Washington State University

School of Electrical Engineering and Computer Science

CptS 451 – Introduction to Database Systems

Online

Dr. Sakire Arslan Ay

Homework-6

Name: Nam Jun Lee

Student Number: 11606459

Question1: Indexing

Consider the following relational schema for a portion of a university database:

Prof(ssno, pname, office, age, sex, specialty, dept_did)

Dept(did, dname, budget, num_majors, chair ssno)

So, I created a database called hw6, made each table, and put in any samples (13 rows in Prof & 4 rows in Dept):

```
CREATE TABLE prof (
        ssno VARCHAR(4) PRIMARY KEY,
        pname CHAR(20),
        office VARCHAR(20),
        age INTEGER,
        sex CHAR(6),
        speciality VARCHAR(20),
        dept_did VARCHAR(4),
        FOREIGN KEY (dept_did) REFERENCES dept (did));
CREATE TABLE dept (
        did    VARCHAR(4) PRIMARY KEY,
        dname VARCHAR(20),
        budget DECIMAL(8,2),
        num_majors INTEGER,
        chair_ssno VARCHAR(4));

INSERT INTO prof
VALUES ('p131', 'Alies', 'BS Hall 122', 33, 'Female', 'Accounting', 'a123');
INSERT INTO prof
VALUES ('p121', 'Jason', 'BS Hall 132', 41, 'Male', 'Finance', 'a123');
INSERT INTO prof
VALUES ('p141', 'John', 'BS Hall 222', 50, 'Female', 'Marketing', 'a123');
```

```
INSERT INTO prof
VALUES ('p111', 'Calvin', 'BS Hall 321', 31, 'Female', 'Management', 'a123');
INSERT INTO prof
VALUES ('p151', 'Julidan', 'COS Hall 421', 53, 'Male', 'Network', 'a223');
INSERT INTO prof
VALUES ('p161', 'Hopkins', 'COS Hall 423', 42, 'Female', 'Digital design', 'a223');
INSERT INTO prof
VALUES ('p171', 'Ronald', 'COS Hall 435', 37, 'Male', 'Data structure', 'a223');
INSERT INTO prof
VALUES ('p181', 'Emy', 'MAT Hall 111', 28, 'Female', 'Linear algebra', 'a333');
INSERT INTO prof
VALUES ('p191', 'Morina', 'MAT Hall 142', 35, 'Female', 'Math modeling', 'a333');
INSERT INTO prof
VALUES ('p201', 'Eijah', 'PY Hall 122', 30, 'Male', 'General physics', 'a423');
INSERT INTO prof
VALUES ('p211', 'Robert', 'PY Hall 212', 58, 'Male', 'Modern physics', 'a423');
INSERT INTO prof
VALUES ('p212', 'Andy', 'PY Hall 324', 39, 'Female', 'Mechanics', 'a423');
INSERT INTO prof
VALUES ('p213', 'Andrew', 'PY Hall 320', 52, 'Male', 'Thermal physics', 'a423');

INSERT INTO dept
VALUES ('a123', 'Business', 2300.42, 4, 'p111');
INSERT INTO dept
VALUES ('a223', 'Computer Science', 3700, 3, 'p161');
INSERT INTO dept
VALUES ('a333', 'Mathmatics', 1800.42, 2, 'p181');
INSERT INTO dept
VALUES ('a423', 'Physics', 2800.12, 4, 'p211');
```

1. List the names, ages, and offices of professors of a user-specified sex (male or female) who have a user-specified research specialty (e.g., artificial intelligence). Assume that the university has a diverse set of faculty members, making it very uncommon for more than a few professors to have the same research specialty.

SQL Query:

SELECT pname, age, office

FROM prof

WHERE prof.sex = 'Female' AND prof.speciality = 'Accounting'; /*user-specified*/

<u>Unclustered hash index on speciality and sex in Prof</u>

2. List all the department information for departments with professors in a user specified age range.

SQL Query:

SELECT *

FROM dept as dt, prof as pf

WHERE dt.did = pf.dept_did AND

pf.age > 45 AND pf.age < 55; /*user-specific*/

<u>Clustered B+ tree index on dept_did and age in Prof. And unclustered hash index on did in Dept.</u>

3. List the department id, department name, and chairperson name for departments with a user-specified number of majors.

SQL Query:

SELECT dt.did, dt.dname, dt.chair_ssno

FROM dept as dt

WHERE dt.num_majors = 4 /*user-specific*/

<u>Unclustered hash index on num_majors in Dept.</u>

4. List the lowest budget for a department in the university.

SQL Query:

SELECT dt.did, dt.dname, MIN(dt.budget)

FROM dept as dt, dept as dt1

WHERE dt.budget = (SELECT MIN(dt1.budget) FROM dept as dt1)

GROUP BY dt.did;

<u>Clustered B+ tree index on budget in Dept.</u>

5. List all the information about professors who are department chairpersons.

SQL Query:

SELECT *

FROM dept, prof

WHERE dept.chair_ssno = prof.ssno;

<u>Unclustered B+ tree index on chair_ssno in Dept. And unclustered hash index on ssno in Prof.</u>

Question 2: Storage and Indexing

(a) Heap file

(i) Total attributes size = 40 *4 * 100000 = 16000000 bytes

Block size = 16000 bytes; # of pages = 16000000 / 16000 = 1000

Hence, cost of file scan = **<u>1000D</u>**

(ii) Equality search (sid = ‘25700’): 1000D * 0.5 = **<u>500D</u>**

(iii) Range search (sid <= ‘25700’):

Uniformly distributed between ‘100’ and ‘204,900’;

Range condition formula: (upper – lower) / (X2 – X1):

((25700 - 100) * 1000) / (204900 – 100) = 25600000 / 204800 = **<u>125D</u>**

(b) Clustered B+ tree

(i) Alternative1, attributes size = 40 * 1 * 100000 = 4000000 bytes

Block size = 16000; # of pages = 4000000 / 16000 = 250

Physical data pages: 1.5 times more than original data file

Hence, cost of file scan = 250* 1.5 = <u>**375D**</u>

(ii) Equality search (sid = '25700'): DlogF 1.5B;

= D (log10 (1.5 * 375)) = D(log10(562.5)) = D(2.75)

= <u>**2.75D**</u>

(iii) Range search (sid <= '25700'): D(logF 1.5B + # matching pages)

\# matching pages: ((25700 – 100) * 375) / (204900 – 100)

= 46.875 = 47 <- Round it

D(log10 (1.5*375) + 47) = D(log10(562.5) + 47) = D(2.75 + 47)

= 49.75D

Also, Height of the B+ tree is 3. Therefore, 49.75 + 3 = <u>**52.75D**</u>

(c) Unclustered B+ tree

(i) Alternative1, attributes size = 40 * 1 * 100000 = 4000000 bytes

Block size = 16000

\# of pages = 4000000 / 16000 = 250

Physical data pages: 1.5 times more than original data file

Unclustered tree index have unsorted (BD) and sorted(4BD):

Hence, cost of file scan <u>unsorted</u> = 250* 1.5 = <u>**375D**</u> &

<u>sorted</u> = 250*1.5*4 = <u>**1500D**</u>

(ii) Equality search (sid <= '25700'): D(logF 0.15B+1)

D(log10 (0.15 * 375) + 1) = D(2.75 + 1) = 3.75D

Also, height of the B+ tree is 3: 3.75 + 3 = <u>**6.75D**</u>

(ii) Range search (sid <= '25700'): D(logF 0.15B + # matching records)

\# matching records: ((25700 – 100) * 375) / (204900 – 100)

= 46.875 = 47 <- Round it

D(log10 (0.15 * 375) + 37) = D(1.75 + 37) = 38.75D

Also, height of the B+ tree is 3: 38.75 + 3 = <u>**41.75D**</u>