Answer the questions in the spaces provided on the page. If you run out of room for an answer, continue on the back of the page.

Instructions:

• DO NOT START THE EXAM UNTIL TOLD TO DO SO

• You only need to answer 5 of the 6 questions.

• On one of the questions make a large slash through it, which indicates that it should not be graded.

• On every page (including the first and last page), write your first and last name, before answering the question. Unnamed pages may be lost.

• If you start to answer a question and then change your mind, please cross out the attempt and write DO NOT GRADE across it.

Figure 1: http://xkcd.com/1129/
Question 1: Schedules ................................................................. 8 points

(a) (2 points) What is the difference between a serial and a serializable schedule?

**Solution:** Serial schedules don’t intersperse actions from different transactions. Serializable schedules can, but the final result will be the same as some ordering of a serial schedule.

(b) (1 point) Which of the four ACID principles are violated by a non-serializable schedule?

- [ ] Atomicity
- [ ] Consistency
- [x] Isolation
- [ ] Durability

(c) (2 points) When does a conflict occur between two transactions?

**Solution:** A conflict occurs between two actions if both transactions are involved with the same element in the database and one of them is a write.

(d) (2 points) Define what conflict-serializable means.

**Solution:** Two schedules are called conflict-equivalent if one can be transformed into the other by a sequence of non-conflicting swaps of adjacent actions.

(e) (1 point) Are all serializable schedules conflict-serializable?

**Solution:** No, all conflict-serializable schedules are serializable by definition, but not all serializable schedules are conflict-serializable. If there are *arithmetic coincidences*, then a serializable schedule may not be conflict-serializable.

Points earned: __________ out of a possible 8 points
Question 2: Precedence ................................................................. 8 points

(a) (2 points) What are the 3 conditions that determine if an action \((A_1)\) from one transaction takes precedence over an action \((A_2)\) in a different transaction?

Solution:

1. \(A_1\) is ahead of \(A_2\) in the schedule
2. Both \(A_1\) and \(A_2\) involve the same database element
3. At least one of \(A_1\) and \(A_2\) is a write action

(b) (4 points) What are the conflicts and implied transaction precedence for the following schedule:
\[ S: r_1(A); r_2(B); w_1(A); w_2(A); w_3(A); w_3(B); r_1(B); \]

Solution:

- \(r_1(A); w_2(A); \Rightarrow T_1 < T_2\)
- \(r_1(A); w_3(A); \Rightarrow T_1 < T_3\)
- \(r_2(B); w_3(B); \Rightarrow T_2 < T_3\)
- \(w_1(A); w_2(A); \Rightarrow T_1 < T_2\)
- \(w_1(A); w_3(A); \Rightarrow T_1 < T_3\)
- \(w_2(A); w_3(A); \Rightarrow T_2 < T_3\)
- \(w_3(B); r_1(B); \Rightarrow T_3 < T_1\)

(c) (2 points) Draw the Precedence Graph for the schedule and indicate if it is conflict-serializable.

Solution:
Not conflict-serializable (cycles present).

Points earned: __________ out of a possible 8 points
Question 3: Simple Locks ................................................................. 8 points

(a) (2 points) According to two-phase locking, when can locks not be acquired?
   √ After the first unlock action
   ○ Before all read and write actions
   ○ Before the last read or write action
   ○ Only upon commit or rollback

(b) (2 points) According to strict two-phase locking, when can unlocks be performed?
   √ Only upon commit or rollback
   ○ Before all read and write actions
   ○ Before the last read or write action
   ○ After the first unlock action

(c) (4 points) For this question there is only one type of lock (an exclusive lock).
   For the following schedule, output all of the read and write actions with the needed
   lock and unlock actions (i.e. $l_1(A)$ and $u_1(A)$). Only lock/unlock when forced to
   by other transactions.
   S: $r_1(A); w_1(B); w_2(C); r_1(A); w_2(A); r_3(B)$;

<table>
<thead>
<tr>
<th>Solution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $l_1(A)$; 5. $l_2(C)$; 9. $l_2(A)$; 13. $r_3(B)$;</td>
</tr>
<tr>
<td>2. $r_1(A)$; 6. $w_2(C)$; 10. $w_2(A)$; 14. $u_2(A)$;</td>
</tr>
<tr>
<td>3. $l_1(B)$; 7. $r_1(A)$; 11. $u_1(B)$; 15. $u_2(C)$;</td>
</tr>
<tr>
<td>4. $w_1(B)$; 8. $u_1(A)$; 12. $l_3(B)$; 16. $u_3(B)$;</td>
</tr>
</tbody>
</table>

Note there are some minor swaps that would also result in a correct answer.

Points earned: __________ out of a possible 8 points
Question 4: Multiple Types of Locks ..................................................... 8 points

(a) Below are three transactions (and six total actions). Unlocks can only happen after all the actions in a transaction have taken place (strict two-phase locking). For this problem, there are shared locks \((sl_1(A))\) and exclusive locks \((xl_1(A))\) and either/both are unlocked with \((u_1(A))\).

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(r_1(A))</td>
<td>(r_2(A))</td>
</tr>
<tr>
<td></td>
<td>(w_1(A))</td>
<td>(w_2(C))</td>
</tr>
</tbody>
</table>

Below is the order of each of the action are interleaved. For each part, output the necessary locks and unlocks that the transaction should perform to make the action take place (don’t forget to include the action itself).

i. (1 point) \(T_2\):

**Solution:** \(sl_2(A); r_2(A);\)

ii. (1 point) \(T_1\):

**Solution:** \(sl_1(A); r_1(A);\)

iii. (1 point) \(T_3\):

**Solution:** \(xl_3(B); w_3(B);\)

iv. (1 point) \(T_2\):

**Solution:** \(xl_2(C); w_2(C); u_2(A); u_2(C);\)

v. (1 point) \(T_1\):

**Solution:** \(xl_1(A); w_1(A); u_1(A);\)

vi. (1 point) \(T_3\):

**Solution:** \(xl_3(C); w_3(C); u_3(B); u_3(C);\)

(b) (2 points) When can an exclusive lock be downgraded to a shared lock?

**Solution:** Never! An exclusive lock can only be released, never downgraded. Doing so would be a violation of two-phased locking.
Question 5: Transaction Modes .................................................. 8 points

For the table below...

After the statement is completed by the associated connection, write which connections are holding each type of lock in the table. If no connection holds a type of lock, leave it blank. A connection can only hold one lock at a time.

<table>
<thead>
<tr>
<th>ID</th>
<th>SQL statement</th>
<th>Shared</th>
<th>Reserved</th>
<th>Exclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CREATE TABLE ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BEGIN TRANSACTION;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>INSERT INTO ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SELECT ...</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>BEGIN IMMEDIATE TRANSACTION;</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DELETE FROM ...</td>
<td></td>
<td></td>
<td>1 2</td>
</tr>
<tr>
<td>2</td>
<td>ROLLBACK TRANSACTION;</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>BEGIN TRANSACTION;</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>BEGIN TRANSACTION;</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>COMMIT TRANSACTION;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SELECT ...</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SELECT ...</td>
<td></td>
<td>2 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>INSERT ...</td>
<td></td>
<td></td>
<td>2 3</td>
</tr>
<tr>
<td>2</td>
<td>COMMIT TRANSACTION;</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>COMMIT TRANSACTION;</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>BEGIN EXCLUSIVE TRANSACTION;</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>SELECT ...</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>UPDATE ...</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>COMMIT TRANSACTION;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 6: Deadlocks ................................................................. 8 points

Our database is running into problems, as many transactions are waiting for locks held
by other transactions. Here is what each transaction is waiting for:

- $T_1$ is waiting on $T_4$
- $T_2$ is waiting on $T_7$
- $T_3$ is waiting on $T_2$
- $T_4$ is waiting on $T_1$
- $T_5$ is waiting on $T_8$
- $T_6$ is waiting on $T_2$
- $T_7$ is waiting on $T_6$
- $T_8$ is not waiting

(a) (3 points) Draw the Wait-For graph for transactions $T_1$–$T_8$.

(b) (3 points) Which transactions are deadlocked?

\[
\begin{align*}
\checkmark T_1 & \quad \checkmark T_2 & \quad \checkmark T_3 & \quad \checkmark T_4 \\
\checkmark T_5 & \quad \checkmark T_6 & \quad \checkmark T_7 & \quad \checkmark T_8 \\
\end{align*}
\]

(c) (2 points) What needs to be done to resolve the deadlock?

**Solution:** Either $T_1$ or $T_4$ needs to be rolled back. And one of $T_2$, $T_6$, or $T_7$ needs to be rolled back.

Points earned: __________ out of a possible 8 points
Question 7: Deadlock Resolution........................................8 points

Below is a time line as to when transactions were started, received locks, and requested a lock. For this problem you can assume all locks are exclusive. You can also assume transaction restarts are handled after the events described.

1. $T_W$ starts
2. $T_W$ gets lock on A
3. $T_X$ starts
4. $T_X$ gets lock on B
5. $T_W$ wants lock on B, waits on $T_X$
6. $T_Z$ starts
7. $T_Z$ gets lock on C
8. $T_Z$ wants lock on B, rollback!

(a) (2 points) Which deadlock resolution strategy is being used?

√ Wait-Die  ○ Wound-Wait

(b) (2 points) If the same events occurred using the other deadlock resolution strategy, what would the time line be?

Solution:

1. $T_W$ starts
2. $T_W$ gets lock on A
3. $T_X$ starts
4. $T_X$ gets lock on B
5. $T_W$ wants lock on B, wounds (rolling back $T_X$), gets lock on B
6. $T_Z$ starts
7. $T_Z$ gets lock on C
8. $T_Z$ wants lock on B, wait on $T_W$

(c) (2 points) Before the rollback on the original Step 8, draw the Wait-For Graph for the transactions.

Solution:

```
  $T_W$  $T_X$
     /\    /
    /   /  \
   $T_Z$
```

(d) (2 points) If you want to have fewer rollbacks (but perhaps undoing more work) which strategy should you use?

○ Wait-Die  √ Wound-Wait

Points earned: __________ out of a possible 8 points
There are two transactions ($T_i$ and $T_j$). $T_i$ started before $T_j$. Both $T_i$ and $T_j$ performing reads and/or writes on database element $E$.

(a) (2 points) Give an example of a Read-too-late event and why it is a problem.

Solution: $T_i$ tries to read element $E$, but $T_j$ wrote to element $E$ first. Because $T_i$ started first it should have read the old value of $E$, but it can’t because $T_j$ changed it.

(b) (2 points) Give an example of a Write-too-late event.

Solution: $T_i$ tries to write element $E$, but $T_j$ read from the element $E$ first. Because $T_i$ started first it should have changed the old value of $E$, before $T_j$ had the opportunity to read it. Now $T_j$ has read the wrong (old) value.

(c) (1 point) Why are physically unrealizable behaviors a problem?

Solution: Because they result in non-serializable schedules and hence break isolation.

(d) (1 point) What is the Thomas Write Rule?

Solution: If a transaction needed to write to an element that has already been written to by a transaction with a later time stamp, then the write doesn’t have to take place. It doesn’t matter because it would have been overwritten anyways.

(e) (2 points) When does an optimistic scheduler (time stamp/validating) outperform a pessimistic scheduler (locking)?

Solution: If conflicts between transactions are rare (mostly reads or transactions which touch different parts of the database) than optimistic schedulers will have fewer transaction delays. However, it may perform more rollbacks, so rollback performance should not be a factor.
Question 9: Legal Optimistic Schedules ........................................ 8 points
Below is a time line of when two transactions ($T_i$ and $T_j$) performed actions on the database.

1. $T_i$ begins transaction
2. $r_i(A)$
3. $T_j$ begins transaction
4. $w_i(B)$
5. $w_j(A)$
6. $T_i$ commits transaction
7. $T_j$ commits transaction

For proposed action, indicate if that action were added (in isolation from the other proposed actions) would the resulting schedule result in physically unrealizable behavior.

(a) (1 point) $w_i(A)$ between steps 3 and 4    $\sqrt{\text{Legal}}$    ○ Physically Unrealizable
(b) (1 point) $w_j(B)$ between steps 3 and 4    ○ Legal    $\sqrt{\text{Physically Unrealizable}}$
(c) (1 point) $w_j(B)$ between steps 4 and 5    $\sqrt{\text{Legal}}$    ○ Physically Unrealizable
(d) (1 point) $r_j(B)$ between steps 4 and 5    $\sqrt{\text{Legal}}$    ○ Physically Unrealizable
(e) (1 point) $w_j(C)$ between steps 4 and 5    $\sqrt{\text{Legal}}$    ○ Physically Unrealizable
(f) (1 point) $r_i(A)$ between steps 5 and 6    ○ Legal    $\sqrt{\text{Physically Unrealizable}}$
(g) (1 point) $w_i(A)$ between steps 5 and 6    ○ Legal    $\sqrt{\text{Physically Unrealizable}}$
(h) (1 point) $r_j(A)$ between steps 6 and 7    $\sqrt{\text{Legal}}$    ○ Physically Unrealizable

Points earned: __________ out of a possible 8 points
Question 10: Entity/Relationship Diagram .............................................. 8 points

We are creating a database to record which farms raise which types of animals. Below are some facts that need to be represented in the database:

• Each farm has a name and a unique address
• Each farm may raise multiple animals, and each animal can be uniquely identified by the combination of its species and id number.
• Each animal is raised by one farm.
• Each animal also has a nickname, but this may not be unique to each.
• Some animals are breeders, meaning that we know its sex and the season in which it can be breed.
• Some animals are producers, meaning that they produce one or more products (e.g. eggs, milk, wool).
• The products that can be produced have a unique id, as well as, a name and price.

Draw an E/R diagram illustrating the structure of a database capturing the above information.

Solution:
[Diagram of an E/R diagram showing entities and relationships]

The weird arrow (between raises and Farm) is suppose to be a rounded arrow (representing exactly one).

Points earned: __________ out of a possible 8 points
Question 11: Converting E/R Diagram ............................... 8 points
Below is an E/R representing information about pets and their owners.

(a) (6 points) Write a relational database schema representing the E/R diagram. Be sure to combine relations when possible. Use the Object-Oriented method to create relations for the subclasses.

Solution:

- Pets(Name, Age, AdoptionDate, OwnerFirstName, OwnerLastName)
- Purebred(Name, Age, AdoptionDate, OwnerFirstName, OwnerLastName, Registration)
- Trained(Name, Age, AdoptionDate, OwnerFirstName, OwnerLastName, Teacher)
- PurebredAndTrained(Name, Age, AdoptionDate, OwnerFirstName, OwnerLastName, Registration, Teacher)
- Knows(TrickName, PetName, OwnerFirstName, OwnerLastName)

(b) (2 points) Into what relation(s) would you put a pet that is purebred and trained?

Solution: Only into the relation named PurebredAndTrained.

Points earned: __________ out of a possible 8 points
Question 12: Dependencies .................................................. 8 points

Below is a relation about pets and owners.

<table>
<thead>
<tr>
<th>Pet</th>
<th>Species</th>
<th>Age</th>
<th>Owner</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoe</td>
<td>Dog</td>
<td>8</td>
<td>Josh</td>
<td>123-4567</td>
</tr>
<tr>
<td>River</td>
<td>Cat</td>
<td>11</td>
<td>Josh</td>
<td>123-4567</td>
</tr>
<tr>
<td>Harry</td>
<td>Rat</td>
<td>4</td>
<td>Emily</td>
<td>246-1357</td>
</tr>
<tr>
<td>Ron</td>
<td>Rat</td>
<td>4</td>
<td>Emily</td>
<td>246-1357</td>
</tr>
<tr>
<td>Snape</td>
<td>Rat</td>
<td>3</td>
<td>Emily</td>
<td>246-1357</td>
</tr>
</tbody>
</table>

(a) (6 points) Which functional dependencies are obeyed?

- √ Pet → Species
- √ Species → Owner Phone
- √ Phone → Owner Phone
- ○ Species → Pet
- √ Pet → Phone
- √ Age → Owner

(b) (2 points) The above relation obeys this multivalued dependency (Owner →→ Phone).

What other rows must also be added if the row below is added? Fill in the additional dependent rows as needed (you may not need them all).

<table>
<thead>
<tr>
<th>Pet</th>
<th>Species</th>
<th>Age</th>
<th>Owner</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mal</td>
<td>Bird</td>
<td>12</td>
<td>Josh</td>
<td>555-5555</td>
</tr>
<tr>
<td>Mal</td>
<td>Bird</td>
<td>12</td>
<td>Josh</td>
<td>123-4567</td>
</tr>
<tr>
<td>Zoe</td>
<td>Dog</td>
<td>8</td>
<td>Josh</td>
<td>555-5555</td>
</tr>
<tr>
<td>River</td>
<td>Cat</td>
<td>11</td>
<td>Josh</td>
<td>555-5555</td>
</tr>
</tbody>
</table>

Points earned: __________ out of a possible 8 points
Question 13: Closures ............................................. \(8 \text{ points}\)
Here’s a relation (R), its attributes and its functional dependencies (F):

\[ R(A, B, C, D, E) \]

\[ C \ D \rightarrow B \]

\[ A \rightarrow D \]

\[ E \rightarrow C \]

(a) (2 points) Which of the following are in the attribute set closure \(\{AB\}^+\)?

\[ \checkmark \ \{A\} \quad \checkmark \ \{AB\} \quad \checkmark \ \{D\} \]

(b) (2 points) Which of the following are in the functional dependency closure of \(F\) \((F^+)\)?

\[ \checkmark \ C \ D \rightarrow B \quad \checkmark \ A \rightarrow A \quad \checkmark \ D \rightarrow B \]

\[ \checkmark \ A \ C \rightarrow B \quad \checkmark \ A \ E \rightarrow B \quad \checkmark \ E \rightarrow C \ D \]

(c) (2 points) Which of the following are superkeys?

\[ \checkmark \ \{ABCDE\} \quad \checkmark \ \{AE\} \]

\[ \checkmark \ \{A\} \quad \checkmark \ \{AB\} \]

\[ \checkmark \ \{BCE\} \quad \checkmark \ \{ABE\} \]

(d) (2 points) Which of the following are keys?

\[ \checkmark \ \{ABCDE\} \quad \checkmark \ \{AE\} \]

\[ \checkmark \ \{A\} \quad \checkmark \ \{AB\} \]

\[ \checkmark \ \{BCE\} \quad \checkmark \ \{ABE\} \]

Points earned: __________ out of a possible 8 points
Question 14: Lossless Joins ................................................................. 8 points

Here’s a relation (R), its attributes and its functional dependencies (F):

R(A, B, C, D, E)

C D → B
A → D
E → C

(a) (4 points) Which of the following sets of relations maintain the lossless join property?

√ R₁(ABCDE), R₂(ABCD)
√ R₁(AD), R₂(ABCE)
○ R₁(AB), R₂(BCDE)
○ R₁(CD), R₂(ABCE)

(b) (2 points) Which of the following sets of relations are entirely in Boyce-Codd Normal Form?

○ R₁(ABCDE)
○ R₁(AD), R₂(ABCE)
○ R₁(CDB), R₂(ACDE)
√ R₁(CDB), R₂(AD), R₃(CE), R₂(AE)

(c) (2 points) Which of the following sets of relations are dependency preserving?

√ R₁(ABCDE)
○ R₁(AD), R₂(ABCE)
√ R₁(CDB), R₂(ACDE)
√ R₁(CDB), R₂(AD), R₃(CE), R₂(AE)

Points earned: __________ out of a possible 8 points
Question 15: Decomposition .................................................. 8 points

Here’s a relation (R), its attributes and its functional dependencies (F):

R(A, B, C, D, E)
C D → B
A → D
E → C

(a) (4 points) Decompose the above relation using the Boyce-Codd Normal Form decomposition.

Solution:

1. $R_0$(ABCDE) not in BCNF (C D → B; violated), break into $R_1$ and $R_2$
2. $R_1$(CDB) in BCNF
3. $R_2$(ACDE) not in BCNF (A → D; violated), break into $R_3$ and $R_4$
4. $R_3$(AD) in BCNF
5. $R_4$(ACE) not in BCNF (E → C; violated), break into $R_5$ and $R_6$
6. $R_5$(EC) in BCNF
7. $R_6$(AE) in BCNF

Answer = \{CDB\}, \{AD\}, \{EC\}, \{AE\}

(b) (4 points) What is the key for the relation $R$?

Solution:

1. \{ABCDE\} is a superkey
2. \{ACDE\} is a superkey
3. \{ACE\} is a superkey
4. \{AE\} is a superkey
5. \{A\} is not a superkey
6. \{E\} is not a superkey

Answer = \{AE\} is a key

Points earned: ___________ out of a possible 8 points
If you have finished early, feel free to bring your exam to an instructor.
   Or you can draw a picture of your favorite Pokémon.
Or you can write a haiku about your love of Boyce-Codd Normal Form.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedules</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Precedence</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Simple Locks</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Multiple Types of Locks</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Transaction Modes</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Deadlocks</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Deadlock Resolution</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Optimistic Scheduling</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Legal Optimistic Schedules</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Entity/Relationship Diagram</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Converting E/R Diagram</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Dependencies</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Closures</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Lossless Joins</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Decomposition</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>