CS 4221: Database Design

The Relational Model

Ling Tok Wang
National University of Singapore

Topics:

- **Basic concepts in Relational Model**
  - FD, transitive dependency, key, primary key, updating anomalies, properties of FDs
- **Normal Forms**
  - 1NF, 2NF, 3NF, BCNF; redundancy in NF relations
- **Decomposition Approach**
  - Universal Relation Assumption, problems of decomposition approach
- **Sythesizing Approach**
  - FD inference rules, closure of FDs, closure of attributes, FD membership test, criteria for normalization, local/global redundancy, Bernstein's Algorithm and its weak points
- **4NF**
  - MVDs, MVD inference rules, properties of FDs and MVDs, decomposition approach, MVDs and hierarchical model
- **5NF and DKNF**
  - Will not be covered/examined due to time limit

- Will show many commonly misunderstood important concepts and errors.

First Normal Form (1NF) Relation

Defn: Given sets of **atomic** (i.e. non-decomposable) elements D_1, D_2, …, D_n (**not necessarily distinct**), R is a **first normal form** (1NF) relation on these n sets if it is a **set** of **ordered** n-tuples $< d_1, d_2, \ldots, d_n >$ such that

$d_i \in D_i \quad \forall\, i = 1, 2, \ldots, n.$ (**Note**: $\forall$ means "for all")

Thus $R \subseteq D_1 \times D_2 \times \ldots \times D_n$
where x is the Cartesian product operator.

- **Note:** A **set** has **no** duplicates. An n-tuple is **ordered** means the orders of the n components of the tuple are important.

Defn: D_1, …, D_n are called the **domains** of R. Each domain may be assigned a unique role name, called an **attribute** of R.

Defn: For any tuple in R, the value of an attribute named **B** is referred to as a **B-value**.

For a set of attributes **X** $= \{B_1, \ldots, B_m\}$, the values of the attributes in X of any tuple in R is referred to as an **X-value**.

E.g. A relation **Take** which contains information on courses taken by students. Take is a 1NF relation.

There are different ways to express the relation Take:

(1) Take $\subseteq$ char(10) x char(6) x char(30) x char(60) x int (domains)

(2) Take $\subseteq$ Student# x Course# x S-name x C-desc x Mark (attributes)

- (3) Take (Student#, Course#, S-name, C-desc, Mark) (attributes)

Take	Student#	Course#	S-name	C-desc	Mark
	95001	CS1101	Tan CK	Programming	75
	95023	CS1101	Lee SL	Programming	58
	94257	CS2103	Tan CK	Data Stru	64
	…				

Defn: A set of attributes Y of R is said to be **functionally dependent** (**FD**) on a set of attributes X of R if each X-value in R has associated with **exactly one** Y-value in R **at any time.**

This is denoted by

$$\mathbf{X \rightarrow Y}$$

and is called a **functional dependency** of R.

- **Q:** Why "at any time"?

Defn: A functional dependency $X \rightarrow Y$ is said to be trivial if $Y \subseteq X$.

- **Q:** Why call it "trivial"?

Defn: A functional dependency $X \rightarrow Y$ of R is said to be a **full dependency** of R (or Y is **fully dependent** on X) if it is a non-trivial FD and there exists no proper subset X' of X such that $X' \rightarrow Y$.

Defn: A set of attributes K of a relation R is said to be a **candidate key** (or simply **key**) of R if all attributes of R are functionally dependent on K and there exists **no** proper subset K' of K such that all attributes of R are functionally dependent on K'.

Defn: If there are more than one key for a relation, one of the keys is designated as the **primary key** of the relation.

- **Q:** How do we choose the primary key of a relation? What are the selection criteria?

Defn: An attribute of R is called a **prime attribute** (or **prime**) if it is contained in **some** key of R. All other attributes of R are called **non-prime attributes** of R.

Example 1.

Let Take be a relation with the set of attributes:

{STUDENT#, COURSE#, S-NAME, C-DESCRIPTION, MARK}

We have the following functional dependencies in Take:

STUDENT# → S-NAME
COURSE# → C-DESCRIPTION
STUDENT#, COURSE# → MARK

❖ **Q:** How can we find/know these FDs?
Can we use some data mining techniques to find FDs in a RDB?
Why each student only has one name?

{STUDENT#, COURSE#} is the only key of the relation.

STUDENT# and COURSE# are primes, the rest are non-primes.

Q: Do the below FDs also hold in the relation Take?

STUDENT#, COURSE# → S-NAME
STUDENT#, COURSE# → C-DESCRIPTION
STUDENT#, S-NAME, COURSE# → MARK

- **Insertion anomaly** – if a new course is created but no students have taken this course, then we cannot enter the information about this course because the use of **null values** or **undefined values** in the primary key could cause problem.
- **Deletion Anomaly** - similiar
- **Rewriting anomaly** - similiar

These three anomalies are called the **updating anomalies.**

❖ **Q:** What causes these updating anomalies?

- One process which attempts to remove these undesirable updating anomalies from the relation is called **normalization.**
- The relation Take can be decomposed into (**Q:** How?)

R1 (STUDENT#, S-NAME)
R2 (COURSE#, C-DESCRIPTION)
R3 (STUDENT#, COURSE#, MARK)

❖ **Notation:** A contiguous underline indicates a key of the relation.
E.g. In R3, attributes STUDENT# and COURSE# form **a** key of the relation R3.

The above 3 relations do not have updating anomalies. Prove it!

Second Normal Form (2NF) Relation

Defn: A first normal form relation is called a **second normal form** (**2NF**) relation if and only if every non-prime attribute of R is fully dependent on **each** key of R.

Note that the relation Take in Example 1 is **not** in 2NF.

Take (STUDENT#, COURSE#, S-NAME, C-DESCRIPTION, MARK)

For example, S-Name is a non-prime and it is not fully dependent on the key {STUDENT#, COURSE#}. **Q:** Why?

The name of a student is duplicated if the student takes more than one course.

Example 2. **SP (S#, Sname, P#, Pname, Price)**

A supplier with supplier number (S#) and name (Sname) supplies a part with part number (P#) and name (Pname) with a price (Price). FDs in relation SP are:

S# → Sname (A supplier only has one name)

P# → Pname (A part only has one name)

S#, P# → Price (A supplier supplies a part with one price at any one time)

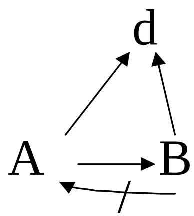
{S#, P#} is the only key of the relation SP. Prove it!

Relation SP is not in 2NF as Sname is not fully dependent on the key. Q: Why?

There are redundant information on Sname and Pname in SP.

Third Normal Form (3NF) Relation

Defn: Let A and B be two **distinct sets** of attributes (i.e. not identical) of a relation R, and d be an attribute of R which does not belong to A or B such that

- 

  $A \rightarrow B$ $\quad$ $B \rightarrow d$ $\quad$ $B \not\rightarrow A$

Then we say that d is **transitively dependent** on A under R, and $A \rightarrow d$ is a **transitive dependency**.

- **Intuitive meaning**: A transitive dependency can be derived from other FDs, so it is redundant and can be removed.

Notation: $B \not\rightarrow A$ means A is not functionally dependent on B.

- **Q:** What if we have $B \rightarrow A$ instead?

Defn: A relation is in Codd **third normal form** (**3NF**) if and only if it is in 2NF and **each** non-prime attribute of R is **not** transitively dependent on **each** key of R.

Note: All the three relations:

R1 (STUDENT#, S-NAME)
R2 (COURSE#, C-DESCRIPTION)
R3 (STUDENT#, COURSE#, MARK)

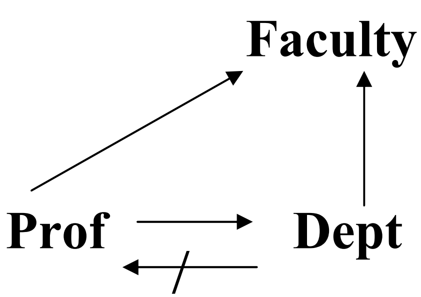
in Example 1 are in 3NF. Prove it!

Example 3. R (Prof, Dept, Faculty)

We have the below FDs: (**Q:** How to find them?)

Prof $\rightarrow$ Dept, Faculty
Dept $\rightarrow$ Faculty

❖ **Q:** Why Prof $\rightarrow$ Dept ? Is it true in any university?



Note that R is in 2NF but not in 3NF because

Prof $\rightarrow$ Faculty

is a transitive dependency.

We decompose this relation into

R1 (Prof, Dept)
R2 (Dept, Faculty)

They are both in 3NF.

Boyce-Codd normal form (BCNF) Relation

Defn: A relation R is in **Boyce-Codd normal form** (**BCNF**) if and only if it is in 1NF and for every attribute set A of R, if **any** attribute of R **not** in A is functionally dependent on A, then **all** attributes in R are functionally dependent on A.

Q: Are the below 3 relations in BCNF?

R1 (STUDENT#, S-NAME)
R2 (COURSE#, C-DESCRIPTION)
R3 (STUDENT#, COURSE#, MARK)

Q: Are the below 2 relations in BCNF?

R1 (Prof, Dept)
R2 (Dept, Faculty)

❖ **Q:** Are there updating anomalies in a BCNF relation?
The answer is still yes but in fewer cases. **Q:** Why?

Example 4.

Consider the relation STJ with the below FDs:

STJ (STUDENT, TEACHER, SUBJECT)

Assume that we have the below constraints:

1. For each subject, each student of that subject is taught by only one teacher.

   STUDENT, SUBJECT $\rightarrow$ TEACHER

2. Each teacher teaches only one subject.

   TEACHER $\rightarrow$ SUBJECT

3. Some subjects are taught by more than one teacher

   SUBJECT $\not\longrightarrow$ TEACHER

Q: What are the keys of the relation SPJ? Primes ?

Q: Is it in 3NF?

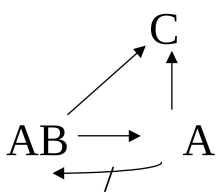
Q: Is it in BCNF?

❖ **Q:** If a relation is not in BCNF, can we always normalize it to a set of BCNF relations? **Ans:** Not always.

Example 5.

R ($\underline{A, B}$, C, D, F)

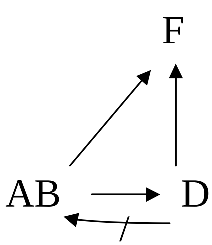
with AB $\rightarrow$ CDF, A $\rightarrow$ C, D $\rightarrow$ F



- R is **not in 2NF** since C is not fully dependent on the key AB.

  Decompose it, we get:

  R_1 ($\underline{A}$, C) and R_2 ($\underline{A, B}$, D, F)



- R_2 is **not in 3NF** since AB $\rightarrow$ F is a transitive dependency. Decompose it, we get

  R_1 ($\underline{A}$, C), R_{21} ($\underline{A, B}$, D), R_{22} ($\underline{D}$, F)

- All are in 3NF. **Q:** Are they also in BCNF?

E.g. R (<u>A, B</u>, C, D) with $AB \rightarrow CD$ and $D \rightarrow B$

R is in 3NF but not in BCNF since $D \rightarrow B$ but $D \not\rightarrow C$

Q: What are the keys ? **Hint:** There are 2 keys.

E.g. Enrol (<u>S#, C#,</u> Sname, Mark)

where S#, C# $\rightarrow$ Sname is a transitive dependency and the relation Enrol is not in 3NF.

In fact, it is not in 2NF also. **Q:** Why?

Decomposition & Synthesizing Method

- for Relational Database Design

- Three common methods for relational database schema design are the decomposition method, the synthesizing method, and the Entity-Relationship Approach.
- **The decomposition method** is based on the assumption that a database can be represented by a **universal relation** which contains all the attributes of the database (this is called **the universal relation assumption**) and this relation is then **decomposed** into smaller relations in order to remove redundant data.
- **The synthesizing method** is based on the assumption that a database can be described by a given set of attributes and a given set of functional dependencies, and 3NF or BCNF relations are then **synthesized** based on the given set of dependencies.

  Note: Synthesizing method assumes universal relation assumption also.
- We will discuss the **Entity-Relationship Approach** later.
- Examples 3 & 5 use the decomposition method.

Properties of **Universal Relation Assumption**

- Decomposition method and synthesizing method do not change any attribute name and do not delete any attribute or add new attributes to the database.
- Two attributes with the same name from 2 relations are referred to some same attribute in the universal relation, i.e. they are from the same attribute and of the same semantics (same meaning).
- Two attributes with different names from 2 different relations or from a relation are referred to two different attributes in the universal relation, and they have different semantics.

Example: A database SP has the below 3 relations:

Supplier (<u>Code</u>, Sname),
Part (<u>Code</u>, Pname, Color)
Supply (<u>Supplier, Part</u>, Price)

This database SP does not satisfy the universal relation assumption.

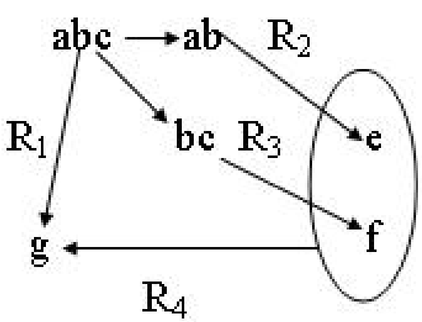
Q: Why? Bad design on attribute names.

Some properties of normal form relations:

1. BCNF $\Rightarrow$ 3NF $\Rightarrow$ 2NF $\Rightarrow$ 1NF (prove them!)
2. A set of 3NF relations always exists for a given set of functional dependencies, but it is **not true** for Boyce-Codd norm form relation set.

   E.g. The relation R (s, j, t)
   with functional dependencies
   $s\ j \rightarrow t, \qquad t \rightarrow j$
   is in 3NF but has no BCNF relation set which
   covers the given functional dependencies

3. Even BCNF relations can suffer from the updating anomalies



E.g. Let R = { $R_1(\underline{a,b,c},g,h)$, $R_2(\underline{a,b},e)$, $R_3(\underline{b,c},f)$, $R_4(\underline{e,f},g)$ }
with the set of full dependencies:
$\mathbb{G} = \{ abc \rightarrow g,\ abc \rightarrow h,\ ab \rightarrow e,\ bc \rightarrow f,\ ef \rightarrow g \}$

❖ **Note:** All the relations in R are in BCNF.
However, there are two different ways to find the g-value of any given {a,b,c}-value via different relations. So, there are redundancies and R has updating anomalies. In fact, g in R1 is superfluous and can be removed.

Properties of FDs (inference rules)

Defn: Given a relation R having a set of attributes $\mathbb{A}$ and a given set of functional dependencies $\mathbb{F}$, the **closure** of $\mathbb{F}$, denoted by $\mathbb{F}^+$, is defined as follows:

(1) $\mathbb{F} \subseteq \mathbb{F}^+$

(2) **Projectivity**: $\forall$ X, Y $\subseteq \mathbb{A}$

If $Y \subseteq X$ then $X \to Y \in \mathbb{F}^+$

Note: $\forall$ means "for all"

(3) **Transitivity**: $\forall$ X, Y, Z $\subseteq \mathbb{A}$

If $X \to Y$, $Y \to Z \in \mathbb{F}^+$

Then $X \to Z \in \mathbb{F}^+$

(4) **Union** (or **Additivity**): $\forall$ X, Y, Z $\subseteq \mathbb{A}$

If $X \to Y$, $X \to Z \in \mathbb{F}^+$

Then $X \to Y \cup Z \in \mathbb{F}^+$

(5) No other functional dependencies are in $\mathbb{F}^+$.

Result: $\mathbb{F}^+$ is **sound** and **complete**. **Q: What are their meanings?**

Q: What is the meaning of "**closure**"?

Another definition for the **closure** of $\mathbb{F}$ (**Armstrong's Axioms**):

(1) $\mathbb{F} \subseteq \mathbb{F}^+$
(2) **Reflexivity**: $X \rightarrow X \in \mathbb{F}^+ \quad \forall X \subseteq \mathbb{A}$
(3) **Augmentation**: $\forall X, Y, Z \subseteq \mathbb{A}$
If $X \rightarrow Z \in \mathbb{F}^+$ then $X \cup Y \rightarrow Z \in \mathbb{F}^+$
(4) **Pseudo-transitivity**: $\forall X, Y, Z, W \subseteq \mathbb{A}$
if $X \rightarrow Y \in \mathbb{F}^+$, $Y \cup Z \rightarrow W \in \mathbb{F}^+$
then $X \cup Z \rightarrow W \in \mathbb{F}^+$
(5) No other FDs are in $\mathbb{F}^+$

Result: The above 2 definitions for the closure of $\mathbb{F}$ are equivalent.

Note: We usually simply write $X \cup Y$ as "X, Y" or {X, Y}.

Defn: Two sets of attributes A and B of a relation are said to be **functionally equivalent** if and only if

$$A \rightarrow B \in \mathbb{F}^{+} \text{ and } B \rightarrow A \in \mathbb{F}^{+}$$

A and B are said to be ***properly* functionally equivalent** if and only if A and B are functionally equivalent and $\nexists$ $A_1 \subset A$ and $B_1 \subset B$ such that $A_1 \rightarrow B \in \mathbb{F}^{+}$ or $B_1 \rightarrow A \in \mathbb{F}^{+}$

Note: $\exists$ means there exists, and $\nexists$ means there does not exist

Result: A relation R is in 3NF if and only if **each** non-prime attribute is not transitivity dependent on an **arbitrarily chosen** key of R. (Prove it!)

- **Q:** What is the use of this result?

E.g. Let $\mathbb{A}$ = {A, B, C}, $\mathbb{F}$ = {A → B, B → C}

$\mathbb{F}^+$ = { A → A, B → B, C → C,
AB → A, AB → B, AB → AB,
BC → B, BC → C, BC → BC,
AC → A, AC → C, AC → AC,
ABC → A, ABC → B, ABC → C,
ABC → AB, ABC → AC, ABC → BC,
ABC → ABC, /* all the above FDs are trivial
A → B, B → C, A→ C, A → BC,
A → AC, A → AB, A → ABC, B → BC,
/* all the below FDs are non full dependencies
AC → B, AC → BC, AC → AB, AC → ABC,
AB → C, AB → BC, AB → AC, AB → ABC }

Q: Do we need find the closure of a set of FDs during normalization?

Note: There are too many FDs in the closure. We don't really need to find the closure. However it is important test whether a FD is in a closure or not.

Q: What is the intuitive meaning of "a FD is in the closure of a set of FDs"?

❖ FD **Membership Problem**:

Given a set of FDs $\mathbb{F}$ defined on $\mathbb{A}$, $X \subseteq \mathbb{A}$ and $y \in \mathbb{A}$, is $X \to y \in \mathbb{F}^+$?
i.e. can $X \to y$ be derived from $\mathbb{F}$?

Example: Let $\mathbb{G} = \{ AB \xrightarrow{1} C,\ C \xrightarrow{2} D, DE \xrightarrow{3} F, A \xrightarrow{4} E\}$

Show $AB \rightarrow F \in \mathbb{G}^+$.

Note: The numbers are used to identify the FDs.

Solution:

$$\begin{aligned} AB &\xrightarrow{1} ABC \xrightarrow{2} ABCD \\ &\xrightarrow{4} ABCDE \xrightarrow{3} ABCDEF \\ &\rightarrow F \end{aligned}$$

$\therefore\ AB \rightarrow F \in \mathbb{G}^+$

Q: How to prove each step using the FD inference rules?

Detailed steps for proving $AB \rightarrow F \in \mathbb{G}^+$

(1) Prove $AB \xrightarrow{1} ABC$

Since $AB \rightarrow AB$ (by projectivity)

$AB \rightarrow C$ (given)

so $AB \rightarrow ABC$ (by additivity)

(2) Prove $ABC \xrightarrow{2} ABCD$

Since $C \rightarrow D$ (given)

$ABC \rightarrow C$ (by projectivity)

so $ABC \rightarrow D$ (by transitivity)

Also $ABC \rightarrow ABC$ (by projectivity)

so $ABC \rightarrow ABCD$ (by additivity)

(3) Prove $AB \xrightarrow{1,2} ABCD$

From (1) we have $AB \rightarrow ABC$

From (2) we have $ABC \rightarrow ABCD$

so $AB \rightarrow ABCD$ (by transitivity)

(4) …

Note: The proof is too long. Any better way?

Alternative Solution to prove $X \rightarrow Y$ in G^+

❖ **Defn:** Given a set of attributes $\mathbb{X}$, the **closure** of $\mathbb{X}$ relative to $\mathbb{G}$ is defined as:

$$\mathbb{X}^+ = \{ y \in \mathcal{A} \mid X \rightarrow y \in \mathbb{G}^+ \}$$

Q: What is the intuitive meaning of the closure of X?

Q: How to construct X^+ relative to a given set of FDs G?

❖ **Alternative Solution:** To test $X \rightarrow Y$ in G^+, we can just test whether Y is in X^+, the closure of X relative to G.

E.g. Let $\mathbb{G} = \{ AB \xrightarrow{1} C,\ C \xrightarrow{2} D,\ DE \xrightarrow{3} F,\ A \xrightarrow{4} E \}$

$$\begin{aligned} \{A\ B\}^+ &\overset{1}{=} \{A\ B\ C\}^+ \overset{2}{=} \{A\ B\ C\ D\}^+ \\ &\overset{4}{=} \{A\ B\ C\ D\ E\}^+ \overset{3}{=} \{A\ B\ C\ D\ E\ F\}^+ \\ &= \{A\ B\ C\ D\ E\ F\} \end{aligned}$$

$$\therefore \quad AB \rightarrow F \in G^+$$

Three Criteria for Normalization

(1) **Reconstructibility (or losslessness).**

If an original relation R is split into n relations $R_1, R_2, \ldots, R_n$, then $R_i = R[A_i]$ (where [ ] is the projection operator)

and $R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n = R$

where A_i is the attribute set of R_i $\forall$ i = 1, 2, …, n

and $\bowtie$ is the **join** operator

Note: The join operator is also denoted by *****.

Defn: Two sets of FDs, $\mathbb{F}$ and $\mathbb{G}$ are **equivalent** if and only if $\mathbb{F}^+ = \mathbb{G}^+$.

If $\mathbb{F}$ and G are equivalent, we say $\mathbb{F}$ **covers** $\mathbb{G}$, $\mathbb{G}$ covers $\mathbb{F}$, $\mathbb{F}$ is a **cover** of $\mathbb{G}$, or $\mathbb{G}$ is a cover of $\mathbb{F}$.

❖ (2) **Covering**.

$$\mathbb{F}^+ = (\mathbb{F}_1 \cup \mathbb{F}_2 \ldots \cup \mathbb{F}_n)^+$$

where $\mathbb{F}$ is the set FDs for the original relation R and $\mathbb{F}_i$ is the set of FDs in relation R_i $\forall$ i = 1, 2, …, n.

(3) Each relation is free of redundant attributes (i.e. no local redundancy – no redundancy within each relation).

❖ **Q:** Is it ture that **$(F \cup G)^+ = F^+ \cup G^+$** for any two sets of FDs F and G?

❖ **Note:** In fact, free of **local redundant** attributes is not enough, **global** redundancy (i.e. redundancy among relations) may still exist. (see **LTK normal form**)

Ref: Tok Wang Ling, Frank W Tompa, Tiko Kameda, An Improved Third Normal Form for Relational Databases, ACM TODS, vol 6, no 2, pp329-346, 1981.

Example Given R (A, B, C) with $C \rightarrow A$

R is in 3NF, but not in BCNF

If we decompose R into 2 relations

R_1 (C, A) and R_2 (C, B)

then we lose the FD $AB \rightarrow C$.

This violates the covering criteria. Why?

Synthesizing Third Normal Form Relations

(by Philip A. Bernstein, TODS 1979)

Algorithm

1. (**Eliminate extraneous attributes**). Let $\mathbb{F}$ be the given set of FDs where the right side of each FD is a single attribute.
Eliminate **extraneous attributes** from the left side of each FD in $\mathbb{F}$, producing the set $\mathbb{G}$.

2. (**Finding covering**). Find a **non-redundant** covering $\mathbb{H}$ of $\mathbb{G}$.

3. (**Partition**). Partition $\mathbb{H}$ into groups such that all of the FDs in each group have identical left sides.

4. (**Merge equivalent keys**). Let J = Φ.

   For each pair of groups, say H_i and H_j with left sides X and Y resp.
   If X and Y are properly equivalent, then
   (a) **merge** H_i and H_j together
   (b) add $X \rightarrow Y$ and $Y \rightarrow X$ to J
   (c) if $X \rightarrow Z \in H$ and $Z \in Y$, then delete $X \rightarrow Z$ from $\mathbb{H}$.
   Similarly, if $Y \rightarrow Z \in H$ and $Z \in X$, then delete $Y \rightarrow Z$ from $\mathbb{H}$.

❖ **5. (Eliminate transitive dependencies).**
Find a minimal $\mathbb{H}' \subseteq \mathbb{H}$ such that
$(\mathbb{H}' \cup J)^+ = (\mathbb{H} \cup J)^+$
Then add each FD of J into its corresponding group of $\mathbb{H}'$.

6. (Construct relations)
Each group in $\mathbb{H}'$ forms a relation.
Each set of attributes that appears on the left side of any FD in the group is a key of the relation formed by the group. They are called explicit keys.

Note: There may have more than one key for some relations constructed.

Result: The relations produced by step 6 are all in 3NF.

Result: The number of relations produced is minimum.

❖ **Q: What is the difference between "minimal" and "minimum"?**

Example 1

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

Step 1. (Eliminating extraneous attributes)

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

(since $A\,E \rightarrow A\,B\,E \in \mathbb{F}^{+}$)

Step 2. (Find covering)

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

(since $A \rightarrow C \in (G - \{A \rightarrow C\})^{+}$)

Step 3 (Partition)

$$H_1 = \{ A \rightarrow B \}$$
$$H_2 = \{ B \rightarrow C, B \rightarrow D \}$$
$$H_3 = \{ D \rightarrow B \}$$
$$H_4 = \{ A\,E \rightarrow F \}$$

Step 4 (Merge groups)

B and D are properly equivalent

$J = \{ B \rightarrow D, D \rightarrow B \}$

$H_1 = \{A \rightarrow B\}$

$H'_2 = H_2 \cup H_3 - \{B \rightarrow D, D \rightarrow B\}$

$= \{B \rightarrow C\}$

$H_4 = \{AE \rightarrow F\}$

Step 5 (Eliminate transitive dependencies)

None! (You should verify this).

Step 6 (Construct relations)

R_1 (<u>A</u>, B)

R_2 (<u>B</u>, <u>D</u>, C)

R_3 (<u>A, E</u>, F)

Example 2 (need step 5)

Given $\mathbb{F} = \{X_1 X_2 \rightarrow AD,\ CD \rightarrow X_1 X_2,$
$A X_1 \rightarrow B,\ B X_2 \rightarrow C,\ C \rightarrow A\}$

Step 1. $\mathbb{G} = \mathbb{F}$

Step 2. $\mathbb{H} = \mathbb{G}$

Step 3
$H_1 = \{X_1 X_2 \rightarrow AD\}$
$H_2 = \{CD \rightarrow X_1 X_2\}$
$H_3 = \{A X_1 \rightarrow B\}$
$H_4 = \{B X_2 \rightarrow C\}$
$H_5 = \{C \rightarrow A\}$

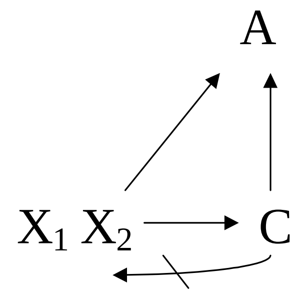
Step 4
$J = \{X_1 X_2 \rightarrow CD,\ CD \rightarrow X_1 X_2\}$
$H'_1 = H_1 \cup H_2 - J$
$= \{X_1 X_2 \rightarrow A\}$
$H_3 = \{A X_1 \rightarrow B\}$
$H_4 = \{B X_2 \rightarrow C\}$
$H_5 = \{C \rightarrow A\}$

- Step 5 (Eliminate TD)

We can eliminate $X_1X_2 \rightarrow A$

since $X_1 X_2 \rightarrow CD$, $C \rightarrow A$

and $C \not\rightarrow X_1 X_2$



so we get

$J = \{X_1 X_2 \rightarrow CD, CD \rightarrow X_1 X_2\}$

$H'_1 = \phi$

$H_3 = \{A X_1 \rightarrow B\}$

$H_4 = \{B X_2 \rightarrow C\}$

$H_5 = \{C \rightarrow A\}$

Step 6

R_1 ($\underline{X_1, X_2}$, $\underline{C,D}$) Note: 2 keys: $\{X_1, X_2\}$ and $\{C, D\}$

R_2 ($\underline{A, X_1}$, B)

R_3 ($\underline{B, X_2}$, C)

R_4 ($\underline{C}$, A)

- **Note:** If we omit step 5, then R_1 will be

  R_1 ($\underline{X_1, X_2}$, $\underline{C,D}$, A)

  Which is not in 3NF. Why?

Some shortcomings of Bernstein's algorithm

- **Shortcoming 1.** Bernstein's algorithm does not guarantee reconstructibility (or losslessness).

Example 3. Given R (Course#, Preq#, Cname, Cdesc) with

$$\mathbb{F} = \{\text{Course\#, Preq\#} \rightarrow \text{Cname}$$
$$\text{Course\#} \rightarrow \text{Cname, Cdesc}\}$$

Step 1 $\mathbb{G} = \{\text{Course\#} \rightarrow \text{Cname, Cdesc}\}$

Step 2 $\mathbb{H} = \mathbb{G}$

:

Step 6 R_1 (<u>Course#</u>, Cname, Cdesc)

Note: We lose information about Preq#.

- **Q:** How to resolve this problem?

In fact we have Course# $\twoheadrightarrow$ Preq#

(**Note**. It is a multi-valued dependency, to be discussed later. Bernstein's algorithm does not handle MVDs).

We need another relation:

R_2 (<u>Course#, Preq#</u>)

❖ **Shortcoming 2.** Bernstein's algorithm does not find **all** the keys.

Example 4. Given R (A, B, C, D)
with $\mathbb{F} = \{ AB \rightarrow CD, C \rightarrow B \}$
Apply the algorithm, we will get
R_1 (<u>A, B</u>, C, D)
R_2 (<u>C,</u> B)

❖ In fact, {A, C} is also a key of R_1.
This is called an implicit key.

❖ **Note:** R_1 is not in BCNF.

❖ **Note:** To find all the keys of a relation is NP-complete.

Q: What is the meaning of NP-complete? A term from complexity theory.

❖ **Shortcoming 3.** Bernstein's algorithm does not remove all the superfluous attributes (i.e. redundant attributes).

Example 5. Given $\mathbb{F} = \{ AD \rightarrow B, B \rightarrow C, C \rightarrow D, AB \rightarrow E, AC \rightarrow F \}$

Step 1 $\mathbb{G} = \mathbb{F}$

Step 2 $\mathbb{H} = \mathbb{G} = \mathbb{F}$

:

Step 6 R_1 ($\underline{A, B}$, C, D, E, F)

R_2 ($\underline{B}$, C)

R_3 ($\underline{C}$, D)

❖ **Note:** C is superfluous in R1, but R1 is in 3NF. However, D is not superfluous. Remove C from R_1 and get

R'_1 ($\underline{A, B}$, D, E, F)

Note: Ling & Tompa & Kameda method removes **all** superfluous attributes.

❖ **Shortcoming 4.** The set of relations produced by the algorithm depends on the non-redundant covering found.

Example 6. Given $\mathbb{F} = \{AD \to B, B \to C, C \to D, AB \to \mathbf{E}, AC \to \mathbf{F}, AD \to \mathbf{F}, AC \to \mathbf{E}\}$

Case 1 If $\mathbb{H} = \{AD \to B, B \to C, C \to D, AB \to E, AC \to F\}$

Then the set of relation is

R_1 (<u>A, B</u>, C, D, E, F)

R_2 (<u>B,</u> C)

R_3 (<u>C,</u> D)

Case 2 If $\mathbb{H} = \{AD \to B, B \to C, C \to D, AB \to E, AD \to F\}$

Then the set of relations is

R'_1 (<u>A, B</u>, D, E, F)

R_2 (<u>B,</u> C)

R_3 (<u>C,</u> D)

Case 3 If $\mathbb{H} = \{AD \to B, B \to C, C \to D, AC \to F, AC \to E\}$

Then we have

R''_1 (<u>A, C,</u> D, B, E, F)

R_2 (<u>B,</u> C)

R_3 (<u>C,</u> D

- **Note** that AB is a key but it is not found by the algorithm.

Case 4 If $\mathbb{H} = \{AD \to B, B \to C, C \to D, AC \to E, AD \to F\}$

Then we have

R'''_1 (<u>A, C,</u> D, B, E, F)

R_2 (<u>B,</u> C)

R_3 (<u>C,</u> D)

- **Note** that AB is a key but it is not found by the algorithm.

Note that Case 2 gives the best solution. What is the meaning?

❖ **Shortcoming 5.** A BCNF relation set may contain **superfluous attributes**, i.e. redundant attributes which can be removed.

Example: Given a set of relations

R_1 (<u>Model#, Serial#</u>, **Price**, Color)
R_2 (<u>Model#</u>, Name)
R_3 (<u>Serial#</u>, Year)
R_4 (<u>Name, Year</u>, Price)

❖ **Note:** All relations are in BCNF, but R_1 contains a superfluous attribute **Price**, i.e. Price can be removed from R_1 without losing any information. How to prove it?

❖ **Note:** 3NF and BCNF are defined for **individual relations** but not the **whole relational schema**.

Ref: Ling, Tompa, & Kameda method takes the **whole relational schema** into consideration and removes superfluous attributes.

Note. Some relations generated by Step 6 may have more than one key. We need to choose their preliminary key. Why and how to choose?

- **Q:** Any impact on other relations after choosing primary key for some relation which has more than one key?

E.g. A database schema generated by Bernstein's Algorithm has the below relations:

Student (NRIC, S#, Name, DOB)
Course (C#, Title, Desc)
Take (NRIC, C#, Grade)

Note that Student relation has two keys, i.e. NRIC and S#. We choose S# as its preliminary key, and we also need to change NRIC in Take relation to S# and the relation Take becomes

Take (S#, C#, Grade)

Q: Why?

Fourth Normal Form (4NF) Relation

E.g. The meaning of a given record in the below unnormalized relation (shown on the LHS) is:

the indicated courses are taught by all of the indicated teachers, and uses all the indicated text books.

Its normalized relation CTX is shown on the RHS.

Unnormalized relation **(a nested relation)**

Course	Teacher	Text
Physics	{ Dr. Lee, Dr. Chan}	{Basic Mechanics, Applied Physics}
Math	{Dr. Black}	{Modern Algebra, Geometry}

CTX - normalized relation

Course	Teacher	Text
Physics	Dr. Lee	Basic Mechanics
Physics	Dr. Lee	Applied Physics
Physics	Dr. Chan	Basic Mechanics
Physics	Dr. Chan	Applied Physics
Math	Dr. Black	Modern Physics
Math	Dr. Black	Geometry

Notes:

1. CTX has the following property:
   if $(c, t_1, x_1) \in CTX$ and $(c, t_2, x_2) \in CTX$
   then $(c, t_1, x_2) \in CTX$ and $(c, t_2, x_1) \in CTX$
2. A lot of redundant data in CTX.
3. CTX is in BCNF.

Defn: Given a relation R with attributes A, B, and C, the **multivalued dependency** (**MVD**)

$$R.A \twoheadrightarrow R.B \quad \text{or simply} \quad A \twoheadrightarrow B$$

holds in R if and only if the set of B-values matching a given (A-value, C-value) pair in R depends only on A-value,

i.e. if $(a, b_1, c_1) \in R$, $(a, b_2, c_2) \in R$
then $(a, b_1, c_2) \in R$, $(a, b_2, c_1) \in R$

Another way to view MVD:

Defn: Let R (A, B, C) be a relation and A, B, C be sets of attributes of R, not necessarily disjoint.

Let $B_{ac} = \{\, b \mid (a, b, c) \in R \,\}$ /* a and c are some **A** and **C** values

The **MVD** $\mathbf{A} \twoheadrightarrow \mathbf{B}$ is said to hold for R (A, B, C) if and only if B_{ac} depends on a only,

i.e. $B_{ac} = B_{ac'}$ for all a, c, c' values of attributes A and C, whenever B_{ac} and $B_{ac'}$ are both **non-empty**.

- We sometime use the embedded MVD notation

$$A \twoheadrightarrow B \mid C$$

Note: Pronounce | as independent of. A multi-determines B and independent of C.

- The two definitions for MVD are equivalent.
- For the relation CTX (Course,Teacher,Text), we have

$$\text{Course} \twoheadrightarrow \text{Teacher}$$
$$\text{Course} \twoheadrightarrow \text{Text}$$
i.e. $$\text{Course} \twoheadrightarrow \text{Teacher} \mid \text{Text}$$

Q: What is the intuitive meaning?

❖ **Notes:** (1) $X \twoheadrightarrow \emptyset$ and $X \twoheadrightarrow Y$ hold for $R(X, Y)$.

(2) $X \twoheadrightarrow Y$ whenever $Y \subseteq X \subseteq R$ for R,

there we use R to represent all attributes of relation R also.

These are called **trivial multivalued dependencies**.

Note: $\emptyset$ is the symbol for the empty set.

❖ **Note:** Many text books define trivial MVD using (2).

Recall: A functional dependency $X \rightarrow Y$ is said to be trivial if $Y \subseteq X$.

Defn. A relation R is in **fourth normal form** (**4NF**) if and only if any non-trivial MVD $X \twoheadrightarrow Y$ holds in R implies X is a **superkey** of R,

i.e. $X \rightarrow a$ for **all** attribute a of R.

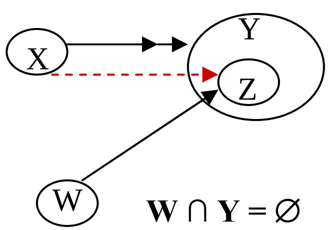
Recall: A relation R is in BCNF iff any non-trivial FD $X \rightarrow Y$ holds in R implies $X \rightarrow a$ for **all** attribute a of R.

Note: A superkey is a key or a superset of a key.

Inference Rules for Multivalued Dependencies

Let R be a relation with attribute set A.

1. (**Complementation**)
   If $X \twoheadrightarrow Y$ then $X \twoheadrightarrow A - X - Y$ (Note: "–" is the set difference)
2. (**Augmentation**)
   If $X \twoheadrightarrow Y$ and $V \subseteq W$
   then $WX \twoheadrightarrow VY$ (Note: WX means W union X, i.e. W and X together)
3. (**Transitivity**)
   $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$ then $X \twoheadrightarrow Z - Y$
4. (**Replication**)
   If $X \rightarrow Y$ then $X \twoheadrightarrow Y$
5. (**Coalescence**) If $X \twoheadrightarrow Y$, $Z \subseteq Y$, and
   for some W disjoint from Y and $W \rightarrow Z$
   then $X \rightarrow Z$ holds also.



Note: These 5 rules plus the 3 rules of Armstrong's Axioms for FDs are **sound** and **complete** for FDs and MVDs.

Result: 4NF relation is also in BCNF.

Theorem. $X \twoheadrightarrow Y$ holds for relation R (X, Y, Z) if and only if R is the **join** of its **projections** R_1 (X, Y) and R_2 (X, Z).

Note: We call $\{R_1, R_2\}$ is a non-loss decomposition of R. R can be reconstructed by joining R_1 and R_2.

- **Corollary.** If a relation is not in 4NF, then there is a non-loss decomposition of R into a set of 4NF relations.

- **Note:** However, it may **not cover** all the given FDs.

E.g. The relation STJ (S, J, T) with

$SJ \rightarrow T$ and $T \rightarrow J$

STJ is not in BCNF so it is not in 4NF.

We can decompose it into two 4NF relations:

R_1 (T, J) and R_2 (T, S)

- R_1 and R_2 form a non-loss decomposition of STJ. However they do not cover the FD: $SJ \rightarrow T$. Bad!

E.g. The relation CTX(<u>course, teacher, text</u>) is in BCNF but not in 4NF since we have:

$$\text{course} \twoheadrightarrow \text{teacher} \mid \text{text}$$

i.e. $\text{course} \twoheadrightarrow \text{teacher}$

and $\text{course} \twoheadrightarrow \text{text}$

❖ Q: How do we know the MVDs?

We can decompose the relation into 2 relations:

CT(<u>course, teacher</u>)

CX(<u>course, text</u>)

Both relations are in 4NF.

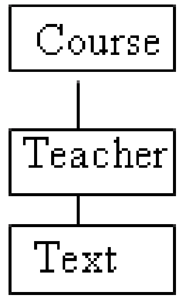
Note that the MVD

$$\text{course} \twoheadrightarrow \text{teacher} \mid \text{text}$$

does not exist in the decomposed relations CT or CX.

❖ **Intuitive meaning** of the MVD: The text books of a course are independent of who are the teachers of the course (perhaps the textbooks of a course are decided by the curriculum committee).

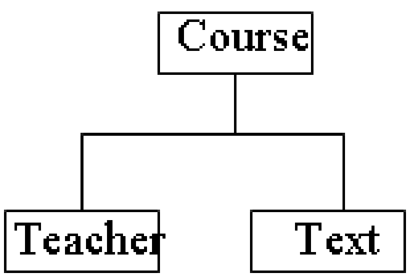
- The relation CTX (course, teacher, text) is similar to the below hierarchical model (and XML):



This is a wrong design in hierarchical model.

Recall that the contiguous underline indicate all the attributes form the key of the relation. It is an all key relation.

- Below is a a correct design:



Note: It can be translated into 2 relations:

CT(Course, Teacher)

CX(Course, Text)

E.g. Let R be a relation

R(employee, child, salary, year)

A tuple < e, c, s, y > in the relation R indicates c is a child of employee e and e got a salary s in year y.

Note that R is in BCNF but not in 4NF, and

$$\text{employee} \twoheadrightarrow \text{child}$$
$$\text{employee} \twoheadrightarrow \{\text{salary, year}\}$$

- **Q:** How do we know/discover these 2 MVDs?

We can decompose R into

R_1 (<u>employee, child</u>)

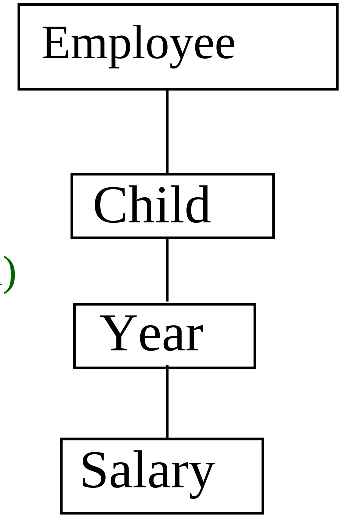
R_2 (<u>employee, salary, year</u>)

Both relations are in 4NF.

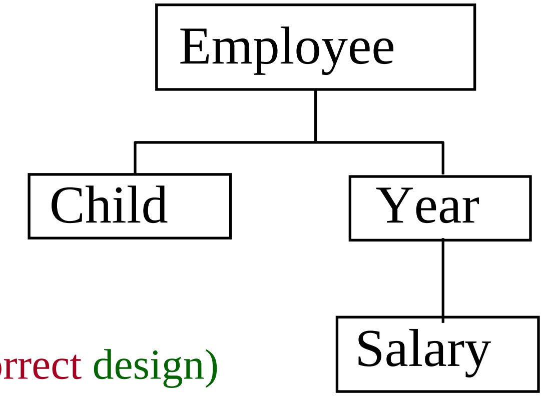
Note that in the above relation, an employee may have more than one salary adjustment within one year.

Q: What if an employee can only has one salary adjustment in January? Any impact on the FDs and MVDs?

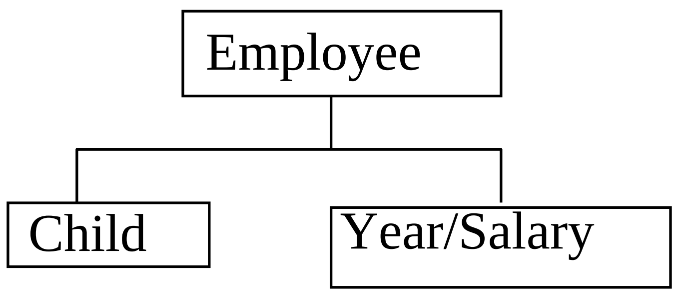
3 possible hierarchical database designs (or XML) of the relation R:



❖ (wrong design)



(correct design)



(another correct design)

More Properties of MVDs

Result: $\varnothing \twoheadrightarrow Y$ in $R(Y, Z)$ iff R is the cartesian product of its projection $R_1(Y)$ and $R_2(Z)$. **Prove it!**

Q: **What is the intuitive meaning of this MVD?**

Note: If $\varnothing \twoheadrightarrow Y$ in $R(Y, Z)$ then $Y_{\varnothing z} = Y_z = \{y \mid (y, z) \in R\} = R[Y]$.

Note: A binary relation is definitely in 3NF but not necessarily in 4NF. How about in BCNF? **Yes. Prove it!**

Result: If $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$

then,
$X \twoheadrightarrow Y \cup Z$ (**multivalued union rule**)
$X \twoheadrightarrow Y \cap Z$ (**multivalued intersection rule**)
$X \twoheadrightarrow Y - Z$ (**multivalued difference rule**)
$X \twoheadrightarrow Z - Y$

Prove them!

Example. Let R(A, B, C, G, H, I) with the following set of dependencies $D = \{ A \twoheadrightarrow B,\ B \twoheadrightarrow HI,\ CG \rightarrow H\}$

(1) Prove $A \twoheadrightarrow CGHI \in D^+$

Since $A \twoheadrightarrow B$, by the complementation rule,
we have $A \twoheadrightarrow R - B - A$
i.e. $A \twoheadrightarrow CGHI \in D^+$
where R means all attributes of the relation R.

Q: Is $A \twoheadrightarrow CGH \in D^+$?

Q: In general, does $A \twoheadrightarrow BC$ imply $A \twoheadrightarrow B$?

(2) Prove $A \twoheadrightarrow HI \in D^{+}$

Since $A \twoheadrightarrow B$ and $B \twoheadrightarrow HI$

By the multivalued transitivity rule, we have

$A \twoheadrightarrow HI - B$

i.e. $A \twoheadrightarrow HI \in D^{+}$

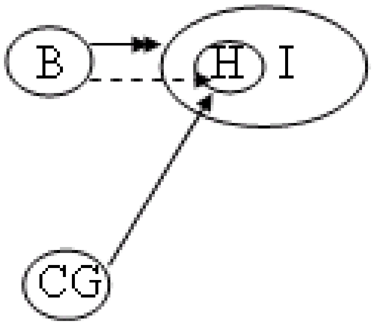
(3) Prove $B \rightarrow H \in D^{+}$

Since $B \twoheadrightarrow HI$

$H \subseteq HI$

$CG \rightarrow H$

$CG \cap HI = \varnothing$



By the coalescence rule, we have

$B \rightarrow H \in D^{+}$

(4) Prove $A \twoheadrightarrow CG \in D^{+}$

By (1) we have $A \twoheadrightarrow CGHI \in D^{+}$

By (2) we have $A \twoheadrightarrow HI \in D^{+}$

By the difference rule, we have

$A \twoheadrightarrow CGHI - HI \in D^{+}$

i.e. $A \twoheadrightarrow CG \in D^{+}$

4NF Decomposition Algorithm (Korth's book page 206)

Given a relation R with a set of FDs and MVDs D

Step 1. (**Initialization**)

result := {R};

done := false;

Step 2. (**Test for non-trivial MVD**)

WHILE (not done) DO

IF (there is a relation $R_i \in$ result that is not in 4NF)

THEN BEGIN

- LET $X \twoheadrightarrow Y$ be a non-trivial MVD that holds on R_i such that

  $X \rightarrow R_i \notin D^+$; /* i.e. X is not a superkey

  /* need to decompose the relation R_i into 2 smaller relations

  SET result := (result – R_i) $\cup$ (R_i – Y) $\cup$ (Relation formed by XY)

END;

ELSE done := true;

- **Q:** How to know relation R_i is not in 4NF? I.e. how to find such MVD $X \twoheadrightarrow Y$ that holds on R_i in Step 2?

- **Note:** There may have several such MVDs, can we just choose anyone of them?

Example. Let $R = (A, B, C, G, H, I)$

$\mathcal{D} = \{A \twoheadrightarrow B,\ B \twoheadrightarrow HI,\ CG \rightarrow H\}$.

Clearly, R is not in 4NF. Why?

(1) Since $A \twoheadrightarrow B$ and A is not a key of R (i.e., $A \rightarrow R \notin \mathcal{D}^+$), using 4NF decomposition algorithm we get

$R_1(\underline{A, B})$ and $R_2(A, C, G, H, I)$

Note that R_1 is in 4NF.

(2) R_2 is not in 4NF (since $CG \rightarrow H$, therefore $CG \twoheadrightarrow H$ in R_2 and CG is not a key of R_2)

Decompose R_2 to get

R_{21} $(\underline{C, G}, H)$ and R_{22} (C, G, A, I)

Note: R_{21} is in 4NF.

(3) We have shown that $A \twoheadrightarrow HI \in D^+$ earlier.

Hence $A \twoheadrightarrow I$ (**prove it!**) holds in R_{22}.

Also A is not a key of R_{22}, R_{22} is not in 4NF. Decompose it into:

R_{221} $(\underline{A, I})$ and R_{222} $(\underline{C, G, A})$

Both are in 4NF.

❖ **Q:** What happen if we first choose $B \twoheadrightarrow HI$ to split the relation? Try it.

- **Note:** The 4NF decomposition algorithm is not a dependency preserving decomposition.

E.g. The relation

SJT (student, subject, teacher)

with D = {teacher $\rightarrow$ subject,
student, subject $\rightarrow$ teacher}

If we use the 4NF decomposition algorithm, we will get

R_1 (<u>teacher</u>, subject)

R_2 (<u>teacher, student</u>)

- The resulting relations do not cover the original FD

  student, subject $\rightarrow$ teacher.

Another method to find 4NF relations

1. Normalize the relation R into a set of 3NF and/or BCNF relations based on the given set of FDs.

2. For each relation, if all attributes belong to the same key and there exists non-trivial MVDs in the relation, then decompose the relation into 2 smaller relations.

- **Q:** How to find such non-trivial MVDs?

- **Q:** How about the covering criteria for normalization?

- **Note**: MVDs are relation sensitive.
  What is the meaning of “relation sensitive”?

- **Note:** When we normalize relations using FDs, we must **maintain** (**cover**) the non-trivial FDs. However, when we normalize relations to 4NF, we want to remove non-trivial MVDs.

❖ MVDs are relation sensitive

Recall that we have 2 MVDs in the relation

CTX (course, teacher, text)

and CTX is not in 4NF.

However, if we add one more attribute, say percentage, to the relation and it becomes

CTX' (course, teacher, text, percentage)

A tuple (c,t,x,p) in the relation CTX' means teacher t teaches course c and p percentages of his material is from text book x. We have the FD:

course, teacher, text $\rightarrow$ percentage

❖ Note that now the two MVDs (in CTX):

course $\twoheadrightarrow$ teacher & course $\twoheadrightarrow$ text

are no longer hold in CTX'. **Q:** Why? Prove it.

The relation CTX' is in 4NF.

This shows MVDs are relation sensitive.

However, we still have course $\twoheadrightarrow$ teacher | text in CTX'.

The Chase Algorithm

- An elegant solution for dependency membership test involving FDs and MVDs.
- Given a set of FDs and MVDs $\mathcal{D}$, does another dependency d (FD or MVD) follow from $\mathcal{D}$ (i.e. d in $\mathcal{D}^+$)?
- **FD Membership Test.** If d is a FD of the form A $\rightarrow$ B, we create a table (i.e. relation) which has all the attributes in $\mathcal{D}$ with 2 tuples which have the same A-value.

  Our objective is to test whether the B-values of these 2 tuples are the same after "applying" the FDs and MVDs in $\mathcal{D}$ to the tuples in the table.

  If yes, then d in $\mathcal{D}^+$ else d is not in $\mathcal{D}^+$.
- **MVD Membership Test.** If d is a MVD of the form A $\rightarrow\rightarrow$ B, we create a table which has all the attributes in $\mathcal{D}$ with 2 tuples which have the same A-value.

  Our objective is to test after applying the FDs and MVDs in $\mathcal{D}$, whether there are 2 new tuples in the table which have the same attribute values of the two original tuples except their B-values are swapped.

  If yes, then d is in $\mathcal{D}^+$ else d is not in $\mathcal{D}^+$.

- **Apply an FD** in $\mathcal{D}$ of the form $X \rightarrow Y$. If there are 2 tuples in the table with same X-value, set their Y-values the same.
- **Apply an MVD** in $\mathcal{D}$ of the form $X \rightarrow\rightarrow Y$. If there are 2 tuples in the able with same X-value, we add 2 new tuples with all the same attribute values except their Y-values are swapped.

Example: Prove that if $A \rightarrow\rightarrow BC$ and $D \rightarrow C$, then $A \rightarrow C$.

In order to prove $A \rightarrow C$, we create 2 tuples in the relation with the same A-value. Our objective is to prove that c1=c2.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2

Since $A \rightarrow\rightarrow BC$, apply the MVD rule, we add 2 tuples into the relation.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2
a	b2	c2	d1
a	b1	c1	d2

A	B	C	D
a	b1	c1	d1
a	b2	c1	d2
a	b2	c1	d1
a	b1	c1	d2

Since $D \rightarrow C$, and the 1st and 3rd tuples have the same D-value, so their C-value should be set to equal, i.e. c1=c2.

So, we have proved that $A \rightarrow C$.

Example: Prove that if A→→B and B→→C, then A→→C in relation R(A,B,C,D).

In order to prove A→→C, we create 2 tuples with same A-value in a relation and then show the 2 tuples (a, b1, c2, d1) and (a, b2, c1, d2) are in the relation.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2

Since A→→B we add 2 tuples.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2
a	b2	c1	d1
a	b1	c2	d2

Since B→→C, we add 2 + 2 tuples.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2
a	b2	c1	d1
a	b1	c2	d2
a	***b1***	***c2***	***d1***
a	b1	c1	d2
a	***b2***	***c1***	***d2***
a	b2	c2	d1

The 2 tuples (***a, b1, c2, d1***) and (***a, b2, c1, d2***) are now in the relation. So we have proved that A→→C

Example (Counter example by chase).

Prove or disprove the statement:

If $A \to\to BC$ and $CD \to B$ then $A \to B$.

In order to prove or disprove $A \to B$, we create 2 tuples with same A-value in a relation and find out whether we can conclude b1=b2.

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2

Since $A \to\to BC$ add 2 tuples

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2
a	b2	c2	d1
a	b1	c1	d2

We cannot further apply the FD: $CD \to B$ to the relation, so the relation remains unchanged.

Since this relation satisfies the two given dependencies but it does not satisfy $A \to B$. This relation is a counter example.

So, the above statement is not true.

Summary on FDs and MVDs in Database Design

- How can we **find FDs** in a RDB? Can we use some data mining techniques to find FDs in a RDB?
- How to **choose the primary key** of a relation? What are the criteria?
- Are there updating anomalies in a BCNF relation?
- If a relation is not in BCNF, can we always normalize it to a set of BCNF relations?
- What are the **normalization criteria** in database schema design?
- Free of **local redundant** attributes is not enough, **global redundancy** may still exist. 3NF and BCNF relations are defined on individual relations, not the whole database, so they may contain global redundant attributes.
- What are the main differences between **decomposition** vs. **synthesizing** methods? What are their weak points?

Summary (cont.)

- How do we **find non-trivial MVDs** in a relation?
- MVDs are relation sensitive.
- If a relation is not in 4NF, then there is a **non-loss decomposition** of R into a set of 4NF relations. However, it may **not cover** all the given FDs.
- When we normalize relations involving onlyFDs, we must maintain (cover) all the non-trivial FDs. However, when we normalize relations to 4NF, we want to **remove non-trivial MVDs**.
- The **Chase** Algorithm for FD/MVD membership test.

Some other normal forms

- Fifth Normal Form (5NF) or called Project-Join Normal Form (PJNF).
- Domain-Key Normal Form (DKNF)
- For your reading pleasure. They will not be covered/examined.

Fifth Normal Form (Project-Join Normal Form)
(5NF, PJNF)

(will not be covered/examined)

There exist relation that cannot be non-loss decomposed into two relations, but can be non-loss decomposed into three or more relations.

Example Let us consider the relation

STOCK(Agent, Company, Product)

We assume that:

1. Agents represent companies.
2. Companies make products.
3. Agents sell products
4. **If an agent sells a product and he represents the company making that product, then he sells that product for that company.**

Note: It is an all key relation. There is no FD or MVD in the relation.

Relation instances:

STOCK (Agent,	Company,	Product)
a_1	c_1	p_1
a_1	c_2	p_1
a_1	c_1	p_3
a_1	c_2	p_4
a_2	c_1	p_1
a_2	c_1	p_2
a_3	c_2	p_4

REP (Agent,	Company)
a_1	c_1
a_1	c_2
a_2	c_1
a_3	c_2

MAKE (Company,	Product)
c_1	p_1
c_1	p_2
c_1	p_3
c_2	p_1
c_2	p_4

SELL (Agent,	Product)
a_1	p_1
a_1	p_3
a_1	p_4
a_2	p_1
a_2	p_2
a_3	p_4

Notes:

(1) There is no FD or MVD in the relation STOCK

(2) The relation is in 4NF.

(3) There are redundant data in the relation.

(4) However, the relation can be non-loss decomposed into 3 relations, namely

REP (<u>Agent, Company</u>)
MAKE (<u>Company, Product</u>)
SELL (<u>Agent, Product</u>)

Q: How do you know this?

(5) REP $\bowtie$ MAKE $\bowtie$ SELL = STOCK

Defn: Let R be a relation and $R_1, \ldots, R_n$ be a decomposition of R. We say that R satisfies the **join dependency** $*\{R_1, R_2, \ldots, R_n\}$ iff

$$\bowtie_{i=1}^{n} R_i = R$$

(or $R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n = R$

or $R_1 * R_2 * \ldots * R_n = R$)

Defn: A join dependency (JD) is **trivial** if one of the R_i is R itself.

Note: When $n = 2$, the join dependency of the form $*\{R_1, R_2\}$ is equivalent to a multivalued dependency.

Example. The relation STOCK(Agent, Company, product) satisfies the join dependency:

$*\{R_1$(Agent, Company), R_2(Agent, Product), R_3(Company, Product)$\}$

However, there is **no MVD** in the relation.

Defn: A relation R is in **fifth normal form** (**5NF**) or called **Project-Join normal form** (**PJNF**) iff every non-trivial join dependency in R is implied by the candidate keys of R.

i.e. whenever a non-trivial join dependency $*\{R_1, R_2, \ldots, R_n\}$ holds in R, implies **every** R_i (all the attributes of R_i) is a superkey for R.

Example: The relation STOCK(Agent, Company, Product) is not in 5NF.

Results: (1) A 5NF relation is in 4NF.

❖ (2) Any relation can be non-loss decomposed into an equivalent collect of 5NF relations, if covering criteria (of FDs) is not required.

Example: The relation Stock can be non-loss decomposed into 3 relations:

REP (Agent, Company)
SELL (Agent, Product)
MAKE (Company, Product)

All are in 5NF.

Domain-Key Normal Form (DKNF)

(will not be covered/examined)

Note that FDs, MVDs and JDs are some sorts of **integrity constraints**. There are other types of constraints:

(1) Domain constraint - which specifies the possible values of some attribute.

E.g. The only colors of cars are blue, white, red, grey.

E.g. The age of a person is between 0 and 150.

(2) Key constraint - which specifies keys of some relation.

Note: All key declarations are FDs but not reverse.

(3) General constraints - any other constraints which can be expressed by the first order logic.

E.g. If the first digit of a bank account is 9, then the balance of the account is greater than 2500.

Defn: Let D, K, G be the set of domain constraints, the set of key constraints, and the set of general constraints of a relation R.

R is said to be in **domain-key normal form** (DKNF) if

$$D \cup K \quad \text{logically implies} \quad G.$$

i.e. all constraints can be expressed by only domain constraints and key constraints.

Example. Let Acct(<u>acct#</u>, balance) with acct# $\rightarrow$ balance and a general constraint:

"if the first digit of an account is 9, then the balance of the account is $\geq$ 2500."

- Relation Acct is not in DKNF.
- To create a DKNF design, we split the relation horizontally into 2 relations:

Regular_Acct (<u>acct#</u>, balance)

Key = {acct#}

Domain constraint: the first digit of acct# is not 9.

Special_Acct (<u>acct#</u>, balance)

Key = {acct#}

Domain constraints:

(1) t he first digit of acct# is 9, and.

(2) balance $\geq$ 2500.

Both relations are in DKNF. **Why?**

All constraints can now be enforced as domain constraints and key constraints.

Q: How to enforce them?

Note: We can rewrite the definitions of PJNF, 4NF, and BCNF in a manner which shows them to be special case of DKNF.

E.g. Let $R=(A_1, \ldots, A_n)$ be a relation.

Let dom(A_i) denote the domain of attribute A_i and let all these domains be infinite.

Then all domain constraints D are of the from

$A_i \subseteq dom(A_i)$.

Let the general constraints be a set G of FDs and MVDs .

Let K be the set of key constraints.

R is in 4NF iff it is in DKNF with respect to D, K, G.

(i.e. every FD and MVD is implied by the domain constraints and key constraints.)

Note: PJNF and BCNF can be rewritten similarly.

Q: How about 3NF?

Theorem

Let R be a relation in which $dom(A)$ is infinite for each attribute A.

If R is in DKNF then it is in PJNF.

Thus if all domains are infinite, then

$$\text{DKNF} \Rightarrow \text{PJNF} \Rightarrow \text{4NF} \Rightarrow \text{BCNF} \Rightarrow \text{3NF}$$