CSCI4333 Database Design & Implement

Lecture Twenty-three: Normalization

Instructor: Dr. Yifeng Gao

DELETE/UPDATE

- UPDATE Sailors SET rating = 3 WHERE Sid = 11;

- DELETE FROM Sailors WHERE Sid = 11;

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

the existence of *integrity constraints* ***(e.g., R→ W).***

functional dependencies

Redundancy

- When part of data can be **derived** from other parts, we say *redundancy* exists.
- Can you guess the value "?"

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	?	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

What's the problem, again

- *Update anomaly*: Can we change W in just the 1st tuple of SNLRWH?
- *Insertion anomaly*: What if we want to insert an employee and don't know the hourly wage for his rating?
- *Deletion anomaly*: If we delete all employees with rating 5, we lose the information about the wage for rating 5!

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

⋈

R	W
8	10
5	7

What do we do?

- When redundancy exists, refinement is needed.
  - Main refinement technique: *decomposition* (replacing ABCD with, say, AB and BCD, or ACD and ABD).

- Decomposition should be used judiciously:
  - Is there reason to decompose a relation?
  - What problems (if any) does the decomposition cause?

Recall Couple of Notations

R

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

inst

Functional Dependencies (FDs)

- A functional dependency (FD) has the form: $X \rightarrow Y$, where X and Y are two *sets* of attributes.
  - Examples: rating→hrly_wage, AB →C
- Convention: X, Y, Z etc denote sets of attributes, and A, B, C, etc denote attributes.

- The FD $X \rightarrow Y$ ***is satisfied by a relation instance r if:***
  - for each pair of tuples t1 and t2 in r:

    $t1.X = t2.X$ implies $t1.Y = t2.Y$
  - i.e., given any two tuples in *r*, if the X values agree, then the Y values must also agree. (X and Y are *sets* of attributes.)

Violation of FD by a relation

- The FD $X \rightarrow Y$ ***is NOT satisfied by a relation instance r if:***
  - There exists a pair of tuples t1 and t2 in r such that

    $t1.X = t2.X$ but $t1.Y \neq t2.Y$
  - i.e., we can find two tuples in *r*, such that X values agree, but Y values don't.

Functional Dependencies (FDs)

- ***The FD holds*** over relation name R if, for every ***allowable*** instance *r* of R, *r* satisfies the FD.

- ***The FD DO NOT holds*** over relation name R if, for there exists an ***allowable*** instance *r* of R, *r* does not satisfy the FD.

Functional Dependencies (FDs)

A	B	C
1	1	2
1	1	3
2	1	3
2	1	2

How many *possible* FDs totally on this relation instance?

FDs with A as the left side:	**Satisfied by the relation instance?**
A→A	yes
A→B	yes
A→C	No
A→AB	yes
A→AC	No
A→BC	No
A→ABC	No

Some other FDs

A	B	C
1	1	2
1	1	3
2	1	3
2	1	2

FD	**Satisfied by the relation instance?**
C→B	yes
C→AB	No
B→C	No
B→B	Yes
AC →B	Yes [note!]
…	…

Example: Constraints on Entity Set

- Consider relation obtained from Hourly_Emps:
  - Hourly_Emps (*ssn*, *name, lot, rating, hrly_wage*, *hrs_worked*)
- *Notation*: We will denote this relation schema by listing the attributes: SNLRWH
  - This is really the *set* of attributes {S,N,L,R,W,H}.
  - Sometimes, we will refer to all attributes of a relation by using the relation name. (e.g., Hourly_Emps for SNLRWH)
- Some FDs on Hourly_Emps:
  - *ssn* is the key: S → SNLRWH
  - *rating* determines *hrly_wage*: R → W

Functional Dependencies (FDs)

- An FD, as an integrity constraint, is a statement about *all* allowable relation instances.
  - Must be identified based on semantics of application.
  - Given some instance *r1* of R, we can check if it ***violates*** some FD *f* or not
  - But we cannot tell if *f* ***holds*** over R by looking at an instance!
    - Cannot prove non-existence (of violation) out of ignorance
  - This is the same for all integrity constraints!

Reasoning About FDs

- Given some FDs, we can usually infer additional FDs:
  - $ssn \rightarrow did,\ did \rightarrow lot$ implies $ssn \rightarrow lot$
  - $A \rightarrow BC$ implies $A \rightarrow B$
- An FD f is *logically implied by* a set of FDs F if f holds whenever all FDs in F hold.
  - F^+ = *closure of F* is the set of all FDs that are implied by F.