Name : Gargi Sontakke
G.No : ~~G0B1~~ G01334018.

Database Management Systems Homework 4.

Q 19.1

1. Define the term functional dependency

⇒ functional dependency is a constraint that specifies the relationship between two sets of attributes where one set can accurately determine the value of other sets.
It is denoted as $X \rightarrow Y$, where X is a set of attributes in a relational schema $\underline{R}$, capable of determining the value of Y.

2. Why are some functional dependencies called trivial?

⇒ The dependency of an attribute on a set of attributes is called trivial if the set of attributes include that attribute.
In other words, they contain an attribute that need not be listed.

$X \rightarrow Y$ is a trivial functional dependency, if Y is a subset of X.

for. eg :

roll_no	name	age.
50	A	20
51	B	21
52	C	23

{roll_no, name} → name is trivial, since name will always imply name.

5. $R(A, B, C)$, FD $\{B \rightarrow C\}$

A is the candidate key for R.

⇒

For a relation to be in Boyce-Codd normal form, it has to satisfy the following,

for every FD $X \rightarrow A$ in F

(i) $A \in X$, it is a trivial FD.

(ii) X is a superkey.

The FD given to us is $B \rightarrow C$, so for the relation to be in BCNF, B has to be the key for relation R.

6. $R(A, B, C)$, FD $\{A \rightarrow B \, , \, B \rightarrow A\}$.

Keys : A, B.

→ The above given information means that the relationship is one to one.

Here A corresponds to at most one entity in B and each B ~~co~~ entity corresponds to at most one A entity.

Q 19.2 $R(ABCDE)$

FD $\{A \to B, BC \to E, ED \to A\}$

Part (i) List all the keys for R.

→ ⓐ Consider the first FD, $A \to B$.

solving this,

$A \to B$ Result : $\{AB\}$

if we add C to both sides,

$AC \to BC$ Result : $\{ABC\}$

and from the FDs given $BC \to E$.

$\therefore AC \to E$ Result : $\{ABCE\}$

Now, if we add D to both sides, we get

$ACD \to ED$ Result : $\{ABCDE\}$.

and $ED \to A$

We can see that $ACD \to ED$ functional dependency gives us the whole relational schema ABCDE.

$\therefore$ ACD is a candidate key.

ⓑ similarly, consider $BC \to E$.

$BC \to E$ Result : $\{BCE\}$.

Adding D,

$BCD \to ED$ Result : $\{BCDE\}$.

from the given FDs, $ED \to A$. Result : $\{ABCDE\}$.

$\therefore BCD \to A$

BCD is another candidate key.

(c) $ED \rightarrow A$ Result : $\{ADE\}$

from given FD's, $A \rightarrow B$. Result : $\{ABDE\}$

so far we have $ED \rightarrow A$

$A \rightarrow B$.

$\therefore ED \rightarrow B$ from transitivity.

Adding C to both sides

$EDC \rightarrow BC$ Result : $\{ABCDE\}$.

$\therefore$ We have another key : EDC.

$\therefore$ The candidate keys for R (ABCDE) are ACD, BCD, EDC

Part (iii). For a relation to be in BNCF, $X \rightarrow Y$, X should be a superkey.

Here, none of A, BC or ED are a key.

$\therefore$ The relational schema R (ABCDE) is not in BNCF.

Q 19.5

1) (a) R1 (A, C, B, D, E)

FD { A → B, C → D }

$A^+ = \{AB\}$; $B^+ = \{B\}$ $C^+ = \{C, D\}$ $D^+ = \{D\}$ $E^+ = \{E\}$

$AC^+ = \{ABCD\}$ ..

Here, ABCDE is not in BNCF ~~as it does not form~~

ABCDE is in 1NF.

(b) BNCF decomposition :

$$
\begin{array}{ccc}
 & ABCDE & \\
\swarrow & \searrow & \rightarrow \\
AB & CD & ACE
\end{array}
$$

2) R2 (ABF) , FD { AC → F , ~~BF →~~ B → F }.

The relation is in 1NF.

Checking for 2NF,

~~As~~ here we cannot identify the primary key from the FDs, so it is not in 2NF.

∴ it cannot be in BCNF.

$AB^+ = \{A, B, F\}$, ∴ AB and BF is the primary key.

3) $R_3(A, D)$.

$D \to G$, $G \to H$.

$A^+ = \{A\}$ $D^+ = \{D, G, H\}$, $G^+ = \{G, H\}$, $H^+ = \{H\}$

Now $AD^+ = \{A, D, G, H\}$

This relation is in BNCF. as there are no FDs affecting the schema.

4. $R_4(D, C, H, G)$.

$A \to I$, $I \to A$.

It is already in BNCF as there are no FDs that affect the schema.

5. $R_5(AICE)$

It is in BNCF since there are no FDs that affect the schema.

Q 19.6 Relation schema S : (A, B, C)
tuples : (1,2,3), (4,2,3), (5,3,3).

1) A → B

from the given tuples we get

A → B.	Here two values in A are referring
1 → 2	to the same value in B.
4 → 2	But a single value in A cannot
5 → 3.	map to more than one value in A.

∴ We can say that A is functionally dependent on B, over schema S.

2) BC → A.

(B, C) → A.	Here (2,3) is referring to two different values (1,4) in A.
(2,3) → 1	∴ functional dependency does not
(2,3) → 4.	occur.
(3,3) → 5.	

3) B → C

B → C	Two or more diff values in B
2 → 3	refers to same value in A.
3 → 3	∴ B is functionally dependent on C

2) No, we cannot identify any dependencies that hold over S.

Q 19.7   R(A, B, C, D)

~~Past~~ (i) $C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$.

⇒ (a) $C \rightarrow D$: Result : {CD}.
$C \rightarrow A$: Result : {CA}.
$B \rightarrow C$: Result : {BC}.

But $C \rightarrow D$.
∴ $B \rightarrow D$ (using transitivity) Result : {BCD}.

Also $C \rightarrow A$
∴ $B \rightarrow A$ (using transitivity) Result : {ABCD}.

∴ we have the whole schema ABCD.
∴ Candidate key : B.

(c) The schema R(ABCD) is not BNCF ∵ $C \rightarrow D$ and $C \rightarrow A$ cause violations and dont contain a key.
∴ we have to decompose ABCD into set of BNCF relations.

ABCD → CD, CA, BC

CD: we got this from $C \rightarrow D$

CA: from $C \rightarrow A$

BC: we get this from (ABCD − AD)

∴ ABCD is decomposed into AC, BC, CD

(ii) $B \rightarrow C$, $D \rightarrow A$.

(a) $B \rightarrow C$ Result : $\{BC\}$

$D \rightarrow A$ Result : $\{DA\}$

none of these single valued attributes give us the whole schema

Adding D to $B \rightarrow C$

$BD \rightarrow CD$ $\therefore$ Result : $\{BCD\}$

$\because$ the result consists A D, we can use $D \rightarrow A$.

$\therefore BD \rightarrow ABCD$

$\therefore$ Candidate key : BD.

(c) The candidate key BD is not part of any FD, therefore the relation R(ABCD) is not in BNCF.

~~To preserve the~~ Hence, we need to decompose ABCD into set of relations

ABCD

$\swarrow \quad \downarrow \quad \searrow$

BC — we get this from $B \rightarrow C$

AD — this is obtained from $D \rightarrow A$

BD — we get this relation from ABCD − AC = BD.

The new relations we have are BC, AD and BD.

(iii) $ABC \rightarrow D$, $D \rightarrow A$

(a) $ABC \rightarrow D$ Result : $\{ABCD\}$

~~$D \rightarrow A$~~

$D \rightarrow A$ Result : $\{AD\}$.

Adding B to both sides

$BD \rightarrow AB$ Result : $\{ABD\}$.

Adding C to both sides.

$BDC \rightarrow ABC$ Result $\{ABCD\}$

∴ Candidate keys : ABC and BCD as we can get the whole schema from these.

(c) ABCD is not in BNCF since we have a FD $D \rightarrow A$ and D is not a key, even though we have ABC.

In order to get it in BNCF, we need to decompose.

$$\begin{array}{ccc} & ABCD. & \\ \swarrow & & \searrow \\ AD & & BCD. \\ \text{from } D \rightarrow A & & \text{from } ABCD - A. \end{array}$$

However, if we do this decomposition, we cannot preser[ve] the dependency $ABC \rightarrow D$ ∵ A and BC will be in two different tables.

∴ we cannot perform BNCF decomposition.

(iv) $A \to B$, $BC \to D$, $A \to C$

(a) $A \to \{B\}$ Result : $\{AB\}$.
$A \to C$ Result : $\{ABC\}$.
$BC \to D$. Result : $\{ABCD\}$.

Candidate key : A .

(c) $BC \to D$ violates BNCF since it does not contain the candidate key A .
To get the relation in BNCF, we decompose it into sets of relation

ABCD

BCD . ABC

we get this from $BC \to D$. $(ABCD - D)$.

So the new relations we have are BCD and ABC

(4) $AB \to C$, $AB \to D$, $C \to A$, $D \to \cancel{A}B$.

(a) $AB \to C$ Result : $\{ABC\}$

$AB \to D$ Result : $\{ABCD\}$.

$C \to A$ Result : $\{AC\}$

$BC \to AB$ Result : $\{ABC\}$

and $AB \to D$ Result : $\{ABCD\}$

$D \to B$ Result : $\{BD\}$

$AD \to AB$ Result : $\{ABD\}$.

and $AB \to C$ Result : $\{ABCD\}$

$C \to A$ Result : $\{AC\}$.

$CD \to AD$ Result : $\{ACD\}$.

and $D \to B$ Result : $\{ABCD\}$.

Candidate keys : AB, BC, AD, CD.

(b) $C \to A$ and $D \to B$ cause violations because it does not contain a key.

∴ We decompose it

(i) If we decompose it into AC, BCD, then it violates / does not preserve $AB \to C$ and $AB \to D$ ∵ A and B are in different tables.

(ii) We decompose it further into AC, BD and CD
But, in order to make up for lost dependencies, ABC and ABD, it loses the BNCF form.

∴ there is no BNCF decomposition possible.