LAB 2 Notes

Introduction to Database Design – ER modeling
1) Steps of Designing a Database System
2) Entities, Attributes, Entity Sets, Relationships, Relationship Sets
3) Entity Relationship Diagram – concepts, guidelines for effective designs
4) Additional Feature of ER Model
5) Putting it all together exercises

1) Database Design steps

- **Requirement Analysis**
  - What **data** to be stored in Database? **Analysts + customer**
  - What **applications** to be build on top of it
  - Which operations are subject to **performance** requirements
- **Conceptual Database Design**
  - High level description of **Analysis + constrains** over these data are modeled with the ER model (semantic model used in database design.
  - Goal: Generate description (or **model**) of data that is understandable by both developers and users
  - The design must be precise in order to allow straightforward **translation** into the **relational model** (tables, attributes,..), which is used by the Database
- **Logical Database Design** (**ER schema -> relational database schema**)
  - Generally this step involves the conversion of the conceptual schema → Database schema.
  - Since we consider only the relational model
- **Schema Refinement**
  - Analyze Relational Database Schema and **identify problems**.
  - **Normalize relations** – decomposing into smaller relations
- **Physical Database Design**
  - Make sure that the database meets the **performance** needs / workloads that are expected bye the Analysis. (Indexes, Denormalize Relations)
- **Application & Security Design**

Requirement Analysis	**Conceptual Design ER**	**Logical DB Design (relational)**	**Schema Refinement**	**Physical DD**

2) Entities, Attributes, Entity Sets, Relationships, Relationship Sets

Entity (Relation)

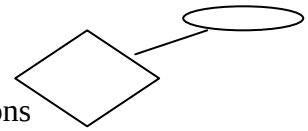
- An entity is a **real world object** that can be **distinguished** from another object given some **attributes**
- e.g. Employee, Manager are not different but Employee, Projects are different

Attribute

- Several attribute characterize an entity. If an attribute is multi-value (address zip, address, aptno) we create an entity.
- Domain -> Possible values
- Key -> Set of attributes that uniquely identifies an entity. (primary, secondary, candidate)

Relationship

- Relates 2 or more entities.
- Descriptive Attributes
- Ternary Relationship -> Involves 3 relations

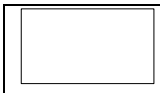
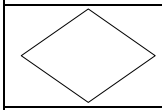
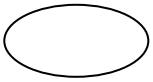


3) Entity-Relationship ER Diagram

- The ER model allows us to **describe the data** involved in a real world enterprise **in terms of objects (entities) and their relationships**.
- Provides the initial framework for developing an initial DB design.
- There are other variations of the ER model that exist, mainly different on the way entities and their relationships are graphically represented, however you should follow book notation.

ER → Data Structure Diagram

ER Notation Explanation

	Entity
	Relationship
	ERD relationship connector
	At-least-One ERD relationship connector
	At-most-One ERD relationship connector
	At-least-One and At-most-One ERD relationship connector
Bold lines (elsewhere)	Weak Entities, Weak Relationships appropriately
Notice:	Please note that the cardinality symbols (1:1, 1:M, M:N) are redundant since bold lines and arrows are also used (as in book), but are used for convenience.
	Ellipsoid - > attribute , key or not

4) Additional Feature of ER Model

- **Key Constrains**

  The "**At least – At most" question**

  1:1 : Each professor works in at most 1 department . In each department at most 1 professor work.

  1:N : Each professor works in at most N departments . In each department at most 1 professor work.

  N:M : Each professor works in at most N departments . In each department at most M professor work.

- **Participation Constrains (partial VS total)**
  1. Employee works in at least 1, at most N Departments (TOTAL participation in relationship)
  2. Employee works in at least 0, at most N Departments (PARTIAL participation in relationship)
- **Weak Entities**
  1. A weak entity can be identified uniquely ONLY by considering the primary of another relation.
  2. Must form a total participation, one-to-many relationship with its owner
- **Class Hierarchies & Aggregation**
  1. **Not used in most designing tools,**
  2. **Might be covered by your teacher.**

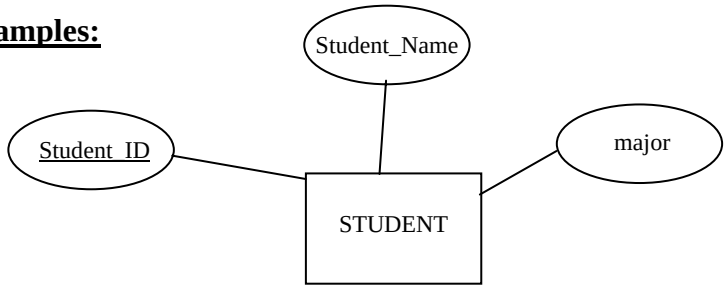
5) Putting it all together

Entity VS Attribute? (e.g. address)
Depends on application.
Rule of Thumb: if multi-value attribute split into Entity

Entity Vs Relationship
Rule of Thumb:
If a relationship is M:N think of making it an entity

- ➔ Easier for your design.
- ➔ During conversion to relational model you will anyway do it

Examples:

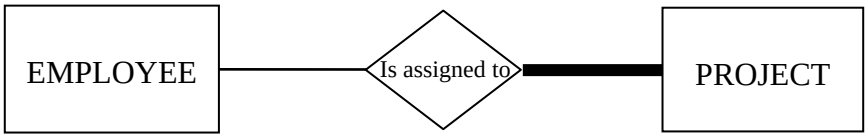


STUDENTS

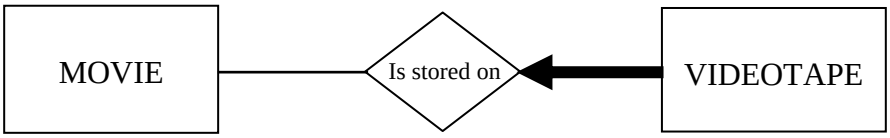
Student_ID	Student_Name	major

A Patient must have a Patient History
A Patient History must belong to one and only one Patient.

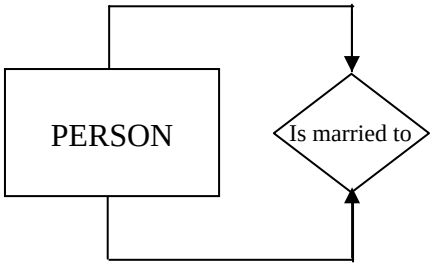
An Employee can be assigned to a project.
A Project must have at least one Employee assigned to it



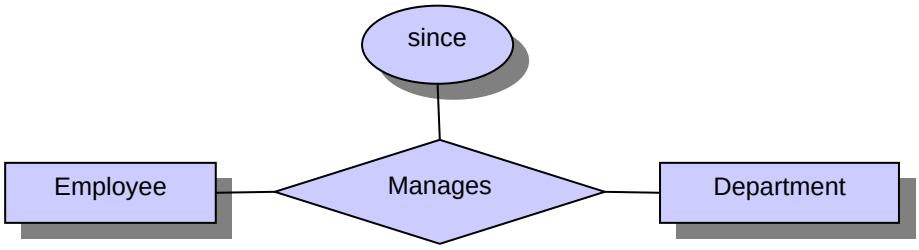
A movie can be stored on several videotapes



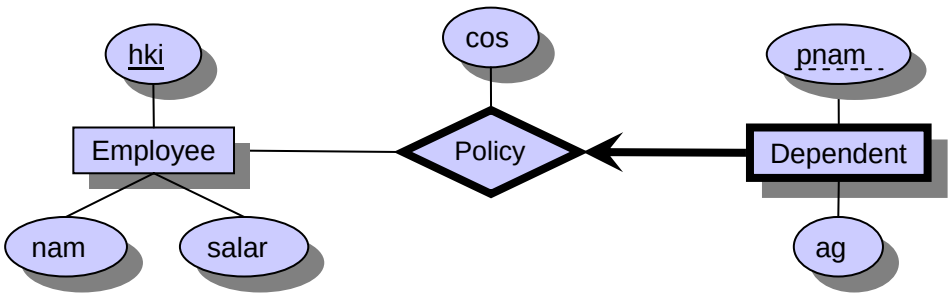
A Person is married to a Person (Unary Relationship)



An Employee manages a Department



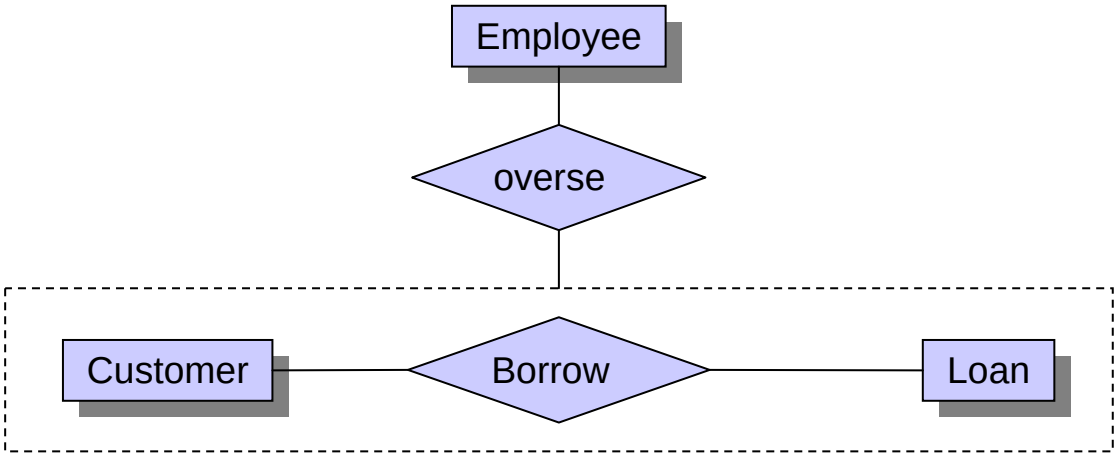
Weak Entity



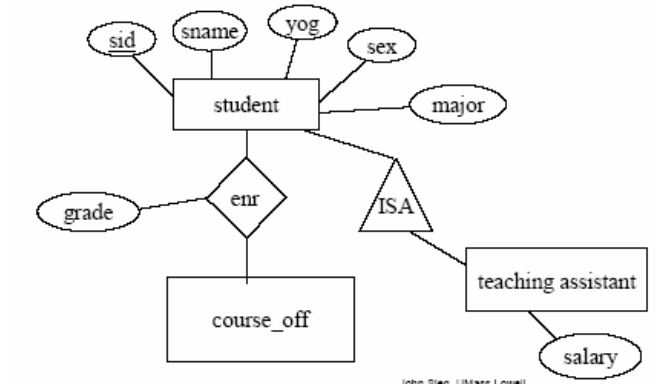
Aggregation

Relationship between an entity and a relationship

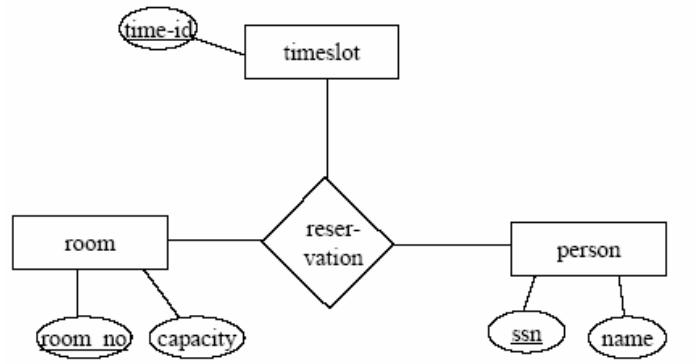
- A Customer borrows a loan
  - Relationship between Customer and Loans
- The bank may assign an Employee, the Loan-officer, to oversee the transaction
  - Relationship between entity Employee and relationship Borrows



Class Hierarchy



Ternary



Complete Example:

Design and ER diagram for this application,

Consider the following set of requirements for a university database that is used to keep track of student's transcripts:

a. The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birth-date, sex, class (freshman, sophomore…graduate), major department, minor department (if any), and degree program (B.A., B.S.,…, Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address and to the student's last name. Both social security number and student number have unique values for each student.
b. Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.
c. Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.
d. Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, …, up to the number of sections taught during each semester.
e. A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3 or 4).

SOLUTION GIVEN DURING LAB OR OFFICE HOURS