The Entity-Relationship Model (E-R Model)

In a Nutshell

E-R model:

- Graphical;
- Models *Entities* (data objects, units of information) and ...
- *Relationships* between *entities*.

In More Detail

Entities

Entity: *object in a real world, different from other objects.* Unit of data.

Entity set: collection of similar *entities*.

Attribute: a single *feature* of an *entity*.

Entities from same *entity set* have the **same** set of *attributes* (i.e. are identifiable by the same set of features).

Attributes take values from **domains**.

Key: *minimal* set of *attributes* whose values uniquely identify an *entity*.

If more than one *key* exists, one of them is designated as **primary key**.

Instance of an *entity set*: set of specific entities.

Relationships

Relationship: *association* between two or more *entities*.

Relationship set: *association* between two or more *entity sets*.

Descriptive attributes: *features* of relationships.

Instance of a *relationship set*: set of *relationships*.

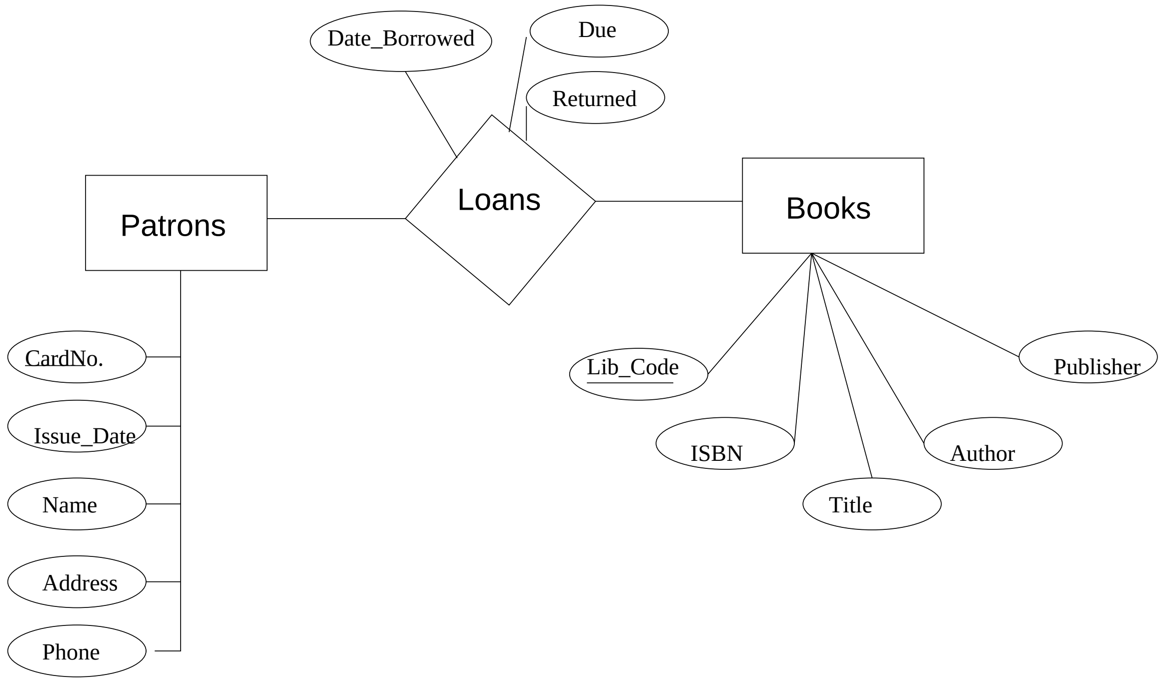


Figure 1: Simple E-R diagram for the Library database.

Entity-Relationship Diagrams: part I

Entity-Relationship Diagram (ER diagram, E-R diagram): graph representing the conceptual model of a database using the conventions outlined below.

- **Entity Set:** Rectangle with entity set *name* inside.
- **Attribute:** Oval with attribute name inside.
- **Primary Key Attribute:** underlined attribute name.
- **Association** of *attributes* with *entity sets*: Edges between ovals and rectangles.
- **Relationship Sets:** Diamond with relationship set name inside.
- **Descriptive Attributes:** *attributes* (ovals) connected directly to *relationship sets* (diamonds).

Example. Figure 1 shows an E-R diagram for a simple library database. The database description is as follows:

> *The library keeps tracks of books and library patrons who borrow books. A book is identified uniquely by a library code. Other information available about the book is its ISBN number, title, author, publisher and the year of publication.*
>
> *A patron of the library must have a membership card. The card number is used to uniquely identify each patron. Other information available about each patron is the name, address, phone, and the issue date of the membership card.*
>
> *Patrons borrow books. For each book loan, the date of the loan, the due date and the date the book was returned are recorded.*

Properties of Entity and Relationship Sets

Entity and relationship sets are subjects to **constraints**. **Constraints** identify special conditions or restrictions on *entities* in an entity set and/or on *relationships* between *entities*.

Constraints on Entity Sets

- Superkeys. A superkey is any collection of attributes, which uniquely identifies each entity in an entity set.
- Candidate Keys. A candidate key is the smallest combination of attributes that uniquely identifies an entity in an entity set. Multiple candidate keys can exist in an entity set.
- Primary Keys: A primary key is one of the candidate keys of an entity set, chosen to be the *chief means of identifying entities in the database*.

Note: Only primary keys of entity sets have graphical representation in E-R diagrams (underlined attributes). However, if *other* important candidate keys exist in an entity set, they must be identified in the conceptual model.

For example, in the Books entity set (see Figure 1), ISBN is a candidate key. This information should be stated explicitly in the Library database design.

Constraints on Relationship Sets

- Multiplicity of (binary) relationship sets. Consider a relationship set R between two entity sets E and F.
  - one-to-one: each entity of E can participate in *at most one* relationship R and each entity of F can participate in *at most one* relationship R. (in this case, each entity from E can be connected to at most one entity from F via R and vice versa).

    Example. In a database tracking basketball season, entity sets Coaches and Teams are in a *one-to-one* relationship is_Coach_of. Each coach can be a head coach at only one team at a time, and each team has only one head coach.
  - many-to-one: each entity of E can participate in *at most one* relationship R, while entities of F can participate in multiple relationships R. (in this case, each entity from E can be connected via R to at most one entity from F, but entities from F can be connected to multiple entities from E via R).

    Example. In a database tracking basketball season, entity sets Players and Teams are in a *many-to-one* relationship Plays_For. Each player can play at only one team at each moment of time, but a team can have multiple players playing for it.
  - many-to-many: each entity of E can participate in *multiple* relationships R and each entity of F can participate in multiple relationships R.

    Example. In a database tracking basketball season, entity sets Players and Games are in a *many-to-many* relationship Plays_in. Each player can play in multiple games, and multiple players play in each game.
  - degree constraint: a degree constraint on the entity set E in R limits the number of entities from E that can be associated by R with the same entity in F. Degree constraints restrict the multiplicity of many-to-many relationship sets.

Example. In a database tracking basketball season, the relationship set Plays_for is a degree constraint ≤ 15 associated with the Players entity set. This constraint states that no more than 15 players can be at the same time on the roster of any team.

- Participation constraints. These constraints specify that **every** entity from a particular entity set must participate in a specific relationship set.

  Example. Consider the basketball season database and the relationship set is_Coach_of between Coaches and Teams entity sets. There is a *participation constraint* on Teams associated with this relationship set: each team must have a head coach. At the same time, there is no participation constraint on Coaches: while a coach cannot coach more than one team at a time, some coaches may be without work for periods of time.

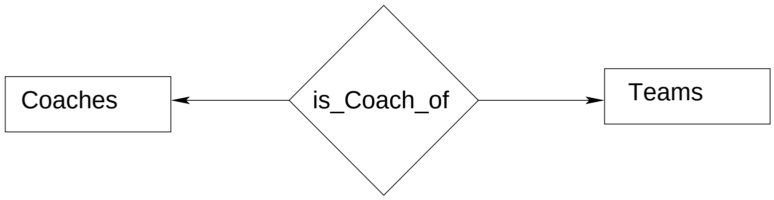
- Referential integrity constraints. Relationship sets contain "pointers" to the entities that form individual relationships. Referential integrity constraints mean that the pointers must not be "dead", i.e., they point to an existing entity in an entity set.

  Example. Consider the basketball season database and the relationship set is_Coach_of between Coaches and Teams entity sets. We can say, that this relationship set entails a *referential integrity constraint* on the Coaches entity set. This means, that an entity for each current head coach *must be present* in the Coaches entity set.

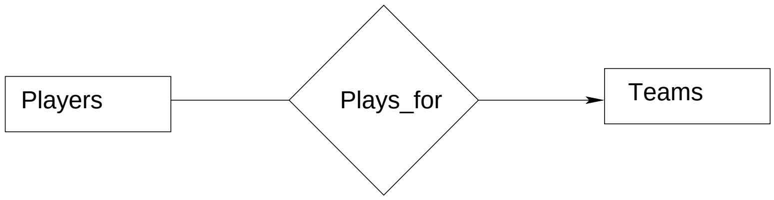
Entity-Relationship Diagrams, Part II

.

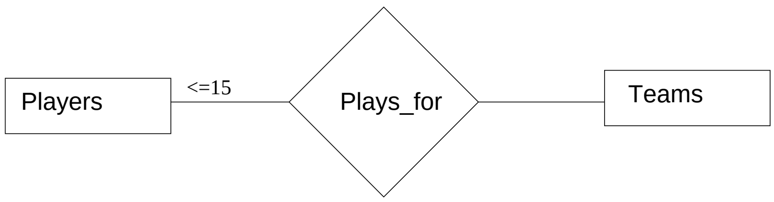
- One-to-one relationship sets are identified by drawing **directed edges** from the relationship set *towards all entity sets*



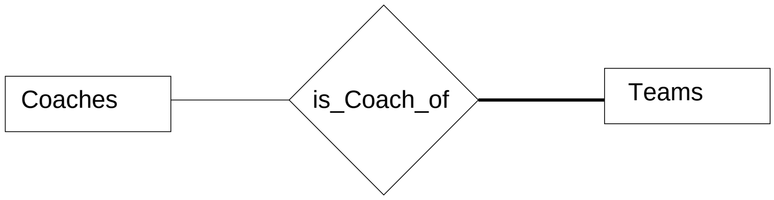
- Many-to-one relationship sets are identified by drawing a **directed edge** from the relationship set *towards the constraining entity set.*



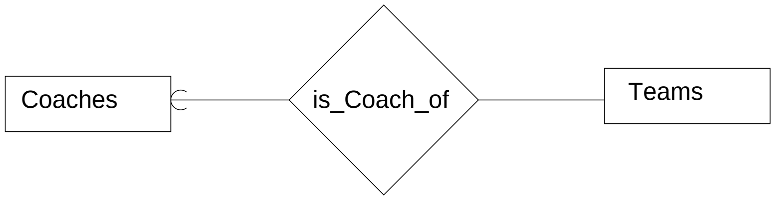
- Degree constraints are identified by **writing the constraint above the edge** connecting the relationship set with the *constrained entity set.*



- Participation constraints are identified by making the edge connecting the relationship set with the *constrained entity set* **bold**.



- Referential Integrity constraints are identified by a **small semi-circular arrow** pointing towards the entity set on which the referential integrity constraint is to be enforced.



Weak Entities

Weak Entities: entities without a *primary key.*

Weak Entity Sets: sets of *weak entities*

- Two (or more) *different weak entities* with *same attributes* can be present in a *weak entity set.*

Identifying Owner: A (strong) *entity* that allows to certify the identity of a *weak entity.*

Identifying Relationship: A *relationship* between a *weak entity* and its *identifying owner* that certifies the identity of a *weak relationship.*

- *One-to-many relationship* (key constraint on weak entity set).
- *Total relationship* for the weak entity set.

Discriminator: a set of attributes in a *weak entity* that identify it in conjunction with the key of its *identifying owner.*

Example Let us revise the description of the Library database, to include the following information.

> *The library issues one library membership per household. Multiple members of the household can use the membership to borrow books. The address and phone number are the features of the household, not individual household members. The library has no way of distinguishing between two patrons with the same name, other than by means of their membership numbers. However, the library wants to distinguish between the memberships and people who use them.*

In this case, the original entity set Patrons needs to be split into two: Memberships and People. The former will keep card number, address, phone and date of issue attributes, while the latter gets the name attribute. But different library patrons may have the same name. Without further identification, they will now be indistinguishable in the People entity set, if we use Name as the primary key for People.

Our solution is to make People a *weak entity set*, and make Memberships its *identifying owner* through an identifying relationship *Has_Membership*. Name becomes a discriminator in the entity set People.

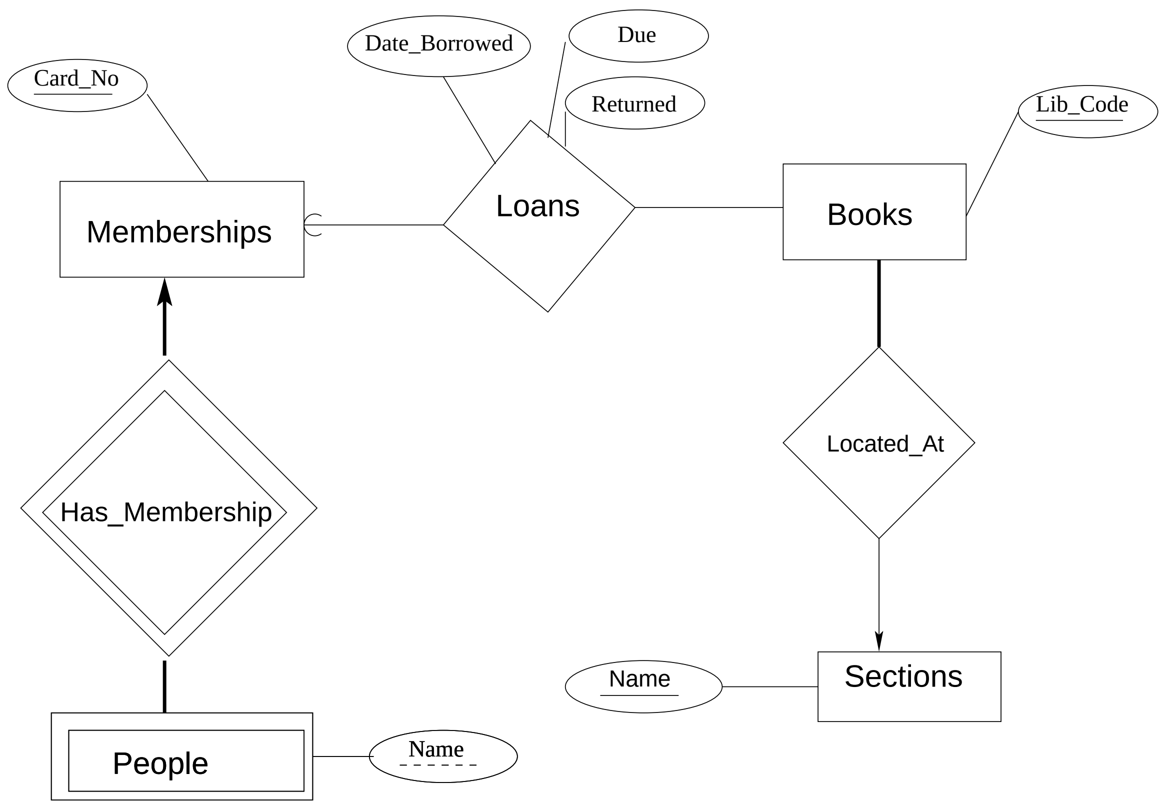


Figure 2: E-R diagram for the updated Library database.

.

Entity-Relationship Diagrams, Part III

.

- **Weak Entity Set**: Double rectangle
- **Identifying Relationship Set:** Double diamond.
- **Discriminator**: Dashed underlined attribute names.

Simplified E-R diagram: E-R diagram showing only primary key, discriminating and identifying attributes of entity sets, weak entity sets and relationship sets respectively.

Example We revise the Library database again, to add one more information to the database description:

> The library is broken into a number of sections. Each section is uniquely identified by its name. Each book has a specific location in exactly one section.

Figure 2 shows the simplified E-R diagram for the updated Library database specifiction (this includes both the weak entity set and the Sections entity set). Notice, that the design also specifies the types of relationship sets. The Loans relationship set is many-to-many and has an integrity constraint on Memberships, the Located_At relationship set is many-to-one and onto (participation constraint) for Books.