Announcements

- New (better?) room: Ryerson 251.
- Assignment 1 is out.
  - Due next Thursday (Oct 4) in class.
  - No late exemptions!
- Office hours:
  - Professor: 3-4pm Tuesdays and Thursdays; also, by appointment.
  - TAs: 5-6pm Monday – Thursday.
- As always, details are on the web!

Entity/Relationship Model

Diagrams to represent designs.
- **Entity** like object, = “thing.”
- **Entity set** like class = set of “similar” entities/objects.
- **Attribute** = property of entities in an entity set, similar to fields of a struct.
- In diagrams, entity set → rectangle; attribute → oval.

![Entity/Relationship Model Diagram](image)

Relationships

- Connect two or more entity sets.
- Represented by diamonds.

![Relationships Diagram](image)

The Big Picture

- Stages of building a database application:
  - Real-world domain.
    - understand client needs.
  - Design data model:
    - using entity-relationship (E/R) model
  - Database data model:
    - using relational model
  - Create schema in DBMS, load data.
  - Open for business!

Beers-Bars-Drinkers Example

- Bars:
  - name, addr, license
- Beers:
  - name, manf
- Drinkers:
  - name, addr
- Frequents:
  - name
- Serves:
  - name

![Beers-Bars-Drinkers Example Diagram](image)
Attributes on Relationships

- Shorthand for 3-way relationship:

\[
\text{Bars} \rightarrow_{\text{Serves}} \text{Beers} \quad \text{Bars} \rightarrow_{\text{Serves}} \text{Prices} \quad \text{Prices} \rightarrow_{\text{Serves}} \text{Beers}
\]

Attributes on Relationships

- A true 3-way relationship.
  - Price depends jointly on beer and bar.
  - Notice arrow convention for multiway relationships: “all other E.S. determine one of these.”
    - Not sufficiently general to express any possibility.
    - However, if price, say, depended only on the beer, then we could use two 2-way relationships: price-beer and beer-bar.
    - Or better: just make price an attribute of beer.

Converting Multiway to 2-Way

- Not required in E/R, but necessary in certain “object-oriented” models.
- Create a new connecting E.S. to represent rows of a relationship set.
  - E.g., (Jimmy's, Bud, $2.50) for the Serves relationship.
- Many-one relationships from the connecting E.S. to the others.

More Design Issues

1. Subclasses.
2. Keys.
3. Weak entity sets.

Subclasses

Subclass = special case = fewer entities = more properties.
- Example: Ales are a kind of beer. In addition to the properties (= attributes and relationships) of beers, there is a “color” attribute for ales.
E/R Subclasses

- *isa* triangles indicate the subclass relation.

```
name  | Beers | manf
Ales  |       |

color | Ales  | Pete's Ale
```

Problems

How should conflicts be resolved?
- Example: *manf* means grower for wines, bottler for beers. What does *manf* mean for “grape beers”?
- Need ad-hoc notation to resolve meanings.
- In practice, we shall assume a tree of entity sets connected by *isa*, with all “isas” pointing from child to parent.

Different Subclass Viewpoints

1. E/R viewpoint: An entity has a *component* in each entity set to which it logically belongs.
   - Its properties are the union of the properties of these E.S.
2. Contrasts with object-oriented viewpoint: An object (entity) belongs to exactly one class.
   - It *inherits* properties of its superclasses.

```
name  | Beers | manf
Ales  |       |

color | Ales  | Pete's Ale
```

Keys

A *key* is a set of attributes whose values can belong to at most one entity.
- In E/R model, every E.S. must have a key.
  - It could have more than one key, but one set of attributes is the “designated” key.
  - In E/R diagrams, you should underline all attributes of the designated key.

Multiple Inheritance

Theoretically, an E.S. could be a subclass of several other entity sets.

```
name  | Beers | name  | Wines
Ales  |       |

color | Ales  | Grape Beers
```

Example

- Suppose *name* is key for *Beers*.
  - Beer name is also key for ales.
    - In general, key at root is key for all.

```
name  | Beers | name  | Wines
Ales  |       |

color | Ales  | Pete's Ale
```
Example: A Multiattribute Key

- Possibly, hours + room also forms a key, but we have not designated it as such.

Logins (Email Addresses)

- Design issue: Under what circumstances could we simply make login-name and host-name be attributes of logins, and dispense with the weak E.S.?

Weak Entity Sets

- Sometimes an E.S. E’s key comes not (completely) from its own attributes, but from the keys of one or more E.S.’s to which E is linked by a supporting many-one relationship.
- Called a weak E.S.
- Represented by putting double rectangle around E and a double diamond around each supporting relationship.
- Many-one-ness of supporting relationship (includes 1-1) essential.
  - With many-many, we wouldn't know which entity provided the key value.

Example: Chain of “Weakness”

- Consider IP addresses consisting of a primary domain (e.g., edu), subdomain (e.g., uchicago), and host (e.g., cs).

Example: Logins (Email Addresses)

- Login name = user name + host name, e.g., evtimov@cs.uchicago.edu.
- A “login” entity corresponds to a user name on a particular host, but the passwd table doesn’t record the host, just the user name, e.g., evtimov.
- Key for a login = the user name at the host (which is unique for that host only) + the IP address of the host (which is unique globally).

Chain of Keys

- Key for primary domain = its name.
- Key for secondary domain = its name + name of primary domain.
- Key for host = its name + key of secondary domain = its name + name of secondary domain + name of primary domain.
All “Connecting” Entity Sets Are Weak

- In this special case, where bar and beer determine a price, we can omit price from the key, and remove the double diamond from ThePrice.
- Better: price is an attribute of BBP.

Design Principles

- Faithfulness to requirements.
  - Remember the design schema should enforce as many constraints as possible. Don't rely on future data to follow assumptions.
  - Example: If registrar wants to associate only one instructor with a course, don't allow sets of instructors and count on departments to enter only one instructor per course.

Good and Bad Design

Good:

Bad (redundancy): repeats manufacturer address for each beer they manufacture.

Good and Bad Design

Bad (needless intermediate):

- Question: Why is it OK to have Beers with just its key as attribute? Why not make set of beers an attribute of manufacturers?