

## QlikView Design Blog



# Authorization using a Hierarchy

Posted by [Henric Cronström](#) in [QlikView Design Blog](#) on Nov 26, 2013 1:58:39 AM

Hierarchies are very common in all database and business intelligence solutions. Often they are used for authorization purposes, i.e. the permissions to see data follows a hierarchy.

One example is an organizational hierarchy. Each manager should obviously have the right to see everything pertaining to their own department, including all its sub-departments. But they should not necessarily have the right to see other departments.



This means that different people will be allowed to see different parts of the organization. The authorization table may look like the following:

| ACCESS | NTNAME   | Person | Position             | Permissions       |
|--------|----------|--------|----------------------|-------------------|
| USER   | ACME\BKO | Bill   | CPO                  | HR                |
| USER   | ACME\DKN | Diane  | CEO                  | CEO               |
| USER   | ACME\DPT | Debbie | Director Engineering | Engineering       |
| USER   | ACME\JOO | John   | CFO                  | Finance           |
| USER   | ACME\LBY | Les    | COO                  | Sales & Marketing |
| USER   | ACME\SDN | Steve  | CTO                  | Product           |

In this case, Diane is allowed to see everything pertaining to the CEO and below; Steve is allowed to see the Product organization; and Debbie is allowed to see the Engineering organization only. Hence, this table needs to be matched against *sub-trees* in the above hierarchy.

Often the hierarchy is stored in an Adjacent Nodes table, and if so, the above problem is easy to solve: Just load the Adjacent nodes table using a [HierarchyBelongsTo](#) and name the ancestor field *Tree*. See the blog post [Unbalanced, n-level hierarchies](#) how this is done.

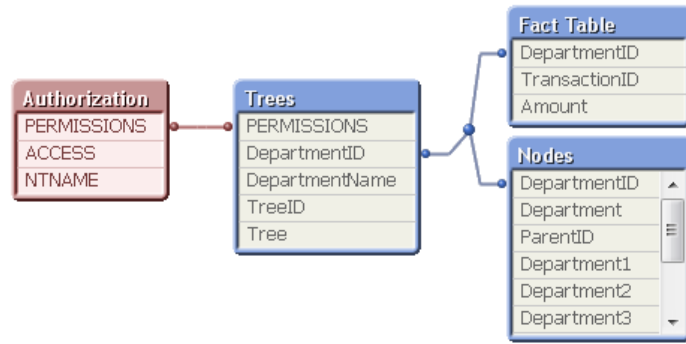
If you want to use [Section Access](#), you need to load an upper case copy of *Tree* and call this new field *PERMISSIONS*. Finally, you need to load the authorization table. These two last steps can be done using the following script lines: (The *TempTrees* table is the table created by the [HierarchyBelongsTo](#).)

```

Trees:
Load *,
    Upper(Tree) as PERMISSIONS
Resident TempTrees;
Drop Table TempTrees;

Section Access;
Authorization:
Load ACCESS,
    NTNAME,
    Upper(Permissions) as PERMISSIONS
From Organization;
Section Application;
    
```

When you have done this, you should have a data model that looks like the following:



The red table is in Section Access and is invisible in a real application. Should you want to use the publisher for the reduction, you can reduce right away on the *Tree* field, without loading the Section Access. In either case, this solution will effectively limit the permissions to only the sub-tree as defined in the authorization table.

But what if you have the hierarchy in a horizontal hierarchy? Then you cannot use the [HierarchyBelongsTo](#).

| DepartmentID | Department         | Board level | Director level   | Department        | Unit               |
|--------------|--------------------|-------------|------------------|-------------------|--------------------|
| 1            | Board              | Board       | -                | -                 | -                  |
| 2            | Audit department   | Board       | Audit department | -                 | -                  |
| 4            | CEO                | Board       | CEO              | -                 | -                  |
| 5            | Finance            | Board       | CEO              | Finance           | -                  |
| 13           | Engineering        | Board       | CEO              | Product           | Engineering        |
| 14           | Quality            | Board       | CEO              | Product           | Quality            |
| 16           | Global Marketing   | Board       | CEO              | Sales & Marketing | Global Marketing   |
| 17           | Sales Department A | Board       | CEO              | Sales & Marketing | Sales Department A |
| 18           | Sales Department B | Board       | CEO              | Sales & Marketing | Sales Department B |
| 23           | Logistics Group    | Board       | CEO              | Service           | Logistics Group    |
| 26           | Salary Group       | Board       | CEO              | HR                | Salary Group       |

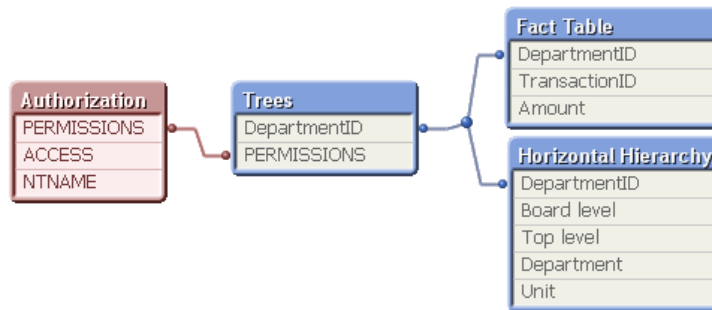
A horizontal hierarchy

The solution is not very different from the above one. The only difference is that you need to create the bridging table manually, e.g. by using a loop:

```

Let vHierarchyDefinition = 'Board level,Director level,Department,Unit';
Let vNumberOfLevels = Len(KeepChar(vHierarchyDefinition,',')) + 1 ;
For vAncestorLevel = 1 to vNumberOfLevels
    Let vAncestor = Subfield(vHierarchyDefinition,',';vAncestorLevel);
    Trees:
    Load distinct
        Upper({$(vAncestor)}) as PERMISSIONS,
        DepartmentID
    Resident [Horizontal Hierarchy]
        Where Len({$(vAncestor)}) > 0;
Next vAncestorLevel
    
```

Having done this, you will have the following data model:



Bottom line is that it is fairly straightforward to implement a hierarchical authorization scheme. See more about Hierarchies in the Tech Brief [Hierarchies](#).

HIC

### 6 Comments



[Kalyan Sundaram](#) Nov 26, 2013 2:06 AM

HIC,  
Very Interesting and intelligent post  
Thanks and Best Regards,  
Kalyan

Like  
(0)



[eduardo ale](#) Nov 26, 2013 8:07 AM

Very Interesting ..  
Thanks  
Eduardo

Like  
(0)



[Car Bal](#) Nov 26, 2013 9:33 AM

A different approach to a common issue.  
Thanks Henric.  
**CB**

Like  
(0)



[Bill Ringer Salalima](#) Nov 27, 2013 8:00 PM

This is interesting, might use it in future development. Kudos!

Like  
(0)



[Karthick Venkatachalam](#) Nov 28, 2013 11:15 PM

Excellent post , Thanks HIC ..

Like  
(0)



[Vikas Mahajan](#) Nov 28, 2013 11:39 PM

Thanks HIC this is very important while we need to have better security.  
100 % important for all organisations.  
Can you post sample QVW with necessary Excel .  
Vikas  
.

Like (0)

### Sections

- Discussion Forums
- Resources
- Ideas

- Groups
- Blogs
- New to QlikView

### Blogs

- Business Discovery
- QlikView Design
- Community Manager Blog

- QlikView Support Updates
- Technical Bulletin
- All Blogs

### QlikView Sites

- QlikView.com
- Qconnect Partner Portal
- Customer Portal
- QlikMarket
- Demos