

Tree storage and retrieval

What do we want from this API?

1. Get a list of parents
2. Find a subtree
3. Move subtrees
4. Show a tree in tree order

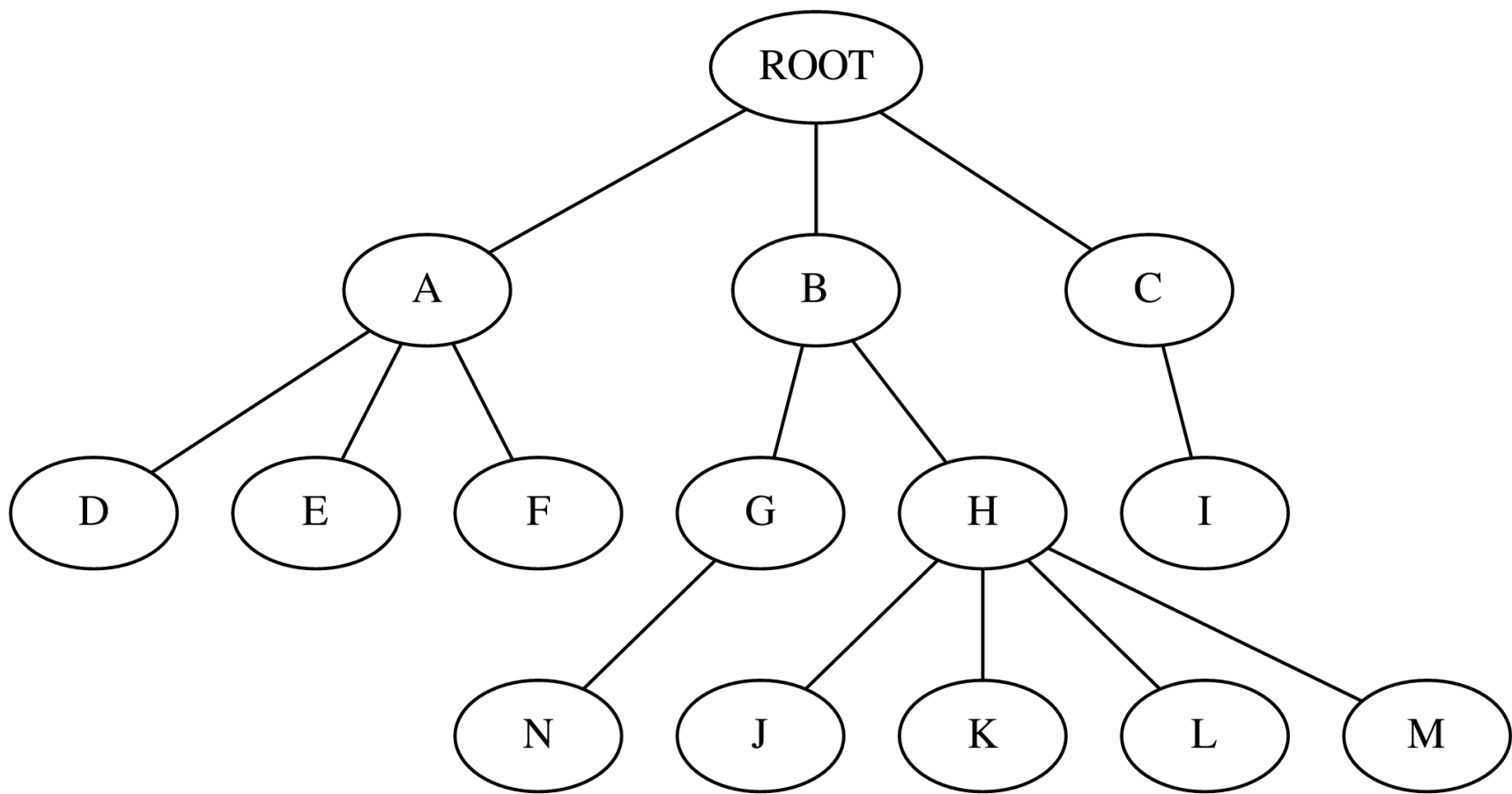
Traditional storage
Node identifier, Parent node identifier

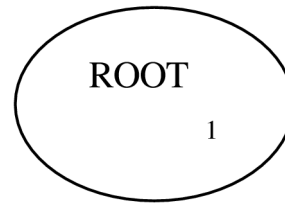
Does it work?

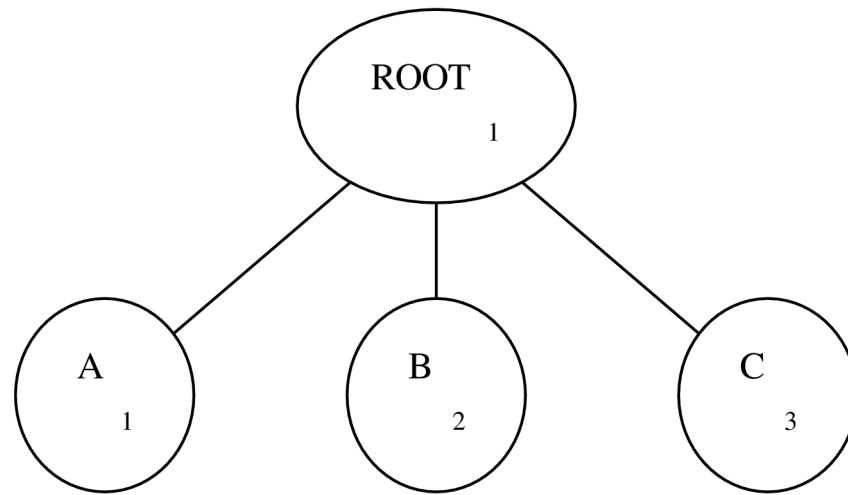
1. Get a list of parents – depth number of queries
2. Find a subtree – recursive
3. Move subtrees – not a problem
4. Show a tree in tree order – recursive

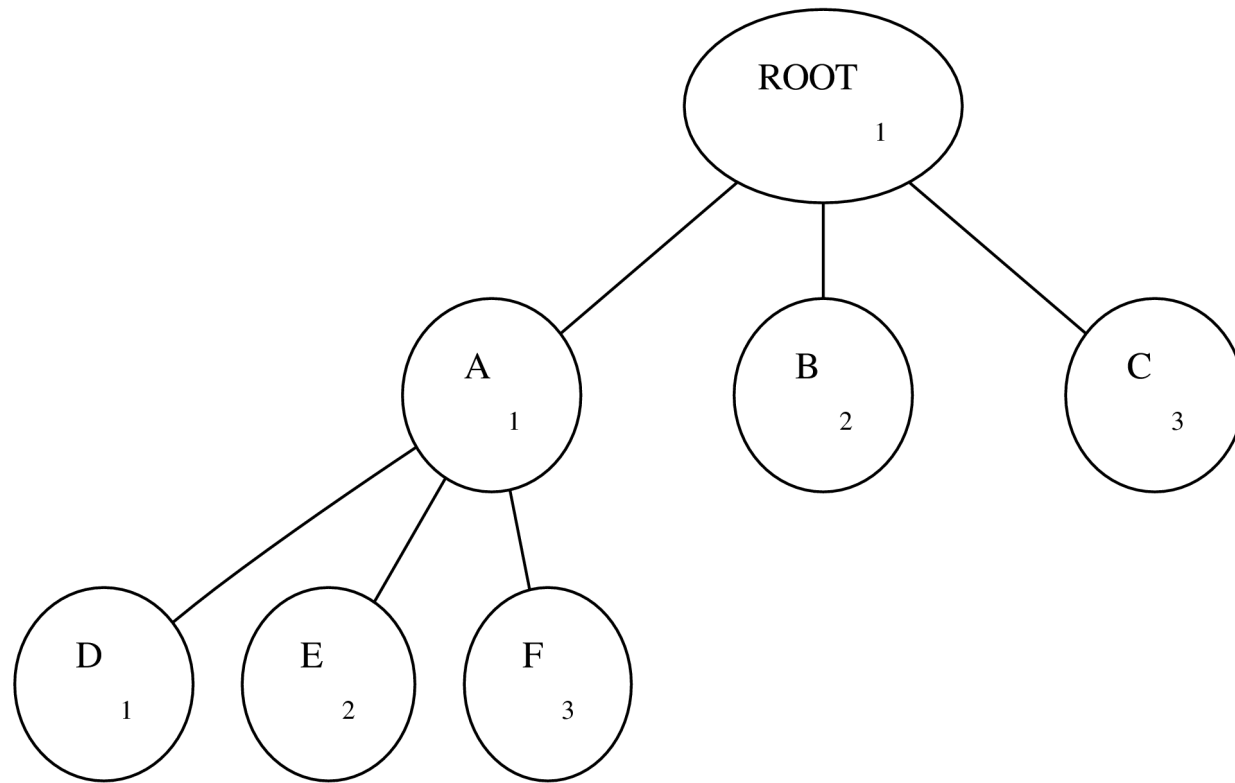
SQL sucks at recursion

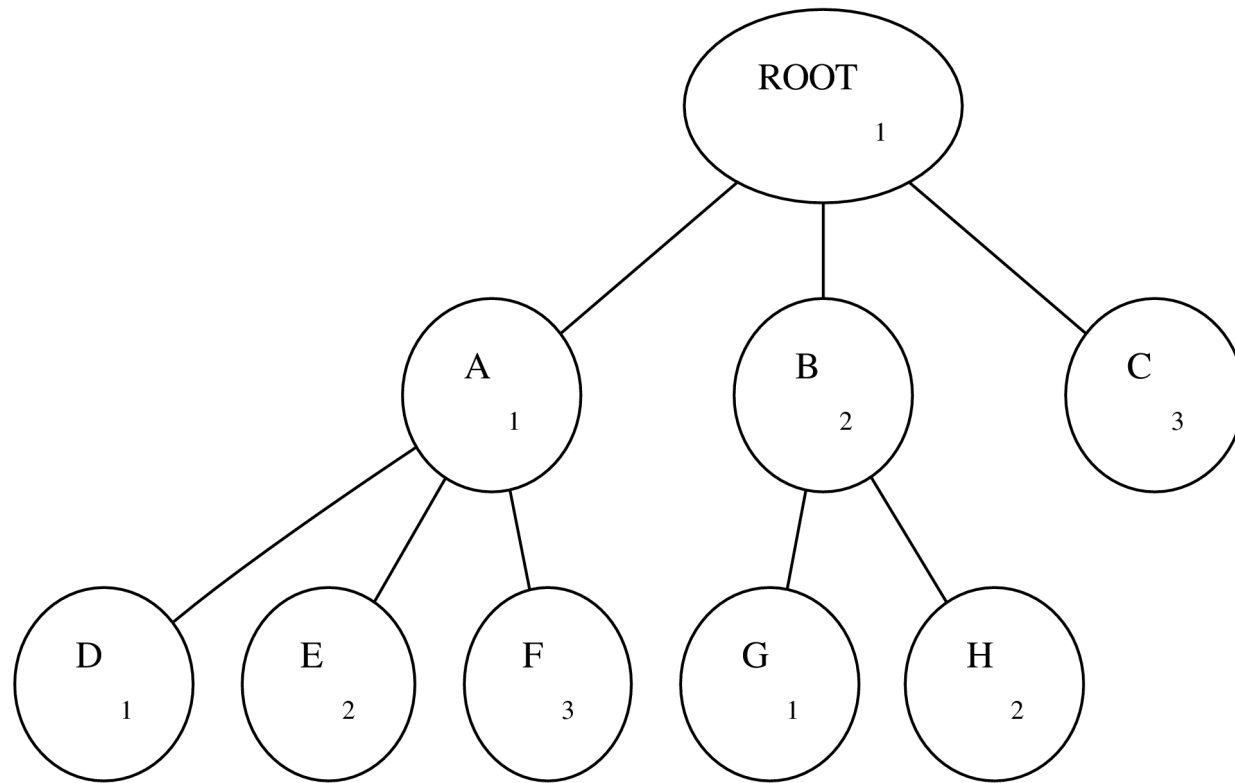
Materialized path

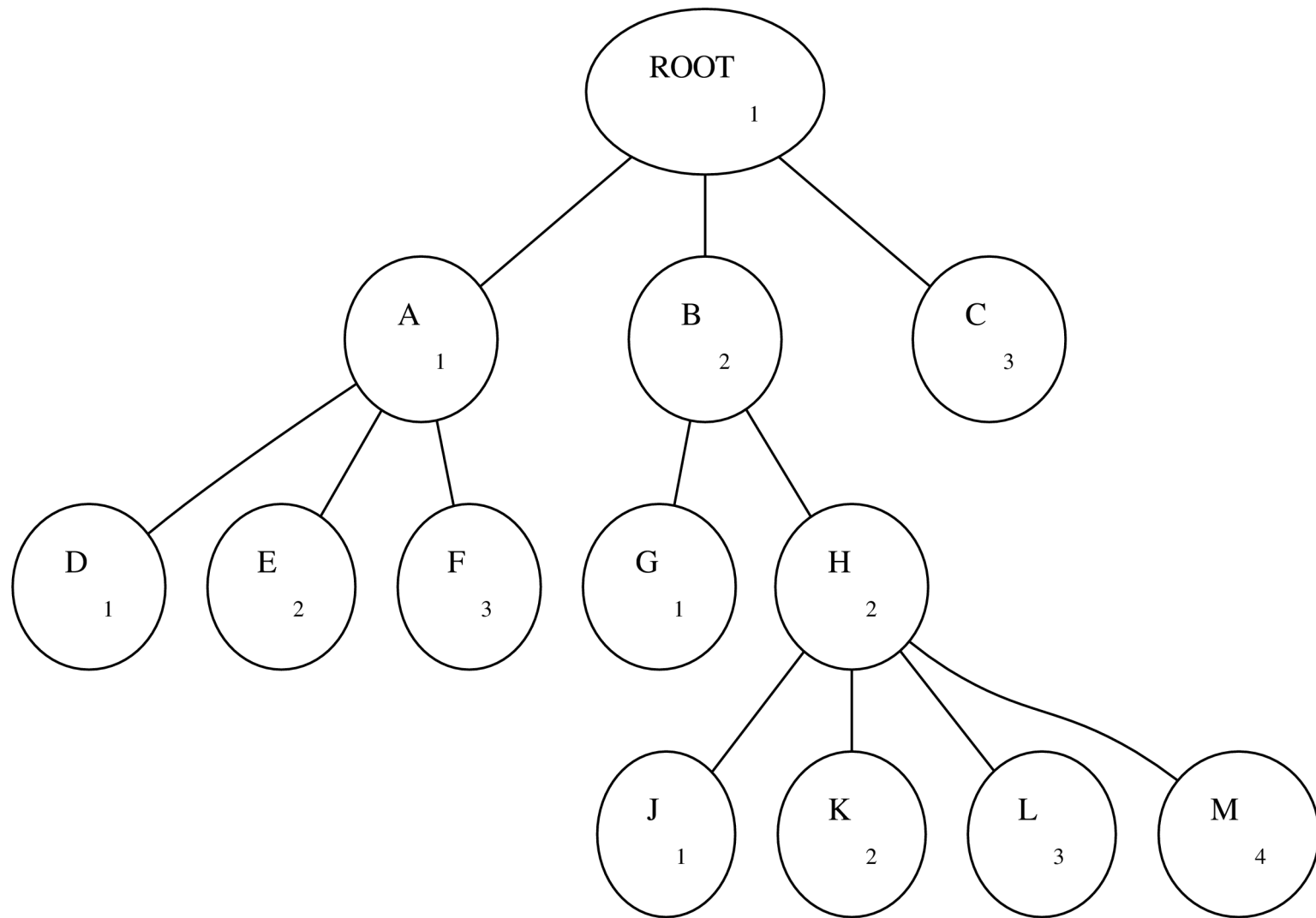


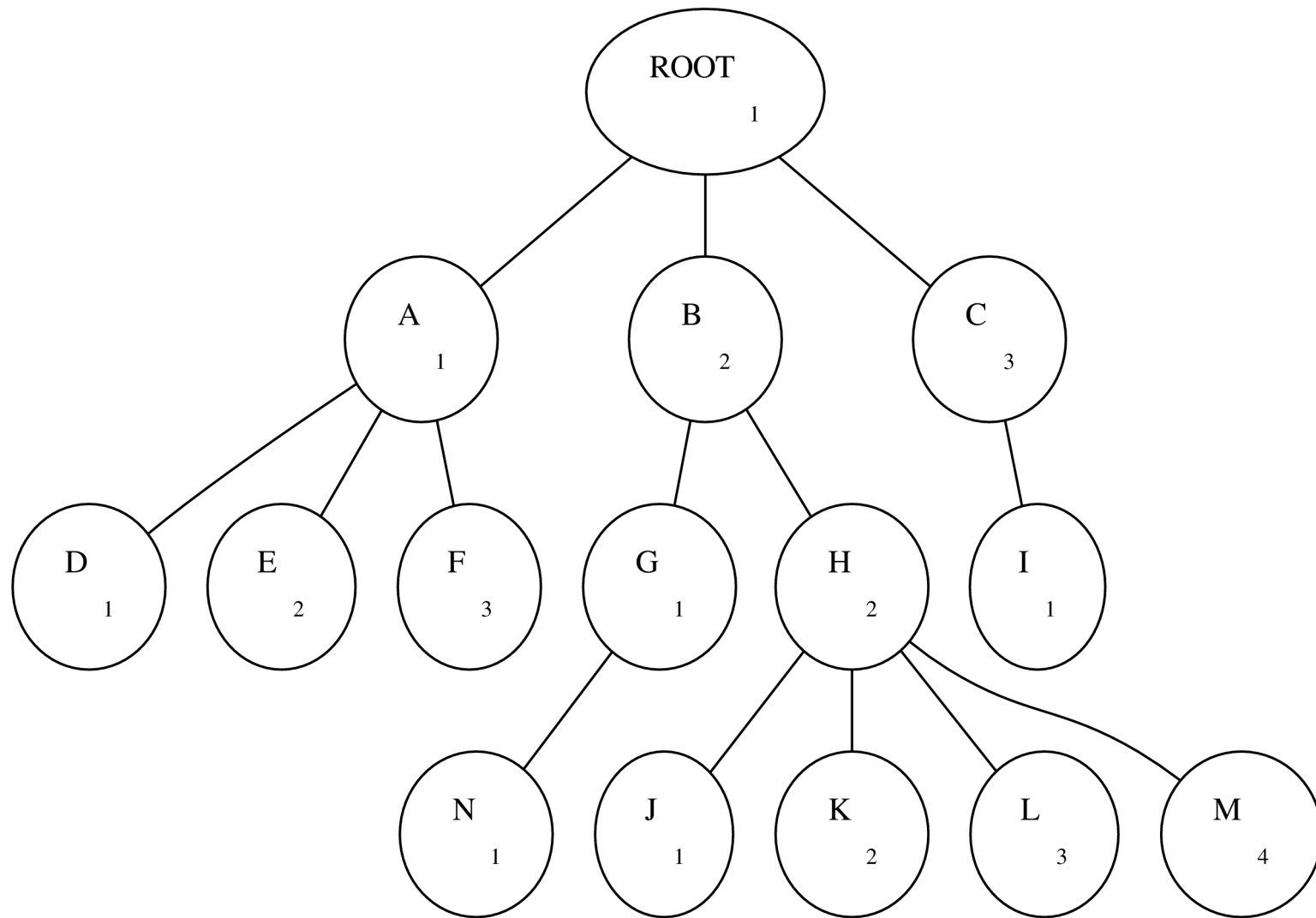


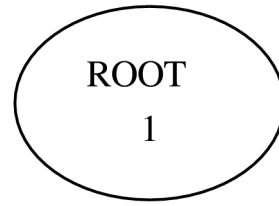


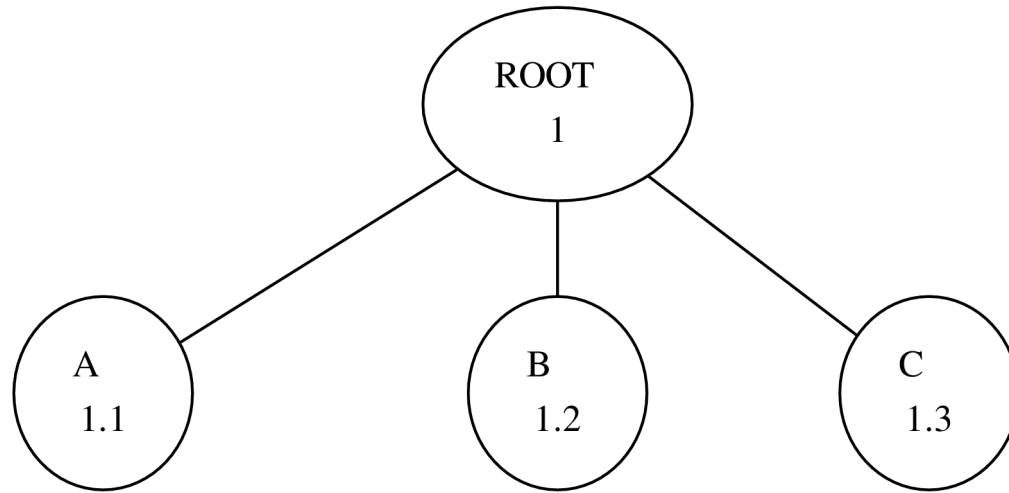


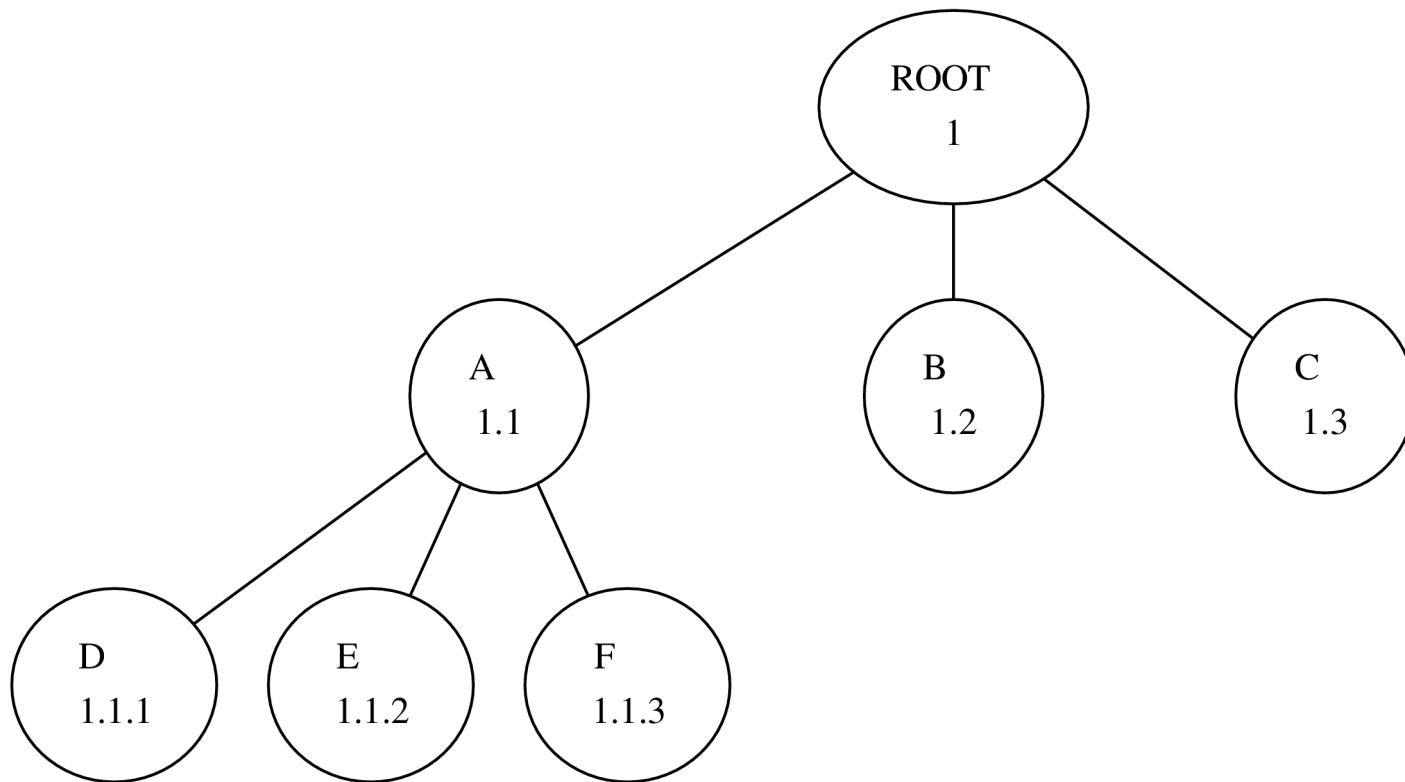


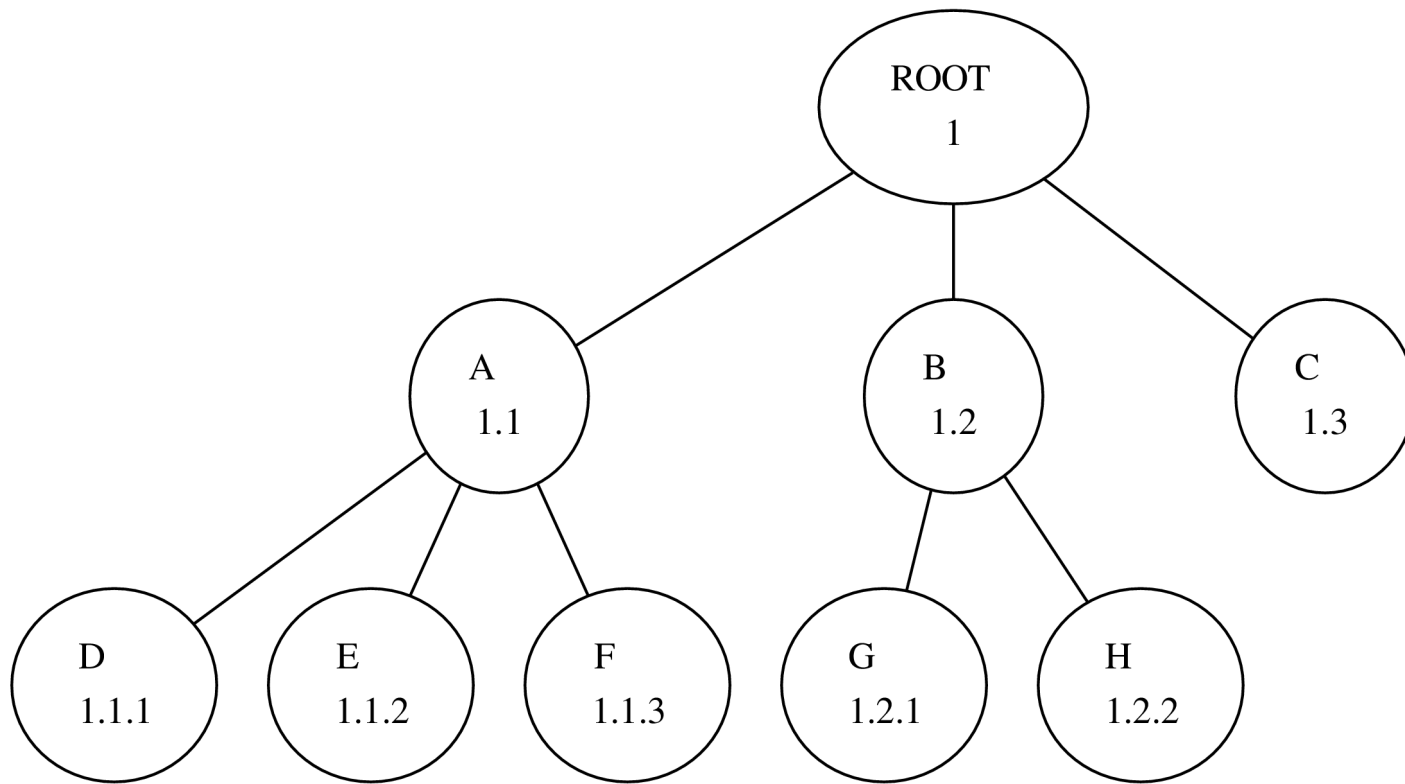


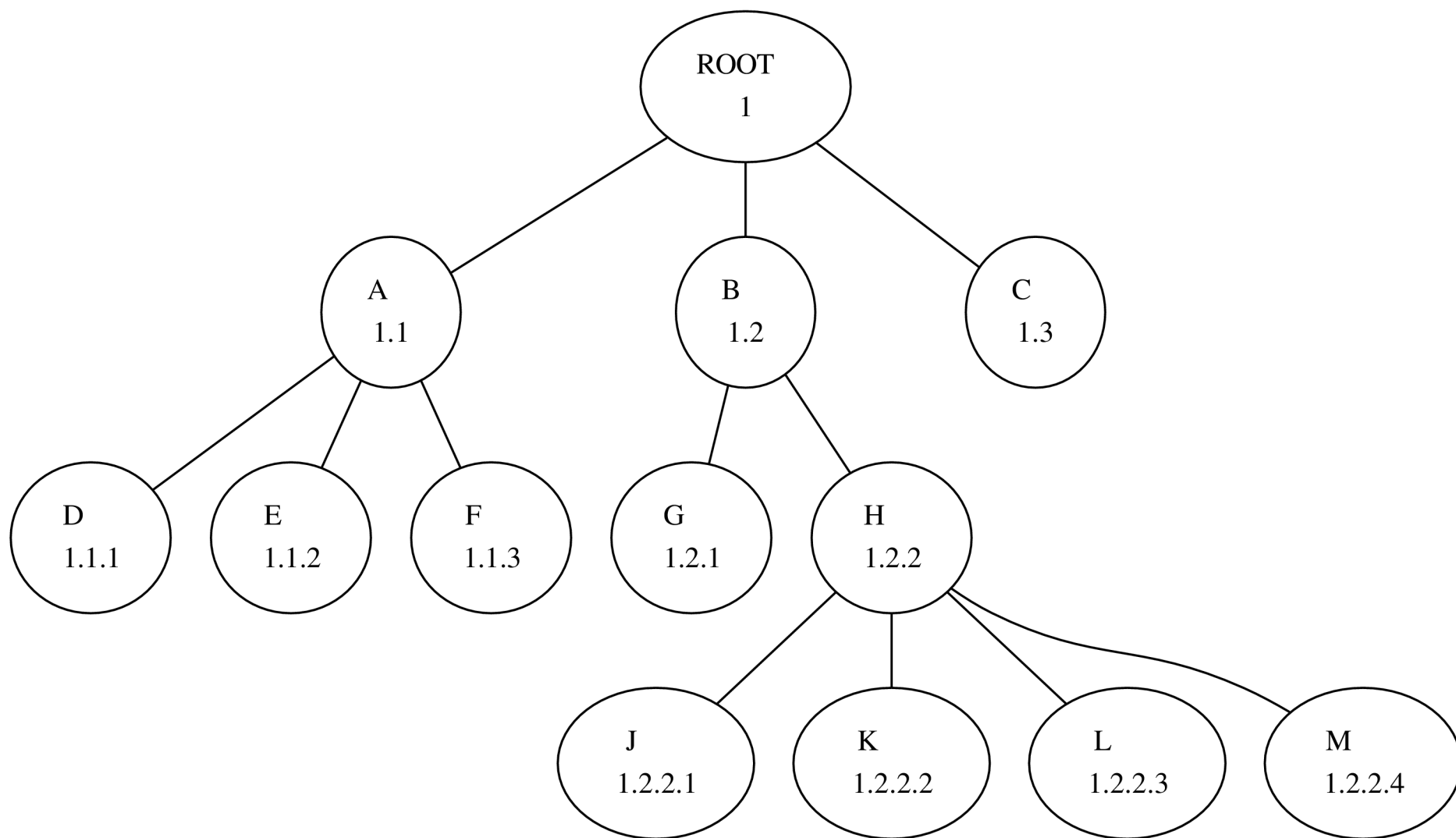


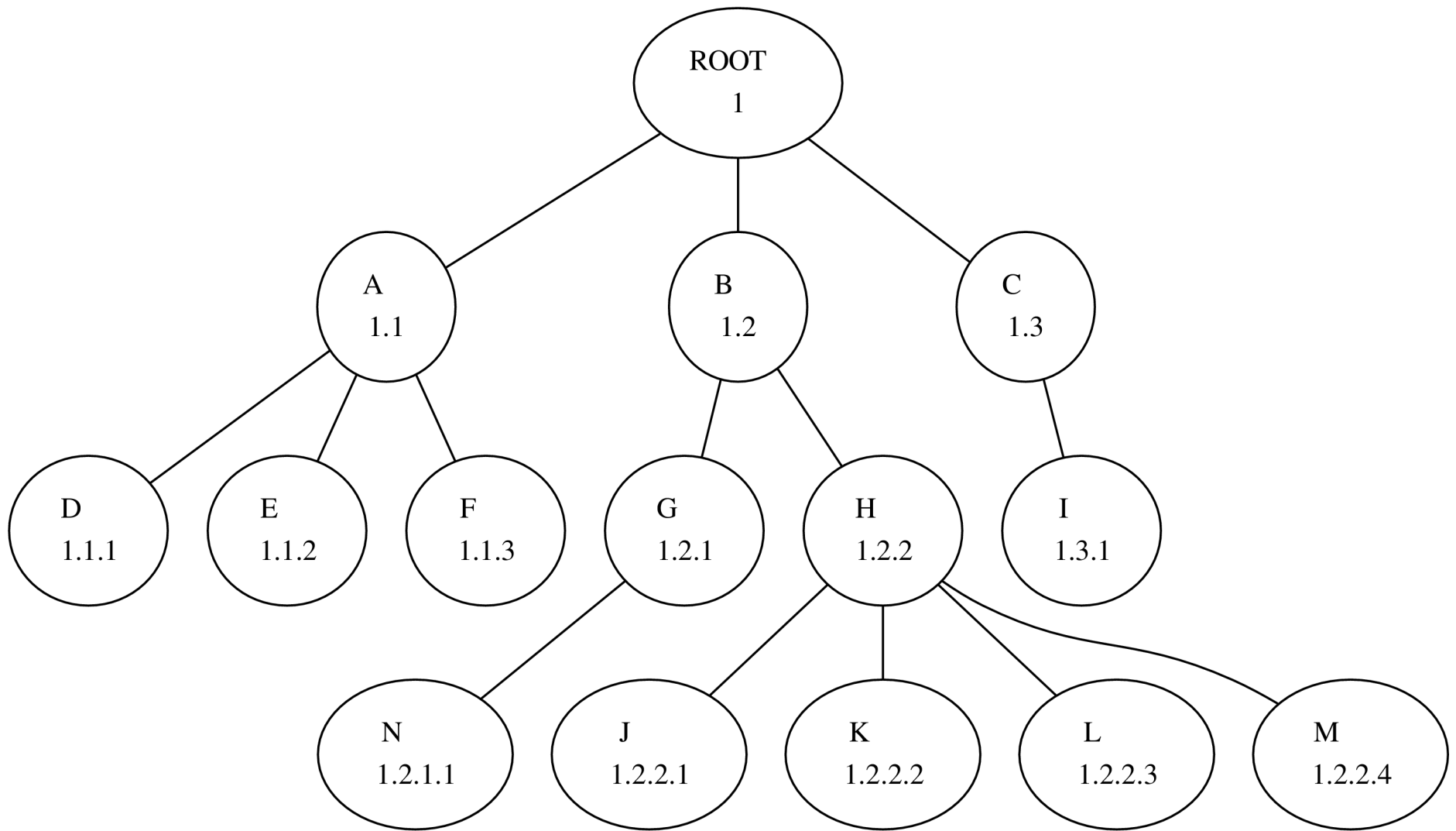












1.12.1

1.2

1.12.1

1.2

The trick in core:
Add length

11.212.11
11.12

This works!

List of parents for node 11.212.257

- 11.212
- 11
- Very easy!

Subtree of 11.212.257

- `SELECT * FROM tree WHERE encoding LIKE '11.212.257.%'`
- Still very easy.
- And the query is indexable.

Move 11.212.257 under 13

- Find the max child of 13, let's say it's 3456
- 11.212.257 becomes 13.3457
- UPDATE tree SET encoding =
 REPLACE('encoding', '11.212.257', '13.3457')
 WHERE encoding LIKE '11.212.257'
- Doable.
- Note: adding a 0. to anchor is advised.

Subtree of 11.212.257 ordered

- `SELECT * FROM tree WHERE encoding LIKE '11.212.257.%' ORDER BY encoding`
- Still very easy.
- And the query is indexable.

We cheated a little

- The materialized path is numbers: 1, 376...
- But instead we store a string: 11.3376
- Storage is ASCII encoding: 49 49 46 51 51 55 54
- Sort holds:
 - $\text{ASCII}(1) = 49$
 - $\text{ASCII}(2) = 50$
 - $49 < 50$.

Core uses base 36 instead of base 10

That's still good

- The number 376 becomes AG
- We add the length: 2AG
- Store this binary data:
50 65 71
- $20 < 21$ is held because 20 becomes K, 21 becomes J and their ASCII encoding is $75 < 76$

We could use UTF-8 encoding instead

UTF-8 encoding

Bits	Number range	Byte 1	Byte 2	Byte 3
7	0000–007F	0xxxxxxx		
11	0080–07FF	110xxxxx	10xxxxxx	
16	0800–FFFF	1110xxxx	10xxxxxx	10xxxxxx

- We store 376 as 197 184
- Two numbers in the same range are compared using the same bits as if they were integers
- Different ranges are determined by first byte

Advantages

- No need for a separator character.
- UTF-8 is quite space efficient.

Disadvantages

- Max number of children is about two billion
- MySQL index size limits the tree depth (usually to 255, can be more)