Entity-Relationship (ER) Model

Problem → ER Model → Tables

- An object-oriented approach
- A visual representation of the design – ER Diagram
- Easily converted to relational model

Example: Problem Description

- Student
  - id, name, address
- Department
  - name
- Classes
  - code, name, quarter, section number
- Class offerings and enrollment

Example: ER Diagram

Entity Set and Attributes

- Entity Set is similar to class in an OO language
- Attributes are the properties of an entity set
  - Similar to the class fields in an OO language
  - Must have simple values like numbers or strings, i.e. cannot be collection or composite type
Keys

- A key is an attribute or a set of attributes that uniquely identify an entity in an entity set.
- Each entity set must have a key.
- If there are multiple keys, choose one of them as the primary key.

Types of Relationships

- Many-to-Many
- Many-to-One / One-to-Many
- One-to-One

Many-to-Many Relationship

- Each entity in E₁ can be related to many entities in E₂.
- Each entity in E₂ can be related to many entities in E₁.

Many-to-One Relationship

- Each entity in E₁ can be related to one entity in E₂.
- Each entity in E₂ can be related to many entities in E₁.

One-to-One Relationship

- Each entity in E₁ can be related to one entity in E₂.
- Each entity in E₂ can be related to one entity in E₁.

Relationship Type Examples

- Students and classes??
- Departments and classes??
- Person and Favorite movie??
Relationship Types in ER Diagram

- Students Take Classes
- Departments Offer Classes
- Person Favorite Movies

An arrow is used to indicate the “one” side

A Closer Look at “One” and “Many”

- **One**
  - 0 or 1
  - Exactly 1 ➔ Referential Integrity
- **Many**
  - 0..N
  - 1..N
  - N..M (*Example??*)

Referential Integrity in ER Diagram

- Departments Offer Classes
- Person Favorite Movies

An circular arrow is used to indicate “Exactly 1”

One vs. Exactly One

- Both lead to foreign key constraint in SQL
  - **One**: foreign key + NULL
  - **Exactly one**: foreign key + NOT NULL
- It’s usually not too important to distinguish the two in ER design

Design Example: Bank Database

Design a database for a bank to keep track of customers and accounts. Each account has an id, and a balance; each customer has a name and address. A customer can own multiple accounts, and an account can be jointly owned by multiple customers.

ER Design (I)

- **Step 1**: identify entity sets, attributes, and relationships.
- **Tips**:
  - **Nouns** tend to be entity sets or attributes
    - Attribute: simple data that can be represented by a single value
    - Entity Set: composite data
  - **Verbs** tend to be relationships
ER Design (II)

- Step 2: determine relationship types
- Step 3: complete entity sets
  - Identify/create keys
  - Add additional attributes if necessary
- Some common problems:
  - No keys
  - Wrong relationship types
  - Collection/composite attributes

Some Common Problems in ER Design

Grades

- Store the grades the students received for their classes
- A grade is a single letter A, B, C, D, or F

Relationship Attributes

- Sometimes it's useful to attach an attribute to a relationship.

Other Relationship Attribute Examples

- From Relationship with Attributes to Entity Set ...
  - Some variations of ER model does not allow relationships to have attributes
... From Relationship with Attributes to Entity Set

- If something needs an attribute, it probably should be an entity set

Grades as an Entity Set

- Need to store more than just the letter

Multiway Relationship

- Why there is an arrow pointing to Grades?

“Arrows” in Multiway Relationships

- In multiway relationships, an arrow points to an entity set E means that if we select one entity from each of the other entity sets in the relationship, those entities are related to at most one entity in E.

Convert Multiway Relationship to Binary Relationship

- A multiway relationship can always be converted to binary relationships by replacing the multiway relationship with an entity set

Compare the Ways to Model Grades

- A. Relationship attribute
- B. Entity set attribute
- C. Entity set in a multiway relationship
- D. Entity set in a binary relationship
Employees and Supervisors

- Each employee has a supervisor
- A supervisor is an employee

Roles

- An entity set may appear in the same relationship more than once.
- Label the edges with names called Roles

Players and Teams

- What's the key for Players??

Weak Entity Set

- Entity set \( E \) is said to be weak if in order to identify entities of \( E \) uniquely, we need to follow one or more many-one relationships from \( E \) and include the key of the related entities from the connected entity sets.

Weak Entity Sets in ER Diagram

- The key of a weak entity set consists of its own key attributes and the key attributes of the supporting set

From Weak to Strong

- We can usually create unique IDs for entity sets
### Subclass

- **Entities**: Users, Students

  - **Attributes**: id, name, email

  - **Relationships**: Superclass must have all the key attributes

### When to Use (and When Not to Use) Subclass

- **A**: salaried employees and hourly employees
- **B**: administrator users and regular users
- **C**: pop songs and country songs
- **D**: beer and wine

### Summary of ER Diagram

- **Entity Set**
  - Attributes, key
  - Weak entity set

- **Relationship**
  - Many-to-Many, Many-to-One, One-to-One
  - Attributes
  - Multiway and binary relationships
  - Subclass

### Design Example 1: Restaurant

<table>
<thead>
<tr>
<th>#</th>
<th>Some Restaurant</th>
<th>Subtotal:</th>
<th>17.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Boiled Pork Wanton</td>
<td>4.95</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Dumpling w/Crabmeat</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Beef Noodle Soup</td>
<td>20.74</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Total</td>
<td>41.74</td>
</tr>
</tbody>
</table>

**GST**: 0.99

**Date**: Jul 09, 2000

**Time**: 03:07PM

**Server**: John

**Time**: 03:07PM

**Order**: #42

**Printed By**: Cashier

### Design Example 2: Folders and Files

- **Folder Structure**:
  - C:
    - \WINNT
    - \Document and Settings
    - \Program Files
      - \yysun
    - \database material
    - \web material
        - file1
        - file2
        - file3
        - file4

### Design Example 3: Price Changes

- **Diagram**:
  - Products
  - id, description, price
  - Price of a product X
  - $ Price of a product X
  - time

**Question**: What if we want to model price changes??