NORMALIZATION: WHY

DO NOT REMOVE
FROM LIBRARY

LIBRARY USE ONLY
NORMALIZATION

Why?

Three types of misbehavior
- UPDATE
- Insertion
- Delete

Each relation should describe a single concept

A relation is in third normal form if every determinant is key

\[ A \rightarrow B \quad \text{determines} \]
\[ \quad \text{determinant} \]
### Normalization

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
<th>Date</th>
<th>Qty ordered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toaster</td>
<td>20.00</td>
<td>1/10/75</td>
<td>2</td>
</tr>
<tr>
<td>Toaster</td>
<td>20.00</td>
<td>2/15/75</td>
<td>5</td>
</tr>
<tr>
<td>Mixer</td>
<td>28.00</td>
<td>4/6/75</td>
<td>3</td>
</tr>
</tbody>
</table>

- Insertion Problem
- Deletion Problem
- Lack of info.
- Loss of info.
IF \( A \rightarrow B \)  
\( B \rightarrow C \)  
THEN \( A \rightarrow C \)  

2. IF \( A \rightarrow B \)  
THEN \( AB \rightarrow B \)  

\[
R(AB, c, D, E) \quad A \rightarrow D  
A \rightarrow E  
AB \rightarrow C  
\]

Let \( X, Y, Z \) be subset of all attributes of a relation \( R \).

**Axioms**

1. **Reflexivity**  
   IF \( Y \subseteq X \subseteq U \) \n   Then \( X \rightarrow Y \)  

2. **Augmentation**  
   IF \( X \rightarrow Y \) and \( Z \subseteq U \)  
   Then \( XZ \rightarrow YZ \)  

3. **Transitivity**  
   IF \( X \rightarrow Y \) and \( Y \rightarrow Z \)  
   Then \( X \rightarrow Z \)
ARMSTRONG'S AXIOMS ARE
SOUND and COMPLETE.

**SOUND:**

If $X \rightarrow Y$ is deduced from $F$ and Axiom $F$ is true,
then $X \rightarrow Y$ is true in any relation in which $F$ is true.

**REFLEXIVITY:** If two tuples of $Y$
agree on $X$, they must agree on
a subset of $X$.

**AUGMENTATION:**

If two tuples agree on $X \ast$
but not on $Y \ast$
then they must agree on $X$ but not on $Y$.

**CONTRACTION** (\(X \rightarrow Y\))

**TRANSITIVITY:**

If two tuples agree on $X$, they
agree on $Y$. If they agree on $Y$, they
agree on $Z$. So $X \rightarrow Z$. 
Functional Dependency (Revisited)

\[ X \rightarrow Y \]

means

\( X \) functionally determines \( Y \) or \( Y \) is functionally dependent on \( X \)

if it is not possible that relation \( R \) has two tuples that agree on value of \( X \)

and disagree on value of \( Y \)

Many-to-one mapping

\[ X \rightarrow Y \]

one-to-one mapping

\[ X \rightarrow Y \]

\[ i \rightarrow x \]
Some more rules

Union

IF $X \rightarrow Y$, $X \rightarrow Z$

THEN $X \rightarrow YZ$

Pseudotransitivity

IF $X \rightarrow Y$

and $WY \rightarrow Z$

Then $XW \rightarrow Z$

Decomposition

IF

$X \rightarrow Y$ and $Z \subseteq Y$

Then $X \rightarrow Z$

IF $F$ is a set of functional dependencies

$F^+$ is the set of all functional dependencies derivable from $F$

$F^+$ is called closure of $F$. 
Proof

Union Rule

\[ x \rightarrow y \Rightarrow x \rightarrow xy \]
\[ x \rightarrow z \Rightarrow xy \rightarrow zy \text{ (or } yz \text{)} \]

So \[ x \rightarrow xy \rightarrow yz \]

\[ \text{QED} \]

Pseudotransitivity Rule

\[ x \rightarrow y \Rightarrow xw \rightarrow yw \]

But \[ yw \rightarrow z \]

So \[ xw \rightarrow z \]

\[ \text{QED} \]

Decomposition Rule

\[ x \rightarrow y \]

Tuples that agree on \( x \) do agree on \( x \) and so they do agree on \( \text{Subset of } y \)

But \[ z \subseteq y \]

So \[ x \rightarrow z \]

\[ \text{QED.} \]
Example

\[ R = (A, B, C, D, E, G) \]

\[ F : \]

\[ AB \rightarrow C \quad D \rightarrow EG \]
\[ C \rightarrow A \quad BE \rightarrow C \]
\[ BC \rightarrow D \quad CG \rightarrow BD \]
\[ ACD \rightarrow B \quad CE \rightarrow AG \]

\[ (BD)^+ = \text{Set of attributes that are dependent on attributes } B, D \]

\[ = ABCDEG = R \]

Thus BD determines R
We call BD as the key of R
A relation by definition is in FIRST NORMAL FORM

- A relation is in 2nd NF if any one of the following is true

  1. The key consists of a single attribute
     2. There are no non-key attributes
     3. Every non-key attribute depends on all of the key

Example

\[ R(A, B) \]  \hspace{2cm} \text{Case 1}
\[ R(A, B, C) \]  \hspace{2cm} \text{Case 2}
\[ R(A, B, C, D) \]  \hspace{2cm} \text{Case 3}

\[ \text{and } AB \rightarrow C \]
\[ AB \rightarrow D \]

- A relation is in 3NF, if it is in 2nd NF and has no transitive dependencies
Mathematically

R is in 3NF

if \( \exists \) key X for R and Y \( \subseteq \) R

and a non-key attribute A not in X or Y

Such that

1. \( X \rightarrow Y \)
   \[ X \quad \underline{A} \]

2. \( Y \rightarrow A \) \( Y \notin X \)
   \[ X \quad \underline{A} \quad Y \]

3. \( Y \nrightarrow X \) \( Y \subseteq X \)
   \[ \underline{X} \quad \underline{Y} \quad \underline{A} \]

If Y is a subset of X

Then R has a partial dependency

If Y is not a subset of X

Then R has a transitive dependency
A set of $F$ is minimal if

a) Every right side is a single attribute

b) For no $x \rightarrow A$, $F - \{x \rightarrow A\} \neq F$

c) For no $x \rightarrow A$, $\exists c x$

\[
[F - \{x \rightarrow A\}] U \{z \rightarrow A\} = F
\]

**Lemma:**

$F$ is covered by $G$, in which no right side has more than one attribute.

If $x \rightarrow A$ in $G$ and $x \rightarrow Y$ in $F$ and $A \in \{x \rightarrow A\}$

By decomposition $x \rightarrow A$ in $F^+$

So $G \subseteq F^+$

Let $Y = A_1 \cdot A_2 \cdots A_n$

If $x \rightarrow Y$ in $F$, then $x \rightarrow A_1$ in $G$.

$x \rightarrow A_2$

$\vdots$

$x \rightarrow A_n$

So $F \subseteq G^+$

So $F^+ = G^+$
Lossless Join Decomposition

Let \( \gamma \) be a relation for scheme \( R \) satisfying dependencies \( D \).

Let \( P = \{ R_1, \ldots, R_n \} \) be a decomposition satisfying \( D \).

Then the decomposition is lossless if

\[
\gamma = \Pi_{R_1}(\gamma) \times \Pi_{R_2}(\gamma) \times \cdots \times \Pi_{R_n}(\gamma)
\]

= Natural Join of its projections on the \( R_i \)'s.

Let \( m_\rho(\gamma) = \Pi_{i=1}^{n} \Pi_{R_i}(\gamma) \)

Decomp is lossless if

\[
\gamma = m_\rho(\gamma)
\]

Let \( \gamma_i = \Pi_{R_i}(\gamma) \)
Lemma

a) \( \gamma \subseteq m\rho(\gamma) \)

b) If \( S = m\rho(\gamma) \), then \( \Pi_{R_\gamma}(S) = \gamma \)

c) \( m\rho(m\rho(\gamma)) = m\rho(\gamma) \)

Testing Lossless Joins

\( R = A_1 \ldots A_n \quad P = (R_1, R_2, \ldots R_k) \)

\[ \begin{array}{c|c|cccc}
\hline
 & A_1 & A_2 & A_j & A_n \\
\hline
R_1 & & & & & \\
R_2 & & & & & \\
& & & \cdots & & \\
R_k & & & & & \\
\hline
\end{array} \]

\[ \begin{array}{c|c|cccc}
\hline
x & a_j & b_j & & \\
\hline
x \neq a_j \text{ if } A_j \in R_2 & & & & \\
x = b_j \text{ if } A_j \in R_2 & & & & \\
\hline
\end{array} \]

Do Recursively:

- If one row is all a's, HALT
- If one row is all b's, HALT

let \( x \rightarrow y \)

look 2 rows that agree on \( x \)

Equate elements of \( y \) if one is c

both a or b

if both are a, leave the as a

if both are b, leave the as b.
\[ R = (A, B, C, D, E) \]

\[
\begin{align*}
R_1 &= AD \\
R_2 &= AB \\
R_3 &= BE \\
R_4 &= CDE \\
R_5 &= AE
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>a_1</td>
<td>b_{12}</td>
<td>b_{13}</td>
<td>a_4</td>
<td>b_{15}</td>
</tr>
<tr>
<td>AB</td>
<td>a_1</td>
<td>a_2</td>
<td>b_{43}</td>
<td>\ell_{25}</td>
<td>b_{25}</td>
</tr>
<tr>
<td>BE</td>
<td>b_{51}</td>
<td>a_2</td>
<td>b_{53}</td>
<td>b_{54}</td>
<td>a_5</td>
</tr>
<tr>
<td>CDE</td>
<td>b_{41}</td>
<td>b_{42}</td>
<td>a_3</td>
<td>a_4</td>
<td>a_5</td>
</tr>
<tr>
<td>AE</td>
<td>a_1</td>
<td>b_{52}</td>
<td>b_{53}</td>
<td>b_{54}</td>
<td>a_5</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
A &\rightarrow C \\
B &\rightarrow C \\
C &\rightarrow D \\
DE &\rightarrow C \\
CE &\rightarrow A
\end{align*}
\]
Proof: Algorithm correctly determines if a decomposition has a lossless join.

One way

Suppose the final table does not have a row with all a's. Let this be a relation \( \gamma \) for \( R \).

Then we must prove \( \gamma \neq m_p(\gamma) \).

Now for each \( R_i \), \( \exists t_i \in \gamma \) with all a's in its row.

\[
mp(\gamma) = \prod_{i=1}^{K} R_i(\gamma) \text{ contains a row with all a's}
\]

So \( \gamma \) with no rows of a \neq mp(\gamma) with a row of all a's.

Reverse

Please read yourself.
Superkey - superset of a key
Candidate key - minimal set of attributes
key - one designated Candidate key

\[ R \left( \text{city}, \text{st}, \text{zip} \right) \]
\[ \text{city}, \text{st} \rightarrow \text{zip} \]
\[ \text{zip} \rightarrow \text{city} \]
\[ \left( \text{city}, \text{st} \right), \left( \text{st}, \text{zip} \right) \text{ are keys} \]

An Att is primary att of \( R \)
if it is a member of any key \( Y \) of \( R \)
Non primary att = not a member of any key \( Y \) of \( R \)
3NF

X is a superkey of R

or A is a prime attr of R

3 Arel is 3NF

if every non-prime attr of R

is

1) Fully functionally dep

on every key of R

2) Non-transitively dep

on every key of R

A rel is 2NF

if every non-prime attr A

in R is not partially dep

on any key of R
Boyce-Codd Normal Form

A rel R with attr F in
in BCNF if

\[ X \rightarrow A \text{ holds in } R \]

and \( A \notin X \)

\[ X \text{ is a superkey of } R \]

\[ X \text{ is or contains the key.} \]

Diff between 3NF and BCNF

3NF allows \( A \) to be prime
if \( X \) is not a superkey.

NP Complete to determine
if \( A \) is in 3NF.
Theorem

Every set of dependencies $F$ is equivalent to a set of dependencies $F'$ that is minimal.

Proof by construction

Step 1
Change $F$ to get single attribute on right side

Step 2
If a dependency $x \rightarrow y$ can be eliminated without changing $F'$, do it.
(you may have several choices)

Step 3
Eliminate attributes from the left side.

$XY \rightarrow z \quad \Rightarrow \quad \text{Eliminate } X$

$Y \rightarrow z$
Theorem

If a relation R is in BCNF
Then it is in 3NF.

Proof
Let R in BCNF and not in 3NF

Then X \rightarrow Y \rightarrow A is in F(Partial or Transitive)
X is a key for R
A \in X or A \in Y and Y \not\rightarrow X

If Y \not\rightarrow X
Then Y does not include the key for R
But Y \rightarrow A violates that R in BCNF

4th NF, Multivalued Dependencies.
**Theorem:**

If \( P = (R_1, R_2) \)

Then \( P \) has a lossless join w.r.t. \( F \)

iff

\[ R_1 \cap R_2 \rightarrow R_1 - R_2 \]

or \( R_1 \cap R_2 \rightarrow R_2 - R_1 \)

\[ \in F^+ \]

**Example**

\( R = (A, B, C) \)

\( F = \{ A \rightarrow B \} \)

\( R_1 (A, B) \)

\( R_2 (B, C) \)

\( R_1 \cap R_2 = B \)

\( R_1 - R_2 = A \)

\( R_2 - R_1 = C \)

\( B \rightarrow A, B \rightarrow C \)

So decomposition is Lossy

\( R_1 (A, B) \)

\( R_2 (A, C) \)

\( R_1 \cap R_2 = A \)

\( R_1 - R_2 = B \)

\( A \rightarrow B \)

\( R_1 \cap R_2 \rightarrow R_1 - R_2 \)

Decomposition has a lossless join
Algorithm for Lossless Join Decomposition into BCNF.

Initially \( P = R. \)

For \( S \in P \) if \( S \) not in BCNF

then \( X \rightarrow A \) holds in \( S \)

\( \exists \) \( X \) does not include a key for \( S \)

and \( A \notin X \)

let attribute \( AK \in S \)

\( \& A \notin X \)

Then \( S = (S_1, S_2) \)

\( \exists \) \( S_1 = (X, A) \)

\( S_2 = (S - A) \neq \emptyset \) \( \because \) contains \( AK \)

Decomposition of \( S \) is \( (S_1, S_2) \)

Keep Iterating till all \( S_i \) in BCNF.
Lemma

\[ P = (R_1, \ldots, R_i, \ldots, R_k) \]

Decomposition (lossless join)

\[ S_1, S_2, \ldots, S_m \]

If \( P \) has a lossless join w.r.t. \( F \)

\[ P_1 = (R_1, \ldots, R_i, S_1, S_2, \ldots, S_m, R_{i+1}, \ldots, R_k) \]

\[ P_1 \text{ has a lossless join w.r.t. } F \]

\[ P_2 = (R_1, \ldots, R_i, \ldots, R_k, R_{k+1}, \ldots, R_n) \]

\( P_2 \) include \( P \) and some more

then \( P_2 \) also has a lossless join w.r.t. \( F \)
Decompositions that Preserve Dependence

Projection of $F$ onto $Z(\pi_z(F))$ is set of dependencies $X \rightarrow Y$ in $F^+$ such that $XY \subseteq Z$.

A decomposition $P$ preserves a set of dependencies $F$ if

$$\bigcup_{i=1}^{R} \Pi_{R_i}(F) \subseteq F$$

$$\left( \bigcup_{i=1}^{R} \Pi_{R_i}(F) \right)^+ = F^+$$
Relation Schema

- Lossless Join Decomposition
- B-C NF

\[ R(C, S, Z) \]

\[ R_1(S, Z) \quad C, S \rightarrow Z \]

\[ R_2(C, Z) \quad Z \rightarrow C \]

Is it a lossless join Decomposition?

Is it a FD preserving Decomposition?
\[ R(\{A, B, C, D\}) \]
\[ P = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A \} \]

\[ R_1(\{A, B\}) \]
\[ F_1: A \rightarrow B \]
\[ \Pi_{R_1}(F^+) = F_1. \]

\[ \Pi_{R_1}(F^+) = F_1. \]

\[ R(\{A, B, C, D\}), P(\{A, B, C, D\}) \]

\[ F: A \rightarrow B, C \rightarrow D. \]

Preserves Functional Dependencies, but yet a lossy join.
\[ R_1 = (C, T, H, R, S, G) \]

\[ C \rightarrow T \]
\[ HR \rightarrow C \]
\[ HT \rightarrow R \]
\[ CS \rightarrow G \]
\[ HS \rightarrow R \]

\[ (HS)^+ = \]

Now \( CS \rightarrow G \) violates that \( R_1 \) in \( BC \).

To break \( CS \rightarrow THR \), project \( F^+ \) on \( C, S, T, H, R \):

\[ C \rightarrow T \]
\[ HR \rightarrow C \]
\[ HT \rightarrow R \]
\[ HS \rightarrow R \]
What is the key for C$^*$THR, HS
C$\rightarrow$T violates that C$^*$THR in BCNF

C$^*$THR

CT$\rightarrow$C$^*$THR

CH$\rightarrow$R
HS$\rightarrow$R
HR$\rightarrow$C

Key HS

CH$\rightarrow$R violates C$^*$THR in BCNF

C$^*$THR

CHR$\rightarrow$CHS

R$_1$ = \{CGS, CT, CHR, CHS\} is in BC

Exponential Complexity

To test if BCNF is NP-COMPLETE.
CE $\rightarrow$ A eliminated because C $\rightarrow$ A

✓ ACD $\rightarrow$ B reduced to CD $\rightarrow$ B

✓ CG $\rightarrow$ D, C $\rightarrow$ A, ACD $\rightarrow$ B

\[ \Downarrow \]

CG $\rightarrow$ CD

\[ \Downarrow \]

CG $\rightarrow$ B

---

**DECOMPOSITION OF RELATION SCHEME**

\[ R = \{ A_1, A_2, \ldots, A_n \} \]

\[ \Downarrow \text{Decompose} \]

\[ P = \{ R_1, R_2, \ldots, R_k \} \exists R_1 \cup R_2 \cup \ldots \cup R_k = R \]

**Example**

\[ R = \{ S, A_1, P \} \]

\[ R_1 = \{ S, A_1 \} \]

\[ R_2 = \{ S, 1, P \} \]
R

| Lossless join
| P = (R₁, R₂, ..., Rₖ)
| Each Rᵢ in BCNF
| ₖ ≥ 50 in 3NF
| Preserve set of dependencies
| \( τ = (S₁, S₂, ..., Sₘ) \)

Let \( τ = σ \cup \{x \mid x \in X \} \)

where \( x \) is the key for \( R \)

All relations in \( τ \) are in 3NF

Decomposition preserves dependencies

And has a lossless Join.

\[ Yᵢ \leq X \cup \{R - X \} \]

\[ Yᵢ \rightarrow Aᵢ \]
Alg 54: Dependency Preserving Decomp.
into 3NF

- If any attribute not in $F$
  it is one decomposition

- If any dependencies contains all attributes of $R$,
  $R$ is one decomp.

- Otherwise

  Decomposition is $X A$
  where $X \rightarrow A$ in $F$

\[ R = \{ C, T, H, R, S, G \} \]

Minimal cover

- $C \rightarrow T$
- $HR \rightarrow C$
- $HT \rightarrow R$
- $CS \rightarrow C$
- $HS \rightarrow R$

\( \Gamma \subseteq \{ CT, HRC, HTR, CSG, HSR \} \)