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
- Conceptual Design
- Logical Design
- Schema Refinement
- Physical Design
- Security Design

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

Problem

- 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?

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? 20? 30?

A: 120 PAGES of schema
Motivation & upcoming conclusion:

• E-R diagrams are excellent documentation tools

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

s

INSTRUCTOR

issn

primary key =
unique identifier ->

STUDENT name ssn

COURSE c-id c-name

INSTRUCTOR issn

but: sections of course (with different instructors)?

COURSE

SECTION s-id

INSTRUCTOR

but: s-id is not unique... (see later)

CMU SCS 15-415/615
Q: how to record that students take courses?
Cardinalities

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

Cardinalities

- COUNTRY has 1 CAPITAL
- PERSON owns N CAR
- STUDENT takes M SECTION
Cardinalities

Book's notation:
- COUNTRY \( \rightarrow \) CAPITAL
- PERSON \( \rightarrow \) CAR
- STUDENT \( \rightarrow \) SECTION

Cardinalities

Book's notation vs 1 to N notation

<table>
<thead>
<tr>
<th>Relation</th>
<th>Book's notation</th>
<th>1 to N notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
<td>1 ( \rightarrow ) CAPITAL</td>
<td>1 ( \rightarrow ) N CAPITAL</td>
</tr>
<tr>
<td>PERSON</td>
<td>M ( \rightarrow ) CAR</td>
<td>N ( \rightarrow ) CAR</td>
</tr>
<tr>
<td>STUDENT</td>
<td>M ( \rightarrow ) SECTION</td>
<td>M ( \rightarrow ) SECTION</td>
</tr>
</tbody>
</table>
‘Total/partial’ participation

- total, total
  - COUNTRY 1:1 has 1:1 CAPITAL
  - PERSON 1:1 owns ?N CAR
  - STUDENT ?N takes ?M SECTION

- partial, total
  - COUNTRY 1:1 has 1:1 CAPITAL
  - PERSON 1:1 owns 0:N CAR
  - STUDENT ?N takes ?M SECTION

- partial, total
  - COUNTRY 1:1 has 1:1 CAPITAL
  - PERSON 1:1 owns 0:N CAR
  - STUDENT 1:N takes 0:M SECTION
‘Total/partial’ participation

Is it ‘legal’?

partial, total

PERSON \(\overset{1:1}{\text{owns}}\) 0:N \(\text{CAR}\)

Is it ‘legal’?

NO! why not?

PERSON \(\overset{1:1}{\text{owns}}\) 0:N \(\text{CAR}\)

Subtle concept: Weak entities

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

\(\text{c-id}\) \(\overset{\text{has}}{\text{N}}\) \(\text{SECTION}\) \(\overset{1}{\text{COURSE}}\)
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')

Employee

Dependent

name

N

has

I

COURSE

c-id

SECTION

N

I

LOYEE

\[ n \]
More details

• self-relationships - example?

EMPLOYEE

manages

1

N

FB user

Has-friend

??

??

??

??
More details

• 3-way and k-way relationships?

More details

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

```
  Employee   Uses   Tool   Project
```

More details

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

```
  User   Reviews   Keyword   App
```

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)
]
Specialization

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

![Diagram of Specialization]

Observations

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

![Diagram of Observations]

More details

- Overlap constraints

  ![Diagram of Overlap]

- Covering constraints

  ![Diagram of Covering]
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
  - with overlap

More details

- **Covering constraints - examples?**
  - Total coverage
  - Partial coverage
Overview

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

Aggregation

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

Aggregation

- 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
                           \ N
                            \ M
                        takes
                            /  \\
                    SECTION
```

```
TAKES
   / \  N
   / \  M
SECTION
```

Binary vs Ternary Relationships

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

If each policy is owned by just 1 employee:

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:

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

What are the additional constraints in the 2nd diagram?

Better design

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

Binary vs Ternary Rel.

why is it bad?

Binary vs. Ternary Relationships
(Contd.)
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:

---

Binary vs. Ternary Relationships (Contd.)

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

CPU  HD  MAKER
```

```
CPU  ?  HD  MAKER
```

```
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

<table>
<thead>
<tr>
<th>(strong) entity set</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>weak entity set</td>
<td>primary key</td>
</tr>
<tr>
<td>relationship set</td>
<td>partial key</td>
</tr>
<tr>
<td>identifying rel. set for weak entity</td>
<td></td>
</tr>
</tbody>
</table>
Summary - cont’d

- Cardinalities
  - N \leq M
  - 1:h \leq 1:h’
- Partial/total cardinalities with limits
  - (not in textbook - FYI)

Summary - cont’d

- IS-A
- Aggregation