Last Name: ZHOU First Name: YIHUA Student ID:83186710

1) [25 Points] By looking at the PHLogger table:

a) List all non-trivial functional dependencies.
Phild -> name
Phild -> street,
Phild -> city,
Phild -> state,
Phild -> pcode

b) What is the highest normal form the PHLogger table is in currently?
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?
Pcode -> city
Pcode -> state

d) What is the highest normal form the PHLogger table is in after adding the new functional dependencies?
2NF
It is sure that in 1NF, because all attributes are single atom. It is in 2NF is because the candidate key is only phild, so each of its non-prime attribute must not partially dependent on part of the candidate keys. Not in 3NF is because there are some transitive dependencies in non-prime attribute, like pcode to city and pcode to state, which violate the rule of 3NF. Not in 3NF then not in BCNF.
In conclusion, the highest normal from is 2NF.

e) Decompose the PHLogger table into multiple tables to the highest normal form possible.
PHLogger(<u>phild</u>, name, street, pcode)
Addr_code(<u>pcode</u>, city, state)

f) After decomposition, what is the highest normal form design that you could produce that is lossless and dependency preserving[3NF/BCNF]? Explain.
BCNF
All FDs are in each relation are now key constraints

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, there is single value in G that can match with multiple values in H
G = 10 can match H=h1 and H=h2

ii) H → M
yes

iii) M → H
No, there is single value in M that can match with multiple values in H
M = m1 can match H=h1 and H=h4a

iv) H → G
No, there is single value in H that can match with multiple values in G
H = h1 can match G = 10 and G = 13

v) M → G
No, there is single value in M that can match with multiple values in G
M = m1 can match H=h1 and H=h4

B. List all potential candidate keys (if there are any) for the above relation.
(G, M)
(G, 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.
(A,C,D)
(A,B,D)
(A,D,E)

B. What is the highest normal form that R satisfies and why?
1NF
It violates 2NF because there exits non-prime attribute that partially dependent on candidate key
(A,C,D), to determine E only needs C, D
(A, B, D), to determine C only needs A, B
(A,D,E) to determine B only needs D,E

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

D. Is your decomposition in BCNF as well?[Yes/No]. Explain.
Yes,
All FDs are now key constrains.

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)

(BC)

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

2NF

Because it violates the rule of 3NF which has transitive dependency between non-prime attribute

D -> E

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.

(B,C, A,D)

(D, E)

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

Yes

All attributes FDs are now key constraints