A schema represents the overall logical structure of an information system. This overall structure can be expressed graphically by an E-R diagram that exists in the real world and is distinguishable from the conceptual model. Express the distinction by associating with each entity a set of attributes that describes the object.

An association among several entities. A **relationship set** is a collection of relationships of the same type, and an **entity set** is a collection of entities of the same type.

Key, candidate key, and primary key apply to entity and relationship sets. They do for relation schemas. Identifying the primary key is not sufficient, since it is composed of attributes of the related entity sets.

**Entities** express the number of entities to which another entity is related via a relationship set. They do not have a sufficient number of attributes to form a primary key. An entity set that has a primary key is termed a **primary key set**.

Entities and attributes of the E-R model offer the database designer numerous ways to represent the enterprise being modeled. Concepts in certain cases, be represented by entities, relationships, or sets of the overall structure of the enterprise may be best modeled by weak entity sets, generalization, specialization, or aggregates. The database designer must weigh the merits of a simple, compact model against a more precise, but more complex, one.

Relationships specified by an E-R diagram can be represented by a schema. For each entity set and for each relationship set, there is a unique relation schema that is assigned the respective entity set or relationship set. This forms the basis for the design of a relational database from an E-R diagram.

**Generalization** define a containment relationship between an entity set and one or more lower-level entity sets. The result of taking a subset of a higher-level entity set to form a lower-level entity set. Generalization is the result of taking the union of two or more (lower-level) entity sets to produce a higher-level entity set. Higher-level entity sets are inherited by lower-level entity sets.

**Aggregation** is a containment relationship in which relationship sets (along with their attributes) are treated as higher-level entity sets, and can participate in a modeling language. UML class diagrams are widely used as well as for general purpose data modeling.

### Review Terms
- Entity-relationship data model
- Entity and entity set
  - Attributes
  - Domain
  - Simple and composite attributes
  - Single-valued and multivalued attributes
  - Null value
  - Derived attribute
  - Superkey, candidate key, and primary key
- Relationship and relationship set
  - Binary relationship set
  - Degree of relationship set
  - Descriptive attributes
  - Superkey, candidate key, and primary key
  - Role
- Recursive relationship set
- E-R diagram
- Mapping cardinality:
  - One-to-one relationship
  - One-to-many relationship
  - Many-to-one relationship
  - Many-to-many relationship
- Participation
  - Total participation
  - Partial participation
- Weak entity sets and strong entity sets
  - Discriminator attributes
  - Identifying relationship
- Specialization and generalization
  - Superclass and subclass
  - Attribute inheritance
  - Single and multiple inheritance
  - Condition-defined and user-defined membership
  - Disjoint and overlapping generalization
  - Total and partial generalization
- Aggregation
- UML
- UML class diagram

### Practice Exercises

7.1 Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.

7.2 Consider a database used to record the marks that students get in different exams of different course offerings (sections).
7.14 Explain the distinctions among the terms primary key, candidate key, and superkey.

7.15 Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

7.16 Construct appropriate relation schemas for each of the E-R diagrams in Practice Exercises 7.1 to 7.3.

7.17 Extend the E-R diagram of Practice Exercise 7.3 to track the same information for all teams in a league.

7.18 Explain the difference between a weak and a strong entity set.

7.19 We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?

7.20 Consider the E-R diagram in Figure 7.29, which models an online bookstore.
   a. List the entity sets and their primary keys.
   b. Suppose the bookstore adds Blu-ray discs and downloadable video to its collection. The same item may be present in one or both formats, with differing prices. Extend the E-R diagram to model this addition, ignoring the effect on shopping baskets.
   c. Now extend the E-R diagram, using generalization, to model the case where a shopping basket may contain any combination of books, Blu-ray discs, or downloadable video.

7.21 Design a database for an automobile company to provide to its dealers to assist them in maintaining customer records and dealer inventory and to assist sales staff in ordering cars.
   Each vehicle is identified by a vehicle identification number (VIN). Each individual vehicle is a particular model of a particular brand offered by the company (e.g., the XF is a model of the car brand Jaguar of Tata Motors). Each model can be offered with a variety of options, but an individual car may have only some (or none) of the available options. The database needs to store information about models, brands, and options, as well as information about individual dealers, customers, and cars.
   Your design should include an E-R diagram, a set of relational schemas, and a list of constraints, including primary-key and foreign-key constraints.

7.22 Design a database for a world-wide package delivery company (e.g., DHL or FedEx). The database must be able to keep track of customers (who ship items) and customers (who receive items); some customers may do both.