Lab 3: Relational Database Design Theory (100 points)

Due Date: **Friday, Feb 7th (11:59 PM)**

Submission

All lab assignments should be turned in online, as a PDF file, through the associated (Lab3 in this case) in **Gradescope**. See the table below for Lab 3 submission opportunities. Note that after the last deadline, Saturday Feb 8th, no further Lab 3 submissions will be accepted at all. That is, we will not accept assignments after that time since we will be publishing the solution at that time. Please turn in all of your work on time! If possible, save your one dropped assignment for the end of the term when you are most likely to want/need it.

Date / Time	**Grade Implications**
Friday, Feb 7th (11:59 PM)	Full credit will be available
Saturday, Feb 8th (11:59 PM)	10% will be deducted

Submission note: please use the provided template to answer each question. In gradescope, mark the answer for each question so grading would be easier.

1) [25 Points] By looking at the PHLogger table:
    a) List all non-trivial functional dependencies.

phlid → name, address_street, address_city, address_state, address_pcode

    b) What is the highest normal form the PHLogger table is in currently?

BCNF highest normal form for the PHLogger table

c) The external consulting experts at DBInstructor, Inc., have noticed that city and state of an address can be inferred by its postal code (zip code). What new functional dependencies would be introduced by codifying this rule?

address_pcode → address_city, address_state

d) What is the highest normal form the PHLogger table is in after adding the new functional dependencies?

2NF with the new dependencies

e) Decompose the PHLogger table into multiple tables to the highest normal form possible.

Table 1: (phlid, name, email address, address_street, address_pcode)
Table 2: (address_pcode, address_city, address_state)

f) After decomposition, what is the highest normal form design that you could produce which is lossless and dependency preserving[3NF/BCNF]? Explain.

After decomposing the PHLogger table into (phlid, name, address_street, password, address_pcode) and (address_pcode, address_city, address_state), the highest normal form achieved is BCNF. This is because every determinant in both tables is a superkey, meaning no partial or transitive dependencies. The decomposition is lossless, because you can reconstruct the original table via joins, It is also dependency-preserving, as all functional dependencies remain intact within the new tables.

2) [25 points] Consider the following relation:

G	H	M
10	h1	m1
10	h2	m2
11	h4	m1
12	h3	m4
13	h1	m1
14	h3	m4

a) Given the current state of the database, for each one of the following functional dependencies answer

a) Does this functional dependency hold in the above relation instance [Yes/No]?

b) If your answer to previous question was no, explain why by listing a tuple that causes a violation.

i) G → H

No, because G=10 maps to both h1 and h2

ii) H →M

Yes, h1 gives the same value for m1 every time h maps to m the same all unique and consistent output

iii) M → H

No, because M=m1 maps to H=h1 and H=h4.

iv) H → G

No, because H=h1 maps to G=10 and G=13.

v) M → G

No, because M=m1 maps to G=10 and G=11.

b) List all potential candidate keys (if there are any) for the above relation.

H -> M
GM -> H
GH -> M

M can be determined by H or GH
GH is the potential candidate key.

3) [25 points] Considering the relation R(A,B,C,D,E) and the following functional dependencies, answer the questions.
   FD1: AB → C
   FD2: CD → E
   FD3: DE → B

A. List all the candidate keys.

(A,C,D)
(A,D,E)
(A,B,D)

B. What is the highest normal form that R satisfies and why?

The highest normal form would be 3NF. It can be 2NF as there are no partial dependencies, as well as it has transitive dependencies which are needed for 3NF of ab -> c and cd -> e which give prime attributes. Thus that means that it cannot be BCNF, but it is proven that it is 3nf

C. If R is not already at least in 3NF, then normalize R into 3NF and show the resulting relation(s) and their candidate keys. Your decomposition should be both join-lossless and dependency-preserving. If R is already in 3NF, just list the candidate keys of R.

(A,C,D)
(A,D,E)
(A,B,D)

D. Is your decomposition in BCNF as well?[Yes/No]. Explain.

No, because you would need a super key to determine all values which we do not have. In other words none of the candidate keys determine c whereas A,B can determine it

4) [25 points] Considering the relation R(A,B,C,D,E) and the following functional dependencies, answer the questions.

FD1: A → BC
FD2: BC → AD
FD3: D → E

A. List all the candidate keys.

(A), (B,C)

B. What is the highest normal form that R satisfies and why?

2NF as we can determine all values such as e by d but there is no direct path to d -> e thus it needs transitivity, this is as D is not a candidate key, thus it is not 3nf

C. If R is not already at least in 3NF, then normalize R into 3NF and show the resulting relation(s) and their candidate keys. Your decomposition should be both join-lossless and dependency-preserving. If R is already in 3NF, just list the candidate keys of R.

In order to get 3NF, you need to make R into

R1(A(PK), B, C, D) – who's Candidate keys are (A) and (BC).
R2(D(PK), E) – who's candidate key is D.

This becomes 3NF as it dependency-preserving, lossless, no redundancy, and every dependency can be determined by a primary key and is included. As well as all dependences can be accessed by the main primary key given A

D. Is your decomposition in BCNF as well?[Yes/No]. Explain.

Yes, R1(A(PK), B, C, D) has the candidate key A, This can be used to get BCD which can be in R1 and used with lossless join to get R2 as who's candidate key is D. With this join you can then determine all attributes including e the only one not directly accessed in R1, thus allowing all to be accessed with no loss and redundancies by the Candidate key A.