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

C. Faloutsos - A. Pavlo  
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  
  user’s needs 
- Conceptual Design  
  high level (ER) 
- Logical Design  
  Tables 
- Schema Refinement  
  Normalization 
- Physical Design  
  Indices etc 
- Security Design  
  Access controls
Problem’

Maintain

• Develop an application for U.G. admin:
  – Student info
  – Who-takes-what class
  – Class rosters
  – Transcripts
• If you are the *new* DBA, what would you rather inherit:

This or this?

```
drop table if exists student;
cREATE TABLE Student
  (ssn fixed,
   name char(20));
drop table if exists takes;
cREATE TABLE Takes
  (ssn fixed,
   cid char(10),
   grade fixed);
```

True story

• Health insurance company
• Wants to catch (some of the abundant) fraud
• Schema:
  – patients, visit doctors, get medicine,
  – Doctors perform operations, …
  – Nurses monitor patients, …
  – etc etc
• Q: How many tables do you think it spans?

True story

• Schema:
  – patients, visit doctors, get medicine,
  – Doctors perform operations, …
  – Nurses monitor patients, …
  – etc etc
• Q: How many tables do you think it spans? 10? 20? 30?
**True story**

- Schema:
  - patients, visit doctors, get medicine,
  - Doctors perform operations, …
  - Nurses monitor patients, …
  - etc etc
- Q: How many tables do you think it spans?
  - 10? ✗ 30?
- A: **120 PAGES** of schema

**Motivation & upcoming conclusion:**

- E-R diagrams are excellent documentation tools

```
Student
   Takes
      Course
```

**Overview**

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

**Tools**

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

```
N
M
```

```
P
```
Example

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

nouns -> entity sets
verbs -> relationship sets

Example

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

nouns -> entity sets
verbs -> relationship sets

Example

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

nouns -> entity sets
verbs -> relationship sets

Example

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

nouns -> entity sets
verbs -> relationship sets
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:

- **COUNTRY** → **has** → **CAPITAL**
- **PERSON** → **owns** → **CAR**
- **STUDENT** → **takes** → **SECTION**

Book’s notation vs 1 to N notation

<table>
<thead>
<tr>
<th>1 to N notation</th>
<th>Book’s notation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COUNTRY</strong> → 1 <strong>has</strong> → 1 <strong>CAPITAL</strong></td>
<td><strong>COUNTRY</strong> → 1 <strong>has</strong> → 1 <strong>CAPITAL</strong></td>
</tr>
<tr>
<td><strong>PERSON</strong> → N <strong>owns</strong> → M <strong>CAR</strong></td>
<td><strong>PERSON</strong> → 1 <strong>owns</strong> → N <strong>CAR</strong></td>
</tr>
<tr>
<td><strong>STUDENT</strong> → M <strong>takes</strong> → N <strong>SECTION</strong></td>
<td><strong>STUDENT</strong> → M <strong>takes</strong> → N <strong>SECTION</strong></td>
</tr>
</tbody>
</table>

‘Total/partial’ participation

<table>
<thead>
<tr>
<th>Total, total</th>
<th>??:1</th>
<th>??:N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COUNTRY</strong> → 1:1 <strong>has</strong> → 1:1 <strong>CAPITAL</strong></td>
<td><strong>COUNTRY</strong> → 1:1 <strong>has</strong> → 1:1 <strong>CAPITAL</strong></td>
<td></td>
</tr>
<tr>
<td><strong>PERSON</strong> → ??:1 <strong>owns</strong> → ??:N <strong>CAR</strong></td>
<td><strong>PERSON</strong> → ??:1 <strong>owns</strong> → ??:N <strong>CAR</strong></td>
<td></td>
</tr>
<tr>
<td><strong>STUDENT</strong> → ??:N <strong>takes</strong> → ??:M <strong>SECTION</strong></td>
<td><strong>STUDENT</strong> → ??:N <strong>takes</strong> → ??:M <strong>SECTION</strong></td>
<td></td>
</tr>
</tbody>
</table>
‘Total/partial’ participation

total, total

PERSON 1:1 owns COUNTRY 1:1 has CAPITAL

partial, total

STUDENT 2:N takes CAR 0:N

??

PERSON CAR

owns partial, total

STUDENT 1:N takes SECTION 0:M

‘Total/partial’ participation

Is it ‘legal’?

partial, total

PERSON 1:1 owns CAR 0:N

‘Total/partial’ participation

Is it ‘legal’?

partial, total

PERSON 1:1 owns CAR 0:N

NO! why not?
Subtle concept: Weak entities

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

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’)

Weak entities

- Other example(s) of weak entities?

- Other example(s) of weak entities?
More details

• self-relationships - example?

More details

• self-relationships - example?

More details

• self-relationships - example?

More details

• 3-way and k-way relationships?
More details

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

```
EMPLOYEE uses TOOL
  N               M
  PROJECT P
```

More details

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

```
?? uses ??
  N               M
  ?? P
```

Other cases?

More details

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

```
user reviews keyword
  N               M
  app P
```

App-store/amazon reviews

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 (eg., ‘dependents’ for EMPLOYEE)
  - **derived** attributes (eg., 15% tip)
]

Overview

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

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?

No overlap

\[
\begin{array}{c}
A \\
B \\
C 
\end{array}
\]

with overlap

\[
\begin{array}{c}
A \\
B \\
C 
\end{array}
\]

More details

- Covering constraints - examples?

Total coverage

\[
\begin{array}{c}
A \\
B \\
C 
\end{array}
\]

Partial coverage

\[
\begin{array}{c}
A \\
B \\
C 
\end{array}
\]
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’?
- A: probably, ‘attribute’ is enough
- A: Entity - we may have many dependents

Entity vs. Relationship

- STUDENT
  - takes
  - SECTION

- TAKES
  - N
  - M
  - OR
  - N
  - I

STUDENT
  - 1
  - N
Binary vs Ternary Relationships

• usually, binary relationships are ‘cleaner’:

If each policy is owned by just 1 employee:

Bad design

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!

- **What are the additional constraints in the 2nd diagram?**

Better design

Binary vs Ternary Rel.

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

### 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?

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

Not in textbook: in practice, often:

\[
\begin{align*}
\text{Parts} & \rightarrow \text{Contract} & \text{Departments} \\
\text{Suppliers} & \\
\text{qty} & \\
\end{align*}
\]

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?

Ternary vs. aggregation

• How would you handle this case?

Ternary vs. aggregation

• How would you handle this case?
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

attribute
primary key
partial key

cardinalities
N
M

l:h
l’:h’
cardinalities with limits

(not in textbook - FYI)
Summary - cont’d

IS-A

aggregation