Entity-Relationship Model
Data Models

• High-level or conceptual data models provide concepts that are close to the way many users perceive data, whereas low-level or physical data models provide concepts that describe the details of how data is stored in the computer.

• Conceptual data models use concepts such as entities, attributes, and relationships.
  – An entity represents a real-world object or concept, such as an employee or a project, which is described in the database.
  – An attribute represents some property of interest that further describes an entity, such as the employee's name or salary.
  – A relationship among two or more entities represents an association among two or more entities, for example, a works-on relationship between an employee and a project.
Modeling

• A database can be modeled as:
  – a collection of entities,
  – relationship among entities.

• An entity is an object that exists and is distinguishable from other objects.
  – Example: specific person, company, event, plant

• Entities have attributes
  – Example: people have names and addresses

• An entity set is a set of entities of the same type that share the same properties.
  – Example: set of all persons, companies, trees, holidays
History of E-R Diagrams

• ER diagrams are visual tools that are used in the Entity-Relationship model initially proposed by Peter Chen in 1976 to create a uniform convention that considers both relational database and network views.
• Chen envisioned the ER model as a conceptual modeling approach that views real world data as systems of entities and relationships.
• Entities are data objects that maintain different relationships with each other.
• Additionally, entities are also described further using attributes.
## Entity Sets *instructor* and *student*

<table>
<thead>
<tr>
<th>instructor_ID</th>
<th>instructor_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>76766</td>
<td>Crick</td>
</tr>
<tr>
<td>45565</td>
<td>Katz</td>
</tr>
<tr>
<td>10101</td>
<td>Srinivasan</td>
</tr>
<tr>
<td>98345</td>
<td>Kim</td>
</tr>
<tr>
<td>76543</td>
<td>Singh</td>
</tr>
<tr>
<td>22222</td>
<td>Einstein</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>student-ID</th>
<th>student_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>98988</td>
<td>Tanaka</td>
</tr>
<tr>
<td>12345</td>
<td>Shankar</td>
</tr>
<tr>
<td>00128</td>
<td>Zhang</td>
</tr>
<tr>
<td>76543</td>
<td>Brown</td>
</tr>
<tr>
<td>76653</td>
<td>Aoi</td>
</tr>
<tr>
<td>23121</td>
<td>Chavez</td>
</tr>
<tr>
<td>44553</td>
<td>Peltier</td>
</tr>
</tbody>
</table>
Entity

• Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.
Relationship Sets

• A **relationship** is an association among several entities

  Example:
  
  44553 (Peltier) \*advisor\* 22222 (Einstein) \*student\* entity

• A **relationship set** is a mathematical relation among \( n \geq 2 \) entities, each taken from entity sets

  \[
  \{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}
  \]

  where \((e_1, e_2, \ldots, e_n)\) is a relationship
  
  – Example:

  \( (44553,22222) \in \text{advisor} \)
Relationship Set advisor

instructor

- 76766 | Crick
- 45565 | Katz
- 10101 | Srinivasan
- 98345 | Kim
- 76543 | Singh
- 22222 | Einstein

student

- 98988 | Tanaka
- 12345 | Shankar
- 00128 | Zhang
- 76543 | Brown
- 76653 | Aoi
- 23121 | Chavez
- 44553 | Peltier
Relationship Sets (Cont.)

• An **attribute** can also be property of a relationship set.
• For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor.
Degree of a Relationship Set

• **binary relationship**
  – involve two entity sets (or degree two).
  – most relationship sets in a database system are binary.

• Relationships between more than two entity sets are rare. Most relationships are binary. (More on this later.)
  
  Example: *students* work on research *projects* under the guidance of an *instructor*.
  
  relationship *proj_guide* is a ternary relationship between *instructor, student, and project*
E-R Diagram with a Ternary Relationship

- instructor
  - ID
  - name
  - salary

- proj_guide

- student
  - ID
  - name
  - tot_cred

project
Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
  - Example:
    
    \[
    \text{instructor} = (\text{ID}, \text{name}, \text{street}, \text{city}, \text{salary})
    \]
    \[
    \text{course} = (\text{course_id}, \text{title}, \text{credits})
    \]

- **Domain** – the set of permitted values for each attribute

- Attribute types:
  - **Simple** and **composite** attributes.
  - **Single-valued** and **multivalued** attributes
    - Example: multivalued attribute: \text{phone_numbers}
  - **Derived** attributes
    - Can be computed from other attributes
    - Example: age, given date_of_birth
Attributes in E-R Diagram

- Attributes are the properties of entities. Attributes are represented by means of ellipses.
- Every ellipse represents one attribute and is directly connected to its entity (rectangle).
Simple Vs Composite Attributes

- Composite attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings.
Composite Attributes in E-R Diagrams

• If the attributes are composite, they are further divided in a tree-like structure.
• Every node is then connected to its attribute.
• That is, composite attributes are represented by ellipses that are connected with an ellipse.
Single Valued and Multi Valued Attributes

• Most attributes have a single value for a particular entity; such attributes are called single-valued.
  – For example, Age is a single-valued attribute of a person.

• A multi-valued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity.
  – For example, the Colors attribute of a car may have between one and three values, if we assume that a car can have at most three colors.
Multivalued Attributes in E-R Diagrams

- **Multivalued** attributes are depicted by double ellipse.
Stored Attributes Vs Derived Attributes

• In some cases, two (or more) attribute values are related—for example, the Age and BirthDate attributes of a person.
  – For a particular person entity, the value of Age can be determined from the current (today's) date and the value of that person's BirthDate.
  – The Age attribute is hence called a derived attribute and is said to be derivable from the BirthDate attribute, which is called a stored attribute.
Derived Attributes in E-R Diagrams

- Derived attributes are depicted by dashed ellipse.
Mapping Cardinality Constraints

• Express the number of entities to which another entity can be associated via a relationship set.
• Most useful in describing binary relationship sets.
• For a binary relationship set the mapping cardinality must be one of the following types:
  – One to one
  – One to many
  – Many to one
  – Many to many
Mapping Cardinalities

One to one

One to many

Note: Some elements in $A$ and $B$ may not be mapped to any elements in the other set
Mapping Cardinalities

Many to one

Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set
Relationships in E-R Diagrams

1. Entity 1 Relationship 1 Entity
2. Entity 1 Relationship N Entity
3. N Relationship 1 Entity
4. N Relationship N Entity
Relationships - Example

- Employee manages Team
- Customer places order
- Employee reports to Department
- Student enrolls in Classes
Keys

• A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.

• A **candidate key** of an entity set is a minimal super key
  – *ID* is candidate key of instructor
  – *course_id* is candidate key of course

• Although several candidate keys may exist, one of the candidate keys is selected to be the **primary key**.
Weak Entity Sets

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

• The existence of a weak entity set depends on the existence of a **identifying entity set**
  
  – It must relate to the identifying entity set via a total, one-to-many relationship set from the identifying to the weak entity set
  
  – **Identifying relationship** depicted using a double diamond

• The **discriminator (or partial key)** of a weak entity set is the set of attributes that 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

Figure 2.16  E-R diagram with a weak entity set.

Weak Entity Example in ER diagrams
Representing Entity Sets With Simple Attributes

• A strong entity set reduces to a schema with the same attributes
  \textit{student}(\textit{ID, name, tot\_cred})

• A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set
  \textit{section} (\textit{course\_id, sec\_id, sem, year})
Representing Relationship Sets

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.

- Example: schema for relationship set *advisor*

  \[
  \text{advisor} = (s\_id, i\_id)
  \]
Specialization

An entity set may include sub-groupings of entities that are distinct in some way from other entities in the set.

For instance, a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set.

The E-R model provides a means for representing these distinctive entity groupings. Consider an entity set person, with attributes name, street, and city.

A person may be further classified as one of the following:
• Customer
• Employee
Specialization - Example

Diagram showing the relationship between Person, Student, and Teacher.
Specialization - Example

- Person
  - Name
  - Age
  - Gender

  IS A

- Student
  - RollNo.

- Teacher
  - EmpID
Generalization

• This commonality can be expressed by generalization, which is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets.

• In our example, person is the higher-level entity set and customer and employee are lower-level entity sets.

• Higher- and lower-level entity sets also may be designated by the terms super-class and subclass, respectively.
Generalization - Example

- Pigeon
- Sparrow
- Dove

Birds
Figure 2.17  Specialization and generalization.
Extended ER Features: 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.
Total Generalization or Specialization

- Each higher-level entity must belong to a lower-level entity set.

- The account generalization is total: All account entities must be either a savings account or a checking account.
Partial Generalization or Specialization

- Some higher-level entities may not belong to any lower-level entity set.
- When the generalization is partial, a higher-level entity is not constrained to appear in a lower-level entity set.
- The work team entity sets illustrate a partial specialization.
- Since employees are assigned to a team only after 3 months on the job, some employee entities may not be members of any of the lower-level team entity sets.
Disjoint and Overlapping Generalizations

• **Disjoint.** A *disjointness constraint* requires that an entity belong to no more than one lower-level entity set. In our example, an *account* entity can satisfy only one condition for the *account-type* attribute; an entity can be either a savings account or a checking account, but cannot be both.

• **Overlapping.** In *overlapping generalizations*, the same entity may belong to more than one lower-level entity set within a single generalization. For an illustration, consider the employee work team example, and assume that certain managers participate in more than one work team.
Figure 2.20 Symbols used in the E-R notation.
Database Design: Example of Bank

Step 1: Requirements Collection

- The bank is organized into branches.
- Each branch is located in a particular city and is identified by a unique name.
- Bank customers are identified by their customer-id values.
- The bank stores each customer’s name, and the street and city where the customer lives.
- Customers may have accounts and can take out loans.
- A customer may be associated with a particular banker, who may act as a loan officer or personal banker for that customer.
- Bank employees are identified by their employee-id values.
- The bank administration stores the name and telephone number of each employee, the names of the employee’s dependents, and the employee-id number of the employee’s manager.
- The bank also keeps track of the employee’s start date and, thus, length of employment.
The bank offers two types of accounts—savings and checking accounts.

Accounts can be held by more than one customer, and a customer can have more than one account.

Each account is assigned a unique account number.

The bank maintains a record of each account’s balance, and the most recent date on which the account was accessed by each customer holding the account.

In addition, each savings account has an interest rate, and overdrafts are recorded for each checking account.

A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number.

For each loan, the bank keeps track of the loan amount and the loan payments.

Although a loan payment number does not uniquely identify a particular payment among those for all the bank’s loans, a payment number does identify a particular payment for a specific loan.
Step 2: Identification of Entity Sets

• The *branch* entity set, with attributes *branch-name*, *branch-city*, and *assets*.
• The *customer* entity set, with attributes *customer-id*, *customer-name*, *customerstreet*, and *customer-city*. A possible additional attribute is *banker-name*.
• The *employee* entity set, with attributes *employee-id*, *employee-name*, *telephonenumber*, *salary*, and *manager*. Additional descriptive features are the multi-valued attribute *dependent-name*, the base attribute *start-date*, and the derived attribute *employment-length*.
• Two account entity sets—*savings-account* and *checking-account*—with the common attributes of *account-number* and *balance*; in addition, *savings-account* has the attribute *interest-rate* and *checking-account* has the attribute *overdraft-amount*.
• The *loan* entity set, with the attributes *loan-number*, *amount*, and *originatingbranch*.
• The weak entity set *loan-payment*, with attributes *payment-number*, *paymentdate*, and *payment-amount*. 
Step 3: Identification of Relationship Sets

- Borrower, a many-to-many relationship set between customer and loan.
- Loan-branch, a many-to-one relationship set that indicates in which branch a loan originated. Note that this relationship set replaces the attribute originatingbranch of the entity set loan.
- Loan-payment, a one-to-many relationship from loan to payment, which documents that a payment is made on a loan.
- Depositor, with relationship attribute access-date, a many-to-many relationship set between customer and account, indicating that a customer owns an account.
- Cust-banker, with relationship attribute type, a many-to-one relationship set expressing that a customer can be advised by a bank employee, and that a bank employee can advise one or more customers.
Figure 2.22  E-R diagram for a banking enterprise.
Summary of Symbols Used in E-R Notation

- **E**: entity set
- **R**: relationship set
- **E R**: relationship set for weak entity set
- **E R E**: total participation of entity set in relationship

**Attributes**:
- Simple (A1)
- Composite (A2)
- Multivalued (A3, denoted by [A3])
- Derived (A4)

**Primary Key**:
- A1

**Discriminating Attribute**:
- A1

**Relationship**:
- Arrows indicate the direction of the relationship.
Symbols Used in E-R Notation (Cont.)

- **R** (relationship)
- **E** (entity)
- **Roles**
  - many-to-many relationship
  - one-to-one relationship
- **Role Name**
- **Role Indicator**
- **Cardinality Limits**
- **ISA: Generalization or Specialization**
- **Total (Disjoint) Generalization**
- **Disjoint Generalization**
Crow’s Foot/Martin/Information Engineering style

Entity

Attribute
Attribute
Attribute

Multiplicity of many

Optional

Multiplicity of one

Optional

Multiplicity of many

Mandatory

Multiplicity of one

Mandatory
### Example – Crow’s Foot

**Marks**
- mark id: integer
- student id: integer
- subject id: integer
- date: date/time
- mark: integer

**Students**
- Student id: integer
- first name: varchar
- last name: varchar
- group id: integer

**Subjects**
- subject id: integer
- title: varchar

**Subject/teacher**
- subject id: integer
- teacher id: integer
- group id: integer

**Groups**
- group id: integer
- name: varchar

**Teachers**
- teacher id: integer
- first name: varchar
- last name: varchar