 2.1.3 Nested Interval Model

Vadim Tropashko has stated that Nested Intervals generalize Nested Sets and they are immune to hierarchy reorganization problem [7] [8] [9] [10]. He proved that it is possible to use rational numbers. The problem in this model is calculating the each values of split. We invited to find the Greatest Common Divisor (GCD) to keep the numerators and denominators as small as possible [8] [4].

### 2.1.4 Materialized Path

Materialized path is a encoding where each tree node is labeled with the path from the node to the root [11]. Materialized Path could be represented a character string separate by any separator for example “/”. Querying trees with

Materialized Path technique necessity to parsing data field string. In this section we present the Path enumeration Model.

This model is very efficiently, was first published by Stefan Gustafsson on a site of SQL Server, then Tom Moreau and Itzak Ben-Gan developed have in their book [3][4][5][6].

This model saves the path from the root to each node as a string (see Figure 3). Either the table PathTree representing this model (see the Table 3):

```
CREATE TABLE PathTree (
    Node INT NOT NULL PRIMARY KEY,
    Path VARCHAR (MAX) NOT NULL
);
```

- The field “Node”: Is the reference of the node stored in the table data.
- The field “Path”: Is the path from the root to node, the information stored in the column PATH is a sequence of alphabet separated by a separator such as for example: “/”.
- MAX: is the maximum of character contained in field Path.

The Path Enumeration Model is has an improvement of the Adjacency List Model. However, is sensitive of the updating queries.

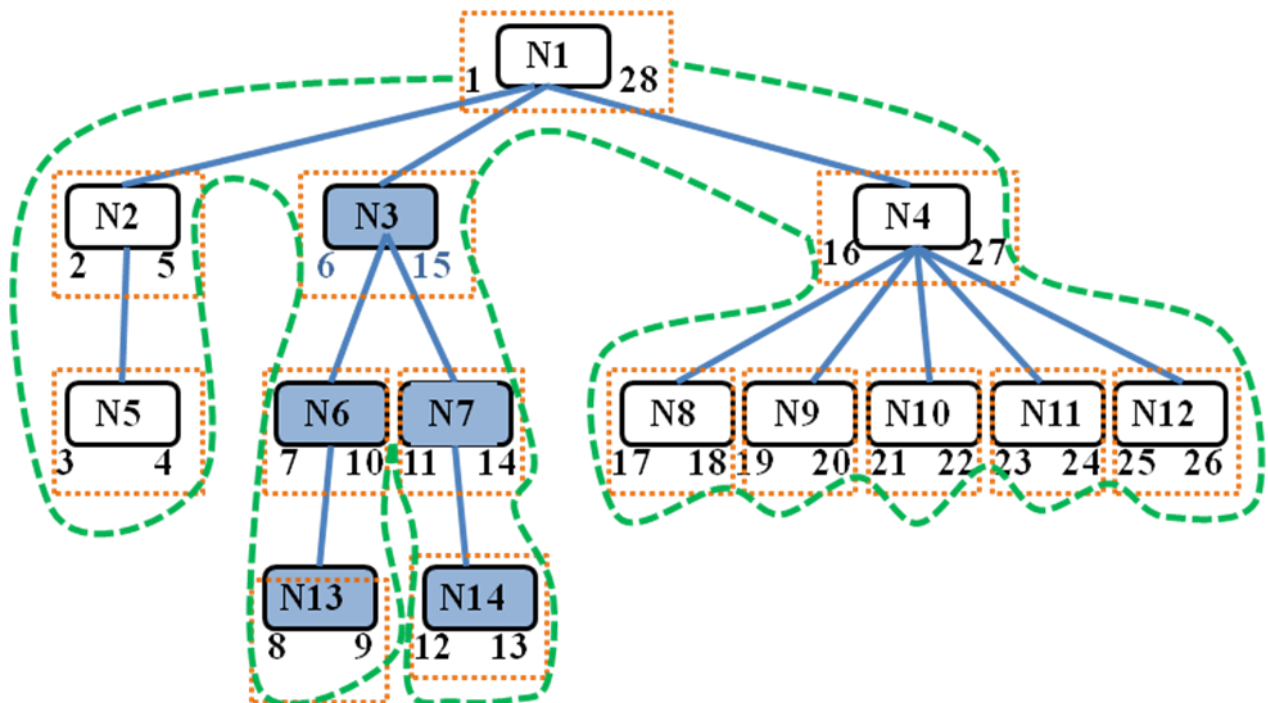


Fig 2: The subtree N3 with the Nested Set Model

**Table 3. Path Enumeration Model**

Node	The Path field
1	/a
2	/a/b
3	/a/c
4	/a/d
5	/a/b/e
6	/a/c/f
7	/a/c/g
8	/a/d/h
9	/a/d/i
10	/a/d/j
11	/a/d/k
12	/a/d/l
13	/a/c/f/m
14	/a/c/g/n

## 2.2 SQL Recursive

They can be classified into two Subcategories: SQL standard and derived SQL.

### 2.2.1 SQL Standard (Recursive with CTE)

In the third version of standard SQL named SQL3 or SQL99 is added the Common Table Expression (CTE). It is also a query expression that is given a name, just like a derived table. In this approach, we must use the recursive Common Table Expression [11]. A recursive definition of a set has two parts: The fixed point, and by applying a rule to the previous step results. The syntax for this construct includes a UNION [ALL]:

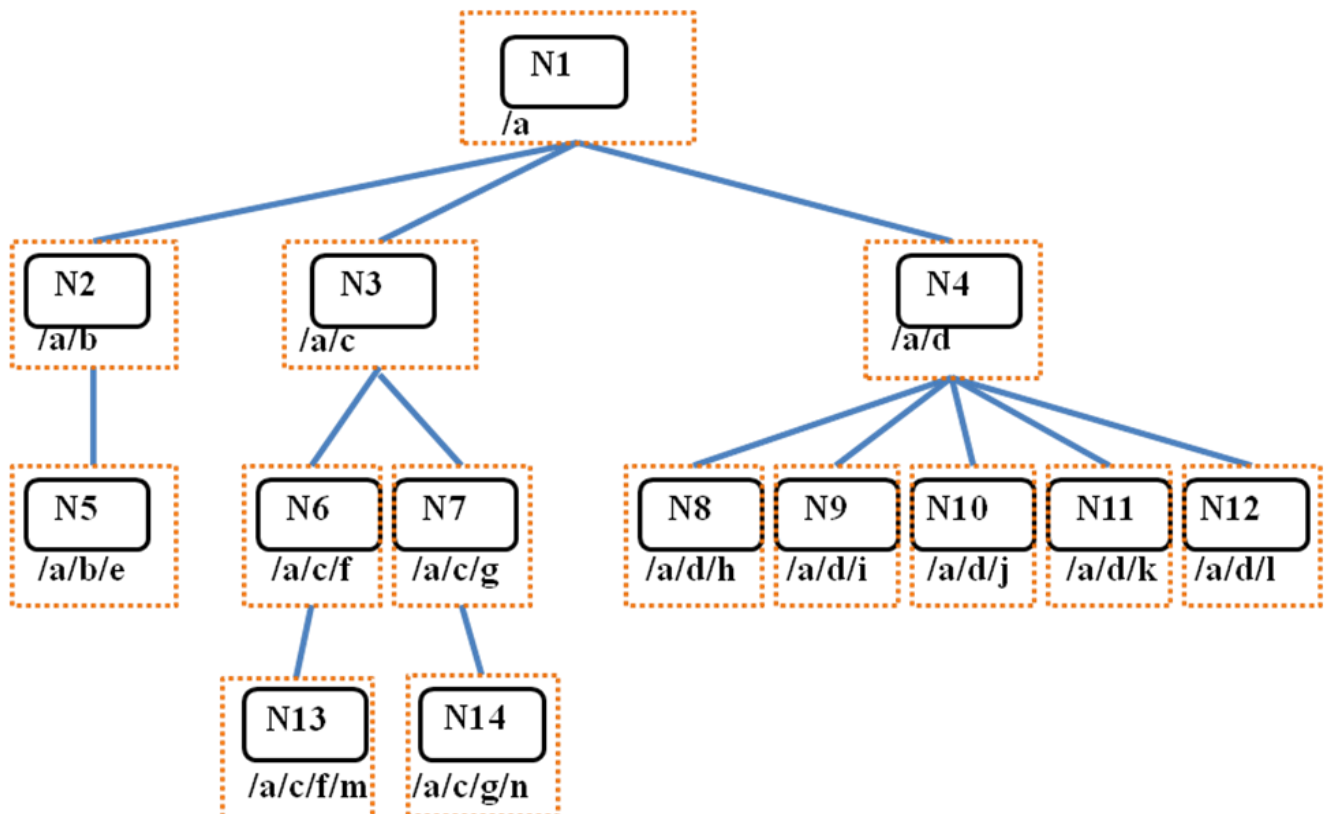
<cte select statement>::=

**WITH RECURSIVE** <fixed point cte element>

**UNION** [ALL]

<step cte element>

<select statement>;



**Fig 3: Path Enumeration Model**

### 2.2.2 Recursive with derived SQL (T-SQL, PL/SQL)

This approach is based on the models already cited, for example: Nested Set Model and Adjacency List Model. The SQL used depends on database management systems (DBMS).

To managing the hierarchies in SQL Server, in Sybase [12], we used the T-SQL, LEFT JOINT, and we have the choice between all models:

- Adjacency List Model
- Nested Set Model
- Nested interval model
- Other model

## 3. CONCLUSION AND FUTURE WORK

This work presents a study of the approach to handling the hierarchies and trees data in relational database. We have given an advantage and disadvantage of those models. This paper allowed us to synthesize the most interesting of these models to modeling the data trees. All this approaches are sensitive of the updating queries.

Hence the necessity in future works:

- ✓ To improve of approaches that is not sensitive to updating queries.
- ✓ To generalize these representations in object databases and semi-structured, especially to parsing the XML files.

## 4. REFERENCES

- [1] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [2] Celko, J., 1999. *SQL for Smarties: Advanced SQL Programming*. Morgan Kaufmann Publishers, San Francisco, CA, 576pp.
- [3] Celko, J., 2008. *Thinking in Auxiliary Sets, Temporal, and Virtual Tables in SQL*. Morgan Kaufmann Publishers, Burlington, MA 01803-4255.
- [4] Celko, J., 2004. *Trees & Hierarchy in SQL for Smarties*. Morgan Kaufmann Publishers, San Francisco, CA 94111.
- [5] Celko.J., 2005. *SQL for Smarties: Advanced SQL Programming third edition*. Morgan Kaufmann Publishers, San Francisco, continuation 400, CA 94111.
- [6] Celko.J., 2011. *SQL for Smarties: Advanced SQL Programming fourth edition*. Morgan Kaufmann Publishers, Burlington, MA 01803,
- [7] V. Tropashko. June 2005. *Nested Intervals Tree Encoding in SQL*, SIGMOD.
- [8] V. Tropashko. *Nested Intervals Tree Encoding with Continued Fraction*. <http://arxiv.org/pdf/cs.DB/0402051>
- [9] V. Tropashko. 2003. *Trees in SQL: Nested Sets and Materialized Path*
- [10] V. Tropashko, 2004. *Nested Intervals with Farey Fractions*. <http://arxiv.org/html/cs.DB/0401014>
- [11] J. Roy. 2003. *Using the Node Data Type to Solve Problems with Hierarchies in DB2 Universal Database* <http://www.ibm.com/developerworks/data/library/techarticle/0302roy/0302roy.html>
- [12] Mohammad G. Ali, December 2011. *Evolution of Database Emerging to Sybase Adaptive Server Enterprise and Ensuring Better Server Performance Tuning and Query Optimization*. *International Journal of Computer Applications*