ER to Relational Conversion

© Department of Computer Science
Northern Illinois University
September 2000
Entity-Relationship Model

Person

Faculty

Staff

Student

Dept.

Address

Semester

Course

Grade

Section

is-related-to

(0,n)

(0,m)

(1,1)

(1,m)

has

(0,1)

(0,m)

(0,n)

(0,p)

(1,1)

(0,1)

(0,m)

(0,n)

(0,m)

(1,1)

(0,m)

(0,m)

(0,m)

(0,m)

(0,m)

(0,m)

(0,m)

credit-hours

teaches

enrolled-in

completed

within

reports-to

is-related-to
Entity-Relationship Model

**ENTITIES**

- **Person**
  - SSN (Identifier)
  - Name
  - Birth-Date
  - Beginning Date

- **Address**
  - Type (discriminator)
  - Street
  - City
  - State
  - Zip

- **Faculty**
  - SSN (Identifier)
  - Contact hours
  - Tenure status

- **Staff**
  - SSN (Identifier)
  - Position

- **Student**
  - SSN (Identifier)
  - Overall GPA
  - Major
Entity-Relationship Model

- Dept.
  - Dept-Code (ID)
  - Dept-Name
  - Dept-Address
  - Dept-Chair
- Course
  - Crse-Code (ID)
  - Crse-Title
  - Crse-Max-Credit-Hours
  - Crse-Var-Hours-Code
  - Crse-Fee
- Section
  - Sect-Code (ID)
  - Sect-Credit-Hours
  - Sect-Meet-Time
  - Sect-Meet-Day
- Semester
  - Sem-Yr (ID)
  - Sem-Session (ID)
Entity-Relationship Model

RELATIONSHIPS with attributes

- **Student enrolled-in Section**
  - **Credit-hours**
    - In a variable credit section this attribute would be used to hold the credit hours for which a specific student is enrolled.

- **Completed**
  - **Grade**
    - A student is allowed to take a course more than once.
ER to Relational Conversion

1. Consider all strong entities not subtypes (do not consider “date” entities here)
2. Consider sub-type entities
   - two methods
3. Consider weak entities
4. Consider One-to-many binary relationships
5 Consider many-to-many binary relationships
6 Consider relationships greater than binary (other than those involving “date” entities)
7 Consider relationships greater than binary involving a “date” entity
8 Consider recursive relationships
Consider All Strong Entities not Subtypes

- create a new relation
- name of the relation is the name of the entity
- attributes of entity become attributes of relation
- primary key of relation is entity identifier
Consider All Strong Entities not Subtypes
Consider Sub-type Entities (First Method)

- treat as a strong entity
- primary key is the entity identifier
- primary key is also a foreign key referencing the relation created from the supertype entity
Consider Sub-type Entities (First Method)
Consider Sub-type Entities (Second Method)

- combine into the relation created from the supertype entity as a composite attribute
Consider Sub-type Entities

- may combine the two methods within the conversion of the sub-types of a single ISA
Consider Weak Entities

• create a new relation
• name of the relation is the name of the weak entity
• attributes of entity become attributes of relation
Consider Weak Entities

- primary key of the relation is the concatenation of the primary key of the relation created from the strong entity and the discriminator of the weak entity
- the attribute which is the primary key of the relation created from the strong entity is also a foreign key
Consider Weak Entities

<table>
<thead>
<tr>
<th>SSN</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td></td>
</tr>
</tbody>
</table>

Person (1,1) has (1,m) Address

Table: Address

- SSN
- Type
- ...
Consider One-to-many Binary Relationships

• The primary key of the relation created from the “one” entity becomes a foreign key in the relation created from the “many” entity.
Consider One-to-many Binary Relationships

Faculty
(1,1)
teaches
(0,m)
Section

Faculty
SSN ……
PK

Section
Sect-ID Teach-SSN ……
PK FK
Consider Many-to-many Binary Relationships

• Create a new relation for the relationship whose primary key is the concatenation of the entity-ids of the related entities.

• The primary key attributes are also foreign keys into the relations created from the related entities.
Consider Many-to-many Binary Relationships

- The name of the new relation should reflect the relationship name.
- The intersection data of the relationship become non prime attributes of the relation.
Consider Many-to-many Binary Relationships

**Diagram:***

- **Student**
  - enrolled-in (0,m)
  - credit-hours

- **Section**

**Table:***

<table>
<thead>
<tr>
<th>SSN</th>
<th>Sect-ID</th>
<th>Credit-hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
<td></td>
</tr>
</tbody>
</table>
Consider Relationships Greater than Binary

- Create a new relation for the relationship.
- The primary key of the new relation depends upon the cardinalities of the relating entities.
A student used exactly one notebook for each course. He/she may be in many courses with many different notebooks. But each notebook belongs to one student and one course.
Consider Relationships Greater than Binary

Student \(\xrightarrow{(1,1)}\) Notebook \(\xrightarrow{(1,1)}\) Course

Completes

<table>
<thead>
<tr>
<th>SSN</th>
<th>Crse-ID</th>
<th>Notebook-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>

One Possibility
A student is assigned to one project within each course. A student may be working on many projects but each is for a different course. There may be many students assigned to a project but each project is for a given course.
Consider Relationships
Greater than Binary

Student

Course

Project

assigned-to

(1,m)

(1,1)

Student

Crse-ID

Proj-ID

SSN

PK

PK

FK

FK

FK

assigned-to

SSN

Crse-ID

Proj-ID

PK

PK

FK

FK

FK

PK
Consider Relationships
Greater than Binary

Each engineer working on a particular project has exactly one manager, but each manager of a project may manage many engineers, and each manager of an engineer may manage that engineer on many projects.

Proj-ID, Engin-ID $\rightarrow$ Mgr-ID
Consider Relationships greater than Binary

Manager -- (1,1) -- manages -- (1,m) -- Project
            
            Engineer -- (1,n) -- manages

PK                  PK                  FK
Proj-ID             Engin-ID            Mgr-ID
PK                  PK                  FK
FK                  FK                  FK
No functional dependencies between entities.

A student can complete many courses in a semester. A student may repeat a course in different semesters. A course can have many students enrolled in it in a semester.
Consider Relationships Greater than Binary

<table>
<thead>
<tr>
<th>Stud-ID</th>
<th>Crse-ID</th>
<th>Sem-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>
Consider Relationships Greater than Binary Including a Date Entity
Consider Relationships Greater than Binary Including a Date Entity

Notice Date is NOT a foreign key in the Visit table. (If it were, we would need to have a table of all dates used which is not practical.)
Consider Recursive Relationships

• Treat as the comparable type of relationship
  – one-to-many
    • use a foreign key
  – many-to-many
    • create a new relation for the relationship
Consider Recursive Relationships

Dept.

reports-to

(0,1)  (0,m)

Dept-ID
Reportsto-Dept
PK
FK
Consider Recursive Relationships

- Person
  - SSN
  - Name
  - (0,m)

- is-related-to

- Relative-SSN
  - SSN
  - Relative-SSN
  - PK
  - FK
### Relational Model

#### Person
- **SSN**
- **Name**
- **Staff**

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Address
- **SSN**
- **Type**

<table>
<thead>
<tr>
<th>SSN</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
</tbody>
</table>

#### Faculty
- **SSN**

<table>
<thead>
<tr>
<th>SSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
</tr>
</tbody>
</table>

#### Section
- **Sect-ID**
- **Teach-SSN**
- **Crse-ID**

<table>
<thead>
<tr>
<th>Sect-ID</th>
<th>Teach-SSN</th>
<th>Crse-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>

#### Completed
- **SSN**
- **Crse-ID**
- **Sem-Yr**
- **Sem-Sess**
- **Grade**

<table>
<thead>
<tr>
<th>SSN</th>
<th>Crse-ID</th>
<th>Sem-Yr</th>
<th>Sem-Sess</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td>PK</td>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
<td>FK</td>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>

---

**Section**

- **Sect-ID**
- **Teach-SSN**
- **Crse-ID**

<table>
<thead>
<tr>
<th>Sect-ID</th>
<th>Teach-SSN</th>
<th>Crse-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>
### Relational Model

#### enrolled-in

<table>
<thead>
<tr>
<th>SSN</th>
<th>Sect-ID</th>
<th>Credit-hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
<td></td>
</tr>
</tbody>
</table>

#### Dept

<table>
<thead>
<tr>
<th>Dept-ID</th>
<th>Reports-to-Dept</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>FK</td>
</tr>
</tbody>
</table>

#### is-related-to

<table>
<thead>
<tr>
<th>SSN</th>
<th>Relative-SSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>

#### Student

<table>
<thead>
<tr>
<th>SSN</th>
<th>Relative-SSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td></td>
</tr>
</tbody>
</table>

#### Course

<table>
<thead>
<tr>
<th>Crse-ID</th>
<th>Dept-ID</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>FK</td>
<td></td>
</tr>
</tbody>
</table>
