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 -> BC

A	B	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 t1[X]=t2[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 $t_1[K]=t_2[K]$)

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

FD Example

- Rule out the FD based on the tables

Eid	Ename
1	a
2	b
3	b

Eid -> Ename

Ename -> Eid

FD Example

- Rule out the FD based on the tables

Eid	Ename
1	a
2	b
3	b

Eid -> Ename

~~Ename -> Eid~~

A	B
1	1
1	2
2	2

A -> B

B -> A

FD Example

- Rule out the FD based on the tables

Eid	Ename
1	a
2	b
3	b

Eid -> Ename

~~Ename -> Eid~~

A	B
1	1
1	2
2	2

~~A -> B~~

~~B -> A~~

FD Example

- Rule out the FD based on the tables

A	B	C
1	1	4
1	2	4
2	1	3
2	2	3
2	4	3

A -> B

B -> C

B -> A

C -> B

C-> A

A -> C

FD Example

- Rule out the FD based on the tables

A	B	C
1	1	4
1	2	4
2	1	3
2	2	3
2	4	3

~~A -> B~~

~~B -> C~~

~~B -> A~~

~~C -> B~~

C-> A

A -> C

FD Example

- Rule out the FD based on the tables

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

XZ -> X

XY -> Z

Z -> Y

Y -> Z

XZ -> Y

FD Example

- Rule out the FD based on the tables

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

XZ -> X

XY -> Z

~~Z -> Y~~

Y -> Z

~~XZ -> Y~~

FD Example

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

- B does not functionally determine C
- 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	B
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	B
1	a
1	a
2	a
3	b
1	a
1	a

→

A	B
1	a
2	a
3	b

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 -> Y and Y -> Z, then X -> 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 -> YZ, then X -> Y and X -> 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!