Database Technologies

Session3

Contents

- Data Redundancy
- Data Anomalies
- Database Constraints (Unique, Not Null, Foreign Key, Default, Check*)
- Functional Dependency
- Normalization
- Need for Normalization
- Normal Forms (1st NF, 2nd NF, 3rd NF, BCNF) with examples, Introduction to 4th and 5th NF
- DML (INSERT/UPDATE/DELETE)

Normalization

- Normalization is a data analysis technique to design a database system.
  - An analytical technique used during logical database design
  - Offers a strategy for constructing relations and identifying keys
- Normalization is a technique for producing relational schema with the following properties:
  - No Information Redundancy
  - No Update Anomalies

Why Normalize - 1

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	roy	5
7499	ray	20-Feb-81	prch	purchase	roy	5
7521	jain	02-Apr-82	prch	purchase	roy	5
7654	gupta	28-Sep-79	info	infoserv	rao	6.5

redundancy

Why Normalize - 2

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	roy	5
7499	ray	20-Feb-81	prch	purchase	roy	5
7521	jain	02-Apr-82	prch	purchase	roy	5
7654	gupta	28-Sep-79	info	infoserv	rao	6.5

deletion anomaly

attributes are lost because of the deletion of other attributes

Why Normalize - 2

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	roy	5
7499	ray	20-Feb-81	prch	purchase	roy	5
7521	jain	02-Apr-82	prch	purchase	roy	5

deletion anomaly

Why Normalize - 3

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	roy	5
7499	ray	20-Feb-81	prch	purchase	roy	5
7521	jain	02-Apr-82	prch	purchase	roy	5
7654	gupta	28-Sep-79	info	infoserv	rao	6.5

Update Anomaly exists when one or more instances of duplicated data is updated, but not all.

update anomaly

Why Normalize - 3

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	apte	5
7499	ray	20-Feb-81	prch	purchase	apte	5
7521	jain	02-Apr-82	prch	purchase	apte	5
7654	gupta	28-Sep-79	info	infoserv	rao	6.5

update anomaly

Why Normalize - 4

emp code	emp name	join date	dept code	dept name	dept mngr	dept bdgt
7369	shah	17-Dec-80	prch	purchase	roy	5
7499	ray	20-Feb-81	prch	purchase	roy	5
7521	jain	02-Apr-82	prch	purchase	roy	5
			info	infoserv	rao	6.5

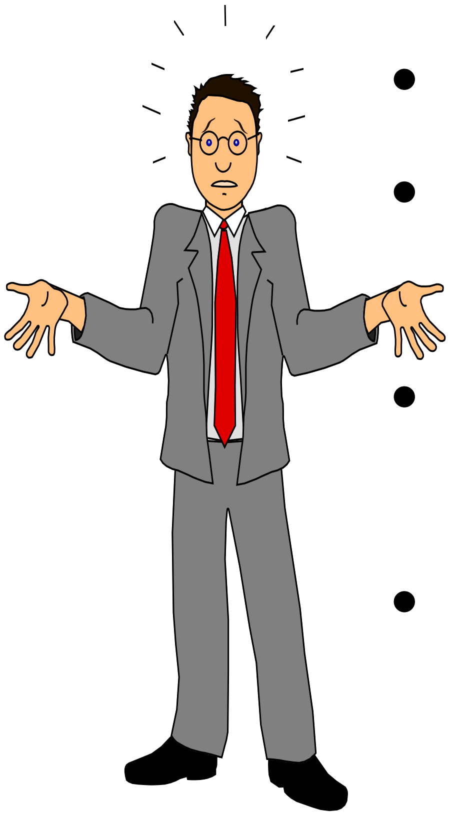
Insert Anomaly occurs when certain attributes cannot be inserted into the database without the presence of other attributes

insertion anomaly

Normalisation Stages

- Process involves applying a series of tests on a relation to determine whether it satisfies or violates the requirements of a given normal form.
  - When a test fails, the relation is decomposed into simpler relations that individually meet the normalization tests.
  - The higher the normal form the less vulnerable to update anomalies

Normal Form??



- “restriction” on a relation
- a relation that satisfies certain rules/conditions
- a relation that exhibits certain properties
- depending on the conditions it satisfies/the properties it exhibits, the relation is said to be in the “n^{th} Normal Form”

Normal Forms

- 1 NF
- 2 NF
- 3 NF
- BCNF

Functional Dependency

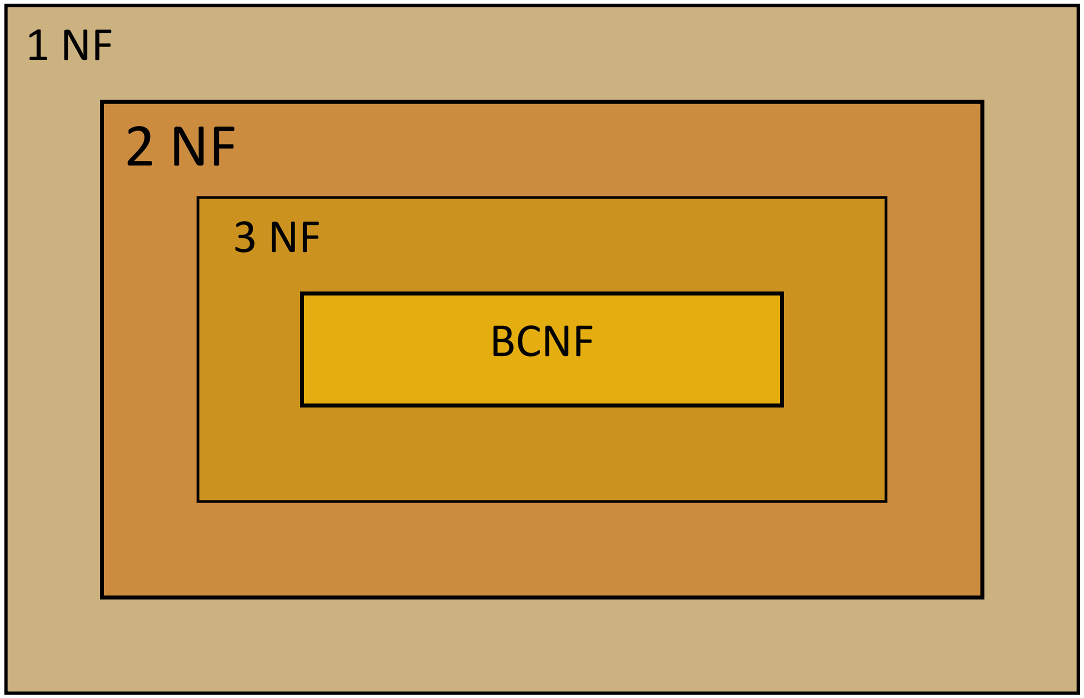
- 4NF

Multi-valued Dependency

- 5NF

Join Dependency

Hierarchy of Normal Forms



Normal Forms are **INCREMENTAL**

Hierarchy of Normal Forms

- It must be emphasised here, that the definition of a Normal Form is INCREMENTAL.
- You cannot have some relation that is in X-normal form, but not in (X-1) Normal Form.
- If a relation is in 3 NF, it also has to be in 2NF, which means it is also in 1NF.

Integrity Constraints

- Databases are structured stores of data
- Data must be accurate
  - Semantic accuracy v/s Syntactic accuracy
    - consider an attribute `age` of type `int`. Any integer would be **syntactically correct**; but if the attribute pertains to the age of say, drivers, then any value less than 18 would be **semantically incorrect**.
- Integrity constraints are the business rules of the problem-domain.

Domain Constraints

- Integrity constraints guard against accidental damage to the database, by ensuring that authorized changes to the database do not result in a loss of data consistency.
- Domain constraints are the most elementary form of integrity constraint.

Referential Integrity

- Ensures that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.
- Formal Definition
  - Let $r_1(R_1)$ and $r_2(R_2)$ be relations with primary keys K_1 and K_2 respectively.
  - The subset α of R_2 is a foreign key referencing K_1 in relation r_1, if for every t_2 in r_2 there must be a tuple t_1 in r_1
    - such that $t_1[K_1] = t_2[\alpha]$.
    - Referential integrity constraint: $\Pi_{\alpha}(r_2) \subseteq \Pi_{K_1}(r_1)$

more on Integrity Constraints

- 3 types of Integrity Constraints are of interest:
  - Functional Dependencies (FD)
  - Multi-Valued Dependency (MVD)
  - Join-Dependency (JD)
- FDs are the most commonly used integrity constraints in normalization.

Functional Dependencies

- Defines a constraint between two (sets of) attributes of a relation
- Require that the value for a certain set of attributes determines uniquely the value for another set of attributes.
- Written as "$X \rightarrow Y$"* (read "X determines Y")

* here, "X" is called the "determinant" and "Y", the "consequent"

FD: Example

- Consider a "Student" relation with the schema {`S_ID`, `Name`, `DateOfBirth`, `Address`, `City`}
- Also consider the "ExamResults" relation with the schema {`S_ID`, `ExamID`, `MaxMarks`, `ObtMarks`}
- Typical FDs that hold over these schemas:
  - {`S_ID`} → {`Name`, `DateOfBirth`, `Address`, `City`}
  - {`S_ID`, `ExamID`} → {`ObtMarks`}
  - {`ExamID`} → {`MaxMarks`}

FD: Defined

- Let R be a relation schema
  - $\alpha \subseteq R$, $\beta \subseteq R$
- The functional dependency
  - $\alpha \rightarrow \beta$

  holds on R if and only if for any legal relations r (R), whenever any two tuples t 1 and t 2 of r agree on the attributes α, they also agree on the attributes β.

  That is,

  $t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$
- K is a superkey for relation schema R if and only if $K \rightarrow R$
- K is a candidate key for R if and only if
  - $K \rightarrow R$, and
  - for no $\alpha \subset K$, $\alpha \rightarrow R$

Trivial and nontrivial dependency

- Trivial
  - If an FD X $\rightarrow$ Y holds where Y subset of X, then it is called a trivial FD.
  - If the right hand side (dependent) is subset of left hand side (determinant)
- Non-trivial
  - If an FD X $\rightarrow$ Y holds where Y is not subset of X, then it is called non-trivial FD.

Axioms of Functional Dependency - I

- Given a set F set of functional dependencies, there are certain there functional dependencies that are logically implied by F .
- The set of all functional dependencies logically implied by F is the closure of F (F +)
- **Axiom of Reflexivity**

  If $Y \subseteq X$, then $X \rightarrow Y$
- **Axiom of Augmentation**

  If $\mathbf{X} \rightarrow \mathbf{Y}$ and $\mathbf{W} \rightarrow \mathbf{Z}$, then $\mathbf{XW} \rightarrow \mathbf{YZ}$
- **Axiom of Transitivity**

  If $\mathbf{X} \rightarrow \mathbf{Y}$ and $\mathbf{Y} \rightarrow \mathbf{Z}$, then $\mathbf{X} \rightarrow \mathbf{Z}$

Axioms of Functional Dependency - II

- **Union Rule**

  If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$

- **Decomposition Rule**

  If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

Attribute Set Closure

- Closure of an attribute set **α**, denoted by α^+ , are:

  - the set of attributes that are determined by **α**

  OR

  - the set of attributes that are functionally dependent on **α**
- Attribute set closure helps in identifying the super keys in a relation

Attribute Set Closure - Example

Given: Relation R(A, B, C, D)

FDs { $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$ }

Here,

- $A^+ = \{A, B, C, D\}$
- $B^+ = \{B, C, D\}$
- $(AB)^+ = \{A, B, C, D\}$
- $(CD)^+ = \{C, D\}$
- $D^+ = \{D\}$

Super keys:

A, AB, ABC, ABCD

Minimal Super key

is candidate key

Candidate key: A

Decomposition

- Most common method of normalization; involves the splitting of original schema into subset schemas.
- Decomposition is not arbitrary; subset schemas (and remainder if any, of original schema) must conform to certain rules
- Decide whether a particular relation R is in “good” form.
- In the case that a relation R is not in “good” form, decompose it into a set of relations of R1,R2, ..., Rn such that
  - each relation is in good form
  - the decomposition is a lossless-join decomposition

Decomposition : *Desirable Properties*

- Lossless-Join
  - the join of the subset schemas when computed, must yield exactly the original relation - not a tuple more, not a tuple less.
- Dependency Preservation
  - decomposition should be such that none of the functional dependencies are lost

for more info:

An Introduction to Database Systems, C J Date, Ch. 10, 11

Lossless-Join Decomposition

Let `R{A,B,C}` be a relation.

`If` `R` is equal to the join of the projections `R1{A,B}` and `R2{A,C}`, **`then this decomposition is Lossless`**.

Lossless-Join Decomposition

Let **R** be a relation

Let **R** be decomposed into two relations

R1 and **R2** .

If $[\mathbf{R1} \cap \mathbf{R2}]^+$ = {all attributes of **R1**} or

{all attributes of **R2**},

then the decomposition is **<u>Lossless</u>**.

Lossless-Join Decomposition - Example

Consider the relation **R(A,B,C,D)** and FDs $\{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$. Consider the decomposition of **R** into **R1(A,B,C)** and **R2(B,C,D)**. Is this decomposition lossless in nature?

- Here $R1 \cap R2 = \{B, C\}$
- $(BC)^+$ $= \{B, C, D\}$ $= R2$
- Hence this decomposition is lossless.

Example

S	STATUS	CITY
S3	30	Paris
S5	30	Athens

S	Status
S3	30
S5	30

S	CITY
S3	Paris
s5	Athens

Lossless

S	STATUS
S3	30
s5	30

STATUS	CITY
30	Paris
30	Athens

Lossy

Which supplier has which city

Lossless-Join Decomposition

Consider the relation
INFO {ID,City,Ppln}
with the FDs
ID → City
City → Ppln

Possible decompositions:

- { {ID,City}, {ID,Ppln} }
- { {ID,City}, {City,Ppln} }
- { {ID,Ppln}, {City,Ppln} }

INFO

ID	City	Ppln
41	Bangalore	5,00,000
42	Trichy	2,50,000
43	Coimbatore	5,00,000
44	Mumbai	8,00,000
45	Mumbai	8,00,000
46	Bangalore	5,00,000
47	Bangalore	5,00,000

Decomposing "INFO"

Decomposed into { {**ID**,**Ppln**}, {**City**,**Ppln**} }

INFO_Ppn

ID	Ppln
41	5,00,000
42	2,50,000
43	5,00,000
44	8,00,000
45	8,00,000
46	5,00,000
47	5,00,000

City_Ppn

City	Ppln
Bangalore	5,00,000
Trichy	2,50,000
Coimbatore	5,00,000
Mumbai	8,00,000

“Recomposing INFO”

INFO = INFO_Ppn $\bowtie$ City_Ppn

INFO_Ppn

ID	City	Ppln
41	Bangalore	5,00,000
41	**Coimbatore**	**5,00,000**
42	Trichy	2,50,000
43	Coimbatore	5,00,000
43	**Bangalore**	**5,00,000**
44	Mumbai	8,00,000
45	Mumbai	8,00,000

ID	City	Ppln
46	Bangalore	5,00,000
46	**Coimbatore**	**5,00,000**
47	Bangalore	5,00,000
47	**Coimbatore**	**5,00,000**

{{ID,Ppln}, {City,Ppln}} was a lossy decomposition

Unnormalized

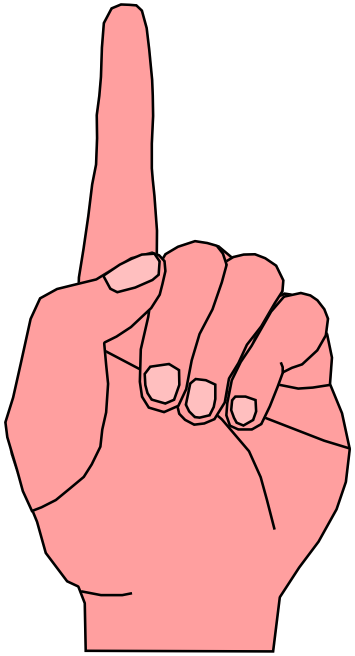
emp_data

emp code	emp name	grade code	grade stage	desig code	change date	grade descr	desig descr
7369	shah	12 --- ---	3 --- ---	slasst --- ---	20-Feb-82 --- ---	Asst_A --- ---	Sls Asst --- ---
7499	ray	23 --- ---	2 --- ---	supr --- ---	17-Jan-95 --- ---	Offcr A --- ---	Superint. ---

multi-valued attributes

not in 1NF!

Unnormalized → 1NF



- Eliminate variable repeating fields and groups so that all attributes take atomic values

A relation is said to be in "first normal form" (1NF) if and only if all its attributes assume only atomic(indivisible) values.

Unnormalized → 1NF

emp_data

emp code	emp name	grade code	grade stage	desig code	change date	grade descr	desig descr
7369	shah	12 --- ---	3 --- ---	slasst --- ---	20-Feb-82 --- ---	Asst_A --- ---	Sls Asst --- ---
7499	ray	23 --- ---	2 --- ---	supr --- ---	17-Jan-95 --- ---	Offcr A --- ---	Superint. --- ---

Unnormalized → 1NF

emp

emp code	emp name
7369	shah
7499	[illegible]

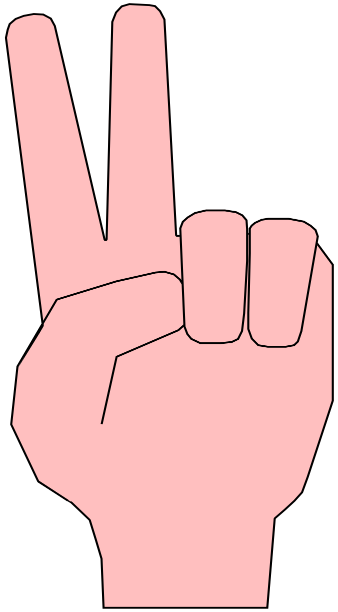
in 1NF!

emp_hist

emp code	grade code	grade stage	desig code	change date	grade descr	desig descr
7369	12	3	slas	20-Feb-82	Asst_A	Sls_Asst
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Offcr A	Superint.
7499	---	---	---	---	---	---
[illegible]	---	---	---	---	---	---
[illegible]	---	---	---	---	---	---

in 1NF!

1NF → 2NF



Eliminate fields that are facts about only a ***subset*** of the key so that all non-key domains are fully functionally dependent on the primary key.

A relation is said to be in 2NF if and only if it is in 1NF and every non-key attribute is **fully functionally dependent (No partial dependency)** on any key.

1NF

- ☑ emp
- ☑ emp_hist

emp_hist

?

emp code	grade code	grade stage	desig code	change date	grade descr	desig descr
7369	12	3	slas	20-Feb-82	Asst_A	Sls_Asst
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Offcr A	Superint.
7499	---	---	---	---	---	---
7499	---	---	---	---	---	---
7499	---	---	---	---	---	---

`Grade_descr` is ***not fully functionally dependent*** on the primary key.

not in 2NF!

Eliminating partial dependencies

1NF

- ☑ emp
- ☑ emp_hist

emp_hist

emp code	grade code	grade stage	desig code	change date	grade descr	desig descr
7369	12	3	slas	20-Feb-82	Asst_A	Sls_Asst
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7369	---	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Offcr A	Superint.
7499	---	---	---	---	---	---
7499	---	---	---	---	---	---
7499	---	---	---	---	---	---

Eliminating partial dependencies

1NF

- ☑ emp
- ☑ emp_hist

emp_hist

emp code	grade code	grade stage	desig code	change date
7369	12	3	slas	20-Feb-82
7369	---	---	---	---
7369	---	---	---	---
7369	---	---	---	---
7369	---	---	---	---
7499	23	2	supr	17-Jan-95
7499	---	---	---	---
7499	---	---	---	---
7499	---	---	---	---

desig descr
Sls_Asst

Superint.

1NF

- ☑ emp
- ☑ emp_hist

grade

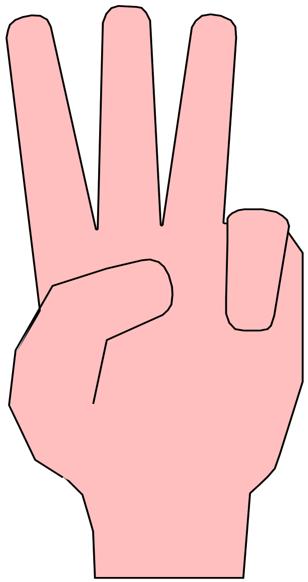
grade code	grade descr
12	Asst_A
23	Offcr A

in 2NF

emp_hist

emp code	grade code	grade stage	desig code	change date	desig descr
7369	12	3	slas	20-Feb-82	Sls_Asst
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Superint.
7499	---	---	---	---	---
7499	---	---	---	---	---
7499	---	---	---	---	---

2NF → 3NF



Eliminate non-key fields that are facts about other non-key fields, so that all non-key domains are mutually independent.

A relation is said to be in 3NF if and only if it is in 2NF and for every nonkey attribute is non transitively dependent on primary key.

2NF

- ☑ emp
- ☑ emp_hist
- ☑ grade

`desig_descr` is ***transitively dependent*** on the primary key.

not in 3NF

emp_hist

emp code	grade code	grade stage	desig code	change date	desig descr
7369	12	3	slas	20-Feb-82	Sls_Asst
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Superint.
7499	---	---	---	---	---
7499	---	---	---	---	---
7499	---	---	---	---	---

Eliminating Transitive Dependencies

2NF

- [x] emp
- [x] emp_hist
- [x] grade

emp_hist

emp code	grade code	grade stage	desig code	change date	desig descr
7369	12	3	slas	20-Feb-82	Sls_Asst
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7369	---	---	---	---	---
7499	23	2	supr	17-Jan-95	Superint.
7499	---	---	---	---	---
7499	---	---	---	---	---
7499	---	---	---	---	---

Eliminating Transitive Dependencies

2NF

- ☑ emp
- ☑ emp_hist
- ☑ grade

in 3NF

desig

desig code	desig descr
slas	Sls_Asst
supr	Superint.

emp_hist

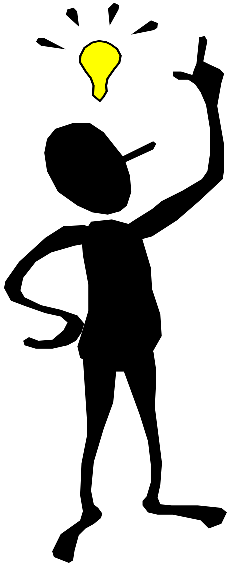
emp code	grade code	grade stage	desig code	change date
7369	12	3	slas	20-Feb-82
7369	---	---	---	---
7369	---	---	---	---
7369	---	---	---	---
7369	---	---	---	---
7499	23	2	supr	17-Jan-95
7499	---	---	---	---
7499	---	---	---	---
7499	---	---	---	---

3NF → BCNF*

- 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 (i.e., $\beta \subseteq \alpha$)
  - α is a superkey for R

*BCNF = Boyce Codd Normal Form

3NF → BCNF*



Eliminate key fields that are facts about other (key) fields so that every determinant becomes a superkey.

*BCNF = Boyce Codd Normal Form

A relation is said to be in BCNF if and only if, for every nontrivial FD $A \rightarrow B$, 'A' is a superkey.

Problem-Domain (BCNF)

Consider a sales-management scenario:

- There are several product categories.
- Products are sold in several cities.
- Each city has several agents.
- Each product category is sold in each city by several retail outlets.
- A given product category is distributed in a given city by one and only one agent.
- A given agent will operate in one and only one city.
- A given agent can stock & sell more than one product category for the same city.

Problem-domain (BCNF)

An appropriate schema would be

Supply {<u>**Product**</u>, <u>**City**</u>, **Outlets**, **Agent**}

Supply

Product	City	Outlets	Agent
Noodles	Margao	155	Daulat
Chocolates	Margao	110	Keshav
Baby Foods	Margao	235	Daulat
Ketchup	Panjim	163	Magsons
Noodles	Panjim	195	Aletta
Chocolates	Vasco	102	Pranav

<u>FDs</u>:

$\{P, C\} \rightarrow \{O, A\}$

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

$\boxed{\{A, P\} \rightarrow \{C\}}$

not in BCNF

3NF → BCNF

Supply

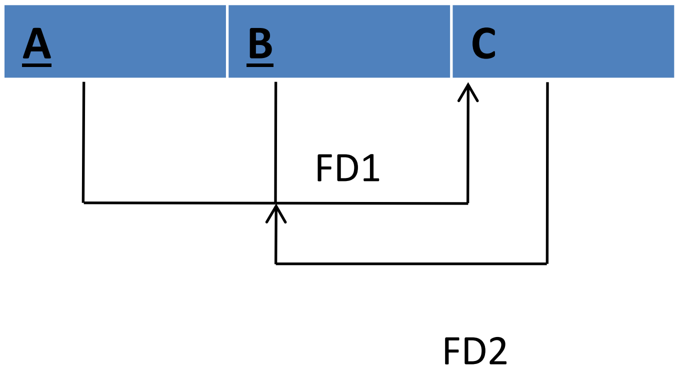
Product	City	Outlets
Noodles	Margao	155
Chocolates	Margao	110
Baby Foods	Margao	235
Ketchup	Panjim	163
Noodles	Panjim	195
Chocolates	Vasco	102

Agent	City
Daulat	Margao
Keshav	Margao
Magsons	Panjim
Aletta	Panjim
Pranav	Vasco

A_Product

Agent	Product
Daulat	Noodles
Daulat	Baby Foods
Keshav	Chocolates
Magsons	Ketchup
Aletta	Noodles
Pranav	Chocolates

BCNF



- Schematic relation in 3NF but not in BCNF

B being prime attribute (member of some candidate key)

Beyond BCNF...

- CTX is in BCNF, it has no functional dependency since it is an all key relation
- a course is taught by multiple teachers
- the course uses multiple textbooks
- there is redundancy

Schema CTX

CTX

course	teachers	texts
Phy	Green	T1
Phy	Green	T2
Phy	Brown	T1
Phy	Brown	T2
Math	Green	T1
Math	Green	T2

More redundancies …

COURSE	TEACHERS	TEXTS
Physics	{Prof. Green, Prof. Brown}	{Basic Mechanics, Principles of Optics}
Math	{Prof. White}	{Basic Mechanics, Vector Analysis, Trigonometry}

Note: There are no FDs in this relation.

More redundancies …

CTX

COURSE	**TEACHERS**	**TEXTS**
Physics	Prof. Green	Basic Mechanics
Physics	Prof. Green	Principles of Optics
Physics	Prof. Brown	Basic Mechanics
Physics	Prof. Brown	Principles of Optics
Math	Prof. White	Basic Mechanics
Math	Prof. White	Vector Analysis
Math	Prof. White	Trigonometry

<u>Constraint:</u> IF tuples (c, t1, x1), (c, t2, x2) both appear

THEN tuples (c, t1, x2), (c, t2, x1) both appear also

- Redundancy :
  - TO add information that a physics course can be taught by a new teacher
    - Insert two new tuples

Multi-valued Dependency

IF tuples (c, t1, x1), (c, t2, x2) both appear

THEN tuples (c, t1, x2), (c, t2, x1) both appear also

`Course` $\twoheadrightarrow$ **`teachers`**

`Course` $\twoheadrightarrow$ **`texts`**

They are <u>independent</u> MVDs

Multi-valued Dependency

- Defines a constraint between two (sets of) attributes of a relation
- Constraint in turn, defined by the semantics of the problem-domain
- Helps identify redundancy that cannot be identified by mere FD analysis
- MVDs are generalization of FDs
  - Every FD is MVD but converse is not true

- Written as “$X \twoheadrightarrow Y$” (read “X multidetermines Y”)

- For a relation R (A,B,C) the MVD
  - A $\twoheadrightarrow$ B holds if and only if MVD A $\twoheadrightarrow$ C also holds

MVD: Defined

given a relation schema R, and two attribute sets X, Y such that

$X \subseteq R$, $Y \subseteq R$,

then, the MVD “$X \rightarrow Y$” means

$\forall$t1 ,t2, t3, t4 $\in$ r (‘r’ is an instance of ‘R’)

the following is true

t1 [X] = t2 [X] = t3 [X] = t4 [X]

t1 [Y] = t3 [Y]

t3 [R-Y] = t2 [R-Y]

t4 [Y] = t2 [Y]

t4 [R-Y] = t1 [R-Y]

BCNF $\Rightarrow$ 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 hold:
  - $\alpha \twoheadrightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$)
  - α is a superkey for schema R
- If a relation is in 4NF it is in BCNF

BCNF → 4NF

CTX

course	teacher	text
Phy	Green	T1
Phy	Green	T2
Phy	Brown	T2
Phy	Brown	T1
Math	white	T3
Math	White	T4

CT

course	teacher
Phy	Green
Phy	Brown
Math	White

CX

course	text
Phy	T1
Phy	T2
Math	T3
Math	T4

In 4NF!

Eliminating **independent**, multi-valued facts

- Not in 4NF because the following non trivial multivalued dependency holds on them and Course is not a superkey

$$\texttt{Course} \twoheadrightarrow \texttt{texts}$$

$$\texttt{Course} \twoheadrightarrow \texttt{teachers}$$

- After decomposition the two relations are in 4NF because the MVD are trivial MVD(i.e their union produces the original relation)

Normalization : a recap

- leads to a design that caters to ad-hoc queries.
- prevents update anomalies and data inconsistencies.
- facilitates data independence.
- penalises data retrieval.
- rule-based, but intricately influenced by semantics of the attributes.

Normal Forms : a recap

1NF

Eliminate repeating groups; attributes must have only atomic values.

2NF

Eliminate partial dependencies; all non-key attributes must be functionally dependent on ***entire*** primary key, not on a subset thereof.

3NF

Eliminate transitive dependencies; all non-key domains must be mutually independent.

Normal Forms : a recap

BCNF

Eliminate FDs with prime attributes in the consequent; all determinants must be super keys.

4NF

Eliminate independent, multi-valued attributes.

References - I

An Introduction to DB Systems (7/e)

C. J. Date

Addison-Wesley (Pearson-Asia Education)

Database System Concepts (3/e)

Silberschatz, Korth, Sudarshan

McGraw-Hill International Edition

Database Management Systems

Raghu Ramakrishnan

McGraw-Hill International Edition

Fundamental of Database Systems

Elmasri

Navathe