2 Normalization

We have discussed functional dependencies in chapter 1. In this chapter, we will dive deep and learn further things related to Functional dependencies.

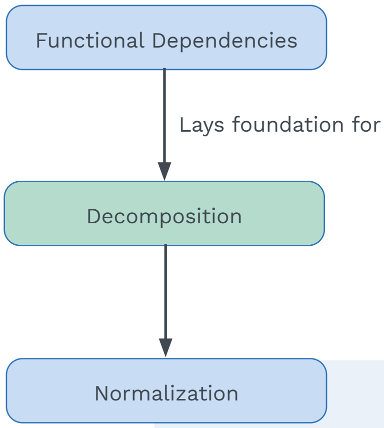


Fig. 2.1

First of all, let's revise some important points from previous chapter.

- Functional dependency is defined as a constraint between sets of attributes in a relation.
- We use '→' notation, which means LHS is functionally dependent on RHS.

We have covered following things previously,

- Functional dependencies
- Types of functional dependencies

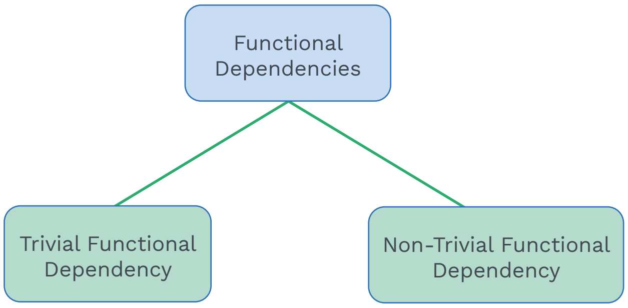


Fig. 2.2

- Rules of functional dependency
- Attribute closure
- Minimal cover
- A Problem caused by redundancy
- Equivalence set of functional dependency

Normalization

- "Normalization is used to organise data properly in tables.
- It is used to reduce the redundancy from a relation or set of relations."
- It is used to eliminate anomalies like insertion, deletion and updation.

Note:

Two or more relations when stored in single table may cause redundancy.
We can understand anomalies with the following example:

Let us consider there are three relations:

i) Sid → Sname, Sage
ii) Uid → Uname, Professor
iii) Sid, Uid → Fees

Combining these three independent relations into a single RDBMS table.

S_id	S_name	S_age	Uid	Uname	Professor	Fee
1	Arya	18	101	DU	Rosen	1000
2	Arya	18	101	DU	Rosen	1200
3	Bhumi	19	102	DU	Rosen	1200
4	Swati	17	102	MDU	Galvin	9000
4	Swati	17	103	MDU	Galvin	5000

Redundancy

Table 2.1

Now,

1) If we want to delete any student's data, the corresponding course data will also be deleted. This is a deletion anomaly. Similarly, if we are going to delete Uid, then corresponding Sid will also get deleted.

2) If we want to insert a university, then we have to add student information as well, which will cause an insertion anomaly.

3) If we want to update, for example, student age, this may cause inconsistency and give rise to an updation anomaly.

OR

If we want to delete/update referenced attribute value used by some referencing attribute, RDBMS won't allow deletion from referenced relation.

Normalization is process of decomposing relation into sub-relation, which must be in normal form.

Note:
If redundancy is reduced, then database anomalies can also be reduced.

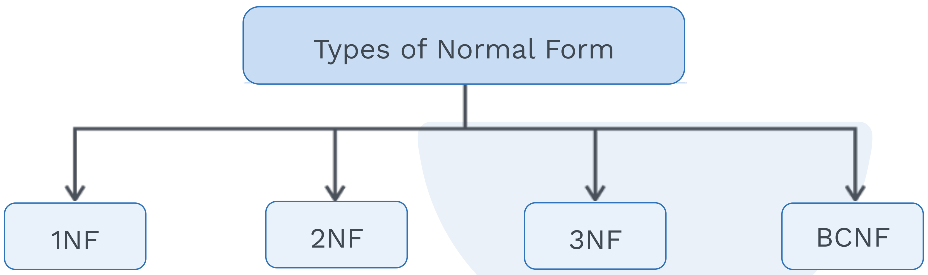


Fig. 2.3 Types of Normal Forms

2.1 FIRST NORMAL FORM (1NF)

"Any relation is said to be in 1NF if it has atomic value. Formally, an attribute of a table cannot hold multiple / Composite / Complex values."

Example: Following relation is not in 1NF because of attribute "S_course" having multiple values for a single student.

S_id	S_name	S_age	S_contact	S_cource
1	Abhi	20	9801010101	DS, Algo
2	Swati	19	9802020202	OS, CoA
3	Swarg	17	9803030303	TOC, CD
4	Shivam	21	9804040404	CN, DBMS

Table 2.2 First Normal Form

The above table can be converted into 1NF.

S_id	S_name	S_age	S_contact	S_cource
1	Abhi	20	9801010101	DS
1	Abhi	20	9801010101	Algo
2	Swati	19	9802020202	OS
2	Swati	19	9802020202	CoA
3	Swarg	17	9803030303	TOC
3	Swarg	17	9803030303	CD
4	Shivam	21	9804040404	CN
4	Shivam	21	9804040404	DBMS

Table 2.3

This is in 1NF, but the primary key will be {Sid, S_name, S_contact, S_course}

Issue with 1NF: 1NF is based on redundancy, Other normal forms came into existence, because of this problem To get rid of these anomalies, Normalization came into the picture.

Rack Your Brain

Consider the following table which is not in 1NF, how many rows will be there when converted to 1NF.

Student_id	Student_name	Contact
1	A	0101010101, 0202020202
2	B	0303030303
3	C	0404040404, 0505050505
4	D	0606060606

2.2 SECOND NORMAL FORM (2NF)

A relation will be in 2NF iff

- "It is in 1NF
- It has no Partial Dependency

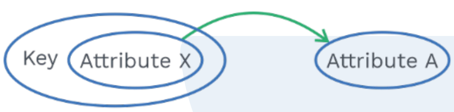
Partial dependency: "When a proper subset of a candidate is functionally dependent on non-prime attribute, this type is termed as Partial Dependency

OR

$$X \rightarrow A$$

X: Proper subset of a candidate key.

A: Non-prime attribute



Partial dependency (A is not the key)

Fig. 2.4

Example: Suppose a table student is given with FD = {Sid $\rightarrow$ Sname}

Student

Sid	Sname	Course_id
S_1	n_1	C_1
S_1	n_1	C_2
S_2	n_2	C_2
S_2	n_2	C_3

Table 2.4

Here, Candidate key = {(Sid, Course_id)}

Prime attribute = {Sid, Course_id}

Non-prime attribute = {Sname}

Given FD = [Sid $\rightarrow$ Sname]

Here, Sid is proper subset of candidate key and Sname is a non-prime attribute, therefore giving Partial Dependency,

SOLVED EXAMPLES

Q2 **Given a relation R(ABCDEFG) is already in 1NF with set of functional dependencies F = {AB → C, B → F, A → C, E → G}, check if relation is in 2NF**

Sol: Here, Candidate key = {ABDE}

Therefore, Prime attribute = {A, B, D, E}

Non-prime attribute = {C, F, G}

Here, AB → C, B → F, A → C, E → G are partial dependencies

Therefore, given relation is not in 2NF.

Rack Your Brain

Consider the table student with following functional dependencies

S_id → S_name
S_id → S_address
S_id → S_contact

Determine whether the relation is in 2NF or not?

Previous Years' Question

A database of research articles in a journal uses the following schema.
(VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, YEAR, PRICE)

The primary key is (VOLUME, NUMBER, STARTPAGE, ENDPAGE) and the following functional dependencies exist in the schema.

(VOLUME, NUMBER, STARTPAGE, ENDPAGE) → TITLE
(VOLUME, NUMBER) → YEAR
(VOLUME, NUMBER, STARTPAGE, ENDPAGE) → PRICE

The database is redesigned to use the following schemas
(VOLUME, NUMBER, STARTPAGE, ENDPAGE, TITLE, PRICE)
(VOLUME, NUMBER, YEAR)

Previous Years' Question_continued

Which is the weakest normal form that the new database satisfies, but the old one does not?

1) 1NF **2)** 2NF

3) 3NF **4)** BCNF

Sol: 2)

2.3 THIRD NORMAL FORM (3NF)

A relation R is in 3NF if for every FD X → A that holds over R, where X is a subset of the attributes of R and A be an attribute of R, one of the following statement is true:

- X is a trivial FD, or
- X is super key, or
- A is part of some key for R.

"Formally, we can say X → a; for this to be in 3NF, either X has to be superkey or 'a' has to be a prime attribute.

Suppose that a dependency X → A causes a violation of 3NF. There are two cases.

i) X is a proper subset of some key K, i.e. partial dependency.

ii) X is not a proper subset of any key, i.e. transitive dependency because it means we have a chain of dependencies K → X → A."

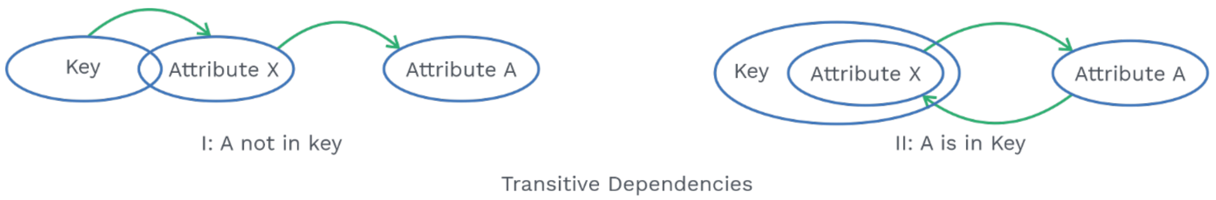


Fig.2.5

Q3 **Given a relation R (PQRSTU) with functional dependencies F = {PQ → R, R → S, S → T, Q → U}.**
Check for 3NF.

Sol: Here, Candidate key = {PQ}

Prime attribute = {P, Q}

Non-prime attribute = {R, S, T, U}

Now, Q → U is a Partial Dependency

R → S, S → T give rise to transitive dependency

Therefore, it is neither in 2NF nor in 3NF.

Rack Your Brain

Consider the table student with the following functional dependencies

S_id → S_name
S_id → S_address
S_id → S_contact

Determine whether the relation is in 3NF or not?

2.4 BOYCE-CODD NORMAL FORM (BCNF)

BCNF stands for Boyce-Codd Normal Form. Given a relation R and functional dependency, X → Y is in BCNF iff.

- $X \supseteq Y$, trivial
- X is a superkey

Note:

Every BCNF is in 3NF. There are very few examples which are in 3NF but not in BCNF.

If the relation is not in 1 NF, then it does not satisfy RDBMS rules.

Q4 **Given a relation R (ABCD) with set of functional dependencies F = {AB → CD, D → A}. Check for BCNF.**

Sol: Here, Candidate key = {AB, BD}

Prime attribute = {A, B, D}

Non-prime attribute = {C}

In FD, D → A, D is not a superkey.

Therefore, it is not in BCNF.

Rack Your Brain

Consider the table student with the following functional dependencies

S_id → S_name
S_id → S_address
S_id → S_contact

Determine whether the relation is in BCNF or not?

Rack Your Brain

Consider the following statements:
S1: Every relation which is in BCNF is also in 3NF.
S2: Every relation which is in 4NF is also in BCNF.

Which of the following option is True?

1) Only S1 is true
2) Only S2 is true
3) None of them is true
4) Both of them are true

Rack Your Brain

Given a relation R (PQRSTUVWXY) and FD set F = {PQ → R, P → ST, Q → T, T → VW, S → XY}.
Check for 2NF, 3NF and BCNF.

BCNF and multivalued dependencies:

- Where redundancy occurs because of presence of multivalued attributes, then this comes under 4NF.
- Multivalued Dependency is represented by either using '→ →' or ' →→ '.

Multivalued dependency:

When two attributes are independent of each other but on a third attribute, this gives rise to multi valued dependency.

Properties of multivalued dependency:

i) If $A \rightarrow\rightarrow B$ then $A \rightarrow\rightarrow D$ when $D = R$ [Complement property]

ii) If $A \supseteq B$ then $A \rightarrow\rightarrow B$

$A \cup B$ then $A \rightarrow\rightarrow B$

iii) Augmentation property:

$A \rightarrow\rightarrow B$ and $C \supseteq D$ (This is trivial dependency, $C \rightarrow D$ iff $D \subseteq E$) but augmentation property holds true for the non-trivial property as well, $AC \rightarrow\rightarrow BD$

iv) Transitivity property:

If $A \rightarrow\rightarrow B$ and $B \rightarrow\rightarrow C$ then $A \rightarrow\rightarrow C$

v) If $A \rightarrow B$, then $A \rightarrow\rightarrow B$

Note:

- If $A \rightarrow\rightarrow BC$ then MVD does not follow $A \rightarrow\rightarrow B$ and $A \rightarrow\rightarrow C$.
- If $A \rightarrow\rightarrow B$ and $A \rightarrow\rightarrow C$, MVD does not follow $A \rightarrow\rightarrow BC$.

Fourth normal form(4NF)

It is an extension of BCNF. A relation is in 4NF iff

i) It is in BCNF

ii) For MVD $A \rightarrow\rightarrow B$. A should be super key, and it should be determined using functional dependency, not using MVD.

Example: Consider the following table T,

T

A	B	C
1	P	R
1	P	S
1	Q	R
1	Q	S
2	Q	S

Let T is decomposed into T_1 and T_2

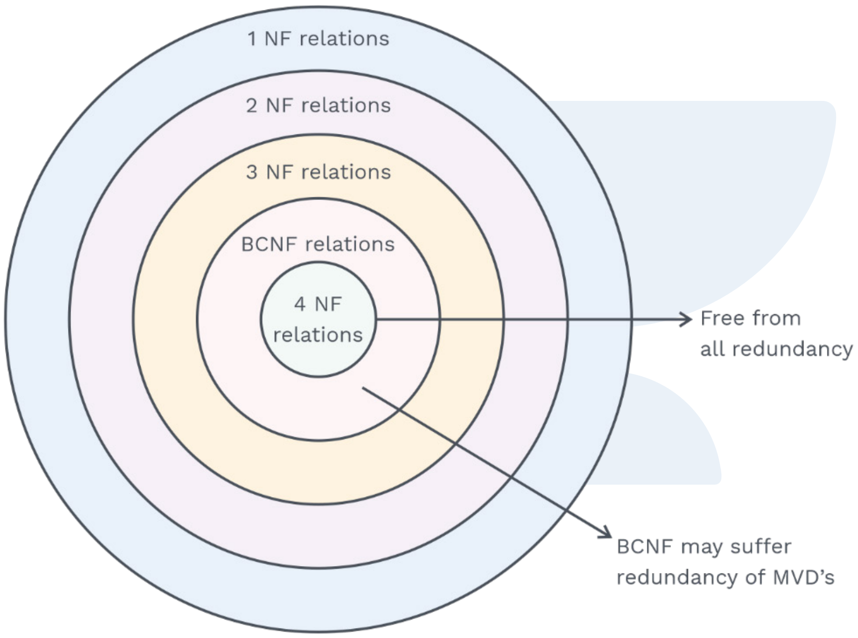
T_1

A	B
1	P
1	Q
2	Q

T_2

A	B
1	R
1	S
2	S

Here, Multivalued Dependency = {A →→ B, A →→ C}



Previous Years' Question

Given the following two statements:

S1: Every table with two single-valued attributes is in 1NF, 2NF, 3NF and BCNF.

S2: AB → C, D → E, E → C is a minimal cover for the set of functional dependenciesAB → C, D → E, AB → E, E → C.

Which one of the following is correct?

1) S1 is TRUE, and S2 is FALSE. **2)** Both S1 and S2 are TRUE.

3) S1 is FALSE, and S2 is TRUE. **4)** Both S1 and S2 are FALSE.

Sol: 1)

Previous Years' Question

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?

1) Both I and II **2)** I only

3) II only **4)** Neither I nor II

Sol: 3)

2.5 PROPERTIES OF DECOMPOSITION

When some relation is not in normal form, then that relation is decomposed into multiple relations, which is used to eliminate anomalies, redundancy and inconsistencies.

The two properties of decomposition are :

- Lossless Decomposition
- Dependency Preserving decomposition

Lossless join decomposition

Decomposition without any loss of data is termed lossless join decomposition.

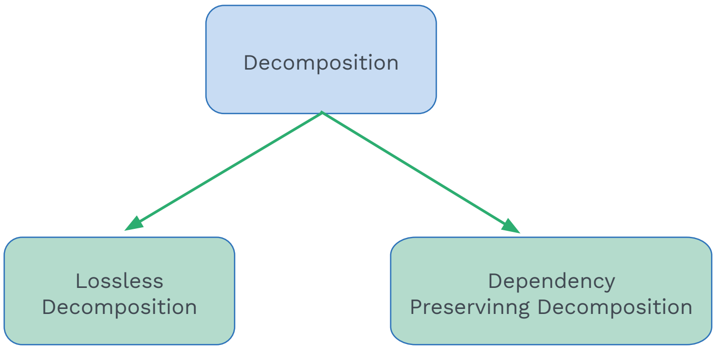
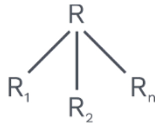


Fig. 2.5

Example: Generalization,



$[R_1 \bowtie R_2 \bowtie R_3 \ldots \bowtie R_n] \in R$

- If $(R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n) = R$, then it is lossless join decomposition.
- If $(R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n) \supset R$, then, it is lossy decomposition.

Note:
In lossy decomposition, some extra tuples are generated, called spurious tuples.

Example: Let a relation R with attributes emp_id, emp_name, dept_id.

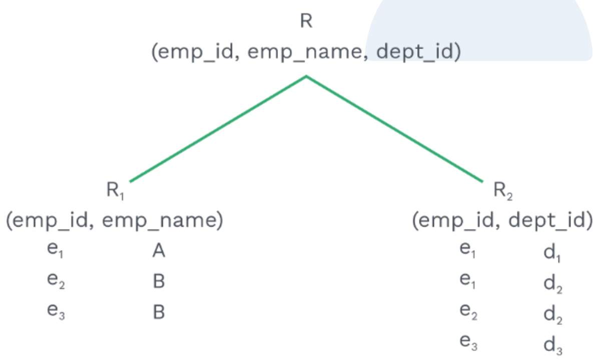
R

Emp_id	Emp_name	Dept_id
e_1	A	d_1
e_1	A	d_2
e_2	B	d_2
e_3	B	d_3

Table 2.5

- The candidate key of the above relation will be emp_id, dept_id

Case I: Let this relation is decomposed into two relations



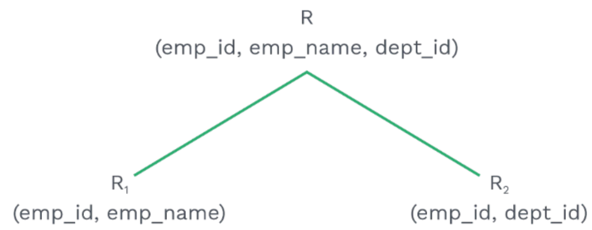
- Now joining R_1 and R_2, we get;

Emp_id	Emp_name	Dept_id
e_1	A	d_1
e_1	A	d_2
e_2	B	d_2
e_3	B	d_3

Table 2.6

$R_1 \bowtie R_2$ = R. that is lossless decomposition

Case II: Let the given relation R is decomposed into,



R_1

Emp_id	Emp_name
e_1	A
e_1	A
e_2	B
e_3	B

R_2

Emp_id	Dept_id
e_1	d_1
e_1	d_2
e_2	d_2
e_3	d_3

Now, $R_1 \bowtie R_2$ =

Emp_id	Emp_name	Dept_id
e_1	A	d_1
e_1	A	d_2
e_2	B	d_2
e_2	B	d_3
e_3	B	d_2
e_3	B	d_3

Table 2.7

Now, as we can see $(R_1 \bowtie R_2) \supset R$, therefore lossy join decomposition.

SOLVED EXAMPLES

Q1 **Given a relation R(A, B, C) with functional dependency {A → B, B → C}. Check if given relation is lossy or lossless, if relation is decomposed into**

i) R_1(AB) and R_2(BC) **ii) R_1(AC) and R_2(BC)**

Sol: Let's try to solve this question by taking counter example,

Now,

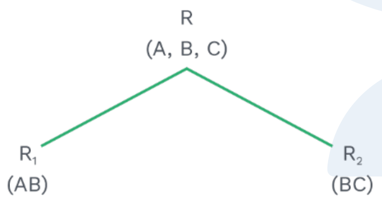
R

A	B	C
1	2	1
3	3	1
4	5	3
5	5	3
9	4	3

Candidate key = A

Now,

i) $R_1(AB)$ and $R_2(BC)$



R_1

A	B
1	2
3	3
4	5
5	5
9	4

R_2

B	C
2	1
3	1
5	3
5	3
4	3

$\Rightarrow$ Taking the join of R_1 and R_2, we get R. Therefore, lossless join decomposition.

ii) $R_1(AC)$ and $R_2(BC)$

As C is the common attribute and is not unique. Therefore, decomposition is lossy, $(R_1 \bowtie R_2) \supset R$.

Note:

Any relation is lossless iff

- $(R1 \cup R_2) = R$
- $(R1 \cap R_2) = R_1$ or $(R1 \cap R_2) = R_2$, formally, either $R1 \cap R_2$ is superkey/ candidate key of R_1 or $R1 \cap R_2$ is superkey of R_2.

Q2 **Consider a relation R(ABCDE) with functional dependencies, {AB → C, C → D, B → E} test whether given decompositions are lossless or lossy. Consider a relation R(ABCDE) with functional dependencies, {AB → C, C → D, B → E}**

Sol:

i) R1(ABC) and R2(CD)

$\Rightarrow$ $R_1 \cup R_2 \neq R$, $R_1 \cup R_2$ is giving ABCD, E being lost.

Therefore, lossy.

ii) R_1(ABC) and R_2(DE)

$\Rightarrow$ As, $R_1 \cup R_2 = R$, but no common attribute $(R_1 \cap R_2 = \phi)$

Therefore, lossy.

iii) R_1(ABC) and R_2(CDE)

$\Rightarrow$ As, $R_1 \cup R_2 = R$

$R_1 \cap R_2 = C$

But C is not a Candidate key, neither of R_1, nor R_2.

Therefore, lossy.

iv) R_1(ABCD) and R_2(BE)

$\Rightarrow$ $R_1 \cup R_2 = R$

$R_1 \cap R_2 = B$, which is the candidate key of R_2.

Therefore, lossless join decomposition.

v) R_1(ABCD) and R_2(ABE)

$\Rightarrow$ $R_1 \cup R_2 = R$

$R_1 \cap R_2 = \{A, B\}$

Now, $(AB)^+$ = ABCDE, superkey in both the relations

Therefore, lossless join.

Rack Your Brain

Consider the following relation R (A, B, C, D, E)

A	B	C	D	E
1	101	A	Alia	CSE
2	102	B	Swati	EE
1	103	A	Shruti	ME
3	104	C	Swayam	CE
4	105	D	Shivam	IT

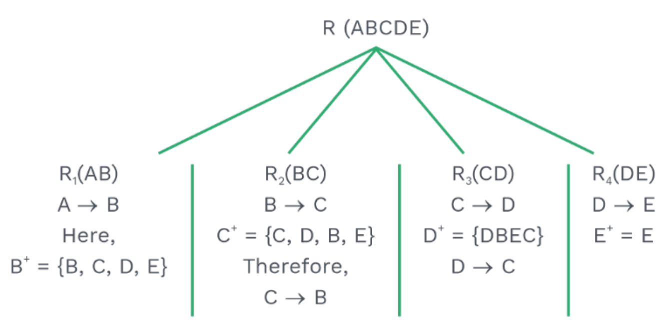
If the given relation is decomposed into two relations, R_1 (ABC) and R_2 (CDE), then check if they form lossless/lossy decomposition.

Dependency preserving decomposition:
A relation R is decomposed into a number of relations R_1, R_2, ... R_n with F_1, F_2, ... F_n Functional dependencies.
Then, $\{F_1 \cup F_2 \cup F_3 \cup \quad F_n\} \subseteq F$

- If $\{F_1 \cup F_2 \cup F_3 \quad \cup F_n\} = F$, then decomposition is dependency preserving decomposition, i.e. every FD of F is present in sub relation.
- If $\{F_1 \cup F_2 \cup F_3 \quad \cup F_n\} \subset F$, then decomposition is not dependency preserving decomposition.

Example: Given a relation R(ABCDE) with functional dependencies, {$A \rightarrow B$, $B \rightarrow C$,$C \rightarrow D$, $D \rightarrow BE$} decomposed to R_1(AB), R_2(BC), R_3(CD), R_4(DE). Check whether Dependency is preserved or not?

Sol:



Now, $A \rightarrow B$
$B \rightarrow C$
$C \rightarrow D$
$D \rightarrow BE$, because $D^+ = \{DBEC\}$

Therefore, $F_1 \cup F_2 \cup F_3 \cup F_4 = F$

Hence, Dependency is preserved.

Rack Your Brain

Given a relation R(ABCDEF) with functional dependencies $\{A \rightarrow BCDE, BC \rightarrow ADF, B \rightarrow F, D \rightarrow E\}$ is decomposed into ABC, BCD, BF, DE. Check if dependency can be preserved or not.

Rack Your Brain

Consider a table Employee:

Emp_id	Emp_name	Salary
1	A	50,000
2	B	55,000
3	C	60,000

This relation is decomposed into the relations emp_name and salary.

emp_name

Emp_id	Emp_name
A	1
B	2
C	3

salary

Emp_id	Salary
1	50,000
2	55,000
3	60,000

Check whether it satisfies Dependency preserving property or not?

Previous Years' Question

Let R(A, B, C, D) be a relational schema with the following functional dependencies:

$A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$ and $D \rightarrow B$. The decomposition of R into (A, B), (B, C), (B, D)

1) gives a lossless join and is dependency preserving

2) gives a lossless join but is not dependency preserving

3) does not give a lossless join but is dependency preserving

4) does not give a lossless join and is not dependency preserving

Sol: 3)

Previous Years' Question

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow TS$

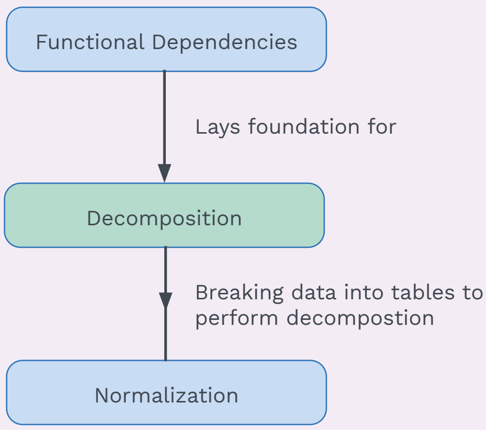
Which of the following functional dependencies can be inferred from the above functional dependencies?

1) $PS \rightarrow T$ **2)** $R \rightarrow T$

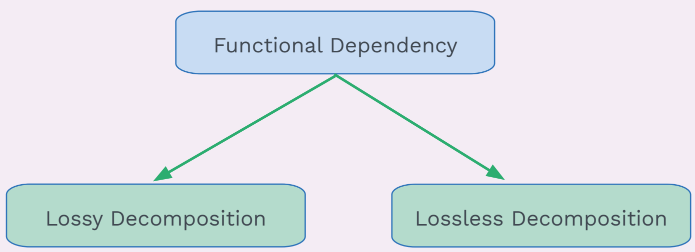
3) $P \rightarrow R$ **4)** $PS \rightarrow Q$

Sol: 1)

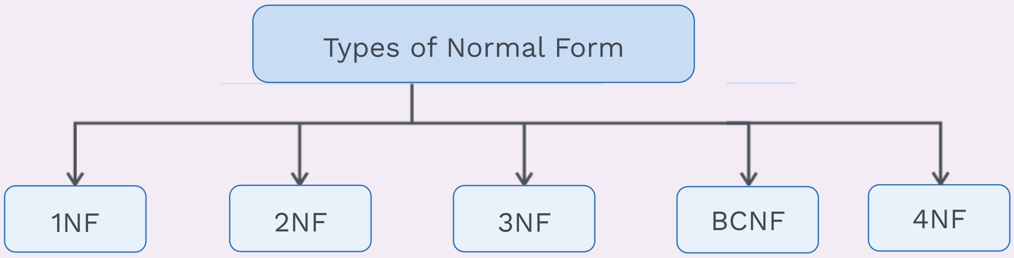
Chapter Summary

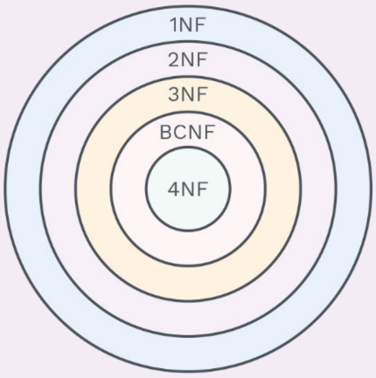
- 

- "Properties of decomposition:
  i) Lossless decomposition.
  ii) Dependency preserving decomposition."

- 

- Lossless decomposition:
  i) If $(R_1 \bowtie R_2 \bowtie R_3 \bowtie \ldots \bowtie R_n) = R$ then it is lossless Join decomposition
  ii) if $(R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n) \supset R$ then it is lossy.
- Dependency preserving decomposition
  i) If $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} = F$, then decomposition is dependency preserving property.
  ii) If $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} \subset F$, then it is not dependency preserving.

- 
Types of Normal Form
1NF
2NF
3NF
BCNF
4NF

- 
1NF
2NF
3NF
BCNF
4NF