MIDTERM II
CS 186 Introduction to Database Systems

NAME: ________________________  STUDENT ID: ________________

IMPORTANT: Circle the last two letters of your class account:

   cs186  a b c d e f g h i j k l m n o p q r s t u v w x y z
   a b c d e f g h i j k l m n o p q r s t u v w x y z

DISCUSSION SECTION DAY & TIME: ___________  TA NAME: ___________

This is a closed book examination – but you are allowed one 8.5” x 11” sheets of notes (double sided). You should answer as many questions as possible. Partial credit will be given where appropriate. There are 100 points in all. You should read all of the questions before starting the exam, as some of the questions are substantially more time-consuming than others.

Write all of your answers directly on this paper. Be sure to clearly indicate your final answer for each question. Also, be sure to state any assumptions that you are making in your answers.

GOOD LUCK!!!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Recovery</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>2. SQL</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>3. Relational Algebra</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>4. Query Evaluation and Optimization</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
Question 1 – Recovery [7 parts, 31 points total]

Consider the following content of a log produced using Write-Ahead Logging. Assume that the log contains all operations since the start of the DBMS. Note that at this point, no system crash has occurred:

<table>
<thead>
<tr>
<th>LSN</th>
<th>Transaction ID</th>
<th>Content</th>
<th>prevLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>T1</td>
<td>update P5</td>
<td>null</td>
</tr>
<tr>
<td>20</td>
<td>T2</td>
<td>update P3</td>
<td>null</td>
</tr>
<tr>
<td>30</td>
<td>T1</td>
<td>update P5</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>T2</td>
<td>commit</td>
<td>20</td>
</tr>
</tbody>
</table>

Now consider four statements:

A. P5 appears in the dirty page table with recLSN 10
B. P5 appears in the dirty page table with recLSN 30
C. P5 appears in the dirty page table, but we don’t have enough information to know what the recLSN would be.
D. None of the above

a) [3 points] Consider the state of the system during normal operation after the log record with LSN 40 is written. If the buffer manager uses a “STEAL” / “NO FORCE” policy, which one of the above statements is guaranteed to be true? Why?

D. In steal/no-force, P5 and P3 can be flushed to disk anytime.

b) [3 points] Consider the state of the system during normal operation after the log record with LSN 40 is written. If the buffer manager uses a “NO STEAL” / “NO FORCE” policy, which one of the above statements is guaranteed to be true? Why?

A. In no-steal, P5 must not be flushed before T1 commits.

c) [3 points] Consider the state of the system during normal operation after the log record with LSN 40 is written. If the buffer manager uses a “NO STEAL” / “FORCE” policy, which one of the above statements is guaranteed to be true? Why?

A. In no-steal, P5 must not be flushed before T1 commits.
Question 1 – Recovery (continued)

Below is the state of the log after a system crash has occurred. Assume STEAL/NO FORCE buffer management and that the log contains all operations since the start of the DBMS.

<table>
<thead>
<tr>
<th>LSN</th>
<th>XactId</th>
<th>Content</th>
<th>prevLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>T1</td>
<td>update P5</td>
<td>null</td>
</tr>
<tr>
<td>20</td>
<td>T1</td>
<td>update P5</td>
<td>10</td>
</tr>
<tr>
<td>30</td>
<td>T2</td>
<td>update P4</td>
<td>null</td>
</tr>
<tr>
<td>40</td>
<td>T3</td>
<td>update P1</td>
<td>null</td>
</tr>
<tr>
<td>50</td>
<td>T2</td>
<td>Commit</td>
<td>30</td>
</tr>
<tr>
<td>60</td>
<td>T2</td>
<td>End</td>
<td>50</td>
</tr>
<tr>
<td>70</td>
<td>--</td>
<td>begin_checkpoint</td>
<td>--</td>
</tr>
<tr>
<td>80</td>
<td>--</td>
<td>end_checkpoint</td>
<td>--</td>
</tr>
<tr>
<td>90</td>
<td>T4</td>
<td>update P3</td>
<td>null</td>
</tr>
<tr>
<td>100</td>
<td>T1</td>
<td>update P2</td>
<td>20</td>
</tr>
<tr>
<td>110</td>
<td>T1</td>
<td>Abort</td>
<td>100</td>
</tr>
<tr>
<td>120</td>
<td>T4</td>
<td>update P4</td>
<td>90</td>
</tr>
<tr>
<td>130</td>
<td>T1</td>
<td>CLR: undo LSN 100, undoNextLSN = 20</td>
<td>110</td>
</tr>
</tbody>
</table>

d) [5 points] Assuming that no dirty pages were written to disk prior to the crash, What are contents of the transaction table contained in the checkpoint ending at LSN 80?

<table>
<thead>
<tr>
<th>PageID</th>
<th>RecLSN</th>
<th>XactId</th>
<th>LastLSN</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>40</td>
<td>T1</td>
<td>20</td>
<td>running</td>
</tr>
<tr>
<td>P4</td>
<td>30</td>
<td>T3</td>
<td>40</td>
<td>running</td>
</tr>
<tr>
<td>P5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Question 3 – Recovery (continued)**

e) [5 points] What are the contents of the Dirty Page Table (DPT) and the transaction table (TT) at the end of the analysis stage?

<table>
<thead>
<tr>
<th>PageID</th>
<th>RecLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>40</td>
</tr>
<tr>
<td>P4</td>
<td>30</td>
</tr>
<tr>
<td>P5</td>
<td>10</td>
</tr>
<tr>
<td>P2</td>
<td>100</td>
</tr>
<tr>
<td>P3</td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XID</th>
<th>LastLSN</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>130</td>
<td>abort</td>
</tr>
<tr>
<td>T3</td>
<td>40</td>
<td>running</td>
</tr>
<tr>
<td>T4</td>
<td>120</td>
<td>running</td>
</tr>
</tbody>
</table>

f) [6 points] During Redo:

1. At what LSN does Redo begin?
   
   10

2. What operations (LSNs) are redone (again, assuming no updates made it out to disk before the crash)?
   
   10, 20, 30, 40, 90, 100, 120, 130

3. Show any new log records that are written (if none, write “none”). For CLRs, be sure to show the undoNextLSN.

   none

g) [6 points] During Undo:

1. What log records (LSNs) are read?
   
   130, 120, 90, 40, 20, 10

2. What operations (LSNs) are undone (do not include CLRs)?
   
   120, 90, 40, 20, 10

3. Show any new log records that are written (If none, write “none”). For CLRs, be sure to show the undoNextLSN.
   
   140  T4 CLR: undoLSN 120, undoNextLSN 90
   150  T4 CLR: undoLSN 90, undoNextLSN NULL
   160  T3 CLR undoLSN 40, undoNextLSN NULL
   170  T1 CLR undoLSN 20, undoNextLSN 10
   180  T1 CLR undoLSN 10, undoNextLSN NULL
Question 2 – SQL [5 parts, 25 points total]
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) [4 points] Which of the following queries produces the CIDs of courses that have no registered students. (One or more options are correct) Circle the letters of your answer(s).

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

b) [4 points] Which of the following queries are equivalent to the query:

SELECT DISTINCT profname FROM Courses

(One or more options are correct) Circle the letters of your answer(s).

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
Question 5 – SQL (continued)

Recall the schema about students and courses:

Students (sid, sname, street, city, age, gender)
Registered (sid, cid, grade)
Courses (cid, cname, profname)

c) [4 points] 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 in a course.

d) [8 points] 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) [5 points] 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;
Question 3 – Relational Algebra [2 parts, 12 points total]
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) [5 points] 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”.

\[ \pi_{\text{sid}, \text{sname}}(\sigma_{\text{grade} = 'A'} \land \text{profname} = 'Hilfinger')(S \bowtie R \bowtie C) \]

b) [7 points] 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.

\[ [\pi_{\text{sid}}(\sigma_{\text{cid} = 'CS162'}(R)) \cap \pi_{\text{sid}}(\sigma_{\text{cid} = 'CS186'}(R))] - \pi_{\text{sid}}(\sigma_{\text{cid} = 'CS162'} \land \text{cid} = 'CS186')(R)] \]
Question 4 – Query Optimization [7 parts, 32 points total]

Consider our student database schema (with primary keys underlined):

STUDENTS (sid, sname, street, city, age, gender)
REGISTERED (sid, cid, credits) where sid and cid are foreign keys
COURSES (cid, cname, profname)

With the following statistics:
- The STUDENTS relation has 10,000 tuples; 20 tuples of STUDENTS fit in one page
- NKeys(city) for STUDENTS is 300 (i.e., there are 300 distinct values for city)
- NKeys(age) for STUDENTS is 70
- NKeys(gender) for STUDENTS is 2
- Low(age) for STUDENTS is 11
- High(age) for STUDENTS is 80
- The REGISTERED relation has 40,000 tuples; 40 tuples of REGISTERED fit in one page
- The COURSES relation has 500 tuples; 40 tuples of COURSES fit in one page

a) [5 points] Assuming that only one page of REGISTERED and one page of STUDENTS can be kept in memory at any given time, how many I/Os will the page-oriented nested loops join using REGISTERED as the outer relation perform to answer the query? (You should ignore the cost of writing the final join answer out to disk.) CIRCLE YOUR ANSWER

The outer relation has 1000 pages. Inner relation 500 pages.

\[ 1000 + 1000 \times 500 = 501,000 \text{ IOs} \]

b) [5 points] Now assume that we have B= 21 memory pages available for the join. Using the “Block Nested Loop Join” (where “blocks” of the outer can be multiple pages) with REGISTERED as the outer, how many I/Os will be required? (You should ignore the cost of writing the final join answer out to disk and no output buffer is needed.) CIRCLE YOUR ANSWER

Outer relation has 1000 pages, \( 1000/(21-1) = 50 \) blocks. Inner relation has 500 pages.

\[ 1000 + 1000/(21-1) \times 500 = 26,000 \text{ IOs} \]

c) [6 points] Consider the following query:

SELECT gender, COUNT(*) FROM STUDENTS GROUP BY gender

Calculate the total number of I/Os required to answer this query using hybrid hash-based grouping using the special first partition (\(<\text{GroupVal}, \text{TransVal}\>) approach described during lecture. Assume that the final output of the query does not need to be written to disk. CIRCLE YOUR ANSWER

Since there are only two buckets, we can fit both of them into the first partition of hybrid hashing. There is no rehashing step necessary.

Total IOs = read all pages for STUDENTS = 500.
Question 4 – Query Optimization (continued)

Now, consider the following indexes:

- A *clustered* B+Tree is defined on composite key `<sid,cid>` for REGISTERED
- An *unclustered* B+Tree index is defined on the `city` attribute for STUDENTS
- A *clustered* B+Tree index is defined on the `age` attribute for STUDENTS

For parts d, e, and f, fill in the expected answer size according to the System R approach, an *efficient method* (file scan, index look up, etc) for answering this query and state how many disk accesses will be required. Be sure to state which index(es) if any you are using. *State any assumptions you are making.* Part of your grade will depend on the efficiency of your solution.

**d) [4 points]**

```sql
SELECT *
FROM REGISTERED
WHERE cid = "CS405";
```

<table>
<thead>
<tr>
<th>Estimated # Tuples in Answer</th>
<th>Method and access path Used</th>
<th>Estimated # I/Os Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000 / 500 = 80</td>
<td>File scan (can’t use index since the index is sorted by sid first and then cid)</td>
<td>1000 pages</td>
</tr>
</tbody>
</table>

**e) [4 points]**

```sql
SELECT *
FROM REGISTERED
WHERE sid = "01234567";
```

<table>
<thead>
<tr>
<th>Estimated # Tuples in Answer</th>
<th>Method and access path Used</th>
<th>Estimated # I/Os Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000 / 10000 = 4</td>
<td>Use the clustered composite key index <code>&lt;sid, cid&gt;</code></td>
<td>3 – 6 (cost of finding the index and add one for data page)</td>
</tr>
</tbody>
</table>

**f) [4 points]**

```sql
SELECT *
FROM STUDENTS
WHERE city = ‘Berkeley’ and age > 40
```

<table>
<thead>
<tr>
<th>Estimated # Tuples in Answer</th>
<th>Method and access path Used</th>
<th>Estimated # I/Os Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000 * (1/300) * ((80-40)/(80-10) ~) = 19</td>
<td>Use unclustered index on city and then select age on the fly</td>
<td>10000 / 300 ~ = 33 tuples ~ = 33 I/Os (unclustered index) + cost of btree ~ = 35 – 37</td>
</tr>
</tbody>
</table>

**g) [4 points]** For the query in part (f) If this was a database for UC Berkeley, would you expect the System R estimate for answer size to be an *over-estimate*, an *under-estimate* or an *accurate* estimate of the true answer size? *Clearly* indicate your answer and *Briefly* justify it.

Any reasonable explanation is accepted. For example, age is overestimated since most UC Berkeley students are younger than 40. City is likely underestimated since most UC Berkeley students live in Berkeley.