ER-to-Relational Mapping

Anne Denton

Department of Computer Science
North Dakota State University
Outline

1. Mapping of ER-model to Relational Model
   - Mapping Entities
   - Mapping Relationships

2. Example
   - Company Database Example
Table of Contents

1. Mapping of ER-model to Relational Model
   - Mapping Entities
   - Mapping Relationships

2. Example
   - Company Database Example
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**

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid {CK}</td>
</tr>
<tr>
<td>name</td>
</tr>
<tr>
<td>first_name</td>
</tr>
<tr>
<td>middle_initial</td>
</tr>
<tr>
<td>last_name</td>
</tr>
<tr>
<td>/age</td>
</tr>
<tr>
<td>birth_date</td>
</tr>
</tbody>
</table>

**Relational Model**

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
</tr>
<tr>
<td>first_name</td>
</tr>
<tr>
<td>middle_initial</td>
</tr>
<tr>
<td>last_name</td>
</tr>
<tr>
<td>birth_date</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid {CK}</td>
</tr>
<tr>
<td>last_name</td>
</tr>
<tr>
<td>prior_degrees [0..*]</td>
</tr>
</tbody>
</table>

#### Relational Model

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
</tr>
<tr>
<td>sid</td>
</tr>
<tr>
<td>last_name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prior_Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>FK</td>
</tr>
<tr>
<td>student_id</td>
</tr>
<tr>
<td>degree</td>
</tr>
<tr>
<td>PK</td>
</tr>
<tr>
<td>deg_id</td>
</tr>
</tbody>
</table>
Table of Contents

1. Mapping of ER-model to Relational Model
   - Mapping Entities
   - Mapping Relationships

2. Example
   - Company Database Example
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
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**

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td>course_id {CK}, course_name</td>
</tr>
<tr>
<td></td>
<td>has_section 1..1 0..*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section</td>
<td>call_number {CK}, course_name</td>
</tr>
</tbody>
</table>

**Relational Model**

<table>
<thead>
<tr>
<th>Table</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td>PK, course_id, course_name</td>
</tr>
<tr>
<td>Section</td>
<td>PK, call_number, course_id</td>
</tr>
<tr>
<td>Section</td>
<td>FK, course_name, course_id</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Student</th>
<th>sid {CK}</th>
<th>last_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BisonBucksAccount</td>
<td>is_enrolled_in</td>
<td>1..1</td>
</tr>
<tr>
<td>balance</td>
<td>0..1</td>
<td></td>
</tr>
</tbody>
</table>

**Relational Model**

<table>
<thead>
<tr>
<th>Student</th>
<th>sid</th>
<th>last_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>sid</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BisonBucksAccount</th>
<th>bb_account_id {CK}</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>BBAccount</th>
<th>bb_account_id</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>sid</td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

1. Mapping of ER-model to Relational Model
   - Mapping Entities
   - Mapping Relationships

2. Example
   - Company Database Example
Mapping of ER-model to Relational Model

Example: ER Model

Department
- dept_id {CK}
- department_name
- locations [1..*]
- managed by
  - employee_count
- controls

Employee
- empl_id {CK}
- ssn
- name
- fname
- minit
- lname
- address
- salary
- birth_date

Project
- project_id {CK}
- project_name
- /employee_count

Dependent
- dep_id {CK}
- name
- birthdate
- relationship

Supervisor
- supervises
- 0..*

Supervisee

Anne Denton
Example: Relational Model

**Location**
- **PK**: loc_id
- ** FK1**: location_name, dept_id

**Department**
- **PK**: dept_id
- ** FK1**: department_name, manager_id

**Employee**
- **PK**: empl_id
- **FK**: ssn, fname, minit, lname, address, salary, birth_date, dept_id

**Dependent**
- **PK**: dep_id
- **FK1**: name, birth_date, relationship

**Supervises**
- **PK**: supervisorship_id
- **FK1**: supervisor_id
- **FK2**: supervisee_id

**Project member**
- **PK**: pm_id
- **FK1**: project_id
- **FK2**: empl_id

**Project**
- **PK**: project_id
- **FK1**: project_name, dept_id
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.

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.

Should the attribute “employee_count” have been included in the relational model?
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?
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?