CS150A Homework 1 – Writing

School of Information Science and Technology
September 29, 2022

1 SQL (40 PTS)

Consider the following relations with underlined primary keys:

Student (sname, sid, gpa, level, deptid)
Dept (dname, deptid)
Course (cid, cname, deptid, units)
Takes (sid, cid)

1. Write a SQL query that returns the names (i.e., snames) of students who have taken more courses outside their department than inside their department. For this question, you can assume that all students in the database have taken at least one course inside their department.

```
CREATE VIEW OutsideCourseCount AS
SELECT s.sid COUNT(*) AS count
FROM Student AS s, Dept AS d, Course AS c, Takes AS t
WHERE s.sid = t.cid
AND t.cid = c.cid
AND c.deptid != s.deptid
GROUP BY s.sid

CREATE VIEW InsideCourseCount AS
SELECT s.sid COUNT(*) AS count
FROM Student AS s, Dept AS d, Course AS c, Takes AS t
WHERE s.sid = t.cid
AND t.cid = c.cid
AND c.deptid = s.deptid
GROUP BY s.sid

SELECT s.sname
FROM Student AS s, OutsideCourseCount AS o, InsideCourseCount AS i
WHERE o.count > i.count
AND o.sid = i.sid
AND s.sid = o.sid
```

2. Which of the following queries returns the department numbers of those departments for which there are no courses being offered? (The correct answer may be more than one).

   A. SELECT D.deptid
      FROM Dept D, Course C

WHERE D.deptid NOT EQUAL C.deptid;

B. SELECT C.deptid, COUNT(C.deptid)
FROM Course C
GROUP BY C.deptid
HAVING COUNT (C.deptid) = NULL;

C. SELECT C.deptid
FROM Course C
WHERE C.deptid NOT IN (SELECT * FROM Dept);

D. SELECT D.deptid
FROM Dept D
WHERE NOT EXISTS (SELECT * FROM Course C
WHERE C.deptid = D.deptid);

E. None of the above

3. Which of the following queries returns the id of the student with the highest GPA? (The correct answer may be more than one).

A. SELECT S.sid
FROM Students S
WHERE S.gpa = MAX(S.gpa);

B. SELECT S.sid, MAX(S.gpa);
FROM Students S
GROUP by S.gpa

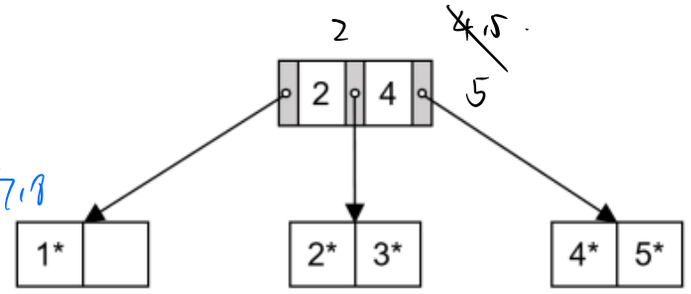
C. SELECT S.sid
FROM Student S
WHERE S.gpa > ALL (SELECT S.gpa FROM Student S);

D. SELECT S.sid
FROM Student S
Where S.gpa = (SELECT MAX(S.gpa)
FROM Student S);

E. None of the above

2 INDEX AND B+ TREES (30 PTS)

1. The following B+ Tree (height=1) has order 1 (max fan-out of 3) and each leaf node can hold up to 2 entries. Answer each of the follow questions independently of each other.



A. What value(s) would be in the root node if we were to insert 0?

2,4

B. What value(s) would be in the root node if we were to insert 6?

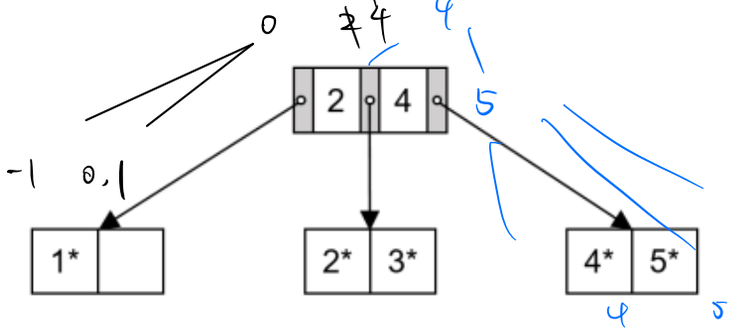
4

C. ~~Starting with the height 1 tree in the picture above, suppose we start inserting keys 6, 7,8, … and so on. After inserting what key will the height of the tree become 3?~~

2. What is the resulting B+ tree after bulk loading the keys: 3, 8, 5, 19, 1, 20, 11, 7, 14? Assume we are building an order d = 1 index with a fill factor of 2/3.

2 INDEX AND B+ TREES (30 PTS)

1. The following B+ Tree (height=1) has order 1 (max fan-out of 3) and each leaf node can hold up to 2 entries. Answer each of the follow questions independently of each other.



   A. What value(s) would be in the root node if we were to insert **-1** and **0**?

   2

   B. What value(s) would be in the root node if we were to insert **6** and **7**?

   4

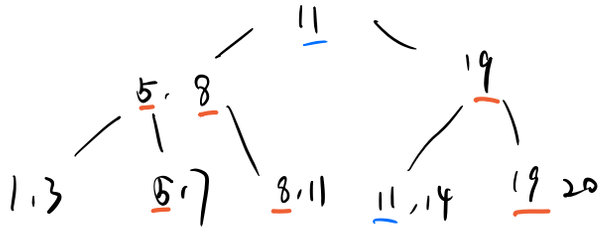
   C. Starting with the height 1 tree in the picture above, suppose we start inserting keys 6, 7,8, … and so on. After inserting what key will the height of the tree become **4**?

   18

2. What is the resulting B+ tree after bulk loading the keys: 3, 8, 5, 19, 1, 20, 11, 7, 14? Assume we are building an order d = 1 index with a fill factor of 2/3.

order d=1 max fan-out = 2d+1 = 3 $3 \times \frac{2}{3} = 2$

Sorted: 1, 3, 5, 7, 8, 11, 14, 19, 20



3 File Organization (30 pts)

Suppose you are playing a game. The same sets of distinct records represented as integers are stored in a heap file M and a sorted file N respectively with no wasted space.

Now you have an integer I with value c and you want to eliminate the record with the same value c that exists in these two files. This will require first finding this record with value c among the records in each file. Then after adding these two equivalent records to get $2c$, you will replace the original record as $2c$. Note that you should keep the sorted file ordered after these operations, and we assume adding won't take any cost. Consider the situation that all the records are smaller than $2c$ in each file.

We use **B** to denote the number of data blocks, and **R** to denote the number of records per block. The average time to read or write a disk block is **D**.

1. What is the average cost of these operations under the optimal approach for heap file M? Give your result and brief explanations.

   find c: equality search: $\frac{BD}{2}$

   delete original c: D

   Insert 2c: 2D cost:

   $\Rightarrow \frac{BD}{2} + 3D$

2. What is the average cost of these operations under the optimal approach for sorted file N? Give your result and brief explanations.

   find c: $D \log_2 B$

   delete original c: $2 * \frac{BD}{2}$ → read, write

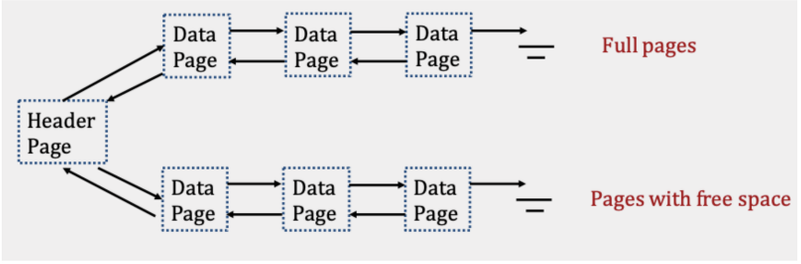
   Insert 2c: $D\log_2 B + D$. (since 2c biggest)

   cost: $2D\log_2 B + BD + D$.

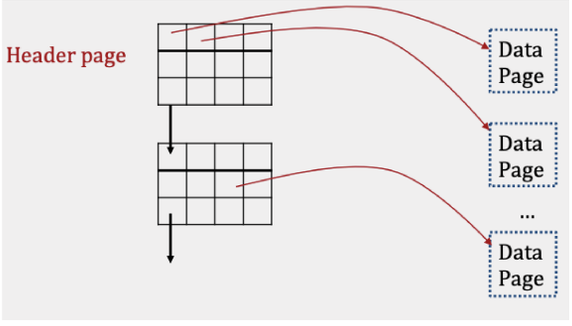
3 File Organization (30 pts)

For heap files, we have two implementation methods:

- Linked List Implementation



- Page Directory Implementation



1. Suppose we have a heap file A implemented with a linked list. It has 6 pages with free space, at least one of which has enough space to insert a record, and 12 full pages. In the worst case, writing a record to the final page with free space will require how many I/Os? Give your result and brief explanations.
   (Hint: Consider what happens when after writing, the page becomes full and assume that the header page can insert at the beginning of the full pages linked list.)

12 I/O.

7 I/o to find final page. 1 header, 6 free pages.

1 I/o write the record. to final page

1 I/o change pointer of previous free page.

1 I/o change header (final can be full and move to

2 I/o. full linked-list head: read, write full-head)

change to previous

2. Suppose we have a heap file B implemented with a page directory. One page in the directory can hold 16 page entries. There are 86 pages in file A in total. In the worst case, writing a record to a page with free space will require how many I/Os (assuming at least one free page with enough space to insert a record exists)? Give your result and brief explanations.

9 I/O. $\lceil \frac{86}{16} \rceil = 6$.

6 I/o to find free page directly.

directory contains # of free page for each page

1 read the page with free page.

1 write to that page

1 write page directory