ERD
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
The relational model (a short reminder)

- Logical database design
- We may deduce the relational model from the conceptual model
  - For instance, by extracting tables from an ERD diagram
- Basic definitions:

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Relation (table)</th>
<th>Schema</th>
<th>Tuple</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

236363 - Database Systems - Spring 2016
Part 1 - Extracting Tables from ERD.
Part 2 – ERD questions.
Extracting Tables from ERD
A table example

- \( t_1 = (\text{foo, bar, baz, }\{ \text{x, y} \}) \)
- \( t_2 = (\text{quz, bar, foo, }\{ \text{y, z} \}) \)

- \( t_1[a_1] = (\text{baz}) \)
- \( t_2[a_2] = (\{ \text{y, z} \}) \)

- \( t_1[a] = (\text{baz, }\{ \text{x, y} \}) \)
- \( t_2[\overline{a}] = (\text{quz, bar}) \)
Entities

$k_1, \ldots, k_n, a_1, \ldots, a_m$

$\vec{k} = (k_1, \ldots, k_n)$

$\vec{a} = (a_1, \ldots, a_m)$

- A short representation:
Entities

• Translation to a table:

<table>
<thead>
<tr>
<th>$\bar{k}$</th>
<th>$\bar{a}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>...</td>
</tr>
<tr>
<td>$k_n$</td>
<td>$a_1$</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>$a_m$</td>
</tr>
</tbody>
</table>

• Constraints:

$$t_1[k] = t_2[k] \Rightarrow t_1[a] = t_2[a]$$
Entities

- We assume $\overline{k}$ cannot be empty
- $\overline{a}$ 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 $a_i$: $t[a_i] \in D$
  • for a multi-valued attribute $a_j$: $t[a_j] \in P(D)$
    • a powerset.
Another Representation of a Multi-Valued Attribute

- A table without multi-valued attributes:

<table>
<thead>
<tr>
<th>$k^-$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>year</td>
</tr>
<tr>
<td>Cold Mountain</td>
<td>2003</td>
</tr>
<tr>
<td></td>
<td>genre</td>
</tr>
<tr>
<td></td>
<td>Drama</td>
</tr>
</tbody>
</table>

- Tables - one for each multi-valued attribute:

<table>
<thead>
<tr>
<th>name</th>
<th>year</th>
<th>screenshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold Mountain</td>
<td>2003</td>
<td>screenshot1</td>
</tr>
</tbody>
</table>
For each $\vec{k}$, $\vec{a}$ may be empty
Translation to a table:
Relationships – many to one

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

<table>
<thead>
<tr>
<th>$\bar{k}_1$</th>
<th>$\bar{a}_R$</th>
<th>$\bar{k}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></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 $E_2$

<table>
<thead>
<tr>
<th>$E_1$:</th>
<th>$k_1$</th>
<th>$a_1$</th>
<th>$k_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$E_2$:</th>
<th>$k_2$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relationships – degree constraint

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

<table>
<thead>
<tr>
<th>$E_2$</th>
<th>$\overline{k}_2$</th>
<th>$\overline{a}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>$\overline{k}_1$</td>
<td>$\overline{a}_R$</td>
</tr>
</tbody>
</table>
Note that it always holds that:

- \( \pi_{k_2}(R) \subseteq \pi_{k_2}(E_2) \)
- \( \pi_{k_1}(R) \subseteq \pi_{k_1}(E_1) \)
Aggregations

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

E1: \[
\begin{array}{|c|c|}
\hline
k_1 & a_1 \\
\hline
\end{array}
\]

E2: \[
\begin{array}{|c|c|}
\hline
k_2 & a_2 \\
\hline
\end{array}
\]

E3: \[
\begin{array}{|c|c|}
\hline
k_3 & a_3 \\
\hline
\end{array}
\]

R: \[
\begin{array}{|c|c|c|}
\hline
k_1 & k_2 & a_R \\
\hline
\end{array}
\]

S: \[
\begin{array}{|c|c|c|c|}
\hline
k_1 & k_2 & k_3 & a_S \\
\hline
\end{array}
\]
Weak Entities

Translation to tables:

<table>
<thead>
<tr>
<th>$\overline{k}_1$</th>
<th>$\overline{a}_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\overline{k}_1$</th>
<th>$\overline{k}_2$</th>
<th>$\overline{a}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\overline{k}_1$</th>
<th>$\overline{k}_2$</th>
<th>$\overline{k}_3$</th>
<th>$\overline{a}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Weak Entities

Constraints:

\[ \pi_{k_1}(E_2) \subseteq \pi_{k_1}(E_1) \]

\[ \pi_{k_1,k_2}(E_3) \subseteq \pi_{k_1,k_2}(E_2) \]

Translation to tables:

<table>
<thead>
<tr>
<th></th>
<th>( k_1 )</th>
<th>( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( a_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_2 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ISA

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

\[ \begin{align*}
E_1 & \quad | \quad k_1 \mid a_1 \\
E_2 & \quad | \quad a_2 \\
E_3 & \quad | \quad a_3
\end{align*} \]
ISA – Translations and Constraints

Dessert:

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

Ice cream:

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

Cake:

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

Constraints:

\[ \pi_{\text{name}}(\text{Cake}) \subseteq \pi_{\text{name}}(\text{Dessert}) \]

\[ \pi_{\text{name}}(\text{Ice cream}) \subseteq \pi_{\text{name}}(\text{Dessert}) \]
ERD Questions
Question 1 - ERD
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?
Question 1e – 1st try

How can we enforce that each child will be raised by at most one pair of parents?
Question 1e \(-2^{\text{nd}}\) try

How can we enforce that each child will be raised by at most one pair of parents?
Question 1e – 3rd try
(non-modern family)
How can we enforce that each child will be raised by at most one pair of parents?

![Diagram showing the relationship between Person, Couple, and Child.]