Chapter 7

Database Design

and

The E–R Model
Goals

- Facilitate DB design and represent the overall logical structure of the DB.

**Definition** Entities are specific objects in the real world that are represented in the DB.

**Definition** An entity is represented by a set of attributes whose properties are used to describe an entity.

**Definition** The domain of an attribute is a set of permitted values for that attribute.

**Definition** A relationship relates two or more (distinct) entities with a specific meaning.

- the *degree* (binary, ternary, ...) of a relationship is the number of participating entities.
- may have its own attributes, called descriptive attributes
- can relate 2 entities of the same type (in a recursive relationship), and thus each entity plays a different role in the relationship.
Entity-Relationship Model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Entity Set</td>
</tr>
<tr>
<td></td>
<td>Weak Entity Set</td>
</tr>
<tr>
<td></td>
<td>Relationship Set</td>
</tr>
<tr>
<td></td>
<td>Attribute</td>
</tr>
<tr>
<td></td>
<td>Key Attribute</td>
</tr>
</tbody>
</table>
Entity-Relationship Model

Symbol

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>$R$</td>
</tr>
<tr>
<td>$E_1$</td>
<td>$R$</td>
</tr>
<tr>
<td>$E_1$</td>
<td>$R$</td>
</tr>
<tr>
<td>$R$</td>
<td>$E_2$</td>
</tr>
</tbody>
</table>

Mapping Cardinality

- 1:1 for $E_1:E_2$ in $R$
- 1:$n$ for $E_1:E_2$ in $R$
- $n:m$ for $E_1:E_2$ in $R$

Roles ($A$ and $B$)
Attributes

- An entity is represented by a set of attributes, which is descriptive properties possessed by all members of an entity set.

Example.

\[
\text{customer} = (\text{customer-id, customer-name, customer-street, customer-city}) \\
\text{loan} = (\text{loan-number, amount})
\]

- Domain – the set of permitted values for each attribute

- Attribute types:
  - Simple and composite attributes (e.g., name: first, middle, last).
  - Single-valued and multi-valued attributes (e.g., phone-numbers)
  - Derived attributes
    - can be computed from other attributes (e.g., age, given date of birth)
Composite Attributes

Example.

Composite Attributes

- name
  - first-name
  - middle-initial
  - last-name

Composite Attributes

- address
  - street
  - city
  - state
  - postal-code

Component Attributes

- street-number
- street-name
- apartment-number
Composite, Multivalued & Derived Attributes

Example.
Relationship Sets with Attributes

Example

![Diagram showing relationship sets with attributes]
Mapping Cardinality

Figure. An instance of the m:1 relationship set Works_for

Figure. The 1:1 relationship set Manages
Mapping Cardinality

**Figure** The \( M:N \) relationship set Works_on

**Employee**                            **Works_on**                                    **Project**

\( e_1 \) \hspace{1cm} \( e_2 \) \hspace{1cm} \( e_3 \) \hspace{1cm} \( e_4 \) \hspace{1cm} \( e_5 \) \hspace{1cm} \( e_6 \) \hspace{1cm} \( e_7 \) \hspace{1cm} \( e_8 \) \hspace{1cm} \( e_9 \) \hspace{1cm} \( e_{10} \)

\( r_1 \) \hspace{1cm} \( r_2 \) \hspace{1cm} \( r_3 \) \hspace{1cm} \( r_4 \) \hspace{1cm} \( r_5 \) \hspace{1cm} \( r_6 \) \hspace{1cm} \( r_7 \) \hspace{1cm} \( r_8 \) \hspace{1cm} \( r_9 \) \hspace{1cm} \( r_{10} \)

\( p_1 \) \hspace{1cm} \( p_2 \) \hspace{1cm} \( p_3 \) \hspace{1cm} \( p_4 \) \hspace{1cm} \( p_5 \) \hspace{1cm} \( p_6 \) \hspace{1cm} \( p_7 \) \hspace{1cm} \( p_8 \) \hspace{1cm} \( p_{10} \)

**Figure** The recursive relationship set Supervision-Employee plays 2 roles.

‘1’: Supervisor

‘2’: Supervisee
Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
  - e.g., participation of loan in borrower is total
    - every loan must have a customer associated to it via borrower

- **Partial participation**: some entities may not participate in any relationship in the relationship set (default)
  - e.g., participation of customer in borrower is partial

![Diagram](diagram.png)
Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints, but not vice versa (due to the explicit $min..max$ constraint specification)
Weak Entity Sets

- The existence of a weak entity set depends on the existence of an identifying (strong) entity set
  - Must relate to the identifying entity set via a total, many-to-one relationship set from the weak to the identifying entity set
  - Identifying relationship depicted using a double diamond

- An entity set that does not have a primary key is referred to as a weak entity set.

- The discriminator (or partial key) of a weak entity set is the set of attributes that (along with the primary key of its owner entity set) distinguishes among all the entities of a weak entity set.

- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set’s discriminator.
Weak Entity Sets

- We depict a weak entity set by *double rectangles*.
- We underline the *discriminator* of a weak entity set with a dashed line.
  - e.g., *payment-number* – discriminator of the *payment* entity set
- Primary key: for *payment* (*loan-number, payment-number*)
Weak Entity Sets

A weak entity set can participate

- in (regular) relationships other than the identifying relationship
- as owner in an identifying relationship with another weak entity set.
- with more than one identifying entity set and each weak entity is identified by a combination of the identifying entities.

A weak entity set can be expressed as a multivalued, composite attribute of the strong entity set

- More appropriate if (i) it participates in only the identifying relationship, or (ii) its content can never be null.
- Less appropriate if it participates in relationships other than the identifying relationship.
Weak Entity Sets

- Note: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship.

  - If loan-number were explicitly stored, payment could be made a strong entity, but then the relationship between payment and loan would be duplicated by an implicit relationship defined by the attribute loan-number common to payment and loan.

- A weak relationship set should have no descriptive attributes, since any required attributes can be associated with the weak entity set.
Weak Entity Sets

- **Another Example.**
  - In a university, a *course* is a strong entity and a *course-offering* can be modeled as a weak entity.
  - The *discriminator* of *course-offering* would be *semester* (including year) and *section#* (multi-sections).
  - If we model *course-offering* as a weak entity, we would model *course#* as an attribute.
    - Then the relationship with *course* would be implicit in the *course#* attribute.
Entity-Relationship Model

**Definition** A *weak entity set* is a set of weak entities, each of which does not have a primary key of its own.
- Each weak entity is existence-dependent on a strong entity

**Definition** The *primary key* of a weak entity set is formed by the primary key of its strong entity set + its *partial key* (discriminator).
- A *partial key* is a set of attributes that can uniquely identify weak entities related to the same strong entity.

**Definition** *(Keys for Relationship Sets)*

Let $R$ be a relationship set relating entity sets $E_1, E_2, \ldots, E_n$.
Let $\text{attr}(R) = \{a_1, \ldots, a_m\}, m \geq 0$, be the set of (descriptive) attributes in $R$.
Let $\text{fk}(R) = \text{primary_key}(E_1) \cup \text{primary_key}(E_2) \cup \ldots \cup \text{primary_key}(E_n)$

**Case 1.** $\text{Attr}(R) = \emptyset$
- If $R$ is $n$-$m$, then $\text{primary_key}(R) = \text{fk}(R)$.
- If $R$ is 1-$n$, then $\text{primary_key}(R)$ is the primary_key of the “$n$-side.”
- If $R$ is 1-1, then $\text{primary_key}(R) = \text{primary_key}(E_i), 1 \leq i \leq n$, i.e., one of the primary keys of the entity sets.

**Case 2.** $\text{Attr}(R) \neq \emptyset$
- Same as (1) except $S \subseteq \text{Attr}(R)$ may become part of the primary key.
Entity-Relationship (E-R) Diagram

- **Graphically** represents the overall logical structure of a DB

- Representing E-R by **tables**
**E-R Representation**

*Strong entity set* $E$: $a_1, a_2, \ldots, a_n$

*Weak entity set* $A$: $a_1, a_2, \ldots, a_n$
  depending on the strong entity set $B$ w/ primary key $b_1, \ldots, b_m$

*Relationship set* $R$: $E_1, E_2, \ldots, E_m$
  and set of attrs $fk(R)$ and $Attr(R)$

*Relationship set linking weak and strong entity sets*

*Recursive relationship set* $R$: $E_1, E_1$

*Containment (ISA) relationship set* linking higher-level and lower-level entity sets

**Table**

*Table: $E$;*
Columns: $a_1, a_2, \ldots, a_n$;
Rows: entities of $E$.

*Table: $A$;*
Columns: $a_1, \ldots, a_n, b_1, \ldots, b_m$;
Row: entities of $A$.

*Table: $R$;*
Columns: $fk(R)$ and $Attr(R)$;
Row: relationships of $R$.

None

*Table: $R$;*
Columns: $fk(R)$;
Row: relationships of $R$.

None
Specialization

- **Top-down design process**: we designate *subgroupings* within an entity set that are distinctive from other entities in the set.

- These subgroupings become *lower-level* entity sets that have attributes or participate in relationships that do not apply to the *higher-level* entity set.

- Depicted by a *triangle* component labeled ISA (e.g., *customer* “is a” *person*).

- **Attribute inheritance** – a lower-level entity set *inherits* all the *attributes* and *relationship participation* of the higher-level entity set to which it is linked.
An Entity-Relationship Diagram

Example

- STUDENT
  - Name
  - SSNO
  - Status
  - Major

- ISA
  - Assistantship
  - Advisor

- Grad_Std
- Undergrad_Std
  - College
  - Minor
Specialization

Another Example.
Generalization

- A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set.

- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

- The terms specialization and generalization are used interchangeably.
Constraints on a Specialization/Generalization

- **Constraints** on which entities can be members of a given lower-level entity set.

  - **Condition-defined** (i.e., pre-defined constraint)
    - e.g., all customers over 65 years are members of *senior-citizen* entity set; *senior-citizen* ISA person.
  - **User-defined** (default)

- **Constraints** on whether or not entities may belong to more than one lower-level entity set within a single generalization.

  - **Disjoint**
    - an entity can belong to only one lower-level entity set
    - in E-R diagram by writing *disjoint* next to the ISA triangle
  - **Overlapping** (default)
    - an entity can belong to more than one lower-level entity set
Constraints on a Specialization/Generalization

- Completeness constraint -- specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.
  
  - **total**: an entity must belong to one of the lower-level entity sets
  
  - **Partial** (default): an entity need not belong to one of the lower-level entity sets
Symbols of E-R Model

- Many to Many Relationship
- Many to One Relationship
- One to One Relationship
- Cardinality Limits
- Role Indicator
- ISA (Specialization or Generalization)
- Total Generalization
- Disjoint Generalization
Redundancy of Tables

- *Many-to-one* and one-to-many relationship sets that are **total** on the **many**-side can be represented by adding an *extra attribute* to the many side, containing the **primary key** of the one side

- e.g.: Instead of creating a table for relationship **account-branch**, add an attribute **branch** to the entity set **account**
Redundancy of Tables

- For one-to-one relationship sets, either side can be chosen to act as the “many” side
  - i.e., extra attribute can be added to either of the tables corresponding to the two entity sets

- If participation is *partial* on the many side, replacing a table by an extra attribute in the relation corresponding to the “many” side could result in *null values*

- The table corresponding to a *relationship set* linking a weak entity set to its identifying strong entity set is *redundant*.
  - e.g., the payment table already contains the information that would appear in the loan-payment table (i.e., the columns loan-number and payment-number).
Composite and Multivalued Attributes

- **Composite attributes** are flattened out by creating a separate attribute for each *component attribute*
  
  - e.g., given entity set *customer C* with *composite attribute name* and *component attributes first-name and last-name*, the table corresponding to *C* has two attributes
    
    \[ \text{name.first-name} \text{ and name.last-name} \]

- **A multivalued attribute** *M* of an entity *E* is represented by a separate table *EM*
  
  - Table *EM* has attributes corresponding to the *primary key* of *E* and an attribute corresponding to *multivalued attribute M*
  
  - e.g., *multivalued attribute dependent-names* of *employee* is represented by a table
    
    \[ \text{employee-dependent-names(employee-id, dname)} \]
  
  - Each value of the multivalued attribute maps to a separate row of the table *EM*. e.g., (John, \{Mary, Paul\}) maps to two rows: (John, Mary) and (John, Paul)
Method 1:

- Form a table for the higher level entity.
- Form a table for each lower level entity set, include primary key of higher level entity set and local attributes.
- Example.

<table>
<thead>
<tr>
<th>Table</th>
<th>Table Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>name, street, city</td>
</tr>
<tr>
<td>customer</td>
<td>name, credit-rating</td>
</tr>
<tr>
<td>employee</td>
<td>name, salary</td>
</tr>
</tbody>
</table>

- Drawback: getting all information about, e.g., employee, requires accessing two tables.
Representing Specialization as Tables

- **Method 2:**
  - Form a table for each entity set with all local and inherited attributes

<table>
<thead>
<tr>
<th>Table</th>
<th>Table Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>name, street, city</td>
</tr>
<tr>
<td>customer</td>
<td>name, street, city, credit-rating</td>
</tr>
<tr>
<td>employee</td>
<td>name, street, city, salary</td>
</tr>
</tbody>
</table>

- If specialization is *total*, no need to create table for generalized entity (person)

- **Drawback:** street and city may be stored redundantly for persons who are both customers and employees
An Entity-Relationship Diagram

EMPLOYEE
- Name
- Sex
- Address
- Salary
- StartDate
- Hours
- Ssn
- Bdate

DEPENDENT
- Name
- Sex
- BirthDate
- Relationship

DEPENDENTS_OF
- Participate in

SUPERVISOR
- Name
- Number

SUPERVISION
- SUPERVISEE

SUPERVISEE
- Name
- Number

DEPARTMENT
- Name
- Number
- Locations

CONTROLS
- PROJECT
- Name
- Number
- Location

PROJECT
- Name
- Number
- Location

SPORTS
- Title
- Fee

WORKS_FOR
- MANAGES
- WORKS_ON

Participate in

Title

Fee

First

Last

Minit
E-R Diagram for a Banking Enterprise

[Diagram of E-R model for a banking enterprise showing entities such as customer, account, loan, and branches, with relationships and attributes detailed in the diagram.]