CS 4221: Database Design

The Relational Model

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

$$d_i \in D_i \;\forall\; i = 1, 2, \ldots, n.$$

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

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

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

- 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 which contains information on courses taken by students.

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

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

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

Take	**Student#**	**Course#**	**S-name**	**C-desc**	**Mark**
	95001	CS101	TanCK	Programming	75
	95023	CS101	LeeSL	Programming	58
	94257	CS203	TanCK	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 one time**.

This is denoted by

$$X \rightarrow Y$$

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

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.

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, C-MARK}

- We have the following functional dependencies in Take:

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

Q: How can we find/know these FDs? What is the meaning of the third FD? Why each student only has one name? Are they trivial FDs?

- {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# → C-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. **Q:** What problem? Why?
- **Deletion Anomaly**
- **Rewriting anomaly**

These three anomalies are called the **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 (<u>STUDENT#</u>, S-NAME)
R2 (<u>COURSE#,</u> C-DESCRIPTION)
R3 (<u>STUDENT#, COURSE#,</u> C-MARK**)**

Notation: A contiguous underline indicates a key of the relation.
Attributes STUDENT# and COURSE# form a key of the relation R3.

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

Second Normal Form Relation

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

Note that the relation Take in the previous example is **not** in 2NF.

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

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)

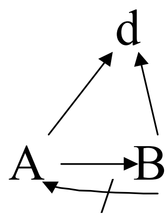
{S#, P#} is the only key of the relation SP.

SP is **not** in 2NF as Sname is not fully dependent on the key.

Redundant information on Sname and Pname in SP. Q: Why?

Third Normal Form 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



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

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.

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

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.

Boyce-Codd normal form (BCNF) Relation

Note: All the three relations:

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

in the previous example are in 3NF. Why?

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 above 3 relations in BCNF?

Question: Are there updating anomalies in a BCNF relation? The answer is still yes but lesser cases. Why?

Example 3.

Consider the relation STJ with the below FDs:

STJ (STUDENT, SUBJECT, TEACHER)

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

   STUDENT, SUBJECT → TEACHER

2. Each teacher teaches only one subject.

   TEACHER → SUBJECT

   SUBJECT ↛ TEACHER

3. Some subjects are taught by several teachers

Questions: What are the keys of the relation SPJ? Primes ? 3NF? BCNF?

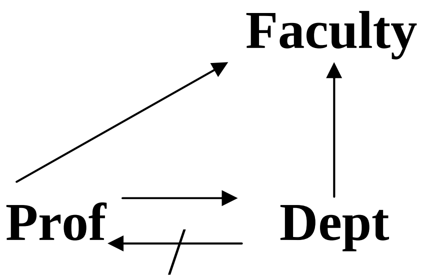
Example 4.

R (<u>Prof</u>, Dept, Faculty)

We have the below FDs: (Why?)

Prof → Dept, Faculty

Dept → Faculty



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

Prof → Faculty

is a transitive dependency.

We decompose this relation to

R1 (<u>Prof</u>, Dept)

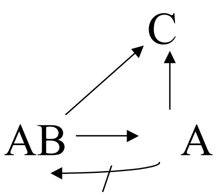
R2 (<u>Dept</u>, Faculty)

They are in 3NF.

Q: Are they in BCNF?

Example 5.

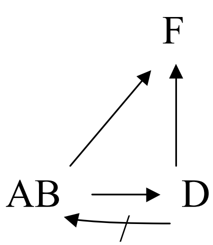
R ($\underline{A, B}$, C, D, F)
with AB $\rightarrow$ CDF, A $\rightarrow$ C, D $\rightarrow$ F



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

  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 (A, B, 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

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

e.g. Enrol (S#, C#, 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 4 & 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 the 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.

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

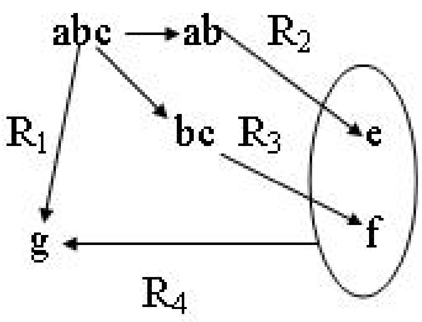
$s\,j \rightarrow t, \qquad t \rightarrow j$

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(<u>a,b,c</u>,g,h), R_2(<u>a,b</u>,e), R_3(<u>b,c</u>,f), R_4(<u>e,f</u>,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.

Properties of FDs

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 $X \rightarrow Y \in \mathbb{F}^+$

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

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

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

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

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

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

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

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

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.

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

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

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 $B_1 \subset B \ni$
$A_1 \to B \in \mathbb{F}^+$ or $B_1 \to A \in \mathbb{F}^+$

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?

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

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

$\mathbb{F}^+ = \{$

$A \rightarrow A$, $B \rightarrow B$, $C \rightarrow C$,
$AB \rightarrow A$, $AB \rightarrow B$, $AB \rightarrow AB$,
$BC \rightarrow B$, $BC \rightarrow C$, $BC \rightarrow BC$,
$AC \rightarrow A$, $AC \rightarrow C$, $AC \rightarrow AC$,
$ABC \rightarrow A$, $ABC \rightarrow B$, $ABC \rightarrow C$,
$ABC \rightarrow AB$, $ABC \rightarrow AC$, $ABC \rightarrow BC$,
$ABC \rightarrow ABC$, /* all the above FDs are trivial

$A \rightarrow B$, $B \rightarrow C$, $A \rightarrow C$, $A \rightarrow BC$,

/* all the below FDs are non full dependencies

$A \rightarrow AC$, $A \rightarrow AB$, $A \rightarrow ABC$, $B \rightarrow BC$,
$AC \rightarrow B$, $AC \rightarrow BC$, $AC \rightarrow AB$, $AC \rightarrow ABC$,
$AB \rightarrow C$, $AB \rightarrow BC$, $AB \rightarrow AC$, $AB \rightarrow ABC$ $\}$

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.

Membership Problem

Given a set of FDs $\mathbb{F}$ defined on $\mathbb{A}$, $X \subseteq \mathbb{A}$ any $y \in \mathbb{A}$,
is $X \rightarrow y \in \mathbb{F}^+$?

i.e. can $X \rightarrow y$ be derived from $\mathbb{F}$?

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

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

Note: The numbers are to indicate a particular FD.

Solution:

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

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

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

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

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

$\because AB \to AB$ (by projectivity)

$AB \to C$ (given)

$\therefore AB \to ABC$ (by additivity)

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

$\because C \to D$ (given)

$ABC \to C$ (by projectivity)

$\therefore ABC \to D$ (by transitivity)

Also $ABC \to ABC$ (by projectivity)

$\therefore ABC \to ABCD$ (by additivity)

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

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

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

$\therefore AB \to ABCD$ (by transitivity)

(4) …

Alternative Solution to show $X \rightarrow Y$ 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: 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.

$$\{A\ B\}^+ \overset{1}{=} \{A\ B\ C\}^+ \overset{2}{=} \{A\ B\ C\ D\}^+$$
$$\overset{4}{=} \{A\ B\ C\ D\ E\}^+ \overset{3}{=} \{A\ B\ C\ D\ E\ F\}^+$$
$$= \{A\ B\ C\ D\ E\ F\}$$
$$\therefore\ AB \rightarrow F \in G^+$$

Criteria for Normalization

(1) Reconstructibility (or losslessness)

If an original relation R is split into $R_1, R_2, \ldots, R_n$,

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** iff $\mathbb{F}^+ = \mathbb{G}^+$

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 (ie no local redundancy).

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 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

Example 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)

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 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.

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

Result: The number of relations produced is minimum.

Example 1 Given $\mathbb{F} = \{A \to B, A \to C, B \to C, B \to D, D \to B, ABE \to F\}$

Step 1. (Eliminating extraneous attributes)

$$\mathbb{G} = \{A \to B, A \to C, B \to C, B \to D, D \to B, AE \to F\}$$

(since $AE \to ABE \in \mathbb{F}^+$)

Step 2. (Find covering)

$$\mathbb{H} = \{A \to B, \quad B \to C, B \to D, D \to B, AE \to F\}$$

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

Step 3 (Partition)

$$H_1 = \{A \to B\}$$
$$H_2 = \{B \to C, B \to D\}$$
$$H_3 = \{D \to B\}$$
$$H_4 = \{AE \to F\}$$

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!

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

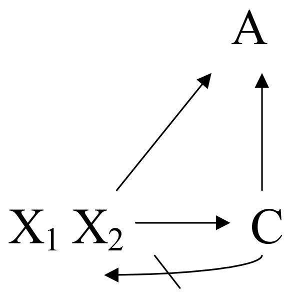
Step 4
$$J = \{X_1 X_2 \rightarrow CD,\ CD \rightarrow X_1 X_2\}$$
$$H'_1 = H_1 \cup H_2 - J$$
$$= \{X_1 X_2 \rightarrow A\}$$
$$H_3 = \{A X_1 \rightarrow B\}$$
$$H_4 = \{B X_2 \rightarrow C\}$$
$$H_5 = \{C \rightarrow A\}$$

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

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

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

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

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?

Note 1. Bernstein's algorithm does not guarantee reconstructibility (or losslessness)

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

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

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

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

:

:

Step 6 R_1 (Course#, Cname, Cdesc)

Note. We lose information about Preq#.

In fact

Course# →→ Preq#

We need another relation

R_2 (Course#, Preq#)

Note 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 (A, B, C, D)

R_2 (C, 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.

Note 3. Bernstein's algorithm does not remove all the superfluous attributes.

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

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.

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

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

Note 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

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

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

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

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

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

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

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

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

Then we have

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

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

R_3 ($\underline{C,}$ D

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

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

Then we have

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

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

R_3 ($\underline{C,}$ 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?

Note: BCNF relations may contain **superfluous attributes**, i.e. redundant attributes which can be removed.

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)

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?

Notes.

- 3NF and BCNF are defined for **individual relations** but not the **whole relational schema**.
- Ling, Tompa, & Kameda method takes the **whole relational schema** into consideration and removes superfluous attributes.

Fourth Normal Form Relation (4NF)

E.g. The meaning of a given record in the below unnormalized relation is:

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

Unnormalized 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**)

$$R.A \twoheadrightarrow R.B$$

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(X, Y, Z) be a relation and X, Y, Z be sets of attributes of R, not necessarily disjoint.

Let $Y_{x\,z} = \{ y \mid (x, y, z) \in R \}$ /* x and z are some X and Z values

The **MVD** $\mathbf{X} \twoheadrightarrow \mathbf{Y}$ is said to hold for R(X, Y, Z) if and only if $Y_{x\,z}$ depends on X

i.e. $Y_{x\,z} = Y_{x\,z'}$ for all x, z, z′ values of attributes X and Z, such that $Y_{x\,z}$ and $Y_{x\,z'}$ are **non-empty**.

- We sometime use

  $X \twoheadrightarrow Y \mid Z$

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

  Course $\twoheadrightarrow$ Teacher

  Course $\twoheadrightarrow$ Text

  i.e. Course $\twoheadrightarrow$ Teacher | Text

Q: What is the intuitive meaning?

Notes. (1) $X \twoheadrightarrow \varnothing$ 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**.

(Recall: A functional dependency $X \rightarrow Y$ is said to be trivial if $Y \subseteq X$).
Note: $\varnothing$ is the symbol for the empty set.

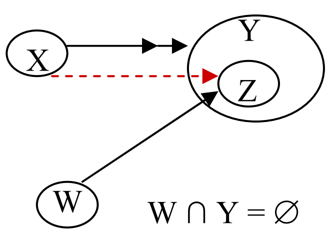
Defn. A relation R is in **fourth normal form** (**4NF**) if and only if any nontrivial MVD $X \twoheadrightarrow Y$ holds in R implies X is a **superkey** of R.
i.e. $X \rightarrow a$ for all attribute a of R.

(Recall: A relation R is in BCNF iff any nontrivial FD $X \rightarrow Y$ holds in R implies $X \rightarrow a$ for all attribute a of R)

Inference Rules for Multivalued Dependencies

Let R be a relation with attribute set A.

1. (**Complementation**)
   If $X \twoheadrightarrow Y$ then $X \twoheadrightarrow A - X - Y$
2. (**Augmentation**)
   If $X \twoheadrightarrow Y$ and $V \subseteq W$
   then $WX \twoheadrightarrow VY$
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).

Corollary. If a relation is not in 4NF, then there is a nonloss 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)

However the resulting relations do not cover the FD:

SJ $\rightarrow$ T

E.g. The relation CTX(course, teacher, text) is in BCNF but not in 4NF

course $\twoheadrightarrow$ teacher | text

i.e. course $\twoheadrightarrow$ teacher

and course $\twoheadrightarrow$ text

We can decompose the relation into 2 relations

CT(course, teacher)

and

CX(course, text)

Both relations are in 4NF.

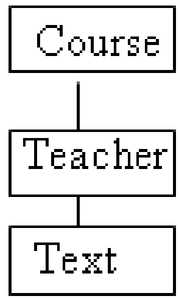
Note that the MVD

course $\twoheadrightarrow$ teacher | text

does not exist in CT or CX.

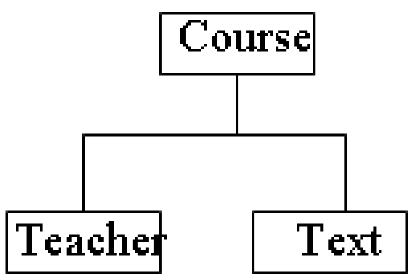
Intuitive meaning of the MVD: The text books of a course are independent on who are the teachers of the course (may be the textbooks of a course are decided by the curriculum committee).

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



It is a wrong design in hierarchical model

Below is a a correct design:



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

employee $\twoheadrightarrow$ child

employee $\twoheadrightarrow$ {salary, year}

(**Q:** How do we know these 2 MVDs?)

We can decompose R into

R_1 (employee, child)

and

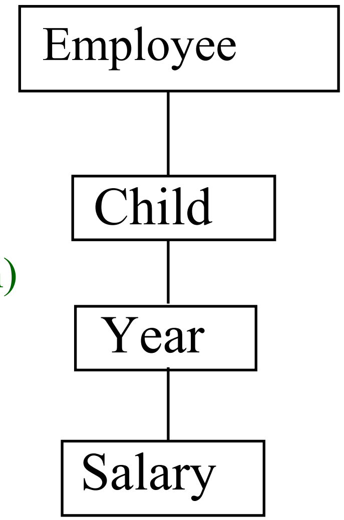
R_2 (employee, salary, year)

Both relations are in 4NF.

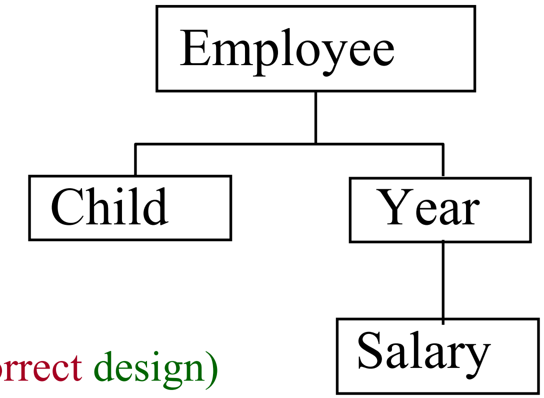
Note that an employee can have more than one salary adjustment within one year.

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

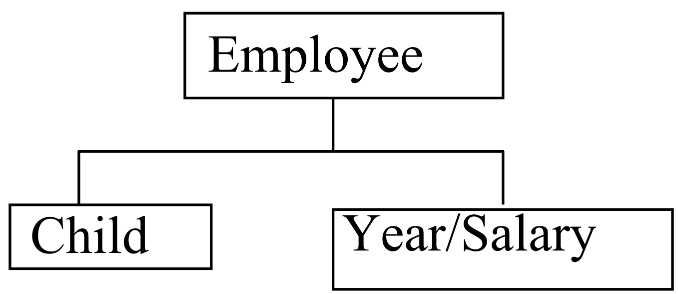
3 possible hierarchical database designs of the relation R:



(wrong design)



(correct design)



(another correct design)

More Properties of MVDs

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

Note. A binary relation is definitely in 3NF but not necessarily in 4NF. How about 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.

Question: 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 $B \twoheadrightarrow HI$

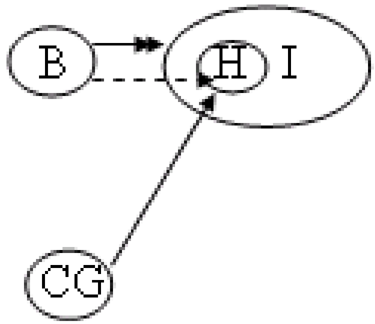
$H \subseteq HI$

$CG \rightarrow H$

$CG \cap HI = \varnothing$

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 Ri ∈ result that is not in 4NF)
    THEN BEGIN
        LET  X ->> Y   be a nontrivial MVD that holds on R_i such that
                X → R_i ∉ D+;
          /* need to decompose the relation R_i into 2 smaller relations
        SET  result := (result – R_i) ∪ (R_i – Y) ∪ (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?

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?

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 (teacher, subject)

R_2 (teacher, student)

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"?

Fifth Normal Form (Project-Join Normal Form)

There exist relation that cannot be nonloss-decomposed into two projections, but can be nonloss-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**.

Relation instances:

STOCK (Agent, Company, Product)

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)

Agent	Company
a_1	c_1
a_1	c_2
a_2	c_1
a_3	c_2

MAKE (Company, Product)

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)

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) The relation can be nonloss-decomposed into 3 relations, namely

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

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

Defn: Let R be a relation and $R_1, \ldots, R_n$ be a decompsition 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 nontrivial join dependency in R is implied by the candidate keys of R.

i.e. whenever a nontrivial 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 nonloss-decomposed into an equivalent collect of 5NF relations, if covering criteria (of FDs) is not required.

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

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

All are in 5NF.

Domain-Key Normal Form

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

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

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

Example. Let Acct(acct#, balance) with acct# → balance and a general constraint:

" if the first digit of an account is 9, then the balance of the account is ≥ 2500."

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

Regular_Acct (acct#, balance)

Key = {acct#}

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

Special_Acct (acct#, balance)

Key = {acct#}

Domain constraints:

(1) balance ≥ 2500

(2) the first digit of acct# is 9.

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

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

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.

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

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