Three Schema Model

- ANSI/SPARC introduced the three schema model in 1975
- It provides a framework describing the role and purpose of data modeling

Three Schema Model (cont.)

- External schema or user view
  - Representation of how users view the database
- Conceptual schema
  - A logical view of the database containing a description of all the data and relationships
  - Independent of any particular means of storing the data
  - One conceptual schema usually contains many different external schemas
- Internal schema
  - A representation of a conceptual schema as physically stored on a particular product
  - A conceptual schema can be represented by many different internal schemas

E-R Model

- Entity-Relationship model is a set of concepts and graphical symbols that can be used to create conceptual schemas
- Four versions
  - Original E-R model by Peter Chen (1976)
  - Extended E-R model: the most widely used model
  - Information Engineering (IE) by James Martin (1990)
  - IDEF1X national standard by the National Institute of Standards and Technology
  - Unified Modeling Language (UML) supporting object-oriented methodology

The Extended E-R Model

Figure 2.15 Example Entity-Relationship Diagram

Example: E-R Diagram

Figure 2.10 Showing Attributes in an Entity-Relationship Diagram
Entities

- Something that can be identified and the users want to track
  - Entity class is a collection of entities described by the entity format in that class
  - Entity instance is the representation of a particular entity
- There are usually many instances of an entity in an entity class

Example: Entity

```
<table>
<thead>
<tr>
<th>CUSTOMER</th>
<th>entity instance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customer Name</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td>City</td>
</tr>
<tr>
<td></td>
<td>Zip</td>
</tr>
<tr>
<td></td>
<td>Contact Name</td>
</tr>
<tr>
<td></td>
<td>Contact Phone</td>
</tr>
</tbody>
</table>
```

Attributes

- Description of the entity's characteristics
- All instances of a given entity class have the same attributes
  - Composite attribute: attribute consisting of the group of attributes
  - Multi-value attributes: attribute with more than one possible value

Identifiers

- Identifiers are attributes that name, or identify, entity instances
- The identifier of an entity instance consists of one or more of the entity's attributes
- An identifier may be either unique or non-unique
  - Unique identifier: the value identifies one and only one entity instance
  - Non-unique identifier: the value identifies a set of instances
- Composite identifiers: Identifiers that consist of two or more attributes

Relationships

- Entities can be associated with one another in relationships
  - Relationship classes: associations among entity classes
  - Relationship instances: associations among entity instances
- Relationships can have attributes
- A relationship class can involve many entity classes
- Degree of the relationship is the number of entity classes in the relationship

Example: Degree of the relationship

- Relationships of degree 2 are very common and are often referred to by the term binary relationships
Binary Relationships

- 1:1
- 1:N
- N:M

Recursive Relationship

- Recursive relationships are relationships among entities of a single class

Cardinality

- Maximum cardinality indicates the maximum number of entities that can be involved in a relationship
- Minimum cardinality indicates that there may or may not be an entity in a relationship

Weak Entities

- Weak entities are those that must logically depend on another entity
- Weak entities cannot exist in the database unless another type of entity (strong entity) also exists in the database
- ID-dependent entity: the identifier of one entity includes the identifier of another entity

Example: Weak Entities
Subtype Entities

- Subtype entity is an entity that represents a special case of another entity, called supertype
- Sometimes called an IS-A relationship
- Entities with an IS-A relationship should have the same identifier

Example: Subtype Entities

![Subtype Entities](image)

IDEF1X Standard

- IDEF1X (Integrated Definition 1, Extended) was announced as a national standard in 1993
- It defines entities, relationships, and attributes in more specific meanings
- It changed some of the E-R graphical symbols
- It includes definition of domains, a component not present in the extended E-R model
- Four Relationship Types
  - Non-Identifying Connection Relationships
  - Identifying Connection Relationships
  - Non-Specific Relationships
  - Categorization Relationships
- Products supporting IDEF1X: ERWin, Visio, Design/2000

Example: IDEF1X

![IDEF1X](image)
Non-Identifying Connection Relationships

- Represent relationship with a dashed line from a parent to a child entity.
- Default cardinality is 1:N with a mandatory parent and an optional child.
  - 1 indicates exactly one child is required.
  - Z indicates zero or one children.

Identifying Connection Relationships

- Same as ID-dependent relationships in the extended E-R model.
- Parent’s identifier is always part of the child’s identifier.
- Relationship are indicated with solid lines, child entities are shown with rounded corners (ID-dependent entities only).
Non-Specific Relationships

- Simply a many-to-many relationship
- Relationships are shown with a filled-in circle on each end of the solid relationship line
- Cannot set minimum cardinalities of a non-specific relationship

Categorization Relationships

- A relationship between a generic entity and another entity called a category entity
- Called specialization of generalization/subtype relationships (IS-A relationships) in the extended E-R model
- Within category clusters, category entities are mutually exclusive
- Two types of category clusters:
  - Complete: every possible type of category for the cluster is shown (denoted by two horizontal lines with a gap in-between)
  - Incomplete: at least one category is missing (denoted by placing the category cluster circle on top of a single line, no gap between horizontal lines)

Example: Categorization Relationships

Example: IDEF1X Model With Relationship Names

Example: IDEF1X Model With Relationship Names
Domains

- A domain is a named set of values that an attribute can have.
- It can be a specific list of values or a pre-defined data characteristic, e.g., character string of length less than 75.
- Domains reduce ambiguity in data modeling and are practically useful.
- Two types of domains:
  - Base domain: have a data type and possibly a value list or range definition.
  - Type domain: a subset of a base domain or a subset of another type domain.

Example: Domain Hierarchy

UML-style E-R Diagrams

- The Unified Modeling Language (UML) is a set of structures and techniques for modeling and designing object-oriented programs (OOP) and applications.
- The concept of UML entities, relationships, and attributes are very similar to those of the extended E-R model.
- Several OOP constructs are added:
  - `<Persistent>` indicates that the entity class exist in the database.
  - UML allows entity class attributes.
  - UML supports visibility of attributes and methods.
  - UML entities specify constraints and methods in the third segment of the entity classes.
- Currently, the object-oriented notation is of limited practical value.

Example: UML

- Figure 2.27a UML Representation of a 1:1 Relationship
- Figure 2.27b UML Representation of a 1:N Relationship
- Figure 2.27c UML Representation of an N:M Relationship
Chapter 2
Entity-Relationship Data Modeling:
Tools and Techniques