















### 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



- 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

- · Candidate key constraints
  - Can have other candidate keys declared as well as primary key
  - Don't allow 2<sup>nd</sup> 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

11

10

12

### 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

13

15

17

## 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 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)

Copies: (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?

14

16

- · Value-based constraints?
- · General functional constraints?

In relational model:

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

COS 597A: Principles of Database and Information Systems

> Relational model: Relational algebra



### Queries

A query is a mapping from a set of relations to a relation

Query: relations  $\rightarrow$  relation

- Can derive schema of result from schemas of input relations
- Can deduce constraints on resulting relation that
   must hold for any input relations
- Can identify properties of result relation

19

21

### Relational query languages

- Two formal relational languages to describe mapping
   Relational algebra
  - Procedural lists operations to form query result
     Relational calculus
  - Declarative describes results of query
- Equivalent expressiveness
- Each has strong points for usefulness – DB system query languages (e.g. SQL) take best of both

20

### begin with Relational Algebra

### Basic operations of relational algebra:

- 1. Selection  $\sigma$  :select a subset of tuples from a relation according to a condition
- 2. Projection  $\pi$  :delete unwanted attributes (columns) from tuples of a relation
- 3. cross product X : combine all pairs of tuples of two relations by making tuples with all attributes of both
- 4. Set difference :\* tuples in first relation and not in second
- 5. union U:\* tuples in first relation or second relation
- 6. Renaming  $\rho$ : to deal with name conflicts

\* Set operations:  $D_1 X D_2 \dots X D_k$  of two relations must agree



- relation R
- · predicate P on attributes of R
- · resulting relation
  - schema same as R
  - contains those tuples of R that satisfy P
  - candidate keys and foreign keys in R are preserved
    - · eliminating tuples doesn't cause violations

22



### Projection $\pi_{s}(R)$

- relation R
- · S a list of attributes from R projected attributes
- · resulting relation:
  - scheme is attributes in S
  - contains all tuples formed by taking a tuple from R and keeping only the attributes listed in S
  - relations are sets ⇒ duplicates are removed
     In practice, usually not removed unless explicitly requested
  - if { candidate key projected, constraint preserved foreign
  - if no candidate key is projected,
  - only candidate key may be all attributes in S
     (set model) 24

Studei nstan	nts: ( <u>na</u> ce:	<u>me, ad</u>	dres	<u>is</u> , geno	ler, ag	e, gra	d yr)
name	addr	gender	age	grad yr			
Joe	NY	М	24	2			
Sally		F	25	3			
Joe	NJ	М	23	2			
Jan		F	27	4			
						<u>name</u>	grad yr
T <sub>name, grad yr</sub> (Students): ( <u>name, grad yr</u> )						Joe	2
						Sally	3
						1	4

# $\begin{array}{l} \textbf{Composing operators}\\ \textbf{0} & \textbf{0} \\ \textbf{0} \\ \textbf{0} & \textbf{0} \\ \textbf{0} \\$





