UNIT - III

NITISH SHARMA

Assignment

- Overview of Physical Storage Media,
- File Organization,
- Indexing and Hashing,
- B+ tree Index Files,
- Query Processing Overview,
- Materialized views,
- Database Tuning.

Database

A database is an organized collection of data whose content must be quickly and easily

- Accessed
- Managed
- Updated

A relational database is one whose data are split up into tables, sometimes called **relations**.

3

Normalization

Normalization is a process that improves storage efficiency, data integrity, and scalability of a database design by generating relations that are of higher normal forms.

The **main goal** of Database Normalization is to restructure the logical data model of a database to:

- Reduction of redundant data.
- Ensure data dependencies make sense.
- Reduce the potential for data anomalies.

4

History

Edgar F. Codd first proposed the process of normalization and what came to be known as the **1st normal form** in his paper

"A Relational Model of Data for Large Shared Data Banks"

Edgar F. Codd stated:

"There is, in fact, a very simple elimination procedure which we shall call normalization. Through decomposition non-simple domains are replaced by '*domains whose elements are atomic (non-decomposable) values.'*"

- Edgar F. Codd originally established three normal forms: 1NF, 2NF and 3NF.
- 3NF is widely considered to be sufficient for most applications.

5

Why we need Normalization?

- To reduce the redundant data from database.
- To minimize data loss.
- To reduce the potential of data anomalies.
- To improve storage efficiency, data integrity, and scalability of a database design.

6

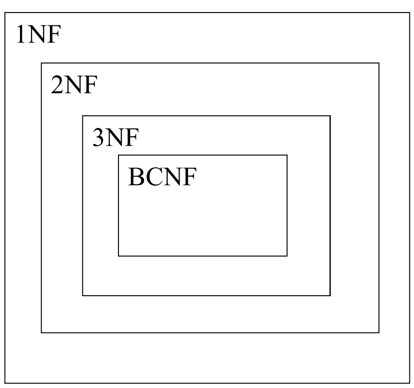
Normalization

Normal Forms are progressive in nature:

1NF is considered the weakest,
2NF is stronger than 1NF,
3NF is stronger than 2NF, and
BCNF is considered the strongest

7

Normalization



a relation in 2NF is also in 1NF

a relation in 3NF is also in 2NF

a relation in BCNF, is also in 3NF

8

Functional Dependency

Consider a relation schema R, and let $\alpha \subseteq R$ and $\beta \subseteq R$.
The functional dependency

$$\alpha \rightarrow \beta$$

holds on schema R if, in any legal relation $r(R)$, for all pairs of tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.

- **Functional dependencies** sometimes are referred to as "equality-generating dependencies".

Example:
Suppose each employee is identified by their unique employee number. We say there is a functional dependency of email address on employee number:

employee number → email address

9

Determinant

Functional Dependency

EmpNum → EmpEmail

Attribute on the LHS is known as the ***determinant***

Attribute on the RHS is known as the ***determiner***

Here, "EmpNum" is a determinant of "EmpEmail"

10

Functional Dependency

<u>EmpNum</u>	EmpEmail	EmpFname	EmpLname
123	al@npiu.com	Alan	Lee
456	ps@npiu.com	Peter	Smith
555	jd@npiu.com	John	Doe
633	zl@npiu.com	Zhnag	Li
787	xf@npiu.com	Xu	Fing

If EmpNum is the PK then the FDs:

EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

must exist.

11

Functional Dependency

EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

FDs are depicted in three different ways

EmpNum → EmpEmail, EmpFname, EmpLname

<u>EmpNum</u>	EmpEmail	EmpFname	EmpLname

12

Introduction to Axioms Rules

- Armstrong's Axioms is a set of rules.
- It provides a simple technique for reasoning about functional dependencies.
- It was developed by William W. Armstrong in 1974.
- It is used to infer all the functional dependencies on a relational database.

13

Various Axioms Rules

A. Primary Rules

Rule 1	**Reflexivity** If A is a set of attributes and B is a subset of A, then A holds B. { A → B }
Rule 2	**Augmentation** If A hold B and C is a set of attributes, then AC holds BC. {AC → BC} It means that attribute in dependencies does not change the basic dependencies.
Rule 3	**Transitivity** If A holds B and B holds C, then A holds C. If {A → B} and {B → C}, then {A → C} A holds B {A → B} means that A functionally determines B.

14

B. Secondary Rules

Rule 1	**Union** If A holds B and A holds C, then A holds BC. If{A → B} and {A → C}, then {A → BC}
Rule 2	**Decomposition** If A holds BC and A holds B, then A holds C. If{A → BC} and {A → B}, then {A → C}
Rule 3	**Pseudo Transitivity** If A holds B and BC holds D, then AC holds D. If{A → B} and {BC → D}, then {AC → D}

15

Example:

Consider relation E = (P, Q, R, S, T, U) having set of Functional Dependencies (FD).

P → Q	P → R
QR → S	Q → T
QR → U	PR → U

Calculate some members of Axioms are as follows,

1. P → T
2. PR → S
3. QR → SU
4. PR → SU

16

Attribute Closure (X^+):-

X^+ = { set of all attributes determined by X }

eg. { $AB \to C$, $C \to D$, $DE \to F$, $BC \to E$; $EG \to H$ }

$(AB)^+$ = { A, B, C , D , E , F }

A, B: as $AB \to AB$ (reflexivity)

C: as $AB \to C$

D: as $C \to D$

E: as $BC \to E$

F: as $DE \to F$

17

(ii) R(ABCDE)

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

By Hit and Trial,

A^+ = { A B C D E } So, A is cand key.

Now

$D \to A$

So, D^+ = { D A B C E } So D is cand key

Now,

$B \to D$

So, B^+ = { B D A C E } So B is cand key

{ A , D , B } are cand keys.

18

Equality of FD Sets:-

Let F and G be two FD sets.

then $\boxed{F \ \& \ G \text{ FD sets are equal iff } F^+ = G^+}$

But it will become very hard to calculate all members of F^+ and G^+, so our problem is not solved.

Another Method:-

F & G FD sets are equal iff

a) F covers G:-

Every FD of G set must be implied in F.

$[F \supseteq G]$

b) G covers F:-

Every FD of F set must be implied in G.

$[F \subseteq G]$

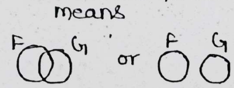
19

---

(4)

F covers G	G Covers F	
✓	✓	$F \equiv G$
✓	×	$F \supset G$
×	✓	$F \subset G$
×	×	F & G Not Comparable

means



but no one is subset of the other

20

Q. $F = \{A \to BCDEF,\ BC \to ADEF,\ B \to F,\ D \to E\}$

$G = \{A \to BC,\ BC \to AD,\ B \to F,\ D \to E\}$

then which is true a) $F \subset G$ b) $F \supset G$ c) $F \equiv G$ d) None

In G,

$A^+ = \{ABCDEF\}$ so, $A \to BCDEF$ is covered

$BC^+ = \{BCADEF\}$ so $BC \to ADEF$ is covered

$B \to F$ & $D \to E$ is covered.

so, G covers F $F \subset G$

In F, $A \to BC$ is covered.

$BC \to AD$ is covered

$B \to F$
$D \to E$ are also covered

So, F covers G.

$F \supset G$

$\because$ $F \subset G$ & $F \supset G$ so, $F \equiv G$.

(c) is correct.

21

Canonical Cover [Minimal Cover]

Minimal set of FDs (F_m) which are logically equal to given FD set F. $\boxed{F_m \equiv F}$

22

eg. $F = \{A \to B, B \to C, A \to C, AB \to C, A \to A\}$

Find minimal cover of F.

sol:-

i) $A \to A$ (Trivial FD)

It should not be included in F.

as if we not have $A \to A$ then also it can't influence FD set F.

or $A \to A$ is obviously derivable from F.

So, remove $A \to A$

ii) $AB \to C$

$B \to C \xRightarrow{\text{Augm.}} AB \to AC \xRightarrow{\text{split}} AB \to C$

$\because AB \to C$ can be determined by $B \to C$

So, remove $AB \to C$

iii) $A \to C$

$A \to B, B \to C \xRightarrow{\text{trans.}} A \to C$

So remove $A \to C$. $\therefore F_m = \{A \to B, B$

23

NOTE:-

Minimal cover of FD set F may not be unique but all minimal covers are logically equal or equal to FD.

i.e.

$(F_{m1} \equiv F_{m2} \equiv F)$

or Expressive Power of F_{m1} = that of F_{m2} = that of F.

24

Functional Dependency

CASE I:

The functional dependency $\alpha \rightarrow \beta$ forms redundancy in R iff

- It is not a trivial F.D. && α is not a superkey.

CASE II:

The functional dependency $\alpha \rightarrow \beta$ doesn't forms redundancy in R iff

- It is a trivial F.D. or α is a superkey.

25

What is decomposition?

- Decomposition is the process of breaking down in parts or elements.
- It replaces a relation with a collection of smaller relations.
- It breaks the table into multiple tables in a database.
- It should always be lossless, because it confirms that the information in the original relation can be accurately reconstructed based on the decomposed relations.
- If there is no proper decomposition of the relation, then it may lead to problems like loss of information.

26

Properties of Decomposition

Following are the properties of Decomposition,

1. Lossless Decomposition
2. Dependency Preservation

27

1. Lossless Decomposition

- Decomposition must be lossless. It means that the information should not get lost from the relation that is decomposed.
- It gives a guarantee that the join will result in the same relation as it was decomposed.

Lossless Join Decomposition

Relation R decomposed into $R_1 R_2 \ldots R_n$

$\boxed{\text{In General, } (R_1 \bowtie R_2 \bowtie R_3 \bowtie \ldots R_n) \supseteq R}$

If $(R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n) = R$

Then Lossless Join Decomposition

If $(R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n) \supset R$

Then lossy join decomposition

28

Example: <Employee_Department> Table

Eid	Ename	Age	City	Salary	Deptid	DeptName
E001	ABC	29	Pune	20000	D001	Finance
E002	PQR	30	Pune	30000	D002	Production
E003	LMN	25	Mumbai	5000	D003	Sales
E004	XYZ	24	Mumbai	4000	D004	Marketing
E005	STU	32	Bangalore	25000	D005	Human Resource

- Decompose the above relation into two relations to check whether a decomposition is lossless or lossy.
- Now, we have decomposed the relation that is Employee and Department.

Relation 1 : <Employee> Table

Eid	Ename	Age	City	Salary
E001	ABC	29	Pune	20000
E002	PQR	30	Pune	30000
E003	LMN	25	Mumbai	5000
E004	XYZ	24	Mumbai	4000
E005	STU	32	Bangalore	25000

- Employee Schema contains (Eid, Ename, Age, City, Salary).

29

Relation 2 : <Department> Table

Deptid	Eid	DeptName
D001	E001	Finance
D002	E002	Production
D003	E003	Sales
D004	E004	Marketing
D005	E005	Human Resource

- Department Schema contains (Deptid, Eid, DeptName).
- So, the above decomposition is a Lossless Join Decomposition, because the two relations contains one common field that is 'Eid' and therefore join is possible.
- Now apply natural join on the decomposed relations.

30

Employee ⋈ Department

Eid	Ename	Age	City	Salary	Deptid	DeptName
E001	ABC	29	Pune	20000	D001	Finance
E002	PQR	30	Pune	30000	D002	Production
E003	LMN	25	Mumbai	5000	D003	Sales
E004	XYZ	24	Mumbai	4000	D004	Marketing
E005	STU	32	Bangalore	25000	D005	Human Resource

Hence, the decomposition is Lossless Join Decomposition.

- If the <Employee> table contains (Eid, Ename, Age, City, Salary) and <Department> table contains (Deptid and DeptName), then it is not possible to join the two tables or relations, because there is no common column between them. And it becomes **Lossy Join Decomposition.**

31

2. Dependency Preservation

- Dependency is an important constraint on the database.
- Every dependency must be satisfied by at least one decomposed table.
- If {A → B} holds, then two sets are functional dependent. And, it becomes more useful for checking the dependency easily if both sets in a same relation.
- This decomposition property can only be done by maintaining the functional dependency.
- In this property, it allows to check the updates without computing the natural join of the database structure.

Relational Schema (R) with FDset F decomposed into $R_1\ R_2 \ldots R_n$ subRelations with FDsets $F_1, F_2 \ldots F_n$ respectively.

In General
$$[F_1 \cup F_2 \cup \ldots \cup F_n] \subseteq F$$

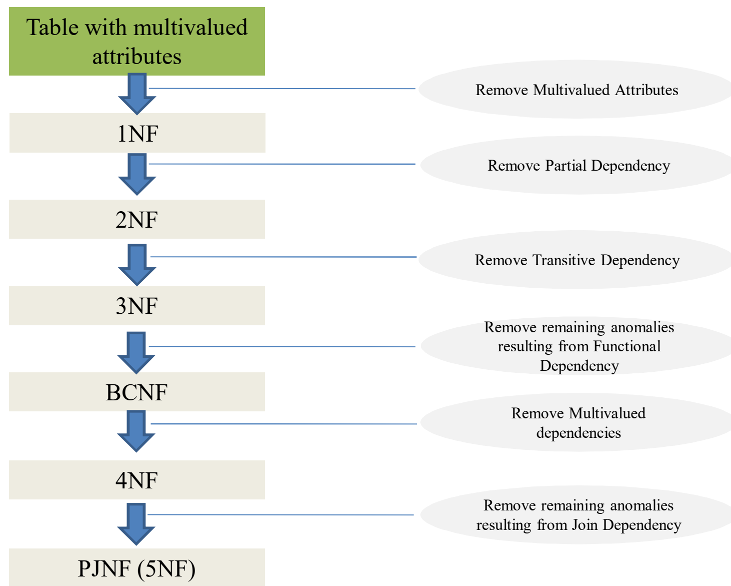
If $[F_1 \cup F_2 \cup \ldots \cup F_n] \equiv F$
then decomposition is Dependency Preserving Decomposition.

No FD is lost becoz of Decomposition

If $[F_1 \cup F_2 \cup \ldots \cup F_n] \subset F$
then it is not Dependency Preserving Decomposition.

32

Normalization



33

First Normal Form

A relation schema R is in **1NF** if the domains of all attributes of R are atomic.

"1NF places restrictions on the structure of relation"

- It does not require additional information such as functional dependencies.

A domain is **atomic** if elements of the domain are considered to be indivisible units.

34

First Normal Form

The following relational table is **not** in 1NF:

EmpNum	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1279	BSc, MSc

EmpDegrees is a multi-valued field:

Employee with EmpNum - 333 has three degrees: *BA, BSc, PhD*

Employee with EmpNum - 679 has two degrees: *BSc* and *MSc*

35

First Normal Form

Employee

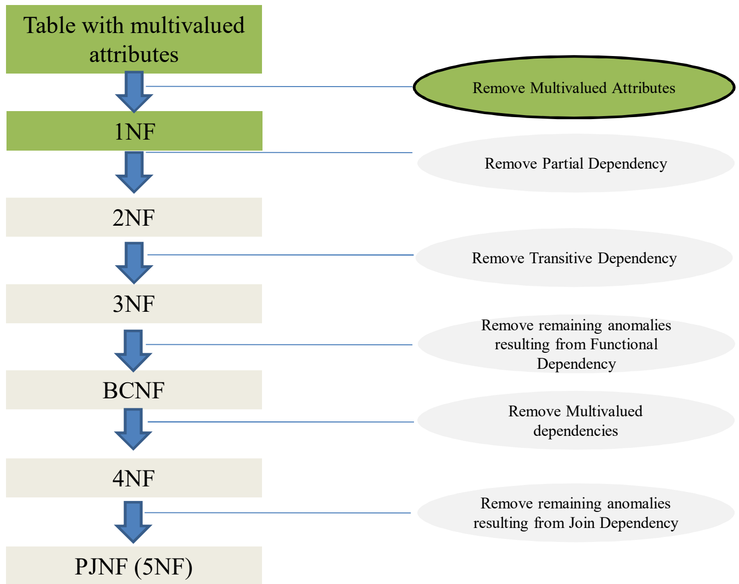
EmpNum	EmpPhone
123	233-9876
333	233-1231
679	233-1231

EmployeeDegree

EmpNum	EmpDegree
333	BA
333	BSc
333	PhD
679	BSc
679	MSc

We can obtain original relation by using Join operation on these relations.

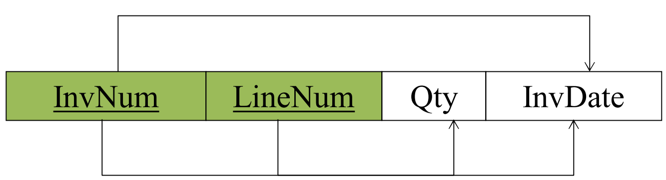
36



37

Partial dependency

A **partial dependency** exists when an attribute B is functionally dependent on an attribute A, and A is a component of a multipart candidate key.



Candidate keys: {InvNum, LineNum}

InvDate is *partially dependent* on {InvNum, LineNum} as InvNum is a determinant of InvDate and InvNum is part of a candidate key

38

Second Normal Form

A relation is in **2NF** if it is in 1NF, and every non-key attribute is fully functional dependent on candidate key.

- It is based on Full Functional dependency.
- 2NF (and 3NF) both involve the concepts of key and non-key attributes.

39

Second Normal Form

Consider this **InvLine** relation (in 1NF)**:**

<u>InvNum</u>	<u>LineNum</u>	ProdNum	Qty	InvDate

InvNum, LineNum $\longrightarrow$ ProdNum, Qty

InvNum $\longrightarrow$ InvDate

Key Attributes: InvNum, LineNum

InvLine is **not in 2NF** due to the presence of partial dependency of InvDate on InvNum

40

Second Normal Form

InvLine

<u>InvNum</u>	<u>LineNum</u>	ProdNum	Qty	InvDate

The above relation has redundancies: the invoice date is repeated on each invoice line.

We can *improve* the database by decomposing the relation into two relations:

<u>InvNum</u>	<u>LineNum</u>	ProdNum	Qty

<u>InvNum</u>	InvDate

41

R(A,B,C)

AB → C
B → C

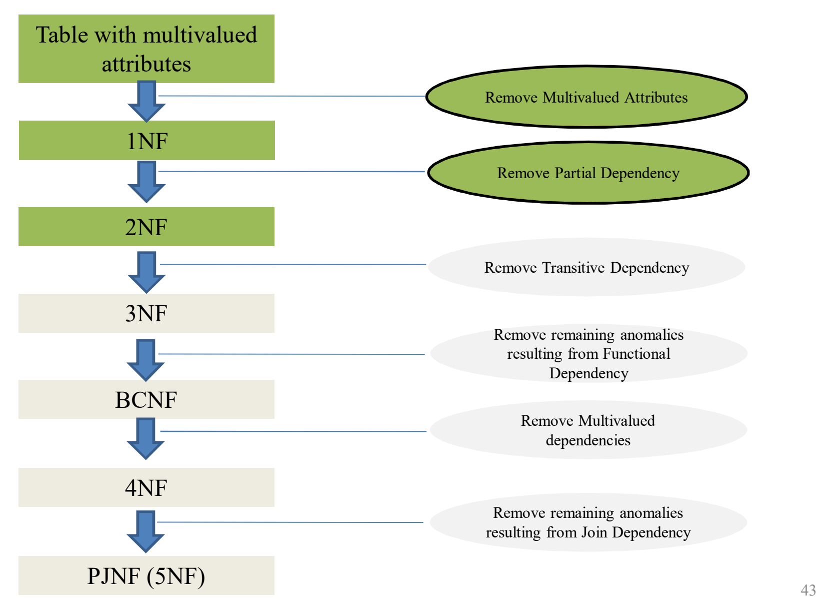
A	B	C
1	U	K
2	U	K
1	P	L
2	P	L

Here, repetition of values for C causes redundancy

A	B
1	U
2	U
1	P
2	P

B	C
U	K
P	L

42



Transitive dependency

Consider attributes A, B, and C, and where

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

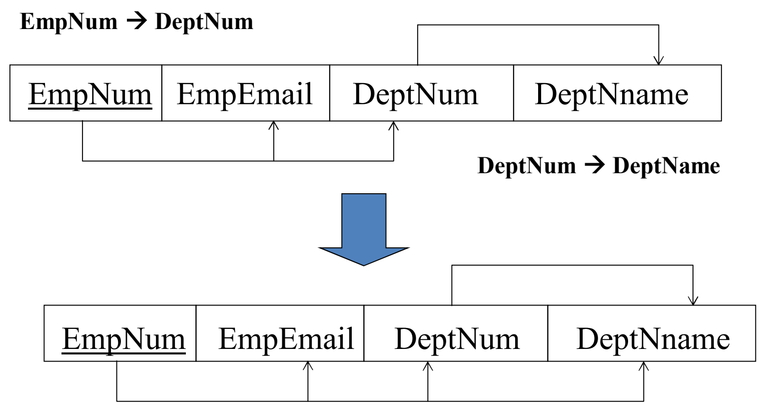
Functional dependencies are transitive, which means that we also have the functional dependency

$$\mathbf{A \rightarrow C}$$

We say that C is transitively dependent on A through B.

44

Transitive dependency



DeptName is *transitively dependent* on EmpNum via DeptNum

EmpNum → DeptName

45

Third Normal Form

A relation schema R is in **3NF** with respect to a set F of functional dependencies if, for all functional dependencies in F^+ of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow \beta$ is trivial functional dependency (that is, $\beta \subseteq \alpha$).
- α is a superkey for schema *R*.
- Each attribute A in $\beta - \alpha$ is contained in a candidate key for *R*.

46

Third Normal Form

<u>EmpNum</u>	EmpName	DeptNum	DeptName

We correct the situation by decomposing the original relation into two 3NF relations. Note the decomposition is *lossless*.

EmpNum	EmpName	DeptNum

DeptNum	DeptName

47

R(A,B,C)

A → B
B → C
A → C

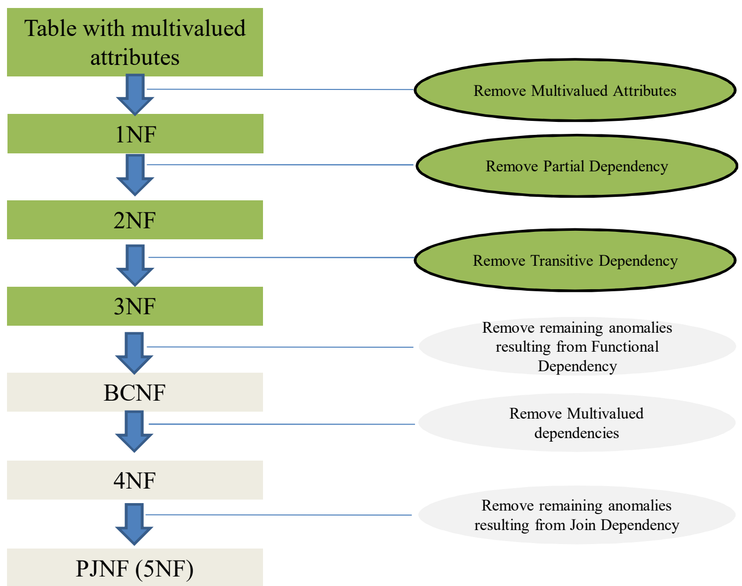
A	B	C
1	U	K
2	U	K
3	P	L
4	P	L

Here, repetition of values for C causes redundancy

A	B
1	U
2	U
3	P
4	P

B	C
U	K
P	L

48



49

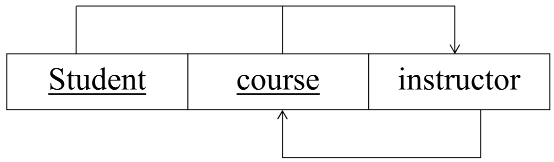
Boyce-Codd Normal Form

A relation schema R is in BCNF with respect to a set F of functional dependencies if, for all functional dependencies in F^+ of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow \beta$ is trivial functional dependency (that is, $\beta \subseteq \alpha$).
- α is a superkey for schema R.

50

Boyce-Codd Normal Form



{Student, course} → Instructor
Instructor → Course

Decomposing into 2 schemas

- {Student,Instructor} {Student,Course}
- {Course,Instructor} {Student,Course}
- {Course,Instructor} {Instructor,Student}

51

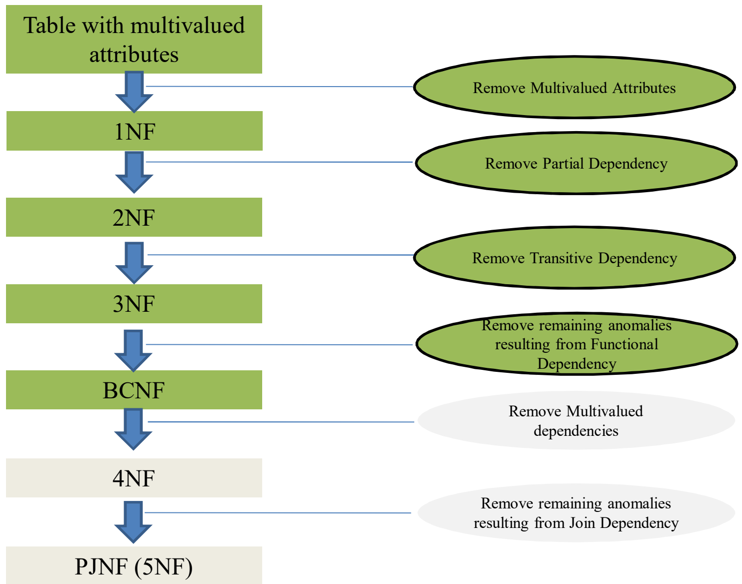
Comparison of BCNF and 3NF

BCNF requires that all non-trivial dependencies be of the form $\alpha \rightarrow \beta$, where α is a superkey.

3NF relaxes this constraint slightly by allowing non-trivial functional dependencies whose determinant is not a superkey.

- It is possible to obtain a 3NF design without sacrificing a lossless join or dependency preservation.
- 3NF allows certain functional dependencies that are not allowed in BCNF.
- Unlike BCNF, 3NF decompositions may contain some redundancy in the decomposed schema.

52



53

Fourth Normal Form (4NF)

A relation schema R is in **4NF** with respect to a set D of functional and multivalued dependencies if, for all multivalued dependencies in D^+ of the form $\alpha \twoheadrightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \twoheadrightarrow \beta$ is trivial multivalued dependency (if $\beta \subseteq \alpha$ or $\beta \cup \alpha = R$).
- α is a superkey for schema R.

- Every 4NF schema is in BCNF.
- Multivalued dependencies sometimes are referred to as "tuple-generating dependencies".

54

Fourth Normal Form (4NF)

Consider this **Movie** relation**:**

MovieName	ScreeningCity	Genre

Candidate Key: {MovieName, ScreeningCity, Genre)

1. All columns are a part of the only candidate key, hence BCNF
2. Many Movies can have the same Genre
3. Many Cities can have the same movie
4. Violates 4NF

MovieName	ScreeningCity	Genre
Hard Code	Los Angles	Comedy
Hard Code	New York	Comedy
Bill Durham	Santa Cruz	Drama
Bill Durham	Durham	Drama
The Code Warrier	New York	Horror

55

Fourth Normal Form (4NF)

We correct the situation by decomposing the original relation into two relations.

1. Move the two multi-valued relations to separate tables
2. Identify a primary key for each of the new entity.

MovieName	ScreeningCity	Genre

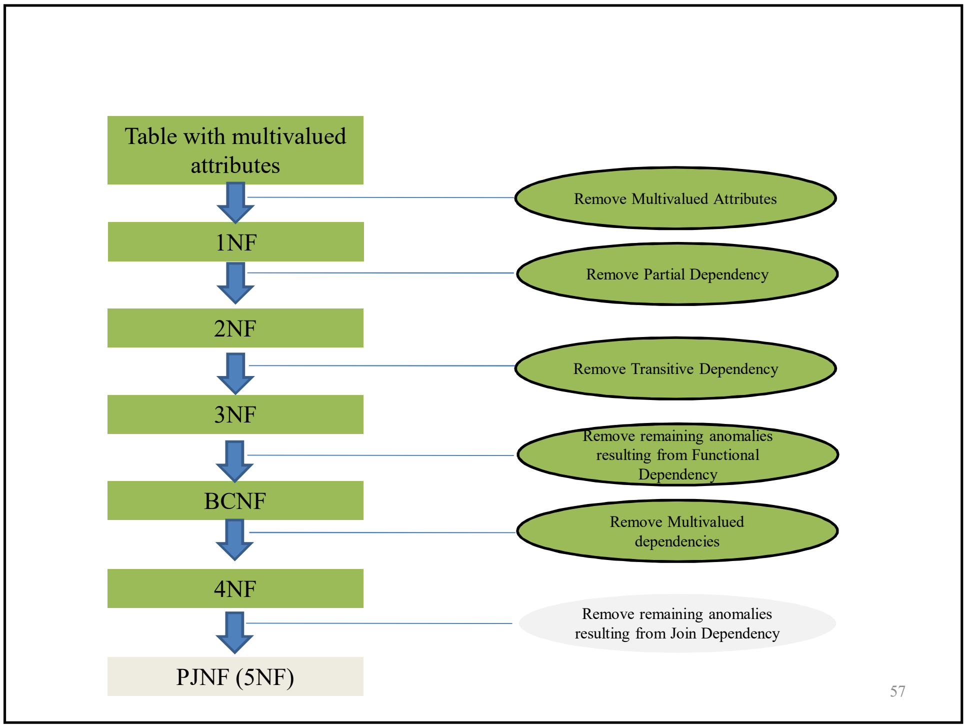
MovieName	ScreeningCity

Movie	ScreeningCity
Hard Code	Los Angles
Hard Code	New York
Bill Durham	Santa Cruz
Bill Durham	Durham
The Code Warrier	New York

MovieName	Genre

Movie	Genre
Hard Code	Comedy
Bill Durham	Drama
The Code Warrier	Horror

56



More Normal Forms

The **4NF** is "ultimate" normal form.

because, multivalued dependency helps us to tackle some forms of repetition of information that cannot be understood in terms of functional dependencies.

There are types of constraints called join dependencies that generalize multivalued dependencies, and lead to another normal form called "**Project-Join Normal Form**"

There is a class of even more general constraints, which leads to a normal form called "**Domain-Key Normal Form**".

58

Analysis of Normal Forms for Redundancy

X→Y Non-trivial F.D. && X not a Superkey (forms redundancy)	**1NF**	**2NF**	**3NF**	**BCNF**
[Proper subset of C.K]→[Non prime attribute]	Allowed	Not allowed	Not allowed	Not allowed
[Non prime attribute]→[Non prime attribute]	Allowed	Allowed	Not allowed	Not allowed
[Proper subset of C.K && Non prime attribute]→[Non prime attribute]	Allowed	Allowed	Not allowed	Not allowed
[Proper subset of C.K]→[Proper subset of other C.K]	Allowed	Allowed	Allowed	Not allowed

59

Database Design Goals

Goals	**1NF**	**2NF**	**3NF**	**BCNF**	**4NF**
0% Redundancy	No	No	No	Yes: over F.D's No: over M.D's	Yes: over F.D's & M.D's
Lossless join decomposition satisfy	Yes	Yes	Yes	Yes	May not
Dependency Preservation satisfy	Yes	Yes	Yes	May not	May not

60

Q. $R(ABCD)$ $F = \{AB \to C \quad C \to A \quad AC \to D\}$
which is the Highest NF of R?

Sol:-
Candidate keys : $\{\underline{AB}, \underline{BC}\}$

Testing for BCNF (atleast one FD with determinant not SK)

SK $AB \to C$ ✓
$C \to A$ ×
not SK $AC \to D$ × } so not BCNF

Testing for 3NF

SK $AB \to C$ Prime ✓
$C \to A$ Prime ✓
$AC \to D$ Not Prime × } so not 3NF
Not SK

Testing for 2NF

Here we take all proper subset of cand. keys and find their closure & check whether they (closure) contain all prime attributes or not.

Cand key $\to$ $AB \Rightarrow$ $A^+ = A$ Prime ✓
$B^+ = B$

$BC \to$ $C^+ = CAD$ Not Prime ×
$C \to D$ will be a partial dependency in R.
$\therefore$ not in 2NF

Hence highest NF of R : 1NF

61

Q. $R(ABCD)$
$\{AB \to C, \; BC \to D\}$

Cand keys : $\{AB\}$

BCNF :
$AB \to C$ ✓
Not SK $BC \to D$ × So not in BCNF

3NF :
$BC \to D$ Not Prime so not in 3NF

2NF :
$AB \to$ $A^+ = A$ ✓ Prime
$B^+ = B$ ✓ So, 2NF

62

Q R(ABCDE)

{A → BC, CD → E, B → D, E → A}

Cand Key: {A, E, CD, BC}

BCNF:

A → BC

CD → E

not sk B → D   so not in BCNF

E → A

3NF:

B → D Prime

A → BC

CD → E Prime

E → A   So 3NF

63