The University of British Columbia

Computer Science 304

Midterm Examination

February 5, 2007

Time: 50 minutes   Total marks: 38

Instructor: Rachel Pottinger

Name ANSWER KEY ______________________ Student No______________

(PRINT) (Last) (First)

Signature____

<u>This examination has 6 pages.</u>

<u>Check that you have a complete paper.</u>

This is a closed book, closed notes exam. No books or other material may be used.

Answer all the questions on this paper.

Give very **short but precise** answers.

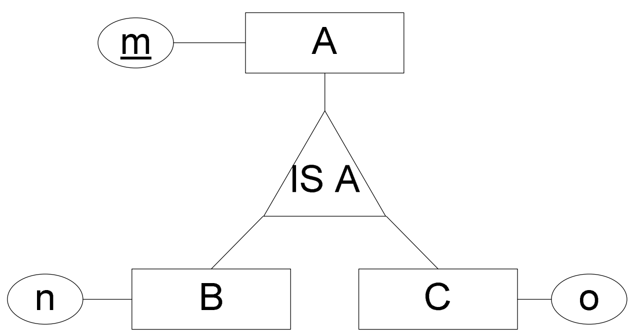
State any assumptions you make

Work fast and do the easy questions first. Leave some time to review your exam at the end.

Good Luck

Question	Mark	Out of
1		6
2		11
3		4
4		10
5		7

1. {6 marks, 1 mark per question} Circle only **one** answer per question – no points will be taken off for incorrect answers (i.e., you might as well guess):

a. If X is a key of a relation R, X is also a superkey of R *True. A superkey is a key plus zero or more additional attributes*	True False
b. A relationship in an ER diagram must be uniquely determined by the entities in that relationship *True. This is why a key of a relationship is at most the keys of the entities involved (though in cases like many-to-one relationships, we may not need all entities)*	True False
c. Every relation that is in BCNF is also in 3NF *True. BCNF is strictly more restrictive than 3NF*	True False
d.  A, B, and C above should be represented by two tables, B(m,n), C(m,o), in a corresponding relational schema if the IS-A relationship is partial *False. See relational slide 47- this method* cannot *be applied if the relationship is partial*	True False
e. MN→O, P→Q, Q→O is a minimal cover for the set of functional dependencies MN→O, P→Q, MN→Q, Q→O. *False. There's no way to derive MN→Q from the first list of dependencies. This question is isomorphic to a question in practice test 4*	True False
f. An insertion anomaly is when it may not be possible to store certain information unless some other, unrelated, information is stored as well *True. See book, page 607*	True False

2. {11 marks} Consider the schema R = (A, B, C, D, E) together with the functional dependencies:

AB→C
CD→A
C→E
C→B

a. {4 marks} What are the key(s) of R? Show your work to prove why each key is a key

Keys:
AB+=ABCE
CD+=CBDAE
C+=CEB
ABD+= ABDCE
Keys: ABD, CD

b. {5 marks} Is R(A, B, C, D, E) in **BCNF**? Why or why not? If not, decompose this relation into BCNF using the algorithm we covered in class and in the book; circle all answers in your final decomposition.

No. AB is not a key of R since AB+ = ABCE (and does not include D). So decompose on AB→C, which yields R1(A,B,C), and R2(A,B,D,E). Is R1 in BCNF? No, C→B violates BCNF for R1, since C is not a key of R1. Decomposing R1 on C→B yields R3(C,B), R4(C,A), both of which are in BCNF since they are both two attribute relations, and every two attribute relation is automatically in BCNF. What about R2? R2 is not in BCNF because the closure of AB = ABCE. When you project this to R2, we see that AB determines E but does not determine D. Therefore, this is a violation of BCNF, and must be decomposed to R5(A,B,E), R6(A,B,D). AB is a key of R5, so despite the projected dependency of AB→E, this is not a violation. R6 is in BCNF because the closure of AB does not include D (the only other attribute in R6), and this is the only interesting closure in R6 other than the trivial closure that the projection of ABD+ on R6 is ABD.

Thus the final decomposition is R3(C,B), R4(C,A), R5(A,B,E), R6(A, B, D)

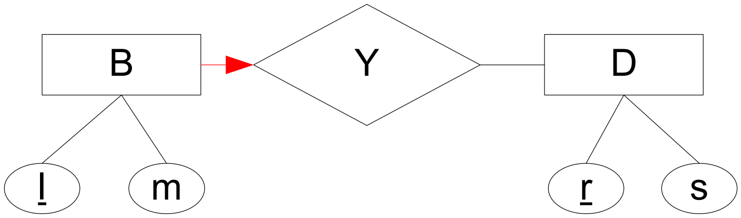
c. {2 marks} Is R(A, B, C, D, E) in **3NF**? Why or why not?

No. The functional dependency C→E violates 3NF because C is not a superkey, and E is not part of any key

3. {4 marks} ER relationship types.

a.

i. {1 mark} Change the ER diagram below so that B to D is a Many to One relationship



An arrow has been added from B to Y; see ER slide 12

ii. {1 mark} Give a set of entities for B and D that violate the above constraint, and explain why they violate it

B:

l	***m***
1	*2*

D:

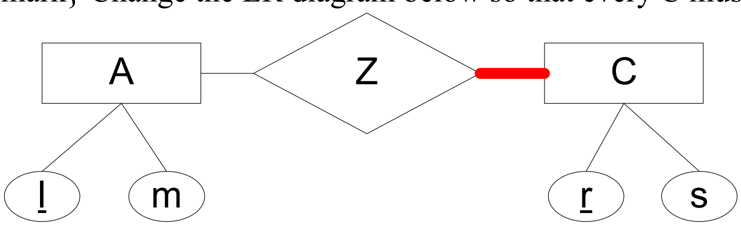
r	***s***
1	*2*
3	*4*

Where we have Y(l,r) = because l should determine r, and it does not

l	***r***
1	*1*
1	*3*

b.

i. {1 mark} Change the ER diagram below so that every C must participate in Z



Where the line between Z and C has been made thick (see ER slide 18)

ii. {1 mark} Give a set of entities for A and C that violate the above constraint, and explain why they violate it

A:

l	***m***
1	*2*

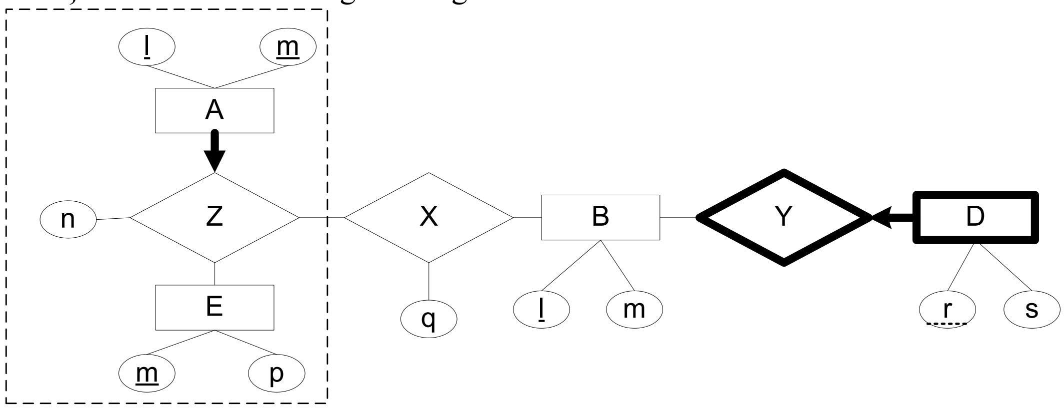
C:

r	***s***
1	*2*
3	*4*

Where we have Y(l,r) = because C(3,4) should participate in Y and it does not.

l	***r***
1	*1*

4. {10 marks} Given the following ER diagram:



Transcribe the ER diagram into a relational schema using the methods discussed in class/the book. State any assumptions that you make – but your assumptions cannot contradict the facts given.

a. {8 marks} Give the SQL DDL necessary to create the relational schema. You do *not* have to include types for any attributes

Tables:

AZ(l,m,n,e-m), E(m,p), X(l,m,q,b-l), DY(r,s,l), B(l,m)

```
CREATE TABLE AZ(
    l,
    m,
    n,
    e-m NOT NULL,
    PRIMARY KEY (l,m),
    FOREIGN KEY (e-m) REFERENCES E)
CREATE TABLE E(
    m,
    p
    PRIMARY KEY(m))
CREATE TABLE X(
    l,
    m,
    q,
    b-l
    PRIMARY KEY(l,m,b-l)
    FOREIGN KEY (l,m) REFERENCES AZ
    FOREIGN KEY (b-l) REFERENCES B)
```

```
CREATE TABLE DY(
    r,
    s,
    l,
    PRIMARY KEY(r,l),
    FOREIGN KEY(l) REFERENCES B
      ON DELETE CASCADE)
CREATE TABLE B(
    l,
    m,
    PRIMARY KEY(l))
```

b. {2 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?

No, all constraints in this diagram can be modeled without using assertions. Assertions are only needed for participation constraints on relationships that are not many-to-one, and the one participation constraint can be modeled using an "is not null" constraint.

5. {7 marks} Consider the following relation instance:

This question is isomorphic to question 2 in practice test 2

A	B	C
Eric	2	Dempster
Eric	2	ICICS/CS
Ting	3	ICICS/CS
Ting	3	Dempster
Ying	5	SUB
Ying	6	Koerner

a) {5 marks} Observe that B → A appears to hold with respect to the given instance. Check to see if all of the following dependencies hold with respect to the instance, *and give a reason if they do not*.

- A → B
  No, because of A = Ying.
- A → C
  No, because of A = Eric, for instance.
- B → C
  No, because of B = 2, for instance.
- C → A
  No because of C = Dempster, for instance.
- C → B
  No, because of C = Dempster, for instance.

b) {2 marks} Determine the minimum number of tuples that can be added to the above instance to invalidate B → A. Demonstrate your answer by showing example(s) of such tuple(s).
(1 mark) It takes two tuples to invalidate a functional dependency. Thus, the minimum number of tuples to add to the instance is 1.
(1 mark) We can add for instance the tuple <Ting, 2, Dempster> to the relation instance.