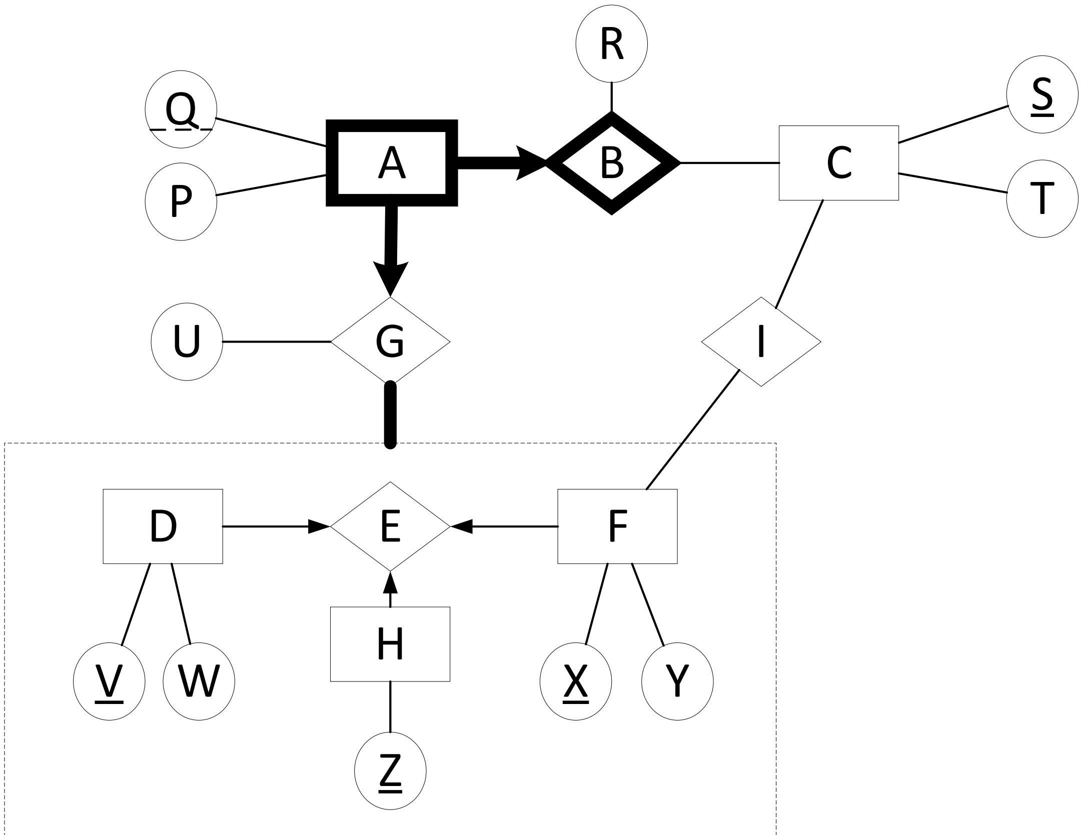
[4 marks] True/False Questions

Statement	True	False
If an attribute is a foreign key, it cannot also be a primary key. *False. An example of this would be the relation for any relationship set that has many to many constraints or the relation resulting from translating a weak entity set.*		
Superkeys cannot be minimal. *False. By definition, a superkey is any set of attributes that can uniquely determine all the attributes in a relation.*		
For R(ABCD), ABCD cannot be the primary key. *False. Primary keys have to be minimal and there is no rule against having all the attributes in a relation be part of the key. For example, when we translate relationships with many to many constraints, we often end up with a relation where all attributes are part of the key (assuming no relationship attribute).*		
For R(ABCD), if D is a candidate key, ABC cannot also be a candidate key. *False. If ABC contained D, that would not be minimal, but there is no restriction on the number of attributes in the candidate key."*		
Based on the rules of ER diagrams discussed in lecture, it is possible for two different entity sets to have attributes that share the same name. *True. E.g., you could have a Student entity set and a Faculty entity set where both have a single attribute called id.*		
Given an ISA relationship with partial and overlapping constraints, the best translation of this relationship into the relational model is to use two relations. *False. You need a relation to represent the parent as there are some entities in the parent entity set that might not participate in the child entities (and thus, need a place to be stored).*		
If an entity has a key and total participation constraint, it must be enforced through a NOT NULL constraint in the SQL DDL statement. *False. Weak entities have a total and key constraint but it is enforced through the fact that the weak entity's primary key contains the owner's primary key.*		
If an aggregated entity has a participation constraint with a relationship, the entities inside the aggregation must also have a participation constraint. *False. Participation constraints are not chained.*		

[5 marks] True/False Questions Part 2

Consider the following ER diagram:



For each of the following statements, clearly indicate whether it is true or false by putting a check mark in the associated box. Consider each statement individually and with regards to the rules/constraints discussed in class.

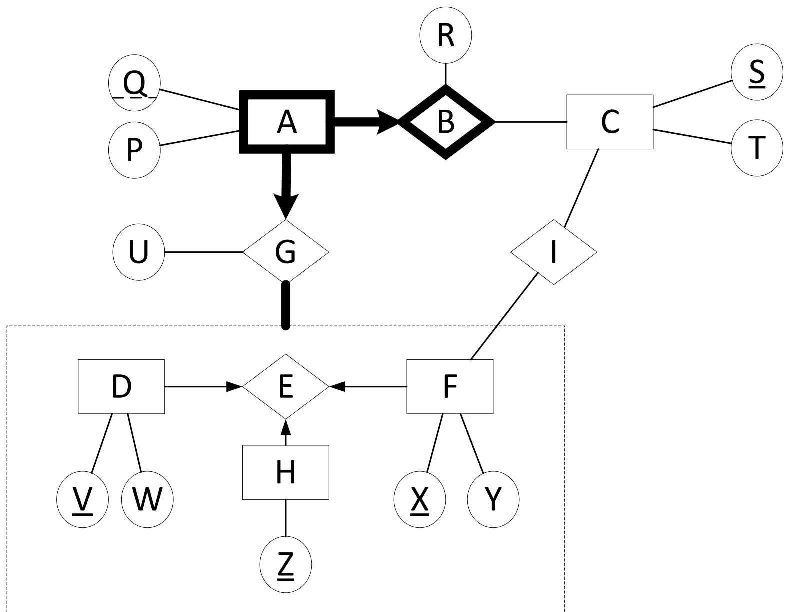
Statement	True	False
X is a candidate key of the relationship set E. *True. The candidate keys of relationship set E are V, Z or X.*		
If there are no entities in entity set D, then there are no relationships in relationship set I. *False. No entities in entity set D does not mean there are no entities in entity set F. The impact of no entities in D would mean that there are no relationships in in E or G.*		
If there is one entity in entity set A, there must be at least one entity in entity set D, H, and F. *True. Since all A's must participate in relationship set G, this means that there has be an associated E relationship. In order to have a E relationship, there has to be at least one entity in D, H, or F.*		

The number of entities in entity set A must always equal the number of relationships in relationship set G. *True. The key constraint on A means that "if we know A, we know what relationship we are talking about in G". Since every A must participate in G, then it follows that the number of relationships in G must equal A.*		
The number of entities in entity set H must always be equal to the number of relationships in relationship set G. *False. E has a many constraint with relationship G. This means that the same E relationship can show up multiple times in relationship set G. It is possible for there to be one relationship in E associated with many different A entities.*		
The number of entities in entity set H must always be equal to the number of relationships in relationship set E. *False. H does not have a total participation constraint so it is possible for there to be H entities that do not show up in E.*		
The number of entities in entity set C is always greater than or equal to the number of entities in entity set A. *False. There could be more than one A associated with each C.*		
There are always more entities in entity set A than there are in C. *False. You don't know and cannot tell from the ER diagram. It would be highly dependent on the data instance.*		
Relationship set I cannot be aggregated because the entity set F is already involved in another aggregation. *False. There is no constraint which prevents relationship set I from being aggregated.*		

[4 marks] Relational Model (True/False)

Consider the same ER diagram, repeated for your convenience, and assume that you have the following:

- a1 and a2 are the only entities of A
- c1 and c2 are the only entities of C
- d1 and d2 are the only entities of D
- f1 and f2 are the only entities of F
- h1 and h2 are the only entities of H



Which of the following relationship sets for B, E, or G are possible according to the diagram, where B={(a1,c1)} means that a relationship between a1 and c1 exists in relationship set B. To extend this to the aggregation G, G{{a1,E{d1, f1, h1}} means that a relationship between a1 and the relationship E(d1, f1, h1}} exists in the relationship set G.

Assume that any relationships not explicitly listed in the question are correct according to the diagram. Write true or false in the blank provided. Ambiguous or unreadable answers will be counted as incorrect.

Question	**True**	**False**
E = {{d1, f1, h1}, G{a1, E{d1, f1, h1}}, G(a2, E{d1, f1, h1}} *True. The total participation constraint on G means that every relationship in E must also appear in G, not that all the entities in D, H, and F must participate in E and also in G.*		
B = {{a1, c1}, {a1,c2}} *False. If we know A, then we know C.*		
B = {{a1,c1},{a2,c1}} *True. All A's are represented, and if we know A, then we know C.*		
E = {{d1,f1,h1}, {d1,f1,h2}} *False. It's not enough to have just one be determined, you have to determine all of them.*		

[22.5 marks] ER to Relational

Consider the ER diagram from the first question, repeated on the next page for your convenience. The following questions are about the translation of the ER diagram to the relational model.

To answer the questions, we recommend using the space on the next page to first fully translate the ER diagram to the relational model. As you are going, if you had to make any choices where there are other plausible options, you may find it helpful to keep track of these choices.

If you use a single relation to represent more than one entity or relationship, put the combined relation in FIRST appropriate row in the table, and put 'N/A' in each box for the rows for the remaining entities/relationships. For example, if you create a single relation to represent both "C" and "G", since "C" is first in the table, write your answers for the combined relation in the row for "C", and put "N/A" in each entry in row "G". If a relation does not have any entry for a box, write "N/A".

PLEASE ALPHABETIZE (I.E., PUT IN ALPHABETICAL ORDER) THE ATTRIBUTES IN A GIVEN CELL.

GRADING NOTES:
Please note that how to solve the problem in order to fill in the table below follows the table. If you read that, it will explain why the answers what they are.

For the option where it is "one of these", it is because there are three different options as to which entity should be combined with E. If you took the first of the options in the explanation below the table, then you should wind up with all of the top choices, etc.

The inconsistency between created relations resulted in a deduction of marks from the answer for entity set E.

Also note that some people marked some columns as blank rather than "N/A". These were counted as incorrect because (1) that was not following the directions and (2) it's impossible to tell someone who meant to tell when something was marked as blank to mean N/A and when they had no idea and just left it blank; hence the instructions.

The relation containing...	All attributes in the relation	Primary key attribute(s)	Foreign key attribute(s)	Attributes that have NOT NULL constraints that must be stated explicitly	Attributes that must be declared as unique
Entity Set A	P Q, R, S, U, Z	Q, S	S, Z	Z	N/A
	P,Q, R, S, U, X	Q, S	S, X	X	N/A
One of these:	P,Q, R, S, U, V	Q ,S	S, V	V	N/A
Entity Set C	S, T	S	N/A	N/A	N/A
Entity Set D	V, W	V	N/A	N/A	N/A
	V,W	V	N/A	N/A	N/A
	V,W,X,Z	V	X,Z	N/A	X, Z

One of these: (in the same order as above)					
Entity Set F	X,Y	X	N/A	N/A	N/A
	V,X,Y,Z	X	V,Z	N/A	V, Z
One of these: (in the same order as above)	X,Y	X	N/A	N/A	N/A

Entity Set H	V,X,Z	Z	V,X	N/A	V,X
	Z	Z	N/A	N/A	N/A
One of these: (in the same order as above)	Z	Z	N/A	N/A	N/A
Relationship Set B	N/A	N/A	N/A	N/A	N/A
Relationship Set E	N/A	N/A	N/A	N/A	N/A
Relationship Set G	N/A	N/A	N/A	N/A	N/A
Relationship Set I	S, X	S, X	S, X	N/A	N/A

To translate this ER diagram to the relational model, we start by creating initial tables for each entity. These will not be the final tables, but it is a good way to start the process:

A(<u>Q</u>, P) ←note: we will be changing this one! Don't worry that it doesn't make sense!
C(<u>S</u>, T)
D(<u>V</u>, W)
F (<u>X</u>,Y)
H(<u>Z</u>)

Next we start with the simplest relationships and try to form tables to model them. Because A is a weak entity that depends on B, we start with B rather than G.

The representation of B combines A and B and include the descriptive attribute of B, R. It also must include the key of C, S, so that it's clear what C is involved in B. However, since A is a weak entity, the key of C must also be included in the key of AB. Thus our representation of B is:

AB(<u>Q</u>, P, R , <u>S</u>)

And we no longer need the separate A table. We also do not need a not null constraint on S because it is part of AB's key and thus cannot be null.

Next we look at E. E is a one to one to one relationship. This means that any of the entities involved will uniquely determine the relationship involved, and we have to pick one of them to combine with E. Any of them will work. We choose the simplest one, and combine E with H.

Thus we wind up with EH(Z, V,X). Because there is no total participation constraint, we retain the separate D and F tables, but V and X must also be declared (separately) as keys using the unique constraint.

If we combine E with D, then we have ED(V, W, Z,X) and retain the separate H and F tables, but W and Z must also be declared (separately) as keys using the unique constraint.

If we combine E with F, then we have EF(X, Y, V, Z) and retain the separate H and D tables, but V and Z must be declared (separately) as keys using the unique constraint.

Next, we resolve the I relationship. It does not matter which entity we have combined with E, the resulting table is:

I(S, X)

This leaves us with a final relationship to deal with: G.

G has two entities/aggregations to connect to, AB and E/whatever entity you combined with. Because this is many to one, you have to combine the tables for AB and E/whatever you combined with. Because this is a total participation between G and A, then there has to be a not null constraint to whatever your key of A is. However, we cannot represent the total participation constraint between E and G without assertions.

Thus, there are three possible correct answers:

1. When we combine E and H:

C(S, T)
I(S, X); foreign keys: S, X
ABG(Q, P, R, S, Z, U), not null constraint on Z; (s is part of the key), Foreign keys: Z, S; S must not b; can't represent total participation with E

EH(Z,V,X), unique V, unique X
D(V,W)
F (X,Y)

2. When we combine E and F:

C(<u>S</u>, T)
I(<u>S</u>,<u>X</u>) foreign keys: S, X
ABG(<u>Q</u>, P, R, <u>S</u>, X, Y, U), not null constraint on X (S is part of the key), Foreign keys X, S; can't represent total participation with E

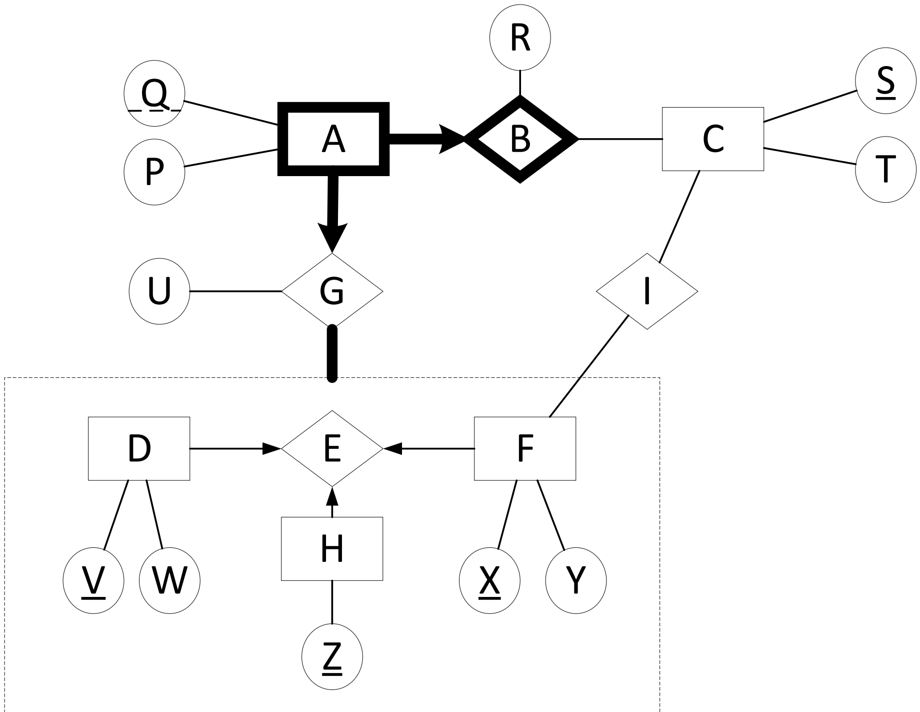
EF(<u>X</u>,Y,V,Z) unique Y, unique Z
D(<u>V</u>,W)
H(<u>Z</u>)

We cannot represent the total participation between A and G without assertions

3. When we combine E and D:

C(<u>S</u>, T)
I(<u>S</u>, <u>X</u>) foreign keys: S, X
ABG(<u>Q</u>, P, R, <u>S</u>, V, U), not null constraint on V (S is part of the key), Foreign keys, V, S; can't represent total participation with E
ED(X, Y, <u>V</u>, Z), unique X, Unique Z
H(<u>Z</u>)
F(<u>X</u>,Y)
We cannot represent the total participation between A and G without assertions

In all three possible cases, we combined one of the entity sets (H/F/D, no total participation) with the ternary relationship E, which makes it possible to have tuples with only entities (H/F/D) that do not participate in the relationship, thus NOT NULLs are not needed.



This space intentionally left blank so you can plan your translation from the ER diagram to the relational model.

[6 marks] Constraints in the Relational Model

Consider your translation from the ER to the relational model in the previous question. List as many answers as are correct in the box. If the answer is "none", then write "none" in the box. If there is more than one possible relational model, consider ***all*** possible relational models.

1. Which of the constraints cannot be represented in the relational model without assertions?

    1. The total participation constraint between A and B
    2. The total participation constraint between A and G
    3. The total participation constraint between E and G
    4. The total participation constraint between E and F
    5. The total participation constraint between E and D
    6. The total participation constraint between E and H
    7. The key constraint between A and B
    8. The key constraint between A and G
    9. The key constraint between E and G
    10. The key constraint between E and F
    11. The key constraint between E and D
    12. The key constraint between E and H

    Answer: 3

2. Which of the constraints must be represented in the relational model with a not null constraint?
    1. The total participation constraint between A and B
    2. The total participation constraint between A and G
    3. The total participation constraint between E and G
    4. The total participation constraint between E and F
    5. The total participation constraint between E and D
    6. The total participation constraint between E and H
    7. The key constraint between A and B
    8. The key constraint between A and G
    9. The key constraint between E and G
    10. The key constraint between E and F
    11. The key constraint between E and D
    12. The key constraint between E and H

    Answer: We removed this problem from being graded because it was covered by the table and made the exam weigh too heavily toward ER/relational model.

3. Which of the constraints must be represented in the relational model with a unique constraint?
   1. The total participation constraint between A and B
   2. The total participation constraint between A and G
   3. The total participation constraint between E and G
   4. The total participation constraint between E and F
   5. The total participation constraint between E and D
   6. The total participation constraint between E and H
   7. The key constraint between A and B
   8. The key constraint between A and G
   9. The key constraint between E and G
   10. The key constraint between E and F
   11. The key constraint between E and D
   12. The key constraint between E and H

Answer: We removed this problem from being graded because it was covered by the table and made the exam weigh too heavily toward ER/relational model.

[2 marks] Functional Dependencies

You are analyzing your peers' UBC learning experiences with their course choices and studying habits. You decided to create a database that will store students' information (ID, Name and GPA) and their overall feedback (Exp) about each course they took (Dept, Cnum, Sect, Prof, Mark).

You started from a relation R (ID, Name, Dept, Cnum, Sect, Prof, Mark, GPA, Exp) with the following relational instance:

ID	Name	Dept	Cnum	Sect	Prof	Mark	GPA	Exp
1	John Doe	CPSC	301	101	Prof. Smith	90	3.7	Excellent
2	Jane Smith	MATH	302	102	Prof. Smith	85	3.5	Interesting
3	Mike Brown	CPSC	301	102	Prof. Kim	75	3	Good
4	Sarah Lee	MATH	302	101	Prof. Kim	92	3.9	Informative
5	Bob Adams	CPSC	301	103	Prof. Smith	80	3.7	Excellent

What functional dependencies cannot be true given the instance above?
Write the letter(s) corresponding to your choices in the box below.

A. ID → Name, GPA — can be true; all IDs are different, so there is no contradiction
B. GPA, Prof → Exp — can be true; Smith and 3.7 both → Excellent
C. Prof → Dept — cannot be true; Smith is both in CPSC & MATH
D. Exp → Dept — Can be true. Both excellents → CPSC
E. Cnum → Dept — Can be true. All 301s are CPSC, and both 302s are MATH
F. Prof → Name, Dept — Cannot be true. Not true for either Prof.
G. All can be true

Answer: C, F

[3 marks] Minimal Keys

Consider the relation R (I, N, D, C, S, P, M, G, E) and the following FDs:

I –> N, G
I, N, D, C –> S
D, C, S –> P
P –> D
E –> D
G, P –> E
C, S, E –> P

Find all the minimal keys for R (I, N, D, C, S, P, M, G, E). Write your final answer in the box, but show your work below. Please list the attributes in alphabetical order within a key and the keys in alphabetical order.

Answer: ICDM, ICEM, ICMP

Left	Middle	Right
I, C, M	N, D, S, P, G, E	

ICM+ = {I, N, C, M, G}
ICMN+ = {I, N, C, M, G}
IDCM+ = {I, N, D, C, S, P, M, G, E}
ICSM+ = {I, N, C, S, M, G}
ICPM+ = {I, N, D, C, S, P, M, G, E}
ICMG+ = {I, N, C, M, G}
ICME+ = {I, N, D, C, S, P, M, G, E}

[12 marks] 3NF Normalization

1. [2 pts] Is the relation R (I, N, D, C, S, P, M, G, E) in 3NF?
   I –>N, G
   I, N, D, C –> S
   D, C, S –> P
   P –> D
   E –> D
   G, P –> E
   C, S, E –> P

   Explicitly state yes or no.
   Provide reasoning (approximately 1 to 4 sentences) to justify your point (based on the definition of the Third normal form).

No, let's take a FD: "I –> N G". 'I' is not a superkey, 'N' and 'G' are not part of the key.

2. [6 pts] Follow the 3-step process discussed in class to find a minimal cover of the set of FDs. Perform each step in the appropriate box below:

<table>
<tr><td>
Step 1

Put FD: “I –> N G” in standard form:
I –> N
I –> G

I N D C –> S
D C S –> P
P –> D
E –> D
G P –> E
C S E –> P
</td></tr>
<tr><td>
Step 2

IDC+ = {I, N, D, C, S, G} includes ‘N’, so we remove ‘N’ from “I N D C –> S”.
I D C –> S

D C S –> P
P –> D
E –> D
G P –> E
C S E –> P
</td></tr>
<tr><td>
Step 3

Without considering the FD: “C S E –> P”
CSE+ = {D, C, S, P, E}, so we can remove that FD.

I –> N
I –> G
I D C –> S
D C S –> P
P –> D
E –> D
G P –> E
</td></tr>
</table>

3. [4 pts] Use the **synthesis method** to bring R (I, N, D, C, S, P, M, G, E) into 3NF, while preserving all functional dependencies from the minimal cover (if you did not find one above, please use the given set of FDs). Write the resulting relations.

Using 3NF Synthesis:

R1 (I, N)
R2 (I, G)
R3 (I, D, C, S)
R4 (D, C, S, P)
R5 (P, D) <– repetitive and must be deleted
R6 (E, D)
R7 (G, P, E)
R8 (I, C, P, M) <– relation that contains all attributes of a key

[4 marks] BCNF Normalization

Consider a Relation S(A, B, C, D, E, F) with the following FDs:

C→B C+=CBF
C→F
AB→C AB+=ABCF
D→E D+=DE

Someone has attempted to start decomposing this into BCNF using the lossless join method by decomposing it into two relations, S1(A,B,C), S2(ABDEF). Answer the following questions:

a. [2 marks] Does this follow the lossless join decomposition rules? If so, why? If not, why not?

Yes. This is a valid decomposition based on the FD AB→C

b. [1 mark] Regardless of whether this is a correct decomposition or not, what are the keys of S2?

The only key of S2 is ABD

c. [1 mark] Regardless of whether this is a correct decomposition or not, is S1 in BCNF? State yes or no.

C→B holds in S1, and C is not a superkey of S1, so S1 is not in BCNF