COS 425:
Database and Information
Management Systems

Relational model continued

Understanding how to use
the relational model

Summary of board example

Relational model:

books: (title, ISBN#, edition, date)
PU branches: (br_name, librarian, hours)
copy: (ISBN#, copy#, condition, purchase date, br_name)
br_name not null
isbn# is a foreign key referencing ***books***
br_name is a foreign key referencing ***PU branches***

because ER key constraints
ER relationship *PU book* folded into ***copy***
(relationship between *books* and weak entity *copy*)
ER relationship *PU holding* folded into ***copy***

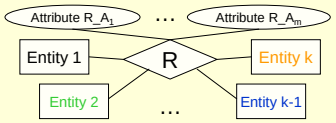
Board example:
Total participation constraints?

PU branches: (br_name, librarian, hours)
copy: (ISBN#, copy#, condition, purchase date, br_name)
br_name not null
isbn# is a foreign key referencing ***books***
br_name is a foreign key referencing ***PU branches***

"br_name not null" - constraint captures
total participation of ***copy*** in *PU holding*
Because *PU holding* represented within ***copy***

total participation of ***PU branches*** in *PU holding*
not representable in pure relational definition

Basic Paradigm



- Each entity becomes a relation
- Relationship becomes

R: { (list of attributes forming key of Entity 1 (denote L_1),
list of attributes forming key of Entity 2 (denote L_2),
…
list of attributes forming key of Entity k (denote L_k),
Attribute R_A_1, …, Attribute R_A_m)
L_1 is a foreign key referencing Entity 1,
…
L_k is a foreign key referencing Entity k }

Note primary key

What about constraints on relationships?

- Key constraint:
  - Simplifies key of corresponding relation
  - Allows folding of relation into key entity
- Total participation constraint:
  - In general, cannot represent in purely relational definition:
    - Domain specification
    - Keys of relations
    - Foreign keys
    - "not null"s

} Constraints have in relational definition

Enforcing relational constraints

- Constraints must be satisfied at all times
- What happens when tuples in relations change?
- Action of changing a relation not part of basic relational model
- Database language implementing model enforces

Enforcement in SQL

SQL commands changing relations:
INSERT, DELETE, UPDATE

- Domain constraints
  - Don't allow attribute value not in domain
    INSERT or UPDATE fails
- "Not null" constraints
  - Special case of domain constraints

Enforcement in SQL

- Key constraints
  - Can have other candidate keys declared as well as primary key
  - Don't allow 2nd tuple with same key value
    INSERT or UPDATE fails
  - Implicit "not null" for attributes in a key
    INSERT or UPDATE fails

Enforcement in SQL

- Foreign key constraints

Suppose Y denotes a set of attributes of relation B that reference the primary key of relation A.

  - Don't allow tuple into B if no tuple in A with matching values for Y
    INSERT or UPDATE fails

Enforcement in SQL

Foreign key constraints continued

- suppose want to remove a tuple in A
- Suppose there is a tuple in B with matching values for Y

Choices (in SQL):

1. Disallow deletion from A
   DELETE or UPDATE fails

Enforcement in SQL

Choices (in SQL) continued:

2. Ripple effect (CASCADE):
   - Remove tuple from A and all tuples from B with matching values for Y
   - DELETE or UPDATE in A causes DELETE in B
3. Substitute value
   - Put "null" (if Y not part of candidate key for B) or other default value for Y in B
   - DELETE or UPDATE in A causes UPDATE in B

Actions for board example?

books: (title, ISBN#, edition, date)

PU branches: (br_name, librarian, hours)

copy: (ISBN#, copy#, condition, purchase date, br_name)
br_name not null
isbn# is a foreign key referencing ***books***
br_name is a foreign key referencing ***PU branches***

What about constraints not expressible in ER model?

- Value based constraints?
- General functional constraints?

In relational model:

- Declaring and enforcing these depend on use of database language
- Use query semantics to check