Database Development Process
### Common Data Models

<table>
<thead>
<tr>
<th>UML/OO</th>
<th>ER</th>
<th>Relational</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>entity type</td>
<td>relation/table</td>
</tr>
<tr>
<td>object</td>
<td>entity</td>
<td>tuple/row</td>
</tr>
<tr>
<td>attribute</td>
<td>attribute</td>
<td>attribute/column</td>
</tr>
<tr>
<td>association</td>
<td>relationship</td>
<td>foreign key</td>
</tr>
<tr>
<td></td>
<td>key attribute</td>
<td>primary key</td>
</tr>
<tr>
<td>inheritance</td>
<td>inheritance</td>
<td>foreign key</td>
</tr>
</tbody>
</table>

- We have standard techniques for translating between data models.
Common Data Models

- UML and ER have graphical notations
  - very good for conceptual design
  - very expressive: capture more application semantics

- Relational model is an implementation model
  - not good for conceptual design
  - not very expressive: many concepts map to same notation
Entity Types

- Entity types → boxes
- Weak entity type → double box

EMPLOYEE

Entity

DEPENDENT

Weak Entity
Entity Types

- Entity types are similar to classes, they describe potential objects (entities) that will appear in the database.
- Weak entity types describe dependent entities, entities that depend on other entities for identity.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Weak Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>DEPENDENT</td>
</tr>
</tbody>
</table>

Entity

Weak Entity
Attributes and Keys

- Attributes ➔ ovals
- Key attributes ➔ underlined name
- Partial key attributes ➔ dotted underlined name

Age
Attribute

SSN
Key Attribute

Date
Partial Key Attribute
Attributes and Keys

- Key attributes must be unique for each entity
- Keys are used to identify particular entities
- Partial keys are only partially unique
  - used for weak entity types

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Key Attribute</th>
<th>Partial Key Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>SSN</td>
<td>Date</td>
</tr>
</tbody>
</table>
Attributes are connected to entity types by lines.
Entity Types and Entities

- Entities are instances of an entity type.

<table>
<thead>
<tr>
<th>Entity Type Name:</th>
<th>EMPLOYEE</th>
<th>COMPANY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name, Age, Salary</td>
<td>Name, Headquarters, President</td>
</tr>
<tr>
<td>Entity Set:</td>
<td>(John Smith, 55, 80k)</td>
<td>(Sunco Oil, Houston, John Smith)</td>
</tr>
<tr>
<td>(Extension)</td>
<td>(Fred Brown, 40, 30K)</td>
<td>(Fast Computer, Dallas, Bob King)</td>
</tr>
<tr>
<td>e₁</td>
<td>.</td>
<td>c₁</td>
</tr>
<tr>
<td>e₂</td>
<td>.</td>
<td>c₂</td>
</tr>
<tr>
<td>e₃</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>(Judy Clark, 25, 20K)</td>
<td>.</td>
<td></td>
</tr>
</tbody>
</table>
Entity Types and Keys

- All regular entity types must have a key attribute or set of key attributes.
- Weak entity types must have partial keys.
- Weak entities get part of their key (and part of their identity) from some related entity.
Sets and Derived Attributes

- Multivalued attributes $\rightarrow$ double lined oval
  - multivalued = set valued
  - that there may be more than one value for the attribute.
- Derived attributes $\rightarrow$ dashed line ovals
  - the attribute is computed from other data
Composite Attributes

- Composite attributes $\rightarrow$ tree
  - composed of other attributes.
  - used for a set of related attributes, when the set is not a conceptual entity
  - the composite doesn’t have identity … it doesn’t have a key
CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

\[
\begin{align*}
\text{CAR}_1 & \quad \text{((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 \{red, black\})} \\
\text{CAR}_2 & \quad \text{((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, \{blue\})} \\
\text{CAR}_3 & \quad \text{((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, \{white, blue\})} \\
\vdots & \\
\vdots & 
\end{align*}
\]
Relationships

- Relationships → diamonds
- Identifying relationship → double diamond
Relationships

- Relationships indicate a meaningful connection between two entity types.
- Relationships may have attributes, but they cannot have key attributes.
- Identifying relationships connect a weak entity type to some other entity type.
  - Indicates where the weak entity gets a key to complete its own partial key.

- WorksOn
- DependentOf
- Relationship
- Identifying Relationship
Participation and Cardinality

- Participation and cardinality define constraints on relationships.
- Participation indicates whether an entity is required to take part in a relationship.
- Cardinality ratios and structural constraints place limits on the number of entities that may participate in a relationship.
Participation Constraints

- Total participation → double or thick line
  - indicates required participation
- Partial participation → thin line
  - indicates optional participation

![Diagram showing participation constraints between Employee, Department, and Project](image-url)
Arrowheads can be used to indicate an upper bound of 1 for participation.

- (not used in our textbook)

- **X** must participate in exactly one **R**

- **X** may participate in at most one **R**
Cardinality ratios specify the maximum number of relationship instances that an entity may participate in:

- **1:1 ratio**
  - EMPLOYEE Manages DEPARTMENT
  - EMPLOYEE 1
  - DEPARTMENT 1

- **n:1 ratio**
  - EMPLOYEE WorksFor DEPARTMENT
  - EMPLOYEE n
  - DEPARTMENT 1

- **n:m ratio**
  - EMPLOYEE WorksOn PROJECT
  - EMPLOYEE n
  - PROJECT m
Structural Constraints

- Structural constraints specify the *minimum* and *maximum* number of relationship instances that an entity may participate in.

**WorksFor**

- **EMPLOYEE** (1,1) \(\rightarrow\) **DEPARTMENT** (4,n)

  An employee must work for exactly 1 department. A department must have at least 4 employees.

**Manages**

- **EMPLOYEE** (0,1) \(\rightarrow\) **DEPARTMENT** (1,1)

  An employee may manage at most 1 department. A department must have exactly 1 manager.
Participation and Cardinality

- There's generally numerous ways to express a relationship constraint.

- (0,N) optional participation in any number of relationships
- (0,1) optional participation in at most one relationship
- (1,N) required participation in at least one relationship
- (1,1) required participation in exactly one relationship
An employee can manage at most one department.
A department must have exactly one manager.
Equivalent Notations

- An employee must work for exactly one department.
- A department must have at least one employee.
Each EMPLOYEE instance is connected to exactly one WORKS_FOR instance.

Each DEPARTMENT instance is connected to at least one WORKS_FOR instance.
It is sometimes convenient to name an entity’s role in a relationship.
- particularly useful in recursive relationships
- removes ambiguity in direction of relationship
Recursive Relationship

1 = supervisor
2 = supervisee
Notation Summary

- **Entity**
- **Weak Entity**
- **Relationship**
- **Identifying Relationship**

- **Attribute**
- **Key Attribute**
- **Multivalued Attribute**
- **Composite Attribute**
- **Derived Attribute**

- Total Participation of \( E_2 \) in \( R \)
- Cardinality Ratio 1: \( N \) for \( E_1; E_2 \) in \( R \)
- Structural Constraint \((\text{min}, \text{max})\) on Participation of \( E \) in \( R \)
Notes:
A LEG (segment) is a nonstop portion of a flight. A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.
Consider a MAIL_ORDER database in which employees take orders for parts from customers. The data requirements are summarized as follows:

- The mail order company has employees, each identified by a unique employee number, first and last name, and ZIP code.
- Each customer of the company is identified by a unique customer number, first and last name, and ZIP code.
- Each part sold by the company is identified by a unique part number, a part name, price, and quantity in stock.
- Each order placed by a customer is taken by an employee and is given a unique order number. Each order contains specified quantities of one or more parts. Each order has a date of receipt as well as an expected ship date. The actual ship date is also recorded.