

3.1 DATA BASE DESIGN

The overall design of the database is called the database schema.

Database system have several schemas, partitioned according to the level of abstraction.

- Physical schema
- Logical schema
- View schema

The relational schema face the several undesirable problems.

(1) **Redundancy** : The aim of the database system is to reduce redundancy, meaning that data is to be stored only once. Storing the data/information many times leads to the wastage of storage space and an increase in the total size of the data stored.

Name	Course	Phone_No	Major	Prof.	Grade
Vijay	160	2374539	Comp Sci	V. Singh	A
Sanjay	170	4277390	Physics	R. Singh	B
Vijay	165	2374539	Comp Sci	S. Singh	B
Gopal	456	3885183	Mathematics	R.J. Lal	A
Santosh	491	8237293	Chemistry	Ved Prakash	C
Santosh	356	8237293	Chemistry	J. Singh	A
Vijay	168	2374539	Comp Sci	Vinay	In prof.

Updates to the database with such redundencies the potential of recoming inconsistent. In the above table the major and phone no. of a student are stored many times in the database.

e.g., The major and phone no. of Vijay stored many times in the database. Thus it is the example of redundancy of data in the database.

(2) **Update Anomalies** : Multiple copies of the same fact may lead to update anomalies or inconsistencies. When an update is made and only some of the multiple copies are updated.

Thus, a change in the phone no. of 'Vijay' must be made for consistency, in all tuples pertaining to the student 'Vijay'. If one-three tuples in table is not change to reflect the new phone-no. of 'Vijay', there will be an inconsistency in the data.

e.g., In the given table the phone-no. of Vijay in all three row is 2374539 if we update the phone-no. of Vijay and two rows. Then database will be inconsistence.

Name	Phone_No.
Vijay	2374539
Vijay	3278435
Vijay	3278435

(3) **Insertion Anomalies** : If this is the only relation in the database showing the association

between a faculty member and the course he or she teaches, the fact that a given professor is teaching in a given course cannot be entered in the database unless a student is registered in the course.

(4) **Deletion Anomalies** : If the only student registered in a given course discontinues the course, the information as to which professor is offering the course will be lost if this is the only relation in the database showing the association between a faculty member and the course she or he teaches.

Example :

Roll-no.	Name	Course	Fee
10	Vijay	DBA	15000
11	Santosh	VB. Net	5000
12	Gopal	VC + +	8000
13	Sanjay	Java	7000

Here, we cannot delete the particular attribute Roll-no. = 11, because after this we cannot access the course, name and fee of that student due to lossing of whole information.

3.2 DECOMPOSITION

The decomposition of a relation schema $R = \{A_1, A_2, \dots, A_n\}$ is its replacement by a set of relation schemes $\{R_1, R_2, \dots, R_m\}$

such that $R_i \leq R, 1 \leq i \leq m$

and $R_1 \cup R_2 \cup R_3 \cup \dots \cup R_m = R$

A relation scheme R can be decomposed into a collection of relation schemes $\{R_1, R_2, R_3, \dots, R_m\}$ to eliminate some of the anomalies contained in the original relation R .

The problems in the relation scheme student can be resolved, if we decompose it with the following relation schemes : such as :

Student-INFO (Name, Phone no. Major)

Transcript (Name, Course, Grade)

Teacher (Course Professor)

These decomposition are bad for the following reasons :

- (i) Redundancy and update anomaly, because the data for the attributes phone no. and major are repeated.
- (ii) Loss of information, because we lose the fact that a student has a given grade in a particular course.

3.3 UNIVERSAL RELATION

Let us consider the problem of designing a database. Such a design will be required to represent a finite number of entity sets. Each entity set will be represented by a number of its attributes. If we refer to the set of all attributes as the universal scheme U then a relation $R(U)$ is called the universal relation.

The universal relation is a single relation made up of all the attributes in the database.

3.4 FUNCTIONAL DEPENDENCY

A functional dependency exists when the value of one thing is fully determined by another.

For example : Given the relation

EMP (Emp No, Emp Name, Salary), attribute EmpName is functionally dependent on attribute Emp No.

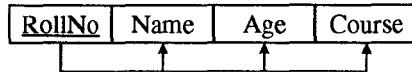
If we know Emp No, we also know the Emp Name. It is represented by

EMP No \rightarrow Emp Name.

A Functional dependency is a constraint between two sets of attributes in a relation from a database.

Types of Functional Dependency :

- (1) **Trivial FD** : A functional dependency of the form $X \rightarrow Y$ is trivial if $Y \subseteq X$.
- (2) **Full Functional Dependency** : A FD $X \rightarrow Y$ is a full functional dependency if removal of any



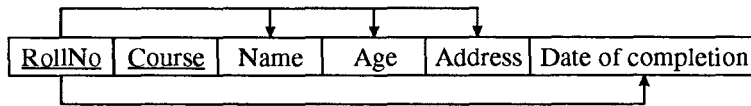
attribute A from X means that the dependency does not hold any more. That is,

Example Full FD :

for any attribute $A \in X$, $(X - \{A\})$ does not functionally determine Y .

$(X - \{A\}) \rightarrow Y$ is called full functional dependency.

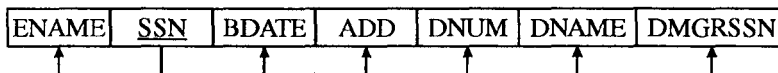
- (3) **Partial FD** : A FD $X \rightarrow Y$ is a partial dependency if some attribute $A \in X$ can be removed from X and then the dependency still hold.



That is if for some $A \in X$, $(X - \{A\}) \rightarrow Y$, then it is called partial dependency.

Example :

- (4) **Transitive Dependency** : A functional dependency $X \rightarrow Y$ in a relation scheme R is a transitive dependence if there is a set of attributes Z that is neither a candidate key nor a subset of any key of



R and both $X \rightarrow Z$ and $Z \rightarrow Y$ hold.

Ex :

The dependency $SSN \rightarrow DMGRSSN$ is transitive through
 $SSN \rightarrow DNUM$ and $DNUM \rightarrow DMGRSSN$

- (5) **Multivalued Dependency** : Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$.

The multivalued dependency $\alpha \twoheadrightarrow \beta$ hold on R if any legal relation $r(R)$, for all pairs of tuples t_1 and t_2 in r s.t. $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r s.t.

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$

$$t_3[\beta] = t_1[\beta]$$

$$t_3[R - \beta] = t_2[R - \beta]$$

$$t_4[\beta] = t_2[\beta]$$

$$t_4 [R - \beta] = t_1 [R - \beta]$$

Example : A table with schema (name, address, car)

Name	Address	Car
Vijay	Noida	Toyota
Vijay	G. Noida	Honda
Vijay	Noida	Honda
Vijay	G. Noida	Toyota

(name, address, car) where

name \twoheadrightarrow address

name \twoheadrightarrow car

3.5 PRIME ATTRIBUTE

An attribute of relation schema R is called a prime attribute of R if it is a member of some candidate key of R .

For Ex. : Work-on

SSN	PNUMBER	HOURS
-----	---------	-------

Both SSN and PNUMBER are prime attributes of work-on.

3.5.1 Non Prime Attribute

An attribute of relation schema R is called non prime attribute if it is not a member of any candidate key.

e.g., Hours is non prime attribute of work-on.

3.6 ARMSTRONG'S AXIOMS

There are following rules that logically implied functional dependency.

Suppose X, Y, Z denotes sets of attributes over a relational schema. Then the rules are :

(1) **Reflexivity :** If X is a set of attributes and $Y \subseteq X$

Then

$$X \rightarrow Y \text{ holds}$$

(2) **Augmentation Rule :**

If $X \rightarrow Y$ holds and Z is a set of attributes, then

$$ZX \rightarrow ZY \text{ holds}$$

(3) **Transitivity Rule :** If $X \rightarrow Y$ and $Y \rightarrow Z$ then

$$X \rightarrow Z \text{ hold}$$

These rules are called Armstrong's axioms.

There are some additional rules are :

(4) **Union rule :** If $X \rightarrow Y$ holds and $X \rightarrow Z$ holds

then $X \rightarrow YZ$ holds

(5) **Decomposition rule :** if $X \rightarrow YZ$ holds,

then $X \rightarrow Y$ holds and $X \rightarrow Z$ holds

(6) **Pseudo transitivity rule :** If $X \rightarrow Y$ holds and $ZY \rightarrow W$ holds

then $XZ \rightarrow W$ holds

For Example, A schema $R = (A, B, C, G, H, I)$ and Functional dependency are $\{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$ we can list several members of P^+ .

(i) Since $A \rightarrow B$ and $B \rightarrow H$ Applying transitivity

We get $A \rightarrow H$ holds

(ii) $\because CG \rightarrow I$, and $CG \rightarrow H$ Applying union rule

We get $CG \rightarrow HI$ holds

(iii) $\because A \rightarrow C$ and $CG \rightarrow I$ Applying pseudo transitivity

We get $AG \rightarrow I$ holds

3.7 CLOSURE OF A SET OF FUNCTIONAL DEPENDENCIES

The set of functional dependencies that is logically implied by F is called the closure of F and is written as F^+

Algorithm : Algorithm to compute the closure of X .

Let X^+ to all the attributes in X .

Determining X^+ , the closure of X under F .

$X^+ := X;$

repeat

old $X^+ := X^+;$

for each functional dependency $Y \rightarrow Z$ in F do

if $Y \subseteq X^+$ then

$X^+ := X^+ \cup Z;$

until $(X^+ = \text{old } X^+);$

Example : Let $R = \{A, B, C, D, E, F\}$ and a set of FDs.

$A \rightarrow BC, E \rightarrow CF, B \rightarrow E, CD \rightarrow EF, F \rightarrow D$

Compute the closure of a set of attribute $\{A, B\}$ under the given set of FDs.

Solution : Let $X = \{A, B\}$

Now initialization of X^+

$\therefore X^+ := X$

$\therefore X^+ = \{A, B\}$

For $A \rightarrow BC, \because A \subseteq X^+$ then

$X^+ = \{A, B\} \cup \{B, C\}$

$X^+ = \{A, B, C\}$

For $B \rightarrow E, \because B \subseteq X^+$ then

$X^+ = X^+ \cup \{E\}$

$= \{A, B, C\} \cup \{E\}$

$X^+ = \{A, B, C, E\}$

For $E \rightarrow CF, \because E \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= \{A, B, C, E\} \cup \{C, F\} \\ &= \{A, B, C, E, F\} \end{aligned}$$

For $F \rightarrow D$, $\therefore F \subseteq X^+$, then

$$\begin{aligned} X^+ &= \{A, B, C, E, F\} \cup \{D\} \\ &= \{A, B, C, D, E, F\} \end{aligned}$$

$$\therefore X^+ = \{A, B, C, D, E, F\}$$

Example : Let $X = BCD$ and $F = \{A \rightarrow BC, CD \rightarrow E, E \rightarrow C, D \rightarrow AEH, ABH \rightarrow BD, DH \rightarrow BC\}$

Compute the closure X^+ of X under F .

$$\therefore R = \{A, B, C, D, E, H\}$$

Solution : We initialize X^+ to X

$$\therefore X^+ = X,$$

$$\therefore X^+ = \{B, C, D\}$$

For $CD \rightarrow E$, $\therefore CD \subseteq X^+$

$$\therefore X^+ = X^+ \cup \{E\} = \{B, C, D\} \cup \{E\} := \{B, C, D, E\}$$

For $D \rightarrow AEH$, $\therefore D \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= X^+ \cup \{A, E, H\} \\ X^+ &= \{A, B, C, D, E, H\} \end{aligned}$$

Now X^+ cannot be augmented any further. Because

$$X^+ = \{A, B, C, D, E, H\} = R = \{A, B, C, D, E, H\}$$

$$\text{Thus } X^+ = \{A, B, C, D, E, H\}$$

3.8 NON REDUNDANT COVERS

Algorithm : Input : A set of FDS F

Output : A non redundant cover of F

$G := F$; (Initialize G to F)

for each $FD X \rightarrow Y$ in G

do

if $X \rightarrow Y \in \{F - (X \rightarrow Y)\}^+$

then $F := \{F - (X \rightarrow Y)\}$;

$G := F$; (G is the non redundant cover of F)

end;

Example : If $F = \{A \rightarrow BC, CD \rightarrow E, E \rightarrow C, D \rightarrow AEH, ABH \rightarrow BD, DH \rightarrow BC\}$. Then find the non-redundant cover for F .

Solution : We find that $(A)^+$ under $\{F - (A \rightarrow BC)\}$

$$\text{Let } X = A \therefore R = \{A, B, C, D, E, H\}$$

$$\therefore X^+ = X$$

$$\therefore X^+ = \{A\}$$

For all FDS, $(A)^+ = \{A\} \neq \{A, B, C, D, E, H\}$

Thus $A \rightarrow BC$ is non redundant.

Now for $CD \rightarrow E$, we find $(CD)^+$ under

$$\{F - (CD \rightarrow E)\}$$

Now $X^+ = \{C, D\}$

For $D \rightarrow AEH, \therefore D \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= X^+ \cup \{A, E, H\} \\ &= \{C, D\} \cup \{A, E, H\} \\ X^+ &= \{A, C, D, E, H\} \end{aligned}$$

For $A \rightarrow BC, \therefore A \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= X^+ \cup \{B, C\} \\ &= \{A, B, C, D, E, H\} = R = \{A, B, C, D, E, H\} \end{aligned}$$

Thus $CD \rightarrow E$ is **redundant** so it is removed.

Now for $DH \rightarrow BC$, we find $(DH)^+$ under $\{F - (DH \rightarrow BC)\}$

$$\therefore X^+ = \{D, H\}$$

$D \rightarrow AEH, \therefore D \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= X^+ \cup \{A, E, H\} \\ X^+ &= \{A, D, E, H\} \end{aligned}$$

$A \rightarrow BC, \therefore A \subseteq X^+$

$$\begin{aligned} \therefore X^+ &= X^+ \cup \{B, C\} \\ X^+ &= \{A, B, C, D, E, H\} = R = \{A, B, C, D, E, H\} \end{aligned}$$

Thus $DH \rightarrow BC$ is redundant removed it.

No remaining FDS can be from the modified F .

Thus a non redundant cover for

$$F \text{ is } \{A \rightarrow BC, E \rightarrow C, D \rightarrow AEH, ABH \rightarrow BD\}$$

3.9 CANONICAL COVER OR MINIMAL SET OF FD'S

A minimal cover of a set of functional dependencies E is a set of functional dependencies F that satisfies the property that every dependency in E is in the closure F^+ of F .

A set of functional dependencies F to be minimal if it satisfies the following condition.

- (i) Every dependency in F has a single attribute for its right-hand side.
- (ii) We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$, where Y is a proper subset of X and still have a set of dependencies that is equivalent to F .
- (iii) We cannot remove any dependency from F and still have a set of dependencies that is equivalent to F .

We can think of a minimal set of dependencies as being a set of dependencies in canonical form and with no redundancies.

A canonical cover is sometimes called minimal.

Example : If $F = \{A \rightarrow BC, CD \rightarrow E, E \rightarrow C, D \rightarrow AEH, ABH \rightarrow BD, DH \rightarrow BC\}$

Find the canonical cover.

Solution : First of all find the non redundant cover

\therefore The nonredundant cover for F is

$$\{A \rightarrow BC, E \rightarrow C, D \rightarrow AEH, ABH \rightarrow BD\}$$

Because $CD \rightarrow E$ and $DH \rightarrow BC$ are redundant FDS.

Now the FD $ABH \rightarrow BD$ can be decomposed into the FDS

$$ABH \rightarrow B \text{ and } ABH \rightarrow D$$

Similarly $A \rightarrow BC$ decompose into

$$A \rightarrow B \text{ and } A \rightarrow C$$

Since $A \rightarrow B$ is in F , we can left reduce the decomposition $ABH \rightarrow B$ and $ABH \rightarrow D$ into

$$AH \rightarrow B \text{ and } AH \rightarrow D$$

Now we also notice that $AH \rightarrow B$ is redundant because $A \rightarrow B$ is already in F .

Now we decompose the FD $D \rightarrow AEH$ into the FDS

$$D \rightarrow A, D \rightarrow E, D \rightarrow H$$

Thus the canonical over is

$$\{A \rightarrow B, A \rightarrow C, E \rightarrow C, D \rightarrow A, D \rightarrow E, D \rightarrow H, AH \rightarrow D\}$$

3.10 NORMALIZATION

The goal of a relational database design is to generate a set of relation schemas that allows us to store information without any redundant (repeated) data. It also allows us to retrieve information easily and more efficiently.

For this we use a approach normal form as the set of rules. These rules and regulations are known as Normalization.

Database normalization is data design and organization process applied to data structures based on their functional dependencies and primary keys that help build relational databases.

Normalization Helps :

- Minimizing data redundancy.
- Minimizing the insertion, deletion and update anomalies.
- Reduces input and output delays
- Reducing memory usage.
- Supports a single consistent version of the truth.
- It is an industry best method of tables or entity design.

Uses : Database normalization is a useful tool for requirements analysis and data modelling process of software development. Thus

The normalization is the process to reduce the all undesirable problems by using the functional dependencies and keys.

Here we use the following normal forms :

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal form (3NF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)
- Sixth Normal Form (6NF)

3.10.1 First Normal Form (1NF)

A relation schema *R* is said to be in First Normal Form (1NF) if the values in the domain of each attribute of the relation are atomic.

For Ex. : Course-INFO

Fact-Dept	Professor	Course Preferences	
		Course	Course-Dept
Comp-Sci	Vijay	353	Comp Sci
		370	Comp Sci
		310	Physics
	Santosh	353	Comp Sci
		320	Comp Sci
		370	Comp Sci
Chemistry	Gopal	456	Chemistry
		410	Mathematics
		370	Comp Sci

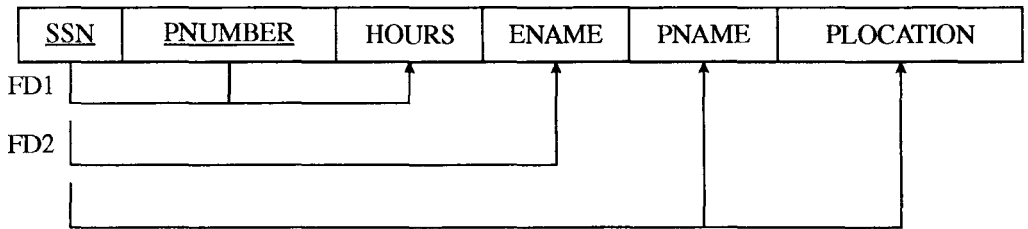
In 1NF : Course INFO

Professor	Course	Fact-Dept	Course
Vijay	353	Comp Sci	Comp Sci
Vijay	370	Comp Sci	Comp Sci
Vijay	310	Comp Sci	Physics
Santosh	353	Comp Sci	Comp Sci
Santosh	320	Comp Sci	Comp Sci
Santosh	370	Comp Sci	Comp Sci
Gopal	456	Chemistry	Chemistry
Gopal	410	Chemistry	Mathematics
Gopal	370	Chemistry	Comp Sci

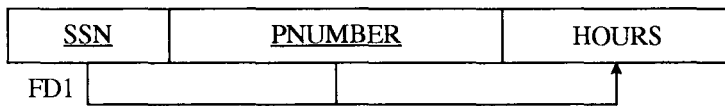
3.10.2 Second Normal Form (2NF)

- 2NF is a normal form in database normalization. It requires that all data elements in a table are full functionally dependent on the table's primary key.
- If data element only dependent on part of primary key, then they are parsed out to separate tables.

- If the table has a single field as the primary key, it is automatically in 2NF.



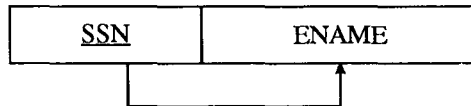
↓ 2NF



A table is in 2NF if and only if

- It is in 1NF
- Each non primary key attribute is full functionally dependent on the primary key.

Example 1 : EMP-PROJ



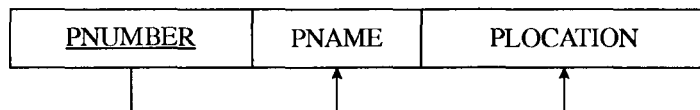
∴ {SSN, PNUMBER} is a primary key and Hours is non key attribute.

(SSN, PNUMBER) → HOURS

Thus HOURS is full FDS on primary key (SSN, PNUMBER)

EP2

SSN → ENAME

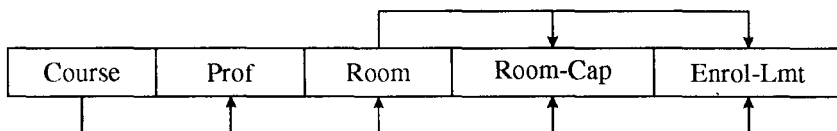


∴ SSN is primary key and NAME is a non key attribute

Since non key attribute ENAME is full FDS on primary key attribute SSN. Thus, it is in 2NF.

EP3

PNUMBER → {PNAME, PLOCATION}



Example 2 :

TEACHER :

Course	Prof	Room	Room-Cap	Enrol-Unt
353	Vijay	A532	45	40
351	Vijay	C320	100	60
355	Santosh	H940	50	45
456	Santosh	B278	50	45
459	Gopal	D110	300	200

TEACHER Relation in Second Normal Form

<u>Course</u>	Prof	Enrol-Unt
353	Vijay	40
351	Vijay	60
355	Santosh	45
456	Santosh	45
459	Gopal	200

(a)

<u>Course</u>	Room
353	A532
351	C320
355	H940
456	B278
459	D110

(b)

<u>Room</u>	Room-Cap
A532	45
C320	100
H940	50
B278	50
D110	300

(c)

3.10.3 Third Normal Form (3NF)

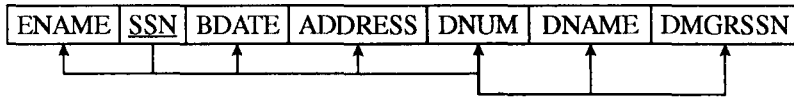
The 3NF is a normal form used in database normalization to check if all the non key attributes of a relation depend only on the candidate keys of the relation.

This means that all non-key attributes are mutually independent or in other words that a non key attribute cannot be transitively dependent on another non-key attribute.

A relation schema R is in 3NF if every non prime attribute of R meets both of the following.

- It is full functionally dependency on every key of R .
- It is non transitively dependent on every key of R .

OR we can say : A relation schema R is in 3NF if, whenever a non trivial functional dependency



$X \rightarrow A$ holds in R .

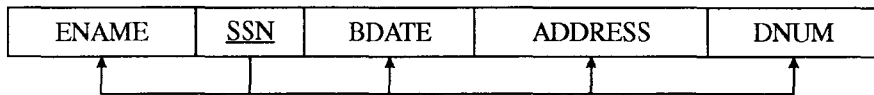
Either

- (a) X is a super key R . OR
- (b) A is a prime attribute of R .

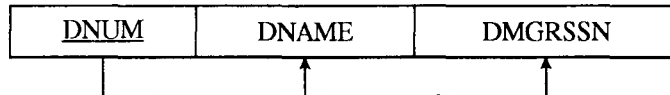
Example 1 : EMP-DEP

The dependency $SSN \rightarrow DMGRSSN$ is transitive through the FDS

$SSN \rightarrow DNUM$ and $DNUM \rightarrow DMGRSSN$



Thus EMP-DEP is not in 3NF because of the transitive dependency of DMGRSSN on SSN Via DNUM.



We can normalize EMP-DEP by decompose into two 3NF said ED_1 and ED_2 respectively.

Hence in 3NF :

- (i) ED_1
- (ii) ED_2

Example : Student relation

Roll_No	Name	Dept	Year	Hostel_Name
1784	Raman	Physics	1	Ganga
1648	Krishnan	Chemistry	1	Ganga
1768	Gopal	Maths	2	Kaveri
1848	Raja	Botany	2	Kaveri
1682	Maya	Geology	3	Krishna
1485	Singh	Zoology	4	Godavari

Here the dependency Roll-No \rightarrow HOSTAL NAME is transitive through
 $ROLL-NO \rightarrow YEAR$ AND $YEAR \rightarrow HOSTAL NAME$

Thus the student Relation is not 3NF.

So we can normalize student Relation by decomposition into two 3NF, STUD1 AND STUD2 respectively.

(i) STUD1 relation

<u>Roll-No</u>	Name	Dept	Year
1784	Raman	Physics	1
1648	Krishnan	Chemistry	1
1768	Gopal	Maths	2
1848	Raja	Botany	2
1682	Maya	Geology	3
1485	Singh	Zoology	4

(ii) STUD2 relation

<u>Year</u>	Hostel_Name
1	Ganga
2	Kaveri
3	Krishna
4	Godavari

3.10.4 Boyce-Codd Normal Form (BCNF)

BCNF is a normal form used in database normalization. It is slightly stronger version of the 3NF.

A table is in BCNF if and only if :

- (a) It is in 3NF and
- (b) For every of its nontrivial functional dependency $X \rightarrow Y$, X is a super key.

OR A relation schema R is in BCNF if whenever a nontrivial functional dependency $X \rightarrow A$ holds in R then

- (1) X is a super key of R .

Ex : Student

<u>Stud_Id</u>	SName	Subject	Grade
10	Vijay	Computer	A
10	Vijay	Physics	B
10	Vijay	Maths	B
20	Gopal	Computer	A
20	Gopal	Physics	A
20	Gopal	Maths	C

There are two candidate keys (Stud_Id, Subject) and (SName, Subject)

In the above relation following FDS are exist :

- SName, Subject \rightarrow Grade
- Stud_Id, Subject \rightarrow Grade
- Stud_Id \rightarrow SName

Now BCNF decompose R into R_1 and R_2 .

R_1

Stud_Id	Subject	Grade
10	Computer	A
10	Physics	B
10	Maths	B
20	Computer	A
20	Physics	A
20	Maths	C

R_2

Stud_Id	SName
10	Vijay
20	Gopal

Ex. 2 : Normalize the relation professor so as it is in BCNF.

PROFESSOR

Prof Code	Department	HOD	Percent Time
P ₁	Physics	Ghosh	50
P ₁	Maths	Krishnan	50
P ₂	Chemistry	Rao	25
P ₂	Physics	Ghosh	75
P ₃	Maths	Krishnan	100
P ₄	Maths	Krishnan	30
P ₄	Physics	Ghosh	70

The FDS are

Prof Code, Department \rightarrow Percent time

Department \rightarrow HOD

The PROFESSOR Relation decompose into two relation PROF 1 and PROF 2 respectively.

PROF1

Professor Code	Department	Percent Time
P ₁	Physics	50
P ₁	Maths	50
P ₂	Chemistry	25
P ₂	Physics	75
P ₃	Maths	100
P ₄	Maths	30
P ₄	Physics	70

PROF2

Department	HOD
Physics	Ghosh
Maths	Krishnan
Chemistry	Rao

Note : Every relation in BCNF is also in 3NF, but a relation is 3NF is not necessarily in BCNF
e.g., The above relations are in BCNF and 3NF also.

TEACH

Student	Course	Instructor
Narayan	Database	Pallaw
Vijay	Database	Navathe
Vijay	OS	Galvin
Vijay	Computer	Gopal
Santosh	OS	Ahmad
Santosh	Database	Pallaw

The dependencies show as

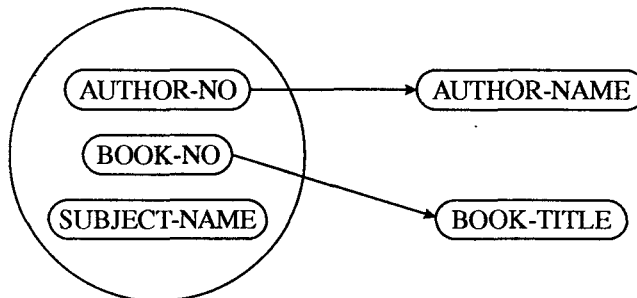
{STUDENT, COURSE} → INSTRUCTOR

INSTRUCTOR → COURSE

But the TEACH Relation is in 3NF but not in BCNF.

3.10.5 Fourth Normal Form (4NF)

- An entity type is in 4NF if it is BCNF and there are non multivalued dependencies between its attribute types.
- Any entity is BCNF is transformed in 4NF
 - (i) Direct any multivalued dependencies.
 - (ii) Decompose entity type.



Example :

Author_No	Book_No	Subject	Book_Title	Author_Name
A ₁	B ₁	Comp Sci	Methods	Vijay
A ₁	B ₁	Maths	Methods	Vijay
A ₂	B ₁	Comp Sci	Methods	Gopal
A ₂	B ₁	Maths	Methods	Gopal
A ₁	B ₂	Maths	Calculus	Santosh

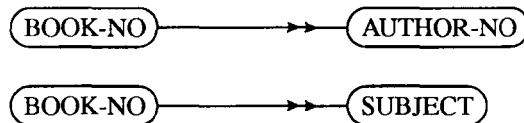
IN BCNF

AUTHOR (Author-No, Author-Name)

BOOK (Book-No, Book-title)

AUTHOR-BOOK-SUBJECT (Author-No, Book-No, Subject)

- Example models that “each AUTHOR is associated with all the SUBJECTS under which the Book is classified.
- The attribute SUBJECT contains redundant values. If SUBJECT were delete from row 1 and 2 the value could be deduced from row 3 and 4.



Multivalued dependencies :

Author_No	Book_No	Subject
A ₁	B ₁	Comp Sci
A ₁	B ₁	Maths
A ₂	B ₁	Comp Sci
A ₂	B ₁	Maths
A ₁	B ₂	Maths

Now The 4th NF will be

Author_No	Book_No
A ₁	B ₁
A ₂	B ₁
A ₁	B ₂

Book_No	Subject
B ₁	Comp Sci
B ₁	Maths
B ₂	Maths

AUTHOR (Author-No, Author-Name)

BOOK (Book-No, Book-Title)

AUTHOR-BOOK (Author-No, Book-No)

BOOK-SUBJECT (Book-No, Subject)

Example 2 : Faculty Relation (BCNF Form)

Faculty	Subject	Institute
Vijay Krishna	DBMS	IILM
Vijay Krishna	Data Structure	IILM
Vijay Krishna	C + +	IILM
Vijay Krishna	DBMS	GIMT
Vijay Krishna	Data Structure	GIMT
Gopal Krishna	Data Structure	GIMT
Gopal Krishna	Java	IILM

4NF

Faculty Subject Relation

Faculty	Subject
Vijay Krishna	DBMS
Vijay Krishna	C + +
Vijay Krishna	Data Structure
Gopal Krishna	Data Structure
Gopal Krishna	Java

Faculty Institute Relation

Faculty	Institute
Vijay Krishna	IILM
Vijay Krishna	GIMT
Gopal Krishna	GIMT
Gopal Krishna	IILM

Relation in 4NF

Note : 4NF eliminates MVD relationships. Thus no table can have more than a single many-to-one or many-to-many relationships which are not directly related.

3.10.6 Fifth Normal Form (5NF)

A table is said to be in the 5NF if and only if it is in 4NF and every Join dependency in it is implied by the candidate key.

Consider the following example :

Psychiatrist-to-Insurer-to-Condition

Psychiatrist	Insurer	Condition
Dr. Vijay	Healthco	Anxiety
Dr. Vijay	Healthco	Depression
Dr. Santosh	Friendly Care	Dementia
Dr. Santosh	Friendly Care	Anxiety
Dr. Santosh	Friendly Care	Depression
Dr. Santosh	Friendly Care	Mood Disorder
Dr. Gopal	Friendly Care	Schizophrenia
Dr. Gopal	Healthco	Anxiety
Dr. Gopal	Healthco	Dementia
Dr. Gopal	Victorian Life	Conversion Disorder

To split the relation into three parts

Psychiatrist-to-Condition

Psychiatrist	Condition
Dr. Vijay	Anxiety
Dr. Vijay	Depression
Dr. Santosh	Dementia
Dr. Santosh	Anxiety
Dr. Santosh	Depression
Dr. Santosh	Mood Disorder
Dr. Gopal	Schizophrenia
Dr. Gopal	Anxiety
Dr. Gopal	Dementia
Dr. Gopal	Conversion Disorder

Psychiatrist-to-Insurer

Psychiatrist	Insurer
Dr. Vijay	Healthco
Dr. Santosh	Friendly Care
Dr. Gopal	Friendly Care
Dr. Gopal	Healthco
Dr. Gopal	Victorian Life

Insurer-to-Condition

Insurer	Condition
Healthco	Anxiety
Healthco	Depression
Healthco	Dementia
Friendly Care	Dementia
Friendly Care	Anxiety
Friendly Care	Depression
Friendly Care	Mood Disorder
Friendly Care	Schizophrenia
Victorian Life	Conversion Disorder

3.10.7 Sixth Normal Form

This normal form was, as of 2005 only recently proposed.

The 6NF was only defined when extending the relational model to take into account the temporal dimension unfortunately, most current SQL technologies as of 2005 do not take into account this work, and most temporal extensions to SQL are not relational.

3.10.8 Domain/Key Normal Form

Domain/Key Normal Form is a normal form used in database normalization which required that the database contains no constraints other than domain constraints and key constraints.

A domain constraint specifies the permissible values for a given attribute, while a key constraint specifies the attributes that uniquely identify a row in a given table.

The domain/Key NF is the Holy Grail of relational database design, achieved when every constraint on the relation is a logical consequence of the definition of keys and domains, and enforcing key and domain restraints and conditions causes all constraints to be met.

Thus it avoids all non-temporal anomalies It's must easier to build a database in domain/key normal form than it is to convert lesser databases which may contains numerous anomalies.

However, successfully building a domain/key normal form data base remains a difficult task, even for experienced database programmers.

Thus, while the domain/key Normal form eliminates the problems found in most databases it tends to be the most costly normal form to achieve.

3.10.9 Conclusion of Database Normalization

Data Normalization is a technique that ensures some basic properties :

- No duplicate tuples.
- No Nested Relations.

Data Normalization is often used as the only technique for database design-implementation view. A more appropriate approach is to complement conceptual modelling with data Normalization.

3.11 LOSSLESS-JOIN DECOMPOSITION

- Let R be a relation schema.
- Let F be a set of functional dependency of R .
- Let R_1 and R_2 from decomposition of R .

The decomposition is a loseless-Join decomposition of R if at least one of the following FDS are in F^+ .

- (1) $R_1 \cap R_2 \rightarrow R_1$
- (2) $R_1 \cap R_2 \rightarrow R_2$

Why is this true ? Simply put, it ensures that the attributes involved in the natural join ($R_1 \cap R_2$) are a candidate key for at least one of the two relations.

This ensures that we can never get the situation where spurious tuples are generated, as for any value on the join attributes there will be a unique tuple in one of the relations.

Ex. : Let $R = \{A, B, C, D\}$
 $F = \{A \rightarrow B, B \rightarrow C\}$

Suppose decompositions $R_1 = \{A, B\}$, $R_2 = \{B, C\}$ and $R_3 = \{A, D\}$. Find the decompositions is lossless or lossy.

Ans. (1) Consider R_1 and R_3

$$R_1 \cap R_3 = \{A, B\} \cap \{A, D\} = \{A\}$$

Since $A \rightarrow B$ and A is a key in R_1 .

$$R_1 \cap R_3 \rightarrow R_1 = \{A, B\} \qquad \because A \rightarrow B$$

Let us union R_1 and R_3 and form R_4 .

$$\therefore R_4 = \{A, B\} \cup \{A, D\} = \{A, B, D\}$$

The decomposition of (A, B, D) into R_1 and R_3 is lossless-Join.

(2) Next consider R_4 and R_2

$$\text{Now } R_4 \cap R_2 = \{A, B, D\} \cap \{B, C\} = \{B\}$$

Since $B \rightarrow C$ and B is a key in R_2

$$\therefore R_4 \cap R_2 \rightarrow R_2 = \{B, C\} \qquad \therefore B \rightarrow C$$

The decomposition of (A, B, C, D) into R_2 and R_4 is lossless-Join.

Solved Problems

Q. 1. Consider the schema $R = (V, W, X, Y, Z)$ suppose the following FDs hold :

$$F = \{Z \rightarrow V, W \rightarrow Y, XY \rightarrow Z, V \rightarrow WX\}$$

State whether the following decomposition of schema R is loss-less join decomposition. Justify your answer. (UPTU 2003, 05)

Ans. For the decomposition of relation schema R into R_1 and R_2 to lossy or lossless either any one of the following conditions hold.

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

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

(i) Considering the first decomposition

$$R_1 = (V, W, X)$$

$$R_2 = (V, Y, Z)$$

$$R_1 \cap R_2 = \{V\}$$

Since $V \rightarrow WX$ and V is a key in R_1

$$\therefore R_1 \cap R_2 \rightarrow R_1 \text{ because } V \rightarrow VWX = R_1$$

Thus we can say the decomposition

$$R_1 = (V, W, X)$$

$$R_2 = (V, Y, Z) \text{ is lossless-decomposition.}$$

(ii) Now considering the second decomposition

$$R_1 = (V, W, X)$$

$$R_2 = (X, Y, Z)$$

$$R_1 \cap R_2 = \{X\}$$

So either

$$X \rightarrow VWX$$

$$[\because R_1 \cap R_2 \rightarrow R_1]$$

or

$$X \rightarrow XYZ$$

$$[\because R_1 \cap R_2 \rightarrow R_2]$$

But using the given set of FDS, we can not get.

$$\text{Either } X \rightarrow VWX$$

$$\text{or } X \rightarrow XYZ$$

Thus the decomposition

$$R_1 = (V, W, X)$$

$$R = (X, Y, Z) \text{ is lossy decomposition.}$$

An other method : For lossless-join decomposition :

Q. 2. Consider the scheme $R = (A, B, C, D, E)$ suppose following FDS hold :

- $E \rightarrow A$
- $CD \rightarrow E$
- $A \rightarrow BC$
- $B \rightarrow D$

State, whether the following decomposition of R are lossless join decomposition or not, justify

1. $\{(A, B, C), (A, D, E)\}$
2. $\{(A, B, C), (C, D, E)\}$

(UPTU 2002, 03, 04)

Ans. (1) $\{(A, B, C), (A, D, E)\}$

Let $R_1 = (A, B, C)$

$R_2 = (A, D, E)$

Put α in table where attribute is exist in relation and put β where they do not exist in relation.

Now we get the following table :

	A	B	C	D	E
R_1	α_A	α_B	α_C	β_1	β_1
R_2	α_A	β_2	β_2	α_D	α_E

After seeing the table we came to know that column A have common α and $A \rightarrow BC$
So we can put α in Row R_2 column B and C. After that we get the following table.

	A	B	C	D	E
R_1	α_A	α_B	α_C	β_1	β_1
R_2	α_A	α_B	α_C	α_D	α_E

Since Row R_2 has all α .

So the decomposition is lossless.

(2) $\{(A, B, C), (C, D, E)\}$

$R_1 = (A, B, C)$

$R_2 = (C, D, E)$

- Put α in table where attribute is exist in relation.
- Put β where they do not exist in relation.

Now we get the following table :

	A	B	C	D	E
R_1	α_A	α_B	α_C	β_1	β_1
R_2	β_2	β_2	α_C	α_D	α_E

After seeing the table we came to know that, column C have common α .

In given FDs, C does not implies any value, so we can not update table.

So the decomposition

$R_1 = (A, B, C)$

$R_2 = (C, D, E)$ is lossy.

Q. 3. Given $R = (A, B, C, D, E)$ with the FDs.

$$F = \{AB \rightarrow CD, A \rightarrow E, C \rightarrow D\}.$$

Find, if the decomposition of R into

$R_1 = (A, B, C)$, $R_2 = (B, C, D)$ and $R_3 = (C, D, E)$ is lossy or not.

(UPTU 2003-04)

Ans. Given $R = (A, B, C, D, E)$

The decomposition of R into three relations

$$R_1 = (A, B, C)$$

$$R_2 = (B, C, D)$$

$$R_3 = (C, D, E)$$

and

$$F = \{AB \rightarrow CD, A \rightarrow E, C \rightarrow D\}$$

- Put α in table where attribute is exist in relation and
- Put β where they do not exist in relation.

So we get the table.

	A	B	C	D	E
R_1	α_A	α_B	α_C	β_1	β_2
R_2	β_2	α_B	α_C	α_D	β_2
R_3	β_3	β_3	α_C	α_D	α_E

After seeing the table we came to know that, column C have common α and FD $C \rightarrow D$.

So we can put α in Row R_1 column D .

After that we get the new table.

	A	B	C	D	E
R_1	α_A	α_B	α_C	α_D	β_1
R_2	β_2	α_B	α_C	α_D	β_2
R_3	β_3	β_3	α_C	α_D	α_E

Since any row in this table does not contains all α .

Thus this decomposition is lossy.

Q.4. Given $R (A, B, C, D)$ with the FDs

$$F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D\}.$$

Find if the decomposition of R into $R_1 (A, B, C)$ and $R_2 (C, D)$ is lossy or not.

Ans. $R_1 = (A, B, C)$

$$R_2 = (C, D)$$

- Put α in table where attribute is exist in Relation and
- Put β where they do not exist in Relation.

Now we get the following table.

	A	B	C	D
R_1	α_A	α_B	α_C	β_1
R_2	β_2	β_2	α_C	α_D

After seeing the table we came to know that column C have common α and FD $C \rightarrow D$.

So we can put α in Row R_1 column D .
 So after that we get the table.

	A	B	C	D
R_1	α_A	α_B	α_C	α_D
R_2	β_2	β_2	α_C	α_D

Since Row R_1 has all α .
 So the decomposition $R_1 (A, B, C)$ and $R_2 (C, D)$ is lossless.

Q. 5. Given $R (A, B, C, D, E)$ with FDs

$$F = \{AB \rightarrow CD, A \rightarrow E, C \rightarrow D\},$$

the decomposition of R into $R_1 (A, B, C)$, $R_2 (B, C, D)$ and $R_3 (C, D, E)$ is lossless or lossy.

Ans. Given $R (A, B, C, D, E)$

Decomposition of R into three relations

$$R_1 = (A, B, C)$$

$$R_2 = (B, C, D)$$

$$R_3 = (C, D, E)$$

The FD $F = \{AB \rightarrow CD, A \rightarrow E, C \rightarrow D\}$

- Put α in the table where attribute is exist in relation and
- Put β where they do not exist in relation.

We get the following table.

	A	B	C	D	E
R_1	α_A	α_B	α_C	β_1	β_1
R_2	β_2	α_B	α_C	α_D	β_2
R_3	β_3	β_3	α_C	α_D	α_E

After seeing the table we came to know that column C have common α and FD $C \rightarrow D$.
 So we can put α in row R_1 column D .

After that we get the table.

	A	B	C	D	E
R_1	α_A	α_B	α_C	α_D	β_1
R_2	β_2	α_B	α_C	α_D	β_2
R_3	β_3	β_3	α_C	α_D	α_E

No further changes are possible and the final version of the table is the same as the table above.
 Finally we find no rows in the table with all α s.

Hence the decomposition is lossy.

Q. 6. Consider the relational schema

$R (A, B, C, D, E, F, G, H)$ with the FDs

$$F = \{AB \rightarrow C, BC \rightarrow D, E \rightarrow F, G \rightarrow F, H \rightarrow A, FG \rightarrow H\}$$

Is the decomposition of R into $R_1 (A, B, C, D)$, $R_2 (A, B, C, E, F)$ and $R_3 (A, D, F, G, H)$ lossless ?
 (UPTU 2005-06)

Ans. Given $R (A, B, C, D, E, F, G, H)$

The decomposition of R into R_1, R_2, R_3 . S.T.

$$R_2 = (A, B, C, E, F), R_1 = (A, B, C, D)$$

$$R_3 = (A, D, F, G, H)$$

FDs $F = \{AB \rightarrow C, BC \rightarrow D, E \rightarrow F, G \rightarrow F, H \rightarrow A, FG \rightarrow H\}$

- Put α in table where attribute is exist in Relation, and
- Put β where they do not exist in relation.

We get the following table.

	A	B	C	D	E	F	G	H
R_1	α_A	α_B	α_C	α_D	β_1	β_1	β_1	β_1
R_2	α_A	α_B	α_C	β_2	α_E	α_F	β_2	β_2
R_3	α_A	β_3	β_3	α_D	β_3	α_F	α_G	α_H

After seeing the table we came to know that column A have common α . But in given FDs A does not implies any value.

So we cannot update table.

Since any row in this table does not contain all α s.

Hence, this decomposition is lossy.

Q. 7. Given the relational schema $R(A, B, C, D, E)$ and given the following set of FDs defined on R .

$$F = \{A \rightarrow BC, CD \rightarrow E, AC \rightarrow E, B \rightarrow D, E \rightarrow AB\}$$

- (a) Determine a lossless-join decomposition of R .
- (b) Determine a decomposition of R which is not lossless-join.
- (c) Determine if R is in 3NF w.r.to F if it is not violating the 3NF.

(UPTU 2006)

Ans. (a) One out of many possible lossless-join decomposition

Let $R_1 = (A, D, E), R_2 (A, B, C)$

These two decompositions R_1 and R_2 of R is lossless-join decomposition.

This is because

$$\begin{aligned} R_1 \cap R_2 &= \{A, D, E\} \cap \{A, B, C\} \\ &= \{A\} \end{aligned}$$

Since $A \rightarrow BC$ and A is a key in R_2 .

$$\therefore R_1 \cap R_2 = \{A, B, C\} = R_2.$$

(b) The decomposition of R is R_1 and R_2 :

Let $R_1 = \{A, B, C, D\}$ and $R_2 = \{B, E\}$

This decomposition is a lossy decomposition because

$$\begin{aligned} R_1 \cap R_2 &= \{A, B, C, D\} \cap \{B, E\} \\ &= \{B\} \end{aligned}$$

$$\therefore B \rightarrow D$$

Therefore, $B \rightarrow R_1$ or R_2 , i.e., $\{BD \rightarrow R_1 \text{ or } R_2\}$

(c) The candidates key of R are A

because $A \rightarrow BC$

$$\therefore A^+ = \{A, B, C\}$$

$$B \rightarrow D \therefore B \subseteq A^+$$

$$\therefore A^+ = A^+ \cup \{D\}$$

$$A^+ = \{A, B, C, D\}$$

$$AC \rightarrow E \therefore AC \subseteq A^+$$

$$\therefore A^+ = A^+ \cup E$$

$$= \{A, B, C, D, E\} = R.$$

Thus A is candidate key.

Similarly $CD, AC,$ and E are also candidate key. Therefore, the candidate key of R are A, AC, CD and E .

Thus there are no dependencies violating R
Hence it is in 3NF.

Q. 8. For each of the following relation schemas and set of FDs.

(1) R is (A, B, C, D) with FDs.

$$F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A\}$$

(2) R is (A, B, C, D) with FDs

$$F = \{B \rightarrow C, B \rightarrow D\}$$

(i) Identify candidate keys for R .

(ii) Indicate BCNF violations and decompose if necessary.

(iii) Indicate 3NF violations and decompose if necessary.

Ans. (1) Given $R(A, B, C, D)$ with FDs

$$F(A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A)$$

(i) For $A \rightarrow B, \therefore A^+ = \{A, B\}$

$B \rightarrow C, \therefore B \subseteq A^+$

$$\therefore A^+ = A^+ \cup \{C\}$$

$$A^+ = \{A, B, C\}$$

$C \rightarrow D, \therefore C \subseteq A^+$

$$\therefore A^+ = \{A, B, C, D\} = R.$$

$\therefore A \rightarrow ABCD$

Similarly,

$$B \rightarrow ABCD$$

$$C \rightarrow ABCD$$

$$D \rightarrow ABCD$$

Thus the candidate keys are A, B, C, D .

(ii) Since all $\{A, B, C, D\}$ are candidate keys. Thus there is no BCNF violation.

Hence no decomposition is necessary.

(iii) Since A, B, C, D all are candidate keys.

Thus there is no 3NF violation

Hence no decomposition is necessary.

(2) Given $R(A, B, C, D)$ with FDs

$$F = \{B \rightarrow C, B \rightarrow D\}$$

(i) $B^+ = \{B, C, D\}$

Since the closure of $B (B^+)$ does not include all R 's attribute

$$i.e., \quad B^+ = \{B, C, D\} \neq R = \{A, B, C, D\}$$

Hence there are no any candidate key.

(ii) $B \rightarrow C, B \rightarrow D$ both violate BCNF.

because there is no any candidate key.

Thus decomposition is necessary.

Since $AB \rightarrow ABCD$. Thus AB is candidate key.

Therefore one possible BCNF decomposition is

$$\{(A, B), (B, C, D)\}$$

(iii) $B \rightarrow C, B \rightarrow D$ both violate 3NF

Because B is not a super key and CD are not part of a candidate key.

One possible 3NF decomposition is

$$\{(A, B), (B, C, D)\} \text{ or } \{(A, B), (B, C), (B, D)\}$$

Because $AB \rightarrow ABCD$. Thus AB is candidate key.

Q. 9. Are these schema in 3NF ?

(a) $R = \{\text{city, street, zip}\}$

$$F = \{\text{city, street} \rightarrow \text{zip, zip} \rightarrow \text{city}\}$$

(b) $R = (A, B, C)$

$$F = \{A \rightarrow B, B \rightarrow C\}$$

(c) $R = (A, B, C, D)$

$$F = \{B \rightarrow C, B \rightarrow D\}$$

(d) $R = (A, B, C, D)$

$$F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$$

(e) $R = (A, B, C, D)$

$$F = \{AB \rightarrow C, BC \rightarrow D, CD \rightarrow A, AD \rightarrow B\}$$

Ans. (a) $R = \{\text{city, street, zip}\}$

$$F = \{\text{city, street} \rightarrow \text{zip, zip} \rightarrow \text{city}\}$$

For city, street \rightarrow zip

$$\therefore (\text{city, street})^+ = \{\text{city, street, zip}\} \\ = R.$$

Thus, $\{\text{city, street}\}$ is the keys.

Similarly, $\{\text{zip, street}\}$ is also the key.

Therefore, $\{\text{city, street}\}$ and $\{\text{zip, street}\}$ are the keys.

The LHS of city, street \rightarrow zip is a key.

So its OK.

The RHS of zip \rightarrow city is part of a key so its OK.

Therefore, the schema is in 3NF.

(b) $R = (A, B, C)$

$$F = \{A \rightarrow B, B \rightarrow C\}$$

The only key is A . The LHS of $B \rightarrow C$ is not a super key.

The RHS is not part of a key.

Therefore, the schema is not in 3NF.

(c) Given $R = (A, B, C, D)$
 $F = \{B \rightarrow C, B \rightarrow D\}$

The only key is AB .

The LHS of $B \rightarrow C$ is not a super key.

The RHS is not part of a key.

Therefore, the schema is not in 3NF.

(d) Given $R = (A, B, C, D)$
 $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$

The keys are AB, BC and BD .

The LHS of $AB \rightarrow C$ is a key and the RHS of $C \rightarrow D$ is a part of a key. The RHS of $D \rightarrow A$ is part of a key.

Therefore, the schema is in 3NF.

(c) Given $R = (A, B, C, D)$
 $F = \{AB \rightarrow C, BC \rightarrow D, CD \rightarrow A, AD \rightarrow B\}$

The only keys are $\{AB, BC, CD, \text{ and } AD\}$

All the FDs have their LHS as keys.

and RHS have a part of a key.

Therefore the schema is in 3NF.

Q. 10. Give a lossless-join, dependency-preserving decomposition into 3NF of schema R .

$$R = (A, B, C, D, E)$$

$$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

Ans. The schema $R = (A, B, C, D, E)$ is already into 3NF because the candidate key of R are (A, BC, CD, E) .

We may also create a schema from the algorithm.

$$R = \{(A, B, C), (C, D, E), (B, D), (E, A)\}$$

schema (A, B, C) contains a candidate key.

$\therefore R$ is a 3NF dependency-preserving lossless join dependency.

Q. 11. Give a lossless-join dependency preserving decomposition into BCNF of schema R of above question.

Ans. We know that FD $B \rightarrow D$ is nontrivial because $D \subseteq B$, and the LHS is not a superkey. By the algorithm we derive the relation $\{(A, B, C, E), (B, D)\}$ is in BCNF.

Q. 12. Given $R = (A, B, C, D)$
 $F = \{A \rightarrow B, B \rightarrow C\}$

Apply BCNF decomposition.

Ans. (a) Using $A \rightarrow B$ First

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

By $A \rightarrow C$ (which is in F^+) decompose R_2

$$R_3 = (A, C), R_4 = (A, D)$$

The resulting relation schemas are R_1, R_3 and R_4 .

\therefore **Result :** $(A, B), (A, C), (A, D)$

The result is not dependency preserving.

(b) Using $B \rightarrow C$ First

$$R_1 = (B, C), R_2 = (A, B, C)$$

Decompose R_2

$$R_3 = (A, B), R_4 = (A, D)$$

Result : BC, AB, AD

The resulting relation schemas are R_1, R_3, R_4 .

Result : $(B, C), (A, B), (A, D)$

The result is dependency preserving.

Q.13. Compute the closure of the following set F of functional dependencies for relation schema $R = (A, B, C, D, E)$.

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

List the candidate keys for R .

Ans. Starting with $A \rightarrow BC$

We can conclude $A \rightarrow B$ and $A \rightarrow C$

...(i)

$\therefore A \rightarrow B$ and $B \rightarrow D$, then $A \rightarrow D$ (Transitivity)

...(ii)

$\therefore A \rightarrow BC, B \rightarrow D$, then $A \rightarrow CD$.

$\therefore A \rightarrow CD$ and $CD \rightarrow E$, then $A \rightarrow E$ (Transitivity)

...(iii)

$\therefore A \rightarrow A$ we have (Reflexive)

...(iv)

From the above step (i), (ii), (iii), (iv) taking union. We get $A \rightarrow ABCDE$. Thus A is candidate key.

$\therefore E \rightarrow A$ and $A \rightarrow ABCDE$, then
 $E \rightarrow ABCDE$ (Transitivity)

Thus E is a candidate key.

For FD $CD \rightarrow E$

$\therefore CD \rightarrow E$ and $E \rightarrow ABCDE$, then
 $CD \rightarrow ABCDE$ (Transitivity)

So CD is also a candidate key.

$\therefore B \rightarrow D$ and $BC \rightarrow CD$, then
 $BC \rightarrow ABCDE$ (augmentation and transitivity).

So BC is a candidate key.

Therefore the candidate keys are

$$(A, BC, CD, E)$$

Therefore, any FD with A, E, BC or CD on LHS of the arrow is in F^+ , no matter with other attributes appear in the FD.

Allow to represent any set of attributes in R , then F^+ is $BD \rightarrow B, BD \rightarrow D, C \rightarrow C, D \rightarrow D, BD \rightarrow BD, B \rightarrow D, B \rightarrow B, B \rightarrow BD$ and all the FDs of the form

$$A \square \rightarrow \alpha, BC \square \rightarrow \alpha, CD \square \rightarrow \alpha, E \square \rightarrow \alpha$$

where α is any subset of (A, B, C, D, E) .

Q. 14. Consider relation $R = (A, B, C, D, E)$

M is the set of multivalued functional dependencies.

$$M = (A \twoheadrightarrow BC, B \twoheadrightarrow CD, E \twoheadrightarrow AD)$$

Give a lossless join decomposition of schema R into 4NF.

(UPTU 2003, 04)

Ans. Given $A \twoheadrightarrow BC$,

...(i)

$$\therefore A \twoheadrightarrow BC \text{ and } B \twoheadrightarrow CD \quad [\because A \twoheadrightarrow BC \text{ (decomposition) } A \twoheadrightarrow B \text{ and } A \twoheadrightarrow C]$$

$$\therefore A \twoheadrightarrow CD \quad \dots(\text{ii}) \quad \because B \twoheadrightarrow CD$$

by union of (i) and (ii) $A \twoheadrightarrow BCD \quad [\because A \twoheadrightarrow B \text{ and } B \twoheadrightarrow CD \therefore A \twoheadrightarrow CD]$

Given $R(A, B, C, D, E)$

Let the decomposition R into R_1 and R_2

$$R_1 = (A, B, C, D) \text{ (A is key attribute)}$$

$$R_2 = (A, D, E) \text{ (E is key attribute)}$$

For lossless-join decomposition

either $R_1 \cap R_2 \twoheadrightarrow R_1$

or $R_1 \cap R_2 \twoheadrightarrow R_2$

$$\therefore R_1 \cap R_2 = (A, B, C, D) \cap (A, D, E) = \{A, D\}$$

Since $A \twoheadrightarrow BCD$.

$$\therefore R_1 \cap R_2 = (A, B, C, D) \cap (A, D, E) \twoheadrightarrow (A, B, C, D) = R_1$$

Hence, we can say that schema R can be decomposed into (A, B, C, D) and (A, E, D) , which are lossless-join decomposition and are in 4NF.

Q.15. Given $R = \{A, B, C, D, E\}$ and set M of multivalued dependency.

$$M = \{A \twoheadrightarrow BC, B \twoheadrightarrow CD, E \twoheadrightarrow AD\}$$

List all non-trivial MUD in M^+ .

(UPTU 2004)

Ans. $M^+ = \{A \twoheadrightarrow BCD$

$$E \twoheadrightarrow BCD\}$$

For $A \twoheadrightarrow BCD$

given $A \twoheadrightarrow BC \quad \text{(Applying decomposition)}$

$$A \twoheadrightarrow B \quad \dots(\text{i})$$

$$A \twoheadrightarrow C$$

Since $B \twoheadrightarrow CD$

$$\therefore A \twoheadrightarrow B \text{ and } B \twoheadrightarrow CD$$

$$A \twoheadrightarrow CD \quad \text{(transitivity) } \dots(\text{ii})$$

By Union of (i) and (ii) $A \twoheadrightarrow BCD$

Since $E \twoheadrightarrow AD$

Applying decomposition

$$E \twoheadrightarrow A \text{ and } E \twoheadrightarrow D$$

Since $E \twoheadrightarrow A$ and $A \twoheadrightarrow BCD$

$\therefore E \twoheadrightarrow BCD$

(By transitivity)

Q. 16. Explain how FDs can be used to indicate the following :

- A one-to-one relationship set exists between entity set account and customer.
- A many-to-one relationship set exists between entity sets account and customers.

Ans. Let $PK(r)$ denote the primary key attribute of relation r .

- The FDs $PK(\text{account}) \rightarrow PK(\text{customer})$ and $PK(\text{customer}) \rightarrow PK(\text{account})$ indicate a one-to-one relationship,

Because any two tuples with the same value for account must have the same value for customer and any two tuples agreeing on customer must have the same value for account.

- The FDs $PK(\text{account}) \rightarrow PK(\text{customer})$ indicates a many-to-one relationship, because any account value which is repeated will have the same customer value but many account values may have the same customer value.

Q. 17. Consider the following collection of relations and dependencies. Assume that each relation is obtained through decomposition from a relation with attributes ABCDEFGHI and that all the known dependencies over relation ABCDEFGHI are listed for each question. (The questions are independent of each other; obviously, since the given dependencies over ABCDEFGHI are different.) For each (sub) relation : (a) State the strongest normal form that the relation is in. (b) If it is not in BCNF, decompose it into a collection of BCNF relations.

1. $R_1(A, C, B, D, E), A \rightarrow B, C \rightarrow D$
2. $R_2(A, B, F), AC \rightarrow E, B \rightarrow F$
3. $R_3(A, D), D \rightarrow G, G \rightarrow H$
4. $R_4(D, C, H, G), A \rightarrow I, I \rightarrow A$
5. $R_5(A, I, C, E)$

Ans. 1. 1NF. BCNF decomposition: AB, CD, ACE.

2. 1NF. BCNF decomposition: AB, BF

3. BCNF.

4. BCNF.

5. BCNF.

Q. 18. Suppose you are given a relation R with four attributes ABCD. For each of the following sets of FDs, assuming those are the only dependencies that hold for R , do the following: (a) Identify the candidate key (s) for R . (b) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF). (c) If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.

1. $C \rightarrow D, C \rightarrow A, B \rightarrow C$
2. $B \rightarrow C, D \rightarrow A$
3. $ABC \rightarrow D, D \rightarrow A$
4. $A \rightarrow B, BC \rightarrow D, A \rightarrow C$
5. $AB \rightarrow C, AB \rightarrow D, C \rightarrow A, D \rightarrow B$

Ans.

1. (a) Candidate keys: B

(b) R is in 2NF but not 3NF.

(c) $C \rightarrow D$ and $C \rightarrow A$ both cause violations of BCNF. One way to obtain a (lossless) join preserving decomposition is to decompose R into AC , BC , and CD .

2. (a) Candidate keys: BD
 (b) R is in 1NF but not 2NF.
 (c) Both $B \rightarrow C$ and $D \rightarrow A$ cause BCNF violations. The decomposition : AD, BC, BD (obtained by first decomposing to AD, BCD) is BCNF and lossless and join-preserving.
3. (a) Candidate keys : ABC, BCD
 (b) R is in 3NF but not BCNF.
 (c) $ABCD$ is not in BCNF since $D \rightarrow A$ and D is not a key. However, if we split up R as AD, BCD we cannot preserve the dependency $ABC \rightarrow D$. So there is no BCNF decomposition.
4. (a) Candidate keys : A
 (b) R is in 2NF but not 3NF (because of the FD : $BC \rightarrow D$).
 (c) $BC \rightarrow D$ violates BCNF since BC does not contain a key. So we split up R as in: BCD, ABC .
5. (a) Candidate keys : AB, BC, CD, AD
 (b) R is in 3NF but not BCNF (because of the FD : $C \rightarrow A$).
 (c) $C \rightarrow A$ and $D \rightarrow B$ both cause violations. So decompose into: AC, BCD but this does not observe $AB \rightarrow C$ and $AB \rightarrow D$, and BCD still not BCNF because $D \rightarrow B$. So we need to decompose further into: AC, BD, CD . However, when we attempt to revive the lost functional dependencies by adding ABC and ABD , we see that these relations are not in BCNF form. Therefore, there is no BCNF decomposition.

Review Questions

1. What do you mean by functional dependency? Explain with an example and a functional dependency diagram.
2. What is the importance of functional dependencies in database design?
3. What are the main characteristics of functional dependencies?
4. Describe Armstrong's axioms. What are derived rules?
5. Let us assume that the following is given :
 Attribute set $R = ABCDEFGH$
 FD set of $F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow G\}$
 Which of the following decompositions of $R = ABCDEG$, with the same set of dependencies F , is
 - (a) dependency-preserving and
 - (b) lossless-join
 - (a) $\{AB, BC, ABDE, EG\}$
 - (b) $\{ABC, ACDE, ADG\}$
6. What is the lossless or non-additive join property of decomposition? Why is it important?
7. What do you understand by the term normalization? Describe the data normalization process. What does it accomplish?
8. Describe the purpose of normalising data.
9. What are different normal forms?
10. Define 1 NF, 2 NF and 3 NF.

11. Given a relation $R(A, B, C, D, E)$ and $F = (A \rightarrow B, BC \rightarrow D, D \rightarrow BC, DE \rightarrow \phi)$, synthesis a set of 3 NF relation schemes.
12. Define Boyce–Codd normal form (BCNF). How does it differ from 3 NF? Why is it considered a stronger from 3 NF? Provide an example to illustrate.
13. Why is 4 NF preferred to BCNF?
14. A relation $R(A, B, C)$ has FDs $AB \rightarrow C$ and $C \rightarrow A$. Is R is in 3 NF or in BCNF? Justify your answer.
15. Explain the following :
 - (a) Why R_2 is in 2 NF but not 3 NF, where
 $R_2 = (\{A, B, C, D, E\}, \{AB \rightarrow CE, E \rightarrow AB, C \rightarrow D\})$
 - (b) Why R_3 is in 3 NF but not BCNF, where
 $R_3 = (\{A, B, C, D\}, \{A \rightarrow C, D \rightarrow B\})$

