Introduction

- ERD is a formal modeling of data
- Conceptual database design
- The data is often consists of:
  - Entities
  - Relationships
- Examples:
  - Movies, actors, directors, roles, awards
  - Students, courses, lecturers, rooms
  - Persons, statuses, friendships, messages, likes

ERD

- Part 1 - Extracting Tables from ERD.
- Part 2 – ERD questions.

The relational model

- Logical database design
- We may deduce the relational model from the conceptual model
  - For instance, by extracting tables from an ERD diagram
- Basic definitions:
  - Relation (table)
  - Schema
  - Tuple
  - Attribute
  - Vector of all key attributes
  - Vector of all non-key attributes

A table example

- t₁ = (foo, bar, baz, {x,y})
- t₂ = (quz, bar, foo, {y,z})

- t₁[a₁] = (baz)
- t₂[a₂] = ({y,z})

- t₁[k₁] = (x,y)
- t₂[k₂] = (quz, bar)

Extracting Tables from ERD
Entities

- Translation to a table:

<table>
<thead>
<tr>
<th>k</th>
<th>k̅</th>
<th>a̅</th>
</tr>
</thead>
<tbody>
<tr>
<td>k̅</td>
<td>a̅</td>
<td>k̅</td>
</tr>
</tbody>
</table>

- Constraints:
  \[ t_1[k] = t_2[k] \Rightarrow t_1[a] = t_2[a] \]

Entities

- A short representation:

<table>
<thead>
<tr>
<th>E</th>
<th>k</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>k̅</td>
<td>a̅</td>
<td></td>
</tr>
</tbody>
</table>

Translation to a table:

- We assume \( k \) can not be empty
- \( a \) may be empty

Any \( a_i \) may be multi-valued

- Multi-valued attributes cannot be a part of the key
- In domain \( D \), for a table row \( t \):
  - for an attribute \( a_j \); \( t[a] \in D \)
  - for a multi-valued attribute \( a_j \); \( t[a] \in P(D) \)
  - \( a \) a powerset.

Relationships

- For each \( E \), \( a \) may be empty
- Translation to a table:

<table>
<thead>
<tr>
<th>E</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>a̅</td>
<td>b̅</td>
<td>E̅</td>
</tr>
</tbody>
</table>

Another Representation of a Multi-Valued Attribute

- A table without multi-valued attributes:

<table>
<thead>
<tr>
<th>movie</th>
<th>year</th>
<th>genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>year</td>
<td>genre</td>
</tr>
<tr>
<td>Cold Mountain</td>
<td>2003</td>
<td>Drama</td>
</tr>
</tbody>
</table>

- Tables - one for each multi-valued attribute:

<table>
<thead>
<tr>
<th>movie</th>
<th>year</th>
<th>screenshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>year</td>
<td>screenshots</td>
</tr>
<tr>
<td>Cold Mountain</td>
<td>2003</td>
<td>screenshots</td>
</tr>
</tbody>
</table>
Relationships – unique reference
- An $E_1$ relates to precisely one $E_2$
- Add a foreign key to the table correlated with $E_2$

Translation to a table:

\[ b \quad a \quad a_1 \]

Relationships – many to one
- An $E_1$ can relate to at most one $E_2$
- Translation to a table:

\[ b \quad a \quad a_1 \]

No longer a key

Relationships – degree constraint
- Note that it always holds that:
  - $\pi_{k_1}(R) \subseteq \pi_{k_2}(E_2)$
  - $\pi_{k_2}(R) \subseteq \pi_{k_1}(E_1)$

Relationships – degree constraint
- Each entity of type $E_1$ must relate (with $R$) to at least one entity of type $E_2$
  - $\pi_{k_2}(E_2) \subseteq \pi_{k_1}(R)$

Aggregations

<table>
<thead>
<tr>
<th>$E_1$</th>
<th>$b_1$</th>
<th>$a_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_2$</td>
<td>$b_2$</td>
<td>$a_2$</td>
</tr>
<tr>
<td>$E_3$</td>
<td>$b_3$</td>
<td>$a_3$</td>
</tr>
<tr>
<td>$R$</td>
<td>$b_4$</td>
<td>$a_4$</td>
</tr>
<tr>
<td>$S$</td>
<td>$b_6$</td>
<td>$b_7$</td>
</tr>
</tbody>
</table>

Aggregations
- Turns the relationship into an entity with attributes of the relationship
Weak Entities

Translation to tables:

<table>
<thead>
<tr>
<th>E₁</th>
<th>E₂</th>
<th>E₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>b₁</td>
<td>b₂</td>
</tr>
<tr>
<td>a₂</td>
<td>b₃</td>
<td>b₄</td>
</tr>
<tr>
<td>a₃</td>
<td>b₅</td>
<td>b₆</td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>name</th>
<th>Birth_date</th>
<th>network.name</th>
<th>user.name</th>
<th>country</th>
<th>tracking</th>
<th>interested</th>
<th>cause</th>
<th>tracking.name</th>
<th>network.name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISA

ISA – a branching weak entity without key components in the subclass

ISA – Translations and Constraints

Dessert:

<table>
<thead>
<tr>
<th>name</th>
<th>calories</th>
</tr>
</thead>
</table>

Ice cream:

<table>
<thead>
<tr>
<th>name</th>
<th>flavor</th>
</tr>
</thead>
</table>

Cake:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
</tr>
</thead>
</table>

Constraints:

\[ \pi_{name}(\text{Cake}) \subseteq \pi_{name}(\text{Dessert}) \]
\[ \pi_{flavor}(\text{Ice cream}) \subseteq \pi_{flavor}(\text{Dessert}) \]
Question 1a
Modify the ERD diagram such that for each university its place will be stored (without adding new entities).

Question 1b
Can parents raise 2 children that were born at the same date?

Question 1c
Can parents raise 3 children that were born at the same date?

Question 1d
How many parents can raise one child?

Question 1e
How can we enforce that each child will be raised by at most one pair of parents?
How can we enforce that each child will be raised by at most one pair of parents?

**Question 1e – 2nd try**

![Diagram](image1)

**Question 1e – 1st try**

![Diagram](image2)

**Question 1e – 3rd try**

![Diagram](image3)