CS186 Midterm Review

Lu Cheng, Victor Zhu
Evan Sparks, Daniel Haas, Liwen Sun

So...what's on the midterm?

Pretty much everything, but let's break it down.

Disks, Memory, Buffers

- Disks, Memory, Buffers
  - Buffer replacement policy
    - MRU
    - LRU
    - Clock - We didn’t cover details in lecture. Just know that it’s an estimation of LRU
    - 2Q - Again, not the details
    - Sequential Flooding
    - Why are buffer replacement handled by DBMS as opposed to OS?
  - File, pages, tuples

Practice

Discussion Worksheet 1 - LRU and MRU

Indexing

- Indexing
  - Hash based indexing
    - Static Hashing
    - Linear Hashing
    - Extendible Hashing
  - Tree based indexing
    - B+ Tree
      - How to insert, how to delete
    - ISAM Tree
      - How to insert, how to delete
  - Alt 1 vs Alt 2 vs Alt 3

Practice

from spring 2011 midterm 1

https://tbp.berkeley.edu/examfiles/cs/cs186-sp11-mt1-Franklin-soln.pdf

Practice

a) [3 points] An extendible hashing structure can never contain overflow pages.

True or False

Practice

a) [3 points] An extendible hashing structure can never contain overflow pages.

True or False

Practice

b) [3 points] Your company asks you to design a hashing mechanism to index old archive data. You know that you will not performing insertion or deletions on the data, but will be querying it for equality searches on the search key. Which hashing method is most appropriate?

A. Static

B. Extendible

C. Linear

Practice

b) [3 points] Your company asks you to design a hashing mechanism to index old archive data. You know that you will not performing insertion or deletions on the data, but will be querying it for equality searches on the search key. Which hashing method is most appropriate?

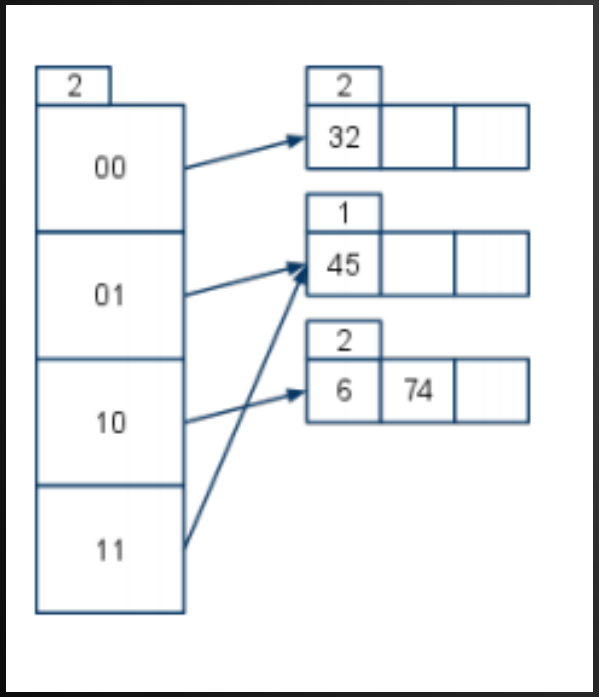
A. Static

B. Extendible

C. Linear

Practice

c) [3 points] Consider the Extendible Hashing structure below. What is the maximum number of keys can you insert before the size of the directory must double? (no explanation needed)



8 keys

File Organization

- File Organization
  - Fixed Length Records
  - Variable Length Records
    - How do we encode these?
  - Slotted Directory
  - Sorted Files vs Heap Files
    - Average I/O costs for operations
  - Clustered vs Unclustered
    - Operation costs
- Know basics of memory hierarchy
  - Why it exists

Practice

Qn 4 from

https://hkn.eecs.berkeley.edu/examfiles/cs186_sp03_f.pdf

Practice

Apartment (aid, capacity)

GraduateStudent (SID, age, sex, dept, GPA, aid)

— GraduateStudent.aid is a foreign key referencing Apartment.

— There are very few “single” apartments (i.e. where capacity=1)

— 50% of the Graduate Students are males.

Pick the access method that would benefit the query most in terms of I/O performance

Query 1: List aids of apartments with capacity=1

i. Clustered B+Tree index for Apartment on aid field

ii. Unclustered B+Tree index for Apartment on capacity field

iii. Clustered B+Tree index for Apartment on capacity field.

Apartment (aid, capacity)

GraduateStudent (SID, age, sex, dept, GPA, aid)

- very few “single” apartments
- 50% of the Graduate Students are males

Query 1: List aids of apartments with capacity=1

i. Clustered B+Tree index for Apartment on aid field

ii. Unclustered B+Tree index for Apartment on capacity field

iii. Clustered B+Tree index for Apartment on capacity field.

Clustered B+Tree index for Apartment on capacity field

Query 2: List SIDs of male graduate students

i. No indexes. Use a file scan on GraduateStudent.

ii. Unclustered B+Tree index for GraduateStudent on sex

iii. Clustered B+Tree index for GraduateStudent on SID

Apartment (aid, capacity)

GraduateStudent (SID, age, sex, dept, GPA, aid)

- very few “single” apartments
- 50% of the Graduate Students are males

Query 2: List SIDs of male graduate students

i. No indexes. Use a file scan on GraduateStudent.

ii. Unclustered B+Tree index for GraduateStudent on sex

iii. Clustered B+Tree index for GraduateStudent on SID

i. No indexes. Use a file scan on GraduateStudent.

Query 3: List depts of graduate students staying in apartment aid=4.

i. Clustered B+Tree index for GraduateStudent on aid

ii. No indexes. Use a file scan on GraduateStudent table.

iii. Unclustered B+Tree index for GraduateStudent on <aid, dept>

Apartment (aid, capacity)

GraduateStudent (SID, age, sex, dept, GPA, aid)

- very few “single” apartments
- 50% of the Graduate Students are males

Query 3: List depts of graduate students staying in apartment aid=4.

i. Clustered B+Tree index for GraduateStudent on aid

ii. No indexes. Use a file scan on GraduateStudent table.

iii. Unclustered B+Tree index for GraduateStudent on <aid, dept>

iii. Unclustered B+Tree index for GraduateStudent on <aid, dept>

SQL + Relational Algebra

- SQL
  - This is a database class. Know your SQL!
  - Where, Except, Intersection, Union, Exists, In, Group By, Having, >Any, >All…
- Relational Algebra
  - Projection, Selection, Union, Intersection, Division, Subtraction

Practice

Q2 from Spring 2011 MT2

https://tbp.berkeley.edu/examfiles/cs/cs186-sp11-mt2-Franklin-soln.pdf

SQL Practice

Consider the following schema about students and courses. Primary keys are underlined.

Students (sid, sname, street, city, age, gender)

Registered (sid, cid, grade)

Courses (cid, cname, profname)

Note that all of the SQL queries in parts a and b are syntactically valid

a) Which of the following queries produces the CIDs of courses that have no registered students? (One or more options are correct)

A. SELECT c.cid
FROM Courses c LEFT OUTER JOIN Registered r ON c.cid = r.cid
HAVING COUNT(*) > 0;

B. SELECT cid FROM Registered
EXCEPT
SELECT cid FROM Courses;

C. SELECT cid FROM Courses
WHERE cid IN
(SELECT cid FROM Registered
GROUP BY cid HAVING COUNT(*) = 0);

D. SELECT c.cid FROM Courses c
WHERE NOT EXISTS
(SELECT cid FROM Registered r WHERE r.cid = c.cid);

E. None of the above

Answer: D

b) Which of the following queries are equivalent to the query: SELECT DISTINCT profname FROM Courses (One or more options are correct)

A. SELECT profname FROM Courses GROUP BY profname;

B. SELECT profname FROM Courses
UNION SELECT profname FROM Courses;

C. SELECT DISTINCT profname FROM Courses
UNION ALL SELECT profname FROM Courses

D. SELECT DISTINCT profname FROM Courses WHERE NULL = NULL

E. None of the above

Answer: A, B

c) When would the following two queries return different results for a given database instance? A one sentence answer should be sufficient!!!

SELECT s.sname FROM Students s LEFT OUTER JOIN Registered r ON s.sid = r.sid

SELECT s.sname FROM Students s, Registered r WHERE s.sid = r.sid

If a student is not registered for a course.

d) In the space below, write a SQL query that returns the name and SID of every student enrolled in the class ‘CS186’ whose age is greater than the average age of all the students enrolled in that class. (‘CS186’ is a CID.)

```
SELECT S.sname, S.sid
FROM Students S, Registered R
WHERE S.sid = R.sid AND R.cid = ‘CS186’ AND
        S.age > (SELECT AVG(S2.age)
                 FROM Students S2, Registered R2
                 WHERE S2.sid = R2.sid AND R2.cid =  ‘CS186’);
```

e) In the space below, write a SQL query that returns for all students who have registered for two or more courses, their sid, sname and the number of courses they have registered for.

```
SELECT S.sid, S.sname, count(*)
FROM Students S, Registered R
WHERE S.sid = R.sid
GROUP BY S.sid, S.sname
HAVING count(*) >= 2;
```

Relational Algebra Practice

Q3 from Spring 2011 MT2

https://tbp.berkeley.edu/examfiles/cs/cs186-sp11-mt2-Franklin-soln.pdf

Recall the schema about students and courses from the previous question:

Students (sid, sname, street, city, age, gender)

Registered (sid, cid, grade)

Courses (cid, cname, profname)

a) In the space below, write a Relational Algebra expression that returns the sid and sname of all students who received an “A” in a course taught by “Hilfinger”.

$$\pi_{sid,sname}(\sigma_{grade='A' \wedge profname='Hilfinger'}(S \bowtie R \bowtie C))$$

b) In the space below, write a Relational Algebra expression that returns the sid of all students who have taken both CS162 and CS186 (where CS162 and CS186 are "cid"s) but no other courses. Do not use any unnecessary relations.

$$[\pi_{sid}(\sigma_{cid='CS162'}(R)) \cap \pi_{sid}(\sigma_{cid='CS186'}(R))] - \pi_{sid}(\sigma_{cid\neq'CS162' \wedge\ cid\neq'CS186'}(R))$$

Out of Core + Query Analysis

- External Merge Sort
  - Know the formulas and why
- External Hashing
  - Know the formulas and why
- Extremely Detailed Understanding of Tournament Sort
  - J/K
- Joins
  - Calculating I/Os for each type of Join
  - Know the formulas and why

Joins

Name	Formula
Simple NL	$(p_r * [R]) * [S] + [R]$
Page NL	$[R] * [S] + [R]$
Block NL	$\frac{[R]}{(B-2)} * [S] + [R]$
Sort-Merge NL	$3[R] + 3[S]$
Hash-Join	$3[R] + 3[S]$

Practice

Any question on this week's discussion is fair game!