Carnegie Mellon Univ.
Dept. of Computer Science
15-415/615 - DB Applications

C. Faloutsos
Lecture#2: E-R diagrams

Problem
• Develop an application for U.G. admin:
  – Student info
  – Who-takes-what class
  – Class rosters
  – Transcripts
• How do you proceed?
  – (Which role(s) are you playing?)

Database Design
• Requirements Analysis
• Conceptual Design
• Logical Design
• Schema Refinement
• Physical Design
• Security Design
Database Design

- Requirements Analysis
- **Conceptual Design**
- Logical Design
- Schema Refinement
- Physical Design
- Security Design

  - user’s needs
  - high level (ER)
  - Tables
  - Normalization
  - Indices etc
  - Access controls

Overview

- concepts
  - Entities
  - Relationships
  - Attributes
  - Specialization/Generalization
  - Aggregation
  - ER modeling questions

Tools

- Entities (‘entity sets’)
- Relationships (‘rel. sets’)
- and mapping constraints
- attributes
Example

Students, taking courses, offered by instructors; a course may have multiple sections; one instructor per section

nouns -> entity sets
verbs -> relationship sets

STUDENT

name
ssn...

INSTRUCTOR

issn

primary key =
unique identifier ->

underline

STUDENT

c-id

c-name

COURSE

primary key =
unique identifier ->

underline

but: sections of course (with different instructors)?
but: s-id is not unique... (see later)

Q: how to record that students take courses?
Cardinalities

- 1 to 1 (example?)
- 1 to N
- N to M
Cardinalities

Book’s notation vs 1 to N notation

<table>
<thead>
<tr>
<th>Cardinalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
</tr>
<tr>
<td>has 1</td>
</tr>
<tr>
<td>CAPITAL</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>PERSON</td>
</tr>
<tr>
<td>owns N</td>
</tr>
<tr>
<td>CAR</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>STUDENT</td>
</tr>
<tr>
<td>takes M</td>
</tr>
<tr>
<td>SECTION</td>
</tr>
</tbody>
</table>

Book’s notation:

<table>
<thead>
<tr>
<th>Cardinalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
</tr>
<tr>
<td>has CAPITAL</td>
</tr>
<tr>
<td>PERSON</td>
</tr>
<tr>
<td>owns CAR</td>
</tr>
<tr>
<td>STUDENT</td>
</tr>
<tr>
<td>takes SECTION</td>
</tr>
</tbody>
</table>

Book’s notation vs 1 to N notation

<table>
<thead>
<tr>
<th>Cardinalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
</tr>
<tr>
<td>has 1</td>
</tr>
<tr>
<td>CAPITAL</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>PERSON</td>
</tr>
<tr>
<td>owns N</td>
</tr>
<tr>
<td>CAR</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>STUDENT</td>
</tr>
<tr>
<td>takes M</td>
</tr>
<tr>
<td>SECTION</td>
</tr>
</tbody>
</table>

Book’s notation vs 1 to N notation
Weak entities

- ‘section’ has no unique-id of its own! (?)

COURSE
\[ c\text{-id} \]
SECTION
\[ N \]
has
\[ I \]
COURSE
\[ s\text{-id} \]

Weak entities

- ‘weak’ entities: if they need to borrow a unique id from a ‘strong entity’ - thick box.
- ‘c-id’ + ‘s-id’: unique id for SECTION
- partial key (eg., ‘s-id’) - dashed underline
- identifying relationship (eg., ‘has’)

More details

- self-relationships - example?
More details

• self-relationships - example?

\[ \text{EMPLOYEE} \overset{\text{manages}}{\rightarrow} \text{EMPLOYEE} \]

More details

• 3-way and k-way relationships?

More details

• 3-way and k-way relationships? Rare, but possible:

\[ \text{EMPLOYEE} \overset{\text{uses}}{\rightarrow} \text{TOOL} \overset{\text{uses}}{\rightarrow} \text{PROJECT} \]
Overview

• concepts
  – Entities
  – Relationships
  – Attributes
  – Specialization/Generalization
  – Aggregation
  – ER modeling questions

More details - attributes

• key (or primary key): unique identifier
• underlined, in the ER diagram
• [not in textbook - FYI:
  – multivalued or set-valued attributes (e.g., ‘dependents’ for
    EMPLOYEE)
  – derived attributes (e.g., 15% tip)
]
Specialization

- eg., students: part time (#credit-hours) and full time (major)

Observations

- Generalization: exact reverse of ‘specialization’
- attribute inheritance
- could have many levels of an IS-A hierarchy

More details

- Overlap constraints
- Covering constraints
More details

- **Overlap constraints**
  - can an entity belong to both ‘B’ and ‘C’?
- **Covering constraints**
  - can an ‘A’ entity belong to neither ‘B’ nor ‘C’?

More details

- **Overlap constraints - examples?**

More details

- **Covering constraints - examples?**
Overview

• concepts
  – Entities
  – Relationships
  – Attributes
  – Specialization/Generalization
  – Aggregation
  – ER modeling questions

Aggregation

• computer model (w/ CPU and HD)
  • and Maker (eg., Dell, HP)

• treat a relationship as an entity
  • used to express a relationship among relationships
Overview

- concepts
  - Entities
  - Relationships
  - Attributes
  - Specialization/Generalization
  - Aggregation
- ER modeling questions

Conceptual design

- Entity vs attribute
- Entity vs relationship
- Binary or ternary relationships?
- Aggregation?

Entity vs. attribute

- Entity EMPLOYEE (w/ emp#, name, job_code, ...)
- Q: How about ‘spouse’ - entity or attribute?
- Q: How about ‘dependents’?
Entity vs. attribute

- Entity EMPLOYEE (w/ emp#, name, job_code, ...)
- Q: How about ‘spouse’ - entity or attribute?
- A: probably, 'attribute' is enough
- Q: How about ‘dependents’?
- A: Entity - we may have many dependents

Entity vs. Relationship

- STUDENT
- SECTION
- takes

- STUDENT
- TAKES
- N

- SECTION
- 1

Binary vs Ternary Relationships

- usually, binary relationships are ‘cleaner’:
Binary vs. Ternary Relationships

If each policy is owned by just 1 employee:

Policies

Dependents

Covers

Employees

Key constraint on Policies would mean policy can only cover 1 dependent!
**Binary vs. Ternary Relationships**

If each policy is owned by just 1 employee:

- **Bad design**

Key constraint on Policies would mean policy can only cover 1 dependent!

- **Better design**

What are the additional constraints in the 2nd diagram?

**Binary vs Ternary Rel.**

- But sometimes ternary rel. can not be replaced by a set of binary rel’s:

**Binary vs. Ternary Relationships (Contd.)**

Why is it bad?
Binary vs. Ternary Relationships (Contd.)

- S “can-supply” P, D “needs” P, and D “deals-with” S does not imply that D has agreed to buy P from S.
- How do we record qty?

Not in textbook:
in practice, often:

---

Not in textbook:
in practice, often:
Binary vs. Ternary Relationships (Contd.)

Not in textbook: in practice, often:

Ternary vs. aggregation

- use aggregation, if we want to attach a relationship to a relationship
- (see book for example)
- (in practice, again we create a unique-id and resort to binary relationships)

Ternary vs. aggregation

- How would you handle this case?
Ternary vs. aggregation

• How would you handle this case?

```
+-------------------+
| COMP. MODEL       |
| HD                |
+-------------------+
    |               |
    | MAKER          |
    +-------------------+
```

Ternary vs. aggregation

• How would you handle this case?

```
+-------------------+
| COMP. MODEL       |
| CPU              |
+-------------------+
    |               |
    | HD             |
    +-------------------+
        |               |
        | MAKER          |
```

Ternary vs. aggregation

• How would you handle this case?

```
+-------------------+
| COMP. MODEL       |
| HAS_CPU           |
| CPU              |
+-------------------+
    |                 |
    | HD              |
    +-------------------+
        |               |
        | MAKER          |
```

```
Summary

• E-R Diagrams: a powerful, user-friendly tool for data modeling:
  – Entities (strong, weak)
  – Attributes (primary keys, discriminators, derived, multivalued)
  – Relationships (1:1, 1:N, N:M; multi-way)
  – Generalization/Specialization; Aggregation

Summary - cont’d

- (strong) entity set
- weak entity set
- relationship set
- identifying rel. set for weak entity
- attribute
- primary key
- partial key
Summary - cont’d

- Cardinalities
- Partial/Total

<table>
<thead>
<tr>
<th>N</th>
<th>M</th>
</tr>
</thead>
</table>

- IS-A

- Aggregation

(Not in textbook - FYI)