

Managing Hierarchical Data in MySQL

Mike Hillyer - MySQL AB

PHP Quebec 2005

About Me

Mike Hillyer, BSc

- Member of the MySQL AB Documentation Team
- MySQL Core and Pro Certified
- Top MySQL Expert at www.experts-exchange.com
- Resident MySQL Expert at SearchDatabase.com
- <http://www.openwin.org/mike/aboutme.php>

About You

How many of you...

- Currently use MySQL?
- Another RDBMS?
- Manage hierarchical data?
- Will be managing hierarchical data?
- Know what hierarchical data is?

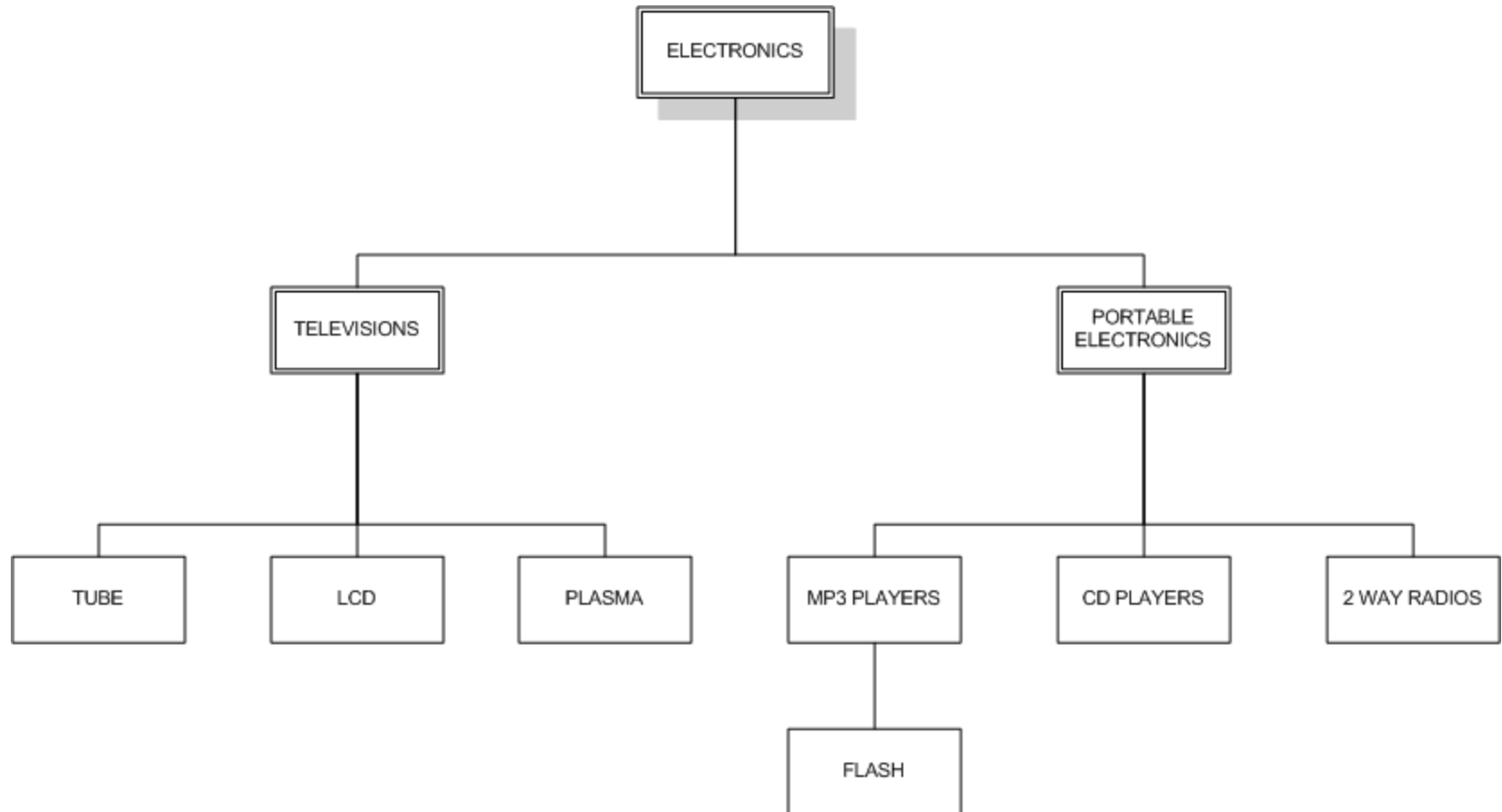
About This Session

- <http://www.openwin.org/mike/presentations/>
- <http://www.vbmysql.com/articles/managing-hierarchical-data-in-mysql.html>
- What is hierarchical data?
- The Adjacency List Model
- The Nested Set Model
- Querying the Nested Set Model
 - Adding New Nodes
 - Deleting Nodes
 - Additional Resources

What is Hierarchical Data?

- Data that maintains a parent-child relationship
- Nodes have zero or one parent, zero to many children
- Only the root node has no parent
- Various types of data
 - Mailing list/forum threads
 - Organizational charts
 - Content management categories
 - Product categories

Product Categories



The Adjacency List Model

category_id	name	parent
1	ELECTRONICS	NULL
2	TELEVISIONS	1
3	TUBE	2
4	LCD	2
5	PLASMA	2
6	PORTABLE ELECTRONICS	1
7	MP3 PLAYERS	6
8	FLASH	7
9	CD PLAYERS	6
10	2 WAY RADIOS	6

Full Tree

```
SELECT t1.name AS lev1, t2.name as lev2, t3.name as lev3, t4.name as lev4
FROM category AS t1
LEFT JOIN category AS t2 ON t2.parent = t1.category_id
LEFT JOIN category AS t3 ON t3.parent = t2.category_id
LEFT JOIN category AS t4 ON t4.parent = t3.category_id
WHERE t1.name = 'ELECTRONICS';
```

lev1	lev2	lev3	lev4
ELECTRONICS	TELEVISIONS	TUBE	NULL
ELECTRONICS	TELEVISIONS	LCD	NULL
ELECTRONICS	TELEVISIONS	PLASMA	NULL
ELECTRONICS	PORTABLE ELECTRONICS	MP3 PLAYERS	FLASH
ELECTRONICS	PORTABLE ELECTRONICS	CD PLAYERS	NULL
ELECTRONICS	PORTABLE ELECTRONICS	2 WAY RADIOS	NULL

6 rows in set (0.00 sec)

Leaf Nodes

```
SELECT t1.name
FROM category AS t1 LEFT JOIN category as t2
ON t1.category_id = t2.parent
WHERE t2.category_id IS NULL;
```

name
TUBE
LCD
PLASMA
FLASH
CD PLAYERS
2 WAY RADIOS

Single Path

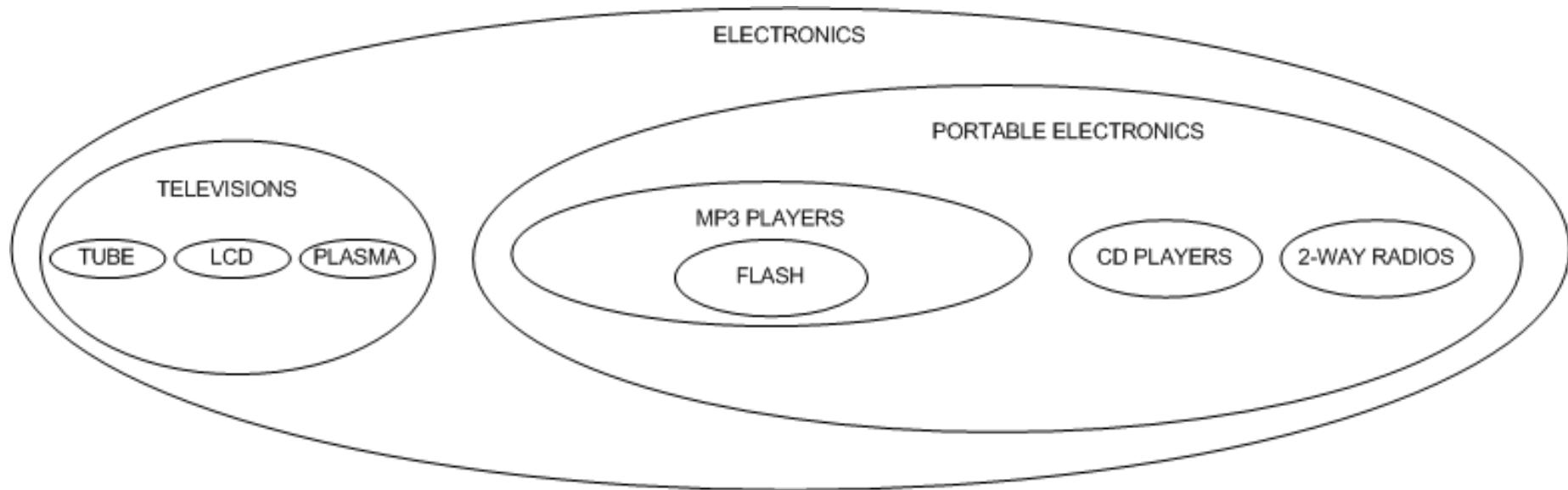
```
SELECT t1.name AS lev1, t2.name as lev2,
       t3.name as lev3, t4.name as lev4
FROM category AS t1
LEFT JOIN category AS t2 ON t2.parent = t1.category_id
LEFT JOIN category AS t3 ON t3.parent = t2.category_id
LEFT JOIN category AS t4 ON t4.parent = t3.category_id
WHERE t1.name = 'ELECTRONICS' AND t4.name = 'FLASH';
```

```
+-----+-----+-----+-----+
| lev1      | lev2                      | lev3      | lev4  |
+-----+-----+-----+-----+
| ELECTRONICS | PORTABLE ELECTRONICS | MP3 PLAYERS | FLASH |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Limitations of Adjacency Lists

- Adjacency list model not normalized.
- Deleting nodes can orphan children.
- Querying paths and trees requires multiple self-joins.
- Procedural code required for most use-cases.

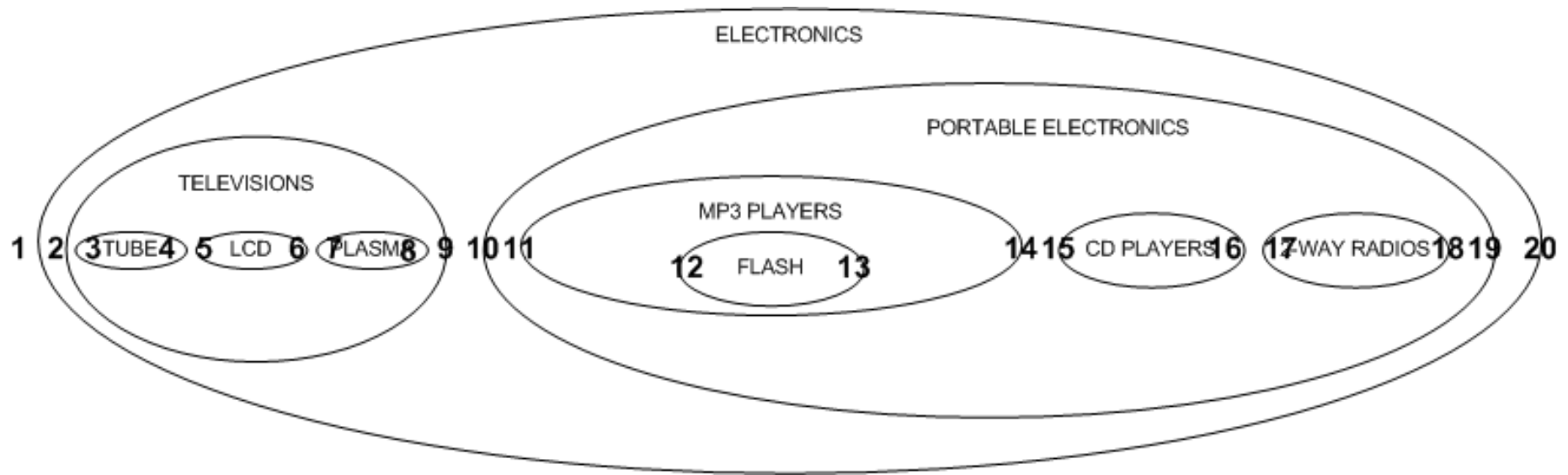
Nested Sets



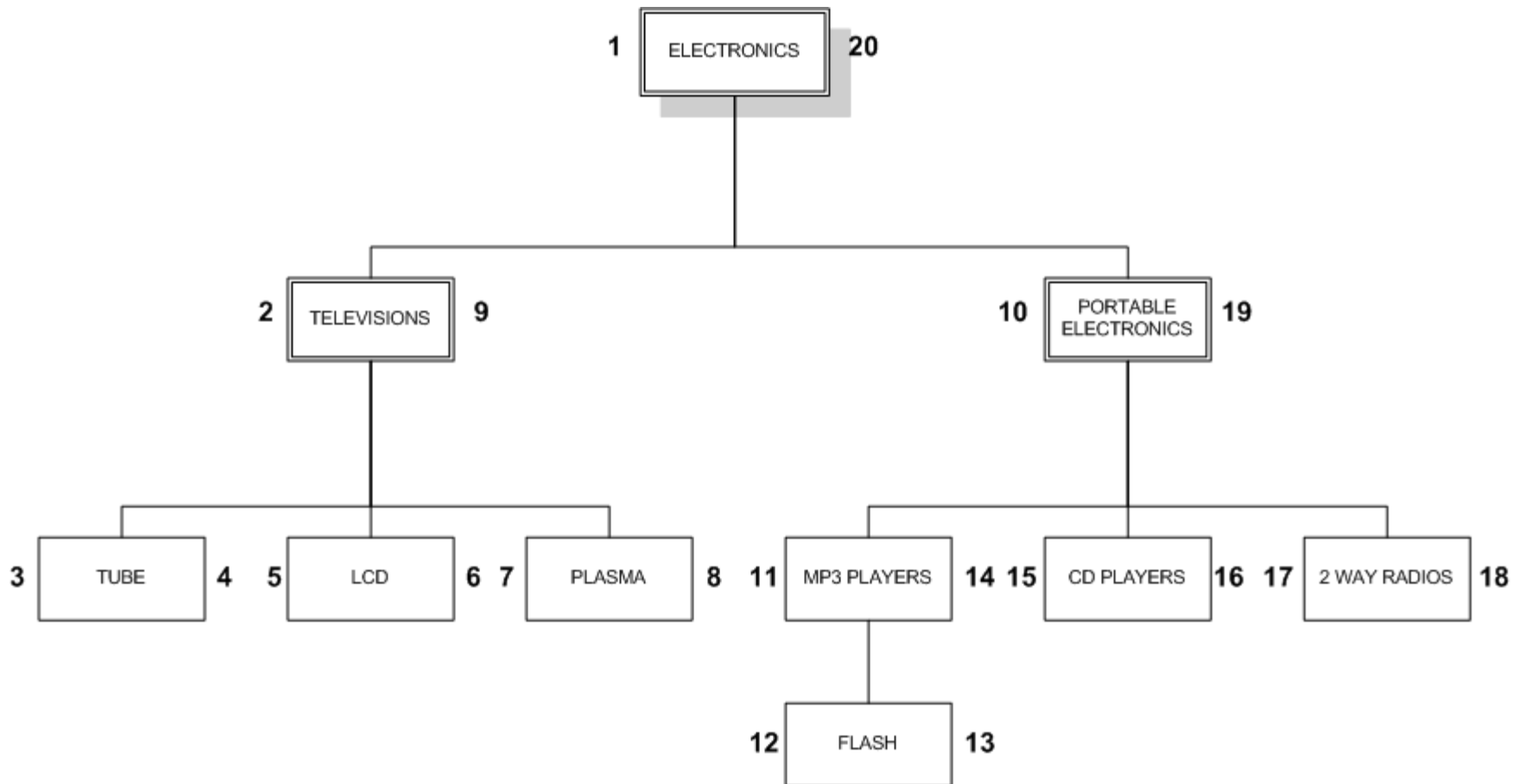
Nested Set Table

category_id	name	lft	rgt
1	ELECTRONICS	1	20
2	TELEVISIONS	2	9
3	TUBE	3	4
4	LCD	5	6
5	PLASMA	7	8

Numbered Sets



Numbered Tree



Full Tree

```
SELECT node.name FROM nested_category AS node, nested_category AS
parent
WHERE node.lft BETWEEN parent.lft AND parent.rgt
AND parent.name = 'ELECTRONICS'
ORDER BY node.lft;
```

+-----+
name
+-----+
ELECTRONICS
TELEVISIONS
TUBE
LCD
PLASMA
PORTABLE ELECTRONICS
MP3 PLAYERS
FLASH
CD PLAYERS
2 WAY RADIOS
+-----+

Leaf Nodes

```
SELECT name
FROM nested_category
WHERE rgt = lft + 1;
```

+	-----	+
	name	
+	-----	+
	TUBE	
	LCD	
	PLASMA	
	FLASH	
	CD PLAYERS	
	2 WAY RADIOS	
+	-----	+

Node Depth

```
SELECT node.name, (COUNT(parent.name) - 1) AS depth
FROM nested_category AS node, nested_category AS parent
WHERE node.lft BETWEEN parent.lft AND parent.rgt
GROUP BY node.name
ORDER BY node.lft;
```

name	depth
ELECTRONICS	0
TELEVISIONS	1
TUBE	2
LCD	2
PLASMA	2
PORTABLE ELECTRONICS	1
MP3 PLAYERS	2
FLASH	3
CD PLAYERS	2
2 WAY RADIOS	2

Indented Nodes

```
SELECT CONCAT( REPEAT(' ', COUNT(parent.name) - 1), node.name) AS name
FROM nested_category AS node, nested_category AS parent
WHERE node.lft BETWEEN parent.lft AND parent.rgt
GROUP BY node.name
ORDER BY node.lft;
```

```
+-----+
| name          |
+-----+
| ELECTRONICS    |
|   TELEVISIONS |
|     TUBE       |
|     LCD        |
|     PLASMA     |
| PORTABLE ELECTRONICS |
|   MP3 PLAYERS  |
|     FLASH      |
|   CD PLAYERS   |
|   2 WAY RADIOS |
+-----+
```

Depth of a Sub-Tree

```
SELECT node.name, (COUNT(parent.name) - (sub_tree.depth + 1)) AS
depth FROM nested_category AS node,
       nested_category AS parent,
       nested_category AS sub_parent,
       (
           SELECT node.name, (COUNT(parent.name) - 1) AS depth
           FROM nested_category AS node,
                nested_category AS parent
           WHERE node.lft BETWEEN parent.lft AND parent.rgt
           AND node.name = 'PORTABLE ELECTRONICS'
           GROUP BY node.name ORDER BY node.lft
       ) AS sub_tree
WHERE node.lft BETWEEN parent.lft AND parent.rgt
      AND node.lft BETWEEN sub_parent.lft AND sub_parent.rgt
      AND sub_parent.name = sub_tree.name
GROUP BY node.name ORDER BY node.lft;
```

Depth of a Sub-Tree - II

name	depth
PORTABLE ELECTRONICS	0
MP3 PLAYERS	1
FLASH	2
CD PLAYERS	1
2 WAY RADIOS	1

Immediate Subordinates

```
GROUP BY node.name  
HAVING depth <= 1  
ORDER BY node.lft;
```

name	depth
PORTABLE ELECTRONICS	0
MP3 PLAYERS	1
CD PLAYERS	1
2 WAY RADIOS	1

Aggregate Functions

```
SELECT * FROM product;
```

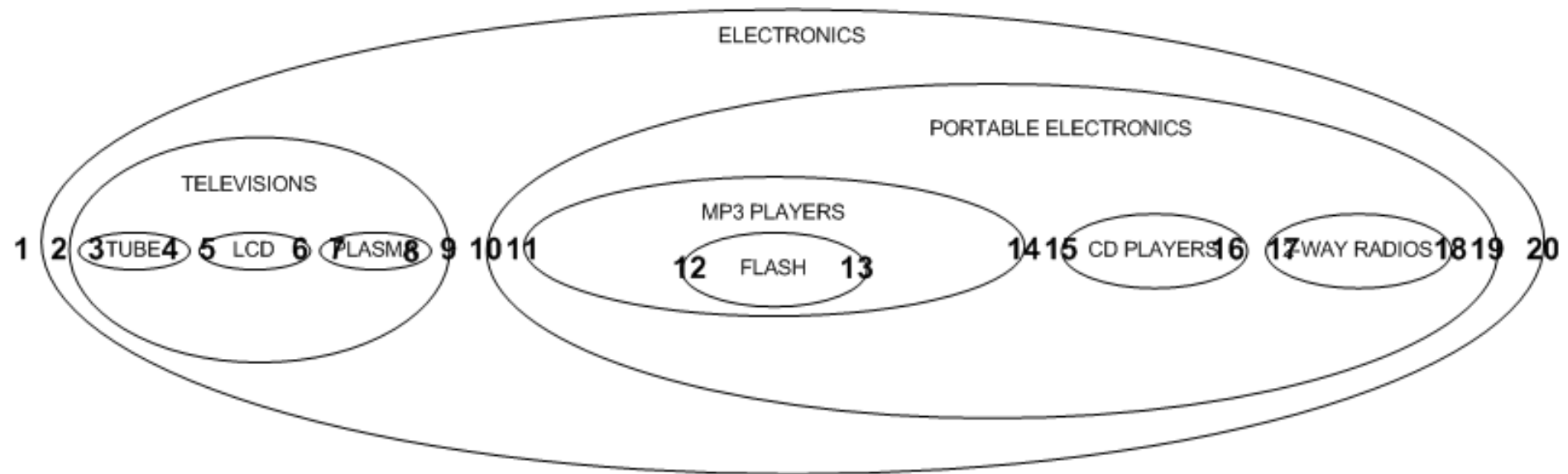
product_id	name	category_id
1	20" TV	3
2	36" TV	3
3	Super-LCD 42"	4
4	Ultra-Plasma 62"	5
5	Value Plasma 38"	5
6	Power-MP3 128mb	7
7	Super-Shuffle 1gb	8
8	Porta CD	9
9	CD To go!	9
10	Family Talk 360	10

Aggregate Functions - II

```
SELECT parent.name, COUNT(product.name)
FROM nested_category AS node,
nested_category AS parent, product
WHERE node.lft BETWEEN parent.lft AND parent.rgt
AND node.category_id = product.category_id
GROUP BY parent.name ORDER BY node.lft;
```

name	COUNT(product.name)
ELECTRONICS	10
TELEVISIONS	5
TUBE	2
LCD	1
PLASMA	2
PORTABLE ELECTRONICS	5
MP3 PLAYERS	2
FLASH	1
CD PLAYERS	2
2 WAY RADIOS	1

Adding Nodes



Adding a Sibling

```
LOCK TABLE nested_category WRITE;
```

```
SELECT @myRight := rgt
FROM nested_category
WHERE name = 'TELEVISIONS';
```

```
UPDATE nested_category
SET rgt = rgt + 2
WHERE rgt > @myRight;
```

```
UPDATE nested_category
SET lft = lft + 2
WHERE lft > @myRight;
```

```
INSERT INTO nested_category(name, lft, rgt)
VALUES('GAME CONSOLES', @myRight + 1, @myRight + 2);
```

```
UNLOCK TABLES;
```

name	
ELECTRONICS	
TELEVISIONS	
TUBE	
LCD	
PLASMA	
GAME CONSOLES	
PORTABLE ELECTRONICS	
MP3 PLAYERS	
FLASH	

Adding a Child

```
LOCK TABLE nested_category WRITE;
```

```
SELECT @myLeft := lft
FROM nested_category
WHERE name = '2 WAY RADIOS';
```

```
UPDATE nested_category
SET rgt = rgt + 2
WHERE rgt > @myLeft;
```

```
UPDATE nested_category
SET lft = lft + 2
WHERE lft > @myLeft;
```

```
INSERT INTO nested_category(name, lft, rgt)
VALUES('FRS', @myLeft + 1, @myLeft + 2);
```

```
UNLOCK TABLES;
```

name
ELECTRONICS
TELEVISIONS
TUBE
LCD
PLASMA
GAME CONSOLES
PORTABLE ELECTRONICS
MP3 PLAYERS
FLASH
CD PLAYERS
2 WAY RADIOS
FRS

Deleting a Leaf Node

```
LOCK TABLE nested_category WRITE;
```

```
SELECT @myLeft := lft, @myRight := rgt, @myWidth := rgt -  
lft + 1  
FROM nested_category  
WHERE name = 'GAME CONSOLES';
```

```
DELETE FROM nested_category  
WHERE lft BETWEEN @myLeft AND @myRight;
```

```
UPDATE nested_category  
SET rgt = rgt - @myWidth  
WHERE rgt > @myRight;
```

```
UPDATE nested_category  
SET lft = lft - @myWidth  
WHERE lft > @myRight;
```

```
UNLOCK TABLES;
```

name
ELECTRONICS
TELEVISIONS
TUBE
LCD
PLASMA
PORTABLE ELECTRONICS
MP3 PLAYERS
FLASH
CD PLAYERS
2 WAY RADIOS
FRS

Deleting a Parent and all Children

```
LOCK TABLE nested_category WRITE;
```

```
SELECT @myLeft := lft, @myRight := rgt, @myWidth := rgt - lft + 1
FROM nested_category
WHERE name = 'MP3 PLAYERS';
```

```
DELETE FROM nested_category
WHERE lft BETWEEN @myLeft AND @myRight;
```

```
UPDATE nested_category
SET rgt = rgt - @myWidth
WHERE rgt > @myRight;
```

```
UPDATE nested_category
SET lft = lft - @myWidth
WHERE lft > @myRight;
```

```
UNLOCK TABLES;
```

+	-----	+
	name	
+	-----	+
	ELECTRONICS	
	TELEVISIONS	
	TUBE	
	LCD	
	PLASMA	
	PORTABLE ELECTRONICS	
	CD PLAYERS	
	2 WAY RADIOS	
	FRS	
+	-----	+

Deleting a Parent and Promoting all Children

```
LOCK TABLE nested_category WRITE;
```

```
SELECT @myLeft := lft, @myRight := rgt, @myWidth := rgt - lft + 1
FROM nested_category
WHERE name = 'PORTABLE ELECTRONICS';
```

```
DELETE FROM nested_category WHERE lft = @myLeft;
```

```
UPDATE nested_category
SET rgt = rgt - 1, lft = lft - 1
WHERE lft BETWEEN @myLeft AND @myRight;
```

```
UPDATE nested_category
SET rgt = rgt - 2
WHERE rgt > @myRight;
```

```
UPDATE nested_category
SET lft = lft - 2
WHERE lft > @myRight;
```

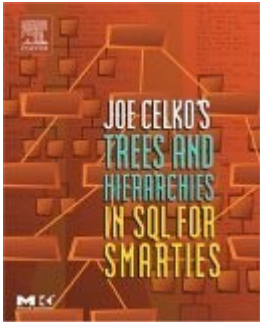
```
UNLOCK TABLES;
```

+-----+
name
+-----+
ELECTRONICS
TELEVISIONS
TUBE
LCD
PLASMA
CD PLAYERS
2 WAY RADIOS
FRS
+-----+

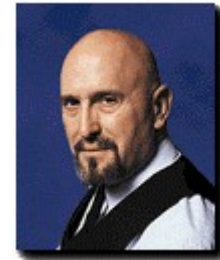
Advanced Use Cases

- Move an entire sub-tree
- Handle high insert/delete volumes
 - Calculate gaps
 - Use floats
 - See the resources next slide

Credit & Resources



Joe Celko's Trees and Hierarchies in SQL for Smarties



<http://www.openwin.org/mike/books/>

<http://www.vbmysql.com/articles/managing-hierarchical-data-in-mysql.html>

<http://www.sitepoint.com/article/hierarchical-data-database>

<http://www.dbmsmag.com/9603d06.html>

<http://www.dbmsmag.com/9604d06.html>

<http://www.dbmsmag.com/9605d06.html>

<http://www.intelligententerprise.com/001020/celko.jhtml>

http://searchdatabase.techtarget.com/tip/1,289483,sid13_gci537290,00.html

http://searchdatabase.techtarget.com/tip/1,289483,sid13_gci801943,00.html

http://www.klempert.de/php/nested_sets_demo

<http://dev.e-taller.net/dbtree/>

Conclusion

- <http://www.openwin.org/mike/presentations/>
- Questions?
- Stick around for the book giveaway!

Book Draw!

Stick around and win a book!