There are 17 questions and 10 pages in this quiz booklet. To receive credit for a question, answer it according to the instructions given. You can receive partial credit on questions. You have 80 minutes to answer the questions.

Write your name on this cover sheet AND at the bottom of each page of this booklet.

Some questions may be harder than others. Attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can’t understand your answer, we can’t give you credit!

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ. NO PHONES, NO LAPTOPS, NO PDAS, ETC.

Do not write in the boxes below

<table>
<thead>
<tr>
<th>1-4 (xx/20)</th>
<th>5-7 (xx/16)</th>
<th>8-10 (xx/20)</th>
<th>11-13 (xx/21)</th>
<th>14-17 (xx/23)</th>
<th>Total (xx/100)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name:
I Short Answer

1. [6 points]: To reduce the number of plans the query optimizer must consider, the Selinger Optimizer employs a number of heuristics to reduce the search space. List three:

   (Fill in the blanks below.)

   A. 

   B. 

   C. 

2. [4 points]: Give one reason why the REDO pass of ARIES must use physical logging.

   (Write your answer in the space below.)

Name:
II Optimistic Concurrency Control

For the following transaction schedules, indicate which transactions would commit and which would abort when run using the parallel validation scheme described in the Kung and Robinson paper on Optimistic Concurrency Control. Also give a brief justification for your answer. You may assume that if a transaction aborts, it does not execute its write phase, rolling back instantly at the end of the validation phase.

3. [4 points]:

\[\begin{array}{c|c|c}
\text{Time} & \text{Read} & \text{Validate} \\
\hline
\text{T1} & \text{Read set: \{A,B\}} & \text{Write set: \{B,C\}} \\
\text{T2} & \text{Read set: \{A,D\}} & \text{Write set: \{A,B\}} \\
\end{array}\]

(Write your answer in the spaces below.)

Transactions that commit: ____________________________
Transactions that abort: ____________________________
Justification:

4. [6 points]:

\[\begin{array}{c|c|c}
\text{Time} & \text{Read} & \text{Validate} \\
\hline
\text{T1} & \text{Read set: \{A,B\}} & \text{Write set: \{B,C\}} \\
\text{T2} & \text{Read set: \{A,B\}} & \text{Write set: \{B,D\}} \\
\text{T3} & \text{Read set: \{D\}} & \text{Write set: \{D\}} \\
\end{array}\]

(Write your answer in the spaces below.)

Transactions that commit: ____________________________
Transactions that abort: ____________________________
Justification:

Name:
III Schema Design

Consider a relational table:

Professor(
    professor_name, professor_id,
    professor_office_id, student_id,
    student_name, student_office_id,
    student_designated_refrigerator_id, refrigerator_owner_id,
    refrigerator_id, refrigerator_size, secretary_name,
    secretary_id, secretary_office )

Suppose the data has the following properties:

A. Professors and secretaries have individual offices, students share offices.
B. Students can work for multiple professors.
C. Refrigerators are owned by one professor.
D. Professors can own multiple refrigerators.
E. Students can only use one refrigerator.
F. The refrigerator the student uses must be owned by one of the professors they work for.
G. Secretaries can work for multiple professors.
H. Professors only have a single secretary.

5. [10 points]: Put this table into 3rd normal form by writing out the decomposed tables; designate keys in your tables by underlining them. Designate foreign keys by drawing an arrow from a foreign key to the primary key it refers to (see example below.) Note that some of the properties listed above may not be enforced (i.e., guaranteed to be true) by a 3NF decomposition.

   (Write your answer in the space below.)

   emp : eid, name, dno
   dept : did, bldg, name

   Example decomposition of emp/dept tables showing primary and foreign keys.

Name:
6. [3 points]: Which of the eight properties (A–H) of the data are enforced (i.e., guaranteed to be true) by the 3NF decomposition and primary and foreign keys you gave above?

(Write your answer in the space below.)

7. [3 points]: What could a database administrator do to make sure the properties not explicitly enforced by your schema are enforced by the database?

(Write your answer in the space below.)

IV External Aggregation

Suppose you are implementing an aggregation operator for a database system, and you need to support aggregation over very large data sets with more groups than can fit into memory.

Your friend Johnny Join begins writing out how he would implement the algorithm. Unfortunately, he stayed up all night working on Lab 3 and so trails off in an incoherent stream of obscenities and drool before finishing his code. His algorithm begins as follows:

```plaintext
input : Table T, aggregation function fname, aggregation field aggf, group by field gbyf
        /* Assume T has |T| pages and your system has |M| pages of memory */
/* Partition phase */
        n ← ⌊|T|/|M|⌋;
        Allocate bufs, n pages of memory for output buffers;
        Allocate files, an array of n files for partitions;
        foreach record r ∈ T do
            h ← hash(r.gbyf) ;
            /* h ∈ [1 . . . n] */
            Write r into page bufs[h] ;
            if page bufs[h] is full then
                Add bufs[h] to file files[h], and set the contents of bufs[h] to empty ;
            end
        end
/* Aggregate phase */
        foreach f ∈ files do
            /* Your code goes here. */
        end
```

Name:
8. [6 points]: Relative to $|T|$, how big must $|M|$ be for the algorithm to function correctly? Justify your answer with a sentence or brief analytical argument.

(Write your answer in the space below.)

9. [8 points]: Assuming the aggregate phase does only one pass over the data, and that $fname = 'AVG$', describe briefly what should go in the body of the for loop (in place of “Your code goes here”). You may write pseudocode or a few short sentences. Suppose the system provides a function $emit(group, val)$ to output the value of a group.

(Write your answer in the space below.)

10. [6 points]: Describe an alternative to this hashing-based algorithm. Your answer shouldn’t require more than a sentence or two.

(Write your answer in the space below.)

Name:
V Cost Estimation

Suppose you are given a database with the following tables and sizes, and that each data page holds 100 tuples, that both leaf and non-leaf B+Tree pages are dense-packed and hold 100 keys, and that you have 102 pages of memory. Assume that the buffer pool is managed as described in the DBMIN Paper (“An Evaluation of Buffer Management Strategies for Relational Database Systems.”, VLDB 1985.)

<table>
<thead>
<tr>
<th>Table</th>
<th>Size, in pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>100</td>
</tr>
<tr>
<td>$T_2$</td>
<td>1000</td>
</tr>
<tr>
<td>$T_3$</td>
<td>5000</td>
</tr>
</tbody>
</table>

11. [15 points]: Estimate the minimum number of I/Os required for the following join operations. Ignore the difference between random and sequential I/O. Assume that B+Trees store pointers to records in heap files in their leaves (i.e., B+Tree pages only store keys, not tuples.)

(Write your answers in the spaces below. Justify each answer with a short analytical argument.)

- Nested loops join between $T_1$ and $T_2$, no indices.

- Grace hash join between $T_2$ and $T_3$, no indices.

- Index nested loops join between a foreign key of $T_2$ and the primary key of $T_3$, with a B+Tree index on $T_3$ and no index on $T_2$.

Name:
VI  ARIES with CLRs

Suppose you are given the following log file.

<table>
<thead>
<tr>
<th>LSN</th>
<th>TID</th>
<th>PrevLsn</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>SOT</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>UP</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>UP</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>-</td>
<td>SOT</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>4</td>
<td>UP</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>CP</td>
<td>dirty, trans</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>-</td>
<td>SOT</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>5</td>
<td>UP</td>
<td>D</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>7</td>
<td>UP</td>
<td>E</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>3</td>
<td>COMMIT</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>8</td>
<td>UP</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>8</td>
<td>CLR</td>
<td>B</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>7</td>
<td>CLR</td>
<td>E</td>
</tr>
</tbody>
</table>

12. [2 points]: After recovery, which transactions will be committed and which will be aborted? 
   (Write your answers in the spaces below)

   Committed: ______________________

   Aborted: ______________________

13. [4 points]: Suppose the dirty page table in the CP record has only page A in it. At what LSN will 
   the REDO pass of recovery begin? 
   (Write your answer in the space below)

   LSN: ______________________

Name:
14. [4 points]: Again, suppose the dirty page table in the CP record has only page A in it. What pages may be written during the REDO pass of recovery?

(Write your answer in the space below)

Pages: ____________________________

15. [4 points]: Once again, suppose the dirty page table in the CP record has only page A in it. What pages may be written during the UNDO pass of recovery?

(Write your answer in the space below)

Pages: ____________________________

VII  Snapshot Isolation

Oracle and Postgres both use a form of transaction isolation called snapshot isolation. One possible implementation of snapshot isolation is as follows:

- Every object (e.g., tuple or page) in the database has a timestamp; multiple copies (“versions”) of objects with old timestamps are kept until no transaction needs to read them again. (For this question, you don’t need to worry about how such old versions are maintained or discarded.)
- When a transaction begins, the system records the transaction’s start time stamp, \( t_{ss} \). Timestamps are monotonically increasing, such that no two transactions have the same timestamp value.
- When a transaction \( T \) writes an object \( O \), it adds the new version of the \( O \) to \( T \)’s local write set. Versions in the local write set are not read by other transactions until after \( T \) has committed.
- When a transaction \( T \) reads an object \( O \), it reads the most recent committed version with timestamp \( \leq t_{ss} \), reading \( O \) from \( T \)’s own local write set if \( O \) has been previously written by \( T \).
- When a transaction \( T \) commits, a new timestamp, \( t_{sc} \), is taken. For every object \( O \) in \( T \)’s local write set, if the most recent version of \( O \) in the database has timestamp \( \leq t_{ss} \), then \( O \) is written into the database with timestamp \( t_{sc} \). Otherwise, \( T \) aborts. Only one transaction commits at a time.
For example, consider the following schedule:

Initial database: objects A and B, both version 0 (denoted $A_0$ and $B_0$)

$T_1(t_{ss} = 1)$  $T_2(t_{ss} = 2)$

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read($A_0$)</td>
<td></td>
</tr>
<tr>
<td>Write(A)</td>
<td></td>
</tr>
<tr>
<td>Read($A_0$)</td>
<td></td>
</tr>
<tr>
<td>Commit ($t_{sc} = 3$)</td>
<td></td>
</tr>
<tr>
<td>Install $A_3$</td>
<td></td>
</tr>
<tr>
<td>Attempt to commit with $t_{sc} = 4$, but abort, because last version of $A$ ($3 &gt; t_{ss} = 2$)</td>
<td></td>
</tr>
</tbody>
</table>

Here, $T_2$ aborts because it tried to write A concurrently with $T_1$.

16. **[10 points]**: Is snapshot isolation conflict serializable? If yes, state briefly why. If not, give an example of a non-serializable schedule.

(Write your answer in the space below.)

17. **[5 points]**: Oracle claims that snapshot isolation is much faster than traditional concurrency control. Why?

(Write your answer in the space below.)

End of Quiz I