ICS 321 Data Storage & Retrieval

Normal Forms 1

Prof. Lipyeow Lim

Information & Computer Science Department

University of Hawaii at Manoa

The Problem with Redundancy

Hourly_Emps

SSN	Name	Lot	Rating	Hourly_wages	Hours_worked
123-22-2366	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

- Suppose hourly wages are determined by rating
- **Redundant storage** : (8,10) stored multiple times
- **Update anomaly** : change hourly wages in row 1
- **Insertion anomaly** : requires knowing hourly wages for the rating
- **Deletion anomaly** : deleting all (8,10) loses info

Using Two Smaller Tables

Hourly_Emps

SSN	Name	Lot	Rating	Hours_ worked
123-22-2366	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

RatingWages

Rating	Hourly_ wages
5	7
8	10

- **Any more anomalies ?** Update, Insertion, Deletion ?
- Remove redundancy by ***decomposition***
  - Since hourly wage is completely determined by rating, factor out hourly wage.
- **Pros**: less redundancy less anomalies
- **Cons**: retrieving the hourly wage of an employee requires a join

Normal Forms

- Helps with the question: do we need to refine the schema ?
- If a relation is in a certain *normal form* (BCNF, 3NF etc.), it is known that certain kinds of problems are avoided/minimized. This can be used to help us decide whether decomposing the relation will help.
- Role of FDs in detecting redundancy:
  - Consider a relation R with 3 attributes, ABC.
    - No FDs hold: There is no redundancy here.
    - Given A → B: Several tuples could have the same A value, and if so, they'll all have the same B value!

Boyce-Codd Normal Form (BCNF)

- Let R denote a relation, X a set of attributes from R, A an attribute from R, and F the set of FDs that hold over R.
- R is in **BCNF** if for all $X \rightarrow A$ in F^+,
  - $A \in X$ (trivial FD) or
  - X is a superkey

The only non-trivial FDs that hold are key constraints

- **Negation**: R is not in BCNF if there exists an $X \rightarrow A$ in F^+, such that $A \notin X$ (non-trivial FD) AND X is not a key

Examples: BCNF

- Are the following in BCNF ?

Firstname	Lastname	DOB	Address	Telephone
John	Smith	Sep 9 1979	Honolulu,HI	808-343-0809

F= { FLD → FLDAT}

Firstname	Lastname	DOB	Street	CityState	Zipcode	Telephone
John	Smith	Sep 9 1979	1680 East West Rd.	Honolulu,HI	96822	808-343-0809

F= { FLD → FLDSCZT, C→Z }

Third Normal Form (3NF)

- Let **R** denote a relation, **X** a set of attributes from R, **A** an attribute from R, **F** the set of FDs for R.
- R is in **3NF** if for all $X \rightarrow A$ in F^+,
  - $A \in X$ (trivial FD) or
  - X is a superkey or
  - A is part of some key
- **Negation**: R is not in 3NF if there exists an $X \rightarrow A$ in F^+, such that
  - $A \notin X$ (non-trivial FD) AND
  - X is not a key AND A is not part of some key
- If R is in BCNF, obviously in 3NF.
- If R is in 3NF, some redundancy is possible. It is a compromise, used when BCNF not achievable (e.g., no ``good'' decomp, or performance considerations).

Example: 3NF

- Which of the following is in 3NF and which in BCNF ?

Firstname	Lastname	DOB	Address	Telephone
John	Smith	Sep 9 1979	Honolulu,HI	808-343-0809

F= { FLD → FLDAT}

Firstname	Lastname	DOB	Street	CityState	Zipcode	Telephone
John	Smith	Sep 9 1979	1680 East West Rd.	Honolulu,HI	96822	808-343-0809

F= { FLD → FLDSCZT, C→Z }

Student	Course	Instructor
Smith	OS	Mark

F= { SC → I, I→C }

Redundancies & Decompositions

Hourly_Emps

SSN	Name	Lot	Rating	Hourly_wages	Hours_worked
123-22-2366	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

Hourly_Emps

SSN	Name	Lot	Rating	Hours_ worked
123-22-2366	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

RatingWages

Rating	Hourly_ wages
5	7
8	10

Decompositions

- Reduces redundancies and anomalies, but could have the following potential problems:
  1. Some queries become more expensive.
  2. Given instances of the decomposed relations, we may not be able to reconstruct the corresponding instance of the original relation!
  3. Checking some dependencies may require joining the instances of the decomposed relations.
- Two desirable properties:
  - Lossless-join decomposition
  - Dependency-preserving decomposition

Lossless-join Decomposition

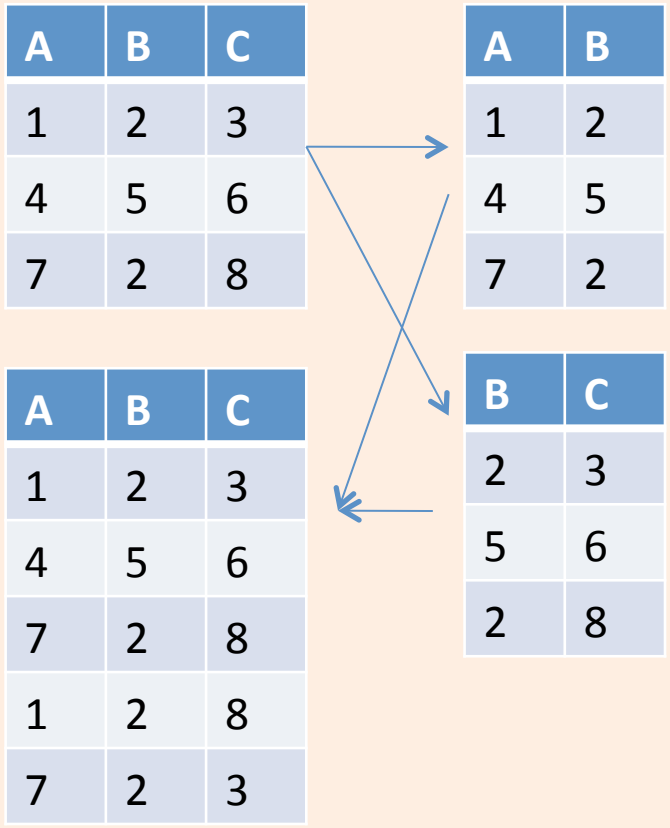
- Decomposition of R into X and Y is *lossless-join* w.r.t. a set of FDs F if, for every instance *r* that satisfies F:

$$\pi_X(r) \text{ join } \pi_Y(r) = r$$

- In general one direction $\pi_X(r)$ join $\pi_Y(r) \subseteq r$ is always true, but the other may not hold.
- Definition extended to decomposition into 3 or more relations in a straightforward way.
- *It is essential that all decompositions used to deal with redundancy be lossless! (Avoids Problem (2).)*

Conditions for Lossless Join

- The decomposition of R into X and Y is lossless-join wrt F if and only if the closure of F contains:
  - $X \cap Y \rightarrow X$, or
  - $X \cap Y \rightarrow Y$
- In particular, the decomposition of R into UV and R - V is lossless-join if $U \rightarrow V$ holds over R.



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

A	B
1	2
4	5
7	2

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

B	C
2	3
5	6
2	8