Introduction to Database Systems  
CSE 444

Lecture 07  
E/R Diagrams

October 10, 2007

Outline

• E/R diagrams  
  – Chapter 2

• From E/R diagrams to relations  
  – Chapters 3.2, 3.3

Database Design

• Why do we need it?  
  – Agree on structure of the database before deciding on a particular implementation.

• Consider issues such as:  
  – What entities to model  
  – How entities are related  
  – What constraints exist in the domain  
  – How to achieve good designs

• Several formalisms exists  
  – We discuss E/R diagrams

Entity / Relationship Diagrams

Objects ⟷ entities  
Classes ⟷ entity sets

Attributes are like in ODL.
Relationships: like in ODL except  
- first class citizens (not associated with classes)
- not necessarily binary

Keys in E/R Diagrams

• Every entity set must have a key
What is a Relation?

- A mathematical definition:
  - if A, B are sets, then a relation R is a subset of \( A \times B \)
  - \( A = \{1,2,3\}, \ B = \{a,b,c,d\}, \ A \times B = \{(1,a),(1,b), \ldots, (3,d)\} \)
  - \( R = \{(1,a), (1,c), (3,b)\} \)

- makes is a subset of Product \( \times \) Company:

Multiplicity of E/R Relations

- one-one:
- many-one
- many-many

Multi-way Relationships

How do we model a purchase relationship between buyers, products and stores?

Can still model as a mathematical set (how?)

Arrows in Multiway Relationships

Q: what does the arrow mean?

A: a given person buys a given product from at most one store
Q: How do we say that every person shops at at most one store?

A: cannot. This is the best approximation. (Why only approximation?)
Entity Set to Relation

Product(name, category, price)

<table>
<thead>
<tr>
<th>name</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>gizmo</td>
<td>gadgets</td>
<td>$19.99</td>
</tr>
</tbody>
</table>

Relationships to Relations

Makes(product-name, product-category, company-name, year)

<table>
<thead>
<tr>
<th>product-name</th>
<th>product-category</th>
<th>company-name</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>gizmo</td>
<td>gadgets</td>
<td>gizmoWorks</td>
<td>1963</td>
</tr>
</tbody>
</table>

No need for Makes. Modify Product:

<table>
<thead>
<tr>
<th>name</th>
<th>category</th>
<th>price</th>
<th>StartYear</th>
<th>companyName</th>
</tr>
</thead>
<tbody>
<tr>
<td>gizmo</td>
<td>gadgets</td>
<td>19.99</td>
<td>1963</td>
<td>gizmoWorks</td>
</tr>
</tbody>
</table>

Multi-way Relationships to Relations

Purchase(product-name, store-name, customer-ssn)

Subclasses

Some objects in a class may be special
- define a new class
- better: define a subclass

Products
  - Software products
  - Educational products

So --- we define subclasses in E/R
Understanding Subclasses

- Think in terms of records:
  - Product
    - field1
    - field2
  - SoftwareProduct
    - field1
    - field2
  - EducationalProduct
    - field1
    - field2
    - field3
    - field4
    - field5

Subclasses to Relations

<table>
<thead>
<tr>
<th>Product</th>
<th>Name</th>
<th>Price</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
<td></td>
</tr>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SoftwareProduct</th>
<th>Sw.Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>platforms</td>
</tr>
<tr>
<td>Gizmo</td>
<td>unix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EducationalProduct</th>
<th>Ed.Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Age Group</td>
</tr>
<tr>
<td>Gizmo</td>
<td>toddler</td>
</tr>
<tr>
<td>Toy</td>
<td>retired</td>
</tr>
</tbody>
</table>

Difference between OO and E/R inheritance

- OO: classes are disjoint (same for Java, C++)

No need for multiple inheritance in E/R

We have three entity sets, but four different kinds of objects.

Modeling UnionTypes With Subclasses

FurniturePiece

Person

Company

Say: each piece of furniture is owned either by a person, or by a company
Modeling Union Types with Subclasses
Say: each piece of furniture is owned either by a person, or by a company
Solution 1. Acceptable, imperfect (What’s wrong ?)

Constraints in E/R Diagrams
Finding constraints is part of the modeling process.
Commonly used constraints:
  - **Keys**: social security number uniquely identifies a person.
  - **Single-value constraints**: a person can have only one father.
  - **Referential integrity constraints**: if you work for a company, it must exist in the database.
  - **Other constraints**: peoples’ ages are between 0 and 150.

Single Value Constraints

Referential Integrity Constraints
Each product made by at most one company.
Some products made by no company

Each product made by exactly one company.
Other Constraints

What does this mean?

Weak Entity Sets

Entity sets are weak when their key comes from other classes to which they are related.

Notice: we encountered this when converting multiway relationships to binary relationships.