LAB 9

Samantha Pope

PART 1 - SELECTING INDEXES

A database contains the following table for former-employee records:

1. Index on start date: 'create index idx_start_date ON EmployeeRecords(Start Date);'

    Why? So we can filter by date employees started faster

2. Make index on start date and end date: 'create index on idx_start_end_date ON EmployeeRecords(Start Date, End Date):'

    Why? So we can find start and end date criteria that are specific. (like bonus for ppl who worked a certain amount or total time worked/etc

A database contains the following table for tracking student grades in classes

1. Make index at grade → 'create index idx_grade ON StudentsGrades(Grade);'

    Why? Query if grade is higher than b or lower than d – compare

Using the same grade database, but now the common queries are different:

1. Make index at classname → 'create index idx_className ON StudentGrades(classname);'

    Why? Order classes by name faster

2. Make index at classname/grade → 'create index idx_className/grade ON StudentGrades(classname,grade);'

    Why? This index will support the query to get all students who earned an 'A' in a certain class. Look up grades in particular class

Queries on the chess database

1. Add index on Elo in players table → 'create index idx_Elo ;'
    a. Why? Support Elo query by players
2. Add index on WhitePlayer on games table → 'create index idx_games ON games(Games,WhitePlayer);'
    a. Optimize join operation where pID = whitePlayer

Queries on the public Library database

select * from Inventory natural join CheckedOut;

No extra indexes for this one → common column in Inventory and Checked Out – these are primary keys for both tables so index doesn't make sense

More library queries:

Assume the only existing indexes are the primary index on each table (despite whatever indexes are on the actual tables). Common queries are:

select * from Inventory natural join CheckedOut where CardNum=2;

select * from Patrons natural join CheckedOut;

Add index (checkedout, cardnum)

Why? First query is based on cardnum so adding index makes it faster.

Second query would also increase the speed of the natural join there

Still more library queries:

var query =from t in db.Titles select new {Title = t.Title,Serials = from i in t.Inventory select i.Serial};

Add index on TitleID in the inventory table

Why? Optimize subquery that selects serial from inventory based on TitleID

PART 2 - B+ TREE INDEX STRUCTURES

How many rows of the table can be placed into the first leaf node of the primary index before it will split?

4096 bytes/ 15(bytes/row) = 273 rows before splitting

What is the maximum number of keys stored in an internal node of the primary index? (Remember to ignore pointer space. Remember that internal nodes have a different structure than leaf nodes.)

4096 bytes / 14(bytes/row) = 292 keys. Internal node can store 29 keys

What is the maximum number of rows in the table if the primary index has a height of 1? (A tree of height 1 has 2 levels and requires exactly one internal node)

293 x 273 = 79,989 rows maximum

What is the minimum number of rows in the table if the primary index has a height of 1? (A tree of height 1 has 2 levels). The minimum capacity of a node in a B+ tree is 50%, unless it is the only internal/leaf node. The minimum number of children of a root node is 2.

2 * (273/2) = 273 so that many rows. Minimum num of children for root node is 2 and each node must b at least 50% capacity

If there is a secondary index on Grade, what is the maximum number of entries a leaf node can hold in the secondary index?

4096/(1 byte/entry) = 4096 entries in the secondary index

Another table

Assume that for some table, rows occupy 128 bytes.

What is the maximum number of leaf nodes in the primary index if the table contains 48 rows?

2048 bytes/ (128 bytes/row) = 16. For 48 rows → 48/16 = 3 leaf nodes. Up to 3 leaf nodes

What is the minimum number of leaf nodes in the primary index if the table contains 48 rows?

4096/(128 bytes/row) = 32. With 48 rows → 48/32 = 2 leaf nodes. Up to 2 leaf nodes