CSCI4333 Database Design & Implement

Lecture Twenty-five: Normalization 3

Instructor: Dr. Yifeng Gao

Normalization

- Consider relation obtained
  - Hourly_Emps(*ssn*, *name, lot, rating, hrly_wage*, *hrs_worked*)
  - call it SNLRHW

- What if we ***know*** rating (R) determines hrly_wage (W)?

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

Redundancy exists because of the existence of *integrity constraints* ***(e.g., FD: R→ W).***

What do we do? Decomposition

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

=

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

$\bowtie$

R	W
8	10
5	7

Decomposition

- Decomposition of R into R_1 and R_2
  - To enforce FD constraint
  - Reduce Redundancy

Decomposition

- Decomposition of R into R_1 and R_2

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

R

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

R_1

R	W
8	10
5	7

R_2

Decomposition

- Decomposition of R into R_1 and R_2
  - How to formally describe this operation in Relational Algebra?

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

R

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

R_1

R	W
8	10
5	7

R_2

Decomposition

- Decomposition of R into R_1 and R_2
  - How to formally describe this operation in Relational Algebra?

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

R

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

R_1

R	W
8	10
5	7

R_2

Projection!

Decomposition

- Decomposition of R into R_1 and R_2

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

R

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

$$R_1 = \pi_{S,N,L,R,H}(R)$$

R	W
8	10
5	7

R_2

Decomposition

- Decomposition of R into R_1 and R_2

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

R

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

$$R_1 = \pi_{S,N,L,R,H}(R)$$

R	W
8	10
5	7

$$R_2 = \pi_{R,W}(R)$$

Decomposition

- Decomposition of R into R_1 and R_2
  - To enforce FD constraint
  - Reduce Redundancy
- Reconstruction from R_1 and R_2
  - Check whether the information is unintentionally modified

Reconstruction

- Formally Describe Reconstruction in Relational Algebra

$$R' = R_1 \bowtie R_2$$

Reconstruct Table

Not So Good Situation: $R \neq R'$

Student_ID	Name	Dcode	Cno	Grade
123-22-3666	Attishoo	INFS	501	A
231-31-5368	Guldu	CS	102	B
131-24-3650	Smethurst	INFS	614	B
434-26-3751	Guldu	INFS	614	A
434-26-3751	Guldu	INFS	612	C

$\neq$

Name	Dcode	Cno	Grade
Attishoo	INFS	501	A
Guldu	CS	102	B
Smethurst	INFS	614	B
Guldu	INFS	614	A
Guldu	INFS	612	C

$\bowtie$

Student_ID	Name
123-22-3666	Attishoo
231-31-5368	Guldu
131-24-3650	Smethurst
434-26-3751	Guldu

Observation

- Given instances of the decomposed relations, we may **not be able to** reconstruct the corresponding instance of the original relation!

Example A

r

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

$\pi_{AB}(r)$

A	B
1	2
4	5
7	2

$\pi_{BC}(r)$

B	C
2	3
5	6

$\pi_{AB}(r) \bowtie \pi_{BC}(r)$

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

Example B

r

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

Example B

r

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

$\pi_{AB}(r)$

A	B
1	2
4	5
7	2

$\pi_{BC}(r)$

B	C
2	3
5	6
2	8

$\pi_{AB}(r) \bowtie \pi_{BC}(r)$

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

Conclusion

- What Different between Example A and Example B? Hint: Consider FD

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

Example A

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

Example B

Conclusion

- What Different between Example A and Example B? Hint: Consider FD

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

Example A

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

Example B

$B \rightarrow BC$ holds in Example A, but not holds in Example B

Conclusion

- In which condition $R' = R$?

$$R$$

$$R_1 = \pi_{A1,A2,..,An}(R)$$

$$R_1 = \pi_{B1,B2,..,Bn}(R)$$

$$R' = R_1 \bowtie R_2$$

Lossless Join Decomposition

- Formally: The decomposition of R into R_1 and R_2 is lossless-join wrt F if and only if F^+ contains:
  - $R_1 \cap R_2 \rightarrow R_1$, or
  - $R_1 \cap R_2 \rightarrow R_2$

A more useful variation

- Suppose we have $U = R_1 \cap R_2$
- The decomposition of R into R_1 and R_2 is lossless-join wrt F if and only if:
  - $U \rightarrow R_1 - U$, or
  - $U \rightarrow R_2 - U$

  holds in R

Lossless Join Decompositions

- If decomposition of R into R_1 and R_2 is *lossless-join* , then we have: $R' = R$

Boyce-Codd Normal Form (BCNF)

We know what is Lossless Join Decomposition.

But how can we always have Lossless Join Decomposition? BCNF Decomposition

Superkey and F^+

- Given $F=\{ A \rightarrow B, B \rightarrow C\}$. Compute F^+ (with attributes A, B, C).

	A	B	C	AB	AC	BC	ABC
A	√	√	√	√	√	√	√
B		√	√			√	
C			√				
AB	√	√	√	√	√	√	√
AC	√	√	√	√	√	√	√
BC		√	√			√	
ABC	√	√	√	√	√	√	√

Attribute closure
$A^+=ABC$
$B^+=BC$
$C^+=C$
$AB^+=ABC$
$AC^+=ABC$
$BC^+=BC$
$ABC^+=ABC$

- An entry with √ means FD (the row) $\rightarrow$ (the column) is in F^+.
- An entry gets √ when (the column) is in (the row)$^+$

Superkey and F^+

- What is the super key of this table?

$A \rightarrow ABC$

	A	B	C	AB	AC	BC	ABC
A	√	√	√	√	√	√	√
B		√	√			√	
C			√				
AB	√	√	√	√	√	√	√
AC	√	√	√	√	√	√	√
BC		√	√			√	
ABC	√	√	√	√	√	√	√

Attribute closure
A^+=ABC
B^+=BC
C^+=C
AB^+=ABC
AC^+=ABC
BC^+=BC
ABC^+=ABC

- A, AB, AC, ABC

Boyce-Codd Normal Form (BCNF)

- Relation R with FDs F is in BCNF if:
  - **for each non-trivial FD** $X \rightarrow A$ in F, **X is a super key for R** (i.e., $X \rightarrow R$ in F^+).
  - An FD $X \rightarrow A$ is said to be "trivial" if $A \in X$.
  - If not all XA are in R, then we don't care.

- In other words, R is in BCNF if the only non-trivial FDs that hold over R are *key constraints*.

- BCNF means no "data" in R can be predicted using FDs alone.

Boyce-Codd Normal Form (BCNF)

- Relation R with FDs F is in BCNF if all FD in F satisfies:
  - **for each non-trivial FD** $X \rightarrow A$ in F **, X is a super key for R** (i.e., $X \rightarrow R$ in F^+).
  - An FD $X \rightarrow A$ is said to be "trivial" if $A \in X$.
  - If not all XA are in R, then we don't care.

- Another useful definition:
  - $X \rightarrow Y$ in F over R violates BCNF if $X \rightarrow Y$ does not satisfy BCNF condition

BCNF Decomposition

- Given a table R, we want to decompose it into multiple BCNF tables.

- If we found a dependency in F cause violation of BCNF, we decompose it into two table.

Decomposition into BCNF

- Consider relation R with FDs F. If $X \rightarrow Y$ in F over R violates BCNF,
- Then: decompose R into R - Y and XY.

r

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

R = ABC

$\xrightarrow{B \rightarrow C}$

$\pi_{AB}(r)$

A	B
1	2
4	5
7	2

R-Y = AB

$\pi_{BC}(r)$

B	C
2	3
5	6

XY = BC

Decomposition into BCNF

- Consider relation R with FDs F. If $X \rightarrow Y$ in F over R violates BCNF, i
- Then: decompose R into R - Y and XY.
- Recursively do it until every table is BCNF

Decomposition into BCNF

- Why we want to Decomposition into BCNF
  - Lossless joint decomposition
  - Always terminate

BCNF Decomposition Example

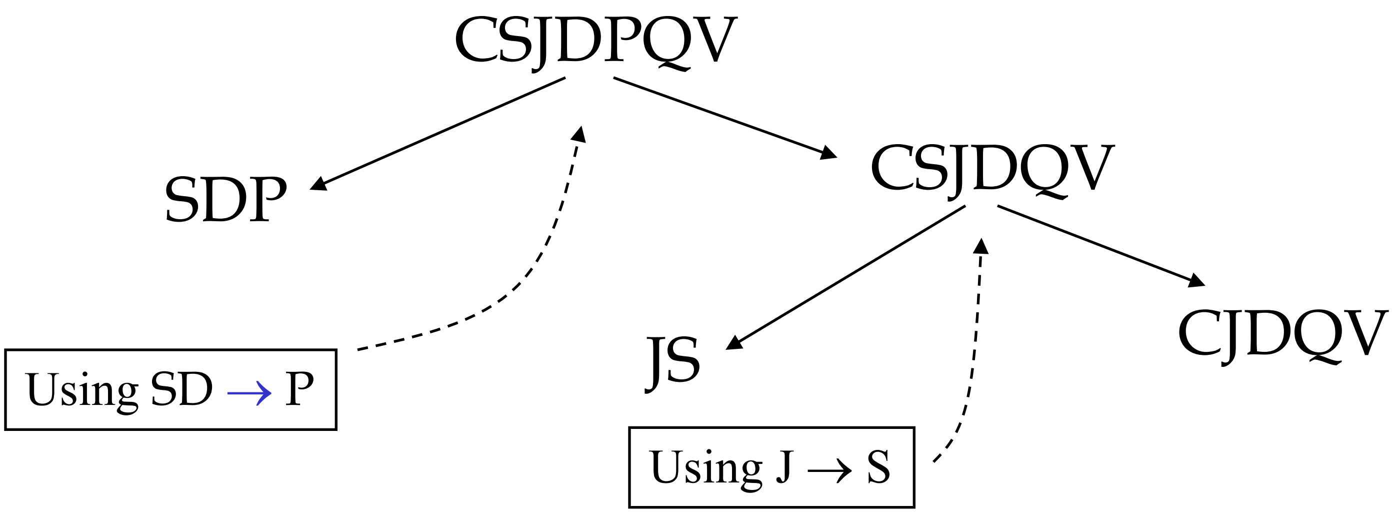
- $R = (A, B, C)$
  $F = \{A \rightarrow B; B \rightarrow C\}$
- Key = $\{A\}$
- R is not in BCNF ($B \rightarrow C$ but B is not a superkey)
- Decomposition
  - $R_1 = (B, C)$
  - $R_2 = (A, B)$

BCNF Decomposition Example 2

- Assume relation schema CSJDPQV:
  Contracts(*contract_id, supplier, project, dept, part, qty, value*)
  - key C ($C \rightarrow CSJDPQV$), JP $\rightarrow$ C, SD $\rightarrow$ P, J $\rightarrow$ S

BCNF Decomposition Example 2

- Assume relation schema CSJDPQV:
  Contracts(*contract_id, supplier, project, dept, part, qty, value*)
  - key C, JP $\rightarrow$ C, SD $\rightarrow$ P, J $\rightarrow$ S
- To deal with SD $\rightarrow$ P, decompose into SDP, CSJDQV.
- To deal with J $\rightarrow$ S, decompose CSJDQV into JS and CJDQV
- A tree representation of the decomposition:



Question Extra

- R(A, B, C, D, E, G, H)
  $F=\{B \rightarrow D, AB \rightarrow C, AC \rightarrow B, BC \rightarrow A, E \rightarrow G\}$

- Answer the BCNF Decomposition of R(ABCDEGH)

Question Extra

- Same as previous
- Not in BCNF
- Decomposition: