ER-to-Relational Mapping

Anne Denton

Department of Computer Science
North Dakota State University

Outline

Anne Denton ER-to-Relational Mapping

Table of Contents

Create a relation for each strong entity type

- Include all simple attributes
- Include only the simple components of composite attributes
- Don't include derived attributes!

ER Model

Student
sid {CK}
name
first_name
middle_initial
last_name
/age
birth_date

Relational Model

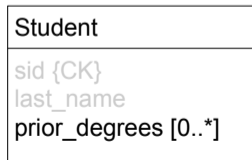| Student	
PK	sid
	first_name
	middle_initial
	last_name
	birth_date

Primary Key Choice

- Normally you add a surrogate key as primary key
  - Additional attribute for which values are assigned automatically (Autonumber in MS Access, Sequence in SQL)
  - If existing attributes should be considered as key, they can be considered as alternate keys by imposing a uniqueness constraint
- If you see reason to use a real world attribute consider
  - Attributes that can be NULL cannot be used as PK
  - Composite keys add compexity
  - Attribute with a text- or other storage-intensive domain are slow and not storage efficient
  - Don't pick an attribute for which values are likely to change (updating of PK values may not be well-supported)
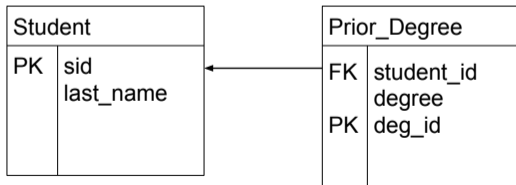
Mapping Multivalued attributes

- Create an additional relation for the attribute
- Add the PK of the corresponding entity as FK
- Primary key choices
    - A surrogate key, i.e. attribute added to act as PK
    - Conventional recommendation: Composite PK consisting of the attribute itself and the FK
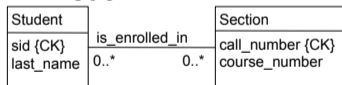
ER Model

Student
sid {CK}
last_name
prior_degrees [0..*]

Relational Model

Student	
PK	sid
	last_name

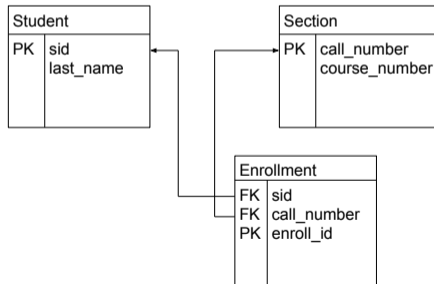| Prior_Degree	
FK	student_id
	degree
PK	deg_id

Table of Contents

Mapping binary many-to-many relationship

- Create a new relation that includes as FKs the two PKs of the participating entities
- If the relationship itself had attributes, those will be included
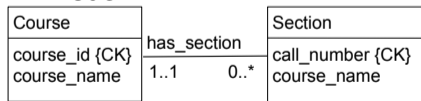
Relational Model

ER Model

Primary Key Choice for Added Relations

- Modern answer
  - Add a surrogate key
  - Important if you have reasons to assume that the relation will be referenced or if it is company policy
- Conventional answer
  - Create a composite PK from two FKs
  - Although it is not strictly necessary to define a PK, defining one is highly recommended
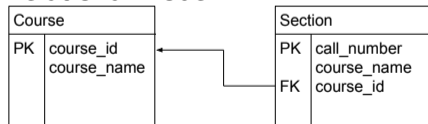
Mapping binary one-to-many relationship

- If there is any reason to believe that the cardinality may change in the future, map it like a many-to-many relation
- If it has partial participation, map it like a many-to-many relation
- If neither of those apply, you may not need an additional table
  - Choose the entity on the many-side and include as FK the PK of the other

ER Model

Course		Section
course_id {CK} course_name	has_section 1..1 0..*	call_number {CK} course_name

Relational Model

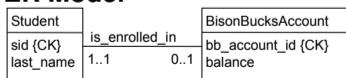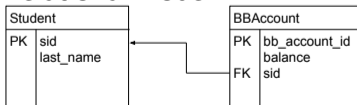| Course			Section	
PK	course_id course_name		PK	call_number course_name
			FK	course_id

Mapping binary one-to-one relationship

- If there is any reason to believe that the cardinality may change in the future, map it like a many-to-many relation
- If it has partial participation on both sides, map it like a many-to-many relation
- If neither of those apply, you may not need an additional table
  - Choose the entity that has mandatory participation and include as FK the PK of the other
  - If both entities have mandatory participation, and the relationship is not recursive, the tables can be merged

ER Model

Student		
sid {CK}	is_enrolled_in	
last_name	1..1	0..1

BisonBucksAccount
bb_account_id {CK}
balance

Relational Model

Student	
PK	sid
	last_name

BBAccount	
PK	bb_account_id
	balance
FK	sid

Anne Denton   ER-to-Relational Mapping

Table of Contents

Example: ER Model
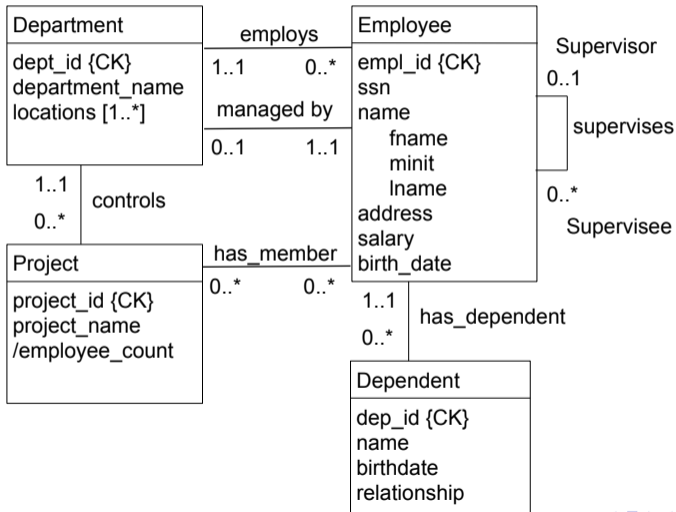
Example: Relational Model
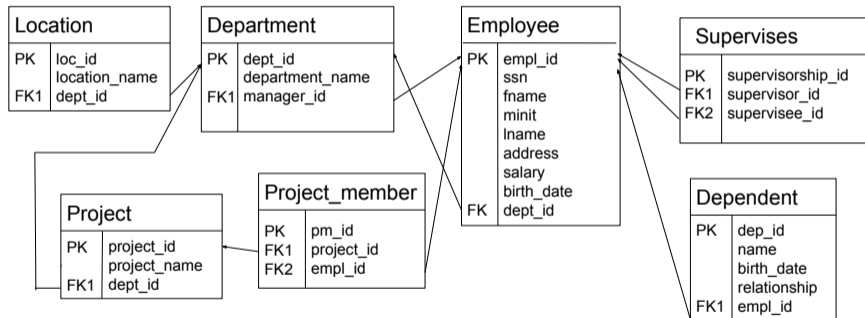
Question 1

In the ER-to-relational mapping of the example company database, Locations were mapped to a separate table.

1. Could "locations" have been an attribute in the department table?

Question 2 (Multiple answers can be correct)

In the ER-to-relational mapping of the example company database, Supervises was mapped to a separate table.

1. Could supervisor have become an attribute in the employee table, given the ER model?
2. Could supervisee have become an attribute in the employee table, given the ER model?

Question 3 (Multiple answers can be correct)

In the ER-to-relational mapping of the example company database, "dept_id" became an attribute in the Employee table.

1. According to the ER model, can that attribute be null?
2. Could "Department membership" of employees have become a separate table?

Question 4 (Multiple answers can be correct)

In the ER-to-relational mapping of the example company database, empl_id became an attribute in the Dependent table.

1. According to expectations of the ER model should empl_id be allowed to be null?
2. Could "Dependent relationship" have become a separate table?

Question 5

In the relational model of the example company database, only fname, minit, and lname are listed in the Employee table, but not the overall name.

1. Should the attribute "name" have been included?

Question 6

In the relational model of the example company database, employee_count is not listed in the Project table, although in the ER Model.

1. Should the attribute "employee_count" have been included relational model?

Question 7 (Multiple answers can be correct)

In the ER-to-relational mapping of the example company database, Project-member became a separate table.

1. Could the project_id have become an attribute of Employee instead?

2. Could the empl_id have become an attribute of Project instead?

3. Is the combination of project_id and empl_id be a candidate key of the Project_member table?

Question 8 (Multiple answers can be correct)

In the ER-to-relational mapping of the example company database, manager was made an attribute of department

1. According to the ER model, can the "manager_id" attribute be null?

2. Would it have been preferable to make "Managed department" an attribute in the Employee table?

3. If "Managed department" was an attribute of the Employee table, could it be null?

4. Could Management have become a separate table?