Exam (Version B) for CSE 480 (2018) KEY

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

Instructions:

• DO NOT START THE EXAM UNTIL TOLD TO DO SO

• You need to answer 6 of the 7 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.

• If you require more space for a solution than allotted, use the back-side of the same page.

Figure 1: http://xkcd.com/1129/
First Name: __________________________

Version B - 2018

Last Name: __________________________

Question 1: Dependencies Validation ........................................... 50 points

<table>
<thead>
<tr>
<th>α</th>
<th>β</th>
<th>γ</th>
<th>δ</th>
<th>ε</th>
<th>θ</th>
<th>κ</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Above is a relation with columns named after Greek Letters. Using the values in the relation, fill-in the bubbles of the dependencies that are **VALID**. Note in any section: zero, one, or many dependences may be valid.

(a) (10 points) \( \sqrt{\theta \rightarrow \theta} \) \( \sqrt{\epsilon \theta \rightarrow \theta} \) \( \bigcirc \theta \rightarrow \epsilon \theta \) \( \sqrt{\delta \rightarrow \delta \epsilon} \)

(b) (10 points) \( \bigcirc \alpha \theta \rightarrow \beta \) \( \bigcirc \beta \delta \rightarrow \theta \) \( \bigcirc \theta \kappa \rightarrow \epsilon \) \( \sqrt{\gamma \delta \rightarrow \delta} \)

(c) (15 points) \( \bigcirc \alpha \kappa \rightarrow \beta \) \( \sqrt{\epsilon \rightarrow \gamma} \) \( \sqrt{\alpha \rightarrow \theta} \) \( \sqrt{\beta \delta \rightarrow \epsilon} \)

(d) (15 points) \( \sqrt{\alpha \theta \rightarrow \beta \delta} \) \( \sqrt{\beta \delta \rightarrow \alpha \theta} \) \( \sqrt{\alpha \rightarrow \theta} \) \( \sqrt{\theta \rightarrow \alpha} \)

Points earned: __________ out of a possible 50 points
Question 2: Functional Dependences ...................................................... 50 points

(a) (20 points) Which of the following are TRUE given this functional dependency holds:
C D → A B?

√ C D → B
○ C → A B
√ C → C
√ C D → A B C D
○ C D → E
○ E → E
√ C D E → E

(b) (10 points) Which of the following are TRUE given this multivalued dependency holds:
C D →→ A B?

○ C D → A B
√ A B →→ C D
○ A B →→ C
○ A →→ C D

(c) (10 points) Which of the following are TRUE about First Normal Form (1NF):

○ SQL ensures that all databases using it are in 1NF.
○ All databases must be in 1NF to allow for atomic actions.
√ An attribute composed of a students’ first and last name can be atomic.
√ Relations in higher normal forms (2NF, 3NF, ...) must conform to 1NF.

(d) (10 points) Which of the following are TRUE about Fourth Normal Form (4NF):

○ 4NF is the normal form that all databases must conform to.
√ 4NF removes the risk of update and deletion anomalies arising from functional dependencies.
√ All relations that conform to 4NF also conform to Boyce-Codd Normal Form.
○ 4NF ensures faster (or equivalent) database performance than Third Normal Form.

Points earned: __________ out of a possible 50 points
Question 3: Converting E/R Diagram .................................................. 50 points

Below is an E/R diagram representing information about companies and embassies.

(a) (40 points) Write a relational database schema representing the E/R diagram. Be sure to combine relations when possible. Use the NULLS method to create relations for the subclasses.

**Solution:**

- Company (ticker, CEO)
- Headquarters (ticker, year, city_name, initials, name) (can be combined with previous)
- City (city_name, initials, name, mayor)
- State (initials, name, est, bird)
- Embassy (guest_country_name, city_name, initials, host_country_name)

(b) (10 points) You have just been tasked with adding a new feature to the above diagram, corporate taxes. Alter the diagram above to allow the database to track the tax rate different companies face in each of the multiple countries they operate in. Below, describe how the schema from (a) would need to be altered to represent this addition.

**Solution:** New relation added: Taxes (ticker, name, rate)

Points earned: ___________ out of a possible 50 points
Question 4: Decomposition .................................................. 50 points

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

R(A, B, C, D, E, F, G)
A B \rightarrow C G
C E \rightarrow B F
A G \rightarrow D

(a) (30 points) Decompose the above relation using the Boyce-Codd Normal Form decomposition. Use the order of F when checking for violations. Full credit requires you to show your work. Be sure the final decomposition is clearly written.

Solution:

1. \( R_0(ABCDEFG) \) not in BCNF \((A B \rightarrow C G)\), break into \( R_1 \) and \( R_2 \)
2. \( R_1(ABCDG) \) and \( R_2(ABEF) \)
3. \( R_1(ABCDG) \) not in BCNF \((A G \rightarrow D)\), break into \( R_3 \) and \( R_4 \)
4. \( R_2(ABEF) \) in BCNF
5. \( R_3(ADG) \) in BCNF
6. \( R_4(ABCG) \) in BCNF

Answer = \( \{ABEF\}, \{ADG\}, \{ABCG\} \)

(b) (10 points) Which of the following are TRUE about the decomposition \( \{CBE\}, \{ACEG\}, \{AGD\}, \) and \( \{CEF\} \):

- ✓ It is a valid lossless-join decomposition.
- ○ It is dependency preserving.
- ✓ It is in Boyce-Codd Normal Form.
- ✓ It is in Fourth Normal Form.

(c) (10 points) Which of the following are TRUE about the decomposition \( \{ABCDG\} \) and \( \{ABEF\} \):

- ✓ It is a valid lossless-join decomposition.
- ○ It is dependency preserving.
- ○ It is in Boyce-Codd Normal Form.
- ○ It is in Fourth Normal Form.

Points earned: ___________ out of a possible 50 points
Question 5: Transaction Modes ........................................ 50 points

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. If an action can’t be granted the necessary locks, cross out that action and proceed as if that action wasn’t in the schedule (don’t perform a rollback).

<table>
<thead>
<tr>
<th>ID</th>
<th>SQL statement</th>
<th>Shared</th>
<th>Reserved</th>
<th>Exclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>CREATE TABLE ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>BEGIN IMMEDIATE TRANSACTION;</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>BEGIN TRANSACTION;</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>SELECT ...</td>
<td>D</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>SELECT ...</td>
<td>D</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>UPDATE ...</td>
<td>D</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>ROLLBACK TRANSACTION;</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>SELECT ...</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>BEGIN DEFERRED TRANSACTION;</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>SELECT ...</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>DELETE ...</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>BEGIN IMMEDIATE TRANSACTION;</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>COMMIT TRANSACTION;</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>SELECT ...</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>INSERT INTO ...</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>BEGIN TRANSACTION;</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

Points earned: ___________ out of a possible 50 points
I'm putting together an app that will allow me to connect people with items to sell (sellers) with people who want to buy items (buyers). Each person tells me what they sell/buy and at what price. I can then accept the buyer's money, give the seller their listed price, facilitate the delivery, and collect the difference between the two monetary amounts as profit. Help me write a SQL query that will yield my most-profitable buyer/seller transactions. Here is some example data:

Table: `market`

<table>
<thead>
<tr>
<th>name</th>
<th>item</th>
<th>state</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Josh</td>
<td>Water</td>
<td>selling</td>
<td>1.00</td>
</tr>
<tr>
<td>Cam</td>
<td>Water</td>
<td>buying</td>
<td>1.10</td>
</tr>
<tr>
<td>Dennis</td>
<td>Water</td>
<td>buying</td>
<td>10.00</td>
</tr>
<tr>
<td>Josh</td>
<td>House</td>
<td>buying</td>
<td>150000.00</td>
</tr>
<tr>
<td>Josh</td>
<td>Ferret</td>
<td>selling</td>
<td>400.00</td>
</tr>
<tr>
<td>Cam</td>
<td>Ball</td>
<td>buying</td>
<td>0.50</td>
</tr>
<tr>
<td>Zizhen</td>
<td>Water</td>
<td>selling</td>
<td>0.90</td>
</tr>
<tr>
<td>Zizhen</td>
<td>House</td>
<td>selling</td>
<td>200000.00</td>
</tr>
<tr>
<td>Jie</td>
<td>Shirt</td>
<td>buying</td>
<td>5.00</td>
</tr>
<tr>
<td>Cam</td>
<td>Shirt</td>
<td>buying</td>
<td>6.00</td>
</tr>
<tr>
<td>Josh</td>
<td>Shirt</td>
<td>selling</td>
<td>3.00</td>
</tr>
<tr>
<td>Dennis</td>
<td>Shirt</td>
<td>selling</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Write a SQL query that returns a row for each profitable transaction/trade using the "market" table. For instance, Zizhen can sell a "Water" to Dennis, and the profit from such a transaction would be $9.10, because Zizhen wants $0.90 and Dennis pays $10. However, Zizhen can't sell a "House" to Josh because, Zizhen is asking for more than what Josh is willing to pay. Order the rows from most-profitable to least (see Expected Rows).

**Solution:**

```sql
SELECT seller.name AS seller, buyer.name AS buyer, seller.item, buyer.price - seller.price AS profit
FROM market AS seller INNER JOIN market AS buyer
ON seller.item = buyer.item
AND seller.state = "selling"
AND buyer.state = "buying"
WHERE profit > 0
ORDER BY profit DESC;

# The subclauses in the ON and WHERE can be interchanged
```

Points earned: __________ out of a possible 50 points
Question 7: Locks ..................................................... 50 points

(a) (10 points) What would happen to the Precedence Graph of a schedule whose actions were reversed?

Solution: The direction of each edge would be reversed.

(b) (10 points) Select the zero or more rules violated by this schedule:
S: \( sl_1(A); r_1(A); xl_1(A); w_1(A); commit_1; u_1(A); commit_2; u_2(B); \)
- Consistency of Transactions
- Legality of Schedule
- Two-Phased Locking
- Strict Two-Phased Locking

(c) (10 points) Select the zero or more rules violated by this schedule:
S: \( xl_1(C); r_2(C); sl_3(D); r_3(D); commit_3; u_3(D); commit_1; u_1(C); \)
\( \sqrt{\text{Consistency of Transactions}} \)
- Legality of Schedule
- Two-Phased Locking
- Strict Two-Phased Locking

(d) (10 points) Select the zero or more rules violated by this schedule:
S: \( sl_4(C); xl_4(D); r_4(C); u_4(C); commit_4; r_4(D); u_4(D); \)
- Consistency of Transactions
- Legality of Schedule
- Two-Phased Locking
\( \sqrt{\text{Strict Two-Phased Locking}} \)

(e) (10 points) Select the zero or more rules violated by this schedule:
S: \( sl_8(E); r_8(E); xl_8(E); u_8(E); xl_8(F); commit_8; u_8(F); \)
- Consistency of Transactions
- Legality of Schedule
\( \sqrt{\text{Two-Phased Locking}} \)
\( \sqrt{\text{Strict Two-Phased Locking}} \)

Points earned: __________ out of a possible 50 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 limerick 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>Dependencies Validation</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Functional Dependences</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Converting E/R Diagram</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Decomposition</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Transaction Modes</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>SQL statement</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Locks</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>