Relational Database Design

CPS 216

Advanced Database Systems

Relational design: a review

- Identifying tuples: keys
- Generalizing the key concept: FDs
- Non-key FDs: redundancy
- Avoiding redundancy: BCNF decomposition
- Preserving FDs: 3NF

2

BNCF = no redundancy?

- *Student* (*SID*, *CID*, *club*)
  - Suppose your classes have nothing to do with the clubs you join
  - FDs?
    - None
  - BNCF?
    - Yes
  - Redundancies?
    - Tons!

SID	*CID*	*club*
142	CPS 216	ballet
142	CPS 216	sumo
142	CPS 214	ballet
142	CPS 214	sumo
123	CPS 216	chess
123	CPS 216	golf
…	…	…

3

Multi-valued dependencies

- A multi-valued dependency (MVD) has the form $X \rightarrow\rightarrow Y$, where X and Y are sets of attributes in a relation R
- $X \rightarrow\rightarrow Y$ means that whenever two tuples in R agree on all the attributes of X, then we can swap their Y components and get two new tuples that are also in R

X	*Y*	*Z*	
a	b1	c1	
a	b2	c2	
a	b1	c2	← Must be in *R* as well
a	b2	c1	
…	…	…	

4

MVD examples

Student (*SID*, *CID*, *club*)

- $SID \rightarrow\rightarrow CID$
- $SID \rightarrow\rightarrow club$
  - Intuition: given SID, CID and club are "independent"
- $SID, CID \rightarrow\rightarrow club$
  - Trivial: LHS $\cup$ RHS = all attributes of R
- $SID, CID \rightarrow\rightarrow SID$
  - Trivial: LHS $\supseteq$ RHS

5

Complete MVD + FD rules

- FD reflexivity, augmentation, and transitivity
- MVD complementation:
  If $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow$ attrs(R) $- X - Y$
- MVD augmentation:
  If $X \rightarrow\rightarrow Y$ and $V \subseteq W$, then $XW \rightarrow\rightarrow YV$
- MVD transitivity:
  If $X \rightarrow\rightarrow Y$ and $Y \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow Z - Y$
- Replication (FD is MVD):
  If $X \rightarrow Y$, then $X \rightarrow\rightarrow Y$
- Coalescence:
  If $X \rightarrow\rightarrow Y$ and $Z \subseteq Y$ and there is some W disjoint from Y such that $W \rightarrow Z$, then $X \rightarrow Z$

Try proving dependencies with these!?

6

An elegant solution: chase

- Given a set of FDs and MVDs D, does another dependency d (FD or MVD) follow from D?
- Procedure
  - Start with the hypotheses of d, and treat them as "seed" tuples in a relation
  - Apply the given dependencies in D repeatedly
    - If we apply an FD, we infer equality of two symbols
    - If we apply an MVD, we infer more tuples
  - If we infer the conclusion of d, we have a proof
  - Otherwise, if nothing more can be inferred, we have a counterexample

Proof by chase

- In $R(A, B, C, D)$, does $A \twoheadrightarrow B$ and $B \twoheadrightarrow C$ imply $A \twoheadrightarrow C$?

Have

	A	B	C	D
	a	b1	c1	d1
	a	b2	c2	d2
$A \twoheadrightarrow B$	a	b2	c1	d1
	a	b1	c2	d2
$B \twoheadrightarrow C$	a	b2	c1	d2
	a	b2	c2	d1
$B \twoheadrightarrow C$	a	b1	c2	d1
	a	b1	c1	d2

Need

A	B	C	D
a	b1	c2	d1
a	b2	c1	d2

Another proof by chase

- In $R(A, B, C, D)$, does $A \rightarrow B$ and $B \rightarrow C$ imply $A \rightarrow C$?

Have

A	B	C	D
a	b1	c1	d1
a	b2	c2	d2

Need: c1 = c2

$A \rightarrow B$: b1 = b2

$B \rightarrow C$: c1 = c2

In general, both new tuples and new equalities may be generated

Counterexample by chase

- In $R(A, B, C, D)$, does $A \twoheadrightarrow BC$ and $CD \rightarrow B$ imply $A \rightarrow B$?

Have

	A	B	C	D
	a	b1	c1	d1
	a	b2	c2	d2
$A \twoheadrightarrow BC$	a	b2	c2	d1
	a	b1	c1	d2

Need: b1 = b2

4NF

- A relation R is in Fourth Normal Form (4NF) if
  - For every non-trivial MVD $X \twoheadrightarrow Y$ in R, X is a super key
  - That is, all FDs and MVDs follow from "key $\rightarrow$ other attributes"
- 4NF is stronger than BCNF
  - Because every FD is also an MVD

4NF decomposition algorithm

- Find a 4NF violation
  - A non-trivial MVD $X \twoheadrightarrow Y$ in R where X is not a super key
- Decompose R into R_1 and R_2, where
  - R_1 has attributes $X \cup Y$
  - R_2 has attributes $X \cup Z$ (Z contains attributes not in X or Y)
- Repeat until all relations are in 4NF
- Almost identical to BCNF decomposition algorithm
- Any decomposition on a 4NF violation is lossless

4NF decomposition example

SID	*CID*	*club*
142	CPS 216	ballet
142	CPS 216	sumo
142	CPS 214	ballet
142	CPS 214	sumo
123	CPS 216	chess
123	CPS 216	golf
…	…	…

Student (*SID*, *CID*, *club*)

4NF violation: *SID* →→ *CID*

Enroll (*SID*, *CID*)
4NF

SID	*CID*
142	CPS 216
142	CPS 214
123	CPS 216
…	…

Join (*SID*, *club*)
4NF

SID	*club*
142	ballet
142	sumo
123	chess
123	golf
…	…

3NF, BCNF, and 4NF

	3NF	BCNF	4NF
Preserves FDs?	Yes	No	No
Redudancy due to FDs?	Possible	No	No
Redundancy due to MVDs?	Possible	Possible	No

Recap

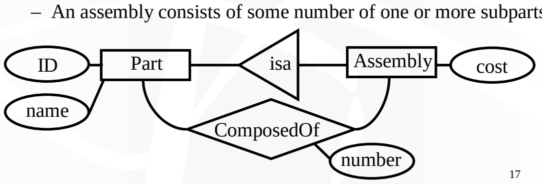
- Another source of redundancy: MVDs
- Reasoning about FDs and MVDs: chase
- Avoiding redundancy due to MVDs: 4NF

A complete design example

- Information about parts and assemblies for a manufacturing company; e.g.:
  - A bicycle consists of one frame and two wheels; the cost of assembly is $30
  - A frame is just a basic part
  - A wheel consists of one tire, one rim, and 48 spokes; the cost of assembly is $40
  - Everything has a part ID and a name

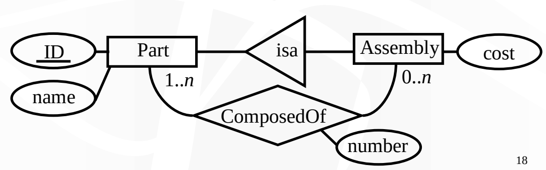
Entities and relationships

- Entities
  - Parts (with ID and name)
  - Assemblies (with ID, name, and cost)
- Relationships
  - An assembly as a whole is a part (with an assembly cost)
  - An assembly consists of some number of one or more subparts

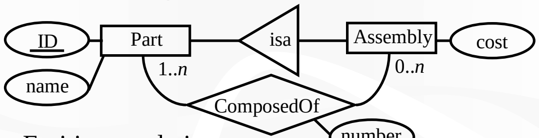


Identify constraints

- ID is a key for parts and assemblies
- An assembly has one or more subparts
- A part can serve as a subpart for zero or more assemblies



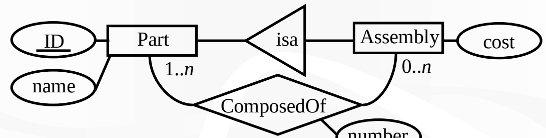
Design relational schema



- Entities to relations
  - *Part* (*ID*, *name*)
  - *Assembly* (*ID*, *cost*)
    - *ID* is inherited from *Part*; *name* is not repeated
- Relationships to relations
  - *ComposedOf* (*assemblyID*, *partID*, *number*)
    - Use keys as "links"

19

Encode constraints



- *Part* (*ID*, *name*)
  - *ID* is a key
- *Assembly* (*ID*, *cost*)
  - *ID* is a key
- *ComposedOf* (*assemblyID*, *partID*, *number*)
  - {*assemblyID*, *partID*} is a key
- Any missing constraints?

20

Apply relational design theory

- *Part* (*ID*, *name*)
  - *ID* is a key
- *Assembly* (*ID*, *cost*)
  - *ID* is a key
- *ComposedOf* (*assemblyID*, *partID*, *number*)
  - {*assemblyID*, *partID*} is a key
- 3NF? BCNF? 4NF?
  - Yes, yes, yes

21

Populate schema with data

Part

ID	*name*
1	bicycle
2	frame
3	wheel
4	tire
5	rim
6	spoke
…	…

Assembly

ID	*cost*
1	30
3	40
…	…

ComposedOf

assemblyID	*partID*	*number*
1	2	1
1	3	2
3	4	1
3	5	1
3	6	48
…	…	…

22

Good design principles

- Avoid redundancy
- Avoid decomposing too much
- KISS
  - Focus on the task and avoid over-design

23