

ER Diagrams to Relations 214/21

Book Notes 4.3 - 4.6

Converting ER Diagram to relation:

Straightforward part:

1. Turn each entity set into a relation with the same set of attributes
2. Replace a relationship by a relation whose attributes are the keys for the connected entity sets

Special Situations:

1. weak entity sets cannot be translated straightforwardly to relations
2. "Is-a" relationships and subclasses require careful treatment
3. Sometimes, we need to combine 2 relations, especially the relation for an entity set E and the relation that comes from a many-one relationship from E to some other entity set

A relation for a given relationship R:

1. For each entity set involved in relationship R, we take its key attributes as part of the schema of the relation for R
2. If the relationship has attributes, then these are also attributes of relation R.

~~AAA~~ Situation: Entity set E with a many-one relationship R from E to F. The relations from E to R will each have the key for E in their relation schema. In addition, the relation for E will have in its schema the attributes of E that are not in the key, and the relation for R will have the key attributes of F and any attributes of R itself. Because R is many-one, all these attributes are functionally determined by the key for E, and we can combine them into one relation with a schema consisting of:

1. All attributes of E

2. The key attributes of F

3. Any attributes belonging to relationship R

Handling weak Entity Sets:

Do 3 things differently:

1. Relation for weak entity set W
itself must include not only the attributes
of W but also the key attributes of the
supporting entity sets

2. Relation for any relationship in which
the weak entity set W appears must
use as a key for W all of its key
attributes, including those of other
entity sets that contribute to W's key

3. A supporting relationship R, from the
weak entity set W to a supporting entity,
need not be converted to a relation at all.
The justification is that, the attributes
of many-one relationship R's relation
will either be attributes for the relation
for W, or can be added to the schema
for W's relation

Rule for Converting weak entities:

- If W is a weak entity set, construct for W a relation whose schema consists of:
 1. All attributes of W
 2. All attributes of supporting relationships for W
 3. For each supporting relationship for W , say a many-one relationship from W to entity set E , all the key attributes of E

* Rename attributes, if necessary
 * No relation for any supporting relationship for W

Converting Subclass structures to Relations:

"Is a" relationship
 \Rightarrow we assume that:

-
1. There is a root entity set for the hierarchy
 2. This entity set has a key that serves to identify every entity represented by the hierarchy
 3. A given entity may have

components that belong to the entity sets of any subtree of the hierarchy, as long as that subtree includes the root

Principle Conversion Strategies:

1. For each entity set E in the hierarchy, create a relation that includes the key attributes from the root and any attributes belonging to E
2. For each possible subtree that includes the root, create one relation, whose schema includes all the attributes of all the entity sets in the subtree
3. Create one relation with all the attributes of all the entity sets in the hierarchy. Each entity is represented by one tuple, and that tuple has a null value for whatever attributes the entity does not have

Keys

• A key for an entity set E is a set K of one or more attributes

s.t. given any two distinct entities e_1 and e_2 in E , e_1 and e_2 cannot have identical values for each of the attributes in the key K .

Requirements:

1. Every entity set must have a key

2. There can be more than one possible key for an entity set.

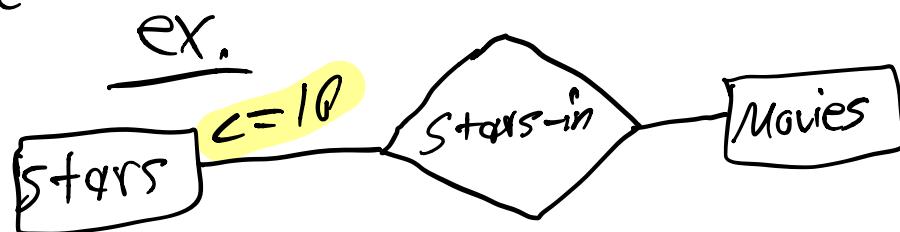
3. When an entity set is involved in a "isa" hierarchy, we require that the root entity have all the attributes needed for a key, and that the key for each entity is found from its component in the root entity set, regardless of how many entity sets in the hierarchy have components for the entity

Referential Integrity:

- \rightarrow \Rightarrow exactly one
- $\rightarrow \blacktriangleleft$ \Rightarrow at most one

Degree Constraints:

* We can attach limits (bounding #'s) to the edges that connect a relationship to an entity set, indicating limits on the number of entities that can be connected to any one entity of the related entity set.



Weak Entity Sets:

Requirements:

It's key consists of:

1. Zero or more of its own attributes
2. Key attributes from entity sets that are reached by certain many-one relationships from E to other entity sets (supporting relationships)

★ Questions: ★

(concepts =)

1. Foreign key (referential integrity):

A foreign key is a key that is a reference to a primary key of another entity set. It indicates a sharing of attributes. Referential integrity is a form of guaranteeing many-one relationships by stating that when adding or deleting tuples in one entity, there is an effect on the other entity w/constraints.

2. Weak entity set:

A weak entity set is an entity set whose attributes cannot uniquely define a primary key. These sets need a supporting relationship entity set that can create a unique primary key for it.

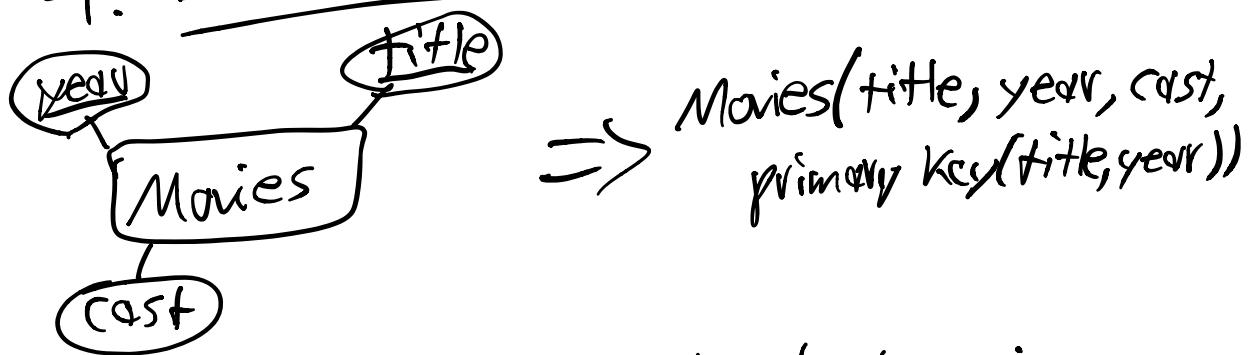
3. Supporting / Identifying Relationship:

A supporting relationship shows that a normal entity set

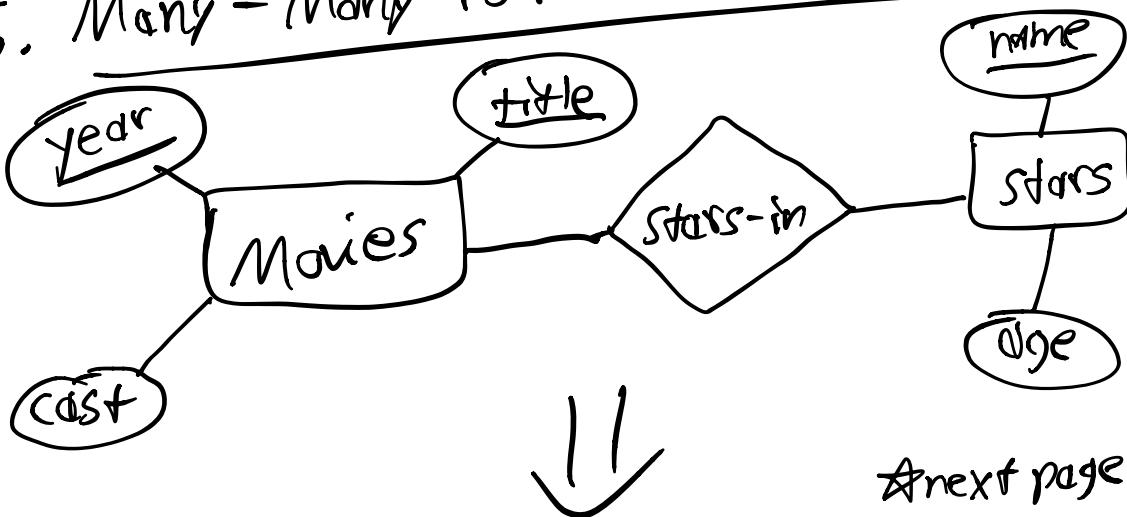
supports a weak entity set
 by offering the weak entity set
 its primary key in the form
 of a foreign key so that the weak
 entity set can use it as part of its
 primary key to uniquely identify
 tuples within its relation.

Translations:

4. Entity set-to relational Schema:

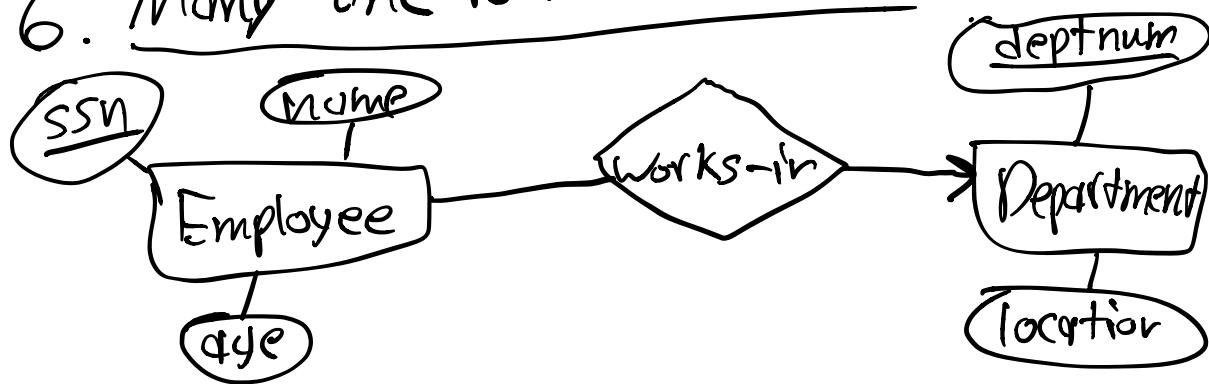


5. Many-Many to relational schema:



Movies(title, year, cast, primary key(title, year))
 Stars(name, age, primary key(name))
 Stars-in(title, year, star-name,
 primary key(title, year, star-name),
 foreign key(title, year) references
 Movies(title, year),
 foreign key(star-name) references
 Stars(name))

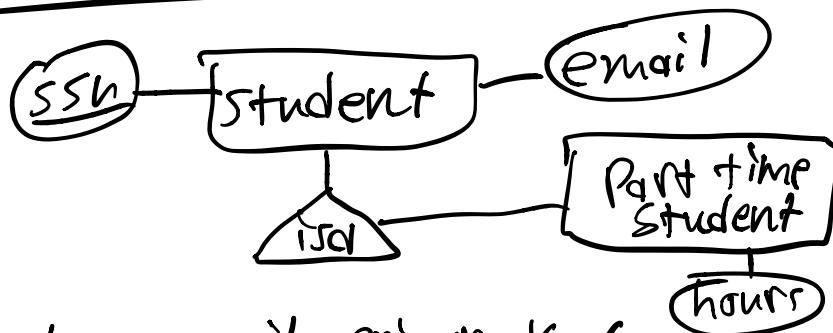
6. Many - One to relational schema:



Employee(ssn, name, age, deptNum,
 primary key(ssn), foreign key(deptNum)
 references Department(deptNum))

Department(deptnum, location, primary key(deptnum))

7. Subclass to relational schema:



$Student(ssn, email, \text{primary key}(ssn))$

$PartTimeStudent(ssn, hours, \text{primary key}(ssn),$
foreign key(ssn) references
 $Student(ssn))$

8. Weak entity set to relational schema:



$Teams(name, \text{primary key}(name))$

$Plays-On(name, number, teamName,$
(Player)
primary key(number, teamName),
foreign key(teamName),
referencing Teams(name))

★ This document contains
notes from book and answers
to weekly questions.