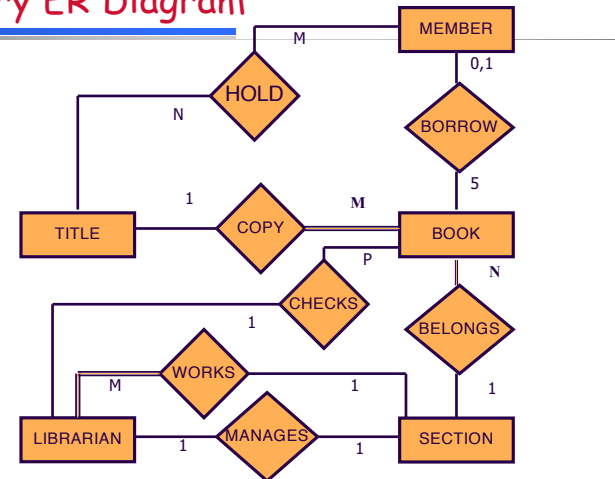


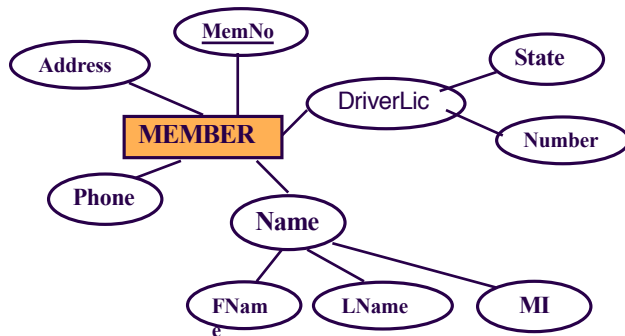
## Case Study: Library Database System

- ❑ Library organized into sections, like art, children, computing, science, etc. Each section has name and a number and its headed by a head librarian
- ❑ Each book title belongs to a section and has a title, authors, ISBN, call number, year and publisher
- ❑ For each copy of the book keep track the current borrower, the due date and the librarian who charged it out.
- ❑ Members have membership number, a driver's license, an address, a phone number and birthday
- ❑ Members can have up to 5 borrowed books and can put a hold request on a book.
- ❑ Librarians have a name, ssn, address, phone

## Library ER Diagram



## Library ER Diagram...



## Entities

1. **TITLE**: CallNumber, Name, Author{(Name(Fname, MI, Lname),Order)}, ISBN, Year, Publisher;
2. **MEMBER**: MemNo, DriverLic(State,No), Name(Fname, MI, Lname), Address, PhoneNumber;
3. **BOOK**: BookID, Edition;
4. **LIBRARIAN**: SSN, Name, Address, Salary, Gender, Date of Birth;
5. **SECTION**: SectNo, Name;

## Weak Entity

- Assume the additional requirement that all the dependents of each librarian are stored in the DB

1. **DEPENDENT**: Name, Date of Birth, Kinship

## Relationships

- COPY**: <TITLE, BOOK> 1:M, PARTIAL/TOTAL;
- BELONGS**: <BOOK, SECTION> N:1, TOTAL/PARTIAL;
- HOLD**: <MEMBER, TITLE> M:N, PARTIAL/PARTIAL, Date;
- BORROW**: <MEMBER, BOOK> 1:5, PARTIAL/PARTIAL, BorrowDueDate;
- CHECKS**: <LIBRARIAN, BOOK> 1:N, PARTIAL/PARTIAL;
- MANAGES**: <LIBRARIAN, SECTION> 1:1, PARTIAL/PARTIAL;
- WORKS**: <LIBRARIAN, SECTION> 1:N, TOTAL/PARTIAL;
- DEPENDS**: <LIBRARIAN, DEPENDENT> 1:N, PARTIAL/TOTAL;
- SUPERVISES**: <supervisor-LIBRARIAN, supervisee-LIBRARIAN> 1:N, PARTIAL/PARTIAL;

## Assumptions/Clarifications:

- One author writes one or more titles.
- Several co-authors write one or more titles.
- A book is a copy of a title. A title can have one or more copies of the book.
- A book has a unique id (not a copy id). If a copy id is used then book is a weak entity type.
- A particular member places a hold on a particular title.
- Not all members necessarily borrow books. Not all books are necessarily borrowed.
- Not all titles need necessarily be of books. However, all books must have a title and only one title.

## Conceptual Database Design & ER-Model

- ER-Model
- ER-Diagrams
- EER Model & Diagrams

## EER Model: Enhanced ER Model

- ❑ The **EER** model introduced the concepts of *superclass* and *subclass* entity types in the ER model
  - **MEMBER (superclass):**
    - LIFE-MEMBER, REGULAR-MEMBER, and SEASON-MEMBER (Subclasses)
  - **LIBRARIAN (superclass):**
    - HEAD LIBRARIANS, SALARY LIBRARIANS, and HOURLY LIBRARIANS (subclasses)

## Why EER Model ?

- ❑ To add more semantic clarity to the design
- ❑ E.g., if only salary-librarians can belong to the librarian guild, then this can be expressed as
  - BelongTo: <SALARY-LIBRARIAN, LIB-GUILD>
  - and not as
  - BelongTo: <LIBRARIAN, LIB-GUILD>
- ❑ Minimize NULL values

## Specialization, Generalization, Inheritance

- ❑ Specialization: identifying subclasses, and their distinguishing characteristics (attributes & relationships)  
(Top-down design)
- ❑ Generalization: aggregate entities to a superclass entity type by identifying their common characteristics  
(Bottom-up design)

## Specialization, Generalization, Inheritance

- ❑ Inheritance: IS\_A (instance) relationship that supports attribute inheritance and relationship participation
  - Single inheritance results in a hierarchy
  - Multiple inheritance results in a lattice



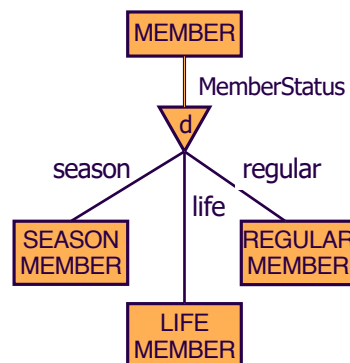
## Inclusion Constraints

- ❑ The disjoint constraint: the subclasses of a superclass are disjoint.
  - This means that an entity can be a member of only one subclass.
  - The entities for each class can be *user-defined* or specified with a *predicate-defined subclass*.
  - In a predicate-defined subclass, we use a selection condition on one or more attributes to define the entities of the subclass. E.g., MembershipStatus
- ❑ The non-disjoint constraints: specify that the subclasses are overlapping and an entity may be a member of more than one subclass.

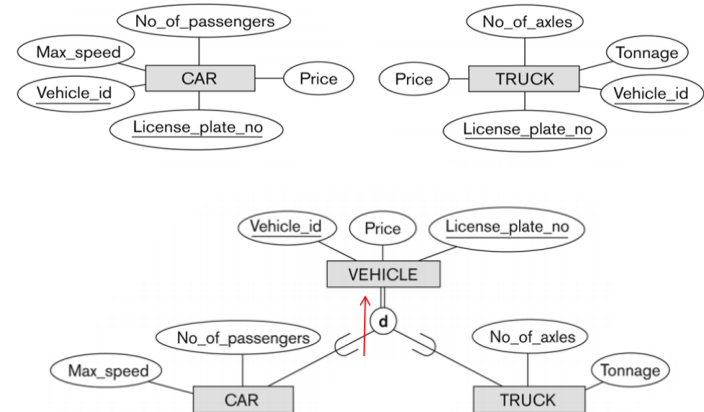
## Completeness Constraints

- ❑ A total specialization: specifies that every entity in the superclass must be a member of some of its subclasses
  - E.g., a librarian must belong to one of the subclasses of LIBRARIAN.
- ❑ A partial specialization: specifies that an entity may not belong to any subclass
  - E.g., an honorary member may not belong to any of the specializations (subclasses) of MEMBER.
- ❑ Superclass via generalization is always total

## EER Diagram: Example 1

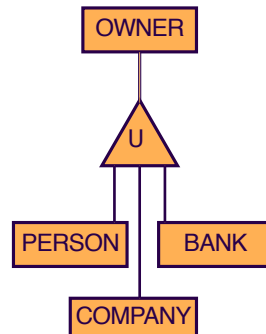


## EER Diagram: Example 2

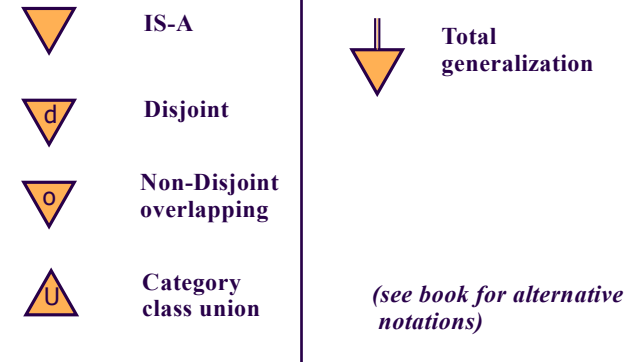


## Union Types or Categories

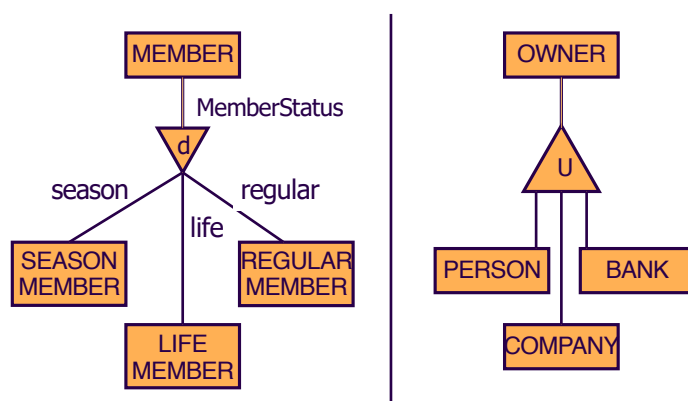
- Collection of entities of distinct entity types
- Category OWNER is a subclass of the set **union** of the entity types: PERSON, BANK, COMPANY
- Multiple Inheritance with superclasses of different types
- An instance in category must exist only in one of the superclasses
- Category can be:
  - Total** or **Partial** (with predicate definition)



## EER Diagrams



## EER Diagram: Examples



## UNIFIED MODELING LANGUAGE & OMT (Object Modeling Technique)

- Describe software modules and their interactions including data requirements via diagrams
- They include *class diagrams* which are similar to EER diagrams — *same concepts different names*
  - Entity = object; Entity type = class
- In UML, a class is a box with three sections:
  - Class name, Object attributes, Object operations
- relationships are called *associations*, relationship instances are called *links* that have *link attributes*
- Relationship constraints are called *multiplicities*
- Two types of relationships: *association* & *aggregation* with directionality of access

## Library UML Diagram

