The University of British Columbia

Computer Science 304

Midterm Examination

January 30, 2012

Time: 50 minutes — Total marks: 40

Instructor: Rachel Pottinger

Name ANSWER KEY ______________________ Student No______________

(PRINT) (Last) (First)

Signature____

This examination has 3 double-sided pages.

Check that you have a complete paper.

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.

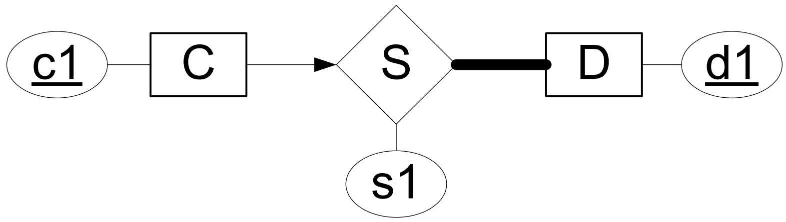
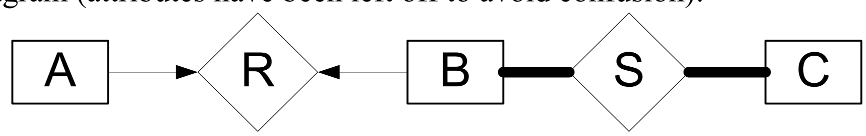
State any assumptions you make

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		10
2		8
3		4
4		8
5		10
Total		40

1. { 10 marks}

a.	All constraints on the following ER diagram can be translated into the relational model with what we know now  *False. We would combine C with S since it is many to one, however, while we can handle a total participation constraint on C by including "d1 is not null" in the ensuing relation, we can't handle this participation constraint*	*TRUE* *FALSE*
b.	Suppose that a1 and a2 are the only entities of A, b1 and b2 are the only entities of B, and c1 and c2 are the only entities of C. If T={(e1,f1)} means a relationship between e1 and f1 exists in relationship set, then R = {}; S = {(b1,c1),(b2,c1), (b2,c2)} is possible according to the following ER diagram (attributes have been left off to avoid confusion):  *True. Both sets of constraints are fine.*	*TRUE* *FALSE*
c.	A weak entity allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships *False; that's aggregation*	*TRUE* *FALSE*
d.	We cannot check a database instance to verify an integrity constraint. *False, the opposite is true*	*TRUE* *FALSE*
e.	In an ER diagram, the primary key of an entity is the key chosen as the principal means to identify entities in an entity set *True. From slides in chapter 3 on keys (34)*	*TRUE* *FALSE*

2. {8 marks} Consider the schema S(A, B, C, D, E,F) together with the functional dependencies:

   ABC→E
   ABC→D
   D→A
   A→E
   E→F

Is S in BCNF? Why or why not? If not, decompose into BCNF using the method shown in class and in the book; circle the answers in your final decomposition. If so, explain why it is in BCNF.

ABC+=ABCDEF
D+=DAEF
A+=AEF
E+=EF

Keys: ABC, BCD. Therefore, the final three FDs violate BCNF since the left hand sides of each are not superkeys for the relation

It is not a valid choice to decompose on ABC→E or ABC→D since these FDs do not violate BCNF. However, D→A violates BCNF because D is not a superkey of S. So decompose R1(AD), R2(BCDEF). R1 is in BCNF (as are all 2 attribute relations), but R2 is not, since, among other things, E is not a key of R2, but E→F is valid in R2. So decompose on E→F. Decomposing R2 yields R3(EF) and R4(BCDE). R3 is in BCNF. R4, however is NOT in BCNF, since D+ includes E, and thus D→E (even though it is not explicitly listed in the FDs), but D is not a key of R4. So decompose, yielding R5(BCD) and R6(DE). R6 is in BCNF, and R5 is in BCNF since there are no non-trivial functional dependencies in R5. Therefore, the final decomposition is R1(AD), R3(EF), R5(BCD),R6(DE).

3. {4 marks} Consider R(A,B,C,D,E) with functional dependencies
   C→D
   DE→B
   AB→C

   is R in 3NF? Why or why not? (Note: I have *NOT* asked you to decompose if R is not in 3NF.)

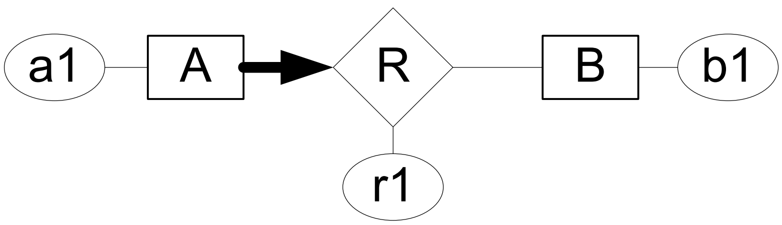
C+=CD
DE+=BDE
AB+=ABCD
Keys = ABE, ACE, ADE

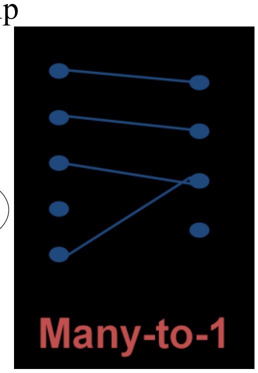
Therefore, R is in 3NF since the right hand side of each functional dependency is in 3NF

4. {8 marks}

For each part below, annotate the related diagram so that it provides the additional requested functionality– ***do not add any additional constraints beyond what is required***. If nothing needs to be done to the diagram or it is impossible to add that constraint in our version of ER diagram, state why. State any assumptions.

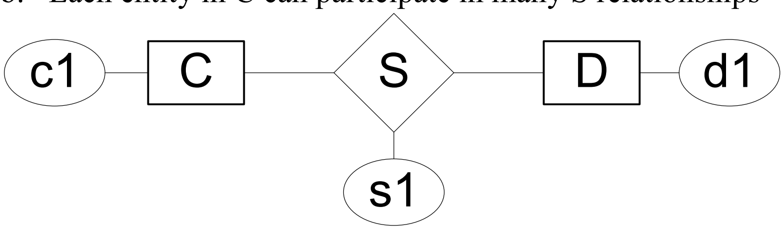
a. Each entity in A participates in exactly one R relationship





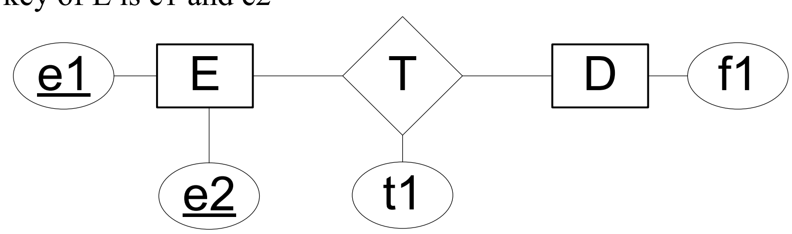
The thick line means that each A must participate in R (i.e., each A must participate in *at least one* relationship). The arrow says that this is on the many side of a many to one relationship (i.e., each A can be related to *at most one* B). So this combination says that A participates in exactly one R relationship.

b. Each entity in C can participate in many S relationships

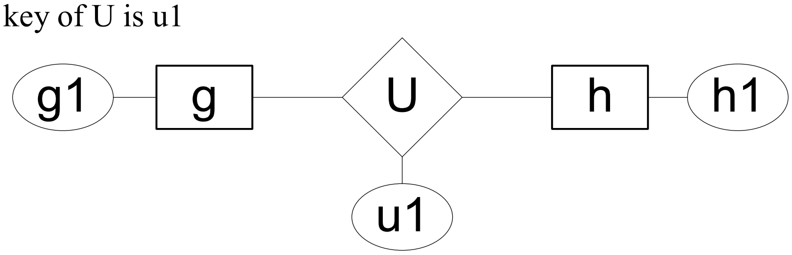


There is nothing to be added; without any restrictions, C is by default part of a many to many relationship.

c. The key of E is e1 and e2



d. The key of U is u1



Can be modeled by underlining u1(see slide on keys in relationships). I also would have taken that you cannot do it, since g and h must determine U. However, adding that U is 1:1 does not help.

5. {10 marks}

Suppose that we have a ternary relationship S between entity sets D, E, and F such that D has a key constraint and E has a key constraint and total participation; these are the only constraints. D has attributes d1 and d2, with d1 being the key; E has attributes e1 and e2, with e1 being the key; and F has attributes f1 and f2, with f1 being the key. S has no descriptive attributes. All attributes are integers. Write SQL statements that create tables corresponding to this information so as to capture as many of the constraints as possible. If you cannot capture some constraint, explain why.

This is isomorphic to problem 3.11 from the book.

Answer: The following SQL statements create the corresponding relations.

```
CREATE TABLE E_S (
e1 int,
e2 int,
d1 int,
f1 int,
PRIMARY KEY (e1),
UNIQUE (d1),
FOREIGN KEY (d1) REFERENCES D,
FOREIGN KEY (f1) REFERENCES F )
CREATE TABLE D (
d1 int,
d2 int,
PRIMARY KEY (d1) )

CREATE TABLE F (
 f1 int,
f2 int,
        PRIMARY KEY (f1) )
```

The first SQL statement folds the relationship S into table E and thereby guarantees the participation constraint on E, since the primary key of E_S cannot be null.