LAB 2 Notes

For students that were not present in the first lab

- TA Web page updated : http://www.cs.ucr.edu/~cs166/
- Mailing list Signup: http://www.cs.ucr.edu/mailman/listinfo/cs166
- The general idea of ER which will be described today was outlined in previous
- Please bring book from now on

Ch.1: Overview of Database Systems
Ch.2: Introduction to Database Design
Ch.3: The Relational Model
Ch.4: Relational Algebra
Ch.5: SQL
Ch.8: Storage and Indexing

Ch.9: Storing Data: Disks and Files
Ch.10: Tree-Structured Indexing
Ch.11: Hash-Based Indexing
Ch.12: Overview of Query Evaluation
Ch.13: External Sorting
Ch.14: Evaluation of Relational Operators
Ch.15: A Typical Relational Query Optimizer
Ch.16: Overview of Transaction

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

<u>1) Database Design steps</u>

- **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 of data that is understandable by both developers and users
  - The design must be precise in order to allow straightforward translation into the **<u>relational model</u>** (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.
  - Ex: Student(ssn, name, numgrade, lettergrade) Functionally-dependent
  - Normalizing relations - Theory
- **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

PUT SYMBOLS ON EACH OF THE BELOW

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

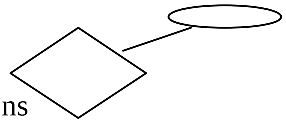
- Several attribute characterize an entity. If an attribute is multi-value (address zip, address, aptno) create an entity.
- Domain -> Possible values
- Key -> Set of attributes that uniquely identifies an entity. (primary, secondary, candidate)

Entity-Set

- Toy, Appliance Department Employees under same set

Relationship

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



Relationship Set

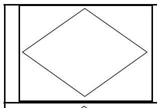
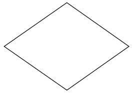
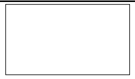
- A set of similar relationships

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 exist, mainly different on the way entities and their relationships are graphically represented.
- You should follow book notation.

ER → Data Structure Diagram

ER Notation Explanation

	M:N Relationship (Entity-Relationship)
	1:1, 1:M or M:1 Relationship
	Entity
—	ERD relationship connector
━	At-least-One ERD relationship connector
→	At-most-One ERD relationship connector

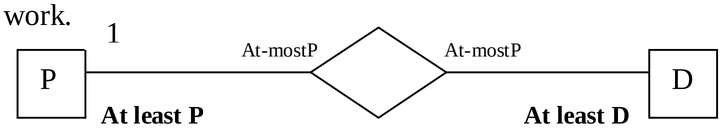
➡	At-least-One and At-most-One ERD relationship connector
Bold lines (elsewhere)	Weak Entities, Weak Relationships appropriately
<u>Notice:</u>	Please note that the cardinality symbols (1:1, 1:M, M:N) are redundant since bold lines and arrows are also used (as in book), but are used for convenience.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

4) Additional Feature of ER Model

- **Key Constrains**

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

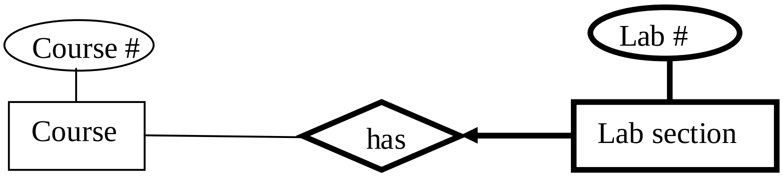
  1:1 : Each professor works in at most 1 department . In each department at most 1 professor work.



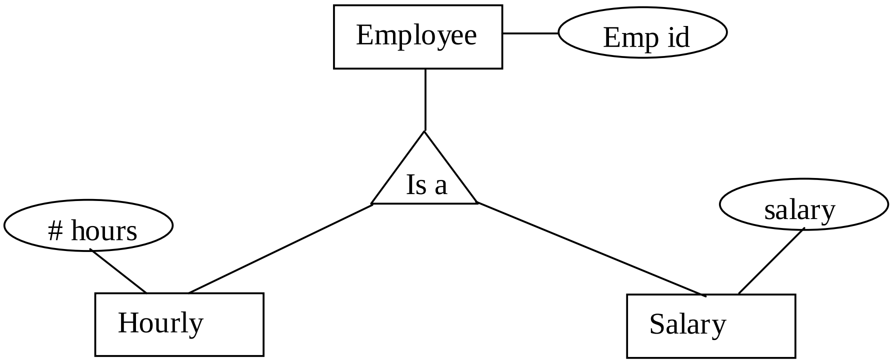
  1:N : Each professor works in at most N departments . In each department at most 1 professor work.

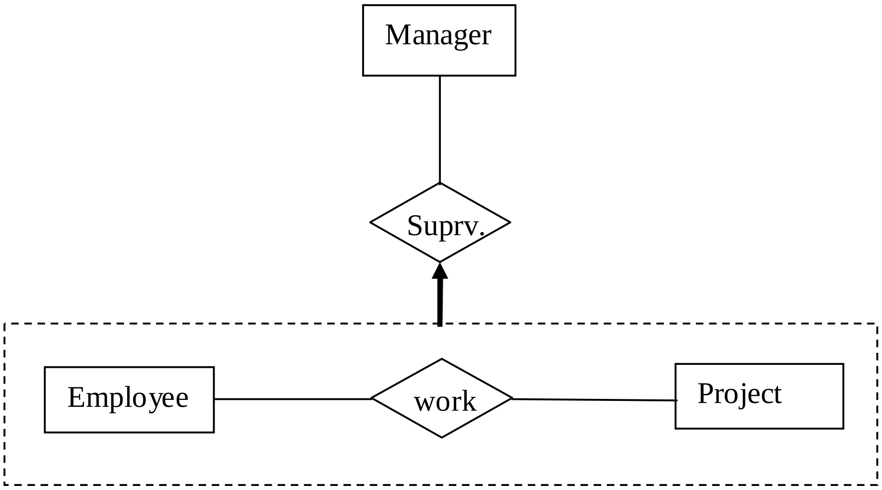
  N:M : Each professor works in at most N departments . In each department at most M professor work.

- **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



- **Class Hierarchies & Aggregation**





5) Putting it all togther

Exercise 2.2 A university database contains information about professors (identified by social security number, or SSN) and courses (identi.ed by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming that no further constraints hold).

1. Professors can teach the same course in several semesters, and each offering must be recorded.
2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)
3. Every professor must teach some course.
4. Every professor teaches exactly one course (no more, no less).

5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.

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
- ➔ Rename by concating : Student, Courses => StudentCourses