Lecture 04: Conceptual Database Design & ER-Model

CS 1555: Database Management Systems

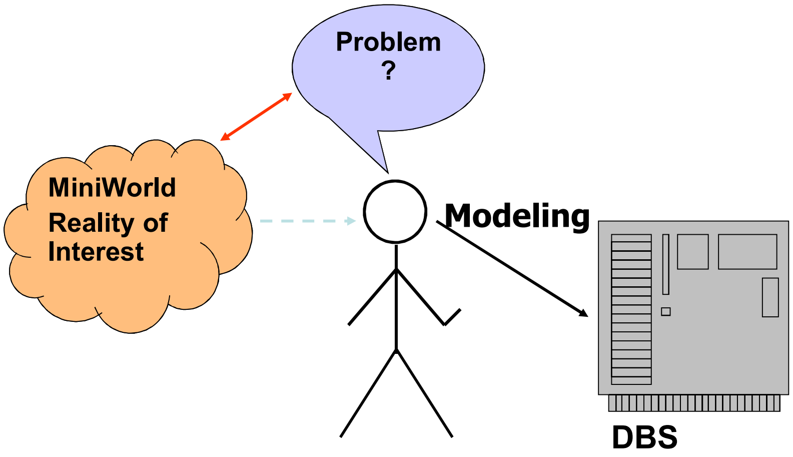
Constantinos Costa

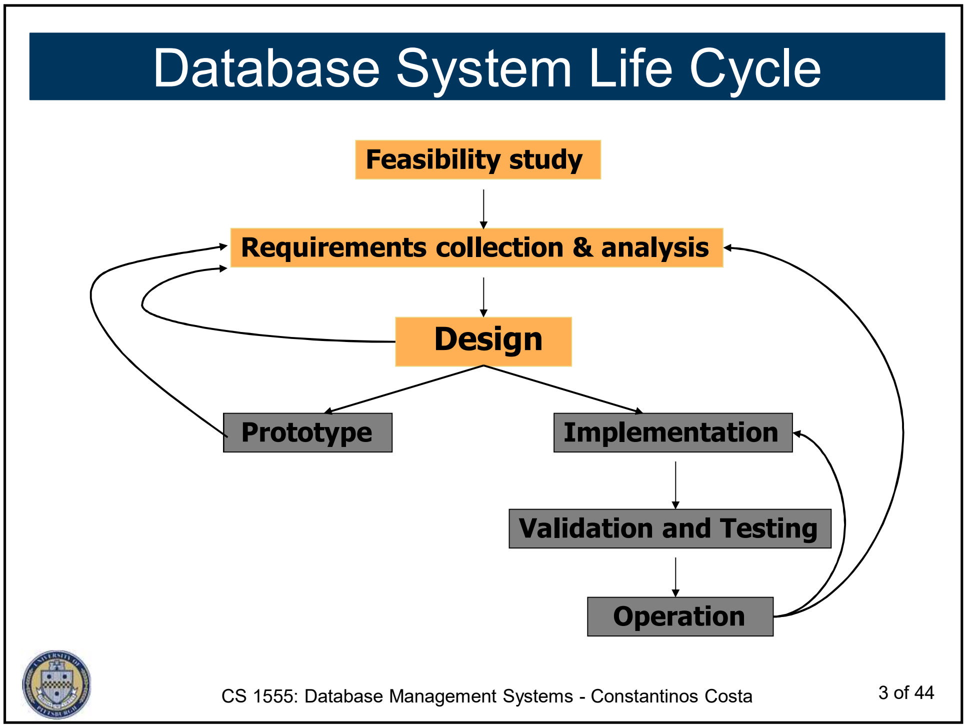
http://db.cs.pitt.edu/courses/cs1555/current.term/

Jan 17, 2019, 16:00-17:15
University of Pittsburgh, Pittsburgh, PA

Lectures based: P. Chrysanthis & N. Farnan Lectures

Database System Design/Data Modeling





Database System Life Cycle

Feasibility study

Requirements collection & analysis

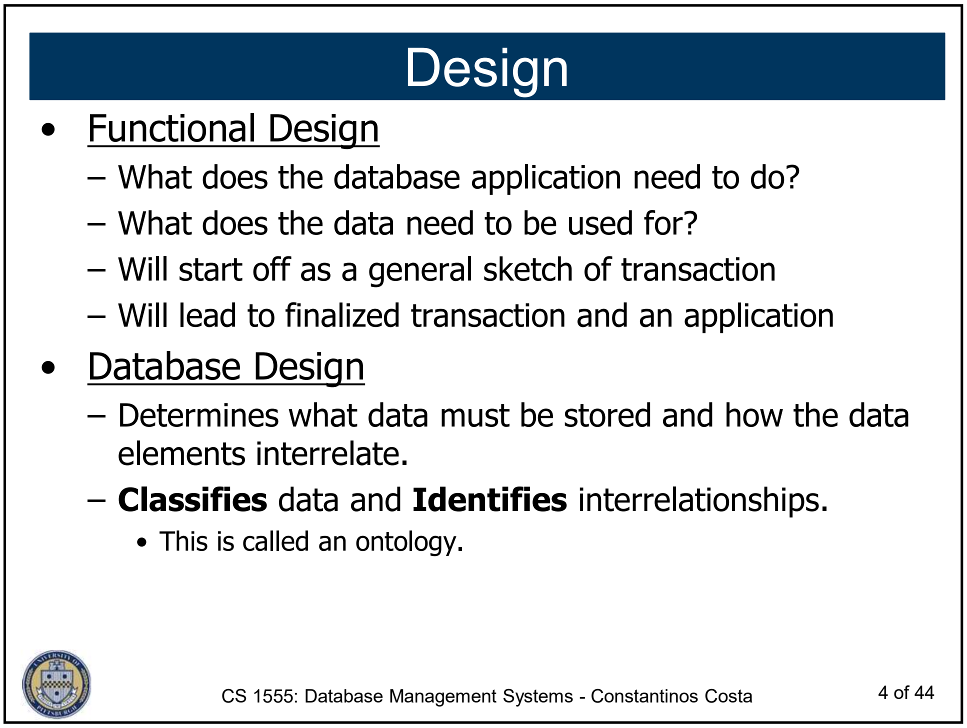
Design

Prototype

Implementation

Validation and Testing

Operation



Design

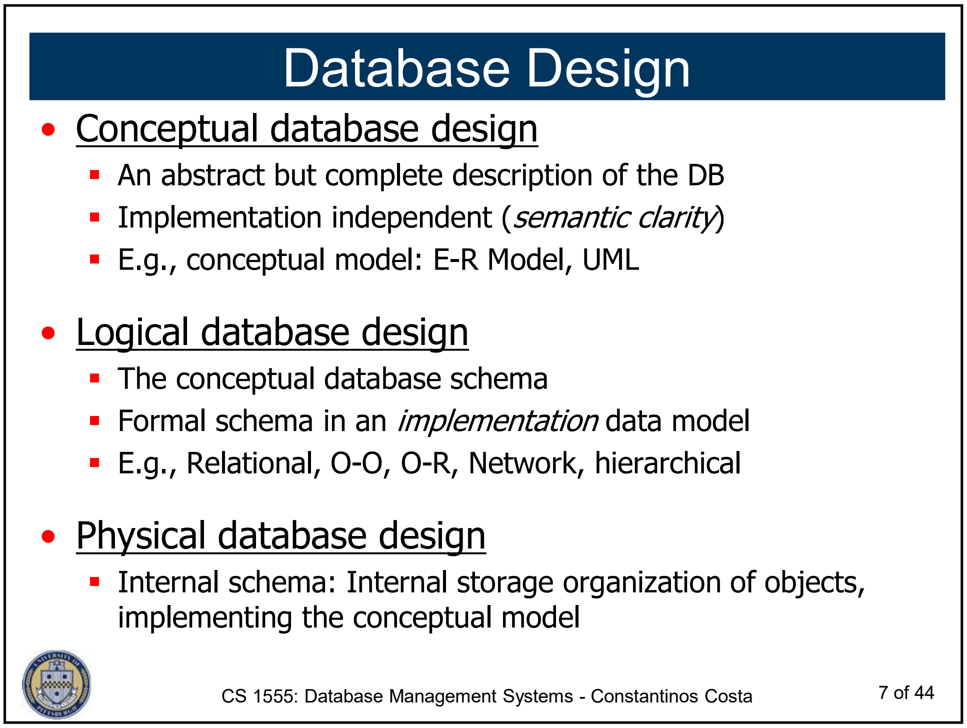
- Functional Design
  - What does the database application need to do?
  - What does the data need to be used for?
  - Will start off as a general sketch of transaction
  - Will lead to finalized transaction and an application
- Database Design
  - Determines what data must be stored and how the data elements interrelate.
  - **Classifies** data and **Identifies** interrelationships.
    - This is called an ontology.

Functional Design

- High-level specification of Transactions
  - DBMS-independent
  - Even diagrams, UML
- Application program design
  - DBMS-specific (db Schema together with DML)
  - Language and environment-specific

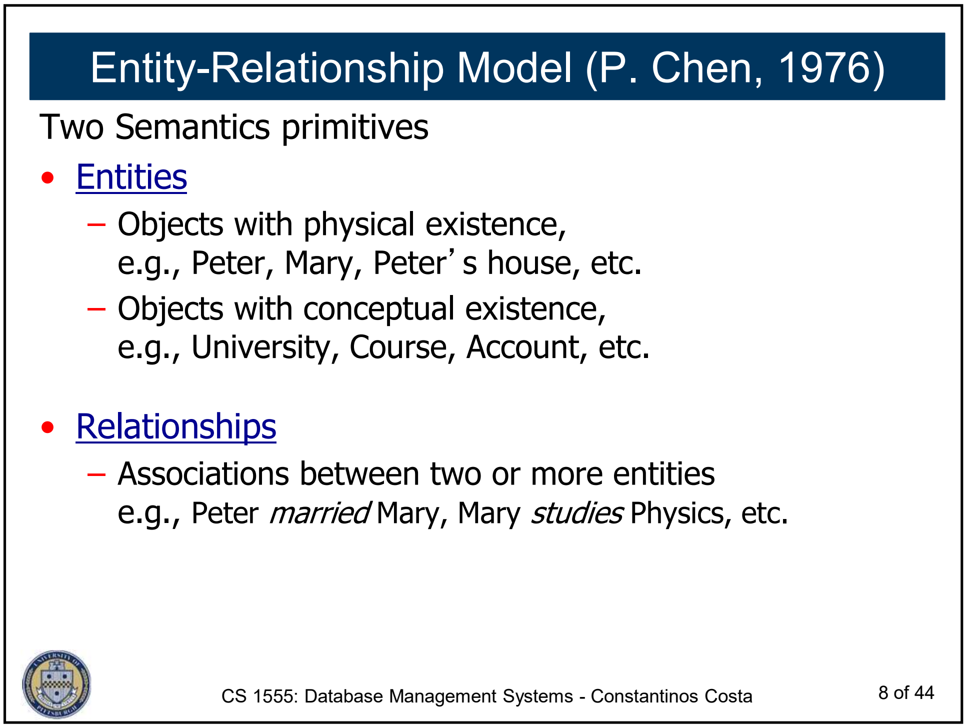
Database Design

- Database design is the activity of specifying the schema of a database in a given data model
- Three categories:
  - Conceptual database design
  - Logical database design
  - Physical database design



Database Design

- <u>Conceptual database design</u>
  - An abstract but complete description of the DB
  - Implementation independent (*semantic clarity*)
  - E.g., conceptual model: E-R Model, UML
- <u>Logical database design</u>
  - The conceptual database schema
  - Formal schema in an *implementation* data model
  - E.g., Relational, O-O, O-R, Network, hierarchical
- <u>Physical database design</u>
  - Internal schema: Internal storage organization of objects, implementing the conceptual model



Entity-Relationship Model (P. Chen, 1976)

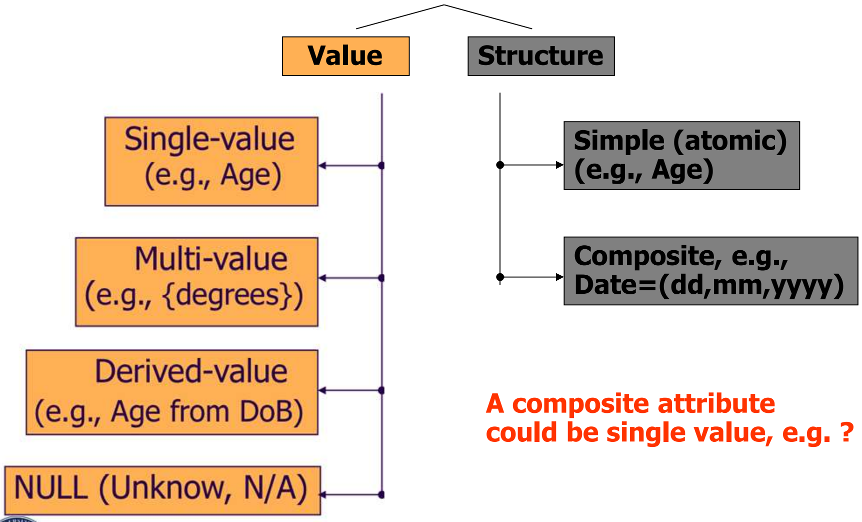
Two Semantics primitives

- <u>Entities</u>
  - Objects with physical existence, e.g., Peter, Mary, Peter’ s house, etc.
  - Objects with conceptual existence, e.g., University, Course, Account, etc.
- <u>Relationships</u>
  - Associations between two or more entities e.g., Peter *married* Mary, Mary *studies* Physics, etc.

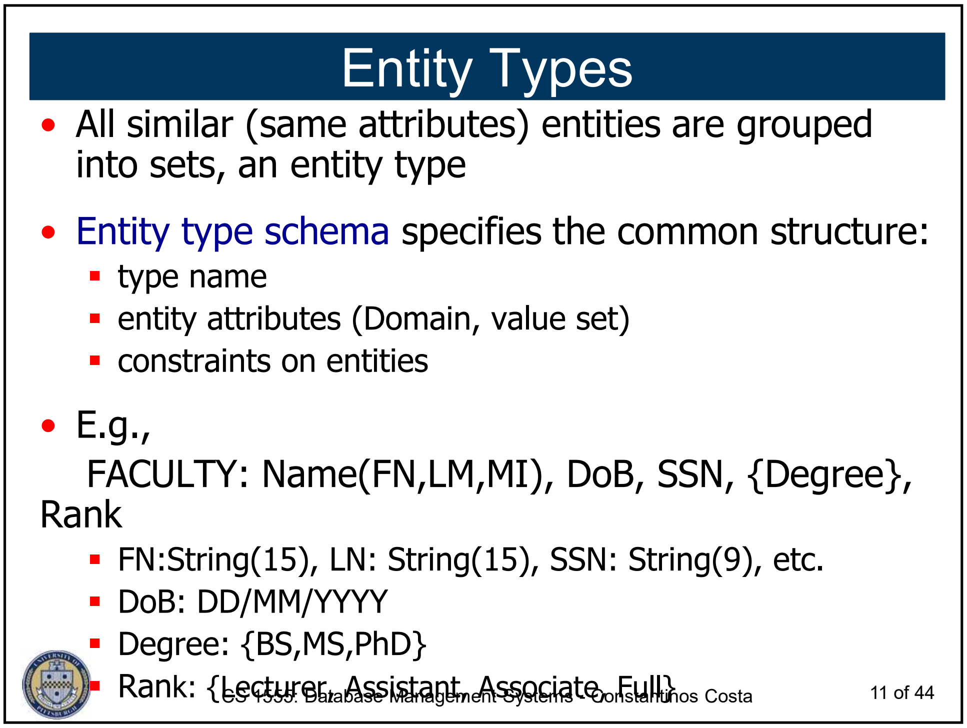
Attributes

- Entities are characterized by their attributes
  - Peter has an age,
  - Mary’s car has a color
- Relationships may also have attributes
  - Peter married Mary on Jan 7

Attribute Classification


Value
Structure
Single-value (e.g., Age)
Multi-value (e.g., {degrees})
Derived-value (e.g., Age from DoB)
NULL (Unknow, N/A)
Simple (atomic) (e.g., Age)
Composite, e.g., Date=(dd,mm,yyyy)

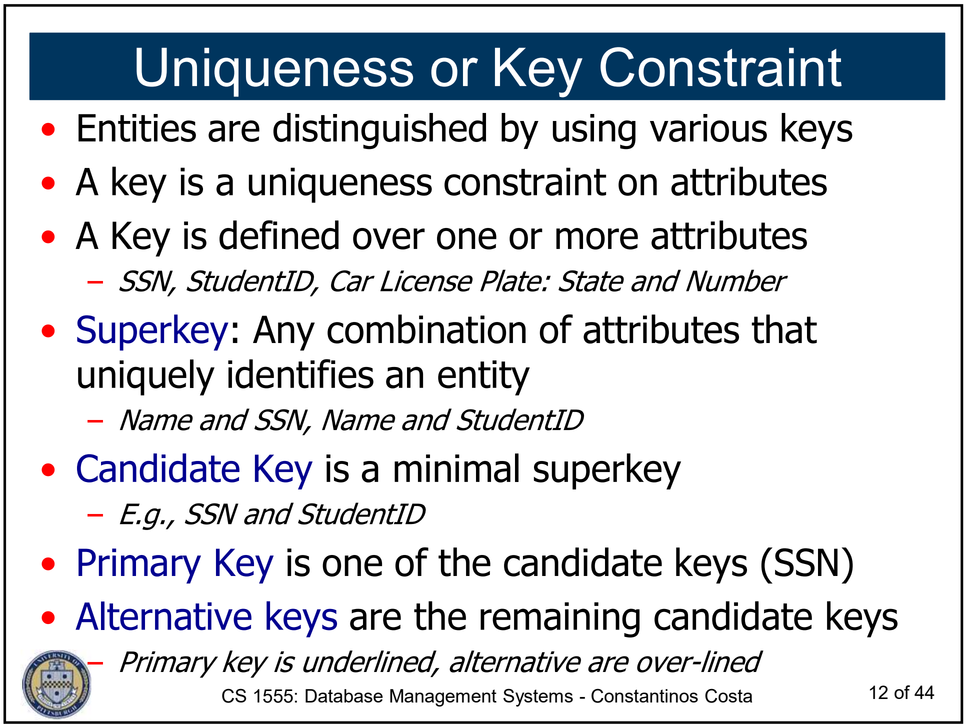
A composite attribute could be single value, e.g. ?



Entity Types

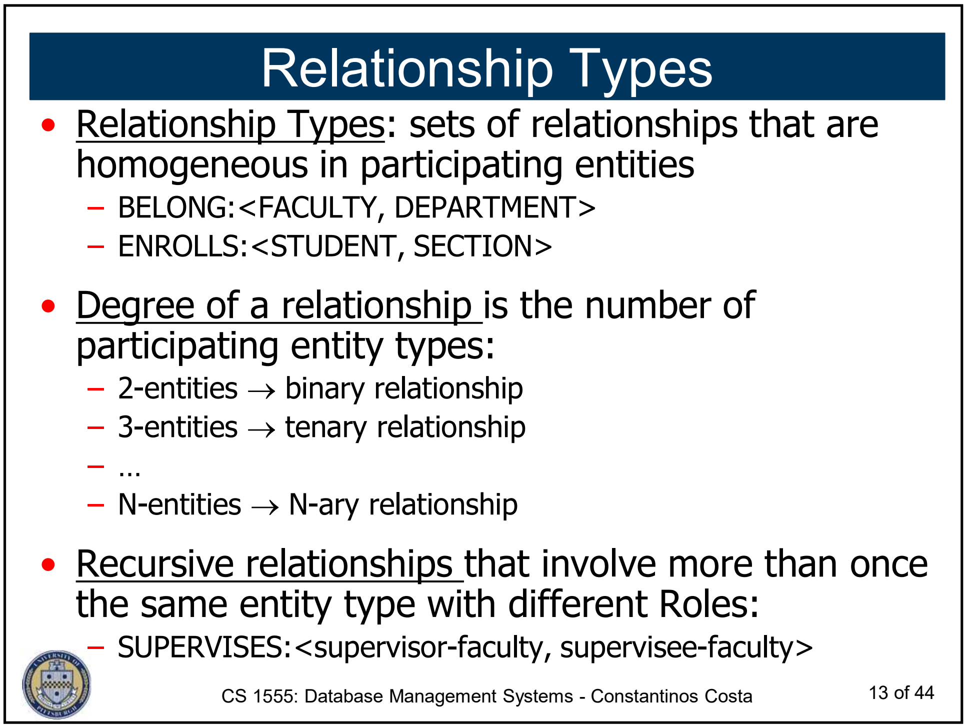
- All similar (same attributes) entities are grouped into sets, an entity type
- Entity type schema specifies the common structure:
  - type name
  - entity attributes (Domain, value set)
  - constraints on entities
- E.g.,

  FACULTY: Name(FN,LM,MI), DoB, SSN, {Degree}, Rank
  - FN:String(15), LN: String(15), SSN: String(9), etc.
  - DoB: DD/MM/YYYY
  - Degree: {BS,MS,PhD}
  - Rank: {Lecturer, Assistant, Associate, Full}



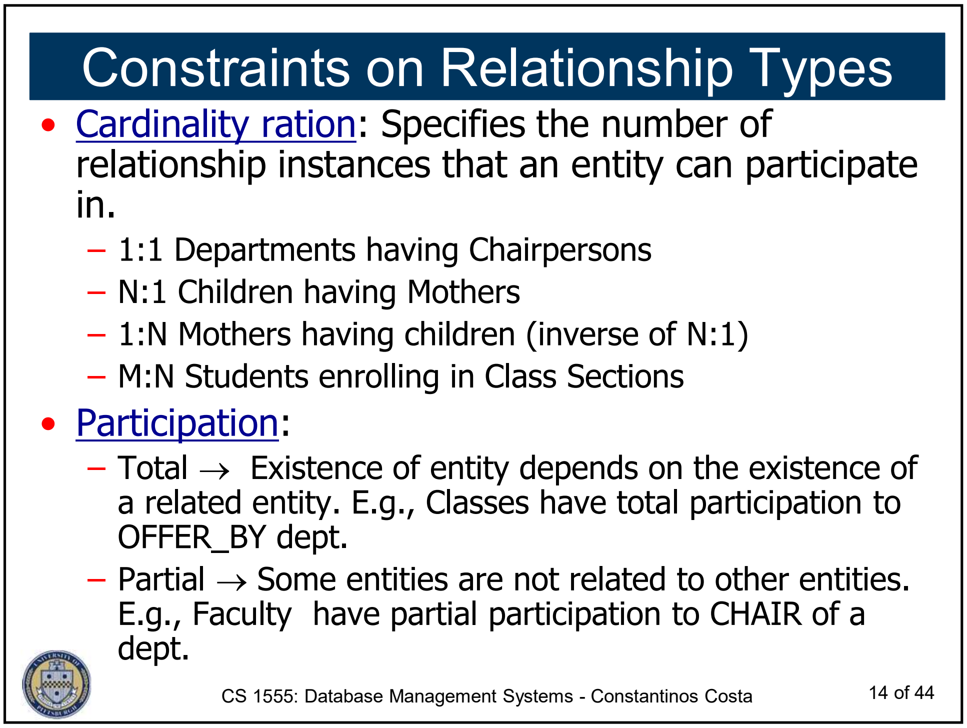
Uniqueness or Key Constraint

- Entities are distinguished by using various keys
- A key is a uniqueness constraint on attributes
- A Key is defined over one or more attributes
  - *SSN, StudentID, Car License Plate: State and Number*
- Superkey: Any combination of attributes that uniquely identifies an entity
  - *Name and SSN, Name and StudentID*
- Candidate Key is a minimal superkey
  - *E.g., SSN and StudentID*
- Primary Key is one of the candidate keys (SSN)
- Alternative keys are the remaining candidate keys
  - *Primary key is underlined, alternative are over-lined*



Relationship Types

- Relationship Types: sets of relationships that are homogeneous in participating entities
  - BELONG:<FACULTY, DEPARTMENT>
  - ENROLLS:<STUDENT, SECTION>
- Degree of a relationship is the number of participating entity types:
  - 2-entities → binary relationship
  - 3-entities → tenary relationship
  - …
  - N-entities → N-ary relationship
- Recursive relationships that involve more than once the same entity type with different Roles:
  - SUPERVISES:<supervisor-faculty, supervisee-faculty>



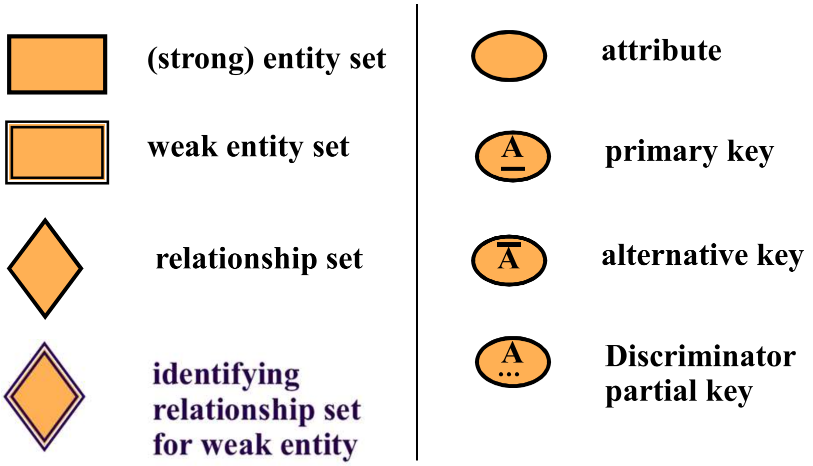
Constraints on Relationship Types

- Cardinality ration: Specifies the number of relationship instances that an entity can participate in.
  - 1:1 Departments having Chairpersons
  - N:1 Children having Mothers
  - 1:N Mothers having children (inverse of N:1)
  - M:N Students enrolling in Class Sections
- Participation:
  - Total → Existence of entity depends on the existence of a related entity. E.g., Classes have total participation to OFFER_BY dept.
  - Partial → Some entities are not related to other entities. E.g., Faculty have partial participation to CHAIR of a dept.

Strong and Weak Entities

- Strong or ordinary Entities:
  - Have independent existence in the mini-world
  - They are part of the care of the application
- Weak Entities:
  - They are dependent on another entity
  - Identify owner is the specific entity on which the weak entity depends
  - No key attribute; are distinguishable through an identifying relationship and a discriminator or partial key
  - Identifying relationship is always total participation
  - It may be represented as multi-value, composite attribute of owner (When isn't this possible?)

ER-Diagrams



(strong) entity set

weak entity set

relationship set

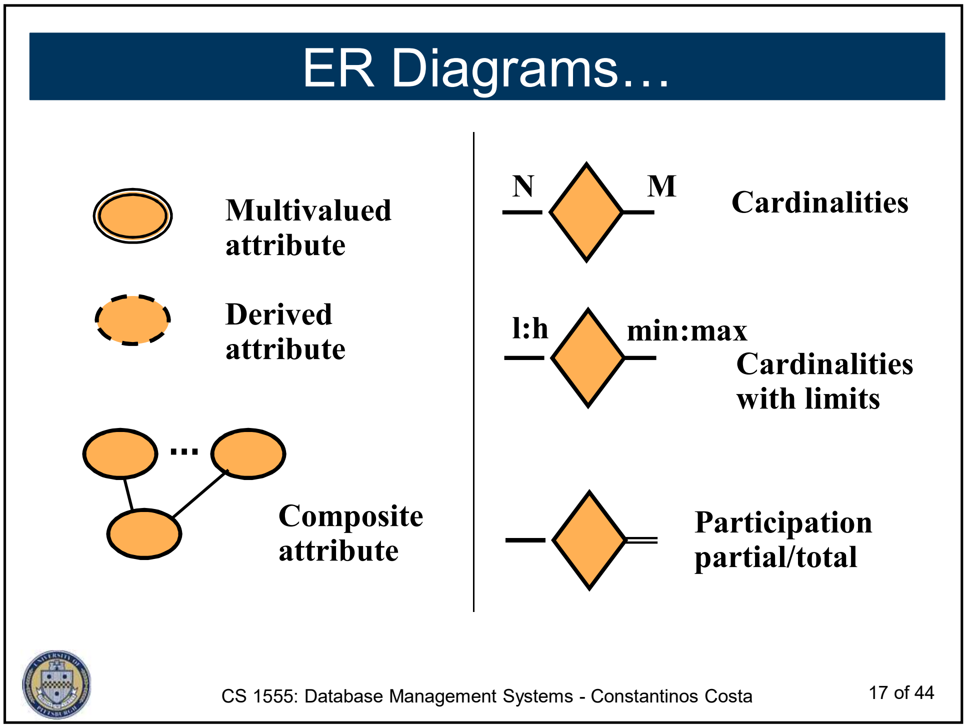
identifying relationship set for weak entity

attribute

primary key

alternative key

Discriminator partial key



ER Diagrams…

Multivalued attribute

Derived attribute

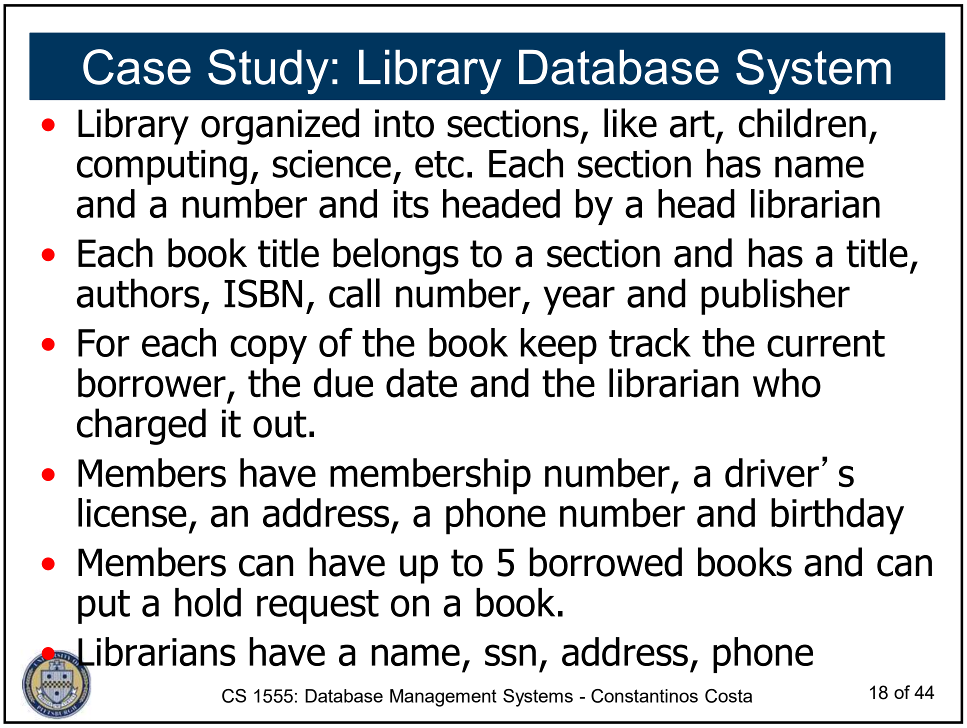
…

Composite attribute

N M Cardinalities

l:h min:max Cardinalities with limits

Participation partial/total



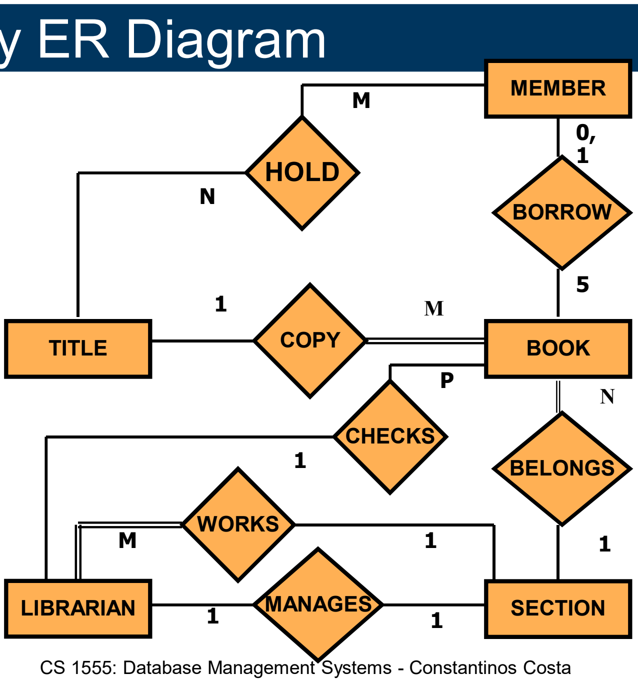
Case Study: Library Database System

- 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

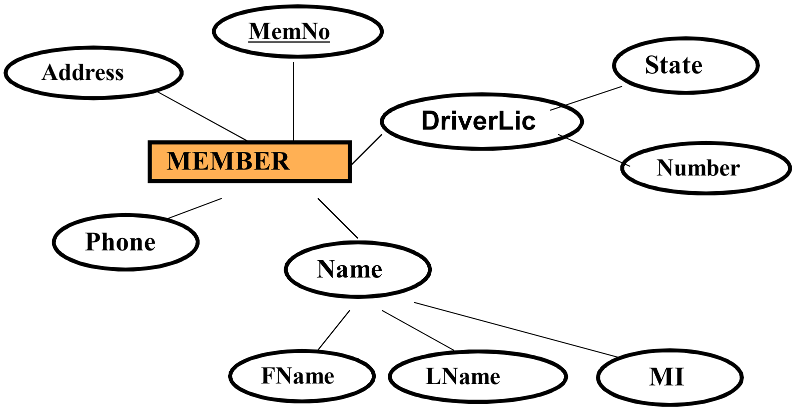
Observation

- nouns -> entity types/sets
  - E.g., Section, librarian
- verbs -> relationship types
  - E.g., belongs, headed

Library ER Diagram

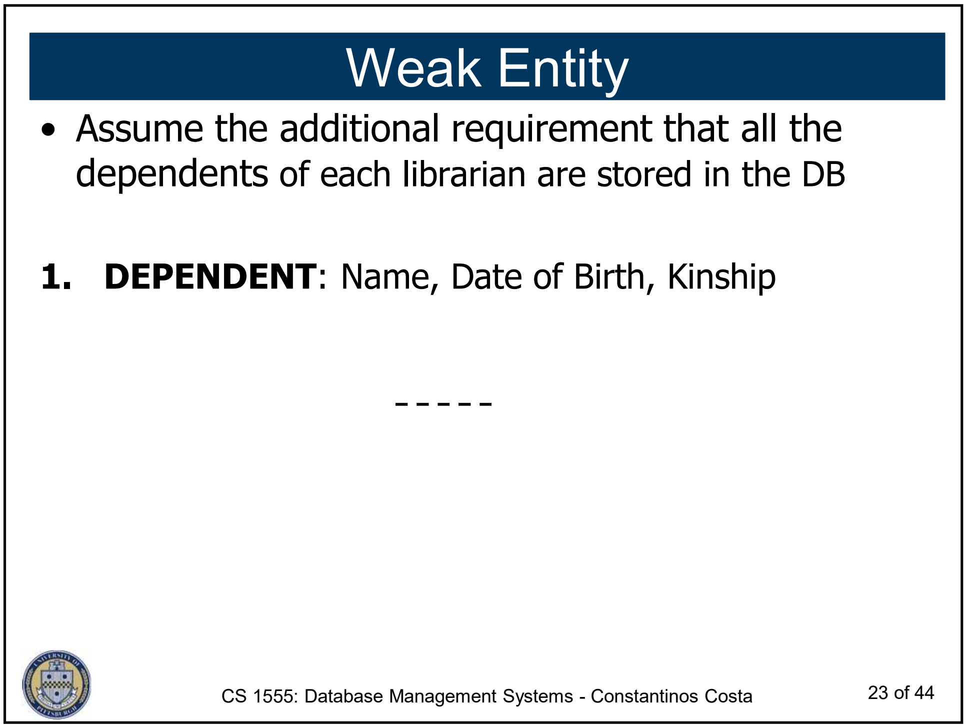


Library ER Diagram…



Entities

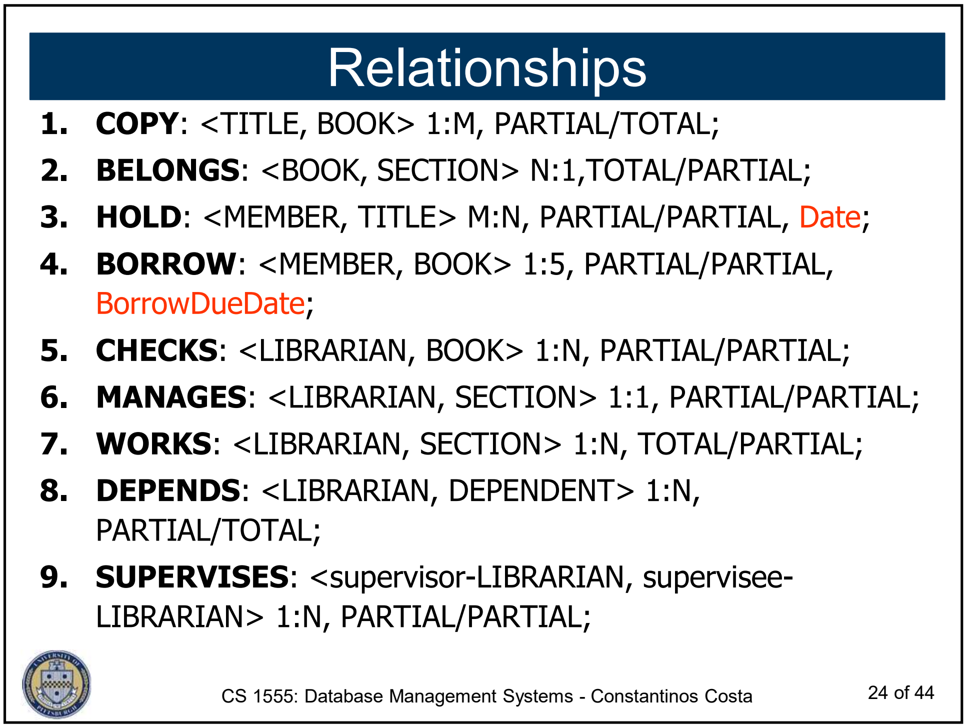
1. **TITLE**: CallNumber, Name, Author{(Name(Fname, MI, Lname),Order)}, ISBN, Year, Publisher;
2. **MEMBER**: MemNo, DriverLic(State,No), Name(Fname, MI, Lname), Address, PhoneNumber;
3. **BOOK**: BookID, Edition;
4. **LIBRARIAN**: SSN, Name, Address, Salary, Gender, Date of Birth;
5. **SECTION**: SectNo,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



Weak Entity

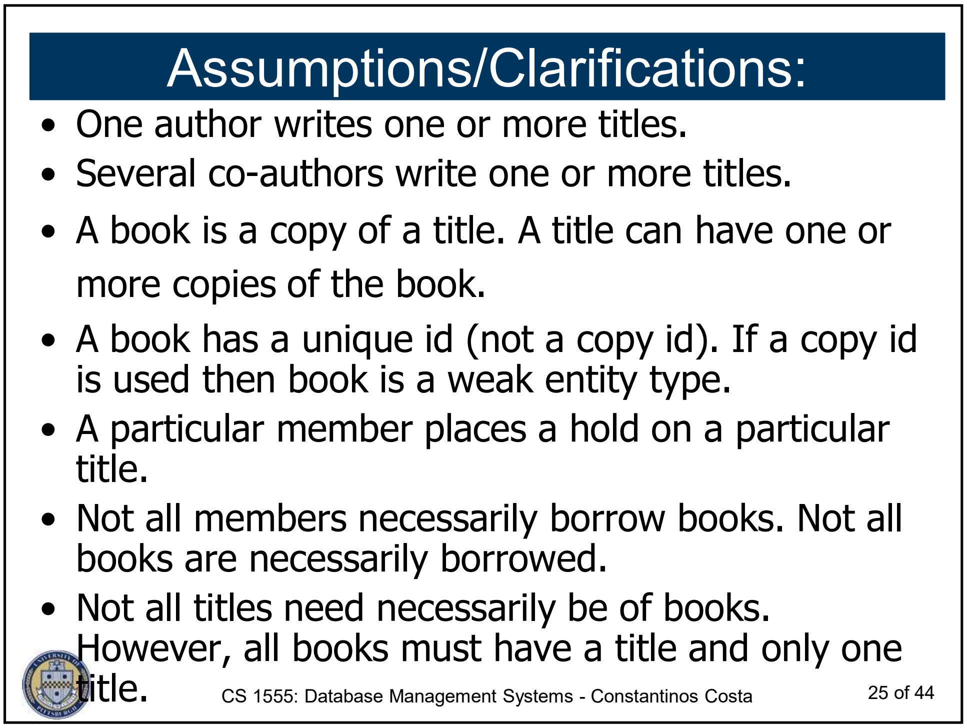
- Assume the additional requirement that all the dependents of each librarian are stored in the DB

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



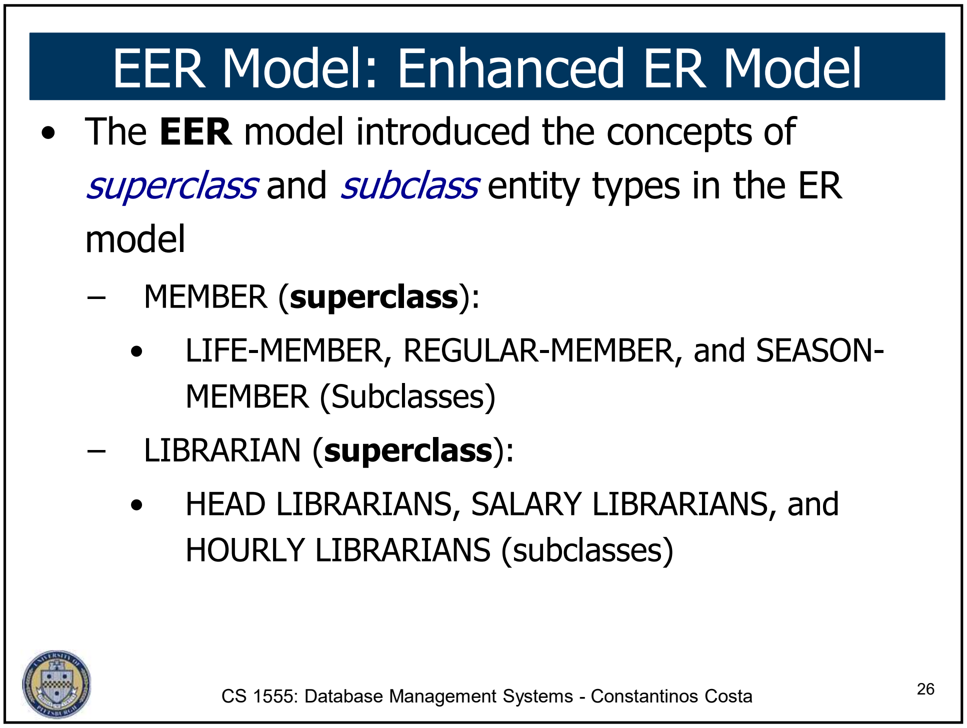
Relationships

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



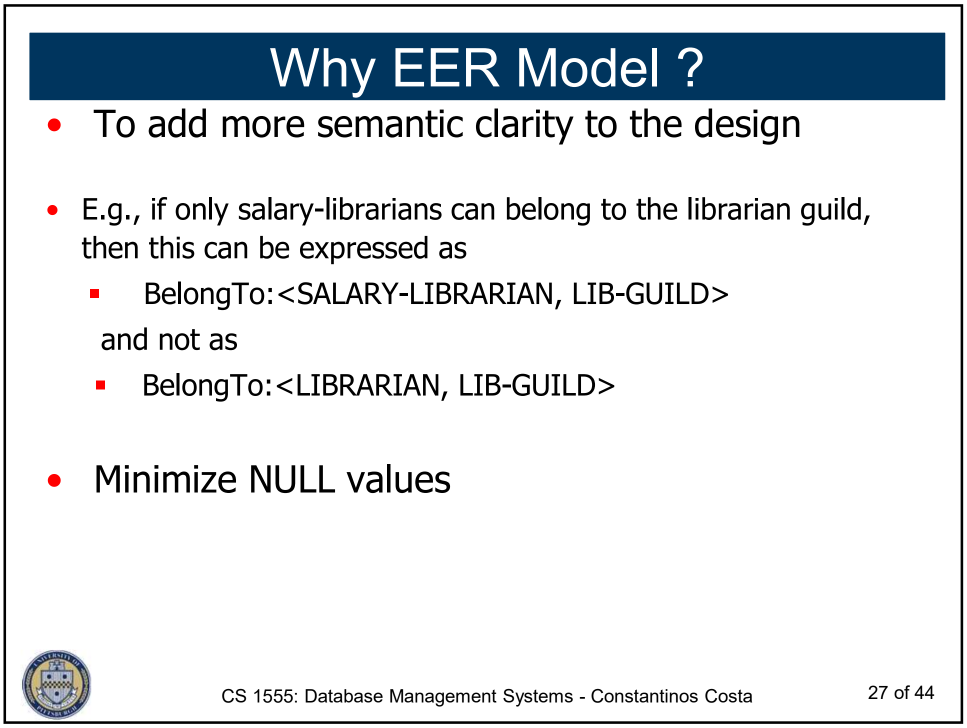
Assumptions/Clarifications:

- 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.



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)



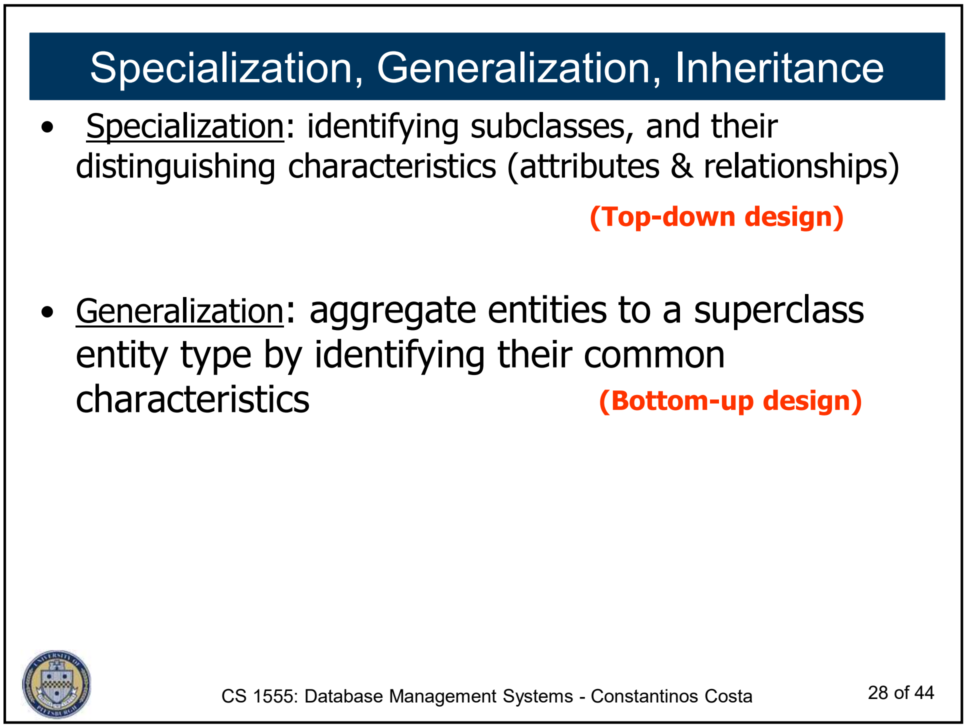
Why EER Model ?

- 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

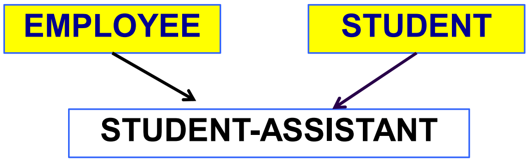
- Specialization: identifying subclasses, and their distinguishing characteristics (attributes & relationships)

  (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

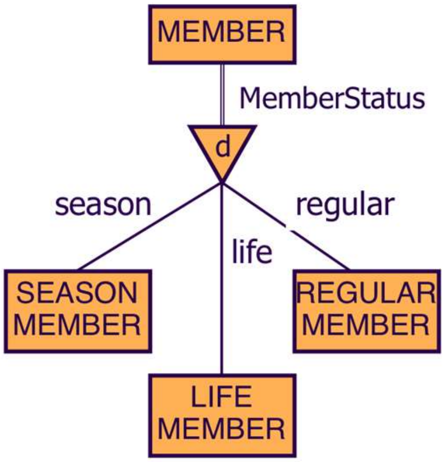
- The *disjoint* constraint: the subclasses of a superclass are disjoint.
  - This means that an entity can be a member of only one subclass.
  - The entities for each class can be *user-defined* or specified with a *predicate-defined subclass.*
  - In a predicate-defined subclass, we use a selection 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.

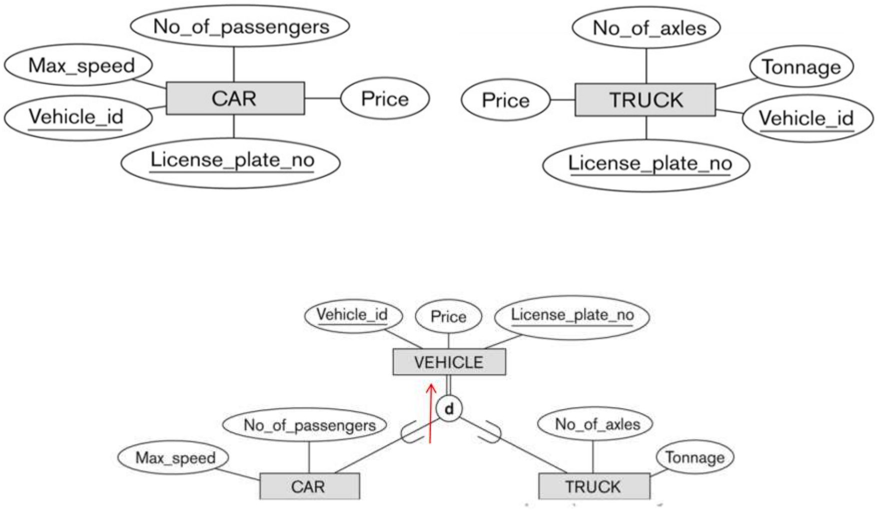
Completeness Constraints

- 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

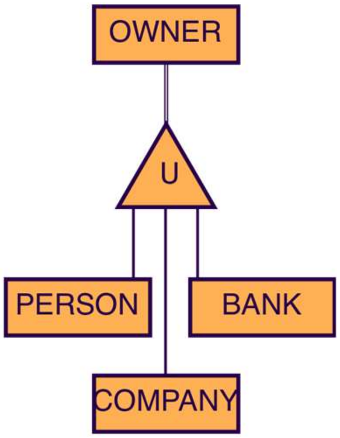


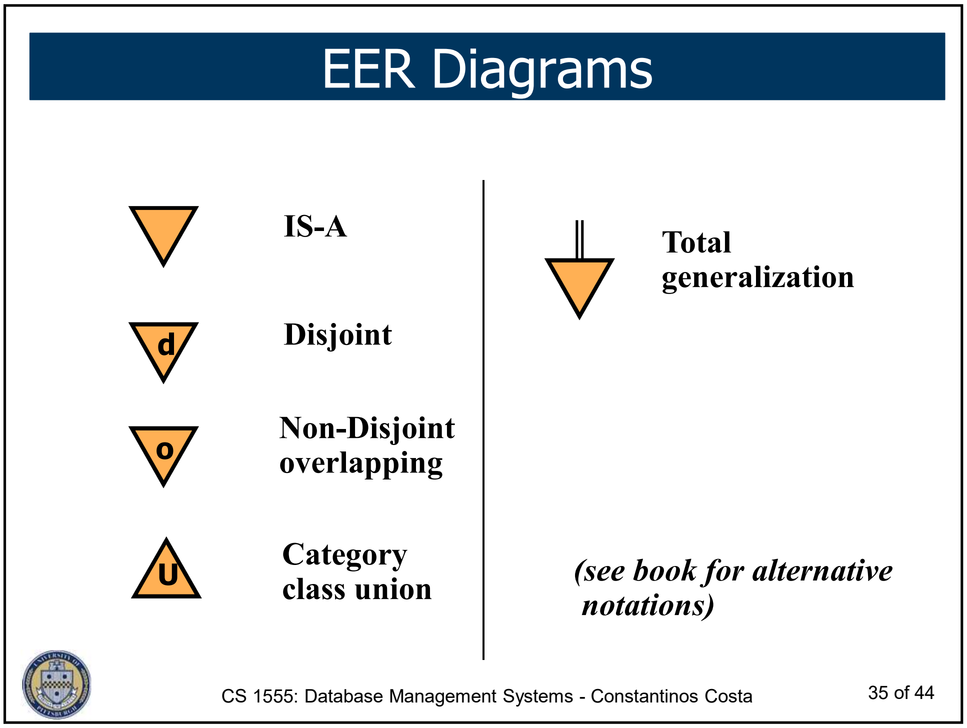
EER Diagram: Example 2



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 exists only in one of the superclasses
- Category can be:
  - **Total** or **Partial** (with predicate definition)





EER Diagrams

IS-A

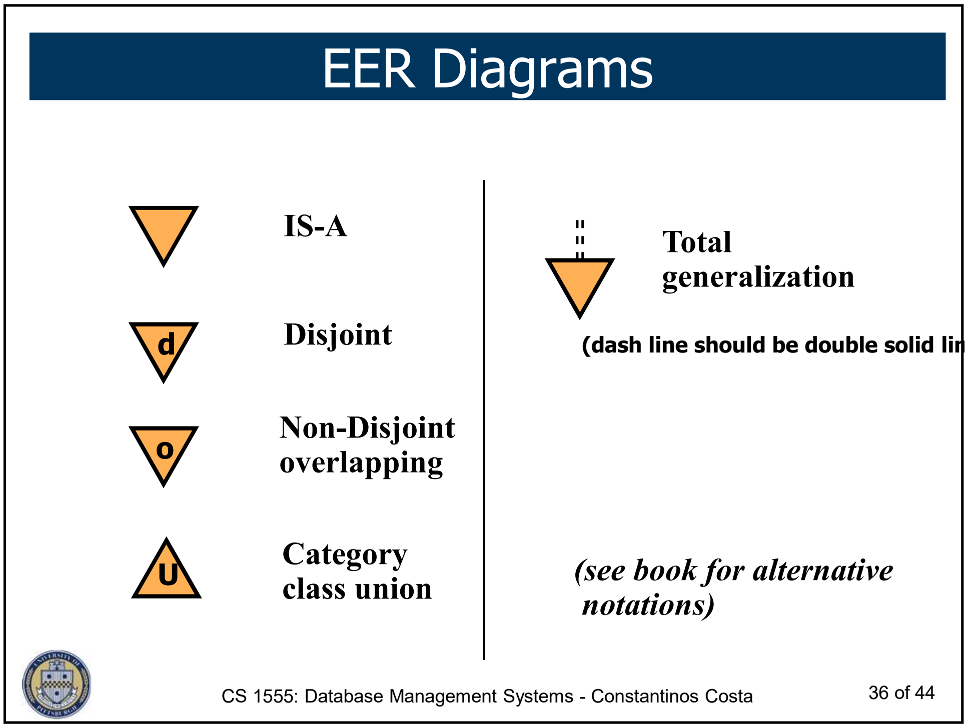
Disjoint

Non-Disjoint overlapping

Category class union

Total generalization

(see book for alternative notations)



EER Diagrams

IS-A

Disjoint

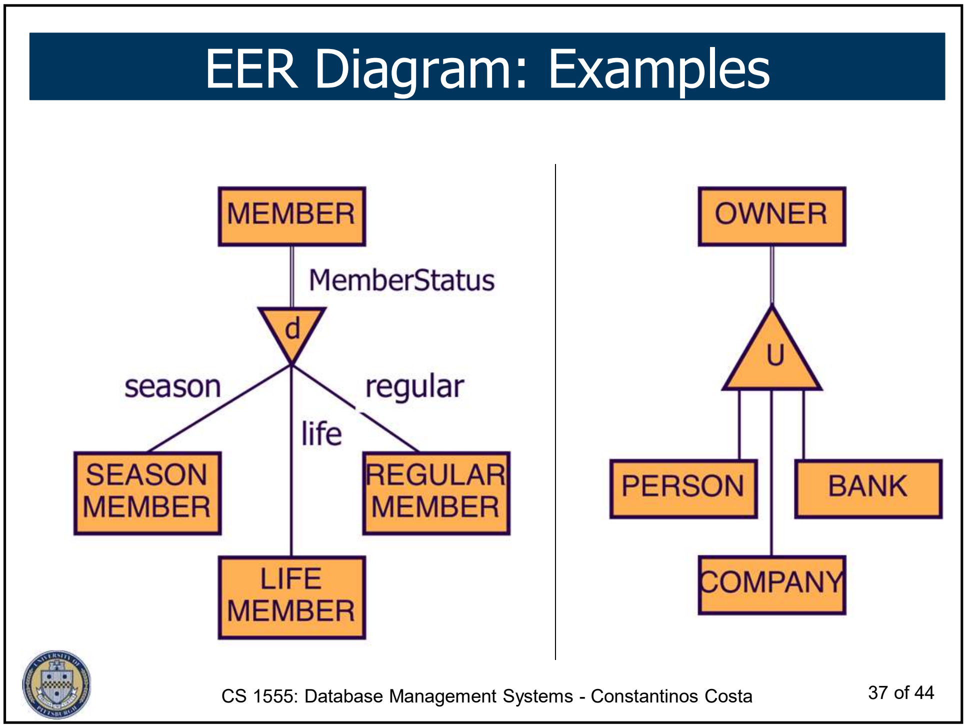
Non-Disjoint overlapping

Category class union

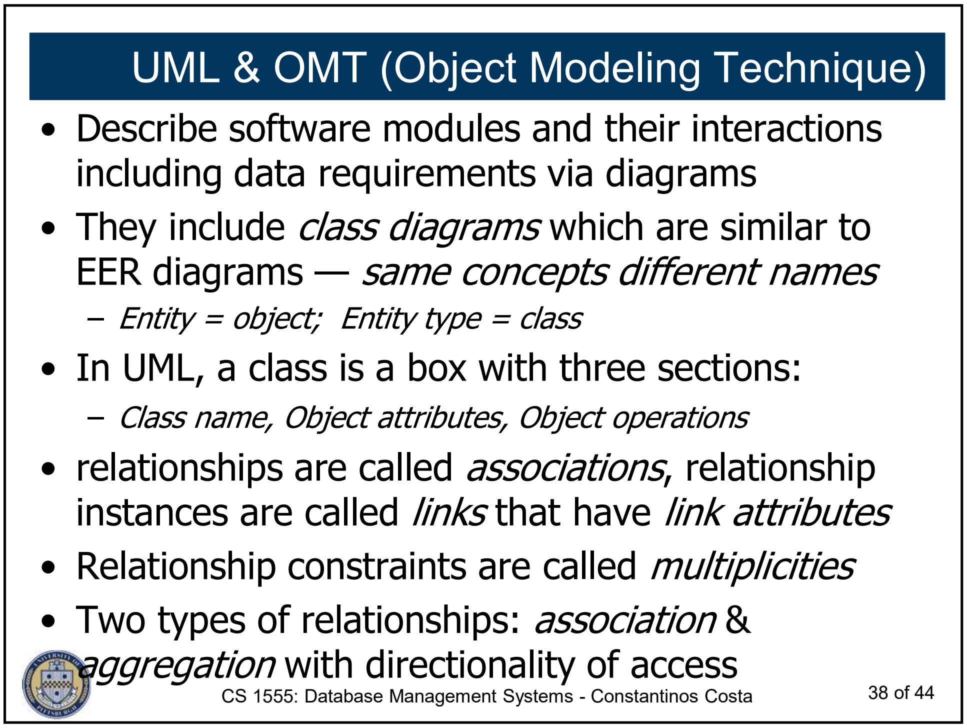
Total generalization

(dash line should be double solid lin

(see book for alternative notations)



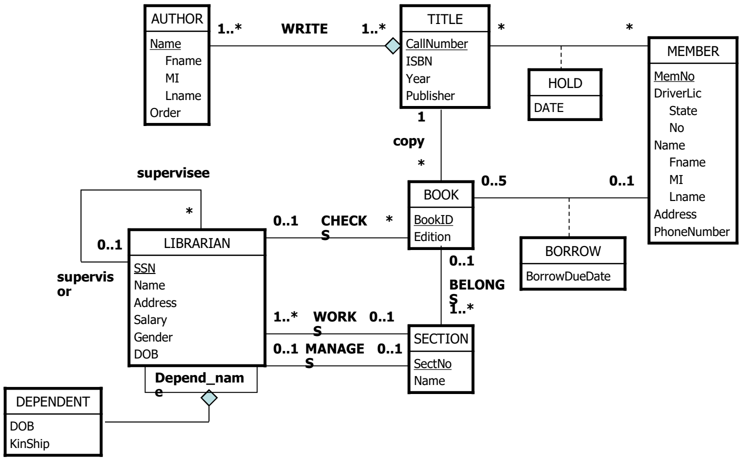
EER Diagram: Examples



UML & 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



AUTHOR
Name
Fname
MI
Lname
Order

1..* WRITE 1..*

TITLE
CallNumber
ISBN
Year
Publisher

* *

HOLD
DATE

MEMBER
MemNo
DriverLic
State
No
Name
Fname
MI
Lname
Address
PhoneNumber

1
copy
*

BOOK
BookID
Edition

0..5 0..1

BORROW
BorrowDueDate

supervisee
*
0..1
supervisor

LIBRARIAN
SSN
Name
Address
Salary
Gender
DOB

0..1 CHECKS *

0..1
BELONGS
1..*

1..* WORKS 0..1

0..1 MANAGES 0..1

SECTION
SectNo
Name

Depend_name

DEPENDENT
DOB
KinShip