Final Exam

(120 minutes)

⊕ Avoid asking questions - If something is unclear, document your assumptions and move on.

⊕ Do not forget to write your name on the first page. Initial each subsequent page.

⊕ Be neat and precise. We will not grade answers we cannot read.

⊕ We will only look at the front page. Use the backs for scratch paper, but will not look at any text except on the front.

⊕ If you have written something incorrect along with the correct answer, you should not expect to get all the points. I will grade based upon what you wrote, not what you meant.

⊕ You should draw simple figures if you think it will make your answers clearer.

⊕ Good luck and remember, brevity is the soul of wit.
1. **[16 pts] True/False:**

(a) To guarantee durability, we must write all the updates done by a transaction to the disk before the transaction is committed.

(b) Using Prepared Statements leaves the system vulnerable to SQL Injection attacks.

(c) All nodes in a B+-tree must be about half full or more.

(d) SSDs are considered non-volatile storage.

(e) RAID level 5 has better write performance than level 0.

(f) In the slotted page structure, records occupy contiguous space at the end of the block.

(g) The UNION operation on `postgres` eliminates duplicates.

(h) FALSE OR (TRUE AND UNKNOWN) = FALSE.

(i) Conflict serializability implies serializability.

(j) The “C” in ACID stands for Consistency, not Concurrency.

(k) The keyword “where” in SQL maps to “σ” in the relational algebra.

(l) A relation schema that is in 3NF is also in BCNF.

(m) Machine learning “classification” refers to assigning data to existing classes

(n) In Spark, “reduceByKey” is an example of a transformation operation.

(o) The buffer manager is in charge of guaranteeing the durability property.

(p) A relation schema that is in BCNF is also necessarily in 3NF.
2. [12 pts] Relations.

(a) [4 pts] Identify the set of relation schemas that faithfully (without redundancy or additional restrictions) captures the E/R diagram.

i. R(attr1, attr2); S(attr3, attr4); rel(attr1, attr3)
ii. R(attr1, attr2, attr3); S(attr3, attr4)
iii. R(attr1, attr2); S(attr3, attr4, attr1)
iv. RS(attr1, attr3, attr3, attr4);

(b) [4 pts] Identify any single foreign-key constraint in the following picture. What purpose does it serve?

(c) [4 pts] Explain the difference between equi-depth vs equi-width selectivity histograms:
3. [20 pts] Short answer

(a) [4 pts] What is the cost in selection w/ B+-tree in terms of $h_i$, $t_T$, $t_S$ and $b$ (number of pages containing matches), assuming primary index, but not a key, searching on equality:

(b) [4 pts] Re-tread: Compute the answer for:

```
SELECT A FROM R r1 WHERE EXISTS
    (SELECT * FROM R r2 WHERE r2.A = r1.A AND r2.C < r1.C);
```

(c) [4 pts] Assume $R$ is 1000 blocks, $S$ is 500 blocks, 100 blocks of memory available, how many blocks would be read using a block nested loop join w/ $R$ as the outer relation?

(d) [4 pts] Assume a disk’s transfer rate is 1GB/sec, seek time 4 ms, rotational delay 3 ms, block size 10MB. How much time would it cost to read 200 MB of sequential blocks, and also how much to read 200 MB of random blocks.

(e) [4 pts] Given the following query in JavaScript-ish code, provide a userid that makes the query syntactically correct but allows it to return all orders, not just the user’s orders.

```
SELECT * FROM orders WHERE uname = CONCAT("", $userid, ", ")
```
4. **[21 pts] Normal forms**

(a) **[5 pts]** Consider relation \( R = (A, B, C, D, E) \) with functional dependencies:
\[ B \rightarrow C, C \rightarrow D, DE \rightarrow AB \]
Identify the correct set of candidate keys:
   i. \{BC, BE, DE\}
   ii. \{BE, CE, DE\}
   iii. \{BCD, CE, DE\}
   iv. \{BD, CDE, BE\}

(b) **[5 pts]** Is this in BCNF? If not, decompose and list and functional dependencies that are not preserved.

(c) **[5 pts]** Consider relation \( R = (A, B, C, D) \) with functional dependencies: \( B \rightarrow C, C \rightarrow D \).
Decompose into 3NF, if not already there:

(d) **[6 pts]** Show \( A \) extraneous in \( AB \rightarrow C \), given \( F = AB \rightarrow C, B \rightarrow D, \) and \( D \rightarrow C \), for \( R = (ABCD) \) (use back for scratch sheet, then copy answer):
5. [20 pts] Valid schedules

(a) [6 pts] Which of the following is true for the following schedule (circle all that apply):

<table>
<thead>
<tr>
<th></th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>read(Q)</td>
<td></td>
<td>write(Q)</td>
<td></td>
</tr>
<tr>
<td>write(Q)</td>
<td></td>
<td></td>
<td>write(Q)</td>
</tr>
</tbody>
</table>

i. serializable
ii. view serializable
iii. conflict serializable
iv. not serializable

(b) [6 pts] In the following schedule, is there a deadlock? If so, give an explicit description of how to break it.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X(v)</td>
<td></td>
<td>X(z)</td>
<td></td>
</tr>
<tr>
<td>S(v)</td>
<td>S(w)</td>
<td>S(v)</td>
<td></td>
<td>S(z)</td>
</tr>
</tbody>
</table>

(c) [6 pts] Consider the following schedule with time-stamp-based concurrency control. **Which transactions are aborted?** Assume $TS(T_1) < TS(T_2) < TS(T_3) < TS(T_4) < TS(T_5)$, and do not assume Thomas’s Write Rule. $r(x)$ and $w(x)$ refer to reads and writes, respectively.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r(Y)$</td>
<td></td>
<td></td>
<td></td>
<td>$w(Y)$</td>
<td></td>
</tr>
<tr>
<td>$r(X)$</td>
<td>$w(X)$</td>
<td></td>
<td>$r(Y)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w(Y)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$r(Y)$</td>
</tr>
</tbody>
</table>

(d) [2 pts] How would this change if we assumed Thomas’s Write Rule?
6. **[26 pts]** Answer questions about the following transactions isolation schemes with RC (read committed), RR (repeatable read), and SI (snapshot isolation). \( x = y = 0 \) initially. “\( w(x)v \)” means a write to \( X \) with value \( v \).

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginTrans()</td>
<td>BeginTrans()</td>
<td>BeginTrans()</td>
</tr>
<tr>
<td>( r_2(x) )?</td>
<td>( w(x)1 )</td>
<td>( r_1(x) )?</td>
</tr>
<tr>
<td>( r_3(x) )?</td>
<td>Commit()</td>
<td>( r_4(x) )?</td>
</tr>
<tr>
<td>( w(x)2 )</td>
<td>BeginTrans()</td>
<td>Commit()</td>
</tr>
<tr>
<td>Commit()</td>
<td>Commit()</td>
<td>Commit()</td>
</tr>
</tbody>
</table>

(a) **[6 pts]** Which transactions abort under each scheme:

i. RC:
ii. RR:
iii. SI:

(b) **[12 pts]** What values are returned by \( r_1, r_2, r_3, \) and \( r_4 \)?

i. RC:
ii. RR:
iii. SI:

(c) **[2 pts]** Which is the least isolated isolation level, and why?

(d) **[6 pts]** Phantom problem. What’s a potential problem w/ the following transaction, assuming read-committed isolation? (Hint: The “set \( x = ... \)” syntax defines a variable.)

```python
begin_transaction()
    set per_employee = (SELECT COUNT(*) FROM employees)
    UPDATE employees SET pay = pay + 100000 / per_employee
commit()
```
7. [12 pts] True/False

(a) Data warehouses can contain tens of thousands of individual servers.

(b) “Shared nothing” systems scale better than shared memory machines.

c) The following graph shows linear scaleout.

![Diagram showing linear scaleout](image)

(d) Data cubes contain summaries at multiple levels.

(e) ORMs implement a thin object layer over an existing database.

(f) SQL multisets provide direct support for composite attributes.

(g) The Principle of Least Privilege refers to using the least powerful key or capability for a given task.

(h) Whitelisting enhances security of form input by screening out specific characters.

(i) Order-preserving encryption allows searches on encrypted data.

(j) “Write-ahead logging” requires the log to be flushed to disk before any data is updated.

(k) The big advantage of recovery checkpoints is that it limits how far back in the log the recovery process must look.

(l) “No-steal” refers to free lists for block allocation.