
data
distributed
database
DBMS
parallel
shared
processing
network
system
architecture
DBMSs
Distributed
DDBMS
systems
performance
processors
access
memory
multiple
centralized
sites
site
single
computer
disk
stored
nothing
technology
applications
computers
processor
may
fragments
typically
major
Parallel
provide
local
sometimes
one
large
number
require
physically
within
users
machines
like
Database
control
scalability
approach
improve
distinction
logical
application
management
transparent
Shared
databases
development
definition
nodes
Figure
interconnection
bottleneck
administered
organizational
logically
decentralized
directly
islands
based
controlled
defined
remote
resources
disks
scalable
architectures
order
split
computing
communication
distribution
reliability
eliminates
fact
across
also
replicated
make
discuss
many
often
sites
referred
operational
Date
com
user
dis
makes
Management
consists
partitioned
used
least
capable
mode
System
must
associated
global
provides
collection
structure
incompatible
way
consider
per

Database System Concept (CSE 3103)

Lecture 04-Day 04

Nazmus Sakib, Assistant Professor, Dept. of CSE, AUST

Reduction to Relation Schemas

- Entity sets and relationship sets can be expressed uniformly as *relation schemas* that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of schemas.
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.
- Each schema has a number of columns (generally corresponding to attributes), which have unique names.

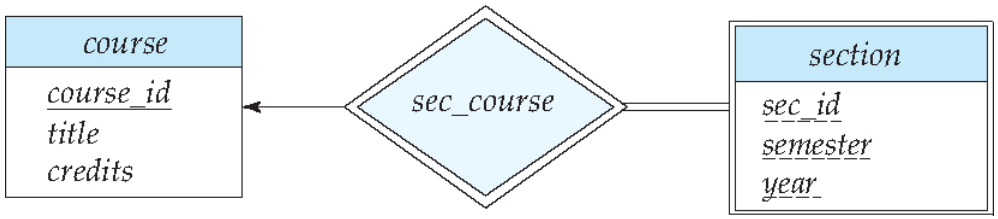
Representing Entity Sets

- A strong entity set reduces to a schema with the same attributes

  student(ID, name, tot_cred)

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

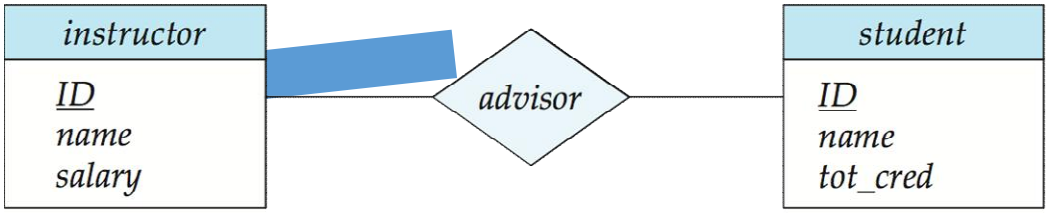
  section (course_id, sec_id, sem, year)



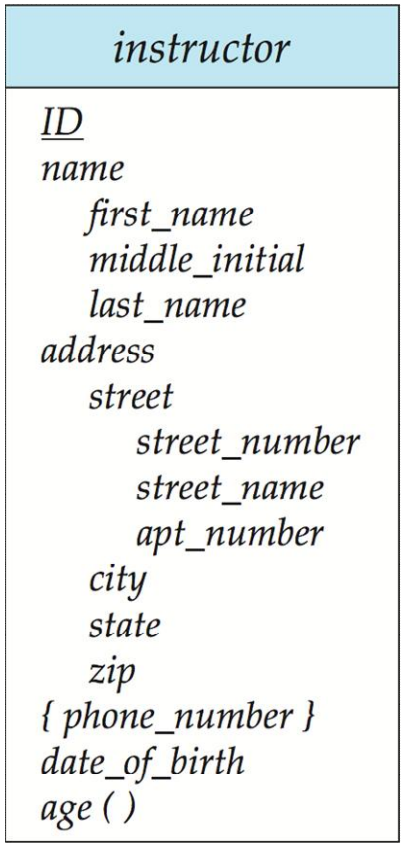
Representing Relationship Sets

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set *advisor*

advisor = (*s_id, i_id*)



Representation of Entity Sets with Composite Attributes



- Composite attributes are flattened out by creating a separate attribute for each component attribute
  - Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*
    - Prefix omitted if there is no ambiguity (*name_first_name* could be *first_name)*
- Ignoring multivalued attributes, extended instructor schema is
  - *instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)*

Representation of Entity Sets with Multivalued Attributes

- A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*
- Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*
- Example: Multivalued attribute *phone_number* of *instructor* is represented by a schema:
  inst_phone= (*ID*, *phone_number*)
- Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*
  - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples:
    (22222, 456-7890) and (22222, 123-4567)