Last Name: Sharma First Name: Kavya Student ID: W1619309

1) [25 Points] By looking at the PHLogger table:
   a) List all non-trivial functional dependencies.
      - phlid (since phlid → name, address_street, address_city, address_state, address_pcode).

   b) What is the highest normal form the PHLogger table is in currently?
      - The multivalued attribute address gets broken down into other attributes (address_pcode, address_city, address_state). This means all the attributes are atomic (1NF). There are no partial dependencies (2NF) and all non-key attributes are dependent on the primary key (3NF).
      - Thus, the highest normal form is BCNF.

   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 since the new FD tells us that address_city and address_state are dependent on address_pcode which is a non-key attribute.

   e) Decompose the PHLogger table into multiple tables to the highest normal form possible.
      - PHLogger(phlid, name, address_street, address_pcode)
        - phlid → name, address_street, address_pcode
      - PCode(address_pcode, address_city, address_state)
        - address_pcode → address_city, address_state
      - Now both tables are in BCNF.

   f) After decomposition, what is the highest normal form design that you could produce that is lossless and dependency-preserving [3NF/BCNF]? Explain.
      - The decomposition is lossless since the tables can be joined due to address_pcode.
      - The decomposition is also dependency-preserving.
      - Thus, the highest normal form that is lossless and dependency-preserving after decomposition is BCNF.

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]?

i) G ⟶ H
No

ii) H ⟶ M
Yes

iii) M ⟶ H
No

iv) H ⟶ G
No

v) M ⟶ G
No

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

vi) G ⟶ H
(10, h2, m2)

vii) H ⟶ M
N/A

viii) M → H
(11, h4, m1)

ix) H → G
(13, h1, m1)

x) M → G
(11, h4, m1)

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

- (GH) uniquely identifies M.
- (GM) uniquely identifies H.

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.
      - The candidate key is (ABD), (ACD), and (ADE).

   B. What is the highest normal form that R satisfies and why?
      - 1NF: All the attributes are atomic.
      - 2NF: No partial dependencies.
      - 3NF: All attributes are prime and they are all superkeys.
      - Thus, the highest normal form 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.
      - N/A

   D. Is your decomposition in BCNF as well?[Yes/No]. Explain.
      - No, the decomposition is not in BCNF as well.

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.
    - The candidate keys are (A) and (BC).

B. What is the highest normal form that R satisfies and why?
    - 1NF: All the attributes are atomic.
    - 2NF: No partial dependencies.
    - 3NF: Violates 3NF since in the relation D ⟶ E, D is not a superkey and E is not a prime attribute.
    - Thus, the highest normal form is 2NF.

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.
    - R1(D,E)
        - This covers the D ⟶ E dependency. R1 is now in 3NF.
    - R2(A, B, C, D)
        - This covers the other 2 dependencies A ⟶ BC and BC ⟶ AD. R2 is now in 3NF.
    - Thus, both of the decomposed relations are lossless and dependency-preserving.

D. Is your decomposition in BCNF as well?[Yes/No]. Explain.
    - Yes, the decomposition is in BCNF as well.
    - R1:
        - For D ⟶ E, D is a superkey and so it is in BCNF.
    - R2:
        - For A ⟶ BC, A is a superkey and so it is in BCNF.
        - For BC ⟶ AD, BC is a superkey and so it is in BCNF.