



Chapter 7: Entity-Relationship Model

E-R to Relational translation

Database System Concepts, 7th Ed.

©Silberschatz, Korth and Sudarshan

See www.db-book.com for conditions on re-use



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. (TYPICALLY, AS WE SHALL SEE IT IS NOT ACTUALLY THE CASE)
- Each schema has a number of columns (generally corresponding to attributes), which have unique names.

⇒ REMEMBER THAT IN THE RELATIONAL MODEL WE JUST HAVE THE FUNDAMENTAL NOTION OF RELATION (TABLE)

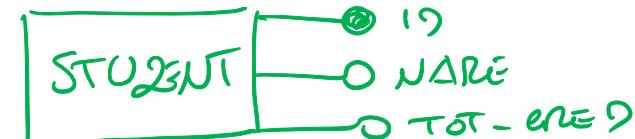


Representing Entity Sets

AS AN EXCEPTION TO THIS RULE, WE WILL SEE HOW TO DEAL WITH ONE-TO-ONE RELATIONSHIPS

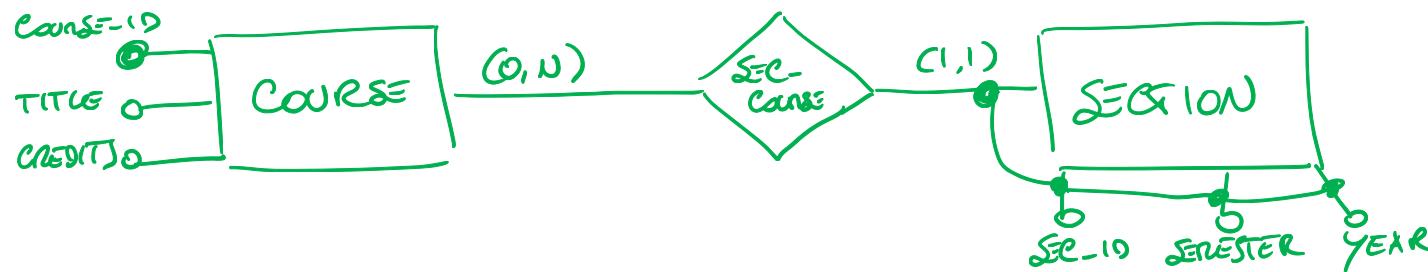
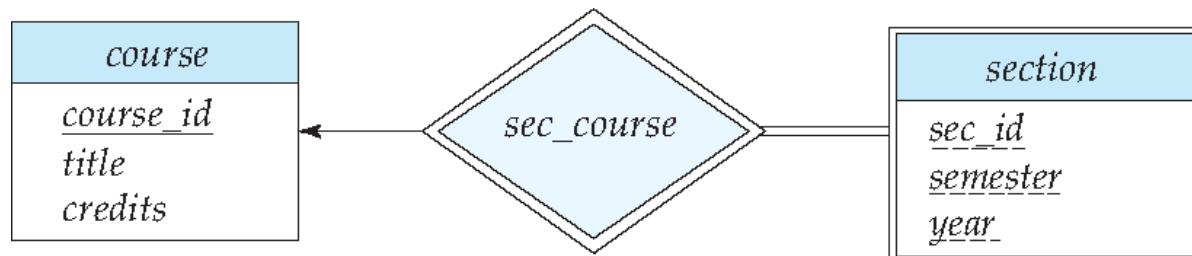
- A strong entity set reduces to a schema with the same attributes

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)





Representation of Entity Sets with Multivalued Attributes

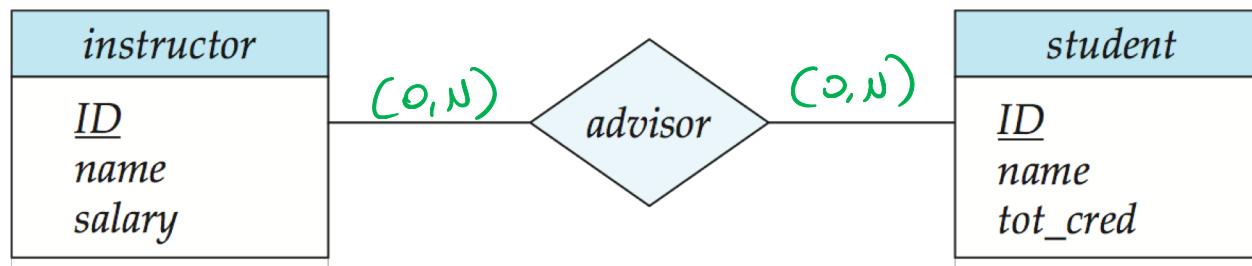
- A multivalued attribute M of an entity E is represented by a separate schema EM
- Schema EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M
- Example: Multivalued attribute $phone_number$ of $instructor$ is represented by a schema:
 $inst_phone = (\underline{ID}, \underline{phone_number})$
- 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)



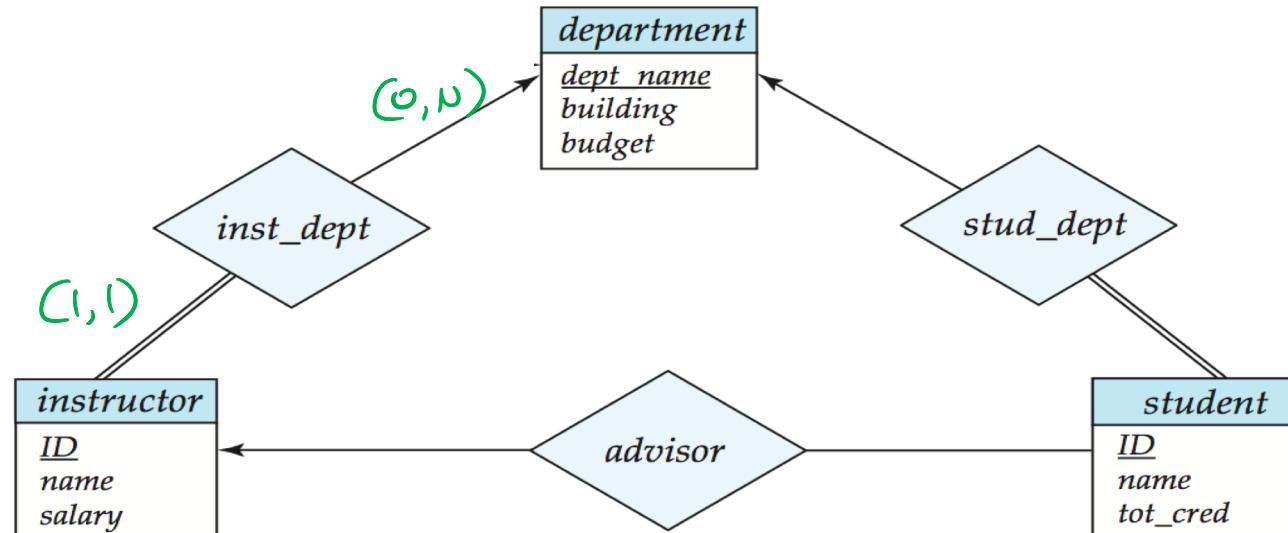
IN THE MANY-TO-MANY CASE, YOU
ALWAYS ADD A NEW RELATION
CORRESPONDING TO THE RELATIONSHIP SET,
IRRESPECTIVE FROM THE PARTICIPATION
CONSTRAINTS

CREATING A SINGLE,
UNIVERSAL TABLE WITH
ALL ATTRIBUTES OF
INSTRUCTOR AND STUDENT
IS NOT A VIABLE OPTION,
DUE TO REDUNDANCY
ISSUES



Redundancy of Schemas

- 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*



INSTRUCTOR (ID, NAME, SALARY, DEPT-NAME)

IF PARTICIPATION OF INSTRUCTOR
IS TOTAL, THEN ADD A NOT-NULL
CONSTRAINT HERE

A note on the ER-to-relational mapping: the case of one-to-one relationships

Angelo Montanari

Dept. of Mathematics, Computer Science, and Physics
University of Udine, Italy

Course on Data Management for Big Data

The mapping of one-to-one relationships



We have to distinguish among 3 different cases:

- ▶ $E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$

where $PK1$ is the key of entity $E1$, $A1$ is an attribute of $E1$, A is an attribute of relationship R , $PK2$ is the key of entity $E2$, and $A2$ is an attribute of $E2$

(the case

$E1(\mathbf{PK1}, A1) - (0, 1) - R(A) - (1, 1) - E2(\mathbf{PK2}, A2)$

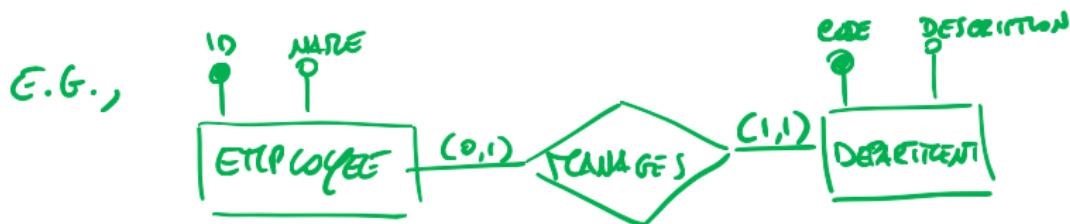
is completely symmetric, and thus ignored)

- ▶ $E1(\mathbf{PK1}, A1) - (0, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$
- ▶ $E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (1, 1) - E2(\mathbf{PK2}, A2)$

The case of $(1, 1) - (0, 1)$ relationships (AND VICE-VERSA)

How can we map the following ER schema into a corresponding relational one (introducing no redundancy, and preserving as much as possible information/constraints)?

$E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$



The case of $(1, 1) - (0, 1)$ relationships

How can we map the following ER schema into a corresponding relational one (introducing no redundancy, and preserving as much as possible information/constraints)?

$E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$

Relational schema:

$E1(\underline{PK1}, A1, PK2, A)$ and $E2(\underline{PK2}, A2)$, where $PK2$ is a foreign key of relation $E1$ that refers to the primary key $PK2$ of relation $E2$.

The case of $(1, 1) - (0, 1)$ relationships

How can we map the following ER schema into a corresponding relational one (introducing no redundancy, and preserving as much as possible information/constraints)?

$E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$

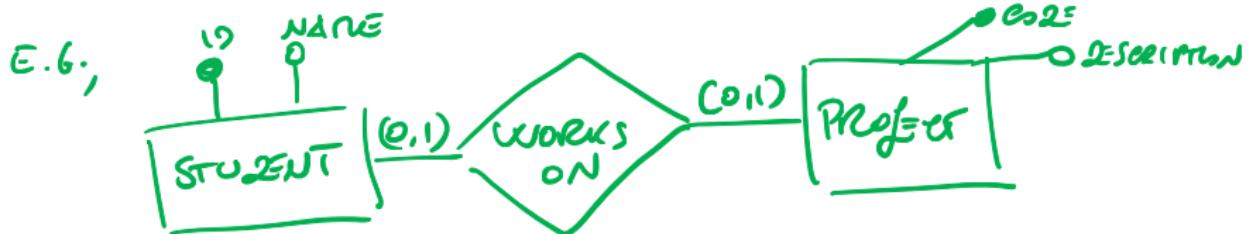
Relational schema:

$E1(\underline{PK1}, A1, PK2, A)$ and $E2(\underline{PK2}, A2)$, where $PK2$ is a foreign key of relation $E1$ that refers to the primary key $PK2$ of relation $E2$.

- ▶ $(1, _)$ on $E1$ side: $PK2$ NOT NULL in $E1$
- ▶ $(_, 1)$ on $E1$ side: $PK1$ is the primary key
- ▶ $(0, _)$ on $E2$ side: $PK2$ foreign key in $E1$ referring to the primary key of $E2$
- ▶ $(_, 1)$ on $E2$ side: $PK2$ UNIQUE in $E1$

The case of (0, 1) – (0, 1) relationships

$E1(\text{PK1}, A1) - (0, 1) - R(A) - (0, 1) - E2(\text{PK2}, A2)$



The case of (0, 1) – (0, 1) relationships

$E1(\mathbf{PK1}, A1) - (0, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$

Relational schema:

$E1(\underline{PK1}, A1, PK2, A)$ and $E2(\underline{PK2}, A2)$, where $PK2$ is a foreign key of relation $E1$ that refers to the primary key $PK2$ of relation $E2$.

The case of $(0, 1) - (0, 1)$ relationships

$E1(\mathbf{PK1}, A1) - (0, 1) - R(A) - (0, 1) - E2(\mathbf{PK2}, A2)$

Relational schema:

$E1(\underline{\mathbf{PK1}}, A1, \mathbf{PK2}, A)$ and $E2(\underline{\mathbf{PK2}}, A2)$, where $\mathbf{PK2}$ is a foreign key of relation $E1$ that refers to the primary key $\mathbf{PK2}$ of relation $E2$.

- ▶ $(0, _)$ on $E1$ side: $\mathbf{PK2}$ **can be NULL in $E1$**
- ▶ $(_, 1)$ on $E1$ side: $\mathbf{PK1}$ is the primary key
- ▶ $(0, _)$ on $E2$ side: $\mathbf{PK2}$ foreign key in $E1$ referring to the primary key of $E2$
- ▶ $(_, 1)$ on $E2$ side: $\mathbf{PK2}$ UNIQUE in $E1$

$E1(\underline{\mathbf{PK1}}, A1)$ and $E2(\underline{\mathbf{PK2}}, A2, \mathbf{PK1}, A)$, where $\mathbf{PK1}$ is a foreign key of relation $E2$ that refers to the primary key $\mathbf{PK1}$ of relation $E1$, works as well.

SO, BASICALLY IT IS THE SAME AS THE PREVIOUS SOLUTION, BUT WE DROP THE NOT-NUL CONSTRAINT

The case of (0, 1) – (0, 1) relationships (contn'd)

How do we choose between the two options? Participation of entities in the relationship can be a criterion.

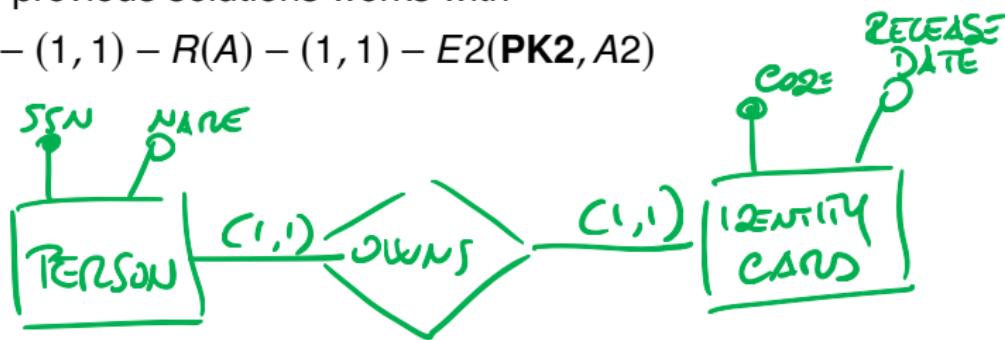
→ EXTEND THE RELATION THAT CORRESPONDS
TO THE ENTITY THAT "PARTICIPATES MOST" TO
THE RELATIONSHIP, SO TO AVOID WASTING
TOO MANY NULL VALUES.

The case of $(1, 1) - (1, 1)$ relationships

No one of the previous solutions works with

$E1(\text{PK1}, A1) - (1, 1) - R(A) - (1, 1) - E2(\text{PK2}, A2)$

E.G.,



The case of (1, 1) – (1, 1) relationships

No one of the previous solutions works with

$E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (1, 1) - E2(\mathbf{PK2}, A2)$

Relational schema:

$R(\underline{PK1}, A1, \underline{PK2}, A2, A)$ (or $R(PK1, A1, \underline{PK2}, A2, A)$)

- simply put all in a single table
- exception to the rule = "each entity set maps to its own table"

The case of $(1, 1) - (1, 1)$ relationships

No one of the previous solutions works with

$E1(\mathbf{PK1}, A1) - (1, 1) - R(A) - (1, 1) - E2(\mathbf{PK2}, A2)$

Relational schema:

$R(\underline{\mathbf{PK1}}, A1, \underline{\mathbf{PK2}}, A2, A)$ (or $R(\mathbf{PK1}, A1, \underline{\mathbf{PK2}}, A2, A)$)

- ▶ $(1, _)$ on $E1$ side: $\mathbf{PK2}$ NOT NULL in R
- ▶ $(_, 1)$ on $E1$ side: $\mathbf{PK1}$ is the primary key (UNIQUE)
- ▶ $(1, _)$ on $E2$ side: $\mathbf{PK1}$ is the primary key (NOT NULL)
- ▶ $(_, 1)$ on $E2$ side: $\mathbf{PK2}$ UNIQUE in R