The Entity-Relationship Model
Aggregation and Class Hierarchies

Aggregation

Aggregation: Participation of a *relationship set* in another *relationship set.*

Example. We continue modifying the Library database.

> *Sometimes, library patrons do not return books on time. In such cases, the library assesses late fees (fines) for the books on-loan. Fees are assessed for each day the book is late, and each assessment needs to be recorded separately, together with the assessed amount.*

To model this, we create (see Figure **??**) a new entity set, Fines, which has two attrbutes, Date_Assessed and Amount. We note that entities from this entity set are associated neither with just the books, nor with just memberships, but rather *with the acts of borrowing books by patrons.* These acts are modeled by the Loans relationship set. We use aggregation to make Loans be "the other end" of the Causes relationship set, which associates the fines with their causes.

We also note that Fines is a *weak entity set*, and that the aggregation of Loans serves as its *identifying owner*, with Date_Assessed being the *discriminator*.

Class Hierarchies

When *two or more entity sets* are viewed as a part of one "virtual" entity set, we talk about **class hierarchy** and **inheritance**.

There are two possible ways to view a *class hierarchy.*

- **Specialization**. Sub-classes represent more specific types of entities that belong to a super-class entity set.
- **Generalization**. Superclass *generalizes* the properties of different subclasses.

Attribute Inheritance: subclasses **inherit** the attributes of the superclass (while adding more specific attributes.

Overlap constraints: determine whether two subclasses are disjoint or not (i.e. are allowed to contain same entities).

- disjoint
- overlapping

Covering constraints: determine whether subclasses collectively store *all* entities of the super-class (i.e., whether there can be superclass entities *not belonging* to any subclass).

- total generalization/specialization.
- partial generalization/specialization.

Example. Our final revision of the Library dataset, adds a class hierarchy to it.

> *The library now revises its membership policies, and establishes two categories of membership: business membership and personal membership. Personal membership is issued to households, while business membership is issued to businesses operating in the area. The library requires collection of significantly more detailed information about a business than it collects about individual memberships, but it also provides additional services to the businesses. Each business can receive only one business membership.*

We model this as follows. First, we define two entity sets for the new types of memberships: Business_Memberships and Personal_Memberships. We move the part of our original model dealing with memberships to the Pesronal_Memberships entity set: it now becomes the identifying owner of the the entity set Individuals — a renamed version of the original People entity set — via the Has_Membership relationship set.

We also create a new entity set Businesses. Unlike Individuals, Businesses is a strong entity set, as we collect enough information about a business to uniquely identify it. We add a relationship set Has_Membership between Businesses and Business_Memberships. This relationship set is one-to-one (each business can have only one membership, and each membership is associated with one business), and onto Business_Memberships. We also add referential integrity constraints on both sides.

Finally, we note, that while two different types of memberships behave differently w.r.t. the members who have them, they do not differ w.r.t. the procedure of loaning books. To model this, we keep the orignal Memberships entity set, and are making it a generalization of Personal_Memberships and Business_Memberships entity sets. The class hierarchy created this way will be disjoint and will have complete coverage.

Entity-Relationship diagrams: Part IV

- **Generalization/specialization**: Triangle with the **ISA** label. Edges connect the triangle to the the parent and the child entity sets in the hierarchy.
- **Aggregation**: Dashed box around the relationship set.

Example. Figure **??** shows the final E-R diagram for the Library dataset. This diagram includes the aggregation of the Loans relationship set, to model the assessment of fines. It also includes the class hierarchy built to model the two different types of membership offered by the library.

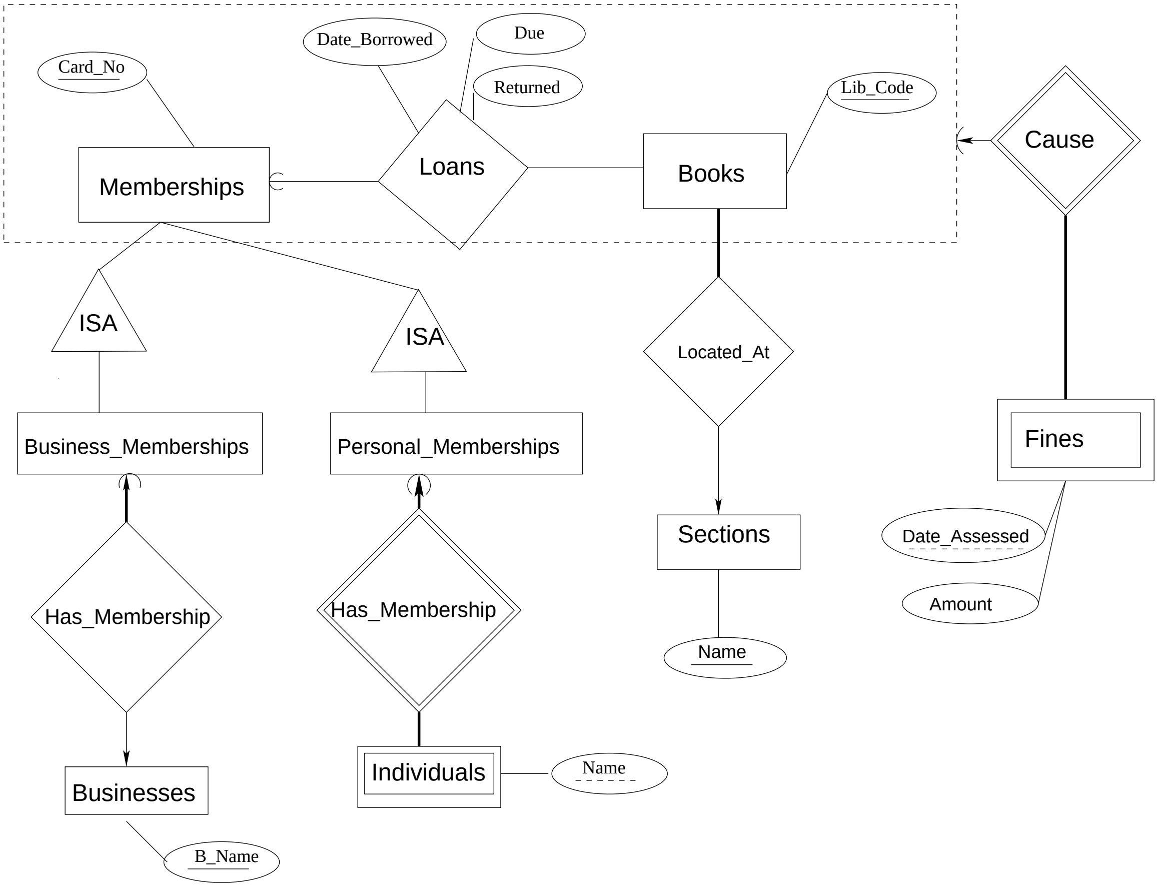

Figure 1: Revised E-R diagram for the Library database. This diagram shows aggregation and a simple class hierarchy.