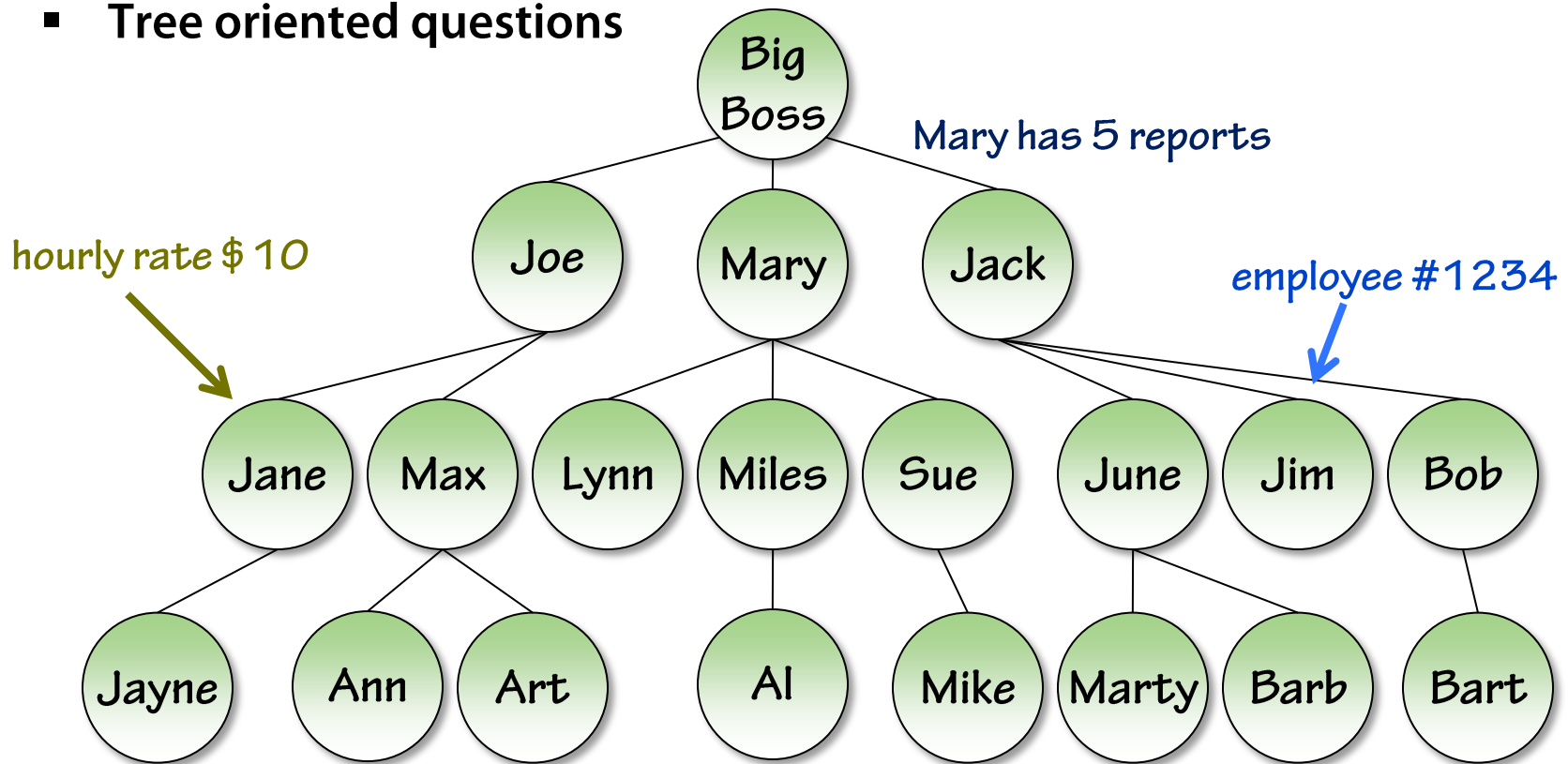# Hierarchies

pluralsight
see what you can learn

Overview

- **SQL Server is relational**
  - trees are in the mind of the questioner
- **Hierarchyid**   *flat*
  - non-recursive queries
  - early 90's

Hierarchyid's are for Tree Oriented Questions

- **Table is set of entities of same type**
  - entity ~ row described by column values
- **Tree oriented questions**

Big Boss

Mary has 5 reports

hourly rate $ 10

Joe   Mary   Jack

employee #1234

Jane   Max   Lynn   Miles   Sue   June   Jim   Bob

Jayne   Ann   Art   Al   Mike   Marty   Barb   Bart

pluralsight
see what you can learn

Why Use Hierarchyid?

+ Tree oriented questions

- converted to range queries or simple lookups

- Size

- parent foreign key typically smaller

```
select * from items where
weight >= 100 and weight < 200
```
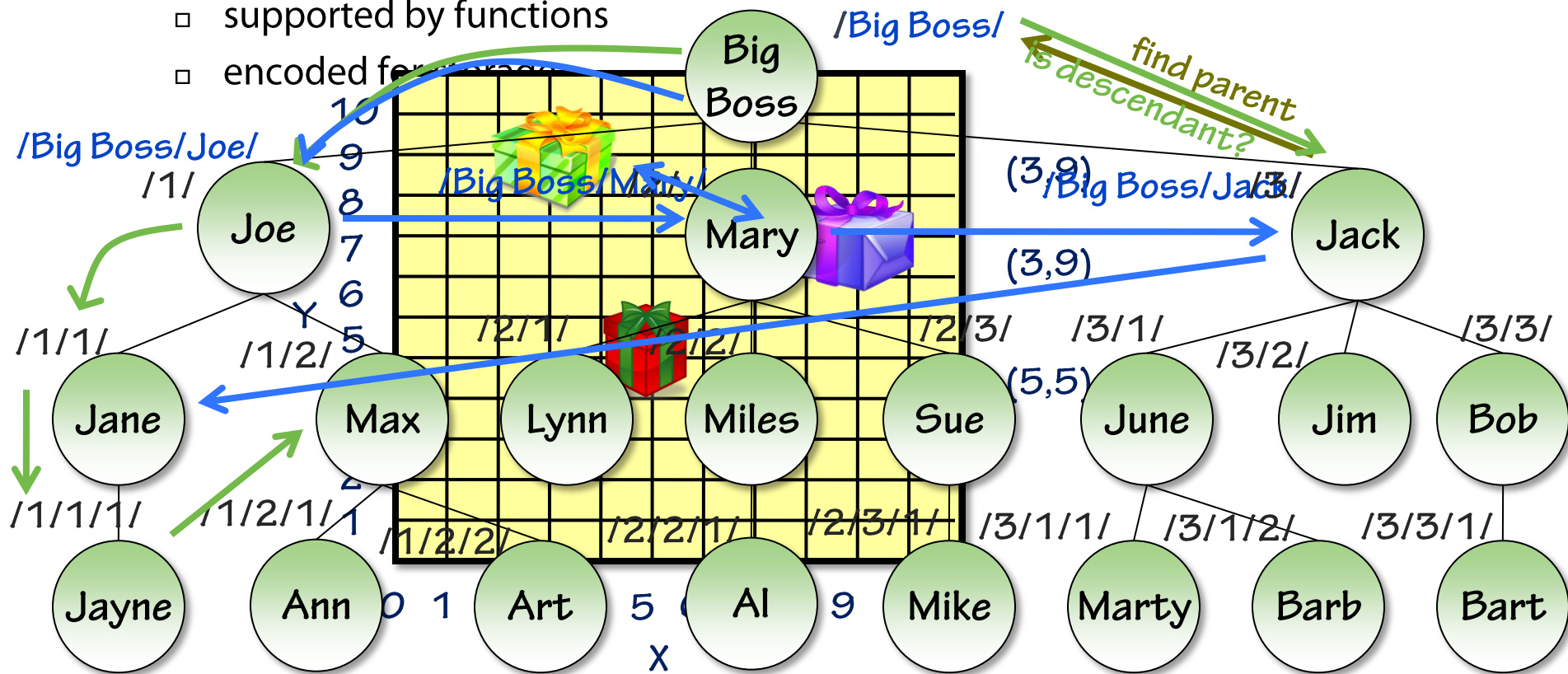
- Maintenance

- integrity

- moving a node

Hierarchyid

- **Tree** *coordinates*
  - like the Cartesian, but for trees
  - supported by functions
  - encoded for storage

/Big Boss/

find parent

Is descendant?

/Big Boss/Joe/

/Big Boss/Mary/

/Big Boss/Jack/

/1/

/1/1/   /1/2/

/2/1/   /2/3/   /3/1/   /3/3/
                           /3/2/

(3,9)

(3,9)

(5,5)

/1/1/1/   /1/2/1/   /2/2/1/   /2/3/1/   /3/1/1/   /3/1/2/   /3/3/1/

/1/2/2/

Big Boss

Joe   Mary   Jack

Jane   Max   Lynn   Miles   Sue   June   Jim   Bob

Jayne   Ann   Art   Al   Mike   Marty   Barb   Bart

10
9
8
7
6
5
Y
2
1

0   1   5   9

X

Finding Descendants

- **CLR based udt**
- **IsDescendantOf**
- **GetLevel**

Mary

```
select name from personnel
where node.IsDescendantOf('/2/') = 1 and
node.GetLevel() = 2
```

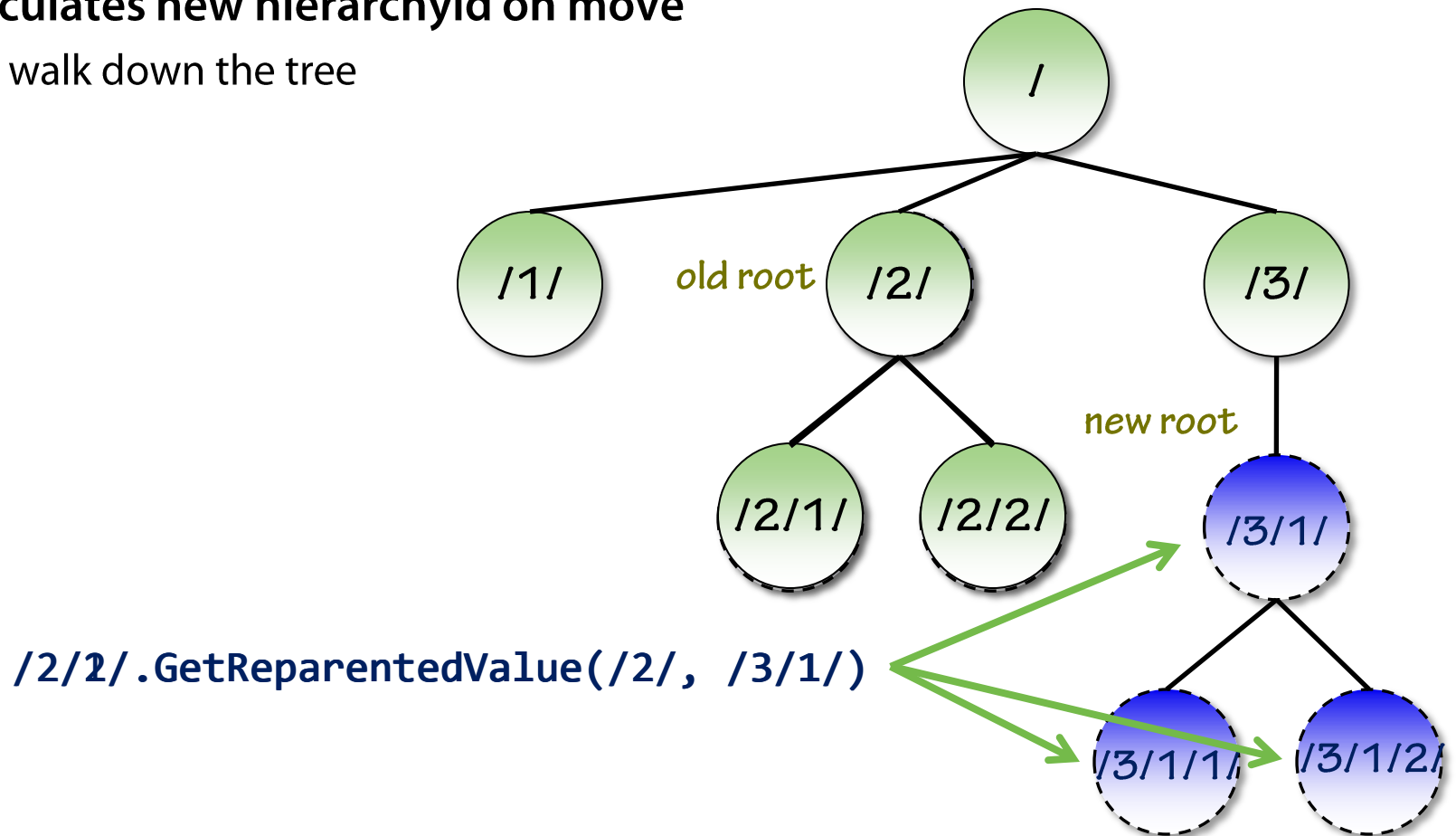next level

Adding Nodes

- **GetDescendant**
  - □ just generates a hierarchyid

parent

new descendant

```
'/2/2/'   select cast('/2/' as hierarchyid)
instance function ➜ .GetDescendant('/2/1/', '/2/3/').ToString()
```

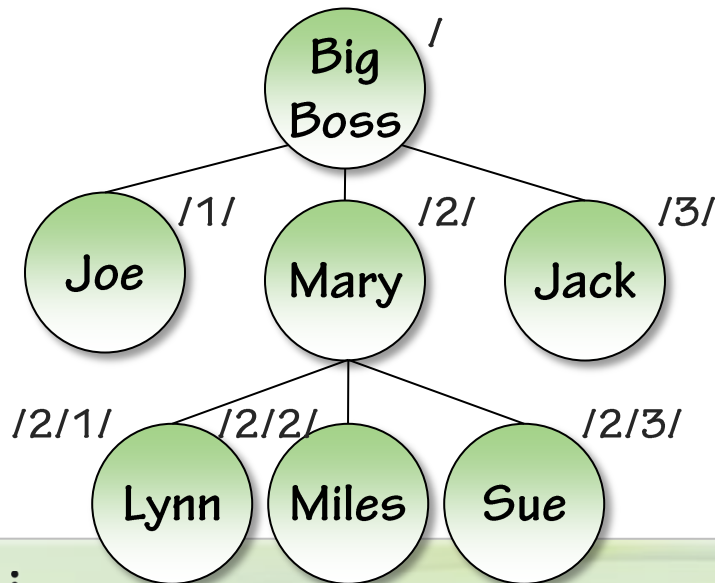*siblings*

better be empty!

GetReparentedValue

- **Calculates new hierarchyid on move**
  - walk down the tree

`/2/2/.GetReparentedValue(/2/, /3/1/)`

Depth & Breadth

- **IsDescendantOf → range query**

```
select node from personnel
where node.IsDescendantOf(Mary) = 1
and Mary.GetLevel() = node.GetLevel()+1
```


depth first

BigBoss
Joe
Jane
Jayne
Max
Ann
Art
Mary
Lynn
Miles
Al
Sue
Mike
Jack

breadth first

Jack 2-/3/
Jane 3-/1/1/
Max 3-/1/2/
Lynn 3-/2/1/
Miles 3-/2/2/
Sue 3-/2/3/
June 3-/3/1/
Jim 3-/3/2/
Bob 3-/3/3/

Summary

- hierarchyid is a numeric path to a row in a tree
- Support functions to ease management/use of hierarchyid's
- Indexes add overhead but speed up tree oriented queries
- You must guarantee the integrity of the tree

References

- **Recursive Hierarchies: The Relational Taboo!**
    - http://www.kamfonas.com/id3.html