

Name : Gargi Sontakke
G. No : 5033 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 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 . i.e. one can do it through A only

one can do it through B only & etc.

for. e.g. : roll-no name age.

50 A 20

51 B 21

52 C 23

$\{ \text{roll-no, name} \} \rightarrow \text{name}$ is trivial, since name will always imply name.

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

A is the candidate key for R.

\Rightarrow

For a relation to be in Boyce-Codd normal form,

it has to satisfy the following stated conditions

for every FD $A \rightarrow X \rightarrow Y$ in functional dependency set

i) if $A \rightarrow X$, it is a trivial FD.

ii) X is a superkey.

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

Let's assume A is the key for relation R.

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

Given that keys for A, B are unique just, draw results of

partial and total

\Rightarrow The above given information means that the relationship

is one to one. domain limit is $y \in X$

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

id	A	id
1	1	1
2	2	2
3	3	3

thus same value limited in domain $\leftarrow \{\text{entity, attr}\}$

entity algmt operators

Q 19.2

$R(A B C D E)$

FD $\{A \rightarrow B, BC \rightarrow E, ED \rightarrow A\}$ $A \leftarrow D \text{ (1)}$

$\{ABC\} : \text{illegal}, A \leftarrow A, \text{ 2nd min w.r.t}$

Part (i) List all the keys for R.

→ (a) Consider the first FD $A \rightarrow B$. See ref. of A.

Solving this,

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

If we add C to both sides,

$AC \rightarrow BC$ Result: $\{ABC\}$ probable

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

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

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

$ACD \rightarrow ED$ Result: $\{ABCDE\}$

ACD, ACD, ACD and $ACD \rightarrow A$ ref. of A is stable set.

We can see that $ACD \rightarrow ED$ functional dependency

Jelwaria gives us the whole relational schema $ABCDEF$.

$\therefore ACD$ is a candidate key

(b) Similarly consider $BC \rightarrow E$.

$BC \rightarrow E$ Result: $\{BCE\}$ ref. of E.

Adding D,

$BCD \rightarrow ED$ Result: $\{BCDE\}$

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

$\therefore BCD \rightarrow A$

BCD is another candidate key.

(iii) R(ABCDE)

① $ED \rightarrow A$ Result: $\{ADE\} \leftarrow \{B, C \leftarrow A\}$ (ii)
from given FD's, $A \rightarrow B$. Result: $\{ABDE\}$

so far we have $EDC \rightarrow A$ if ext. retrieval (i) \leftarrow
 $A \rightarrow B$. with parallel

$\{BA\} : EDC \rightarrow B$ from transitivity.

Adding C to both sides $\leftarrow BC \leftarrow CA$

$EDCA \rightarrow ABC$ Result: $\{ABCDE\}$.

$\{BCA\} : \text{trans} \leftarrow CA$..

∴ we have another key EDC .

$\{BCDA\} : \text{trans} \leftarrow CB \leftarrow CA$

∴ The candidate keys for R(ABCDE) are ACD, BCD, EDC

Part (ii) for a relation to be in 1NF, if $X \rightarrow Y$, X should be a superkey.

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

∴ the relational schema R(ABCDE) is not in 1NF.

$BCD \rightarrow E$ Result: $\{BCDE\}$

$\{BCDA\} : \text{trans} \leftarrow CD \leftarrow BC$

$A \leftarrow BCD$

∴ 1NF violation exists in ABC

Q 19.5

(a, A) & R (8)

$H \leftarrow \emptyset, \emptyset \leftarrow C$

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

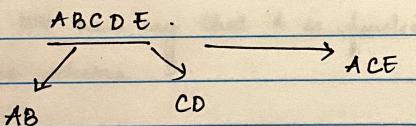
$$\{H\}^+ = H \quad \{H\text{ FD}\} \{A \rightarrow B, C \rightarrow D\} = \{A\}^+ = A$$

$$A^+ = \{AB\}, B^+ = \{B\}, C^+ = \{CD\}^+ = \{D\}^+ = \{E\} \\ AC^+ = \{ABCD\}$$

Here, ABCDE is not in BCNF. ~~as it does not form a relation with A~~
ABCDE is in INF.

$A \leftarrow I, I \leftarrow A$

b) BCNF decomposition:



2) R₂(ABF), FD {AC → E, B → F}.

The relation is in INF.

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) R₃ (A, D).

D → G, G → H.

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

(3, A, G, H, A) is ③ (1)

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

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

4. R₄ (D, C, H, G)

A → I, I → A.

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

5. R₅ (AICE)

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

$$A \rightarrow B \leftarrow C, B \rightarrow AC \Rightarrow \text{BCF. } (BA) \text{ is } ④ (1)$$

This is a mistake and

This is not valid

and meaning with primary domain see next

This is not a primary key, so it is wrong

This is not a primary key so it is :

just primary key is BA after BA ∴ $\{A, B\} = {}^+ BA$

Q 19.6 ~~Ans~~ Relation schema $S : (A, B, C)$ tuples : $(1, 2, 3), (4, 2, 8), (5, 3, 3)$. 2

1) $A \rightarrow B$

from the given tuples we get

$A \rightarrow B$. Here two values in A are referring

$1 \rightarrow 2$ to the same value in B.

$4 \rightarrow 2$ But a single value in A cannot

$5 \rightarrow 3$ map to more than one value in A.

\therefore we can say that A is functionally dependent on B, over schema S.

2) $BC \rightarrow A$.

$(B, C) \rightarrow A$. Here $(2, 3)$ is referring to two different values $(1, 4)$ in A.

$(2, 3) \rightarrow 1$ \therefore functional dependency does not

$(2, 3) \rightarrow 4$. occur

$(3, 3) \rightarrow 5$.

3) $B \rightarrow C$

$B \rightarrow C$ Two or more diff values in B

$2 \rightarrow 3$ refers to same value in A.

$3 \rightarrow 3$ \therefore B is functionally dependent on C

2) No, we cannot identify any dependencies that hold over S. $(S, S, 2), (S, S, 1), (S, S, 1)$: output

$S \leftarrow A(1)$

step 2: output word with morph

for word A we generate out soft $\cdot S \leftarrow A$

• S is either gross or at $\cdot S \leftarrow !$

for word A we generate alpha is and $\cdot S \leftarrow \alpha$

• A is either one vowel word at part $\cdot S \leftarrow e$

\rightarrow no sufficient information in A itself for us to tell if it's a vowel word

$A \leftarrow 28(2)$

out at generation is (S, S) left

• no $(p, !)$ sector therefore $\cdot A \leftarrow (2, 8)$

for word we get downshift $\therefore I \leftarrow (S, S)$

• $A \leftarrow (S, S)$

$I \leftarrow (S, S)$

$E \leftarrow (S, S)$

$S \leftarrow \emptyset(2)$

• A is either file name or out $\cdot I \leftarrow \emptyset$

• A is either name or noun $\cdot E \leftarrow \emptyset$

\rightarrow no sufficient information in S $\therefore E \leftarrow \emptyset$

$$A \leftarrow a, D \leftarrow d \quad (iii)$$

Q 19.7. $R(A, B, C, D)$ for 3 : Always $A \rightarrow D$ & D (ii)

$\{AD\}$: always $A \rightarrow D$

$\{AC\}$: always $A \rightarrow C$

Past (i) $C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$ is present in even

\Rightarrow (a) $C \rightarrow D$: Result : $\{CD\}$ start with see
 $C \rightarrow A$: Result : $\{CA\}$ $A \rightarrow D$ at D present
 $B \rightarrow C$: Result : $\{BC\}$ $AC \leftarrow BC$

But $C \rightarrow D$ is new, in A we have already seen it

$\therefore B \rightarrow D$ (using transitivity) Result : $\{BCD\}$.

Also $C \rightarrow A$

$\therefore B \rightarrow A$ (using transitivity) Result : $\{ABCD\}$.

Q8 : not stable
Q7 runs for group 2 as Q8 not stable (i)

we have the (whole) schema $ABCD$ with different

\therefore Candidate key : B.

Q8 : appears at even see, except with missing set
(ii) The schema $R(ABCD)$ is not in BCNF b/c $C \rightarrow D$ and $C \rightarrow A$

cause violations and don't contain a key.

\therefore we have to decompose $ABCD$ into set of BCNF relations.

ABCD

ABCD

Q8

int top see

as Q8

A is smart b/w A & D

A is smart b/w A & C

BC

Q8 = CA + Q8A
Q8 runs for group 2 as Q8 not stable

we got this

from $C \rightarrow D$ and see unstable

from $C \rightarrow A$

we get this from
 $(ABCD - AD)$

$\therefore ABCD$ is decomposed into AC, BC, CD

(ii)

$$B \rightarrow C, D \rightarrow A.$$

(a) $B \rightarrow C$ Result: $\{BC\}$ ($C, 3, 8, A$) R

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

none of these single-valued attributes give us the whole schema : $A \leftarrow C$ (A) \leftarrow Adding D to $B \rightarrow C$ $A \leftarrow C$ (A) \leftarrow

$$BD \rightarrow CD \quad \text{Result: } \{BCD\} \quad C \leftarrow S$$

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

$$\therefore \{BCD\} \text{ is BCDF (minimal prime)} \quad C \leftarrow S$$

$$\therefore \text{Candidate key: } BD \quad A \leftarrow C \text{ and } A$$

$$\{BCD\} : \text{BCDF (minimal prime)} \quad C \leftarrow S$$

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

To preserve the FNC, we need to decompose $ABCD$ into set of relations ($R(BC)$ and $R(AD)$ and $R(BD)$)

thus $R(BC)$ and $R(AD)$ suggests at most one relation

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 .
 $(CA \rightarrow D)$

BC, CA, DA are preserved in $R(BC)$ \therefore

(iii)

$$ABC \rightarrow D, D \rightarrow A \leftarrow A, C \leftarrow BC, B \leftarrow A$$

(iv)

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

~~$D \rightarrow A \{ \text{Result} : \{ ABCD \} \leftarrow A \}$~~

$D \rightarrow A \{ \text{Result} : \{ AD \} \leftarrow C \leftarrow B \}$

Adding B to both sides

$BD \rightarrow AB \text{ Result: } \{ ABD \} \leftarrow C \leftarrow B \text{ (stabilized)}$

Adding C to both sides

$BDC \rightarrow ABC \text{ Result: } \{ ABCD \} \leftarrow C \leftarrow B \text{ (stabilized)}$

A will stabilize with

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

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

In order to get set in BCNF, we need to decompose.

ABCD

~~get keys and FDs we want see in tables even with old~~

AD BCD

from $D \rightarrow A$ from $ABCD - A$.

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

∴ we cannot perform BCNF decomposition.

(iv)

$$A \rightarrow B, BC \rightarrow D, A \rightarrow CA \leftarrow d, d \leftarrow BC A$$

(a) $A \rightarrow \{B\}$ Result: $\{AB\} \leftarrow BC A$

$A \rightarrow C$ Result: $\{ABC\} \leftarrow A$

$\Leftarrow BC \rightarrow D$. Result: $\{ABCD\}$.

Candidate key

$BC \rightarrow D$ violates BNF since it does not contain the candidate key A .

To get the relation in BNF, we decompose it into sets of relation

$A \leftarrow d$ (it is a new relation)

$BCD \leftarrow ABC$ (it is a new relation)

we get this from $(ABCD \rightarrow d)$ and $BC \rightarrow D$.

So the new relations we have are BCD and ABC

$BCD \leftarrow ABCD - A$

$ABC \leftarrow ABCD - d$

second form is not correct as $BC = BD$.

Now as $d \rightarrow A$ & $d \leftarrow BC$ we can get $d \leftarrow BC$

which is correct as $d \leftarrow BC$

which is correct as $d \leftarrow BC$

(b) Given $AB \rightarrow CA$, $AB \rightarrow D$, $C \rightarrow A$, $D \rightarrow AB$.

(a) Given $AB \rightarrow C$ and Result : $\{ABC\}$.
 $AB \rightarrow D$ and Result : $\{ABCD\}$.

$C \rightarrow A$ and Result : $\{AC\}$

$BC \rightarrow AB$ and Result : $\{ABC\}$

∴ and $AB \rightarrow D$ and Result : $\{ABCD\}$

$D \rightarrow B$ and Result : $\{BD\}$

$AD \rightarrow AB$ and Result : $\{ABD\}$.

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

$C \rightarrow A$ and Result : $\{AC\}$.

$CD \rightarrow AD$ and Result : $\{ACD\}$.

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

Candidate keys : AB , BC , AD , CD .

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

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

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

\therefore there is no BNCF decomposition possible.

$\{ABC\}$: $B \leftarrow A$

$\{ABC\}$: $C \leftarrow B$

$\{ABC\}$: $A \leftarrow BC$

$\{BC\}$: $B \leftarrow C$

$\{BC\}$: $C \leftarrow B$

$\{BC\}$: $C \leftarrow BA$

$\{AC\}$: $A \leftarrow C$

$\{AC\}$: $C \leftarrow A$

$\{AC\}$: $C \leftarrow BA$

$\therefore CD, CA, CB, BA$: good candidates

which is required maintaining given $B \leftarrow C$ and $A \leftarrow C$ (i)

put a neutral item :

in requirement \therefore

1. choose the next, CB, CA which is required in (i)

& BA : $C \leftarrow BA$ and $C \leftarrow CA$ satisfying both requirement

selected requirement in (i)