Entity-Relationship Model & Diagram

Todd S. Bacastow
IST 210: Organization of data
Design Principles

- Setting
  - client has (possibly vague) idea of what he/she wants.

- YOUR task
  - must design a DB that represents these thoughts and *only* these thoughts.
Design Principles (cont.)

1. Avoid redundancy
   - Wastes space and encourages inconsistency
   - Intuition: something is redundant if it could be hidden from view, and you could still figure out what it is from the other data

2. Faithfulness to requirements
   - Remember the design schema should enforce as many constraints as possible. Don’t rely on future data to follow assumptions.
   - Example: if registrar wants to associate only one instructor with a course, don’t allow sets of instructors and count on departments to enter only one instructor per course.
Data Modeling

- Data modeling is a formalism, with:
  - Notation for describing data
  - Set of operations for manipulating data

- Entity-Relationship Model (ER) Entity Relationship model:
  - a semantic model
  - attempts to capture the structure of data
Premises

- The real world consists of *entities* (entities = objects, things)
- Entities in one enterprise are *related*
- Database model = relevant entities + relationships among them
- Entities and relationships have *attributes*
E-R

- Features
  - Relatively easy to model real-world
  - Translates easily to relational data model

- Basic Concepts
  - Entity sets
  - Relationship sets
  - Attributes
Entities

- **Definition**
  - **Entity**: an object that is distinguishable from other objects.
    - a particular student, a particular book
  - **Entity set**: a set of entities *of the same type* that share the same set of properties
    - students, books
Entities

Examples of these entity types are:

- Person:
  - employee
  - student
  - customer
  - supplier
Entity Types

- Place:
  - city
  - state
  - county

- Object:
  - vehicle
  - building
  - airplane
  - animal
Entity Types

- Event:
  - purchase
  - sale
  - registration
  - game

- Concept:
  - account
  - course
  - work group
  - department
Weak Entities

- Weak entities are entities, but with a difference---
- Weak entities only exist because some other entity exists.
- For example:
  - two entities *employee* and *salary-history*
  - *salary-history* is a weak entity
  - the record of an employee’s salary history could only exist if a record of an employee also exists
  - E.g., Joe Smith’s salary history wouldn’t make much sense if Joe Smith doesn’t exist in the data base.
Finding Entities

- Entities are normally located through the various user views identified during analysis.

- Search for entities by:
  - looking for **nouns**
  - finding objects/things that have many instances (i.e., things stored in tables)
  - have descriptive/defining properties (attributes)
Attributes

Definition:

- Attribute: a descriptive property possessed by all members of an entity set.
- Attribute set: the set of all attributes possessed by members of an entity set.
  - *e.g.*, book = (author, title, publisher, barcode, *etc.*)
  - *e.g.*, student = (ID, surname, firstname, course, *etc.*)
  - *e.g.*, subject = (code, name, lecturer, timetable, *etc.*)
Attributes

Properties of Attributes

- Domain - the seat of permitted values for each attribute
- Type:
  - data type
  - simple vs. composite attributes
  - single-valued vs. multivalued
  - derived vs. stored
Simple & Composite Attributes

- Simple attributes cannot be decomposed without loss of meaning, e.g.,
  - Title: Mr., Ms., Mrs., Dr.

- Composite attributes can be further decomposed without loss of meaning, e.g.,
  - Name: first name + last name
  - Date: day + month + year
  - Address: street + city + state + zip
Single-Valued Attributes

- Single-valued attributes occur once only for each entity instance, e.g.
  - ID
  - Name
  - Department
Multi-Valued Attributes

- Multi-valued attributes can occur more than once for the same entity instance; e.g.
  - Sports person plays
  - Hobbies person likes
  - Skills person has achieved
  - Courses person has taken
  - Degrees person completed
Stored & Derived Attributes

- A derived attribute is one that can be calculated from other stored data
  - not usually stored as an attribute
  - occasionally stored to speed information/query requests when these occur frequently
- Maybe shown on ERD as attribute name within a dashed-line ellipse
Relationships

- Relationship: an association among 2 or more entities
  - E1: student - R: *takes* - E2: subject
  - *e.g.* Sara Smith *takes* IST210
Relationship sets

- Relationship set: a relation among entity sets.
  - *e.g.* (Smith, IST210) *takes*

- Relationship sets
  - have degree
  - may have attributes
  - *e.g.* final mark attribute for *takes* relationship
E-R Diagram
The symbols used in an ERD are not standardised so -

- diagrams may appear somewhat different depending upon who created them
- symbols used for our purposes will follow the format
Graphical Elements of the ER Diagram

Its components are:
- **rectangles** representing entity sets.
- **ellipses** representing attributes.
- **diamonds** representing relationship sets.

- **Entity**
  - Customers
  - Accounts

- **Attribute**
  - id
  - name

- **Relationship**
  - Customer Account
Basic Symbols

- Strong Entity
- Weak Entity
- Relationship
- Identifying Relationship
- Associative Entity
- Attribute
- Multivalued Attribute
- Derived Attribute
Relationship

- Connect two or more entity sets
- Represented by diamonds
- Think of the “value” of a relationship set as a table
- One row for each list of entities, one from each set, that are connected by the relationship
Finding Relationships

- Often represented by a **Verb**
- allows answers to questions not available through individual entity types

![Diagram](Diagram.png)

**EMPLOYEE** → **Has** → **DEPENDENT**

Weak Entity
Attributes & Relationships

- Relationship can have 1 or more attributes
  - depend on relationship
  - do not belong to the entities
  - for example the date on which a course was completed belongs to the relationship rather than either of the participating entities
  - shown in same way as entity attributes
Multiplicity or Cardinality

Many–many

Many–one

One–one
## Cardinality Symbology

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td>one</td>
<td><img src="image2" alt="Diagram" /></td>
<td>one or more</td>
</tr>
<tr>
<td><img src="image3" alt="Diagram" /></td>
<td>zero or one</td>
<td><img src="image4" alt="Diagram" /></td>
<td>zero or more</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="Diagram" /></td>
<td>1-1 Relationship</td>
<td>Only one entity of the first (left) type is related to an entity of the other (right) type.</td>
</tr>
<tr>
<td><img src="image6" alt="Diagram" /></td>
<td>n-1 Relationship</td>
<td>Many entities of the first type are related to an entity of the other side.</td>
</tr>
</tbody>
</table>
Example

- In the relation rented to, a videotape is related to 0 or 1 customer (it is either rented out or not rented out).
- In the relation rented, a customer is related to 0 or more videotapes (they can rent 0 or more videotapes).
One-to-One

"Each EMPLOYEE may be a DRIVER".
One-to-Many

"Each CUSTOMER may have several ORDERS".
Many-to-Many

"Some **ORDERS** may contain many parts (**INVENTORY**),
and some parts may be used in many **ORDERS**."
Recursive

"EMPLOYEES may supervise any other EMPLOYEES"
Common Constraints

- Keys → we will discuss next
- Single-value constraints → must have a certain value
  - Null value
- Referential integrity constraints → “broken link to a web page”
- Domain constraints → a negative account balance
- General constraints → limits you set
Keys

**Primary key**

**Foreign key**

**Primary key**

**Student**

- ID: 1
  - Last: Wood
  - First: Bob
  - Grade: C
  - Class: Ist 357

- ID: 2
  - Last: Kent
  - First: Chuck
  - Grade: B
  - Class: Ist 115

- ID: 3
  - Last: Smith
  - First: Jane
  - Grade: A
  - Class: Ist 357

- ID: 4
  - Last: Boone
  - First: Dan
  - Grade: B
  - Class: Ist 357

**Class**

- Name: Ist 357
  - #Stud: 48
  - Instructor: Jones

- Name: Ist 115
  - #Stud: 120
  - Instructor: Brower

- Name: Ist 20
  - #Stud: 120
  - Instructor: Mennis

2/1/2004
Keys

- A *key* is a set of attributes whose values can belong to at most one entity.
- In E-R model, *every* E.S. must have a key.
- More than one key: a set of attributes is the “designated” key.
- In E-R diagrams: underline all attributes of the designated key.
A Multi-attribute Key

- dept+number (e.g. IST 210) form a key for courses.
- Possibly, hours+room also forms a key. But we have not designated it as such.
Weak Entities & Keys

- Weak entity sets:
  - Does not have a primary key
  - are meaningful only in the context of a relationship with an *owning* strong entity set
We Made it!