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
Q: how to record that students take courses?
Cardinalities

• 1 to 1 (example?)
• 1 to N
• N to M

Cardinalities

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

Book's notation:

<table>
<thead>
<tr>
<th>PERSON</th>
<th>owns</th>
<th>CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>takes</td>
<td>SECTION</td>
</tr>
</tbody>
</table>

Cardinalities

Book's notation vs 1 to N notation

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

‘Total/partial’ participation

<table>
<thead>
<tr>
<th>total, total</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
</tr>
<tr>
<td>PERSON</td>
</tr>
<tr>
<td>STUDENT</td>
</tr>
</tbody>
</table>
‘Total/partial’ participation

- total, total
  - COUNTRY : 1:1 has : 1:1 CAPITAL

- partial, total
  - PERSON : 1:1 owns : 0:N CAR

- ??
  - STUDENT : ?:N takes : ?:M SECTION

Weak entities

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

- SECTION : N has : 1 COURSE

- C-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

• 3-way and k-way relationships?

More details

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

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?

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)

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**
  - takes
  - **SECTION**
    - \( N \)

OR

- **STUDENT**
  - **TAKES**
    - \( 1 \)
  - **SECTION**
    - \( N \)

**Binary vs Ternary Relationships**

- usually, binary relationships are ‘cleaner’:

**Binary vs. Ternary Relationships**

If each policy is owned by just 1 employee:
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:

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)
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

- N
- M
- 1:h
- 1':h'
- cardinalities
- with limits
- (not in textbook - FYI)
Summary - cont’d

IS-A

aggregation