Decomposition

2 NF DECOMPOSTION

Q R(ABCD)
AB>D, B>C

Q R(ABCDE)
A>B, B>E, C>D

Q R(ABCDEFGHIJ)(ABD) (1NF)
AB>C, AD>GH, BD>EF, A>I, H>J

3NF DECOMPOSITON

Q R(ABCDE)

A>B, B>E, C>D

Q R(ABCDEFGHIJ)

AB>C, A>DE, B>F, F>GH, D>IJ

Q R(ABCDE)

AB>C, B>D, D>E

BCNF DECOMPOSITION

Q R(ABCD) AB (2 NF) ABC BCD
AB>C, BC>D

Q R(ABCD) AB, BD (3 NF) AFTER DECOMPOSITION AD, BCD
AB>CD, D>A

Q R(ABCD) AB, BD (3 NF) AFTER DECOMPOSITION AD, BCD
AB>CD, D>A

Q R(ABCDE) (A, E, BC, CD) (3 NF) AFTER DECOMPOSITION BD, ABC,
AEA>BC, CD>E, B>D, E>A

Q R(ABCD) (A, B) (2NF) AFTER DECOMPOSITION ABC, CD
A>B, B>A, B>C, C>D

Q R(ABDLPT) (AB) (1 NF) AFTER DECOMPOSITION TL, BPT, AB, AD
B>PT, T>L, A>D

Q R(ABCDE) (AB, BC, DB, EB) (3NF) AFTER BCNF DECOMPOSITION BC, EA, DE, CD
AB>C, C>D, D>E, E>A

Q Consider the Relation R (A, B, C, D) with the following functional dependencies A $\rightarrow$ B, C $\rightarrow$ D, B $\rightarrow$ C. The BCNF decomposition of R is

a) {(A, B), (C, D), (B, C)}
b) {(A, B), (C, D), (A, C)}
c) {(B, C), (A, D), (A, B)}
d) all of the above

Q If the relation R (ABCDE) with FD set {AB - > CDE, A -> C, C->D} is converted into BCNF then the number of foreign keys exist in resulting relations is ____________

Q Consider the schema
R = {S, T, U, V} and the dependencies
$S \rightarrow T$, $T \rightarrow U$, $U \rightarrow V$ and $V \rightarrow S$
If R = (R1 and R2) be a decomposition such that $R1 \cap R2= \phi$, then the decomposition is
(NET-JUNE-2014)

(A) not in 2NF
(B) in 2NF but not in 3NF
(C) in 3NF but not in 2NF
(D) in both 2NF and 3NF

Ans: d

FOURTH NORMAL FORM (4NF)

Multivalued Dependency-

- Multivalued dependencies are a *consequence of first normal form (1NF*) which disallows an attribute in a tuple to have a *set of values (Multiple values)*.
- Denoted by, A ->-> B, Means, for every value of A, there may exist more than one value of B.
- If there is functional dependency from A-> B, then there will also a multivalued functional dependency from A->->B.
- A trivial multivalued dependency X→→Y is one where either *Y* is a subset of *X*, or *X* and *Y* together form the whole set of attributes of the relation.
- *E.g.* let the constraint specified by MVD in relation EMP as
  - *Ename ->-> Pname*
    Ename ->-> Dname

- ***EMP***

Ename	***Pname***	***Dname***
Smith	*X*	*John*
Smith	*Y*	*Anna*
Smith	*X*	*Anna*
Smith	*Y*	*John*

Redundancy due to two independent multivalued dependencies in same Relation.

EMP_PROJECTS

Ename	***Pname***
Smith	**X**
Smith	**Y**

EMP_*DEPENDENT*

Ename	***Dname***
Smith	**john**
Smith	**anna**

NOTE: **The above EMP schema is in BCNF as no functional dependency holds on EMP, but still redundancy due to MVD.**

Hence 4NF is stricter than BCNF.

Consider the following example:

Pizza Delivery Permutations		
Restaurant	**Pizza Variety**	**Delivery Area**
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City
A1 Pizza	Stuffed Crust	Springfield
A1 Pizza	Stuffed Crust	Shelbyville
A1 Pizza	Stuffed Crust	Capital City
Elite Pizza	Thin Crust	Capital City
Elite Pizza	Stuffed Crust	Capital City
Vincenzo's Pizza	Thick Crust	Springfield
Vincenzo's Pizza	Thick Crust	Shelbyville
Vincenzo's Pizza	Thin Crust	Springfield
Vincenzo's Pizza	Thin Crust	Shelbyville

- Each row indicates that a given restaurant can deliver a given variety. The table has no non-key attributes because its only key is {Restaurant, Pizza Variety, Delivery Area}. Therefore, it meets all normal forms up to BCNF.
- If we assume, however, that pizza varieties offered by a restaurant are not affected by delivery area (i.e. a restaurant offers all pizza varieties it makes to all areas it supplies), then it does not meet 4NF. The problem is that the table features two non-trivial multivalued dependencies on the {Restaurant} attribute (which is not a super key). The dependencies are:
  - {Restaurant} →→ {Pizza Variety}
  - {Restaurant} →→ {Delivery Area}
- These non-trivial multivalued dependencies on a non-superkey reflect the fact that the varieties of pizza a restaurant offers are independent from the areas to which the restaurant delivers. This state of affairs leads to redundancy in the table:
- for example, we are told three times that A1 Pizza offers Stuffed Crust, and if A1 Pizza starts producing Cheese Crust pizzas then we will need to add multiple rows, one for each of A1 Pizza's delivery areas.

Varieties By Restaurant	
Restaurant	Pizza Variety
A1 Pizza	Thick Crust
A1 Pizza	Stuffed Crust
Elite Pizza	Thin Crust
Elite Pizza	Stuffed Crust
Vincenzo's Pizza	Thick Crust
Vincenzo's Pizza	Thin Crust

Delivery Areas By Restaurant	
Restaurant	Delivery Area
A1 Pizza	Springfield
A1 Pizza	Shelbyville
A1 Pizza	Capital City
Elite Pizza	Capital City
Vincenzo's Pizza	Springfield
Vincenzo's Pizza	Shelbyville

- If we have two or more multivalued *independent* attributes in the same relation schema, we get into a problem of having to repeat every value of one of the attributes with every value of the other attribute to keep the relation state consistent and to maintain the independence among the attributes involved. This constraint is specified by a multivalued dependency.

4NF-_A relation is in 4NF iff,

- It is in BCNF
- There must not exist any non-trivial multivalued dependency.
- Each MVD is decomposed in separate table, where it becomes trivial MVD.

- A 1992 paper by Margaret S. Wu notes that the teaching of database normalization typically stops short of 4NF, perhaps because of a belief that tables violating 4NF (but meeting all lower normal forms) are rarely encountered in business applications. This belief may not be accurate, however. Wu reports that in a study of forty organizational databases, over 20% contained one or more tables that violated 4NF while meeting all lower normal forms.

Q Multi-valued dependency among attribute is checked at which level? **(NET-JUNE-2005)**
(A) 2 NF **(B)** 3 NF **(C)** 4 NF **(D)** 5 NF

Q Which of the following is false? **(NET-DEC-2014)**
(A) Every binary relation is never be in BCNF.
(B) Every BCNF relation is in 3NF.
(C) 1 NF, 2 NF, 3 NF and BCNF are based on functional dependencies.
(D) Multivalued Dependency (MVD) is a special case of Join Dependency (JD).
Ans: a

Q Match the following: **(NET-DEC-2005)**

(i) 5 NF	**(a)** Transitive dependencies eliminated
(ii) 2 NF	**(b)** Multivalued attribute removed
(iii) 3 NF	**(c)** Contains no partial functional dependencies
(iv) 4 NF	**(d)** Contains no join dependency

(A) i-a, ii-c, iii-b, iv-d
(B) i-d, ii-c, iii-a, iv-b
(C) i-d, ii-c, iii-b, iv-a
(D) i-a, ii-b, iii-c, iv-d

Ans: b

Lossy/Lossless-Dependency Preserving Decomposition

- Because of a normalization a table is Decomposed into two or more tables, but during this decomposition we must ensure satisfaction of some properties out of which the most important is lossless join property/decomposition.
- if we decompose a table r into two tables r_1 and r_2 because of normalization then at some later stage if we want to join(combine) (natural join) these tables r_1 and r_2, then we must get back the original table r, without any extra or less tuple. But some information may be lost during retrieval of original relation or table. For e.g.

R (A, B, C)

A	B	C
1	a	p
2	b	q
3	a	r

R_1 (A, B)

A	B
1	a
2	b
3	a

R_2 (B, C)

B	C
a	p
b	q
a	r

R (A, B, C)

A	B	C
1	a	p
1	a	r
2	b	q
3	a	p
3	a	r

- Decomposition is lossy if $R1 \bowtie R2 \supset R$
- Decomposition is lossy if $R \supset R1 \bowtie R2$

- Decomposition is lossless if $R_1 \bowtie R_2 = R$ "The decomposition of relation R into R1 and R2 is **lossless** when the join of R1 and R2 yield the same relation as in R." which guarantees that the spurious (extra or less) tuple generation problem does not occur with respect to the relation schemas created after decomposition.
- ***This property is extremely critical and must be achieved at any cost.***

A	B	C	D	E
A	122	1	W	A
E	236	4	X	B
A	199	1	Y	C
B	213	2	Z	D

How to check for lossless join decomposition using FD set, following conditions must hold:

- Union of Attributes of R_1 and R_2 must be equal to attribute of R. Each attribute of R must be either in R_1 or in R_2. Att(R1) U Att(R2) = Att(R)
- Intersection of Attributes of R1 and R2 must not be NULL. Att(R1) ∩ Att(R2) ≠ Φ
- Common attribute must be a key for at least one relation (R1 or R2)
- Att(R1) ∩ Att(R2) -> Att(R1) or Att(R1) ∩ Att(R2) -> Att(R2)
- If X ∩ Y forms a superkey of either X or Y, the decomposition of R is a lossless decomposition.

E.g. Detailed example Explaination:-

Consider the following relationship : **R (A,B,C,D)**

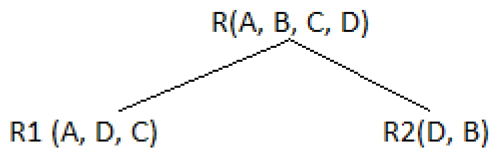
and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2**.



Breaking, table into two tables, one with A, D and C while the other with D and B.

5 NF

A Relational table R is said to be in 5^{th} normal form if
a) it is in 4 NF
B) it cannot be further non-loss decomposed

Dependency Preserving Decomposition

Let relation R be decomposed into Relations R_1, R_2, R_3………… R_N with their respective functional Dependencies set as F_1, F_2, F_3………… F_N, then the Decomposition is Dependency Preserving iff-

$$\{F_1 \cup F_2 \cup F_3 \cup F_4 \ldots\ldots \cup F_N\}^+ = F^+$$

Dependency preservation property, although desirable, is sometimes sacrificed.

Q R (A, B, C)

A--> B, B→C, C-→A

R_1(A, B) AND R_2(B, C)

LOSSLESS AND FD PREVERSING

Q R (A, B, C, D)

AB→CD, D→A

R_1(A, D), R_2(B, C, D)

LOSSLESS AND NOT FD PREVERSING

Q R (A, B, C, D)
A--> B, B→C, C-→D, D-→A
R1(A, B), R2(B, C) AND R3(C, D)
LOSSLESS AND FD PREVERSING

Q R(ABCDEG)(NF)R1(ABC) R2(ABDE) R3(EG)(LS)(DP)
AB>C
AC>B
AD>E
B>D
BC>A
E>G

Q R(ABCDE)(NF) R1(AB) R2(BC) R3(ABCD) R4(EG)(! LS)()
A>BC
C>DE
D>E

E.g.1 R (A, B, C, D)
F={AB->C, C->A, C->D}
Given decomposition as R1(A, C, D) and R2 (B, C)
Solution-
$R_1 \cup R_2 = R$ [True]
$R_1 \cap R_2 = C = \phi$
$C^+ = \{A, C, D\}$ So $C \to R_1$
Hence Given decomposition is LOSSLESS.

E.g. 2 R (A, B, C, D)
F={B->C, D->A}
Decomposition as –
$R_1(B, C)$ and $R_2(A, D)$
Solution-
1.) $R_1 \cup R_2 = R$ [True]
2.) $R_1 \cap R_2 = \phi$ [not satisfied]
Hence LOSSY Decomposition

Q Consider a scheme R (A B C D E)
F → D
D → F
C → A D

AB → C
is decomposed into R, (ABC)& R(CDE)is the decomposition lossy

Q Identify the lossy decomposition R (ABCD)F.D [A → B, B → C D]
a) (ABC) (BD) b) (AB) (BC)(CD) c) (AB) (CD) d) None

Q Identify the lossy decomposition on the relation R(ABCD) with functional dependencies
A →B
B→ C
C → D
a) (ABC) (BD) b) (AB) (BC) (CD) c) (AB)(CD) d) None of these

Q Consider the relation R (ABCD) with the FD set F = {A → BC, B → CD, C→ AD} which is decomposed into set of tables D = {(AB), (BC), CD}. Which of the following is true about the decomposition D?
a) It is lossless and dependency preserving
b) It is lossy but dependency preserving
c) It is lossless but dependencies are nor preserved
d) It is neither lossless nor dependency preserving

Q R(A,B,C,D) is a relation. Which of the following does not have a lossless join, dependency preserving BCNF decomposition? **(Gate - 2001) (2 Marks)**
(A) A->B, B->CD **(B)** A->B, B->C, C->D
(C) AB->C, C->AD **(D)** A ->BCD
Answer: (C)

Q Consider a schema R (A, B, C, D) and functional dependencies A->B and C->D
Then the decomposition of R into R1(AB) and R2(CD) is **(GATE-2001) (2 Marks)**
(A) dependency preserving and lossless join
(B) lossless join but not dependency preserving
(C) dependency preserving but not lossless join
(D) not dependency preserving and not lossless join
Answer: (C)

Q Let the set of functional dependencies F = {QR → S, R → P, S → Q} hold on a relation schema X = (PQRS). X is not in BCNF. Suppose X is decomposed into two schemas Y and Z, where Y = (PR) and Z = (QRS).
Consider the two statements given below.

I. Both Y and Z are in BCNF

II. Decomposition of X into Y and Z is dependency preserving and lossless

Which of the above statements is/are correct? **(GATE- 2019) (1 Marks)**

(a) I only

(b) Neither I nor II

(c) II only

(d) Both I and II

Ans: c

Q Consider a schema R(A, B, C, D) and functional dependencies A → B and C → D. Then the decomposition R1(A, B) and R2(C, D) is **(NET-JUNE-2012)** **(Gate - 2001) (2 Marks)**

(A) Dependency preserving but not lossless join

(B) Dependency preserving and lossless join

(C) Lossless Join but not dependency preserving

(D) Lossless Join

Ans: c

Q Consider a schema R(A, B, C, D) and following functional dependencies.

A → B

B → C

C → D

D → B

Then decomposition of R into R1(A, B), R2(B, C) and R3(B, D) is ___________. **(NET-NOV-2017)**

(1) Dependency preserving and lossless join.

(2) Lossless join but not dependency preserving.

(3) Dependency preserving but not lossless join.

(4) Not dependency preserving and not lossless join.

Ans: a

Q Consider a schema R(MNPQ) and functional dependencies M → N, P → Q. Then the decomposition of R into R1(MN) and R2(PQ) is _________. **(NET-JAN-2017)**

(1) Dependency preserving but not lossless join

(2) Dependency preserving and lossless join

(3) Lossless join but not dependency preserving

(4) Neither dependency preserving nor lossless join.

Ans: 1

Q Which of the following statements is TRUE? **(NET-JULY-2016)**

D1: The decomposition of the schema R(A, B, C) into R1(A, B) and R2 (A, C) is always lossless.

D2: The decomposition of the schema R(A, B, C, D, E) having AD → B, C → DE, B → AE and AE → C, into R1 (A, B, D) and R2 (A, C, D, E) is lossless.

(1) Both D1 and D2
(2) Neither D1 nor D2
(3) Only D1
(4) Only D2

Ans. 4

Only D2 is True because AD is key and present in both the tables. D1 is not always true because FD's not given and if we take B->A and C->A then it is lossy decomposition because no common attributes contain key from one of the tables.

Q Consider the table R with attributes A, B and C. The functional dependencies that hold on R are : A → B, C → AB. Which of the following statements is/are True? **(NET-AUG-2016)**

I. The decomposition of R into R1(C, A) and R2(A, B) is lossless.

II. The decomposition of R into R1(A, B) and R2(B, C) is lossy.

(1) Only I
(2) Only II
(3) Both I and II
(4) Neither I nor II

Ans: c

Q

If a relation with a Schema R is decomposed into two relations R_1 and R_2 such that $(R_1 \cup R_2) = R_1$ then which one of the following is to be satisfied for a lossless joint decomposition (→ indicates functional dependency)

(A) $(R_1 \cap R_2) \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$

(B) $R_1 \cap R_2 \rightarrow R_1$

(C) $R_1 \cap R_2 \rightarrow R_2$

(D) $R_1 \cap R_2 \rightarrow R_1$ and $R_1 \cap R_2 \rightarrow R_2$

Q (NET-DEC-2010)

Match the following :

I.	2 NF	(a)	transitive dependencies eliminated
II.	3 NF	(b)	multivalued attribute removed
III.	4 NF	(c)	contain no partial functional dependencies
IV.	5 NF	(d)	contains no join dependency

Codes :

	I	II	III	IV
(A)	(a)	(c)	(b)	(d)
(B)	(d)	(a)	(b)	(c)
(C)	(c)	(d)	(a)	(b)
(D)	(d)	(b)	(a)	(c)

Q The dependency preservation decomposition is a property to decompose database schema D, in which each functional dependency X → Y specified in F, **(NET-DEC-2010)**
(A) appeared directly in one of the relation schemas Ri in the decomposed D.
(B) could be inferred from dependencies that appear in some Ri.
(C) both (A) and (B)
(D) None of these
Ans c

Q The relation schemas R1 and R2 form a Lossless join decomposition of R if and only if: **(NET-JUNE-2015)**
(a) R1 ∩ R2 ↠ (R1 - R2)
(b) R1 → R2
(c) R1 ∩ R2 ↠ (R2 - R1)
(d) (R2 → R1) ∩ R2
(1) (a) and (b) happens
(2) (a) and (d) happens
(3) (a) and (c) happens
(4) (b) and (c) happens
Ans. 3

Q Relation R is decomposed using a set of functional dependencies, F and relation S is decomposed using another set of functional dependencies G. One decomposition is definitely BCNF, the other is definitely 3NF, but it is not known which is which. To make a guaranteed identification, which one of the following tests should be used on the decompositions? (Assume that the closures of F and G are available).(Gate-2002) (2 Marks)

(A) Dependency-preservation **(B)** Lossless-join

(C) BCNF definition **(D)** 3NF definition

Answer: (B)

Q Suppose R is a relation schema and F is a set of functional dependencies on R. Further, suppose R_1 and R_2 form a decomposition of R. Then the decomposition is a lossless join decomposition of R provided that: **(NET-DEC-2008)**

(A) $R_1 \cap R_2 \rightarrow R_1$ is in F^+

(B) $R_1 \cap R_2 \rightarrow R_2$ is in F^+

(C) both $R_1 \cap R_2 \rightarrow R_1$ and $R_1 \cap R_2 \rightarrow R_2$ functional dependencies are in F^+

(D) at least one from $R_1 \cap R_2 \rightarrow R_1$ and $R_1 \cap R_2 \rightarrow R_2$ is in F^+

Ans: d

Q Select the 'False' statement from the following statements about Normal Forms: **(NET-JUNE-2015)**

(1) Lossless preserving decomposition into 3NF is always possible

(2) Lossless preserving decomposition into BCNF is always possible

(3) Any Relation with two attributes is in BCNF

(4) BCNF is stronger than 3NF

Ans. 2

Q Which one of the following statements about normal forms is FALSE? (GATE-2005) (2 Marks)

(A) BCNF is stricter than 3NF

(B) Lossless, dependency-preserving decomposition into 3NF is always possible

(C) Lossless, dependency-preserving decomposition into BCNF is always possible

(D) Any relation with two attributes is in BCNF

Answer: (C)

Q Choose the correct statements.

I: 3 NF decomposition is always lossless join and dependency preserving.

II: 3 NF decomposition is always lossless join but may or may not be dependency preserving

III: BCNF decomposition always lossless join and dependency preserving.

IV: BCNF decomposition is always lossless join but may or may not be dependency preserving.

a) Both I and III b) Both II and IV c) Both I and IV d) Both II and III

Q Consider the given relation R (ABCDEFG) with attributes as follows: -

A→ Key Attribute
B and C → Atomic Attributes
D and E → Multi – valued Attributes
F and G → Composite Attributes
Which of the following is the correct 1 NF decomposition for the above relation?
a) R1 (ABC), R2(AD), R3 (AE), R4(AF), R5 (AG)
b) R1 (AB), R2(BC), R3 (DE), R4(FG)
c) R1 (ABC), R2 (ADE), R3 (AFG)
d) R1 (ABC), R2 (DEFG)

Database Design goals	1NF	2NF	3NF	BCNF
0% Redundancy	No	NO	No	Yes[due to FD's] NO[due to MVD's]
Lossless Decomposition	YES [Always]	YES [Always]	YES [Always]	YES [Always]
Dependency Preservation.	YES [Always]	YES [Always]	YES [Always]	May not be Always.