Schema Decomposition

R&G Chapter 19

In Last Lecture

- 1^{st} NF, 2^{nd} NF, 3^{rd} NF, and BCNF
- If a relation is not in a desired normal form, we need to decompose the relation.
- Decompositions should be used only when needed, as it can cause potential problems.

Problems with Decompositions

- There are three potential problems to consider:
  1) May be impossible to reconstruct the original relation! (Lossiness)
  2) Dependency checking may require joins.
  3) Some queries become more expensive.

***Tradeoff*: Must consider these issues vs. redundancy.**

Features of a Good Decomposition

- **A good decomposition is**
  - Lossless
  - Dependency preserving

Task #1

- **How to decompose the original relation so that it is lossless (i.e., the join of the decomposed relations is the same as the original relation)?**

Solution to Lossless Decomposition

- The decomposition of R into X and Y is **lossless with respect to F** *if and only if* the $\mathbf{F^+}$ contains:

$$X \cap Y \rightarrow X, \quad \textbf{or}$$

$$X \cap Y \rightarrow Y$$

In other words, the join attributes should be the key of X or Y.

- If $W \rightarrow Z$ holds over R and $W \cap Z$ is empty, then
  - decomposition of R into R-Z and WZ
  - R-Z and WZ are guaranteed to be loss-less (since R-Z and WZ joins at W)

Lossy Decomposition (example)

A	B	C
1	2	3
4	5	6
7	2	8

A	B
1	2
4	5
7	2

B	C
2	3
5	6
2	8

$A \rightarrow B; C \rightarrow B$

$X=\{A, B\}, Y=\{B, C\}, X \cap Y = \{B\}, B \not\rightarrow \{A, B\}$ and $B \not\rightarrow \{B, C\}$

Lossy decomposition!

A	B
1	2
4	5
7	2

Join

B	C
2	3
5	6
2	8

=

A	B	C
1	2	3
4	5	6
7	2	8
1	2	8
7	2	3

Lossless Decomposition (example)

A	B	C
1	2	3
4	5	6
7	2	8

A	C
1	3
4	6
7	8

B	C
2	3
5	6
2	8

$A \rightarrow B$; $C \rightarrow B$

X={A, C}, Y={ B, C}, $X \cap Y$ = {C}, C->{B, C}

Lossless decomposition!

A	C
1	3
4	6
7	8

Join

B	C
2	3
5	6
2	8

=

A	B	C
1	2	3
4	5	6
7	2	8

Lossless Decomposition Exercise 1

- **Relational table R(A,B,C,D,E)**
- **FDs F={AB->C, C->E, B->D, E->A}**
- **R is decomposed into R1(B, C, D) and R2(A, C, E)**
- **Is (R1, R2) a lossless decomposition?**
- **Way of thinking:**
  - Find common attribute: R1 $\cap$ R2 = (C);
  - Check whether C->(B, C, D) or C-> (A, C, E) in F+
  - C+ = (CEA). So it is a lossless decomposition.

Lossless Decomposition Exercise 2

- **Table R(A,B,C,D,E)**
- **FDs F=(A->BC, CD->E, B->D, E->A)**
- **R is decomposed into R1(A, B, C) and R2(A, D, E)**
- **Is (R1, R2) a lossless decomposition?**

Problem #2 of Decomposition

A	B	C
1	2	3
4	5	6
7	2	8

A	C
1	3
4	6
7	8

B	C
2	3
5	6
2	8

$A \rightarrow B$; $C \rightarrow B$

X={A, C}, Y={ B, C}, $X \cap Y$ = {C}, C->{B, C}

Lossless decomposition!

A	C
1	3
4	6
7	8

Join

B	C
2	3
5	6
2	8

=

A	B	C
1	2	3
4	5	6
7	2	8

But, now we can't check $A \rightarrow B$ without doing a join!
(Problem #2 of decomposition!)

BCNF Versus 3NF Decomposition

	BCNF	3NF
Redundancy	NONE	May still have some
Lossless-join decomposition	Guaranteed	Guaranteed
dependency-preserving decomposition	Not guaranteed	Guaranteed

Decomposition into BCNF

Consider relation R with FDs F.

- Step 1:
  - Ensure that each FD in F only contain a single attribute on right-hand side (RHS)
    - This is always doable, for example, if you have AB->CD, spit it into AB->C and AB->D;
- Step 2:
  - If $X \rightarrow Y$ (in F) violates BCNF (i.e., X is not the key of R), decompose R into R - Y and XY (guaranteed to be loss-les).

Repeat Step 1 & 2, until all FDs do not violate BCNF.

It will give a lossless decomposition that consists of BCNF relations (i.e., data redundancy free).

Decomposition into BCNF

Consider the relation R={CSJDPQV}:

– Its primary key is C;
– It has the following FDs: $JP \rightarrow C$, $SD \rightarrow P$, $J \rightarrow S$.

- **Question:**
- **(1) Does R satisfy BCNF?**
- **(2) If not, decompose R into BCNF tables.**

Decomposition into BCNF

Consider the relation R={CSJDPQV}:

- Its primary key is C;
- It has the following FDs: JP $\rightarrow$ C, SD $\rightarrow$ P, J $\rightarrow$ S.

- **Question:**
- **(1) Does R satisfy BCNF?**
- **(2) If not, decompose R into BCNF tables.**
  - To deal with SD $\rightarrow$ P, decompose into SDP, CSJDQV.
  - To deal with J $\rightarrow$ S, decompose CSJDQV into JS and CJDQV
  - So we end up with: SDP, JS, and CJDQV

(note: JP is a candidate key of R, so JP->C does not violated BCNF)