10/10 Questions Answered
Saved on Nov 18 at 6:25 PM

Quiz 8

STUDENT NAME

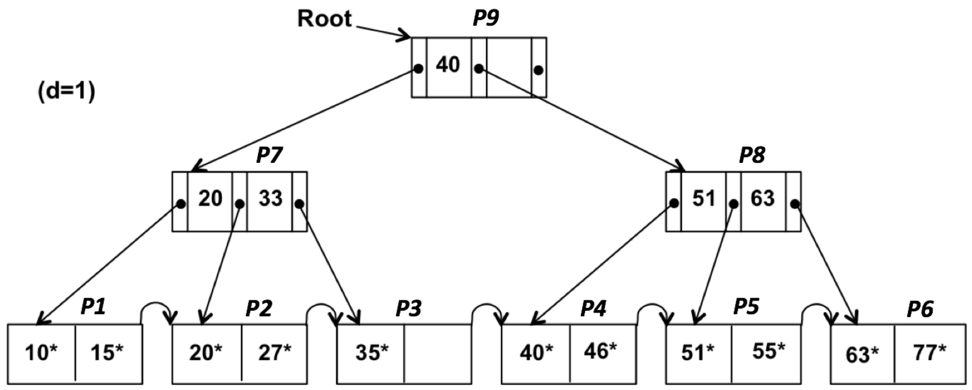
Search students by name or email... ▾

Q1 Tree Indexes 201

5 Points

Below you will find an example of a B+ tree index with ridiculously small pages --- the B+ tree is of order d=1 --- for ease of example-drawing. The pages in the underlying index file are numbered for ease of reference. *Note:* This index resulted from a DBA having said `CREATE INDEX AgeIndex ON Users(age) TYPE BTREE`, but several operations have been performed on the index since that time. The schema of the indexed User table is of the form `Users (userid, name, age)`.

For Gradescope purposes, shown below the tree is an example of how you should represent the B+ tree when answering the following problems. Note that you can use cut/paste from this initial tree to copy it into the answer box and then edit it as needed. (*NOTE*: Before the Endterm you need to make absolutely sure that you are capable of performing cut/paste operations like this! If you have problems in one browser, try another -- this is important! You will not survive the Endterm exam without it. You will be asked to show whole trees in your insert/delete answers -- not just changed nodes -- to simplify grading and enable partial credit if needed.)



```
P9: [P7|40|P8]

P7: [P1|20|P2|33|P3]
P8: [P4|51|P5|63|P6]

P1: [10*|15*]
P2: [20*|27*]
P3: [35*|-- ]
P4: [40*|46*]
P5: [51*|55*]
P6: [63*|77*]
```

(If you need to add pages as you work through the problem, the next page number used in the tree should be P10.)

Q1.1

1 Point

The index leaves were initially 100% full, but one of the users left the web site that the database is supporting --- and as a result, their data record was deleted along with the corresponding index entry. How old was the user who left? (***Hint:*** Look very carefully at the node with a "hole" and also at its parent node.)

33

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q1.2

1 Point

What pages will be read (and in what order) to process the range query `SELECT * FROM Users WHERE age > 21 AND age < 50`?

P9, P7, P2, P3, P4, P5

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q1.3

1 Point

What index pages would be written (newly allocated or updated) to `INSERT INTO Users VALUES (234, 'Hans Duo', 39)`?

P3

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q1.4

1 Point

Starting with the ***original index***, using the notation above, show the index structure that would result from execution of the statement `INSERT INTO Users VALUES (235, 'D. Fourpio', 53)`.

P9: [P7|40|P8|55|P11]

P7: [P1 |20|P2|33|P3]
P8: [P4 |51|P5|--|--]
P11: [P10|63|P6|--|--]

P1: [10*|15*]
P2: [20*|27*]
P3: [35*|--]
P4: [40*|46*]
P5: [51*|53*]

P10: [55*|--]

P6: [63*|77*]

Save Answer Last saved on **Nov 18 at 6:25 PM**

Q1.5

1 Point

Again starting with the ***original index***, using the notation above, show the index structure that would result from executing the statement `DELETE FROM Users WHERE age = 35`.

P9: [P7|40|P8]

P7: [P1|20|P2|--|--]

P8: [P4|51|P5|63|P6]

P1: [10*|15*]

P2: [20*|27*]

P4: [40*|46*]

P5: [51*|55*]

P6: [63*|77*]

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q2 Hashing It Out

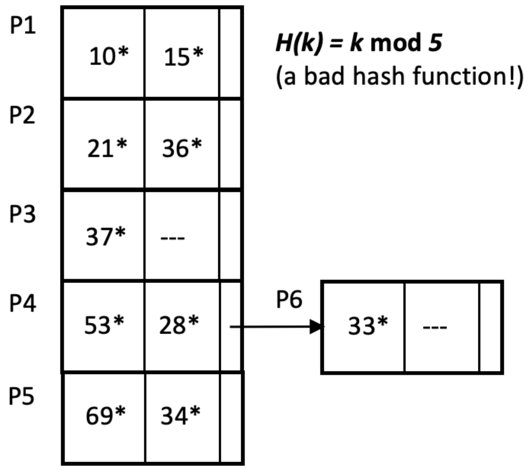
2 Points

Below you will find an example of a static hashed index with small pages for ease of example-drawing. ***NOTE:*** This index resulted from

`CREATE INDEX AgeIndex ON User(age) TYPE HASHED`

being used to create a secondary index. The indexed table's schema is once again

`Users (userid, name, age)`.



For Gradescope purposes, we can represent this structure as follow:

```
P1: [10*|15*|]
P2: [21*|36*|]
P3: [37*|-- |]
P4: [53*|28*|P6]
P5: [69*|34*|]

P6: [33*|-- |]
```

(If you need to add pages as you work through the problem, the next page number used in the tree should be P7.)

Q2.1

1 Point

Given the nature of the index, which of the following would make sense to be *I(k)* for the hashed index's entries of the form*(k, I(k))*? (Check all that apply.)

□ RID

✓ list of RIDs

□ PK

✓ list of PKs

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q2.2

1 Point

Starting with the original index, using the notation above, show the index structure that would result from execution of the statement `INSERT INTO Users VALUES (235, 'Princess Sita', 31)`.

P1: [10*|15*|]
P2: [21*|36*|P7]
P3: [37*|-- |]
P4: [53*|28*|P6]
P5: [69*|34*|]

P7: [31*|-- |]
P6: [33*|-- |]

Save Answer Last saved on **Nov 17 at 7:39 PM**

Q3 Decisions, Decisions

3 Points

Now let's think briefly about physical database design!

Q3.1

1 Point

Which of the two types of indexes (i.e., B+ tree or hashed) from questions 1 & 2 would it make sense to create to support queries of the form `SELECT COUNT(*) FROM Users WHERE age > ?` (Check all that apply.)

✓ B+ tree

□ hashed

□ neither

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q3.2

1 Point

Which of the two types of indexes (i.e., B+ tree or hashed) from questions 1 & 2 would it make sense to create to support queries of the form `SELECT COUNT(*) FROM Users WHERE age = ?` (Check all that apply.)

✓ B+ tree

✓ hashed

□ neither

Save Answer Last saved on **Nov 17 at 5:11 PM**

Q3.3

1 Point

For such queries, should the index be clustered or unclustered in order to minimize this query's I/O cost? (Check all that apply.)

□ clustered

□ unclustered

✓ either one, actually

Save Answer Last saved on **Nov 17 at 5:11 PM**

Save All Answers

Submit & View Submission ❯