Chapter 12

Entity-Relationship Modeling
Chapter 12 - Objectives

- How to use Entity–Relationship (ER) modeling in database design.
- Basic concepts associated with ER model.
- Diagrammatic technique for displaying ER model using Unified Modeling Language (UML).
- How to identify and resolve problems with ER models called connection traps.
- How to build an ER model from a requirements specification.
Concepts of the ER Model

- Entity types
- Relationship types
- Attributes
Entity Type

- **Entity type**
  - Group of objects with same properties, identified by enterprise as having an independent existence.

- **Entity occurrence**
  - Uniquely identifiable object of an entity type.
## Examples of Entity Types

<table>
<thead>
<tr>
<th>Physical existence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staff</td>
</tr>
<tr>
<td>Property</td>
</tr>
<tr>
<td>Customer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conceptual existence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing</td>
</tr>
<tr>
<td>Inspection</td>
</tr>
</tbody>
</table>
ER diagram of Staff and Branch entity types
Relationship Types

- Relationship type
  - Set of meaningful associations among entity types.

- Relationship occurrence
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.
ER diagram of Branch *Has* Staff relationship

```
Staff ← Has → Branch

‘Branch has staff’
```
Conventions for Naming and Directing Relationships

- **Naming Conventions**
  - Try to use action verbs
  - Try not to repeat names (e.g., “has”, “manages”)

- **Direction of Relationship Arrow**: If relationship is 1:n, then direct it from the 1-relation to the n-relation
  - Alternative explanation: Arrow goes toward the relation that has the foreign key
Relationship Types

- **Degree of a Relationship**
  - Number of participating entities in a relationship.

- **Relationship of degree**:  
  - two is binary  
  - three is ternary  
  - four is quaternary.
Binary relationship called $POwns$

‘Private owner owns property for rent’

PrivateOwner  $POwns$  PropertyForRent
Ternary relationship called *Registers*

Staff

Registers

Branch

‘Staff registers a client at a branch’

Client
Another Ternary Relationship
Wrong way to show a ternary relationship
Quaternary relationship called **Arranges**

- **Solicitor**
- **Buyer**
- **Financial Institution**
- **Bid**

'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'
Relationship Types

- **Recursive Relationship**
  - Relationship type where *same* entity type participates more than once in *different roles*.

- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.
Recursive relationship called *Supervises* with role names

‘Staff (Supervisor) supervises staff (Supervisee)’
Entities associated through two distinct relationships with role names

‘Manager manages branch office’

Role name

Manager

Branch Office

Manages

Staff

Branch

Has

Member of Staff

Branch Office

Role name

‘Branch office has member of staff’
Attributes

♦ Attribute
  – Property of an entity or a relationship type.

♦ Attribute Domain
  – Set of allowable values for one or more attributes.
Attributes

- **Simple Attribute**
  - Attribute composed of a single component with an independent existence.

- **Composite Attribute**
  - Attribute composed of multiple components, each with an independent existence.
Attributes

- **Single-valued Attribute**
  - Attribute that holds a single value for each occurrence of an entity type.

- **Multi-valued Attribute**
  - Attribute that holds multiple values for each occurrence of an entity type.
Attributes

- **Derived Attribute**
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.
Keys

- **Candidate Key**
  - Minimal set of attributes that uniquely identifies each occurrence of an entity type.

- **Primary Key**
  - Candidate key selected to uniquely identify each occurrence of an entity type.

- **Composite Key**
  - A candidate key that consists of two or more attributes.
ER diagram of Staff and Branch entities and their attributes
Relationship called *Advertises* with attributes

```
\[ \text{Newspaper} \quad \text{Advertises} \quad \text{PropertyForRent} \]
\[ \text{newspaperName} \quad \text{propertyNo} \]
\[ \text{dateAdvert} \quad \text{cost} \]
```

*Newspaper advertises property for rent*
Structural Constraints

- Main type of constraint on relationships is called \textit{multiplicity}.

- Multiplicity - number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.

- Represents policies (called \textit{business rules}) established by user or company.
Structural Constraints

- Multiplicity is made up of two types of restrictions on relationships: *cardinality* and *participation*.
Structural Constraints

- **Cardinality**
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.

- **Participation**
  - Determines the minimum number of entities that may participate in the relationship.

- **Example**: 0..1: means that a minimum of 0 and a maximum of 1 entity may occur in the relationship.
Multiplicity as cardinality and participation constraints

- 'One branch is managed by one member of staff'
- 'One member of staff manages one branch'
- 'All branches are managed' (mandatory participation for branch)
- 'Not all staff manage branches' (optional participation for staff)

Cardinality

Staff
staffNo
1..1
Manages
Branch
branchNo
0..1

Participation
Structural Constraints

- The most common degree for relationships is binary.

- Binary relationships are generally referred to as being:
  - one-to-one (1:1)
  - one-to-many (1:*)
  - many-to-many (*:*).

- The book is overloading its meaning of (x:y) here—in this context the (x:y) means the maximum number of occurrences of each of the two entity types
  - Example, if staff is 0..1 and branch is 1..*, then the relationship between the two is (1:*), which involves taking the 1 from 0..1 and combing it with the * from 1..*
Multiplicity of Staff Manages Branch (1:1) relationship

Each branch is managed by one member of staff

A member of staff can manage zero or one branch

Staff
- staffNo

Branch
- branchNo

Multiplicity

1..1

0..1
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type

- Staff: `staffNo` (0..1) → Oversees → PropertyForRent: `propertyNo` (0..*)

  - ‘Each property for rent is overseen by zero or one member of staff’
  - ‘Each member of staff oversees zero or more properties for rent’
Multiplicity of Newspaper Advertises PropertyForRent (*:* *) relationship

- "Each property for rent is advertised in zero or more newspapers"
- "Each newspaper advertises one or more properties for rent"

Diagram:
- Newspaper
  - newspaperName
- Advertises
  - 0..* to 1..*
- PropertyForRent
  - propertyNo
The Wrong Way to Show a *:* Relationship

Newspaper 1..1 Lists 1..* Advertisement 0..*

AppearsIn 1..1 PropertyForRent
Structural Constraints

- **Multiplicity for Complex Relationships**
  - Number (or range) of possible occurrences of an entity type in an $n$-ary relationship when other $(n-1)$ values are fixed.
  - May help to write the relationship as a function.

  » Example: There is a relationship $D$ among 3 entities $A$, $B$, $C$.
    - If $|D(A,B)| = 0..1$, then you put $0..1$ next to $C$
Multiplicity of ternary *Registers* relationship

![Entity relationship diagram](image-url)
# Summary of multiplicity constraints

<table>
<thead>
<tr>
<th>Alternative ways to represent multiplicity constraints</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>Zero or one entity occurrence</td>
</tr>
<tr>
<td>1..1 (or just 1)</td>
<td>Exactly one entity occurrence</td>
</tr>
<tr>
<td>0..* (or just *)</td>
<td>Zero or many entity occurrences</td>
</tr>
<tr>
<td>1..*</td>
<td>One or many entity occurrences</td>
</tr>
<tr>
<td>5..10</td>
<td>Minimum of 5 up to a maximum of 10 entity occurrences</td>
</tr>
<tr>
<td>0, 3, 6–8</td>
<td>Zero or three or six, seven, or eight entity occurrences</td>
</tr>
</tbody>
</table>
Problems with ER Models

- Problems may arise when designing a conceptual data model called *connection traps*.

- Often due to a misinterpretation of the meaning of certain relationships.

- Two main types of connection traps are called *fan traps* and *chasm traps*.
Problems with ER Models

- Fan Trap
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.

- Chasm Trap
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.
An Example of a Fan Trap
Semantic Net of ER Model with Fan Trap

- At which branch office does staff number SG37 work?
Restructuring ER model to remove Fan Trap

Restructuring the model resolves the fan trap
SG37 works at branch B003.
An Example of a Chasm Trap
At which branch office is property PA14 available?
ER Model restructured to remove Chasm Trap

Adding the Offers relationship resolves the chasm trap
Semantic Net of Restructured ER Model with Chasm Trap Removed