

# Chapter 10

- **Recursive hierarchies**



- Standard OLTP solution
- DM solution with bridge table
- **Nested sets model (Joe Celko) ... after March 7**
  - Not in text

- Wikipedia references SQL Antipatterns page 44:

*Nested Sets is a clever solution – maybe too clever. It also fails to support referential integrity. It's best used when you need to query a tree more frequently than you need to modify the tree*

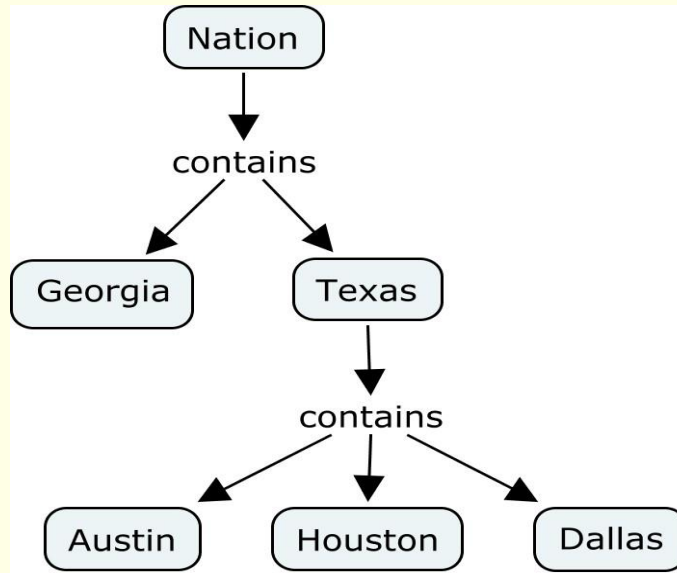
# Nested Sets

- References:

- [http://en.wikipedia.org/wiki/Nested\\_set\\_model](http://en.wikipedia.org/wiki/Nested_set_model)
- Joe Celko's Analytics & OLAP in SQL 
- Joe Celko's SQL for Smarties: Advanced SQL Programming 

# Nested Sets

Consider the geographical hierarchy :



For the above we can create a table with 3 fields:

```
CREATE TABLE Regions (  
  regionName varchar(10) ,  
  left integer ,  
  right integer  
)
```

# Nested Sets

Properties:

The root has  $Left=1$  and  $Right=2*(\text{select count(*) from GeoRegionsDim})$

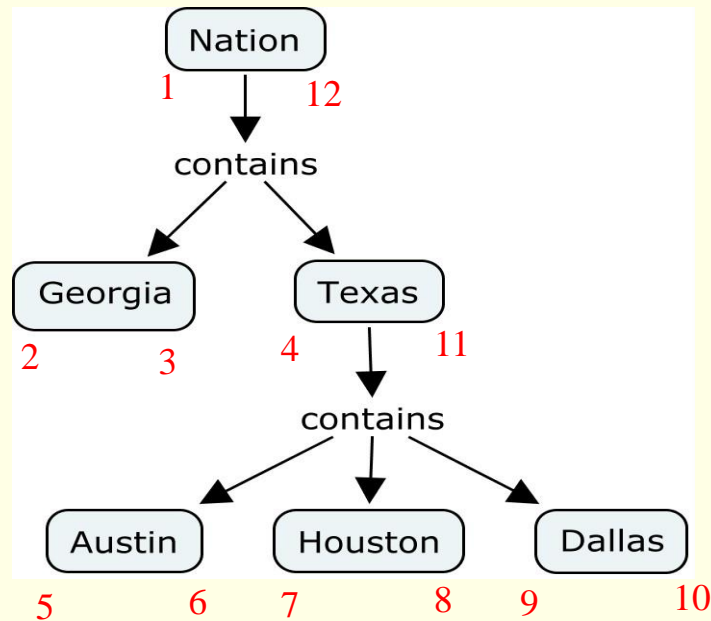
Leaf nodes have  $Left+1=Right$

Setting the left and right values:

1. Set count to 1
2. Traverse the nodes according to a pre-order traversal beginning at root
  - When visiting a node for the first time assign left the value of count, and then increment count by 1
  - When you visit a node for the second time assign right the value of count, and then increment Count by 1
  - Leaf nodes  $Right \leftarrow Left+1$ , and then set Count to  $Right+1$

# Nested Sets

Example:



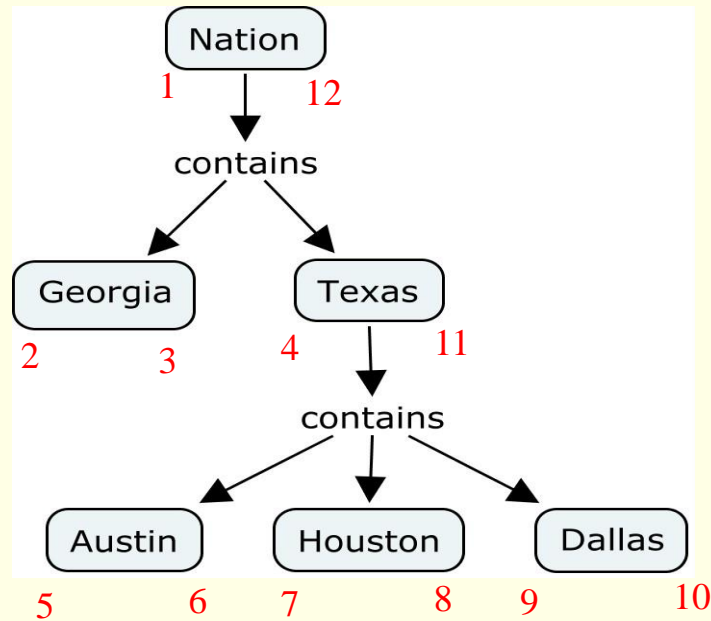
Note:

- The root has Left=1 and Right=2\*(select count(\*) from GeoRegionsDim)
- Consider a subtree headed by node S.
  - If R1 is a node beneath S then R1.Left is between R2.Left and R2.Right. Similarly R1.Right ...
- Leaf nodes have Left+1 equals Right



# Nested Sets

Example:

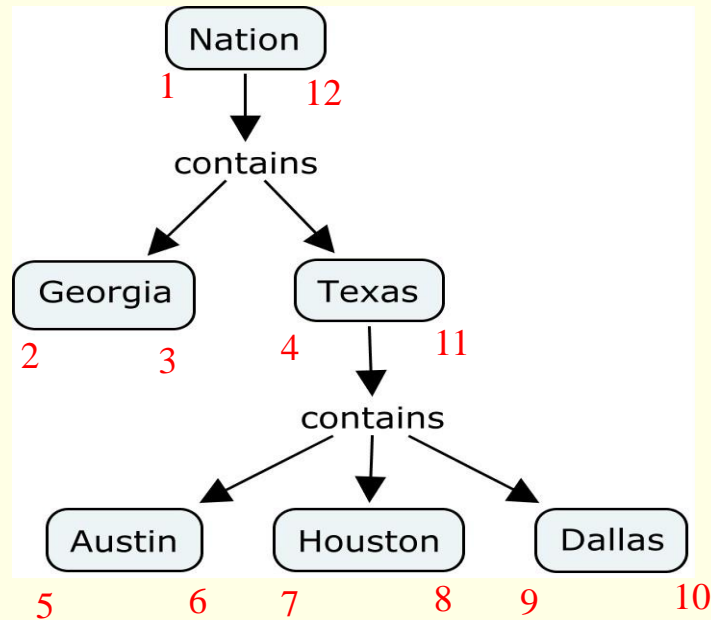


To List the regions within a specific region, say Texas:

```
SELECT w.regionName
FROM Region AS w INNER JOIN Region AS a
      ON w.Left > a.Left AND w.Left < a.Right
WHERE (a.regionName = 'Texas')
```

# Nested Sets

Example:



To find the immediate children of a node is difficult

From Wikipedia ... unless we add a field for depth

```
SELECT DISTINCT Child.Node, Child.Left, Child.Right
```

```
FROM Tree as Child, Tree as Parent
```

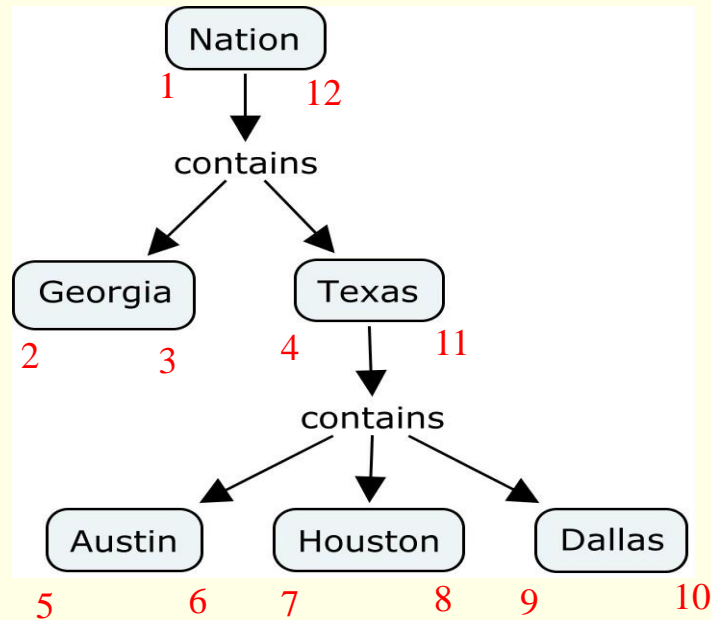
```
WHERE Parent.Left < Child.Left AND Parent.Right > Child.Right
```

```
GROUP BY Child.Node, Child.Left, Child.Right
```

```
HAVING max(Parent.Left) = 1
```

# Nested Sets

Example:



To find the immediate children of a node is difficult

**However**, what if we stored the parent node's key (or ...) in the child?