Understanding Entity Relationship Diagrams

Dr. Chawalit Jeenanunta
Objective

- Notation basics of Crow’s foot diagram
- Understanding relationships
- Generalization hierarchies
Basic Symbols

- **Entity type:**
  - Collection of things of interest: persons, places, things, events
  - Entity: instance or member of an entity type
Basic Symbols

- **Attribute:**
  - Properties of entity types or relationships
  - Data type to indicate the kind of values and permissible operations on the attribute
  - Shown inside entity type or next to relationship
  - Like columns
  - Primary key
Basic Symbols

- **Relationship:**
  - Named association among entities: name is significant (gives it more status)
  - Usually between two entity types: can involve multiple entity types and one
  - Bidirectional:
    - Can be used to navigate in both directions
    - Two names
    - Course to Offering: Has, Provides
    - Offering to Course: IsProvidedFor
    - Which name to use: try to use active verb; not always possible

![Diagram of entity types and relationships]

- **Entity Type**
  - **Course**
    - Attributes: CourseNo, CrsDesc, CrsUnits
    - Primary Key: CourseNo

- **Offering**
  - Attributes: OfferNo, OffLocation, OffTime
  - Relationship name: Has

Dr. Chawalit Jeenanunta
DBMS: Slide 5
Cardinalities

- A constraint on the number of entities that participate in a relationship
- Specify the minimum and maximum cardinalities in both directions
Cardinalities

Instance diagram:
- Shows occurrences of entity types (entities)
- Useful to understand some relationships (similar to sample table usage)
- Lines show relationships among entities
- Course1 related to Offering1, Offering2, and Offering3
- Course2 related to Offering4
- Course3 not related to any offerings
- Course related to a minimum of 0 and maximum of many
- Offering: each related to exactly one course
**Cardinality Notation**

- **Symbols:**
  - Oval: means 0
  - Perpendicular line: means 1
  - Crow's foot: means many (0 or more); unconstrained
  - Some drawing tools support exact cardinalities (numbers)

- **Placement:**
  - Inside symbol: minimum cardinality
  - Outside symbol: maximum cardinality
  - Interpret the far cardinality symbols: near the other entity type
Cardinality Notation

- Course is related to a min of 0 and max of many offerings
- Offering is related to a min of 1 and max of 1 courses (exactly one)
Classification of Cardinalities

- **Minimum cardinality based:**
  - Min cardinality of one: mandatory; makes entity types existent dependent
  - Min cardinality of 0: optional; similar to a FK that allows null values

- **Maximum cardinality based:**
  - Functional: max cardinality of 1; mathematics based
  - 1-M: max cardinalities are 1 and M
  - M-N: max cardinalities are many in both directions
  - 1-1: max cardinality is one in both directions (not common)
### Summary of Cardinalities

<table>
<thead>
<tr>
<th>Classification</th>
<th>Cardinality Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Minimum cardinality $\geq 1$</td>
</tr>
<tr>
<td>Optional</td>
<td>Minimum cardinality $= 0$</td>
</tr>
<tr>
<td>Functional or single-valued</td>
<td>Maximum cardinality $= 1$</td>
</tr>
<tr>
<td>1-M</td>
<td>Maximum cardinality $= 1$ in one direction and maximum cardinality $&gt; 1$ in the other direction.</td>
</tr>
<tr>
<td>M-N</td>
<td>Maximum cardinality is $&gt; 1$ in both directions.</td>
</tr>
<tr>
<td>1-1</td>
<td>Maximum cardinality $= 1$ in both directions.</td>
</tr>
</tbody>
</table>
More Relationship Examples

- **TeamTeaches**:  
  - M-N  
  - Optional in both directions
- **WorksIn**:  
  - 1-1  
  - Optional: office can be empty  
  - Mandatory: faculty must be assigned to an office
Understanding Relationships

- Identification dependency
- M-N relationships with attributes
- Self identifying relationships
- M-way relationships
- Equivalence between M-N and 1-M relationships
Identification Dependency

Concept:
- Some entity types borrow part or entire PK
- Specialized concept: important when it occurs but not too common
- Closely related entities: physical containment
Identification Dependency

- Room is physically contained in a building
- Identification of room includes building
- Others: country-state, order-orderline

Identification Dependency Symbols:
- Solid relationship line for identifying relationships
- Diagonal lines in the corners denote weak entities.
Identification Dependency

- **Symbols:**
  - Weak entity:
    - Borrows part or all of PK
    - Diagonal lines in the corners
  - Identifying relationship:
    - Solid line
    - Indicates the source of PK
    - Ambiguity if entity type participates in more than one relationship
Identification Dependency

Example:
- PK of Room is a combination of RoomNo (local key) and BldgID (borrowed attribute)
- Cardinality of weak entity in the identifying relationship must be 1-1
- Room cannot exist unless associated building exists
- Identification dependency involves existence dependency:
  - Weak entity is existent dependent on other entity
- Also borrows part of all of PK
M-N Relationships with Attributes

- Relationships are first class citizens
  - Can have attributes just like entity types
  - Most typical for M-N relationships
  - Attribute depends on both entity types, not just one entity type
  - 1-M relationships with attributes is controversial
Example:

- EnrGrade: grade recorded for a student in a particular course
- Depends on the combination of Student and Offering
- EnrGrade is not part of the Student or Offering entity types
M-N Relationships with Attributes

- AuthOrder: the order in which the author’s name appears in the title of a book
  - Record order of authors: important in publishing disciplines
  - AuthOrder is part of the Writes relationship (combination of Author and Book)
  - AuthOrder is not part of the Author or Book entity types

b) Writes relationship

```
Author
| AuthNo  |
| AuthName  |

Book
| ISBN |
| Title |

Writes

AuthOrder
```
M-N Relationships with Attributes

- Qty: quantity of part supplied by a supplier; quantity varies by part and supplier

![Diagram showing M-N relationship with Attributes]

**Diagram:**
- **Supplier**:
  - SuppNo
  - SuppName

- **Part**:
  - PartNo
  - PartName

- **Relationship with Qty**

Dr. Chawalit Jeenanunta  DBMS: Slide 21
Instance Diagrams for Self-Referencing Relationships

- **Basic idea:**
  - Associations among members of the same set
  - Specialized concept: important when occurs but not too common

- **Courses:** prerequisite structures

![Instance Diagram](image)
Instance Diagrams for Self-Referencing Relationships

- **Employees**: supervisory relationships

![Instance Diagram]

- (a) **Supervises**
  - Faculty1
  - Faculty2
  - Faculty3
  - Faculty4
  - Faculty5
ERD Notation for self-Referencing Relationships

- Relationship connects entity type to itself
- Position of cardinalities is not important: relationship involves the same entity type
- Otherwise, nothing is different about self-referencing relationships
- Key point:
  - 1-M vs. M-N
  - Use instance diagrams to help you reason
Associative Entity Types for M-way Relationships

- Relationships can involve more than 2 entity types
- Specialized concept: important when occurs but not common
- Difficult concept: easy to use inappropriately
- Track interaction of 3 entity types: 3 way relationship
Associative Entity Types for M-way Relationships

- 3 way relationship tracks who supplies a part on a specified project
- Do not use when only who supplies a part and what parts are used on which projects
- Local purchasing: suppliers chosen for each project rather than centrally
Associative Entity Types for M-way Relationships

- Crow's Foot does not support M-way relationships
- Some ERD notations do support
- Use an associative entity type and identifying relationships
- Associative entity type:
  - Weak entity that depends on two or more entity types
  - Replacement for M-N or M-way relationship
  - Usually borrows entire PK
Relationship Equivalence

- Replace M-N relationship
  - Associative entity type
  - Two identifying 1-M relationships

- M-N relationship versus associative entity type
  - Largely preference
  - Associative entity type is more flexible in some situations
Associative Entity Type Example

- **Enrollment:**
  - Associative entity type
  - Represents M-N relationship between Student and Offering

- **Attendance:**
  - Weak entity
  - PK: Combination of AttDate and PK of Enrollment

- Must use associative entity type for Enrollment rather than M-N relationship:
  - Relate Enrollment to other entity types (Attendance)
  - Cannot connect a relationship to other relationships
  - Use associative entity type when other relationships involved
Generalization Hierarchies

- **Classification:**
  - Group objects by similarity
  - Pervasive in life and business:
    - Understand animal species by similarity of characteristics
  - Conduct business by classifying entities: investments, loans, customers, …

- **Generalization hierarchy:**
  - Shows similarity of entity types
  - Differences in attributes: use when entity types have similar but different attributes
  - Specialized technique: do not overuse; important when occurs but not common
Generalization Hierarchies

- **Vocabulary:**
  - **Supertype:** parent entity type
  - **Subtype:** child entity type (additional attributes)
  - Generalization hierarchy: ISA (not an acronym)
  - SalaryEmp IS AN Employee
  - HourlyEmp IS AN Employee

- **Set inclusion:**
  - Subtypes are subsets of supertypes
  - Set of SalaryEmp entities is a subset of Employee entities
Inheritance

- Subtypes inherit attributes of supertypes (direct and indirect)
- Allows abbreviation of attribute list
- Applies to code (methods) as well as attributes (data)
- Inheritance related to sharing of characteristics (data and code)

- First developed for object-oriented programming languages:
  - Reduce the amount of code by inheriting code from similar objects
  - Later applied to databases

- Notation:
  - Do not show inherited attributes
  - Assume existence: SalaryEmp includes all attributes of Employee
  - Sometimes exception for PK: for emphasis can show it
Generalization Constraints

- **Cardinality of generalization hierarchy:**
  - No need to specify
  - **Subtype:** min and max cardinalities are both 1
  - **Supertype:** min cardinality is zero or one; max cardinality is one

- **Disjointness:**
  - Subtypes have common entities:
    - D means intersection is empty
    - No symbol means intersection is not empty: Faculty and Student

- **Completeness:**
  - Supertype have free standing entities (not in any subtype)
  - C: Complete; union of subtype entities is the set of supertype entities
  - **Nothing:** supertype can have free standing entities; Union of subtypes has few entities than supertype
Generalization Constraints

- Security
  - Symbol
  - SecName
  - LastClose

- Stock
  - OutShares
  - IssuedShares

- Bond
  - Rate
  - FaceValue

Disjointness Constraint

Completeness Constraint
Multiple Levels of Generalization

- Each subtype inherits from direct and indirect supertypes
  - Common inherits from Stock and Security
  - Provide constraints for each level of generalization hierarchy
- Specialized technique:
  - Need for multiple levels is less common than a single level
  - Do not overuse generalization hierarchies
  - Use only to show similarity among attributes
Comprehensive Example

- Demonstrates most of the notation previously shown
  - 1-M relationships: Has, Teaches, Supervises, Registers, Grants
  - Optional relationships: Teaches, Supervises
  - Mandatory relationship: Has, Registers, Grants
  - Self referencing relationship: Supervises
  - Weak entity (also associative entity type): Enrollment
  - Identifying relationships: Registers, Grants
  - Generalization hierarchy: UnivPerson, Student, Faculty
  - Could apply relationship equivalence to transform Enrollment, Registers, and Grants into a M-N relationship with an attribute