The University of British Columbia

Computer Science 304

Final Examination

Instructor: Rachel Pottinger **Time**: 2.5 hours **Total marks**: 48

Name ANSWER KEY **Student No**________________

(PRINT) (Last) (First)

Signature____

This examination has 9 pages of questions printed double-sided (5 pieces of paper total). Check that you have a complete exam.

This is a closed book exam, closed notes exam. Answer all the questions on this paper. Give **short but precise** answers. Write down any assumptions that you make. The marks for each question are given in {}. Use this to manage your time.

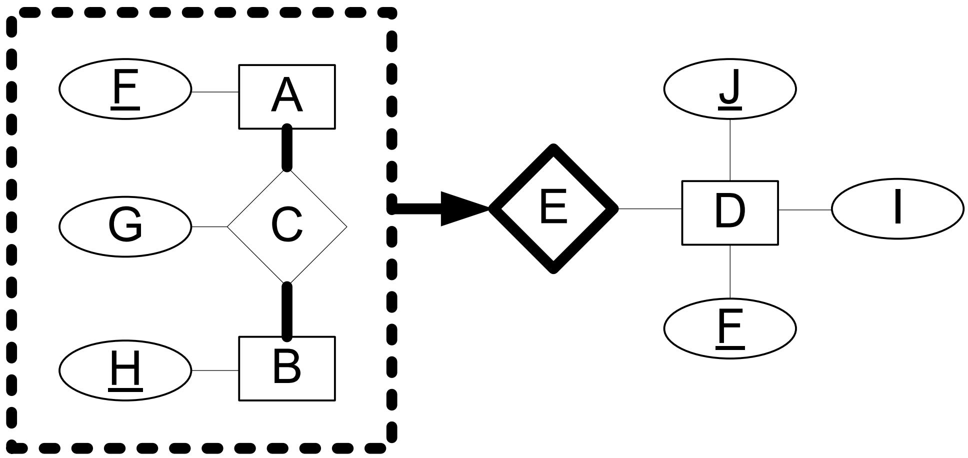
Good Luck!

READ AND OBSERVE THE FOLLOWING RULES:

1. Each candidate should be prepared to produce, upon request, his or her UBC Card.
2. No candidate shall be permitted to enter the examination room after the expiration of one-half hour, or to leave during the first half-hour of the examination.
3. Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
4. **CAUTION** --- Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
   a. Making use of any books, papers or memoranda, calculators or computers, phones, MP3 players, or other memory aid devices, other than those authorized by the examiners.
   b. Speaking or communicating with other candidates.
   c. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

	Mark	**Out of**
2		10
3		10
4		4
5		4
6		4
7		6
8		6
9		4
Total		48

Question 1. {10 marks}



Transform the ER diagram into a relational schema using the methods discussed in class/the book. If there are any conflicting attribute names, rename them something appropriate and easy to understand. State any assumptions that you make – but your assumptions cannot contradict the facts given.

a. {12 marks} What should the relational schema be? You do NOT have to create SQL DDL, just underline the keys and note foreign keys and not null constraints after the relation definition, e.g., you might have "M(n,o): foreign key (o) references R(q), o is not null"

First, consider the entities:
B(H)
A(F)
D(I,F,J)
Then the simple relationship:
C(F,H,G): foreign key (F) references A(F), foreign key (H) references B(H)

Finally, the aggregated weak entity. Because E is a weak entity relationship, then we combine the relationship C (since that's forming the "entity" part of the weak entity) with the relationship E to form table CE(F, H, D-F,G,J): foreign key (F) references A(F), foreign key (H) references B(H),foreign key (D-F,J) references D(F,J)
Note that the table C does not appear in the final answer because it has been subsumed by CE.

b. {4 marks} Are there any constraints in the relational schema that cannot be modeled without using assertions? If so, which constraint(s)? If not, why not?

The constraint that A and B are total in C cannot be represented without assertions.
Note: this is sample question #4 from practice midterm 6

Question 2. {10 marks}

Consider the schema R(A, B, C, D, E) together with the functional dependencies:

BD → A
AB → C
D → A
B → C
C → E

Is R in 3NF? Why or why not? If not, decompose into 3NF using the method we used in class and the book and *circle all relations in your final answer. Show all your work.*

AB+=ABCE
BD+=BDACE
D+=AD
B+=BCE
C+=CE
This question is isomorphic to question 2 from practice midterm 1 # 7
There is no way to get BD any other way, so BD is the only key. But the others do violate 3NF, so we need to decompose.

First we have to take the minimal cover. BD→A is redundant to D→A. AB→C is redundant to B→C. So the only functional dependencies to consider are D→A, B→C, and C→E. Note that because the cover only removes redundant functional dependencies, the original closures still holds. Start with D→A. D is not a key, so decompose: R1(A,D), R2(D,B,C,E). R1 is in BCNF since it is a two attribute relation. R2: B→C still holds, but B is not a key of R2, so decompose: R3(B,C), R4(B,D,E). R3 is in BCNF since it has only two attributes. R4 is not in BCNF since B→E holds in R4, but B is a key of R4. Decompose to R5(B,E), R6(B,D). All are two attribute relations, so all are in BCNF. At this point our answer set is R1(A,D), R3(B,C), R5(B,E), R6(B,D). Now, we consider if there are any functional dependencies that need to be added back in. D→A and B→C are both covered (R1 and R3 respectively). C→E is not. So we add in a new relation R5(C,E), bringing our final answer to R1(A,D), R3(B,C), R5(B,E),R6(B,D), R7(C,E)

Note: this is question 2 from sample midterm #6

Question 3. {4 marks}

Given the relation: Sailors(sid, sname, rating, age), consider the following query: Find the names of sailors with a higher rating than all sailors with age < 21. The following two SQL queries attempt to obtain the answer to this question.

Query A:

```
SELECT  S.sname
FROM    Sailors S
WHERE  S.rating > ANY ( SELECT S2.rating
                        FROM   Sailors S2
                        WHERE S2.age < 21)
```

Query B:

```
SELECT S.sname
FROM   Sailors S
WHERE NOT EXISTS ( SELECT *
                   FROM    Sailors S2
                   WHERE  S2.age < 21 AND S.rating <= S2.rating)
```

For each of the two questions below, circle your answer:

A. Does Query A compute the correct answer? Yes <u>No</u>

B. Does Query B compute the correct answer? <u>Yes</u> No

Note that this is a subset of the question in question 5.5 part 3, only I've reversed the order of the queries.

Only the second query is correct. The first query returns the names of sailors with a higher rating than at least one sailor with age < 21. Note that the answer to the first query does not necessarily contain the answer to the second query. In particular, if all the sailors are at least 21 years old, the first query will return an empty set while the second query will return all the sailors. This is because the NOT EXISTS predicate in the second query will evaluate to true if its subquery evaluates to an empty set, while the ANY predicate in the first query will evaluate to false if its subquery evaluates to an empty set.

Question 4. {4 Marks}

Consider the relation Arc with the following current values:

Arc

x	y
1	2
1	2
2	3
3	4
3	4
4	1
4	1
4	1
4	2

Compute the result of the query:

```
SELECT a1.x, a2.y Count(*)
FROM Arc a1, Arc a2
WHERE a1.y = a2.x
GROUP BY a1.x, a2.y
```

Which of the following is a row in the result? Circle a single correct answer.

a. (1,3,2)

b. (4,2,6)

c. (4,3,1)

d. All of the above

e. None of the above

Note: this is a clicker question – you can compute the correct answer yourself from the "clicker sql scripts" off WebCT Vista.

Question 5. {4 Marks}

Suppose R(a,b) and S(b,c) are relations. Consider the Datalog query:

```
T(x,y) :- R(2,x), S(y,x)
```

Write an equivalent relational algebra expression:

$\pi_{R.b,\ S.b}((\sigma_{a\ =\ '2'}(\ R \bowtie_{R.b\ =\ S.c}\ S))$

Question 6. {6 marks}

Consider the following concurrency protocols: 2PL, Strict 2PL, Time Stamp without the Thomas Write rule, Timestamp with the Thomas Write Rule, and Multiversion Timestamps. For each of the following schedules, circle whether it is allowed under the above protocols. If you cannot decide whether a schedule is allowed based on the listed actions, explain briefly.

The actions are listed in the order they are scheduled and prefixed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.

a. `T1:R(X), T2:R(X), T1:W(X), T2:W(X)`

1. 2PL? Yes <u>No</u> Can't tell:
2. Strict 2PL? Yes <u>No</u> Can't tell:
3. Timestamp w/ TWR? Yes <u>No</u> Can't tell:
4. Timestamp w/o TWR? Yes <u>No</u> Can't tell:
5. Multiversion timestamps? Yes <u>No</u> Can't tell:

b. `T1:W(X), T2:R(Y), T1:R(Y), T2:R(X)`

1. 2PL? <u>Yes</u> No Can't tell:
2. Strict 2PL? Yes <u>No</u> Can't tell:
3. Timestamp w/ TWR? <u>Yes</u> No Can't tell:
4. Timestamp w/o TWR? <u>Yes</u> No Can't tell:
5. Multiversion timestamps? <u>Yes</u> No Can't tell:

This is question 17.3 in the book. Part a is part 1, part b is part 2.

Question 7. {6 marks}

Consider the following sequence of actions using the ARIES protocol:

```
LSN LOG
…  …
00 begin checkpoint
10 end checkpoint
20 update: T1 writes P5
30 update: T2 writes P3
40 T2 commit
50 T2 end
60 update: T3 writes P3
70 T1 abort
<CRASH, RESTART>
```

Assume that the Dirty Page Table and Transaction Table were empty before the start of the log.

a. What is done during Analysis? (Be precise about the points at which Analysis begins and ends and describe the contents of any tables constructed in this phase.)

Analysis determines that the last begin checkpoint was at LSN 00 and starts at the corresponding end checkpoint (LSN 10).
We denote Transaction Table records as (transID, lastLSN) and Dirty Page Table records as (pageID, recLSN) sets.
Then Analysis phase runs until LSN 70, and does the following:
LSN 20 Adds (T1, 20) to TT and (P5, 20) to DPT
LSN 30 Adds (T2, 30) to TT and (P3, 30) to DPT
LSN 40 Changes status of T2 to "C" from "U"
LSN 50 Deletes entry for T2 from Transaction Table
LSN 60 Adds (T3, 60) to TT. Does not change P3 entry in DPT
LSN 70 Changes (T1, 20) to (T1, 70)
The final Transaction Table has two entries: (T1, 70), and (T3, 60). The final Dirty Page Table has two entries: (P5, 20), and (P3, 30).

b. What is done during Redo? (Be precise about the points at which Redo begins and ends.)

Redo starts at LSN 20 (smallest recLSN in DPT).
LSN 20 Changes to P5 are redone.
LSN 30 P3 is retrieved and its pageLSN is checked. If the page had been written to disk before the crash (i.e. if pageLSN >= 30), nothing is re-done otherwise the changes are re-done.
LSN 40,50 No action
LSN 60 Changes to P3 are redone
LSN 70 No action

Note that for all of the changes that have to be redone (LSN 20, 30, and 60), the full check has to be done to see if the data values have been updated since then, as is detailed in the explanation for LSN 30.

c. What is done during Undo? (Be precise about the points at which Undo begins and ends.)

Undo starts at LSN 70 (highest lastLSN in TT). The Loser Set consists of LSNs 70 and 60. LSN 70: Adds LSN 20 to the Loser Set. Loser Set = (60, 20). LSN 60: Undoes the change on P3 and adds a CLR indicating this Undo. Loser Set = (20). LSN 20: Undoes the change on P5 and adds a CLR for this Undo.

Note: this is problem 18.3 part 2 from the book

Question 8. {4 marks}

Consider the following XML DTD:

```
    <?xml version="1.0"?>
<!DOCTYPE Newspaper[
        <!ELEMENT Basket (Cherry+, (Apple | Orange)*) >
            <!ELEMENT Cherry EMPTY>
               <!ATTLIST Cherry flavour PCDATA #REQUIRED>
            <!ELEMENT Apple EMPTY>
               <!ATTLIST Apple colour PCDATA #REQUIRED>
            <!ELEMENT Orange EMPTY>
               <!ATTLIST Orange location 'Florida'>

    ]>
```

Where #PCDATA and CDATA are equivalent.

Consider the following instance, which is designed to adhere to the above DTD.

```
     <Basket>
          <Apple/>
          <Cherry flavour='good'/>
          <Orange/>
     </Basket>
```

a. Is this instance valid? Why or why not?

No. For the instance to be valid, it would need to have one or more cherries first, plus it needs a colour for Apple.

b. Is this instance well-formed? Why or why not?

Yes. It is all parseable XML