Designing Tables Is Not Easy

Problem in Real World

Tables in RDBM

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-Relationship Model

CS422 Principles of Database Systems

Chengyu Sun
California State University, Los Angeles

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??
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

- 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

Example: 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.
Some variations of ER model does not allow relationships to have attributes.

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

**Example: More about Grades**

Make Grades an entity set.

**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
Example: 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

Example: 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
Summary of ER Diagram

- **Entity Set**
  - Attributes, key
  - Weak entity set
- **Relationship**
  - Many-to-Many, Many-to-One, One-to-One
  - Attributes
  - Multiway relationship
  - Subclass

Relational Model

- Proposed by Edgar F. Codd in early 1970's
- Data is stored in tables (a.k.a. relations)
- All major database systems these day are relational

<table>
<thead>
<tr>
<th>student_id</th>
<th>first_name</th>
<th>last_name</th>
<th>birthday</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000001</td>
<td>John</td>
<td>Doe</td>
<td>1970-1-1</td>
</tr>
<tr>
<td>2000002</td>
<td>Jane</td>
<td>Doe</td>
<td>1971-1-1</td>
</tr>
<tr>
<td>2000003</td>
<td>Tom</td>
<td>Smith</td>
<td>1962-2-2</td>
</tr>
</tbody>
</table>

About Relational Model

- Attributes must be of simple type
- No order among attributes
- No order among records

Table (Relation)

- Attributes (fields, columns)
  - id
  - name

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>John</td>
</tr>
<tr>
<td>1001</td>
<td>Jane</td>
</tr>
</tbody>
</table>

Table and Database Schema

- **Table schema**
  - Name of the table, and the names and types of the attributes
  - E.g. `students(id:integer, name:string)`
    or just `students(id, name)`
- **Database schema**
  - Schemas of all the tables in the database
Basic Rules of ER to Relational Conversion

- A entity set is converted to a table
- A many-to-many relationship is also converted to a table, including
  - Its own attributes
  - Key attributes from the associated entity sets
- A many-to-one relationship is merged into the “many” side with a foreign key to the “one” side

Conversion Example: ER Diagram

Conversion Example: Relational Schema

Students( id, name, address )
Departments( name )
Classes( code, name, quarter, section, department_name )
Takes( student_id, code, quarter, section )

More Conversion Examples

Special Cases of Conversion

- One-to-One relationship
- Multiway relationship
- Weak entity set
- Subclass

Converting One-to-One Relationship ...
... Converting One-to-One Relationship

Which one of the following makes more sense??

Faculty( id, name, chair_of_department )
Departments( id, name )

or

Faculty( id, name )
Departments( id, name, department_chair )

Converting Multiway Relationship

Should this relationship be treated as many-to-many or many-to-one??

Converting Weak Entity Set ...

The table for a weak entity set includes its complete key as well as its own non-key attributes
A supporting relationship is redundant and yields no relation

Converting Subclass ...

Object-oriented approach
- One table per class
- Each entity belongs to exact one table

ER approach
- One table per class
- Each entity may appear in multiple tables

NULL approach
- One table per class hierarchy
Object-Oriented Approach

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>John</td>
<td></td>
</tr>
</tbody>
</table>

Users

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Jane</td>
<td>212345678</td>
</tr>
</tbody>
</table>

Students

ER Approach

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>John</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>Jane</td>
<td></td>
</tr>
</tbody>
</table>

Users

<table>
<thead>
<tr>
<th>user_id</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>212345678</td>
</tr>
</tbody>
</table>

Students

NULL Approach ...

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>John</td>
<td>NULL</td>
</tr>
<tr>
<td>1001</td>
<td>Jane</td>
<td>212345678</td>
</tr>
</tbody>
</table>

Users

... NULL Approach

Discriminator field

<table>
<thead>
<tr>
<th>id</th>
<th>user_type</th>
<th>name</th>
<th>cin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>staff</td>
<td>John</td>
<td>NULL</td>
</tr>
<tr>
<td>1001</td>
<td>student</td>
<td>Jane</td>
<td>212345678</td>
</tr>
</tbody>
</table>

Users

Comparison of Subclass Conversion Approaches

- Constraints and data integrity
- Query performance

Q1: find the number of users
Q2: find the number of students

Summary of ER to Relational Conversion

- Basic rules
  - Entity set
  - Many-to-many relationship
  - Many-to-one relationship
- Special cases
  - One-to-one relationship
  - Multiway relationship
  - Weak entity set
  - Subclass
Common Problems in ER Diagram

- Student (id, name, addr)
- Class (id, code)
- Take (student_id, class_id)
- Offer (id, name, department_id)

Common Mistakes of ER Design

- ER diagram
  - Missing arrows
  - Missing keys
  - Redundant foreign keys
- ER to relational conversion
  - Missing tables for many-to-many relationships

Variations of ER Model

- Rule differences
  - Relationships cannot have attributes
  - Anything that have attributes should be an entity set
  - Simplifies ER diagram
  - No multiway relationship
  - Better mapping to OO languages
- Notational differences
  - Old school style
  - Tool friendly style

Old School ER Diagram

- Highlight three distinct components of a diagram
- Difficult to draw
- Can look messy even for a relatively simple schema

Tool Friendly ER Diagram

- Relationships cannot have attributes, and no multiway relationships
- Easier to draw, especially using OO design tools
- Looks cleaner

Design Example 1: Restaurant

- Terminal ID: NC2HHRY
- Merchant ID: 4992414532566624

- VISA
  - AUTH:00559B
  - Inv:000032
- Sale
  - Batch: 0000244
  - Date: JUN 17, 06 Time: 18:44
  - AUTH:00559B
- Base: $36.70
- Tip:
- Total: $40.70
- Chengyu Sun
Design Example 2: Folders and Files

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

Design Example 3: Price That Changes

What if we want to model price that changes??

$ Price of a product X
 time