UCS310 Database Management System

Relational Design

Lecture-17 Date: 20 Feb 2023

Dr. Sumit Sharma
Assistant Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

Recap

- Relation design features
 - Attributes semantics
 - Redundant information in tuples & update anomaly
 - NULL values in tuples
 - Spurious tuples due to loosy decomposition
- Functional Dependencies

Functional Dependencies

- Functional dependencies (FDs) are used to specify formal measures of the "goodness" of relational designs
- FDs and keys are used to define normal forms for relations
- FDs are constraints that are derived from the meaning and interrelationships of the data attributes
- A set of attributes X functionally determines a set of attributes Y
 - if the value of X determines a unique value for Y

Functional Dependencies

- If A is key then,
 - all the other attributes, i.e., B and C can be uniquely identified using A

Means,

$$A \rightarrow BC$$

A	В	C
1		
2	a	b
3		
4		

Functional Dependencies

- X -> Y holds,
 - if whenever two tuples have the same value for X,
 - they must have the same value for Y
- For any two tuples t1 and t2 in any relation instance r(R):
 - If $t_1[X]=t_2[X]$,
 - then t1[Y]=t2[Y]
- X -> Y in R specifies a constraint on all relation instances r(R)
- Written as X -> Y; can be displayed graphically on a relation schema as in Figures (denoted by the arrow:)
- FDs are derived from the real-world constraints on the attributes

Examples of FD constraints

- social security number determines employee name
 - SSN -> ENAME

- project number determines project name and location
 - PNUMBER -> {PNAME, PLOCATION}
- employee ssn and project number determines the hours per week that the employee works on the project
 - {SSN, PNUMBER} -> HOURS

Examples of FD constraints

- An FD is a property of the attributes in the schema R
- The constraint must hold on every relation instance r(R)
- If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples with t1[K]=t2[K])

Functional Dependencies (FDs)

- If A -> B,
 - then A need not be key always

• For every value of A, we should be able to uniquely identifying B values

A	B	C
a	1	d
a	1	e
b	2	f
b	2	g

FDs tells which are the areas that can further decompose

Rule out the FD based on the tables

Eid	Ename
1	a
2	b
3	b

Eid -> Ename

Ename -> Eid

Eid	Ename
1	a
2	b
3	b

$$A \rightarrow B$$

$$B \rightarrow A$$

Eid	Ename
1	a
2	b
3	b

$$A \rightarrow B$$
$$B \rightarrow A$$

A	В	C
1	1	4
1	2	4
2	1	3
2	2	3
2	4	3

$$B \rightarrow A$$

$$C \rightarrow B$$

$$A \rightarrow C$$

A	В	C
1	1	4
1	2	4
2	1	3
2	2	3
2	4	3

$$\begin{array}{c} A \longrightarrow B \\ B \longrightarrow C \\ B \longrightarrow A \\ C \longrightarrow B \\ C \longrightarrow A \\ C \longrightarrow C \end{array}$$

X	Y	Z
1	4	3
1	5	3
4	6	3
3	2	3

$$XZ \rightarrow X$$
 $XY \rightarrow Z$
 $Z \rightarrow Y$
 $Y \rightarrow Z$
 $XZ \rightarrow Y$

X	\mathbf{Y}	Z
1	4	3
1	5	3
4	6	3
3	2	3

$$XZ \rightarrow X$$
 $XY \rightarrow Z$
 $Z \rightarrow Y$
 $Y \rightarrow Z$
 $XZ \rightarrow Y$

A	В	C
1	1	1
1	1	О
2	3	2
2	3	2

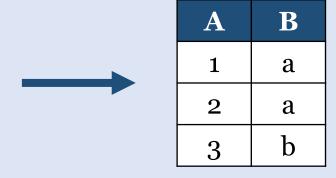
- B does not functionally determineC
- A -> B is valid for the particular instance but may not hold for the entire database
- Therefore, A does not functionally determine B

Functional Dependencies (FDs)

■ If A -> B,

	A	В
t1		
t2		
	If t1 and t2 agree here	Then they must agree here also
	If t1 and t2 disagree here	Then they may agree or Disagree here

A	В
1	a
1	a
2	a
3	b
1	a
1	a



Inference Rules for FDs

- Given a set of FDs F, we can infer additional FDs that hold whenever the FDs in F hold
- Armstrong's inference rules:
 - IR1. (**Reflexive**) If Y *subset of* X, then X -> Y
 - IR2. (Augmentation) If X -> Y, then XZ -> YZ
 (Notation: XZ stands for X U Z)
 - IR3. (**Transitive**) If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

 IR1, IR2, IR3 form a sound and complete set of inference rules

Inference Rules for FDs

Some additional inference rules that are useful:

- (**Decomposition**) If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
- **■** (**Union**) If X -> Y and X -> Z, then X -> YZ
- (Psuedotransitivity) If X -> Y and WY -> Z, then WX -> Z

The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3 (completeness property)

Thanks!