

Lossless-Join Decomposition:

(Recap) Def: For the case of $R = (R_1, R_2)$, we require that for all possible relations r on Schema R

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r)$$

A decomposition of R into R_1 and R_2 is lossless join if at least one of the following dependencies is in F^+ :

- (i) $R_1 \cap R_2 \rightarrow R_1$
 - (ii) $R_1 \cap R_2 \rightarrow R_2$
- } sufficient condition

* Identification of lossy/lossless join

The following conditions must hold.

- (i) $R_1 \cup R_2 = R$
- (ii) $R_1 \cap R_2 \neq \emptyset$ and
- (iii) $R_1 \cap R_2 \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$

Example: Supplier-Parts (S-NO, S-Name, City, P-No, Qty)

fd: $S-NO \rightarrow S-Name$,

$S-NO \rightarrow City$

$S-NO, P-No \rightarrow Qty$

Decomposition: Supplier (S-NO, S-Name, City, Qty)
Parts (P-No, Qty)

Determine lossy/lossless join.

Soln: we have to determine Supplier \bowtie Parts

Supplier Parts

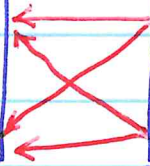
S-NO	S-Name	City	P-NO	Qty
3	Smith	London	301	20
5	Nick	NY	500	50
2	Steve	Boston	20	10
5	Nick	NY	400	40
5	Nick	NY	301	10

Supplier

S-NO	S-Name	City	<u>Qty</u>
3	Smith	London	20
5	Nick	NY	50
2	Steve	Boston	10
5	Nick	NY	40
5	Nick	NY	10

Parts

P-NO	<u>Qty</u>
301	20
500	50
20	10
400	40
301	10



Supplier \bowtie Parts

(i) holds	S-NO	S-Name	City	P-NO	Qty
(ii) holds	3	Smith	London	301	20
	5	Nick	NY	500	50
	5	Nick	NY	20	10 *
	2	Steve	Boston	20	10
	5	Nick	NY	400	40
	5	Nick	NY	301	10
	2	Steve	Boston	301	10 *

[Lossy Join]

We get extra tuples

Attributes common to both R_1 & R_2 : i.e. $R_1 \cap R_2 =$
Qty.

Here none of the f'ds can determine Parts or Supplier given Qty.

\therefore (iii) does not hold.

Try yourself:

Example: Given Supplier-Parts (S-NO, S-Name, City, P-NO, Qty) ^R
 fds: $\{S-NO \rightarrow S-Name, S-NO \rightarrow City, (S-NO, P-NO) \rightarrow Qty\}$

Decomposition: Supplier (S-NO, S-Name, City) ^{R₁}
 Parts (S-NO, P-NO, Qty) ^{R₂}

Determine if lossy/lossless join.

Soln:

Check for:

(i) does $R_1 \cup R_2 = R$

i.e. Supplier \cup Parts = Supplier-Parts
 \rightarrow holds true.

(ii) does $R_1 \cap R_2 \neq \phi$

i.e. Supplier \cap Parts = S-NO
 \rightarrow holds true.

(iii) does $R_1 \cap R_2 \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$

\rightarrow holds true using
 i.e. $R_1 \cap R_2 = S-NO$

using fds: $\left. \begin{array}{l} S-NO \rightarrow S-Name \\ S-NO \rightarrow City \end{array} \right\}$ covers Supplier

\therefore (iii) holds true

Conclude: Lossless join and
 decomposition holds/is valid as it
 preserves all fd's.
 S-NO is the super key.

2NF and Examples: {NOTE: R should be in 1NF}

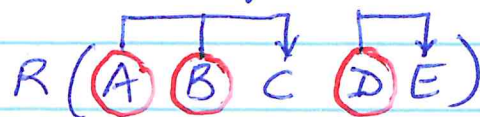
Objective: a) determining if a relation is in 2NF
 b) if it is not in 2NF how decompose the relation such that it is in 2NF.

Example 1: Given $R(AB C D E)$ with fd's $AB \rightarrow C, D \rightarrow E$ determine if it is in 2NF; if not then decompose.

solution: {search for partial dependencies}

Given $R(AB C D E)$ $\{AB \rightarrow C, D \rightarrow E\}$

step 1: use the edge diagram to identify the candidate key.



∴ potential candidate key = ABD

$ABD^+ = ABDCE$ ∴ ABD is the candidate key.

step 2: identify partial dependencies from fd's.

$AB \rightarrow C$

$D \rightarrow E$

∴ if a fd's determinant (i.e. X) contains a subset of the candidate key; then that fd is a partial dependency.

∴ $AB \rightarrow C$ is a partial depen.

$D \rightarrow E$ is a partial depen.

(5)

$\therefore R(ABCDE)$ is not in 2NF.

Converting/Decompose R in 2NF.

$R(ABCDE)$

$\rightarrow R_1(ABC)$ based on partial dep $AB \twoheadrightarrow C$
 $\rightarrow R_2(DE)$ based on pd $D \rightarrow E$
 $\rightarrow R_3(ABD)$ the candidate key; to ensure lossless joins.

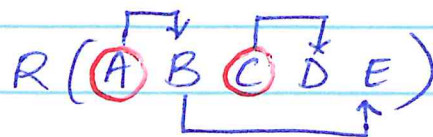
Example 2: $R(ABCDE)$

$A \rightarrow B$

$B \rightarrow E$

$C \rightarrow D$ State (True/False) R is in 2NF

soln:



$AC^+ = ACBED \therefore AC$ is the candidate key.

Given AC^+ :

$A \rightarrow B$ \rightarrow {this is a partial dep.}

$B \rightarrow E$

$C \rightarrow D$ \rightarrow {this is a partial dep.}

$\therefore R$ is not in 2NF. [False]

$R(ABCDE)$

$\rightarrow R_1(ABE)$
 $\rightarrow R_2(CD)$
 $\rightarrow R_3(AC)$

Decomposition is in 2NF

Example 3: $R(A B C D E F G H I J)$

$AB \rightarrow C$

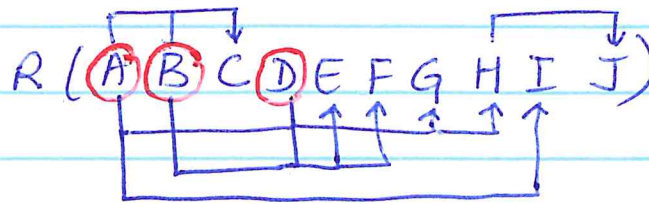
$AD \rightarrow GH$

$BD \rightarrow EF$

$A \rightarrow I$

$H \rightarrow J$

soln:



ABD^+ : $ABDCGH E F I J$ ✓

$AB \rightarrow C$
 $AD \rightarrow GH$
 $BD \rightarrow EF$
 $A \rightarrow I$

} are partial dep. in the fds.

decomposition:

$R(A B C D E F G H I J)$

$\rightarrow R_1(A B C)$

$\rightarrow R_2(A D G H, J)$

$\rightarrow R_3(B D E F)$

$\rightarrow R_4(A I)$

$\rightarrow R_5(A B D)$

Note fd $H \rightarrow J$ is included here.

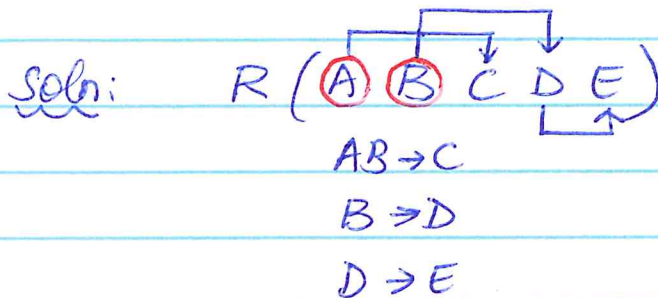
R_1, R_2, R_3, R_4 & R_5 are in 2NF.

3NF Examples: {Note R should be in 2NF}
 {get rid of transitive dependencies}

Objective: a) to determine if R is in 3NF

b) to decompose such that Relations are in 3NF

Example 1: Given $R(ABCDE)$ fd's: $\{AB \rightarrow C, B \rightarrow D, D \rightarrow E\}$
 determine if R is in 3NF and decompose.



Step 1: the candidate key is: AB as $AB^+ = ABCDE$

Step 2: traverse through the fd's to identify fd's that are partially dependant on the candidate key AB.

$AB \rightarrow C$

$B \rightarrow D$

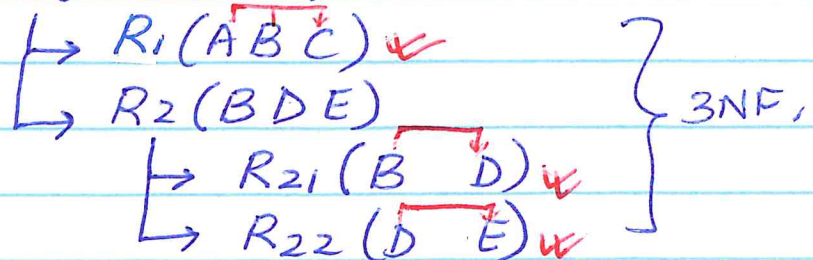
$D \rightarrow E$

{ this fd is partially dep. }

{ this fd is a transitive dep. }

So conclude that R is not in 3NF.

Step 3: Decompose $R(ABCDE)$



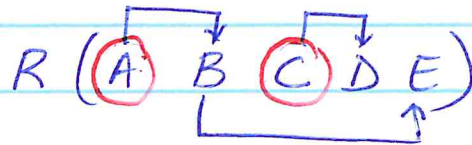
Example 2: $R(A B C D E)$

$A \rightarrow B$

$B \rightarrow E$

$C \rightarrow D$

Soln:



AC^+ : ACBED $\therefore AC$ is the candidate key.

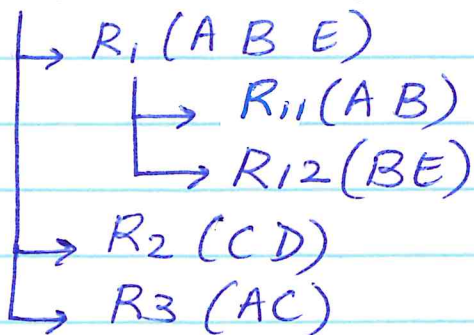
traverse through fd's to identify partial dependencies & transitive dependencies.

$A \rightarrow B$ pd.

$B \rightarrow E$ td.

$C \rightarrow D$ pd. $\therefore R$ is not in 3NF

decomposition: $R(A B C D E)$



$\therefore R \text{ decomp} \rightarrow R_1, R_2, R_3 \Rightarrow 2NF$

$R \text{ decomp} \rightarrow R_{11}, R_{12}, R_2, R_3 \Rightarrow 3NF.$

Example 3: given $R(ABCDEFGHIJ)$ with
fks $AB \rightarrow C$

$A \rightarrow DE$

$B \rightarrow F$

$F \rightarrow GH$

$D \rightarrow IJ$

and AB is the candidate key.

Decompose into 3NF.

Soln: $R(ABCDEFGHIJ)$

- ↳ $R_1(ADEIJ)$
 - ↳ $R_{11}(ADE)$
 - ↳ $R_{12}(DIJ)$
- ↳ $R_2(BFGH)$
 - ↳ $R_{21}(BF)$
 - ↳ $R_{22}(FGH)$
- ↳ $R_3(ABC)$

$\therefore R_{11}, R_{12}, R_{21}, R_{22}, \text{ and } R_3 \text{ are in 3NF}$