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
- Get a list of Students
- The general idea of ER which will be described today was outlined in previous
- Please bring book from now on

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 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.
  - Ex: Student(ssn, name, numgrade, lettergrade) Functionally-dependent
  - Normalizing relations - Theory
• **Physical Database Design**
  o Make sure that the database meets the performance needs / workloads that are expected by the Analysis. (Indexes, Denormalize Relations)

• **Application & Security Design**

<table>
<thead>
<tr>
<th>Requirement Analysis</th>
<th>Conceptual Design ER</th>
<th>Logical DB Design (relational)</th>
<th>Schema Refinement</th>
<th>Physical DD</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M:N</td>
<td>Relationship (Entity-Relationship)</td>
</tr>
<tr>
<td>1:1</td>
<td>1:M or M:1 Relationship</td>
</tr>
<tr>
<td>Entity</td>
<td></td>
</tr>
<tr>
<td>ERD relationship connector</td>
<td></td>
</tr>
<tr>
<td>At-least-One ERD relationship connector</td>
<td></td>
</tr>
<tr>
<td>At-most-One ERD relationship connector</td>
<td></td>
</tr>
<tr>
<td>At-least-One and At-most-One ERD relationship connector</td>
<td></td>
</tr>
<tr>
<td>Bold lines (elsewhere)</td>
<td>Weak Entities, Weak Relationships appropriately</td>
</tr>
<tr>
<td>Notice:</td>
<td>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.</td>
</tr>
<tr>
<td>Ellipsoid - &gt; attribute , key or not</td>
<td></td>
</tr>
</tbody>
</table>

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. Employee(ssn) -- \( \exists \text{Has} \leq \text{Dependents} \) (pname, age)

- **Class Hierarchies & Aggregation**
  1. Not used in most designing tools,
  2. Might be covered by your teacher.

### 5) Putting it all together

**Exercise 1** A university database contains information about professors (identified by social security number, or SSN) and courses (identified 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.

1) 

![Diagram 1]

2) 

![Diagram 2]
3) 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

Exercise 2 E-R diagram for a car-insurance company with the following information:

- people own one or more cars each.
- A car can does not have multiple owners
- People with attributes driver-id, name, address
- Car with attributes license, model, year
- Each car has associated with it zero to any number of recorded accidents.
- Accident with attributes report-number, location, date
- The car owner might not be the driver of the car involved in the accident
• For each accident record the *damage-amount*

**Step one - Define Entity set**
• People - driver-id, name, address  
  • Car - license, model, year  
  • Accident - report-number, location, date

**Step two - Relationship sets**
  • owns - a person can own 0 or more cars. A car has only one owner.  
  • participates - a driver and a car participate in an accident. *damage-amount* is the attribute of *participates*

**Step three - Build ER diagram**

---

**Exercise 2.3** Consider the following information about a university database:

• Professors have an SSN, a name, an age, a rank, and a research specialty.  
• Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.  
• Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).
• Each project is managed by one professor (known as the project’s principal investigator).
• Each project is worked on by one or more professors (known as the project’s co-investigators).
• Professors can manage and/or work on multiple projects.
• Each project is worked on by one or more graduate students (known as the project’s research assistants).
• When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
• Departments have a department number, a department name, and a main office.
• Departments have a professor (known as the chairman) who runs the department.
• Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
• Graduate students have one major department in which they are working on their degree.
• Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.
• Design and draw an ER diagram that captures the information about the university. Use only the basic ER model here, that is, entities, relationships, and attributes. Be sure to indicate any key and participation constraints.

Exercise 2.5 Notown Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database. The company has wisely chosen to hire you as a database designer (at your usual consulting fee of $2,500/day).

• Each musician that records at Notown has an SSN, a name, an address, and a phone number. Poorly paid musicians often share the same address, and no address has more than one phone.
• Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-.at, E-.at).
• Each album that is recorded on the Notown label has a title, a copyright date, a format (e.g., CD or MC), and an album identifier.
• Each song recorded at Notown has a title and an author.
• Each musician may play several instruments, and several musicians may play a given instrument.
• Each album has a number of songs on it, but no song may appear on more than one album.
• Each song is performed by one or more musicians, and a musician may perform a number of songs.
• Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

Design and draw an ER diagram for your schema. The following information describes the situation that the Notown database must model. Be sure to indicate all key and cardinality constraints and any assumptions that you make. Identify any constraints that you are unable to capture in the ER diagram and briefly explain why you could not express them.