W22- CSC 430/530

Lecture 18

Feb 9th, 2022

§ 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 atleast one of the following dependencies is in F^+:

$$\left.\begin{array}{l}(i)\ R_1 \cap R_2 \rightarrow R_1 \\ (ii)\ R_1 \cap R_2 \rightarrow R_2\end{array}\right\} \text{sufficient condition}$$

※ § 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 → S_Name.

S_NO → City

S_NO, P_No → Qty

Decomposition: Supplier (<u>S_NO</u>, S_Name, City, Qty)

Parts (<u>P_No, Qty</u>)

Determine lossy / lossless join.

Soln: we have to determine Supplier ⋈ 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	Qty
3	Smith	London	20
5	Nick	NY	50
2	Steve	Boston	10
5	Nick	NY	40
5	Nick	NY	10

Parts

P_NO	Qty
301	20
500	50
20	10
400	40
301	10

Supplier ⋈ Parts

(i) holds

(ii) holds

S_NO	S_Name	City	P_NO	Qty
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.

∴ (iii) does not hold.

Try yourself:

Example: Given Supplier_Parts (S_NO, S_Name, City, P_No, Qty) R
fds: {S_NO → S_Name, S_NO → City, (S_No, P_No) → Qty}

Decomposition: Supplier (S_NO, S_Name, City) R_1
Parts (S_NO, P_No, Qty) R_2

Determine if lossy / lossless join.

Soln:

Check for:

(i) does $R_1 \cup R_2 = R$
i.e Supplier $\cup$ Parts = Supplier_Parts
↳ holds true.

(ii) does $R_1 \cap R_2 \neq \phi$
i.e. Supplier $\cap$ Parts = S_No
↳ holds true.

(iii) does $R_1 \cap R_2 \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$
↳ holds true using
i.e $R_1 \cap R_2$ = S_No
using fds: S_No → S_Name, S_No → City } covers Supplier

∴ (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(A B C D E) with fd's $AB \to C$; $D \to E$ determine if it is in 2NF; if not then decompose.

solution: {search for partial dependencies}

given: R(A B C D E) $\{AB \to C,\ D \to E\}$

step1: use the edge diagram to identify the candidate key.

R(A B C D E)

§ potential candidate key = ABD

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

step2: identify partial dependencies from fd's.

$AB \to C$

$D \to E$

§ if a fd's determinant (i.e α) contains a subset of the the candidate key; then that fd is a partial dependancy.

∴ $AB \to C$ is a partial depen.

$D \to E$ is a partial depen.

∴ R(ABCDE) is not in 2NF.

Converting / Decompose R in 2NF.

R(ABCDE)
- → R_1(ABC) based on partial dep AB→C
- → R_2(DE) based on pd D→E
- → R_3(ABD) the candidate key; to ensure lossless joins.

Example 2: R(ABCDE)

A → B

B → E

C → D State (True/False) R is in 2NF

soln:

R(ABCDE)

AC^+ = ACBED ∴ AC is the candidate key.

given AC^+:

A → B → {this is a partial dep.}

B → E

C → D → {this is a partial dep.}

∴ R is not in 2NF. [False]

R(ABCDE)
- → R_1(ABE)
- → R_2(CD)
- → R_3(AC)

} Decomposition is in 2NF

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

$AB \to C$

$AD \to GH$

$BD \to EF$

$A \to I$

$H \to J$

soln:

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

$ABD^+ : ABDCGHEFIJ$ ✓✓

$\left.\begin{array}{l} AB \to C \\ AD \to GH \\ BD \to EF \\ A \to I \end{array}\right\}$ are partial dep. in the fds.

decomposition:

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

→ $R_1(A\ B\ C)$

→ $R_2(A\ D\ G\ H, J)$ # Note fd $H \to J$ is included here.

→ $R_3(BD\ E\ F)$

→ $R_4(A\ I)$

→ $R_5(ABD)$

$R_1\ R_2\ R_3\ R_4$ & 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: $\{AB \to C, B \to D, D \to E\}$ determine if R is in 3NF and decompose.

Soln: $R(ABCDE)$

$AB \to C$
$B \to D$
$D \to E$

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

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

$AB \to C$
$B \to D$ {this fd is partially dep.}
$D \to E$ {this fd is a transitive dep.}

So conclude that R is not in 3NF.

Step 3: Decompose $R(ABCDE)$

- $R_1(ABC)$
- $R_2(BDE)$
  - $R_{21}(BD)$
  - $R_{22}(DE)$

} 3NF.

Eample 2: R (A B C D E)

$A \to B$

$B \to E$

$C \to D$

Soln: R (A B C D E)

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

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

$A \to B$ pd.

$B \to E$ td.

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

decomposition: R (A B C D E)

- R_1 (A B E)
  - R_{11} (A B)
  - R_{12} (BE)
- R_2 (C D)
- R_3 (AC)

$\therefore$ R decomp $\to R_1, R_2, R_3 \Longrightarrow$ 2NF

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

Example 3: given $R(ABCDEFGHIJ)$ with fds $AB \to C$

$A \to DE$

$B \to F$

$F \to GH$

$D \to IJ$

and AB is the candidate key.
Decompose into 3NF.

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

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

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