Review
- A data model is a group of concepts for describing data.
- What are the two terms used by ER model to describe a miniworld?
  - Entity
  - Relationship
- What makes a good database design

Topics Next
- Conversion of ER models to Schemas
- Reading Assignment
  - Chapter 3.1, 3.2
A Relation is a Table

Attributes (column headers)

<table>
<thead>
<tr>
<th>name</th>
<th>manf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winterbrew</td>
<td>Pete's</td>
</tr>
<tr>
<td>Bud Lite</td>
<td>Anheuser-Busch Beers</td>
</tr>
</tbody>
</table>

Tuples (rows)

Schemas

- Relation schema = relation name + attributes, in order (+ types of attributes).
- Example: Beers(name, manf) or Beers(name: string, manf: string)

- Database = collection of relations.
- Database schema = set of all relation schemas in the database.
Why Relations?

- Very simple model.
- Often matches how we think about data.
- Abstract model that underlies SQL, the most important database language today.
- But SQL uses bags, while the relational model is a set-based model.

From E/R Diagrams to Relations

- Entity sets become relations with the same set of attributes.
- Relationships become relations whose attributes are only:
  - The keys of the connected entity sets.
  - Attributes of the relationship itself.

Entity Set -> Relation

Relation: Beers(name, manf)
Combining Relations

- It is OK to combine the relation for an entity-set $E$ with the relation $R$ for a many-one relationship from $E$ to another entity set.
- Example: Drinkers(name, addr) and Favorite(drinker, beer) combine to make Drinker1(name, addr, favBeer).

Risk with Many-Many Relationships

- Combining Drinkers with Likes would be a mistake. It leads to redundancy, as:

```
<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>beer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sally</td>
<td>123 Maple</td>
<td>Bud</td>
</tr>
<tr>
<td>Sally</td>
<td>123 Maple</td>
<td>Miller</td>
</tr>
</tbody>
</table>
```

Redundancy
Handling Weak Entity Sets

- Relation for a weak entity set must include attributes for its complete key (including those belonging to other entity sets), as well as its own, nonkey attributes.
- A supporting (double-diamond) relationship is redundant and yields no relation.

Example

```
Logins
  name
  time

Hosts
  name

At
  loginName, hostName, time

Hosts(hostName)
Logins(loginName, hostName, time)
At(loginName, hostName, hostName2)
```

At becomes part of Logins

Must be the same
A (Slightly) Formal Definition

- A **database** is a collection of **relations** (or tables)
- Each **relation** is identified by a name and a list of **attributes** (or columns)
- Each **attribute** has a name and a **domain** (or type)
  - Set-valued attributes not allowed

Schema versus instance

- **Schema** (metadata)
  - Specification of how data is to be structured logically
  - Defined at set-up
  - Rarely changes
- **Instance**
  - Content
  - Changes rapidly, but always conforms to the schema
- Compare to type and objects of type in a programming language

Example

- **Schema**
  - Student (SID integer, name string, age integer, GPA float)
  - Course (CID string, title string)
  - Enroll (SID integer, CID integer)
- **Instance**
  - { h142, Bart, 10, 2.3, h123, Milhouse, 10, 3.1, ... }
  - { hCPS116, Intro. to Database Systems, ... }
  - { h142, CPS116, h142, CPS114, ... }