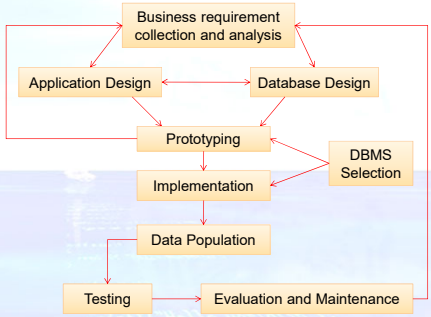
CpSc 4620/6620: Database Management Systems (DBMS) (TEXNH Approach)

ER Model and Database Design

James Wang

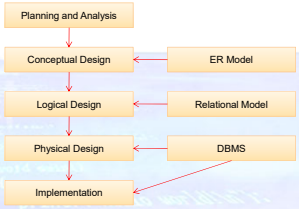
MeTube Project Lifecycle



2

Database Design

- **Goal: design the logical and physical structure of the database to accommodate the user needs for information of a particular application.**
- **Design steps:**



3

Data and Function Modeling

- **The conceptual design process includes data model and function model.**
  - The data model focuses on what data should be stored in the database (Database Schema Design).
  - The function model deals with how the data is processed (queries and operations on the data).
- **Data Modeling can be done by ER Model and Relational Model at different phase of the database design.**

4

Requirement Analysis

- **Data modeling is preceded by planning and analysis, in which the information needed to build a data model is collected.**
- **Although not formally considered as a part of the data modeling by some methodologies, the requirement analysis is always conducted concurrently with ER diagramming, and thus is viewed as an integrated part of data modeling by many practitioners.**

5

Goals of Requirement Analysis

- **Determine the data requirements of the database in terms of primitive objects.**
- **Classify and describe the information about these objects.**
- **Identify and classify the relationships among the objects**
- **Identify rules for the integrity of the data.**
- **Determine the types of transactions that will be executed on the database and the interactions between the data and the transactions (part of function modeling).**

6

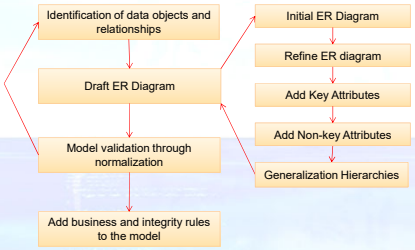
Requirement Collection

- **review of existing documents: Please read the project description.**
- **interviews with end users: Discuss with the instructor or your peers about the MeTube system.**
- **review of existing systems: Review the YouTube system.**
- **Note:**
  - End users don't know entities, attributes, and relationships; they know real-world terms.
  - Be sure understand the real-world needs.
  - Different end-users may think about and view data differently.

7

Construction of Data Model

- **ER model construction:**



8

Entity-Relationship Model

- **History:**
  - P.P. Chen. *The entity-relational model: Toward a unified view of data.* ACM Transactions an Database Systems, 1(1):9--36, 1976.
- **What is ER model?**
  - The ER model is a conceptual data model that views the real world as entities and relationships.
- **Why ER model?**
  - Easy to convert to relational model.
  - Easy to learn.

9

ER Model Basics

- **Entity: Data object or recognizable concept.**
  - Strong or weak (independent or dependent).
  - Instance: An occurrence of an entity.
  - Special types: associative or subtype.
- **Relationship: Association between entities.**
  - Degree (binary, ternary, …, n-ary).
  - Connectivity and cardinality: 1:1, 1:N, M:N.
  - Direction, Type, etc.
  - Existence: optional or mandatory.
- **Attributes: describe the entity of which they are associated.**
- **Generalization Hierarchies.**

10

ER Diagram Notation

- **Strong Entity:** Entity name
- **Weak Entity:** Entity name
- **Attribute:** Attribute name
- **Multi-value attribute:** Attribute name
- **Composite attribute :** Attribute name; Comp #1, Comp #2, Comp #3
- **Derived attribute:** Derived
- **Key attribute:** Key
- **Partial Key:** Partial Key

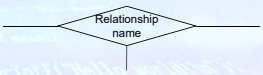
11

ER Diagram Notation (cont.)

- **Relationships:**

- **Identifying Relationships: Relationship connecting at least one weak entity.**

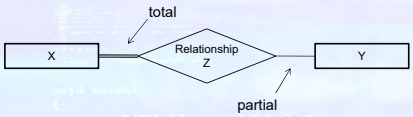
- **N-ary relationship: Relationship associated with more than two entities.**



12

ER Diagram Notation (cont.)

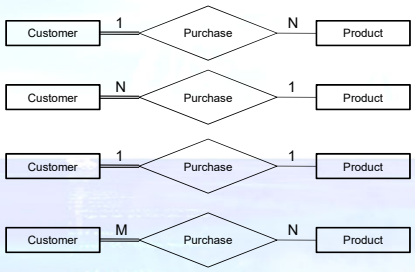
- **Constraints – Participation:**
  - **Total Participation:** Entity X has total participation in Relationship Z, meaning that every instance of X takes part in AT LEAST one relationship. (i.e. there are no members of X that do not participate in the relationship.
  - **Partial Participation**: Entity Y has partial participation in Relationship Z, meaning that only some instances of Y take part in the relationship.
  - **Example:** X = Customer, Y = Product, Z = Purchase.



13

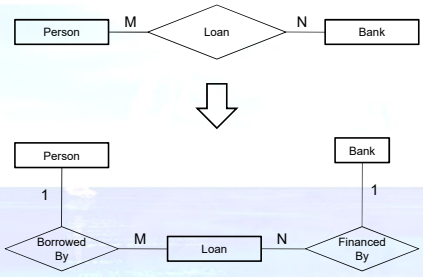
ER Diagram Notation (cont.)

- **Constraints – Cardinality:**



14

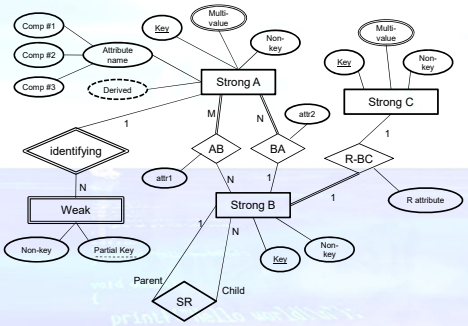
M:N or 1:N?



Question: Is M:N relationship necessary?

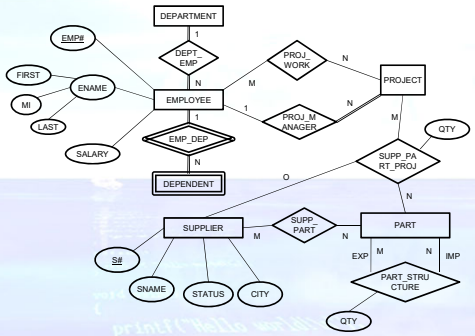
15

ER Diagram Example



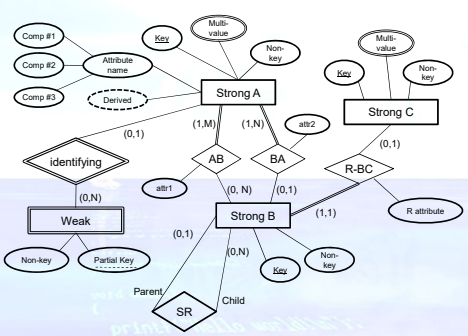
16

ER Diagram Example 2



17

ER Diagram with (min, max) Cardinality



18

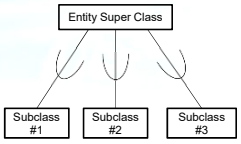
Extended ER Model

- **Extended ER model extends the basic ER model by adding following features:**
  - An entity definition is known as a class.
    - A specific occurrence of an entity is an instance of a class.
  - Classes can be formed into superclass/subclass hierarchies using generalization and specialization.
    - The IS-A relationship.
    - Inheritance of attributes.
    - Constraints on subclass membership.
  - Categories are used to represent a union of classes.

19

Extended ER Diagram

- **Specialization/Generalization:**



- **Each subclass inherits all relationships and attributes from the super-class.**

20

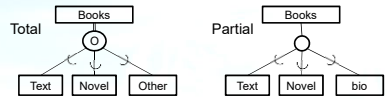
EER Diagram Notation

- **Constraints on Specialization/Generalization:**
  - **Total Specialization** – Every member of the super-class must belong to at least one subclass.
  - **Partial Specialization** – a member of the super-class may not belong to one of the subclasses.
  - **Disjoint** – every member of the super-class can belong to at most one of the subclasses.
  - **Overlapping** – a member of the super-class can belong to more than one of the subclasses.
  - **Multiple Inheritance** – a subclass participates in more than one subclass/super-class relationship, and inherits attributes and relationships from more than one super-class.
  - **Union** – a subclass/super-class relationship can have more than one super-class, but the subclass inherits from at most one of the super-classes.

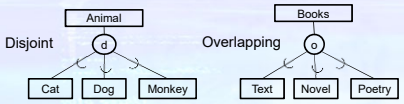
21

ER Diagram Notation (cont.)

- **Total or partial specialization:**



- **Disjoint or overlapping specialization:**



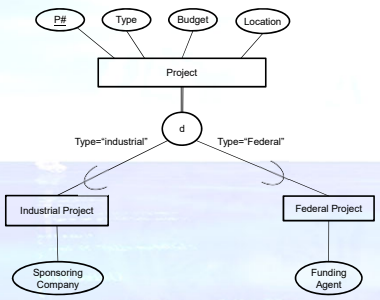
22

Constraints on Subclass Membership

- **The attributes in subclass may be dependent on an attribute value in the superclass:**
  - **Attribute-defined** - Determines membership in a subclass by placing a condition on the value of an attribute in the superclass.
  - **User-defined** - Membership in a subclass does not depend on any specific attribute value. Membership is determined by the user.

23

Attribute-defined Subclass



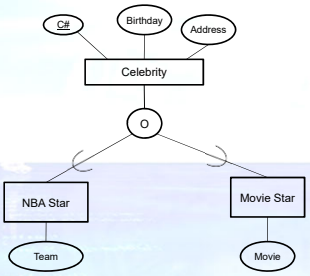
24

Rules for Attribute-defined Subclass

- If the specialization attribute at the superclass level is single-valued, membership at the subclass level is always disjoint.
- If the specialization attribute at the superclass level is multi-valued, membership at the subclass level is always overlapping.
- If the specialization is total, the attribute value in the superclass is required.
- If the specialization is partial, the specialization attribute value in the superclass is optional. The presence of a value, however, implies automatic insertion at the subclass level.

25

User-defined Subclass



26

Rules for Superclass/Subclass Hierarchy

- Deleting an entity from a superclass implies automatic deletion of the entity from all subclasses.
- Deleting an entity from a subclass does not imply deleting the entity from its superclass. However, attributed-defined constraints must not be violated.
- At the superclass level, changing the value of an attribute used for attribute-defined specialization requires appropriate changes in subclass membership.

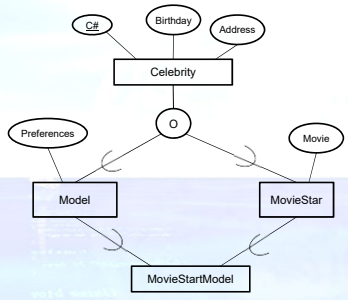
27

Multiple Inheritance

- A subclass can have more than one superclass. It is called specialization lattice.
  - The subclass is referred to as a shared subclass.
  - A specialization lattice demonstrates multiple inheritance.
  - A shared subclass must satisfy the multiple inheritance intersection constraint, where each instance of the shared subclass is an instance of all of its superclasses

28

Specification Lattice

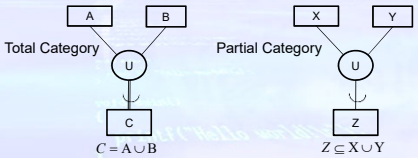


MovieStarModel = Model ^ MovieStar

29

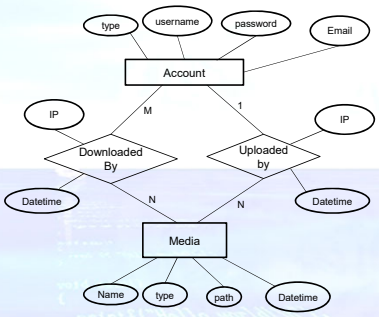
Categories and Categorization

- If a subclass contains instances from two superclass, then this subclass is called a category.
- A category represents a union of its superclasses, where an instance of a category subclass must be an instance of at least one superclass, but is not necessarily a member of all superclasses.



30

MeTube Database ER Model (partial, unrefined)



31

References

- http://www.youtube.com
- An Introduction to Database Systems, Eighth Edition, C. J. Date, Addison Wesley, 2004, ISBN: 0-321-19784-4.
- Suzanne W. Dietrich and Susan D. Urban, *An Advanced Course in Database Systems: Beyond Relational Databases*, Prentice Hall, 2005. ISBN-13: 978-0130428981; ISBN-10: 0130428981

32