classmate
Date 24/3/23
Page

FD's AND NORMALIZATION

Key's & FD: 1 to 39
Normal Form's: 40 to 56
Transaction & Con: 57 to 102
ER Model: 103 to 125
Relational Algebra: 126 to 142
SQL: 143 to 165
File Org. & indexing: 166 to 190

DBMS [Database Management System]

Data —— Facts, Raw material

↓

Information —— Meaningful data or Pre processed data

↓

Database —— Collection of logically related data or Collection of Similar records

↓

DBMS —— Set of programs (s/w) used to access & update the data in an efficient manner.

RDBMS : Relational Database Management System

└→ Relation → Table → Row [Tuple/Record], Column [field/attribute]

Student Table

Roll no	Name	Branch	Gender	CGPA
1	A	CS	M	9
2	B	IT	F	8
3	C	CS	M	7
4	D	IT	F	10
5	E	CS	M	9
6	F	IT	F	8

Arity (Degree) : No. of attributes

Cardinality : No. of Tuple/records

Relational Schema: Table Abstraction / Heading of Table

STUDENT (Roll no, Name, Branch, Gender, CGPA)

Relational Instance: Set of records at particular moments.

1	A	CS	M	9
2	B	IT	F	10
3	C	CS	M	8

$X \longrightarrow Y$ — X functionally determine Y

↑ determinant — determiner / dependent

Functional Dependency [FD] ($X \rightarrow Y$)

Let R be the Relation Schema, x and y be the attribute set of Relation R, t1 and t2 Any two tuple such that $X \rightarrow Y$.

$$\boxed{\text{If } t_1.x = t_2.x \text{ then } t_2.y = t_2.y \text{ must be same.}}$$

1.

x	y
1	5
2	5
3	5
4	7
5	8
6	9
7	8

~~2~~

x	y
1	8
2	9
3	4
4 →	5 ×
5	6
2	9
4 →	6 ×

3.

x	y
1	7
2	7
3	7
4	8
5	8
6	8

~~4~~

x	y
x_1 —	y_5 ×
x_2	y_6
x_3	y_7
x_4	y_8
x_1 —	y_9 ×
x_5	y_{10}

→ In $X \rightarrow Y$ whenever X value repeats then corresponding y value must be same.

Types of Functional Dependencies:

1. Trivial FD
2. Non-Trivial FD
3. Semi-Non Trivial FD.

1. Trivial FD → Always Valid

$X \rightarrow Y$ is trivial FD if and only if $\boxed{X \supseteq Y}$

R.H.S attribute must be part or equal of LHS attribute.

eg: AB → A

AB → B

AB → AB

2. Non Trivial FD

$X \rightarrow Y$ is non-trivial FD if $\boxed{X \cap Y = \phi \text{ and } X \rightarrow Y}$ must satisfy functional dependency definition.

eg:

A → B

Sid → marks

3. Semi Non Trivial FD

$X \rightarrow Y$ is semi non-trivial FD if $\boxed{X \not\supseteq Y \text{ and } X \cap Y \neq \phi}$

eg: AB → BC

AB

A B

Q1. Identify non-trivial FD which is satisfied by the equations/Instances

X	Y	Z
3	3	7
3	1	7
1	3	7
1	1	7
1	3	7

X → Y ✗   Y → X ✗   Z → X ✗

X → Z ✓   Y → Y ✓   Z → Y ✗

X → YZ ✗   Y → XZ ✗   Z → XY ✗

XY → Z ✓

YZ → X ✗

ZX → Y ✗

Ans: X → Z

Y → Z

XY → Z

Q2. Consider the following relation: Given the extension (state) which of the following dependencies may hold in the above relation? If the dependency cannot hold, explain why by specifying the tuples that cause the violation.

A	B	C	Tuple
10	b_1	c_1	1
10	b_2	c_2	2
11	b_4	c_1	3
12	b_3	c_4	4
13	b_1	c_1	5
14	b_3	c_4	6

FD satisfied by the instance
$\Rightarrow B \rightarrow C$

Q3.

A	B	C
1	1	1
1	2	1
2	1	2
2	1	3
1	3	3

FD satisfied by the instance
$\Rightarrow BC \rightarrow A$

Q4.

P	Q	R
6	6	7
6	7	7
7	3	4
8	3	4

FDs that are satisfied by the instance are
$P \rightarrow R$
$Q \rightarrow R$
$PQ \rightarrow R$

Q5.

A	B	C
7	5	6
7	7	6
7	5	7
7	7	7
9	5	6

0 FDs are satisfied by the above instance.

Q6.

A	B	C
2	2	4
2	3	4
3	2	4
3	3	4
3	2	4

FDs satisfied by the above instance:
$A \rightarrow C$
$B \rightarrow C$
$AB \rightarrow C$

Q7.

X	Y	Z
4	4	4
4	7	4
7	4	7
7	4	9
4	9	9.

Non trivial FDs satisfied by the instance : $YZ \to X$

Q8.

X	Y	Z
1	4	2
1	5	8
1	6	3
3	2	2

Functional dependencies satisfied by the instance : $YZ \to X$ and $Y \to Z$

Q9. From the following instance of a relation Schema R(A, B, C) we can conclude that :

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

$\Rightarrow$ We can conclude that B does not functionally determine C.

Note:

* Trivial FDs are always valid.
* Rule out the FDs based on the given data.

Q10. Consider the relation X(P, Q, R, S, T, U, V) with the following set of functional dependencies

$$F = \{ \{P, R\} \to \{S, T\}$$
$$\{P, S, U\} \to \{Q, R\} \}$$

Which of the following is the Trivial FD in F^+ is closure of F?

Ans $\{P, S\} \longrightarrow \{S\}$

Armstrong's Axioms / Inference Rules

* Axioms, or rules of inference, provide a simpler technique for reasoning about functional dependencies.
* In these rules we follow the Greek letters ($\alpha, \beta, \gamma ...$) for set of attributes.
* We can use the following three rules to find logically implied functional dependencies.
* By applying these rules repeatedly, we can find all of F^+ given F. This collection of rules called Armstrong's Axioms in honor of the person who first proposed it.

1. Reflexivity Rule : If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ holds.
2. Argumentation Rule : If $\alpha \rightarrow \beta$ holds and γ is a set of attributes, then $\gamma\alpha \rightarrow \gamma\beta$ holds.
3. Transitivity Rule : If $\alpha \rightarrow \beta$ holds and $\beta \rightarrow y$ (γ) then $\alpha \rightarrow y$ (γ) holds

Additional Rules

* If $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds (union).
* If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds (decomposition).
* If $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds then $\alpha\gamma \rightarrow \delta$ holds (Pseudo transitivity).

All the rules can be inferred from Armstrong's Axioms

→ Inference rules that can be used to infer new dependencies from a given set of dependencies:

IR1: (Reflexive Rule) : If $x \supseteq y$ then $x \rightarrow y$.

IR2: (Augmentation Rule) : $\{x \rightarrow y\} \models xz \rightarrow yz$

IR3: (Transitive Rule) : $\{x \rightarrow y, y \rightarrow z\} \models x \rightarrow z$

IR4: (Decomposition or Projective Rule) $\{x \rightarrow yz\} \models x \rightarrow y, x \rightarrow z$

IR5: (Union or additive rule) : $\{x \rightarrow y, x \rightarrow z\} \models x \rightarrow yz$

IRG: (pseudotransitive rule) : $\{x \to y, wy \to z\} \models wx \to z$

Attribute Closure $[x]^+$

R(ABCDE) [A → B, B → C, C → D, D → E]

$[A]^+ = [ABCDE]$

$[B]^+ = [BCDE]$

$[C]^+ = [CDE]$

$[D]^+ = [DE]$

$[E]^+ = [E]$

→ Let x be the attribute set of Relation R. Set of all possible attributes which are logically or functionally determined by x is called "Attribute Closure" of x $[x]^+$.

Example:

Q1. Let us consider a relation with attributes A, B, C, D, E and F. Suppose that the relation has the FD's AB → C, BC → AD, D → E, and CF → B.

What is the closure of {AB}, that is $[AB]^+$?

Ans $[AB]^+ = [ABCDE]$

Q2. F = { SSN → Ename, Pnumber → (Pname, Plocation), (Ssn, Pnumber) → Hours }.

Ans $\{Ssn\}^+ = [Ssn, Ename]$

$\{Pnumber\}^+ = [Pnumber, Pname, Plocation]$

$\{Ssn, Pnumber\}^+ = [Ssn, Pnumber, Ename, Plocation, Pname, Hours]$

Q3. R(ABCDEFG) F = { AB → C, BC → AD, D → E, E → G, CE → B)

Find Closure of :

(i) $[A]^+ = [A]$

(ii) $[B]^+ = [B]$

(iii) $[C]^+ = [C]$

(iv) $[D]^+ = [DEG]$

(v) $[E]^+ = [EG]$

(vi) $[F]^+ = [F]$

(vii) $[AB]^+ = [ABCDEG]$

(viii) $[BD]^+ = [BDEG]$

(ix) $[CE]^+ = [CEBGAD]$

(x) $[BC]^+ = [BCADEG]$

(xi) $[ACE]^+ = [ACEGBD]$

(xii) $[BDE]^+ = [BDEG]$

Q4. The follouwing functional dependencies are given:
$AB \to CD$, $AF \to D$, $DE \to F$, $C \to G$, $F \to E$, $G \to A$.
Which of the following options are false?

Ans $\{AF\}^+ = \{ACDEFG\}$
$\{AB\}^+ = \{ABCDFG\}$.

Q5. The following functional dependencies are given
$\{PQ \to RS, PU \to S, ST \to U, R \to U, U \to T, V \to P\}$
Which of the following options is/are true?

Ans $\{RU\}^+ = \{PRSTOU\}$ $\{QV\}^+ = \{PQRSU\}$.

Keys Concept.

eg: $R(ABCDE)$ $[A \to B, B \to C, C \to D, D \to E)$
$[A]^+ = [ABCDE] \longrightarrow$ A is a super key
$[B]^+ = [BCDE]$
$[C]^+ = [CDE]$
$[D]^+ = [DE]$
$[E]^+ = [E]$

Super Key: Let R be the Relation Schema and x be the attribute set of Relation R.
→ If $[x]^+$ determines all attributes of Relation R then x is a Super key

Or

→ If all attributes of Relation R is determined by the attribute Closure of x $[x]^+$ then x is a Super key.

Note: * Every key is super key.
* Any superset of super key is also super key.

eg2. $R(ABCDE)$ $[AB \to C,\ C \to D,\ D \to E]$

$[AB]^+ = [ABCDE]$

$[A]^+ = [A]$

$[B]^+ = [B]$

$\boxed{\text{AB is a super key.}}$

Super Key

→ Candidate Key (CK)

→ Primary Key (1 select as)

→ Alternative/Secondary Key (Remaining C keys except Primary key.)

Super Key: If all attribute of Relation R is determined by Attribute (attribute set) Closure of X $[X]^+$ then X is a super key.

* Every key [C.K, (P.K + A.K)] are super key.

* Every Superset of super key is also super key.

Candidate Key: Minimal of super key.

- If any proper subset of super key is also super key then that proper subset is called Candidate key (and so on)
- Superkey : Candidate Key

and

Candidate key + all other attribute Combination.

* Every Candidate key is a super key also but every super key is not a Candidate key.

* Because Candidate key is minimal of super key.

eg: 1 R(ABCDE) [AB → C, C → D, B → EA]

$[AB]^+ = [ABCDE]$

$\boxed{\text{AB is Super key.}}$

Proper subset.

B A

$[A]^+ = [A]$

$[B]^+ = [BEACD]$

Key / Prime attribute = [B]

Non key / Non prime attribute = [A, C, D, E]

$\boxed{\text{B is Candidate key.}}$

eg: 2 R(ABCDE) [AB → C, C → D, B → E]

$[AB]^+ = [ABCDE]$

$\boxed{\text{AB is super key}}$

A B

$[A]^+ = [A]$

$[B]^+ = [BE]$

Key / Prime attribute = [A, B]

Non Key Attribute = [C, D, E]

AB is ~~Super~~ Candidate key so AB is first Super key. AB + all attribute combination is also Super key.

$\boxed{\text{AB is Candidate key.}}$

→ From above example 1 B is a Candidate key so B is first Super key and B + All attribute combination is also Super key.

<u>Key / Prime Attribute</u>: Set of attribute which belongs or present in any Candidate key is called key or Prime attributes.

Non Key/Non Prime-Attribute: Set of attribute which does not belong or not present in any Candidate key is Called non key or non candidate Prime attribute.

Finding Candidate key: First find super key (any Super key) then Check minimal of that Super key.

R(ABCDE) [AB→C, C→D, D→E]

$[AB]^+ = [ABCDE]$ AB: Super key.

$[A]^+ = [A]$ AB is Candidate Key

$[B]^+$ [B]

→ Those attribute not present in right hand side (R.H.S) that attribute must be a part or present in Candidate key

Finding Multiple Candidate keys: First find any Candidate key, and that attributes (C.K attribute) is prime/key attribute.

$\boxed{\text{If } X_{\text{attribute}} \longrightarrow [\text{Prime/Key Attribute}]}$ then multiple C.K's are possible.

Q1. R(ABCDEF) {A→B, B→C, D→CEF} Find the Candidate keys for the relation R?

Ans $[A]^+ = [ABC]$ $[AD]^+ = [ABCDEF]$

$[D]^+ = [DCEF]$ AD is Candidate key

Check: If $X_{\text{attribute}} \longrightarrow$ [Prime attribute] no multiple C.K.S

∴ Only 1 CK (AD).

Q2 R(ABCDE) {AB→C, C→D, D→E, B→A, C→B} Find the Candidate keys for the relation R.

Ans $[AB]^+ = [ABCDE]$ $[A]^+ = [A]$

AB is Super key $[B]^+ = [BACDE]$ B is Candidate key.

Key/Prime attributes = B, C

If X attribute $\rightarrow$ [Prime Attribute]

$\therefore$ $C \rightarrow B$

$[C]^+ = [CBADE]$ C is Candidate key.

$\therefore$ Two Candidate keys : B and C.

Q3. R(ABCD) $\{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A\}$ Find Candidate keys for the relation R?

Ans

$[A]^+ = [ABCD]$

A is Candidate key — ①

$D \rightarrow A$

$[D]^+ = [DABC]$

D is a candidate key — ②

$C \rightarrow D$

$[C]^+ = [C, D, A, B]$

C is Candidate key — ③

$B \rightarrow C$

$[B]^+ = [BCDA]$

B is Candidate key — 4

Prime Attributes = {A, D, C, B}.

$\therefore$ Candidate keys are : A, B, C, D.

Q4. R(ABCDEF) $\{A \rightarrow BCDE, BC \rightarrow AD, D \rightarrow EF\}$

Find the Candidate keys for the relation R

Ans: $[A]^+ = [ABCDEF]$

A is Candidate key.

Key/prime attributes = [A, B, C]

If X attribute $\rightarrow$ [Prime Attribute]

$BC \rightarrow AD$

$[BC]^+ = [BCADEF]$

$[B]^+ = [B]$

$[C]^+ = [C]$

BC is Candidate key

$\therefore$ Candidate keys are:

1. A
2. BC

Q5. R(ABCDE) : $\{AB \rightarrow C, BC \rightarrow D\}$

Find the Candidate keys for the relation R.

Ans. $[AB]^+ = [ABCD]$, E not present in FD.

Note: Whenever any attribute not present in FD then make a part of (Add in) Candidate key.

$[ABE]^+ = [ABCDE]$

$[AB]^+ = [ABCDE]$ ∴ ABE is the only Candidate key

$[AE]^+ = [AE]$

$[BE]^+ = [BE]$

Q6. Consider the following relational Schema R(ABCDEF) with functional dependency {$AB \rightarrow C$, $C \rightarrow D$, $D \rightarrow E$, $E \rightarrow F$, $F \rightarrow B$} The no. of Candidate keys for relation R are?

Ans. $[AB]^+ = [ABCDEF]$ Key/prime attribute = [A, B, C, D, E, F]

$[A]^+ = [A]$, $[B]^+ = [B]$

AB is Candidate key — ①

If X attribute → [prime/attribute]

$F \rightarrow B$

$[AF]^+ = [AFBCDE]$

$[F]^+ = [FB]$

AF is Candidate key — ②

$E \rightarrow F$

$[AE]^+ = [AEFBCD]$

$[E]^+ = [EFB]$

AE is Candidate key — ③

$C \rightarrow D$

$[AC]^+ = [ABCDEF]$

$[C]^+ = [CDEFB]$

AC is Candidate key — ④

$D \rightarrow E$

$[AD]^+ = [ABCDEF]$

$[D]^+ = [DEFB]$

AD is Candidate key — ⑤

∴ Total Candidate keys: 5.

AB, AE, AF, AD, AC.

Q7. R(ABCD) F: {$AB \rightarrow C$, $B \rightarrow D$, $C \rightarrow B$, $D \rightarrow B$}

Find all Candidate keys of relation R.

Ans $[AB]^+ = [ABCD]$ Prime/key attribute = [A, B, C, D]

$[A]^+ = [A]$ $[B]^+ = [BD]$

AB is Candidate key — ①

$C \rightarrow B$

$[AC]^+ = [ACDB]$

$[C]^+ = [CBD]$

AC is Candidate key — ②

$D \rightarrow B$ ∴ Candidate keys are:

$[AD]^+ = [ADBC]$ AD, AB, AC

$[D]^+ = [DB]$

AD is Candidate key - 3

Q8. R(ABCDE): {AB → C, BC → D} Find Candidate keys.

Ans $[AB]^+ = [ABCD]$

E is not present in FD

$[ABE]^+ = [ABCDE]$

∴ ABE is the only Candidate key.

Q9. R(ABCDEFG): {A → B, B → C, C → D, D → E, F → G}

Find Candidate keys for the relation R.

Ans $[A]^+ = [ABCDE]$ Prime/Key attributes = [A, F]

$[F]^+ = [FG]$ non prime attributes = [B, C, D, E, G]

$[AF]^+ = [ABCDEFG]$

AF is Candidate key.

Q10. R(ABCDEFGH): {AB → C, AC → B, AD → E, B → D, BC → A, E → G} Find the Candidate keys for the Relation R.

Ans $[AB]^+ = [ABCDEG]$ Prime attributes = [F, H, A, B, C]

F, H not present in FD then make a pair of CK

$[ABFH]^+ = [ABCDEFGH]$

ABFH is a Candidate key - 1 ∴ Candidate keys = 3.

BC → A AC → B ABFH

[BCFH] - 2 [ACHF] - 3 ACHF

BCFH

Q11. R(ABCDPTL) {B → PT, T → L, A → D} Find the Candidate keys.

Ans $[B]^+ = [BPTL]$ Prime/Key attribute = [A, B, C]

$[A]^+ = [AD]$

C is not present So add it. ∴ ABC is the only Candidate key.

Q12. R(ABCDEFGHIJ) ; {AB → C, A → DE, B → F, F → GH, D → IJ}
Find the Candidate keys for relation R.

Ans [AB]$^+$ = [ABCDEFGHIJ]
[A]$^+$ = [ADEIJ]
[B]$^+$ = [BFGH]
AB is Candidate key

key/prime attributes : [A, B]
non prime = [C, D, E, F, G, H, I, J]

Q13. R(ABCDEFG) {A → B, B → A, C → D, D → E, E → FG}
Find the Candidate key for the relation R.

Ans [AC]$^+$ = [ABCDEFG]
[A]$^+$ = [AB]
[C]$^+$ = [CDEFG]
AC is Candidate key — 1

Key attribute = [C, A, B]
B → A
[BC]$^+$ = [BCADEFG]
[B]$^+$ = [BA]
BC is the Candidate key — 2

∴ Candidate keys are: AC, BC

Q14. R(ABCDEFG) {AB → CDEF, C → ADE, D → EBF, F → DA, BE → AF}

Ans G is not part of FD, so must be present in each Fk.
Candidate keys are: ABG, BEG, FG, DG, CG } 5
Out of 5 select one as primary key
Primary key = (unique + not null)

Q15. R(ABCDEFGH) {A → BC, B → DEF, DE → AGH} Find Candidate keys.

Ans [A]$^+$ = [ABCDEFGH]
A is Candidate key — 1

Prime attributes = [A, B, E, D]
B → DEF
[B]$^+$ = [BDEFACGH]
B is Candidate key — 2

DE → AGH
[DE]$^+$ = [DEAGHBFC]
[D]$^+$ = [D]
[E]$^+$ = [E]
DE is Candidate key — 3

∴ Candidate keys are: 3
A, B, DE
composite key

Q16. R(ABCDE) {A → BC, CD → E, B → D, E → A} Find Ckeys.

Ans. $[A]^+ = [ABCDE]$ Prime attributes = [A, E, C, D, B]

A is Candidate key - 1

CD → E

E → A

$[CD]^+ = [CDEAB]$

$[E]^+ = [EABCD]$

$[C]^+ = [C]$, $[D]^+ = [D]$

E is Candidate key - 2

CD is Candidate key - 3.

B → D

Total Candidate keys are: 4

$[CB]^+ = [ABCDE]$

A, E, CD, CB.

$[B]^+ = [BD]$

CB is Candidate key - 4.

Q17 R(ABCDEFGH) {AB → CD, D → EG, F → H, C → EF, H → A, G → B, A → B} Find Candidate keys.

Ans Total Candidate keys : 4 : [A, H, F, C]

GATE PYQs

1. Consider the relation Schema R(A,B,C) with the following functional dependencies {AB → C, C → A} Determine minimal keys.

2 Candidate keys {AB, BC}.

2. Let R = (A,B,C,D,E,F) be a relation Schema with the following dependencies C → F, E → A, EC → D, A → B. Which of the following is a key for R?

$[EC]^+ = [ABCDEF]$ ∴ EC

3. The relation Schema Student Performance (name, CourseNo, rollNo, grade) has the following functional dependencies

name, Course No → grade

rollNo, CourseNo → grade
name → rollNo
rollNo → name   Find Candidate keys.
2 Candidate keys : {RollNoCourseNo, Name CourseNo}

4. Consider the relation Schema R = {A, B, C, D, E, H} on which the following FDs hold {A → B, BC → D, E → C, D → A}
What are Candidate keys of R?
AEH, BEH, DEH.

5. Let R (ABCDEFPG) be a relational Schema in which of the following functional dependencies are known to hold
{A → BCD, DE → P, C → E, P → C, B → G} Find Candidate keys.
AF is Candidate key

6. Consider a relation R with five attribute V, W, X, Y and Z. The following functional dependencies hold: VY → W, WX → Z, and ZY → V. Which of the following is a candidate key.
$[VXY]^+ = [VXYWZ]$ ∴ VXY is a Candidate key

7. Relation R has eight attributes ABCDEFGH. Fields of R contain Only atomic values.
F = {CH → G, A → BC, B → CFH, E → A, F → EG} is a set of functional dependencies (FDs) so that F is exactly the set of FDs that hold for R. How many Candidate keys does the relation R have?
Total Candidate keys : 4 {AD, ED, FD, BD}.

8. Consider the relation Schema R = {E, F, G, H, I, J, K, L, M, N} and the set of functional dependencies {EF → G, F → IJ, EH → KL, K → M, L → N} on R. What is the key of R?
$[EFH]^+ = [EFHGIJKLMN]$ → Candidate key.

31/7/24

9. A prime attribute of a relation Scheme R is an attribute that appears
In some Candidate key of R.

10. Which of the following is NOT a Super key in a relation Schema with attribute V, W, X, Y, Z and primary key VY?
VWXY → VY is not present so even VWXY is not Candidate key

Membership Set:

F: [......]

→ Let F be the given FD. Any X→Y is a member of FD set F if X→Y logically implied in F.

→ X→Y Logically implied means from the closure of X determine Y
$[X]^+ = [\ldots Y]$ X→Y logically implied / member / valid FD

eg: F: [A→B, B→C]
Check A→C member / valid FD / logically implied or not?
$[A]^+ = [ABC]$ ∴ Yes

Q1. F: [AB→C, C→D, D→E, E→G]
Check: (i) A→C $[A]^+ = [A]$
~~(ii)~~ C→G $[C]^+ = [CDEG]$
~~(iii)~~ AB→G $[AB]^+ = [ABCDEG]$
(iv) B→E $[B]^+ = [B]$

Q2. F: [AB→C, A→DE, B→F, F→GH, D→IJ]
~~(i)~~ A→I $[A]^+ = [ADEIJ]$
(ii) B→I $[B]^+ = [BFGH]$
~~(iii)~~ B→GH
(iv) D→EF $[D]^+ = [DIJ]$
(v) F→J $[F]^+ = [FGH]$
~~(vi)~~ AB→F $[AB]^+ = [ABCDEFGHIJ]$
Only 1, 3, 6 are valid FD for this FD set.

Q3. In a Schema with attribute A, B, C, D and E following set of functional dependencies are given:

$\{A \to B,\ A \to C,\ CD \to E,\ B \to D,\ E \to A\}$

Which of the following functional dependencies is not implied by above set?

Ans: $BD \to CD$ is not implied by the above set.

Q4. Suppose the following functional dependencies hold on a relation U with attribute P, Q, R, S and T: $\{P \to QR,\ RS \to T\}$.

Which of the following functional dependencies are inferred/implied from the above functional dependencies?

Ans:
$PS \to T$ $[PS]^+ = [PSQRT]$ ✓
$P \to R$ $[P]^+ = [PQR]$ ✓
$PS \to Q$ $[PS]^+ = [PSQRT]$ ✓

Equality between 2 FD Set:

F[....] G:[.....]

→ F and G are equals only if: F cover G = True
G cover F = True

→ F cover G: F covers all the FD's of G FD set.

Or

All G FD's should be logically implied in F FD set.

→ G cover F: G covers all the FD's of F FD set.

Or

All F FD's should be logically implied in G FD set.

F Cover G:	True	False	True	False
G Cover F:	False	True	True	False
	$F \supset G$	$G \supset F$	$F = G$	Uncomparable

eg. F : [AB → CD, B → C, C → D]
G : [AB → C, AB → D, C → D]

F cover G

✓ AB → C [AB]⁺ = [ABCD]
✓ AB → D [AB]⁺ = [ABCD]
✓ C → D [C]⁺ = [CD]

} True

G cover F

✓ AB → CD [AB]⁺ = [ABCD]
× B → C [B]⁺ = [B]
✓ C → D [C]⁺ = [CD]

} False

$\boxed{F \supseteq G}$

Q5. Consider the relation Schema A (PQRS) with two set of FD's
F : [P → Q, PQ → R, PR → S, Q → R, Q → P]
G : [PQ → S, PR → Q, Q → S, QS → R]
Which of the following is correct?

Ans

F cover G

PQ → S [PQ]⁺ = [PQRS]
PR → Q [PR]⁺ = [PRQS]
Q → S [Q]⁺ = [QRPS]
QS → R [QS]⁺ = [QSPR]

True

G cover F

P → Q [P]⁺ = [P]
PQ → R [PQ]⁺ = [PQRS]
PR → S [PR]⁺ = [PRQS]
Q → R [Q]⁺ = [QSR]
Q → P [Q]⁺ = [QSR]

False

∴ F cover G is correct.

HW

Q6. Consider a relation Schema R (ACDEH) with two set of FD's.
F : [A → C, AC → D, E → AD, E → H]
G : [A → CD, E → AH]
Which of following is correct?

F cover G

✓ A → CD [A]⁺ = [ACD]
✓ E → AH [E]⁺ = [EADHC]

True

G cover F

A → C [A]⁺ = [ACD]
AC → D [AC]⁺ = [ACD]
E → AD [E]⁺ = [EAHCD]
E → H [E]⁺ = [EAHCD]

} True

(MSQ) ∴ F cover G, G cover F, F & G are equivalent

classmate
Date 01/8/23
Page

Roll no. CGPA Branch Fathername

R(A B C D E F G)

[Table] Name Address Gender

[A→B, A→C, A→F, F→D, A→EG]

$[A]^+ = [ABCDEFG]$

A (Roll no) is Candidate key

Minimal Cover :

Objective of the minimal cover is to eliminate or reduce the redundant FD (extra FD).

Redundant FD : R.F.D is a FD, if we delete that FD from original FD set, then after deletion does not affect the power of FD set.

eg: F: {A→B, B→C, A→C}

A→C is a Redundant FD

G: {A→B, B→C}

$[A]^+ = [ABC]$ so A→C is extra (redundant)

Minimal/Canonical Cover: (Formal defination).

Sets of functional dependencies may have redundant dependencies that can be inferred from others

Q. How to Cross Check your answer is correct or not?

→ Check Equality F Cover G: True if F=G then answer is correct.

G Cover F: True

F: Given FD set (Question)

G: Minimal Cover of F (Minimal Cover of F).

Q.1 $F: \{A \to B,\ B \to C,\ A \to C\}$.

Assume
(i) Let $A \to B$ is R.F.D
$G: [B \to C,\ A \to C]$

F cover G: $B \to C$, $A \to C$

G cover F: × $A \to B$, $B \to C$, $A \to C$

{False.}

∴ $A \to B$ is not R.F.D.

Let's Assume
(ii) $B \to C$ is R.FD
$G: [A \to B,\ A \to C]$

F cover G: $A \to B$, $A \to C$

G cover F: $A \to B$, × $B \to C$, $A \to C$

{False}.

∴ $B \to C$ is not R.F.D

Let's Assume
(iii) $A \to C$ is R.F.D
$G: [A \to B,\ B \to C]$

F cover G: $A \to B$, $B \to C$

G cover F: $A \to B$ ✓, $B \to C$ ✓, $A \to C$ ✓

{True}.

∴ $F = G$

$A \to C$ is R.F.D.

So $A \to B$, $B \to C$ is minimal cover.

Q2. $AB \to C$, $D \to E$, $E \to C$ is a minimal cover for the set of functional dependencies $AB \to C$, $D \to E$, $AB \to E$, $E \to C$.

Ans. $F: [AB \to C,\ D \to E,\ AB \to E,\ E \to C]$
$G: [AB \to C,\ D \to E,\ E \to C]$

F cover G

$AB \to C$ $[AB]^+ = [ABCE]$
$D \to E$ $[D]^+ = [DEC]$
$E \to C$ $[E]^+ = [EC]$
True

G cover F

$AB \to C$ $[AB]^+ = [ABC]$
$D \to E$ $[D]^+ = [DEC]$
× $AB \to E$ $[AB]^+ = [ABC]$
$E \to C$ $[E]^+ = [EC]$ False

$(F \neq G)$

∴ Not Minimal Cover.

Procedure to Find Minimal Cover.

(1) Split the FD such that RHS contain single attribute
$A \to BC$: $A \to B$, $A \to C$

(2) Find the redundant attribute on LHS and delete them
$AB \to C$; A is extra if $[B]^+ = [\ldots A]$
B is extra if $[A]^+ = [\ldots B \ldots]$

③ Find the redundant FD and delete them from FD set.

{A → B, B → C, A → C}

{A → B, B → C}

Ex 1. [A → C, AC → D, E → AD, E → H]

Step 1: Split the FD such that RHS contain single attributes.

A → C, AC → D, E → A, E → D, E → H.

Step 2: Find the redundant attribute on LHS and delete them from FD

AC → D [A]+ = [ACD] ✓ ∴ C is extra.

[C]+ = [C] ∴ A is not extra

{A → D}

Step 3: {A → C, A → D, E → A, E → D, E → H}

Find the R.F.D and delete it from FD set.

① A → C ② A → D ③ E → A ④ ~~E → D~~

[A]+ = [AD] [A]+ = [AC] [E]+ = [EDH] [E]+ = [EAHCD]

⑤ E → H

[E]+ = [EACD]

* Hide the RHS, take closure, if the hidden element arrives in closure, then mark as redundant FD and delete it.

* Once deleted element should not be considered in future.

Minimal Cover: $\boxed{A \to C \quad A \to D \quad E \to A \quad E \to H}$

or

$\boxed{A \to CD \quad E \to AH}$

Ex 2 [AB → CD, A → E, E → C]

Step 1: AB → C, AB → D, A → E, E → C

Step 2: AB → C [A]+ = [AEC]

[B]+ = [B] B is extra (Both)

① ~~AB → C~~ ② AB → D ③ A → E ④ E → C

[AB]+ = [ABDEC] [AB]+ = [ABEC] [A]+ = [A] [E]+ = [E]

Answer:

AB → D

A → E

E → C

Date 2/8/23

Ex: [$A \to BC$, $CD \to E$, $E \to C$, $D \to AEH$, $ABH \to BD$, $DH \to BC$]

Step 1: $A \to B$, $A \to C$, $CD \to E$, $E \to C$, $D \to A$, $D \to E$, $D \to H$, $ABH \to B$, $ABH \to D$, $DH \to B$, $DH \to C$.

Step 2: $\cancel{C}D \to E$ $[C]^+ = [C]$ $[D]^+ = [DC..]$ C is extra

$D\cancel{H} \to B$, $DH \to C$ $[H]^+ = [H]$, $[D]^+ = [DH..]$ H = extra

$A\cancel{B}H \to B$, $A\cancel{B}H \to D$ $[AH]^+ = [AHB..]$ B is extra

$AH \to D$, $AH \to B$ $[A]^+ = [ABC]$ $[H]^+ = [H]$

Step 3:

1. $A \to B$ $[A]^+ = [AC]$
2. $A \to C$ $[A]^+ = [AB]$
3. $D \to E$ $[D]^+ = [DABCH]$
4. $E \to C$ $[E]^+ = [E]$
5. $D \to A$ $[D]^+ = [DEACB]$
6. $D \to H$
7. $AH \cancel{\to} B$ $[AH]^+ = [AHDBC]$
8. $AH \to D$ $[AH]^+ = [AHBC]$
9. $D \cancel{\to} B$ $[D]^+ = [DACBHE]$
10. $D \cancel{\to} C$ $[D]^+ = [DABCE]$

Minimal Cover: {$A \to B$, $A \to C$, $D \to E$, $E \to C$, $D \to A$, $D \to H$, $AH \to D$, }

Note: Minimal cover may or maynot be unique i.e we can have more than one minimal cover.

HW

Q1. Consider the following FD set:
{$P \to QR$, $Q \to PR$, $R \to PQ$} which of the following is/are minimal cover for the above FD set?

a. $P \to Q$, $Q \to R$, $R \to P$
b. $P \to R$, $Q \to R$, $R \to PQ$.
c. $Q \to P$, $P \to R$, $R \to Q$
d. $P \to QR$, $Q \to P$, $R \to P$

Finding Number of Super Keys

Let R be the Relational Schema with n attributes $A_1, A_2 \ldots A_n$
How many super keys are there?

(I) With only Candidate key A_1?

\# Super keys: A_1
$A_1 A_2$
$A_1 A_2 A_3$
$\underbrace{A_1 A_2 \ldots A_n}_{n-1}$

$\boxed{\# \text{Super Keys} = 2^{n-1}}$

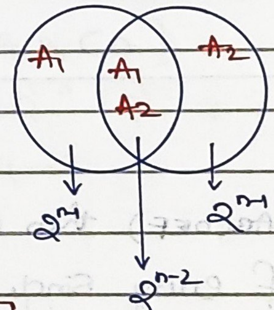
eg: R(ABCD) with CK: A
Super key $= 2^{n-1} = 2^{4-1} = \underline{8}$

A
AD
AC
⋮
ACD
ABD
⋮
ABCD

8 Super keys

(II) With only Candidate key A_1, A_2?

A_1	A_2
$A_1 A_2$	$A_2 A_1$
$A_1 A_2 A_3$	$A_2 A_1 A_3$
$\underbrace{A_1 A_2 \ldots A_n}_{(n-1)}$	$\underbrace{A_2 A_1 A_3 \ldots A_n}_{(n-1)}$

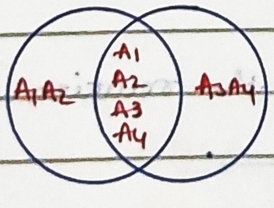


$n(A \cup B) = n(A) + n(B) - n(A \cap B)$

$\boxed{\# \text{Super Keys} = 2^{n-1} + 2^{n-1} - 2^{n-2}}$

eg: R(ABCDE) with CK: A, D
\# super key $= 2^{5-1} + 2^{5-1} - 2^{5-2} = \underline{\underline{24}}$
$(2^{n-1} + 2^{n-1} - 2^{n-2})$

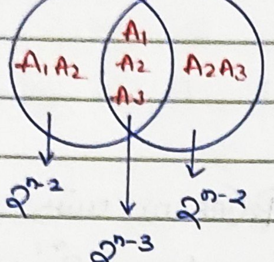
(III) With only Candidate key $A_1 A_2$, $A_3 A_4$?



$\boxed{\# \text{Super keys} = 2^{n-2} + 2^{n-2} - 2^{n-4}}$

eg: R(ABCDE) Candidate keys [AB, DE]
Super keys $= 2^{5-2} + 2^{5-2} - 2^{5-4}$
$= 2^3 + 2^3 - 2^1 = 8 + 8 - 2 = \underline{14 \text{ Super keys}}$

(iv) With only Candidate key A_1A_2, A_2A_3?



$$\boxed{\#\text{ Superkeys} = 2^{n-2} + 2^{n-2} - 2^{n-3}}$$

eg: R(ABCDE) CK {AB, BC}.

Super Keys $= 2^{n-2} + 2^{n-2} - 2^{n-3}$

$= 2^{5-2} + 2^{5-2} - 2^{5-3} = 12$ superkeys

(v) With only Candidate key A_1, A_2, A_3?

$n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(B \cap C) - n(A \cap C) + n(A \cap B \cap C)$

$$\boxed{\#\text{Super keys} = 2^{n-1} + 2^{n-1} + 2^{n-1} - 2^{n-2} - 2^{n-2} - 2^{n-2} + 2^{n-3}}$$

$$\boxed{\text{Maximum Number of Candidate keys} = {}^{n}C_{\lfloor \frac{n}{2} \rfloor}}$$

n = no. of attributes.

eg: R(ABCDEF) then what is the maximum # of candidate keys?

- If every single attribute is a Candidate key $= {}^6C_1 = 6$
- If every 3 attribute forms a CK $= {}^6C_3 = 20$ $\left\{ {}^nC_{\lfloor n/2 \rfloor} \right\}$

Max. no of C.keys.

Q1. The maximum number of superkeys for the relation schema R (EFGH) with E as the key ___8___.

Ans R(EFGH) E is the CK

$\#$ Super key $= 2^{n-1}$

$= 2^{4-1}$

$= 8$ Superkeys

Q2. Consider a Relation R(ABCDE) with following three functional dependencies: $\{AB \rightarrow C,\ BC \rightarrow D,\ C \rightarrow E\}$.
The number of Super keys in the relation R is 8

Ans Candidate key : $[AB]^+ = [ABCDE]$.
No. of Super keys = $2^{n-2} = 2^{5-2} = 8$ Superkeys.

Q3. R(ABCDE) with Candidate keys : A, BC
No of Super keys : $2^{5-1} + 2^{5-2} \cancel{*} - 2^{5-3}$
$= 16 + 8 - 4$
$= 20$ Superkeys

II method :
$\cancel{A}BCDE = 2^4 = 16$
$BC\,DE = 2^2 = 4$
20

Q4. R(ABCDE) and Candidate keys : [A, B, CD]
Super keys = $2^{5-1} + 2^{5-1} + 2^{5-2} - 2^{5-2} - 2^{5-3} - 2^{5-3} + 2^{5-4}$
$= 16 + 16 \cancel{+8} \cancel{-8} - 4 - 4 + 2$
$= 26$ Superkeys.

II method :
$\cancel{A}BCDE = 2^4 = 16$
$\cancel{B}CDE = 2^3 = 8$
$CDE = 2^1 = 2$
26

Note:
Total maximum number of Super keys = $2^n - 1$.

Under the assumption that every single attribute is Candidate key.

Q5. R(ABCD) with Candidate keys = [A, B, C, D]
$\cancel{A}BCD = 2^3 = 8$
$BCD = 2^2 = 4$
$CD = 2^1 = 2$
$D = 2^0 = 1$
15 Super keys.

Q6. R(ABCDE) with Candidate keys [AB, BC].

No. of Super keys $= 2^{5-2} + 2^{5-2} - 2^{5-3}$

$= 2^3 + 2^3 - 2^2$

$= 12$ Superkeys.

ABCDE $= 2^3 = 8$ Superkeys

BCDE $= 2^2 = 4$ Superkeys

12 Superkeys.

Q7. R(ABCDE) with C.K. [AB, CD]

No. of Superkeys

$= 2^{5-2} + 2^{5-2} - 2^{5-4}$

$= 2^3 + 2^3 - 2^1$

$=$ 14 Super keys. ✓

II method : ABCDE $= 2^3 = 8$

CDE $= 2^1 = 2$

10 Superkeys ✗

∴ This method doesn't work always.

Q8. If in a Relation R with n attribute, if every (n-1) attribute, if every (n-1) attribute is a Candidate key then number of Super keys?

Ans. (n-1) attribute is C.K.

$\boxed{{}^nC_{n-1}}$ $\boxed{{}^nC_n}$ $\Rightarrow$ n+1

eg: R(ABC) n=3

AB, BC, AC are Candidate keys.

Super keys = n+1 = 4

AB, BC, AC, ABC Super keys.

Q9. In Relation R with ~~exact~~ n attribute if every (n-2) attribute is Candidate key then number of Super keys?

Ans # Super keys $\Rightarrow {}^nC_{n-2}$ + Any Superset of ${}^nC_{n-2}$

${}^nC_{n-2}$ ${}^nC_n$ ${}^nC_{n-1}$

$\boxed{\#\text{Superkeys} = {}^nC_{n-2} + {}^nC_{n-1} + {}^nC_n}$

Q10. R(ABCDE) if every '3' attribute is Candidate keys then

(i) Find total no. of Candidate keys

(ii) Find total no of Super keys.

Ans. Total Candidate keys $= {}^nC_{n-2} = {}^5C_{5-2} \Rightarrow$ 10 Candidate keys

Total Super keys $= {}^nC_{n-2} + {}^nC_{n-1} + {}^nC_n$

$= {}^5C_3 + {}^5C_4 + {}^5C_5 \rightarrow 10+5+1$

$\Rightarrow$ 16

Q11. If every (n-3) attribute form a candidate key then no. of super key?

Ans # Super keys $= {}^nC_{n-3} + {}^nC_{n-2} + {}^nC_{n-1} + {}^nC_n$.

Q12. R(ABCDE) if every 2 attribute is C.K then total no. of candidate key and no of super keys?

Ans # Candidate keys $= {}^nC_{n-3} = {}^5C_2 = 10$ Candidate keys.

Super keys $= {}^5C_2 + {}^5C_3 + {}^5C_4 + {}^5C_5 \Rightarrow$ 26 Super keys.

Q13 If every single attribute is a candidate key then total no. of super key?

Ans Total no. of super keys $= {}^nC_1 + {}^nC_2 + {}^nC_3 \ldots\ldots {}^nC_n$

Q14. R(ABCD) if every single attribute is a candidate key then no. of super key?

Ans. # Super keys $= {}^4C_1 + {}^4C_2 + {}^4C_3 + {}^4C_4$

$= 4+6+4+1$

$= 15$

Super keys $= 2^n - 1$

$= 2^4 - 1$

$= 15$ Super keys.

Properties of Decomposition:

1. Lossless join decomposition
2. Dependency preserving decomposition.

Lossless Join Decomposition:

1. Basic Concept
2. Binary Method } Methods
3. Chase Test.

Let R be the Relational Schema with instance r, is decomposed into sub relation $R_1, R_2, R_3 \dots R_n$ with instance $r_1, r_2, r_3 \dots n$ respectively.

$$\boxed{\begin{array}{c} \text{If } R_1 \bowtie R_2 \bowtie R_3 \dots \bowtie R_n = R \\ \text{Lossless Join Decomposition.} \end{array}}$$

$$\boxed{\begin{array}{c} \text{If } R_1 \bowtie R_2 \bowtie R_3 \dots \bowtie R_n \supset R \\ \text{Lossy Join Decomposition.} \end{array}}$$

Natural Join ($\bowtie$) $R \bowtie S$

It is performed into 3 steps:

Step 1: Cross product of R and S.

R	S
n_1 Tuple	n_2 Tuple
C_1 Attribute	C_2 Attribute

$$\boxed{\begin{array}{l} R \times S = n_1 \times n_2 \text{ Tuple} \\ \quad C_1 + C_2 \text{ Attribute} \end{array}}$$

Step 2: Select the tuples which satisfy equality condition on all common attributes (From $R \times S$ (step1)).

Step 3: Projection of distinct attribute.

Date: 04/8/23

Q1. R(ABC)

A	B	C
1	5	5
2	5	8
3	8	8

Decomposed into

(1) R_1 (ABC) and R_2 (BC)

$\downarrow$ 3 tuple, 2 Attribute

$\downarrow$ 3 tuple, 2 Attribute.

A	B
1	5
2	5
3	8

B	C
5	5
5	8
8	8

$R \times S \begin{cases} 3 \times 3 = 9 \text{ Tuple} \\ 2+2 = 4 \text{ Attribute.} \end{cases}$

Step 1 $R_1 \times R_2$

	R_1A	R_1B	R_2B	R_2C
1.	1	5	5	5
2.	1	5	5	8
	1	5	8	8
3.	2	5	5	5
4.	2	5	5	8
	2	5	8	8
	3	8	5	5
	3	8	5	8
5.	3	8	8	8

$R_1.B = R_2.B$ Step 2

(1, 2, 3, 4, 5)

Step 3 : Distinct attributes of Step 2

A	B	C
1	5	5
1	5	8
2	5	5
2	5	8
3	8	8

→ Spurious (extra) tuple

We didn't get the original table

∴ Lossy Join.

Q2. R(ABC)

A	B	C
1	5	5
2	5	8
3	8	8

Decomposed into

R_1 (AB) and R_2 (AC)

A	B
1	5
2	5
3	8

A	C
1	5
2	8
3	8

$R_1 \bowtie R_2 \Rightarrow$

R1.A	R1.B	R2.A	R2.B
1	5	1	5
1	5	2	8
1	5	3	8
2	5	1	5
2	5	2	8
2	5	3	8
3	8	1	5
3	8	2	8
3	8	3	8

R1.A = R2.A

A	B	C
1	5	5
2	5	8
3	8	8

$\Rightarrow R$

$\therefore$ Lossless Join

Lossless Join Decomposition (Binary Method).

Let R be the relational schema with FD set F is decomposed into Sub Relation R_1 and R_2.

$R_1 \bowtie R_2$ is Lossless.

if ① $R_1 \cup R_2 = R$

② If common attribute of R_1 and R_2

→ either a Super key of R_1 $\qquad [R_1 \cap R_2]^+ \rightarrow R_1$

$\boxed{\text{or}}$ $\qquad$ or

→ Super key of R_2 $\qquad [R_1 \cap R_2]^+ \rightarrow R_2$

$R_1 \bowtie R_2$ is Lossy Join

① If common attribute of R_1 & R_2

→ Neither Super key of R_1

$\boxed{\text{nor}}$

→ Super key of R_2.

② R(ABCD)

R_1(AB) $\qquad$ R_2(CD) $\Big\}$ Lossy Join.

③ R(ABCDEFGH)

R_1(ABCD) $\qquad$ R_2(BCDG)

Lossy because E, H, & F are missing.

Q1. $A(ABCDEFG)$ $\{AB \to CD, D \to E, E \to FG\}$
Decomposed into $R_1(ABCD)$ and $R_2(DEFG)$.

Ans $R_1(ABCD) \cup R_2(DEFG) = R(ABCDEFG)$ ✓
$R_1(ABCD) \cap R_2(DEFG) = D$
$[D]^+ = [DEFG]$ Super key of R_2 ✓
∴ Lossless Join

Q2. R(ABC)

A	B	C
1	5	5
2	5	8
3	8	8

Decomposed into $R_1(AB)$ and $R_2(BC)$

Ans
(i) $R_1(AB) \cup R_2(BC) = R(ABC)$ ✓
(ii) $R_1(AB) \cap R_2(BC) = B$ ✗
$[B]^+ = B \to$ Not a Super key of R_1 or R_2
∴ Lossy Join

Q3. $R(ABCDEFG)$ $\{AB \to C, C \to D, D \to EFG\}$
Decomposed into $R_1(ABCE)$ and $R_2(DEFG)$

Ans $R_1(ABCE) \cup R_2(DEFG) = R(ABCDEFG$
$R_1(ABCE) \cap R_2(DEFG) = E$
$[E]^+ = [E] \to$ Not a Super key of R_1 or R_2
∴ Lossy Join

Q4. $R(ABCDEFG)$ $[AB \to C, C \to D, D \to EFG]$
$R_1(ABCDE)$ $R_2(BEFG)$

Ans $R_1 \cap R_2 = [BE]^+ = [BE]$
BE not a super key of R_1 & R_2
∴ Lossy Join

Q5. $R(ABCDEG)$ $\{AB \to C, AC \to B, AD \to E, B \to D, BC \to A, E \to G\}$
Decomposed into $R_1(ABC)$ $R_2(ACDE)$ and $R_3(ADG)$

Ans $R_1(ABC) \cap R_2(ACDE) = AC$
$[AC]^+ = [ACB\ldots] \to$ Super key of R_1
$R_{12}(ABCDE) \cap R_3(ADG) = AD$
$[AD]^+ = [ADEG] \to$ Superkey of R_3.

$R_{123}(ABCDEG)$
∴ Lossless Join

$R_1(ABC)$ $R_2(ACDE)$ $R_3(ADG)$

$R_{12}(ABCDE)$ $R_3(ADG)$

$R_{123}(ABCDEG)$ {Lossless Join}

Q6. $R(ABCDE\cancel{F}G)$ $\{AB \to C, AC \to B, AD \to E, B \to D, BC \to A, E \to CG\}$

(i) Decomposed into $R_1(AB)$ $R_2(BC)$ $R_3(ABDE)$ and $R_4(EG)$

(ii) Decomposed into $R_1(AB)$ $R_2(BC)$ $R_3(ABDE)$ and $R_4(ECG)$.

Ans.

(i) $R_1(AB) \cap R_2(BC) = [B]$

$[B]^+ = [BD] \to$ not a Super key of R_1 and not of R_2

$R_1(AB) \cap R_3(ABDE) = [AB]$

$[AB]^+ = [AB\ldots] \to$ Super key of R_1, so Join

$R_{13}(ABDE)$ $R_2(BC)$ $R_4(EG)$

$R_{13}(ABDE) \cap R_4(EG) = [E]$

$[E]^+ = [ECG] \to$ Super key of R_4

$R_{134}(ABDEG) \cap R_2(BC) = [B]$

$[B]^+ = [BD] \to$ not a super key of R_{134} and not super key of R_2

$\therefore$ Lossy Join

(ii) $R_1(AB) \cap R_2(BC) = B$

$[B]^+ = [BD] \to$ not a Superkey of R_1 nor R_2

$R_1(AB) \cap R_3(ABDE) = [AB]$

$[AB]^+ = [AB\ldots] \to$ Super key of R_1

$R_{13}(ABDE)$

$R_{13}(ABDE) \cap R_4(ECG) = E$

$[E]^+ = [ECG] \to$ Super key of R_4

$R_{134}(ABCDEG) \cap R_2(BC) = BC$

$[BC]^+ = [BC\ldots] \to$ Superkey of R_2

$R_{1234}(ABCDEG)$

$\therefore$ Lossless

Lossless Join Decomposition (Chase Test)

→ In Chase test we create a matrix in which column represents the attribute and tuple (row) represent the sub relations.

→ Fill all the cells (entries) / values with any variable (assume 'a') in corresponding attributes of respective sub relation.

→ Now fill the table entries with the help of given FD's.

$x \to y$ – if 2 'x' value (same 2 value of x) is present and one y value 'a' is present then write 'a' in another y.

$x \to y$ $\boxed{\text{If } t_1.x = t_2.y \text{ then } t_1.y = t_2.y \text{ must be same.}}$

→ If 2 same value of x is present but '0' y value is there then insert any other variable (assume 'b') instead of 'a'.

→ If no two same value of x then that FD not applied on that moment.

Note:

* If we get any one tuple with all 'a' entries then lossless join.
* CHASE TEST will stop, if either we get one tuple with all 'a' variable entries (lossless) or there is no further updation in table.

$x \to y$ $\boxed{\text{If } t_1.x = t_2.x \text{ then } t_1.y = t_2.y \text{ must be same.}}$

1.

x	y
a → a	
a → a	

$x \to y$ → If any one y value is present then insert the same y (a) value.

{If 2-same value of x}

→ If '0' y value is present then insert some other variable (assume 'b') instead of 'a'.

2.

x	y
a → b	
a → b	

Q1. R(ABCDEFG) {AB → CD, D → E, E → FG}.
Decomposed into R_1 (ABCD) and R_2 (DEFG) by Chase Test.

Ans

	A	B	C	D	E	F	G
R_1 (ABCD)	a	a	a	a	a	a	a
R_2 (DEFG)				a	a	a	a

$x \to y$
× AB → CD
D → E
✓ E → FG

Getting a tuple with all 'a' entries
So Lossless Join.

Q2. (ABCDEFG) {AB → C, C → D, D → EFG}.
Decomposed into R_1 (ABCE) and R_2 (DEFG) by Chase Test.

Ans

$x \to y$
× AB → C
× C → D
× D → EFG

	A	B	C	D	E	F	G
R_1(ABCE)	a	a	a		a		
R_2(DEFG)				a	a	a	a

Not getting a tuple with all 'a' entries
∴ Lossy Join.

H/W

Q3. R(ABCDEG) {AB → C, AC → B, AD → E, B → D, BC → A, E → CG}
Decomposed into R_1 (AB) R_2 (BC) R_3 (ABDE) and R_4 (ECG)

Ans.

	A	B	C	D	E	G
R_1(AB)	a	a	b_1	a		
R_2 (BC)		a	a	a		
R_3 (ABDE)	a	a	~~b_1~~ a	a	a	a
R_4 (ECG)			a		a	a

$x \to y$.
✓ AB → C
AC → B
AD → E
✓ B → D
BC → A
E → CG
E → C
E → G

Getting a tuple with all 'a' entries
∴ Lossless Join.

Q4. R(ABCD) {A → B, B → C, CD → A}
R_1 (AD) R_2 (AC) R_3 (BCD)

classmate

Date 07/8/2023

Page

Qs. R(ABCDEG) {AB → C, AC → B, AD → E, B → D, BC → A, E → G}
Decomposed into R1(ABC) R2(ACDE) and R3(ADG)

	A	B	C	D	E	G
R1	a	a	a	a		
R2	a	a	a	a	a	a
R3	a			a	a	a

AB → C ~~AC → B~~

~~AD → E~~ BC → A

~~B → D~~ E → G

Getting a tuple with 'a' entries ∴ Lossless

Dependency Preserving Decomposition

Let R be the Relational Schema with FD set F is decomposed into Sub relations R1, R2, R3, ... Rn with FD set F1, F2, F3 ... Fn respectively.

$$\boxed{\text{IF } F_1 \cup F_2 \cup F_3 \ldots \cup F_n = F \quad \text{Dependency Preserving decomposition}}$$

$$\boxed{\text{If } F_1 \cup F_2 \cup F_3 \ldots \cup F_n \subset F \quad \text{Dependency Not Preserved}}$$

Q1. Let R(A, B, C, D, E) be a relational Schema with the following functional dependencies:
{A → B, B → C, C → D, and D → BE.}
Decomposed into R1(AB), R2(BC), R3(CD) and R4(DE).

Ans [A]+ = [ABCDE]
[B]+ = [BCDE]
[C]+ = [CDBE]
[D]+ = [DBEC]
[E]+ = [E]

R1(AB)	R2(BC)	R3(CD)	R4(DE)
A → B	B → C	C → D	D → E
	C → B	D → C	

A → B
B → C
C → D
D → BE

A → B, B → C, C → B, C → D, D → C, D → E

A → B, B → C, C → D, D → E, D → C, C → B, D → B.

∴ Dependency Preserving

Note: In dependency preserving decomposition method, first take the closure of the attributes then write all non trivial FD in respective sub relation.
$x \cap y = \phi$ and $x \to y$ must satisfy FD definition.

Q2. R(ABCDEFG) {AB→C, AC→B, AD→E, B→D, BC→A, E→G}
decomposed into R_1(ABC) R_2(ACDE) and R_3(ADG).

Q3. Consider a schema R(A,B,C,D) and functional dependencies A→B and C→D. Then the decomposition of R into R_1(AB) and R_2(CD) is.

Ans $R_1(AB) \cap R_2(CD) = \phi$.
∴ lossy

$[A]^+ = [AB]$
$[B]^+ = [B]$
$[C]^+ = [CD]$
$[D]^+ = [D]$

R_1(AB)	R_2(CD)
A→B	C→D

(A→B) ∪ (C→D)
A→B, C→D
Dependency Preserving

∴ Dependency preserving & not lossless.

Q4. Let R(ABCD) be the relation schema with the following functional dependencies {A→B, B→C, C→D and D→B}
The decomposition of R into (A,B), (B,C), (B,D).

Ans. $R_1(AB) \cap R_2(BC) = B$
$[B]^+ = [BCD]$, Super key of R_2.
$R_{12}(ABC) \cap R_3(BD) = B$
$[B]^+ = [BDC]$, super key of R_3
∴ R_{123}(ABCD) lossless join and dependency preserving

$[A]^+ = [ABCD]$
$[B]^+ = [BCD]$
$[C]^+ = [CDB]$
$[D]^+ = [DBC]$

R_1(AB)	R_2(BC)	R_3(BD)
A→B	B→C C→B	B→D D→B

A→B, B→C
D→B, C→D

A→B, B→C, C→B, B→D, D→B

∴ D.P

Attribute Closure $[X]^+$: Set of all possible attribute which is logically determined by attribute closure of 'X'.

Closure of FD Set $[F]^+$: Set of all possible FD's which is determined by given FD set is called Closure of FD set.

Closure of FD Set:
- $\boxed{\text{Case I}}$ → When non fd set is given then find $[F]^+$.
- $\boxed{\text{Case II}}$ → When FD set is given then find $[F]^+$.

Case I: When FD set is not given.

eg: R(AB) then find $[F]^+$

ϕ	$\phi \to \phi$	$A \to \phi$	$B \to \phi$	$AB \to \phi$
A	$\phi \to A$	$A \to A$	$B \to A$	$AB \to A$
B	$\phi \to B$	$A \to B$	$B \to B$	$AB \to B$
AB	$\phi \to AB$	$A \to AB$	$B \to AB$	$AB \to AB$

Invalid

$[F]^+ = 13$

$n = 4$ → $(n-1)*n + 1$

Formula: $3 \times 4 + 1 = 13$

Case II: When FD set is given in the question then find $[F]^+$.

eg: R(AB) $[A \to B]$ then find $[F]^+$?

ϕ 0 attribute: $\phi \to \phi$ ⟶ 1

A 1 attribute: $[A]^+ = [AB] = 2^2$ ⟶ 4

B $[B]^+ = [B] = 2^1$ ⟶ 2

AB 2 attribute: $[AB]^+ = [AB] = 2^2$ ⟶ 4

11 Ans

$[F]^+ = 11$

Date: 8/8/2023

eg: R(ABC) {A→B, B→C} then find [F]$^+$ = ?

0 attribute : $\phi \rightarrow \phi$ ⟶ 1

1 attribute : $[A]^+ = [ABC] = 2^3 \rightarrow 8$

$[B]^+ = [BC] = 2^2 \rightarrow 4$

$[C]^+ = [C] = 2^1 \rightarrow 2$

2 Attribute : $[AB]^+ = [ABC] = 2^3 \rightarrow 8$

$[BC]^+ = [BC] = 2^2 \rightarrow 4$

$[AC]^+ = [ABC] = 2^3 \rightarrow 8$

3 Attribute: $[ABC]^+ = [ABC] = 2^3 \rightarrow 8$

43.

Normal Forms:

Normal form is a set of rules used to reduce or eliminate the redundancy.

There are various normal forms:

1. 1NF (first normal form)
2. 2NF (Second normal form)
3. 3NF (Third normal form)
4. BCNF (Boyce Codd normal form)

* Every higher normal form contain the lower normal form.
* If a relation R is in 2NF that means it's already in 1NF.
* If a relation R is in 3NF that means it's already in 2NF & 1NF.
* If a relation R is in BCNF that means R already in 3NF, 2NF and 1NF.

First Normal Form (1NF)

A Relation R is in 1NF if R does not contain any multi valued attribute. or

A relation R is in 1NF if all attribute of R are atomic.

Roll.	Name.	Course.
1.	Vinod	C/Java

Not in 1NF

multi valued attribute.

Roll.	Name	Course
1	Vinod	C++
1	Vinod	Java

R is in 1NF.

* 1NF is ensured by Candidate key.
* By default RDBMS is in 1NF, In 1NF the redundancy level is too high.

$$\boxed{\text{Redundancy level: } 1NF > 2NF > 3NF > BCNF}$$

Possible Non-Trivial FD which Create redundancy:

Case I :	Proper subset of Candidate key	→	Non key/non prime attribute.	Eliminated by 2NF
Case II :	Non Key attribute.	→	Non key attribute	Eliminated by 3NF
Case III :	Proper Subset of One Candidate key.	→	Proper subset of another Candidate key.	Eliminated by BCNF

* 2NF does not allow Case I but allows Case II, Case III.
* 3NF does not allow Case I, II but allows Case III.
* BCNF does not allow Case I, II and III.

Case I : eg: R(ABCDEF) {AB→C, C→DF, B→E}

Candidate key = [AB]

Non prime/non key attributes = [CDEF]

B→E

Proper Subset of CK

non key attribute.

Case II: eg: R(ABC) {A→B, B→C}

Candidate key : [A]

Non prime / non key attribute(s) = [B,C]

B → C

non key attribute.   non key attribute

Case III: eg: R(ABCD) {AB→CD, D→A}

Candidate key : [AB, DB]

Non key attribute = [C]

D → A

proper subset of Candidate key.   proper subset of another C.K.

Q. [FD-I] AB → C , [FD-II] A → C

Identify which one is partial dependency?

Ans. AB → C is partial FD

Partial Dependency: $x \to y$ is partial FD

If $A \in X$

$(X - A) \to y$

if any attribute A is removed from X and you still get Y then it is partial FD

Second Normal Form: (2NF)

It is based on the concept of full functional dependency. A functional dependency $X \to Y$ is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more, that is for any attribute $A \in X$, $(X - \{A\})$ does not functionally determine Y. A functional dependency $X \to Y$ is a partial dependency is some attribute $A \in X$ can be removed from X and the dependency still holds.

$X \to Y$ is Full FD

if $(X \to Y)$ and if $(X - A) \not\to Y$

Q2. Identify which of the following is a partial FD?
R(ABCDEF) {ABC→DE, DE→ABC, AB→D, DE→F, E→C}.

I. ABC→D – Partial FD ∵ AB→D.
II. AE→C – Partial FD ∵ E→C.
III. AF→D – full FD
IV. AB→D – full FD
V. AC→D – full FD.
VI. BC→D – full FD
VII. BE→C – partial FD ∵ E→C
VIII. AB→F. – full FD

Second Normal Form

Definition: A relation schema R is in 2NF if every non prime attribute A in R is fully functional dependent on the primary key of R.

→ A Relation R is in 2NF if:

1. R is in 1NF
2. R does not contain the below type FD-

Proper Subset of Candidate key.	–/→	Non key attribute.

Third Normal Form. (Condition Checking)

A relation schema R is in 3NF if every x→y non trivial FD must satisfy the following condition:

x→y

x : Super key (or) y : key/prime attribute.

BCNF (condition Checking in BCNF)

A relation schema R is in BCNF if x→y is non trivial FD must satisfy : x→y

x : Super key

Date 9/8/2023
Page

Q3. Let R(ABCDEPG) be a relational schema in which the following functional dependencies are known to hold,
{AB → CD, DE → P, C → E, P → C, and B → G}.

Ans Candidate key = [AB]
non key attribute = [C, D, E, P, G]
B → G
Proper subset of candidate key → non key attribute } violation of 2NF.
∴ Not in 2NF

Third Normal Form
Definition: According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no non prime attribute of R is transitively dependent on primary key.

Definition: A Relation Schema R is in third normal form if whenever a non-trivial functional dependency X → A holds in R either (a) X is super key of R, or (b) A is a prime attribute of R.

eg. R(ABC) [A → B, B → C]
Candidate key = [A]
Non key attribute = [B, C]

A → B ✓ 3NF (A is a super key)
B → C ✗ 3NF
not superkey, non prime } B is neither super key nor C is key/prime attribute.

Codd Definition:
A → B → C
* A → C is transitive FD.
* Non prime attribute C is transitively dependent on primary key
So, not in 3NF.

Boyce-Codd Normal Form

Definition: A relation schema R is in BCNF if whenever a non trivial functional dependency $X \to A$ holds in R, then X is a Super key of R.

X : Superkey

Import Points:

2NF Checking	3NF Checking	BCNF Checking
[Proper Subset OF CK.] → [non key attribute] validation of 2NF ∴ Not in 2NF.	R is in 3NF if every non trivial FD must satisfy the following: x : Super key or y : key / prime attribute	R is in BCNF if every $X \to A$ non trivial FD must satisfy the following condition. X : Super key.

eg. R(ABCD) {AB → CD, D → A}

Candidate key = [AB, DB]

3NF Checking:

AB → CD ✓ 3NF AB is Superkey

D → A ✓ 3NF D not superkey but A is prime attribute.

∴ R is in 3NF

BCNF Checking:

AB → CD ✓ BCNF (AB is Superkey)

D → A ✗ BCNF

Because D is not Superkey.

∴ R is not in BCNF.

Q1. Consider a relational table R that is in 3NF, but not in BCNF. Which one of following Statements is true?

Ans R has a non-trivial functional dependency $X \to A$, where X is not a Super key and A is a prime attribute.

Note: If a Relation R has only one Candidate key then R always in 1NF but may or may not in 2NF, 3NF & BCNF.

eg:

R(ABCDE) {AB→C, C→D, B→E}

Candidate key = [AB]

Non Key attribute = [C, DE]

B → E

proper subset of C.K → non key attribute } Not in 2NF

* If in a relation R all candidate keys are Simple (single attribute) Candidate key (no composite key) then relation R always in 2NF but may or may not be in 3NF & BCNF

eg: R(ABCDE) {A~~B~~→B, B→C, C→D, D→E, C→A}.

Candidate Key = [A, C, B]

Here all Candidate keys are Simple candidate keys then R always in 2NF

But not in 3NF : D → E

not Superkey → not prime attribute.

* If in a relation R, all attributes are key/prime attributes then R always is in 3NF but may or may not be in BCNF

eg: R(ABCD) {A→B, B→C, C→A}

Candidate keys = [AD, BD, CD]

Key/prime attributes = [A, B, C, D]

Non key/non prime attributes = [ ]

→ All keys are prime/key attributes so R is in 3NF but not in BCNF.

* If a relation R is in 3NF, and all Candidate keys are Simple Candidate key then R always in BCNF.

eg: R(ABC) {A→B, B→C, B→E}
Candidate key = [A, B, C], Prime attributes = [A, B, C]
Here all attributes are key/prime attributes so R is in 3NF
Also all Candidate Keys are simple candidate keys so R is in BCNF.

A → B
B → C } A, B, C are Superkeys.
C → C

* Binary Relation (Relation with 2 Attributes) is always in BCNF

eg: R(AB) {A→B}
Candidate key = [A]
A→B is superkey
∴ R is in BCNF

eg: R(AB) {B→A}
Candidate key = [B]
B→A, B is Superkey.
∴ R is in BCNF

eg: R(AB) {A→B, B→A}.
Candidate key = [A, B]
A→B, B→A, both S.K
∴ R is in BCNF

* A Relation R with no-non trivial FD is always is in BCNF
(Trivial FD) is in BCNF.
XYZ → XY.
Superkey
∴ BCNF

Design Goal	1NF	2NF	3NF	BCNF
0% Redundancy	×	×	×	✓ {Suffer from multi valued attributes}. (X →→ Y).
Lossless Join	✓	✓	✓	✓
Dependency Preserving.	✓	✓	✓	{May/may not}

Normal Form Decomposition

2NF Decomposition:
Q1. R(ABCDEFGH) {AB→C, C→D, B→E, E→FG, G→H}.
Candidate key = [AB]
Non key/non prime attribute : [C, D, E, F, G, H].

Date 10/8/2023

2NF Checking:

$B \rightarrow E$

Proper Subset of Candidate key → non key attribute.

∴ Not in 2NF

2NF Decomposition.

R(ABCD~~EFGH~~)

$[B]^+ = [BEFG]$

R_1: ABCD R_2: BEFGH

$R_1(ABCD) \cap R_2(BEFGH) = B$

$[B]^+ = [BEFGH]$ Super key of R_2

∴ Lossless + Dependency Preserving + 2NF.

Q2. R(ABCDE) F: $\{A \rightarrow B, B \rightarrow E, C \rightarrow D\}$. Decompose into 2NF.

Candidate Key: [AC]

Non key attribute = [B, D, E]

Check 2NF?

$A \rightarrow B$, $C \rightarrow D$: Proper subset of C.K to non key attribute.

∴ Not in 2NF.

2NF Decomposition

R(A~~B~~C~~D~~~~E~~)

$[A]^+ = [ABE]$

$[C]^+ = [CD]$

R_2: ABE (CK: A) R_1: AC (CK: AC) R_3: CD (CK: C)

$R_1(AC)$ $R_2(ABE)$ $R_3(CD)$

$R_1(AC) \cap R_2(ABE) = A$

$[A]^+ = [ABE]$ Super key of R_2

$R_{12}(ABCE) \cap R_3(CD) = C$

$[C]^+ = [CD]$ Super key of R_3.

$R_{123}(ABCDE)$ — Lossless Join

∴ Dependency Preserving + 2NF + Lossless Join.

Q3. R = (ABCDEFGHIJ) $\{AB \rightarrow C, BD \rightarrow EF, AD \rightarrow GH, A \rightarrow I, H \rightarrow J\}$

Candidate key: [ABD]

Non key attribute = [C, E, F, G, H, I, J]

Checking 2NF?

$AB \rightarrow C$, $BD \rightarrow EF$: Proper subset of C.K to non key attribute

$AD \rightarrow GH$, $A \rightarrow I$: Proper subset of CK to non key attribute

∴ Not in 2NF

2NF Decomposition: R(ABCDEFGHIJ)

$[AB]^+ = [ABCI]$
$[BD]^+ = [BDEF]$
$[AD]^+ = [ADGHIJ]$
$[A]^+ = [AI]$

R1	R2	R3	R4	R5
ABD	ABCI	BDEF	ADGHIJ	AI

$\{AB \to C,\ A \to I\}$ (Further Decomposed) $\{AD \to GH,\ H \to J,\ A \to I\}$

ABC → (R2) AI → (R5) ADGHJ → (R4) AI → (R5)

R1	R2	R3	R4	R5
ABD	ABC	BDEF	ADGHJ	AI

∴ Lossless + 2NF + Dependency Preserving

Q4. R(ABCDEF) $\{AB \to C,\ C \to D,\ B \to EF\}$

Candidate key : [AB] , Non key attribute : [C, D, E, F]

Checking 2NF?

$B \to EF$

Proper subset of CK. non key attribute.

∴ Not in 2NF.

2NF Decomposition

$[B]^+ = [BEF]$ R(ABCDEF)

R1	R2
ABCD	BEF
CK : AB	CK : B

⟹ 2NF + Dependency Preserving + Lossless.

Q5. R(ABCDEFGH) $\{AB \to C,\ C \to D,\ B \to E,\ E \to F,\ A \to GH\}$

Candidate key = [AB]

Non key attribute = [C, D, E, F, G, H]

Checking 2NF?

$B \to E$
$A \to GH$ } Proper subset of CK to nonkey. ∴ Not in 2NF.

2NF Decomposition:

$[B]^+ = [BEF]$ R(ABCD~~EFG~~H)

$[A]^+ = [AGH]$

ABCD	BEF	AGH

$R_1(ABCD)$ $R_2(BEF)$ $R_3(AGH)$

$R_1(ABCD) \cap (R_2(BEF) = [B]^+ = [BEF]$ – Superkey of R_2.

$R_{12}(ABCDEF) \cap R_3(AGH) = [A]^+ = [AGH]$ – Superkey of R_3

$R_{123}(ABCDEFGH)$ – Lossless Join

∴ 2NF + Lossless + Dependency Preserving.

3NF Decomposition:

Q1. $R(ABC)$ $\{A \to B, B \to C\}$.

Candidate key : [A]

Non key attribute [B, C]

Checking in 2NF?

→ R is in 2NF.

Check 3NF?

$B \to C$ B is not S.K

nor C is prime attribute.

∴ Not in 3NF.

3NF Decomposition

R_1: AB

R_2: BC

$R_1(AB) \cap R_2(BC) = [B]^+ = [BC]$

Superkey of R2.

∴ 3NF + Lossless + Dependency preserved.

$R(A, B, C)$ $\{A \to B, B \to C\}$.

A	B	C
1	b_1	x
2	b_1	x
3	b_1	x
4	b_1	x
5	b_1	x
6	c_1	y
7	c_1	y
8	c_1	y
9	c_2	y
10	c_1	y
11	c_1	y.

AB

BC

A	B
1	b_1
2	b_1
3	b_1
4	b_1
5	b_1
6	c_1
7	c_1
8	c_1
9	c_2
10	c_1
11	c_1

B	C
b_1	x
c_1	y

∴ 3NF + Lossless Join + Dependency preserved.