Normalization

- **Normalization** is a step-by-step convertible process of replacing a given collection of relations by successive collections in which the relations have a progressively simple and regular structures.

- Normalization deals with important issues of
  - Database semantics
  - Functional dependence
  - Logical database design

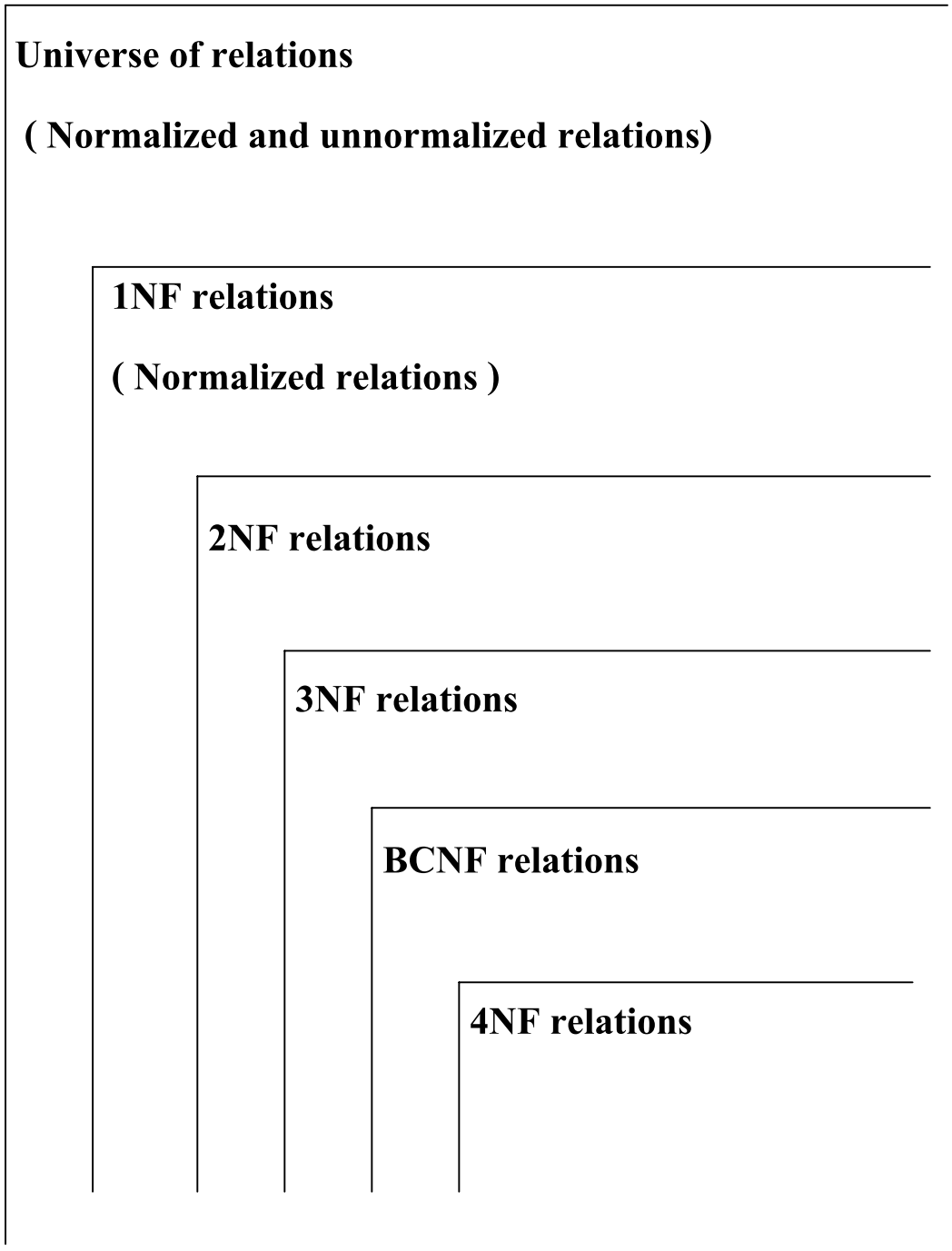
- Normalization theory is built around the concept of **normal forms**.

- A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints.

- Every normalized relation must be in **first normal form (1NF)**.

Normalization

♠ **Normal forms of relations :**



Functional dependence

- **Definition 1**

  Given a relation R, attribute B of R is **functionally dependent (single-valued dependent**) on attribute A of R – in symbols,

  $$R.A \rightarrow R.B \quad \text{or}$$

  $$A \rightarrow B$$

  (read "A **functionally determines** B"

  or "B **is dependent** on A")

  if and only if an A-value in R has associated with it precisely one B-value in R (at any one time). Attributes A and B may be composite.

- **Definition 2**

  Given a relation R, attribute B of R is **functionally dependent** on attribute A of R if and only if, whenever two tuples of R agree on their A-value, they must necessarily agree on their B-value.

Functional dependence

- Given the following relations' structure :

S (s#, sname, status, city)

P (p#, pname, color, weight, city)

SP (s#, p#, qty)

- The functional dependences (FDs) are :

s#	→	sname	relation S
s#	→	status	
s#	→	city	
p#	→	pname	relation P
p#	→	Color	
p#	→	Weight	
p#	→	city	
s#, p#	→	qty	relation SP

Unnormalized relation

- An example of unnormalized relation :

In structure form :

Student(St#, Name, Sex, Address (Number, Street, City))

In tabular form :

Student

St#	Name	Sex	**Address**		
			Number	Street	City
St1	Nantida	F	123	Prachathipat	Hatyai
St2	Waneerat	F	21/33	Silom	Bangkok
St3	Peerapong	M	45	Rama IV	Bangkok
St4	Chanin	M	213	Tanee	Songkla
St5	Wachara	F	23	Mahawong	Chana

- Student is an unnormalized relation because there is another relation (Address) within the relation Student.

Normalization

- Consider the unnormalized relation Student :

Student (St#, Name, Sex, Address (Number, Street, City))

The functional dependencies of Student are :

St# → Name

St# → Sex

St# → Address

- The corresponding normalized relation Student is :

Student (St#, Name, Sex, Number, Street, City)

Now, the functional dependencies of Student are :

St# → Name

St# → Sex

St# → Number

St# → Street

St# → City

1NF relation

- Unnormalized relation :

  Student (St#, Name, Sex, Address (Number, Street, City))

Student

St#	Name	Sex	**Address**		
			Number	Street	City
St1	Nantida	F	123	Prachathipat	Hatyai
St2	Waneerat	F	21/33	Silom	Bangkok
St3	Peerapong	M	45	Rama IV	Bangkok
St4	Chanin	M	213	Tanee	Songkla
St5	Wachara	F	23	Mahawong	Chana

- The corresponding normalized relation (**1NF relation**) :

  Student (St#, Name, Sex, Number, Street, City)

Student

St#	**Name**	**Sex**	**Number**	**Street**	**City**
St1	Nantida	F	123	Prachathipat	Hatyai
St2	Waneerat	F	21/33	Silom	Bangkok
St3	Peerapong	M	45	Rama IV	Bangkok
St4	Chanin	M	213	Tanee	Songkla
St5	Wachara	F	23	Mahawong	Chana

Normal forms

- **First Normal Form (1NF)**

  A relation R is in 1NF if and only if all underlying domains contain **atomic values** only.

- **Second Normal Form (2NF)**

  A relation R is in 2NF if and only if it is in 1NF and every non-key attribute is **fully functionally dependent** on the primary key.

- **Full functional dependence**

  Non-key attribute A of a relation R is fully functionally dependent on the primary key K of R if and only if it is functionally dependent on the primary key K and not functionally dependent on any proper subset of the primary key K.

Normal forms

- **Third Normal Form (3NF)**

  A relation R is in 3NF if and only if it is in 2NF and

  - Every non-key attribute is **non-transitively functionally dependent** on the primary key, or
  - Every **determinant** is a candidate key

- **Transitive functional dependence**

  Let A, B and C are attributes of a relation R, C is said to be transitively functionally dependent on A whenever the functional dependencies

  $A \rightarrow B$ and $B \rightarrow C$ both hold,

  then the **transitive functional dependence**

  $A \rightarrow C$ also holds.

Normal forms

- **Fourth Normal Form (4NF)**

  A relation R is in 4NF if and only if and only if

  - There are no both functional dependency and multi-valued dependency in R, or
  - There is no more than one multi-valued dependency (repeating group) in R.

- **Multi-valued dependency**

  Given a relation R, an attribute B of R is multi-valued dependent on attribute A of R -- in symbols

  R.A $\longrightarrow\!\!\!\!\!\rightarrow$ R.B

  or A $\longrightarrow\!\!\!\!\!\rightarrow$ B

  (read " An A value may **determine** many values of B "

  or " A set of B values is **dependent on** a value of A ")

  if and only if each A-value may have associated with it one or more B-value (at any one time). A and B may be composite.

Full functional dependence

- **Consider the 1NF relation :**

StudentGrade(St#, Sj#, Grade, Name, Sex, GPA)

St1	Sj1	A	Nantida	F	2.45
St1	Sj2	B	Nantida	F	2.45
St1	Sj3	C	Nantida	F	2.45
St2	Sj1	B	Waneerat	F	2.60
St2	Sj3	A	Waneerat	F	2.60
St2	Sj4	C	Wanerrat	F	2.60
St2	Sj6	C	Waneerat	F	2.60
St3	Sj6	C	Peerapong	M	2.76

- **Functional dependencies of StudentGrade are :**

St#, Sj# $\xrightarrow[\textbf{full}]{}$ Grade

St# , Sj# $\xrightarrow[\textbf{not full}]{}$ Name

St# , Sj# $\xrightarrow[\textbf{not full}]{}$ Sex

St# , Sj# $\xrightarrow[\textbf{not full}]{}$ GPA

2NF relations

- **Transfrom the normalized 1NF relations :**

  StudentGrade(St#, Sj#, Grade, Name, Sex, GPA)

  into two corresponding 2NF relations :

  Student(St#, Name, Sex, GPA)

  StdGrade(St#, Sj#, Grade)

- **Example of instances of Student and StdGrade :**

Student (St#, Name, Sex, GPA)

St1	Nantida	F	2.45
St2	Waneerat	F	2.60
St3	Peerapong	M	2.76

StdGrade (St# , Sj# , Grade)

St1	Sj1	A
St1	Sj2	B
St1	Sj3	C
St2	Sj1	B
St2	Sj3	A
St2	Sj4	C
St2	Sj6	C
St3	Sj6	C

Transitive functional dependence

- **Consider the normalized 2NF relation :**

StudentAvsor (St#, Name, Sex, Teacher, Tsex, Tpos)

St1	Nantida	F	Supatta	F	Lect.
St2	Waneerat	F	Supatta	F	Lect.
St3	Peerapong	M	Panya	M	Lect.
St4	Chanin	M	Panya	M	Lect.
St5	Wachara	F	Panya	M	Lect.

- **Functional dependencies of StudentAvsor are :**

St# $\longrightarrow$ Name

St# $\longrightarrow$ Sex

St# $\longrightarrow$ Teacher

St# $\longrightarrow$ Tsex *

St# $\longrightarrow$ Tpos *

Teacher $\longrightarrow$ Tsex

Teacher $\longrightarrow$ Tpos

- Tsex and Tpos are **transitively functionally dependent on** *S#* via Teacher.

3NF relations

- **The 2NF relation that is not a 3NF relation :**

StudentAvsor(St#, Name, Sex , Teacher, Tsex, Tpos)

St1	**Nantida**	**F**	**Supatta**	**F**	**Lect.**
St2	**Waneerat**	**F**	**Supatta**	**F**	**Lect.**
St3	**Peerapong**	**M**	**Panya**	**M**	**Lect.**
St4	**Chanin**	**M**	**Panya**	**M**	**Lect.**
St5	**Wachara**	**F**	**Panya**	**M**	**Lect.**

- **Transform the StudentAdvisor relation into two 3NF relations :**

Student(St#, Name, Sex, Teacher)

St1	**Nantida**	**F**	**Supatta**
St2	**Waneerat**	**F**	**Supatta**
St3	**Peerapong**	**M**	**Panya**
St4	**Chanin**	**M**	**Panya**
St5	**Wachara**	**F**	**Panya**

Advisor(Teacher , Tsex , Tpos)

Supatta	**F**	**Lect.**
Panya	**M**	**Lect.**

Repeating groups (multi-value dependencies)

- **Consider the following facts :**

Emp#	ChildName	MajorGraduated
E1	Ratree Maliwan Sritrang	Mathematics Computer science
E2	Maliwan	Mathematics Statistics

- **Put the above facts into relation :**

Employee (Emp#, ChildName, MajorGraduated)

Emp#	ChildName	MajorGraduated
E1	Ratree	Mathematics
E1	Maliwan	Mathematics
E1	Sritrang	Mathematics
E1	Ratree	Computer science
E1	Maliwan	Computer science
E1	Sritrang	Computer science
E2	Maliwan	Mathematics
E2	Maliwan	Statistics

3NF Relation with one repeating group

- **Consider the following relation :**

  Employee (Emp#, Ename, Sex, ChildName)

Emp#	Ename	Sex	ChildName
E1	Sompong	M	Anongnart
E1	Sompong	M	Ananva
E1	Sompong	M	Apiwat
E2	Pongpat	M	Ananya
E2	Pongpat	M	Apichart
E3	Wacharee	F	Narumon
E3	Wacharee	F	Ananya
E6	Harnnarong	M	Phasai
E6	Harnnarong	M	Apiwat
E6	Harnnarong	M	Apichart

- **The relation Employee is a 3NF relation with the following implicit functional dependencies and multi-valued dependency :**

  Emp# $\longrightarrow$ Name

  Emp# $\longrightarrow$ Sex

  Emp# $\longrightarrow\!\!\!\!\rightarrow$ ChildName

- **Transform the relation Employee into two 4NF relations :**

  Employee(Emp#, Ename, Sex)

  Children(Emp#, ChildName)

4NF relations

- **The 4NF relations Employee and Children in tabular form :**

Employee

Emp#	**Ename**	**Sex**
E1	Sompong	M
E2	Pongpat	M
E3	Wacharee	F
E6	Harnnarong	M

Children

Emp#	**ChildName**
E1	Anongnart
E1	Ananya
E1	Apiwat
E2	Ananya
E2	Apichart
E3	Narumon
E3	Ananya
E6	Phasai
E6	Apiwat
E6	Apichar

3NF relation with two repeating groups

- **Consider the following following relation :**

Employee(Emp#, ChildName, MajorGraduated)

E1	Ratree	Mathematics
E1	Maliwan	Mathematics
E1	Sritrang	Mathematics
E1	Ratree	Computer science
E1	Maliwan	Computer science
E1	Sritrang	Computer science
E2	Maliwan	Mathematics
E2	Maliwan	Statistics

- **Relation Employee is 3NF but not 4NF because there are two implicit multi-valued dependencies :**

Emp# $\longrightarrow\!\!\!\!\!\rightarrow$ ChildName

Emp# $\longrightarrow\!\!\!\!\!\rightarrow$ MajorGraduted

- **Transform relation Employee into two 4NF relations :**

Emp-Child(Emp#, Childname)

Emp-Major(Emp#, MajorGraduated)

Conclusion of relational database design

Do not represent more than one ‘concept’ or ‘entity type’ in a single relation, otherwise you will face the difficulties of anomalies due to updates, insertions, and deletions.

****

Database design and construction processes

The database design and construction processes can be divided into the following steps :

- **Data and semantics requirements analysis**
- **Conceptual database design using**
  - **E-R diagram, or**
  - **O-R diagram**
- **Logical database design**
  - **Relational database structure, or**
  - **Hierarchical database structure, or**
  - **Network database structure**
- **Logic database structure refinement (optional !)**
  - **Normalization process**
- **Integrity constraints and security specification**
- **Construction of logical database schema and subschema via DBMS**
- **Physical database construction by DBMS**