

ER Model-Part 3

Reduction of ER Model to Relational Schema

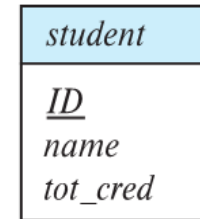
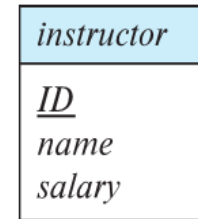
ER to Relational Mapping

- A database which conforms to an E-R diagram can be represented by a collection of schemas.

Representing Entity Sets

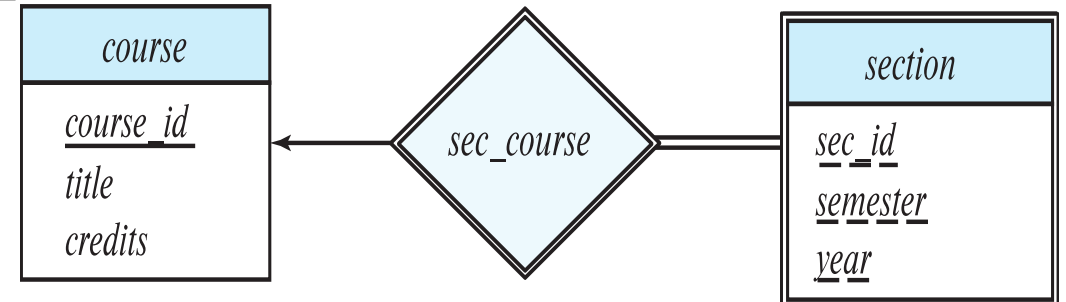
- A strong entity set reduces to a schema with the same attributes

student(ID, name, tot_cred)
Instructor(ID, name, salary)



- 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)



Representation of Entity Sets with Composite Attributes

<i>instructor</i>
<u>ID</u>
<i>name</i>
<i>first_name</i>
<i>middle_initial</i>
<i>last_name</i>
<i>address</i>
<i>street</i>
<i>street_number</i>
<i>street_name</i>
<i>apt_number</i>
<i>city</i>
<i>state</i>
<i>zip</i>
{ <i>phone_number</i> }
<i>date_of_birth</i>
<i>age</i> ()

- 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*)

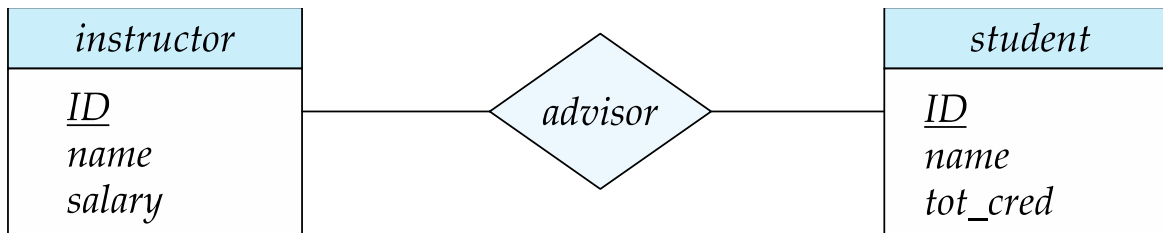
Representation of Entity Sets with Multivalued Attributes

- 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)

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*

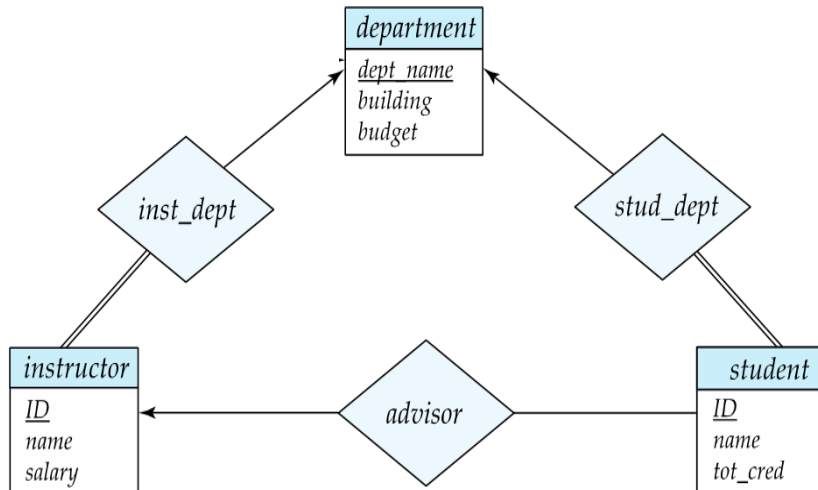
advisor = (*s id*, *i id*)



Representing Relationship Sets

- **Many-to-one and one-to-many relationship** sets that are total on the many-side can be represented by adding an extra attribute to the “many” side, containing the primary key of the “one” side
- Example: Instead of creating a schema for relationship set *inst_dept*, add an attribute *dept_name* to the schema arising from entity set *instructor*

- Example



Instructor(id,name,salary,dname)

Representing Relationship Sets

- For **one-to-one relationship** sets, either side can be chosen to act as the “many” side
 - That is, an extra attribute can be added to either of the tables corresponding to the two entity sets
 - It is better to add the primary key of entity set that is partially participating in the relationship set as a foreign key in the relation corresponding to entity at total participating side.

Mapping of N-ary Relationship Types.

- For each n-ary relationship type R, where $n > 2$, create a new relation S to represent R.
 - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
 - Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Summary –Mapping ER to Relational Schema

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	<i>Entity</i> relation
1:1 or 1:N relationship type	Foreign key (or <i>relationship</i> relation)
M:N relationship type	<i>Relationship</i> relation and <i>two</i> foreign keys
<i>n</i> -ary relationship type	<i>Relationship</i> relation and <i>n</i> foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key