RANK Improvement
WORKBOOK

GATE 2025

Computer Science & Information Technology

Databases

MADE EASY PUBLICATIONS

www.madeeasypublications.org

01 Normalization

1. (b)

$$^{n}C_{n-2} \Rightarrow \frac{n(n-1)(n-2)!}{(n-2)!\times 2} = \frac{n(n-1)}{2}$$

Candidate keys.
If every $(n-2)$ attributes are Candidate keys then every $(n-1)$ and n attributes are also Superkeys

$$\begin{aligned}\text{Total Super keys} &= {}^{n}C_{n-2} + {}^{n}C_{n-1} + {}^{n}C_{n}\\ &= {}^{n}C_{n-2} + n + 1\\ &= \frac{n(n-1)}{2} + n + 1\end{aligned}$$

2. (b)

As R is referring to R_2 and S is primary key of R_2, $\pi_R(r_1) - \pi_S(r_2)$ will give empty relation or empty table as number of values in R column of table r_1 will always refer to of respective values in S column of r_2.

3. (d)

# of super keys of $R \equiv$ (# of super keys over Prime Attributes) $\times 2^{\#\text{of non prime attribute}} \equiv (2^n - 1) \times 2^{n-m}$

4. (26)

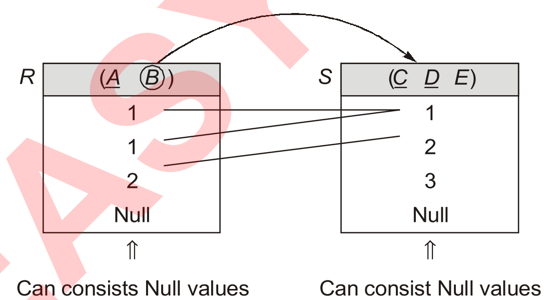
$$^{5}C_2 + {}^{5}C_3 + {}^{5}C_4 + {}^{5}C_5 = 26$$

5. (21)

Candidate keys of R: AD, AE, BD, BE, CD and CE.
Every super-key must consist of at least one of {A, B, C} and {D, E} (since every key has at least one element of each). The number of super-keys is the number of ways to choose 1, 2, or 3 attributes from {A, B ,C} times the number of ways to choose 1 or 2 attributes from {D, E}.

$$\left(\binom{3}{1}+\binom{3}{2}+\binom{3}{3}\right)\cdot\left(\binom{2}{1}\binom{2}{2}\right) = (3+3+1)\cdot(2+1)$$
$$= 21$$

6. (b)



Null value of B column record not references to any record of S. Remaining records of R references to atmost one record of S.

7. (d)

- Referential actions are generally implemented as implied triggers (i.e., triggers with system-generated names, often hidden) As such, they are subject to the same limitations as user defined triggers, and their order of execution relative to other triggers may need to be considered.
- It is true that referential integrity constraints ensure that a value that appears in one relations for a given set of attributes also appears for a certain set of attributes in another relation.

8. (c)

On inserting the tuple (15, 14%, 6) in relation B there will be any violation of integrity.
Constraints as value of attribute $f = 6$ is not present in relation A therefore, f cannot refer to any value which is not present in the referred relation A. Thus, correct option is (c).

9. (b)

I. Since the purpose of the foreign key is to identify a particular row of referenced table, it is generally required that the foreign key is equal to the candidate key in some row of the primary table, or else have no value the NULL value. This rule is called a referential integrity constraints. Thus, foreign keys are used to ensure referential integrity constraint and not NULL or domain integrity.

II. Since multiple rows in the referencing table may refer to the same row in the referenced table, in this case, the relationship between the two tables is called one to many relationship between the referenced table and the referencing table.

10. (2)

$$D^+ = ABCDEH$$

D candidate key

$$(ABH)^+ = ABCDEH$$
$$(AH)^+ = ABCDEH$$

So, there are only 2 candidate keys i.e., D, AH.

11. (3)

Find all keys of relation $R : A_1A_3A_4,\ A_3A_4A_5,\ A_2A_3A_4$.

A_3 is a prime attribute therefore it will be included in every candidate key.

12. (7)

$$(AB)^+ = ABCDEF$$
$$\vdots$$
$$(BC)^+ = ABCDEF$$
$$\vdots$$
$$(BE)^+ = ABCDEF$$
$$\vdots$$
$$(BD)^+ = ABCDEF$$
$$\vdots$$
$$(CF)^+ = ABCDEF$$
$$\vdots$$
$$(CE)^+ = ABCDEF$$
$$\vdots$$
$$(CD)^+ = ABCDEF$$

Candidate keys: $\{\underline{AB}, \underline{BC}, \underline{BE}, \underline{BD}, \underline{CF}, \underline{CE}, \underline{CD}\}$

(7 candidate keys)

13. (b)

The correct matching is

Pseudotransitive rule : If $\{X \to Y,\ YZ \to W\}$ then $XZ \to W$

Augmentation rule : If $X \to Y$ then $XZ \to YZ$ for any Z

Reflexive rule : If $X \supseteq Y$, then $X \to Y$

Union rule : If $\{X \to Y$ and $X \to Z\}$ then $X \to YZ$

14. (b)

Check G covers H:

1. $A \to B$ cannot be derived from G which is present in H, So G not covers H.

Check H covers G:

1. $CA \to B$ can be derived from H i.e. $A \to B$ then $CA \to CB$.
2. $BA \to D$ can be derived from H i.e. $A \to B$, $B \to AC$, $AC \to D$ then $B \to D$ and $AB \to D$.
3. $B \to D$ can be derived from H i.e. $A \to B$, $B \to AC$, $AC \to D$ then $B \to D$.
4. $DB \to C$ can be derived from H i.e. $B \to C$ then $DB \to DC$.

Hence H covers G.

15. (d)

AB uniquery determine C, similarly BC and AC can uniquery determine A and B respectively. Therefore, all the three non-trivial dependencies $AB \to C$, $AC \to B$ and $BC \to A$ hold for the given instance of relation.

16. (a)

Trivial: If a functional dependency $X \to Y$ holds, where Y is a subset of X, then it is caused a trivial FD.

Non-trivial: If a functional dependency $X \to Y$ holds, where Y is not a subset of X, then it is called a non-trivial FD.

Semi non-trivial: Combination of both trivial and non trivial is said to be semi-non trivial FD.

17. (b)

I. No, it is not always true that if $A \rightarrow B$ and $B \rightarrow A$. For eg:

A	*B*
0	2
1	2

Here, *A* determines *B* but *B* cannot determine *A*. Therefore, this statement is not true.

II. We must compute the closure of $A_1, A_2, \ldots\ldots A_n C$. Since $A_1, A_2, A_3 \ldots\ldots A_n \rightarrow B$ is a dependency, surely *B* is in this set, proving $A_1, A_2, \ldots\ldots A_n, C \rightarrow B$ (Augmenting left sides). Therefore, this statement is true.

18. (11)

For the single attributes we have

$A^+ = \{A\}$ $C^+ = \{ACD\}$

$B^+ = \{B\}$ $D^+ = \{AD\}$

Thus, the only new dependency we get with a single attribute on the left is $C \rightarrow A$.

Now, consider pairs of attributes:

$AB^+ = ABCD$, so we get $AB \rightarrow D$.

$AC^+ = ACD$, and $AC \rightarrow D$ is non trivial.

$BC^+ = ABCD$, we get $BC \rightarrow A$ and $BC \rightarrow D$.

$CD^+ = ACD$, $CD \rightarrow A$

For the triple of attributes,

$ABC \rightarrow D$, $ABD \rightarrow C$ and $BCD \rightarrow A$

Thus, we get 11 new dependencies.

19. (b)

$$F = AB \rightarrow C$$
$$A \rightarrow B$$
$$B \rightarrow A$$

have 2 canonical cover

$$F_{min1} = A \rightarrow C$$
$$A \rightarrow B$$
$$B \rightarrow A$$

$$F_{min2} = B \rightarrow C$$
$$A \rightarrow B$$
$$B \rightarrow A$$

20. (d)

Checking $QS \rightarrow P$, $Q^+ = Q$, $S^+ = S$, Hence $QS \rightarrow P$ is essential.

Checking $WS \rightarrow R$, $WS \rightarrow T$

$W^+ \rightarrow WVSRT$, Hence it can be decomposed to $W \rightarrow R$, $W \rightarrow T$

So, the dependencies remained are

$W \rightarrow V$, $W \rightarrow S$, $T \rightarrow S$, $W \rightarrow R$, $W \rightarrow T$, $QS \rightarrow P$

Now, $\{W \rightarrow T, T \rightarrow S\}$ by transitive rule $W \rightarrow S$ can be obtained.

Hence minimal cover is : $W \rightarrow V$, $T \rightarrow S$, $W \rightarrow R$, $W \rightarrow T$, $QS \rightarrow P$

21. (b)

$[AB \rightarrow C$		$[AB \rightarrow C$		$[AB \rightarrow C$
$C \rightarrow A$		$C \rightarrow A$		$C \rightarrow A$
$BC \rightarrow D$		$BC \rightarrow D$		$BC \rightarrow D$
$ACD \rightarrow B$	After $\Longrightarrow$	$CD \rightarrow B$	After $\Longrightarrow$	$BE \rightarrow C$
$BE \rightarrow C$	removal of	$BE \rightarrow C$	removal of	$EC \rightarrow F$
$EC \rightarrow F$	extraneous	$EC \rightarrow F$	redundant	$CF \rightarrow B$
$EC \rightarrow A$	attributes	$EC \rightarrow A$	FD's	$D \rightarrow E]$
$CF \rightarrow B$		$CF \rightarrow B$		Minimal cover
$CF \rightarrow D$		$CF \rightarrow D$		
$D \rightarrow E]$		$D \rightarrow E]$		

22. (4)

Given relation: R(A, B, C, D, E, F, G, H)

$AC \rightarrow G$, $D \rightarrow EG$, $BC \rightarrow D$, $CG \rightarrow BD$, $ACD \rightarrow B$, $CE \rightarrow AG$

Since, $(AC)^+ = ABCD$

So, $ACD \rightarrow B$, here D is extraneous.

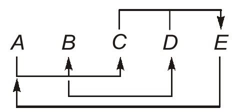
Minimal cover:

1. $\{AC \rightarrow G, D \rightarrow EG, BC \rightarrow D, CG \rightarrow B, CE \rightarrow AF\}$
2. $\{AC \rightarrow G, D \rightarrow EG, BC \rightarrow D, CG \rightarrow D, CE \rightarrow AF\}$
3. $\{AC \rightarrow B, D \rightarrow EG, BC \rightarrow D, CG \rightarrow D, CE \rightarrow AF\}$
4. $\{AC \rightarrow B, D \rightarrow EG, BC \rightarrow D, CG \rightarrow B, CE \rightarrow AF\}$

Total 4 minimal cover.

23. (b)

Calculating the key for R(A, B, C, D, E)



$A^+ \rightarrow$ ABCDE {key}
$B^+ \rightarrow$ BD
$C^+ \rightarrow$ C
$D^+ \rightarrow$ D
$E^+ \rightarrow$ EABCD {key}

Taking 2-attributes:

$(BC)^+ \rightarrow$ BCDEA {key}
$(BD)^+ \rightarrow$ BD
$(CD)^+ \rightarrow$ CDEAB {key}

Key of the relation R = {A, E, BC, CD}.
R is decomposed as R_1(ABC) and R_2(ADE)
$\{R_1 \cap R_2\} \rightarrow \{A\}$
Since A is the candidate key for R_1. Hence the decomposition is a lossless decomposition.
But, this relation is not dependency preserving since the dependency CD $\rightarrow$ E and B $\rightarrow$ D are not preserved.

24. (c)

Check for Lossless join:

(X,Y) and (Y,Z) have common attribute as Y. W to Y $\rightarrow$ Z, Y is a key for (Y,Z).

Hence XYZ can be losslessly decomposed into (X,Y) and (Y,Z).

(X, Y, Z) (Y, W), common attribute is Y.

Y $\rightarrow$ W is a functional dependencies (via Y $\rightarrow$ Z, Z $\rightarrow$ W), and hence Y is a key for (Y, W).

So, decomposition of (X, Y, Z, W) into (X, Y, Z) (Y, W) is lossless.

Thus the given decomposition is lossless.

Check for Dependency preserving:

X $\rightarrow$ Y is present in (X, Y), Y $\rightarrow$ Z is present in (Y, Z), W $\rightarrow$ Y is present in (Y, W), and Z $\rightarrow$ W is indirectly present via Z $\rightarrow$ Y in (Y, Z) and Y $\rightarrow$ W in (Y, W).

The given decomposition is also dependency preserving:

25. (c)

Here is counter example:

R

X	Y	Z	W
1	1	1	1
2	2	1	1

R_1

X	W
1	1
2	1

R_2

Y	W
1	1
2	1

R_3

Z	W
1	1

R_1 Join R_2 Join R_3

X	Y	Z	W
1	1	1	1
2	1	1	1
1	2	1	1
2	2	1	1

26. (b)

$R(PQRST)$ $\{P \rightarrow QR,\ RS \rightarrow T,\ Q \rightarrow S,\ T \rightarrow P\}$

LLJ test:

$$R_1 \cap R_2 = P^+ = PQRST$$

$R_1 \cap R_2$ key for both relations. So, given decomposition lossless join.

Dependency Preserve Test:

$$\begin{array}{ll} R_1(PQR) & R_2(PST) \\ \left\{ P \rightarrow QR \right. & \left. \begin{array}{l} P \rightarrow ST \\ T \rightarrow PS \end{array} \right\} \end{array}$$

Because of decomposition $RS \rightarrow T$, $Q \rightarrow S$ lost. So, not dependency preserving decomposition.

27. (a)

$r_i = \Pi r_i(U)$ implies that $t[R_i] \in r_i$, $1 \le i \le n$.

Thus,

$t[R_1] \bowtie t[R_2] \bowtie t[R_3]\ \ldots \bowtie t[R] \in r_1 \bowtie r_2 \ldots \bowtie r_n$

By the definition of natural Join,

$t[R_1] \bowtie t[R_2] \bowtie \ldots \bowtie t[R_n] = \pi_a[\sigma_\beta(t[R_1] \times t[R_2] \times \ldots \times t[R_n]])$

where condition β is satisfied if values of attributes one tuple. The Cartesian product of single tuples generates one tuple. The selection process is satisfied because all attributes with the same name must have the same value since they are projection from the same tuple. Finally, the projection clause removes duplicate attribute name.

By the definition of decomposition, $U = R_1 \cup R_2 \cup \ldots \cup R_n$, what mean that all attributes of t are in $t[R_1] \bowtie t[R_2] \bowtie \ldots \bowtie t[R_n]$.

That is, t is equal to result of this join. Since t is any arbitrary tuple in u.

$u \subseteq r_1 \bowtie r_2 \bowtie \ldots \bowtie r_n$

28. (a)

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

AB → C, AC → B, AD → E, B → D, BC → A, E → F

R_1(ABC)	R_2(ABDE)	R_3(EF)
AB → C	AD → E	E → F
BC → A	B → D	
AC → B		

Relation is dependency preserving since no dependencies is lost, also lossless decomposition because $R_1 \cap R_2$ = AB is candidate key of R_1, then $R_1 R_2 \cap R_3$ = E is candidate key of R_3.

29. (c)

The primary keys of the relation R are {A, B, E} and {A, B, G}. Therefore, neither (i) nor (iv) are true.

The decomposition {ABCD, AEG} doesn't have and primary key in common, therefore, this decomposition is not lossless but it is dependency preserving. Because the relation R_1(ABCD) will cover FD{AB → C and B → D} while relation R_2(AEG) will cover FD{AG → E and E → G}.

Thus correct option is (c).

30. (d)

Compute closure of all subsets of {ABC}.

A^+ = A AC^+ = ACE

B^+ = B BC^+ = ABCDE

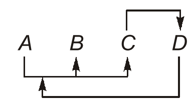
C^+ = ACE

AB^+ = ABCDE

So, the basis for the resulting FD for ABC are: C → A and AB → C. Note the BC → A is true, but follows logically from C → A. Therefore, both (A) and (C) are correct set of FD for S.

31. (d)

Finding the key for R,



Since, B is an essential attribute hence,

$(AB)^+$ → ABCD

$(BC)^+$ → BCDA

$(BD)^+$ → BDAC

Hence, the set of candidate key is {AB, BC, BD}, R is decomposed as,

R_1(CD) ⇒ C → D

R_2(AC) ⇒ C → A

R_3(BC) ⇒ No non-trivial FD's

Since in all the three functional dependencies, the LHS is a key.

Hence, the decomposition is in BCNF.

32. (d)

(1) Consider relation AB → C, B → D, R(A, B, C, D)

Key : AB

R is in 3NF, but not in BCNF.

(2) Consider another relation A → BC, B → E (A, B, C, E)

Key : A,

R is in 2NF but not in 3NF.

(3) Consider another relation A → C, B → D, R(A, B, C, D)

Key : AB:

R is not in 2NF.

33. (d)

- If every attribute is prime, then there will be no transitive dependency, hence the relation will always be in 3NF.
- If every attribute is prime, that doesn't mean the L.H.S. of the functional dependency is a key. Hence BCNF is not necessary.

- If every candidate key will be simple, then there will be no partial dependency, hence relation will be in 2 NF.

34. (1)

1. The redundancy in the set of relation that have been arised after decomposing a relation R into BCNF is more than zero but less than 3NF decomposition if multivalued dependencies are considered.
2. BCNF decompositions are lossless but always need not to be dependency preserving.
3. A prime attribute of a relation schema R is an attribute that appears in some candidate key of R.
4. A relation R is in 3NF, if every non-prime attribute of R, is fully functionally dependent on some key, because then there will be no transitive dependency.

35. (b)

From the question it can be concluded that:
{Doctor #, Patient #, Date} $\rightarrow$ {Diagnosis, Treat_code, charge}
Treat_code $\rightarrow$ charge
Since there is no partial dependency, hence the relation is in 2NF, but since, 'Treat_code' a non-prime attribute. Hence the relation is not in 3NF.

36. (c)

The given relation with functional dependencies is in 3 NF i.e., no transitive and partial function dependency exist but $C \rightarrow A$, violets BCNF i.e., super key $\rightarrow$ any attributes. So, relation R in 3NF but not BCNF.

37. (d)

Functional dependencies of re-designed schemas:

R_1 (ShipName, ShipType) {ShipName $\rightarrow$ ShipType}

R_2 (TripId, ShipName, Cargo) {TripId $\rightarrow$ ShipName, Cargo}

R_3 (ShipName, Date, TripId, Port)

{ShipName, Date $\rightarrow$ TripId, Port}
All R_1, R_2 and R_3 are in BCNF.

38. (c)

Candidate keys for the relation are: PQ, QS and QR
$S \rightarrow P$, prime attribute $\rightarrow$ prime attribute (not allowed in BCNF but allowed in 3NF).
$\Rightarrow$ Relation R is in 3NF but not in BCNF since $S \rightarrow P$ does have a superkey on the left hand side.

39. (4)

Given relation in 1NF

for 2NF	$R_1(ABCDFG)$	$R_2(BDE)$	
	$AB \rightarrow C$	$B \rightarrow D$	
	$BC \rightarrow A$	$D \rightarrow E$	
	$AC \rightarrow B$		
for 3NF	$R_1(ABCFG)$	$R_2(BD)$	$R_3(DE)$
	$AB \rightarrow C$	$B \rightarrow D$	$D \rightarrow E$
	$BC \rightarrow A$		
	$AC \rightarrow B$		
FOR BCNF	$R_1(ABC)$ $R_2(ABFG)$	$R_2(BD)$	$R_3(DE)$
(Lossless join)	$AB \rightarrow C$	$B \rightarrow D$	$D \rightarrow E$
(Dependency	$BC \rightarrow A$		
Preserving)	$AC \rightarrow B$		

40. (a)

(a) The LHS of an FD's are super key, therefore, it is in BCNF.
(b) Here AB is the candidate key, therefore, remaining FD's violates the BCNF rule. Thus, it is not in BCNF.
(c) Here ABE and ADE are super key, since none of the FD satisfying the BCFN rule, therefore it is not in BCNF.

41. (d)

- For given FD, candidate keys are AC, BC and CD.

- For FD $A \rightarrow B$, both A and B are prime attributes, therefore, it is in 3NF but not in BCNF.
- For FD, $D \rightarrow A$, it is in 3NF but not in BCNF
- For FD $BC \rightarrow D$, here the LHS of the FD is the candidate key/super key. Therefore, this FE is in BCNF. It doesn't violet any normal form.
- For FD, $C \rightarrow E$, here C is a prime attribute but E is a non-prime attribute, therefore, it violates 2NF.

42. (d)

(d) the candidate key is C, therefore FD $AC \rightarrow B$, $A \rightarrow D$, $D \rightarrow B$ neither LHS is super key nor RHS is having the prime attribute, therefore it is not in 3NF.

43. (a)

The relation R consists of FD's $B \rightarrow C$ and $CD \rightarrow E$. These all are transitive dependencies therefore it is not in 3 NF. But there is not partial dependency, therefore, it is in 2NF.

44. (d)

The functional dependency "Dept_id, Join_date → Cabin_no." Violates BCNF rule as LHS is not a super key. To remove this dependency, decompose the relation and set up a relation for the non key determinant with attributes functionally dependent on it. Hence, one of the relations should be (Dept_id, Join_date, Cabin-no.).

Hence we can eliminate option (a) and (b). Therefore option (d) is the answer.

45. (d)

The relation R consists FD's : $A \rightarrow B$, $D \rightarrow E$, $E \rightarrow D$. These all are partial dependencies therefore, it is not even in 2NF.
Hence the correct option is (d).

46. (c)

I. Whatever combination of decomposition R_1 and R_2 we take. It will always follow the 3NF conditions i.e. either LHS should be a superkey or RHS should be a prime attribute for the given FD's candidate keys are = {A, B, C}. Therefore, it is in 3NF as well as in BCNF. Therefore, this statement is false.

II. It is false because decomposition in BCNF is always lossless but we cannot say anything about dependency preserving. The FD's may or may not get preserved.

47. (d)

R(ABCD)
FDs:
$AB \rightarrow C$
$BC \rightarrow A$
$(BD)^+ = BD$
$(ABD)^+ = ABDC$
$(CBD)^+ = CBDA$
Candidate keys = {ABD, CBD}
In R, $BC \rightarrow A$ is a non trivial FD and in which BC is not a super key and A is a prime attributes.

■■■■