Clustered Index Internals

Kimberly L. Tripp
OWNER/PRESIDENT – SQLSKILLS.COM
@kimberlyltripp www.sqlskills.com/blogs/kimberly

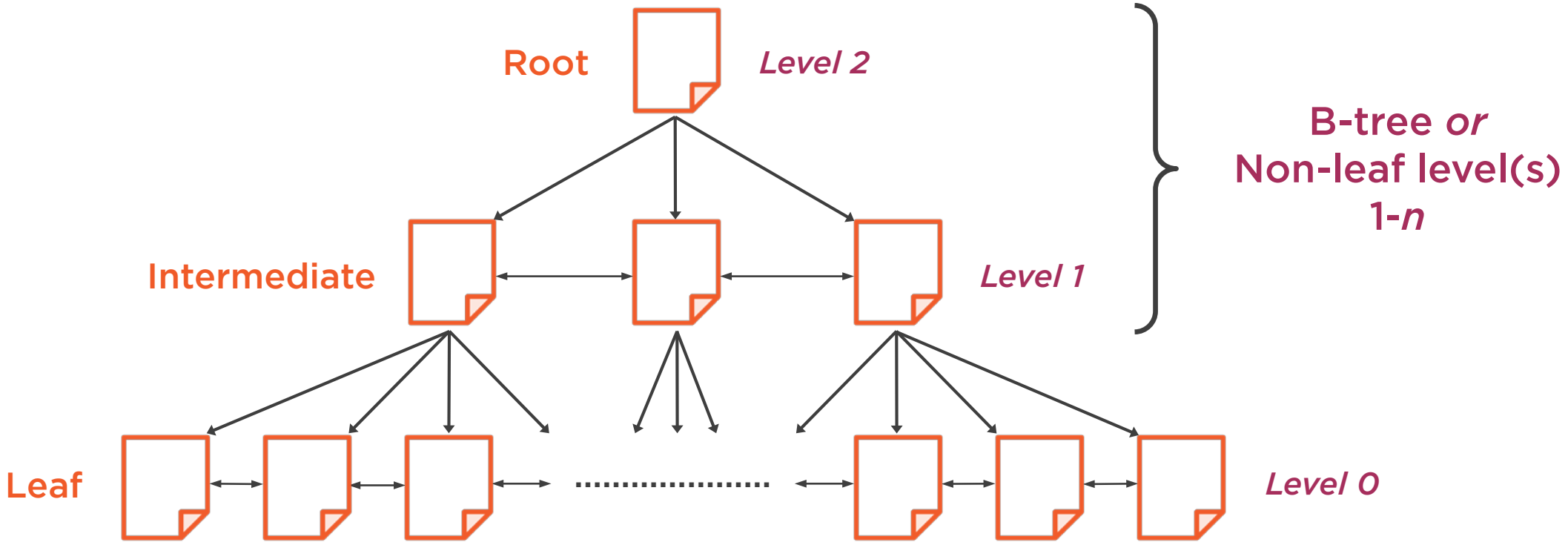
Module Overview

Index levels

Case study

- Building the leaf level
- Building the tree structure

Physical Index Levels



Leaf level: contains something for every row of the table in indexed order

Non-leaf level(s) : contains something, specifically representing FIRST value, from every page of level below (intermediate levels are not a certainty)

Employee Table Case Study

Clustering the Employee table on EmployeeID (identity)

- Investigate the data
- Physically order data
- Add the tree structure
- Complete the math
- Complete the B-tree

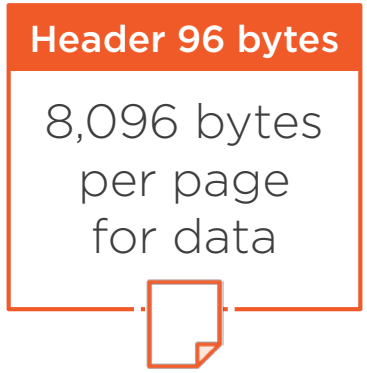
Nonclustered unique constraint for SSN

- In the next module...

Investigating the Data

Average row size
= 400 bytes

```
CREATE TABLE [Employee]
(
[EmployeeID]      int          NOT NULL identity,
[LastName]        nchar(30)    NOT NULL,
[FirstName]       nchar(29)    NOT NULL,
[MiddleInitial]   nchar(1)     NULL,
[SSN]             char(11)     NOT NULL,
…other columns…)
```



$$\frac{\text{8,096 bytes / page}}{\text{400 bytes / row}} = \textbf{20 rows/page}$$

80,000 current employees = 80,000 rows

$$\frac{\text{80,000 employees}}{\text{20 rows / page}} = \textbf{4,000 pages}$$

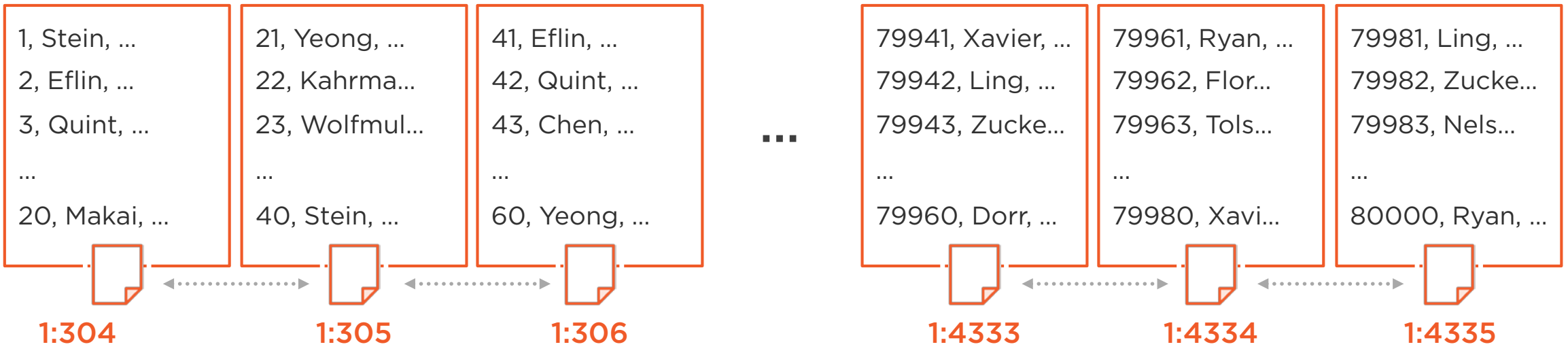
Clustered Employee Table

Step 1: **Physically order data**

Review the index level definitions... Does this seem to match one of the definitions?

Yes! When a table is clustered the data becomes the leaf level of the clustered index!

4,000 pages of Employees in clustering key (EmployeeID) order



Clustered Employee Table

Step 2: **Add the tree structure**

Starting from the leaf level and going up to a root of 1 page

B-tree entry = index key value + pointer + row overhead

Pointer = page pointer of 6 bytes = 2 for fileID + 4 for pageID

Row overhead varies based on many factors (min of 1 byte in the row)

Non-leaf level entry for clustered index on EmployeeID = 11

4 bytes for EmployeeID (int) + 6 bytes for page pointer + 1 byte for row overhead

$$\frac{\text{8,096 bytes / page}}{\text{11 bytes/entry + 2 bytes in slot array}} = \textbf{622 index entries per non-leaf level page}$$

How many entries to store? 4,000

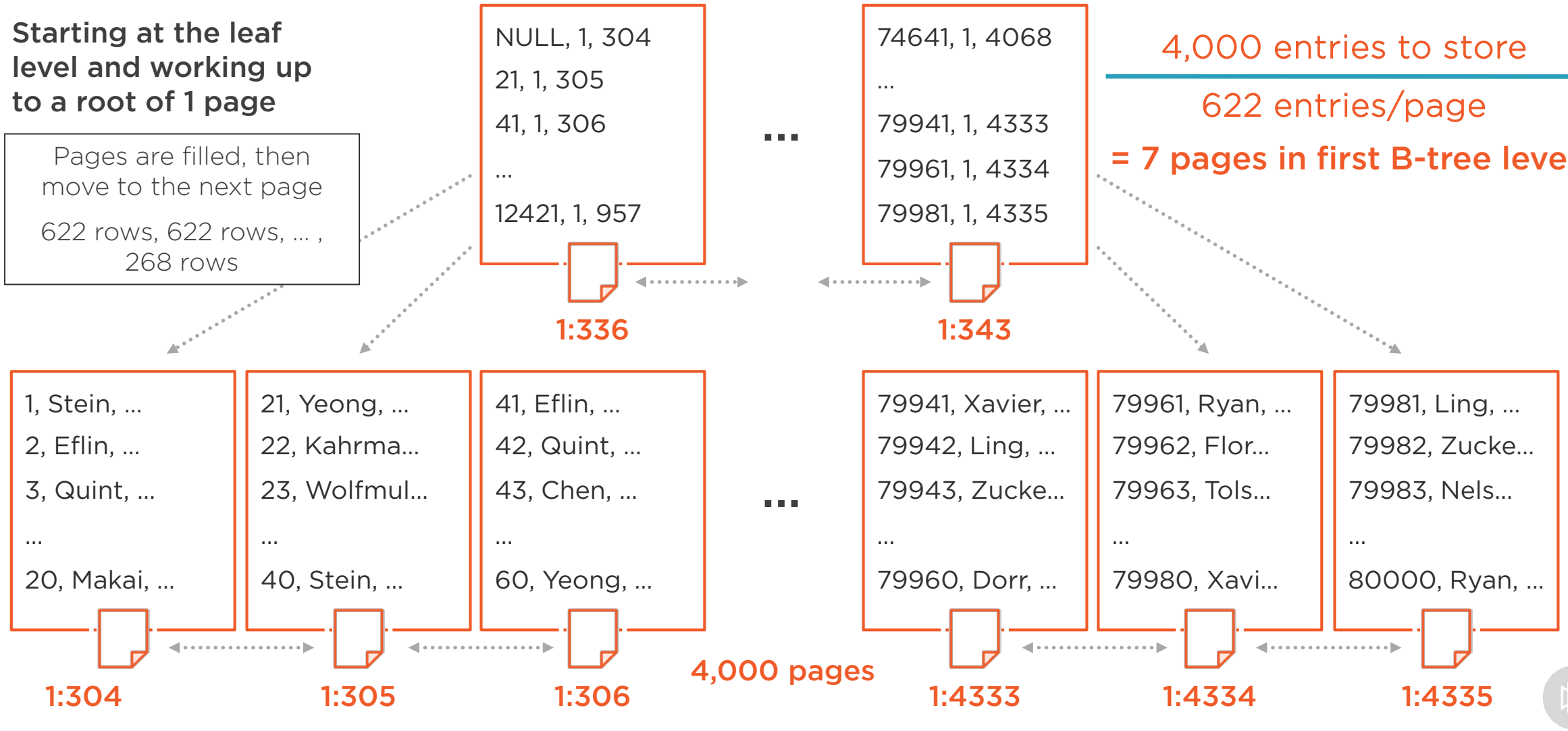
Clustered Employee Table

Step 2: **Add the tree structure**

Starting at the leaf level and working up to a root of 1 page

Pages are filled, then move to the next page

622 rows, 622 rows, … , 268 rows



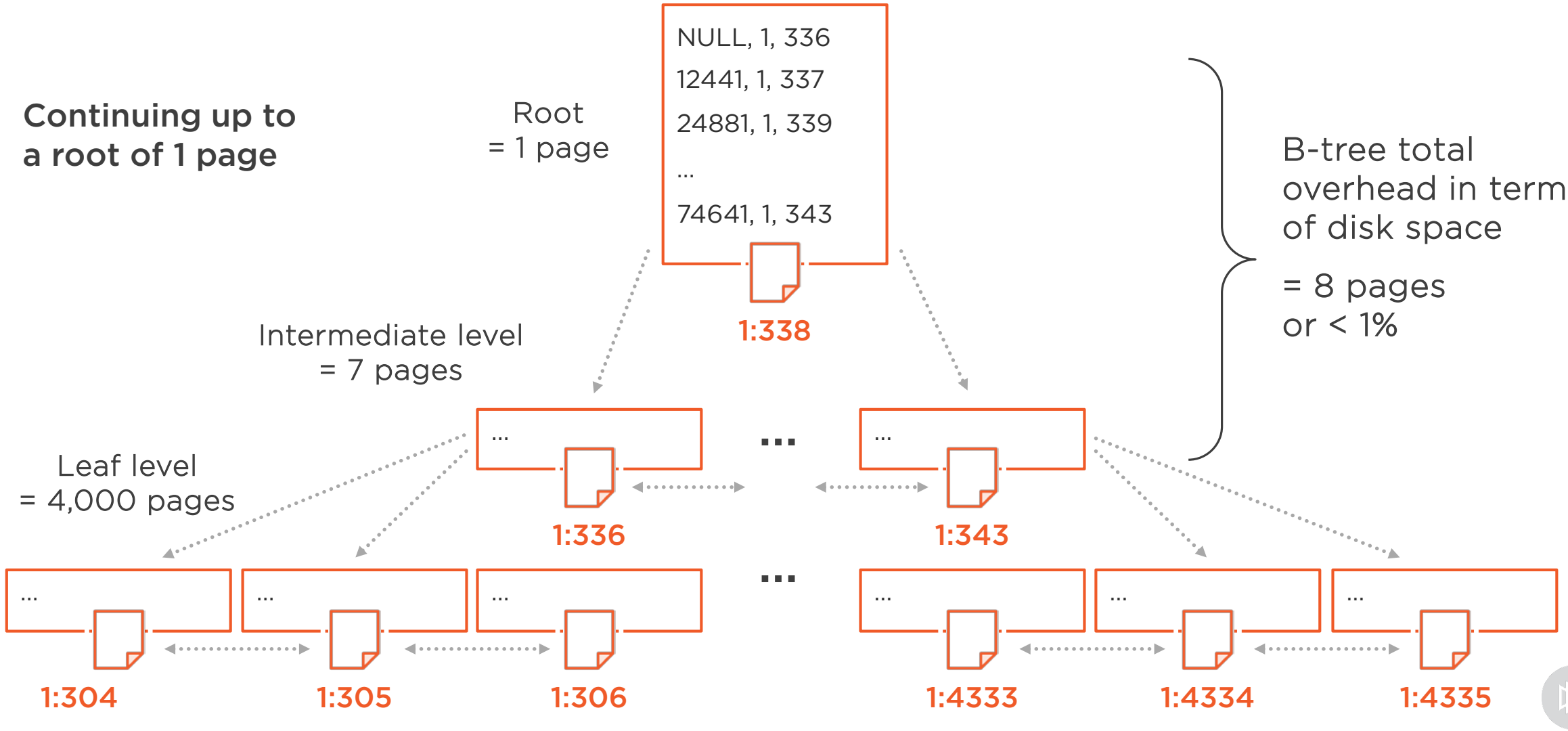
4,000 entries to store

622 entries/page

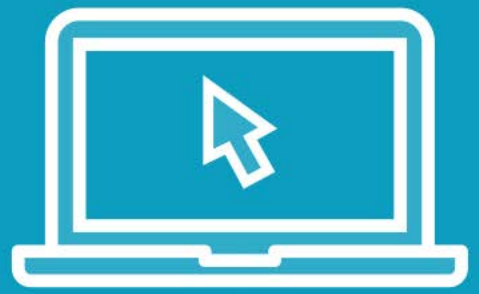
= 7 pages in first B-tree level

Clustered Employee Table

Step 2: **Add the tree structure**



Demo



Clustered index internals

What We Covered

Index levels

Case study

- Building the leaf level
- Building the tree structure