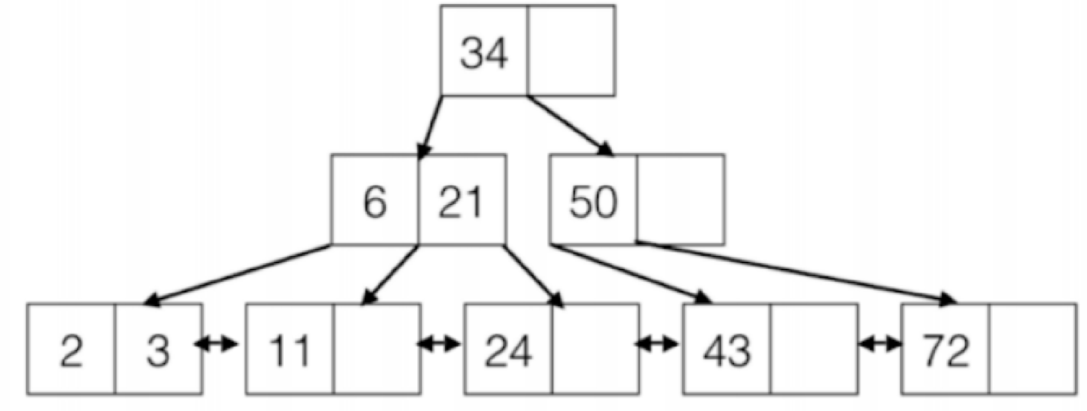
CS W186 Introduction to Database Systems
Fall 2019 Josh Hug

# DIS 3

## 1 Indices (B+ Trees)

Assume we have the following B+ Tree of order 1. Each index node must have either 1 or 2 keys (2 or 3 pointers), and the leaf nodes can hold up to 2 entries.



(a) What is the maximum number of insertions we can do without changing the height of the tree?

(b) What is the minimum number of keys you could insert to change the height of the tree?

## 2 Indices

**Two sets of terminology:**

*Clustered vs. unclustered*

- In a clustered index the index field specifies the sequential order of the table file.
- In an unclustered index the table file is not ordered by the index field.
- One consequence of this is that range queries on a unclustered index are much more inefficient than those on a clustered index.

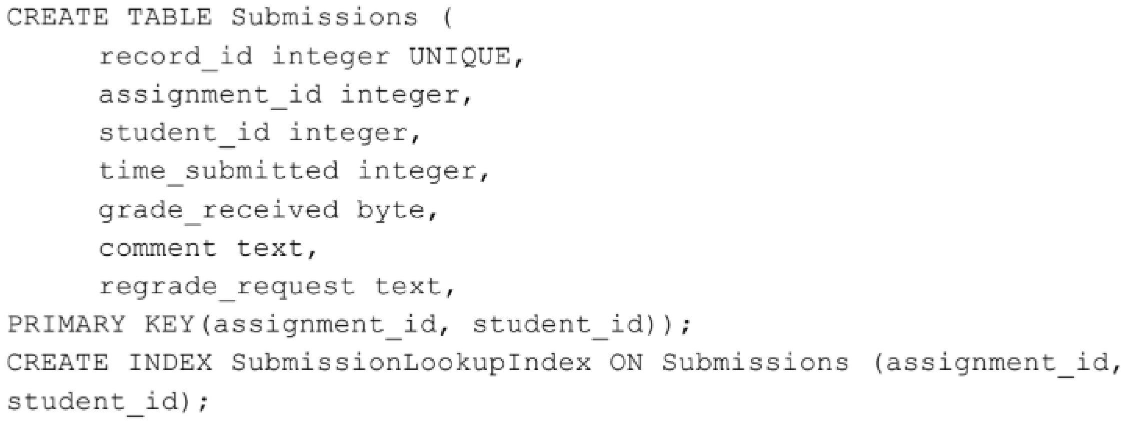
*Three alternatives for storing underlying data: alternatives 1, 2, and 3.*

- (A1) By value: entire row in leaf
- (A2) By reference: (key, rid) pairs, with each key occurring once
- (A3) By reference: (key, [rid1, rid2, ...]) pairs

1. Is it possible to have two clustered indices on separate columns?

2. Suppose we have an alternative 2 unclustered index on (assignment_id, student_id) with a height of 3 (one must traverse 3 index pages to reach any leaf page).

   Here's the schema:



```
CREATE TABLE Submissions (
     record_id integer UNIQUE,
     assignment_id integer,
     student_id integer,
     time_submitted integer,
     grade_received byte,
     comment text,
     regrade_request text,
PRIMARY KEY(assignment_id, student_id));
CREATE INDEX SubmissionLookupIndex ON Submissions (assignment_id,
student_id);
```

   Assume the table and its associated data takes up 12 MB on disk (1 MB = 1024 KB) and that page size is 64 KB. (This includes extra space allocated for future insertions.)

   (a) We want to scan all the records in Submissions. How many I/Os will this operation take?

   (b) UPDATE Students SET grade_received=85 WHERE assignment_id=20 AND student_id=12345; How many I/Os will this operation take?

   (c) In the worst case, how many I/Os does it take to perform an equality search on grade_received?