Relation: Is it a table? A set of tuples?

Beers

name	brewery	abv
Amnesiac	Phillips	8.5
Fat Tug IPA	Driftwood	7.0

```
Beers = (("Amnesiac", "Phillips", 8.5),
         ("Fat Tug IPA", "Driftwood", 7.0))
```

Relational Model: Basic concepts

- Attributes
- Tuple
- Domain
- Relation schema
- Relation instance
- Database
- Database schema

But why "relations"?

- Model is very simple
  - Mathematically tractable (e.g., relations are sets of tuples, not lists)
  - Simplicity permits an efficient implementation
- Model often matches how we think about data
  - (But not always!)
- The model is the one underlying SQL (and SQL is the most important database language today)

Oh dear, the instructor likes beer...

Beers (name: string, brewery: string, abv: real)
Pub (name: string, address: string, URL: string)
Patron (name: string, address: string, phone: string)
Likes (patron: string, beer: string)
Sells (pub: string, beer: string, price: real)
Frequents (patron: string, pub: string)

Oh dear, the instructor likes beer...

Beers (name: string, brewery: string, abv: real)
Pub (name: string, address: string, URL: string)
Patron (name: string, address: string, phone: string)
Likes (patron: string, beer: string)
Sells (pub: string, beer: string, price: real)
Frequents (patron: string, pub: string)

- Example of a constraint ("key")
  - Two tuples in a relation instance may not have the same values in all attributes of the key
  - Convention is the underline the key attribute(s)

Relations in SQL

- Declaring a relation:

```
CREATE TABLE <name> (
    <list of elements>
);
```

- Deleting a relation:

```
DROP TABLE <name>;
```

Relations in SQL: Declaration elements

- Most basic element: the **name** of an attribute, and the **type** of the attribute
- Most common types in SQL:
  - INT or INTEGER
  - REAL or FLOAT
  - CHAR(n) = fixed-length string of n characters
  - VARCHAR(n) = variable-length string of up to n characters

Example: Create table

```
CREATE TABLE Sells (
    pub    CHAR(20),
    beer   VARCHAR(20),
    price  REAL
);
```

SQL values

- INTEGERS and REALs have the expected literal representations
- Strings are also unsurprising, although they do require single quotes
  - Use two single quotes in order to escape a single quote
  - Example: 'Bob''s Bar'
- Booleans are TRUE, FALSE, and UNKNOWN
- Dates and times: DATE, TIME, DATETIME (and sometimes TIMESTAMP)
- Any attribute can take the NULL value...

More about DATEs and TIMEs

- Form of a date: zero-padding is used
- Example: International "Talk Like a Pirate" Day this year is September 19, 2013
  - '2013-09-19'
  - (Some SQL implementations require DATE to precede the literal)
- From of time: 'hh:mm:ss'
  - Optional decimal point available for fractional seconds
- Example: Time at which CSC 370 students are all in their seats for lecture: Eight and half seconds after 10:30 am
  - '10:30:08.5'
  - (Some SQL implementations require TIME to precede the literal)

Relations in SQL: Declaring keys

- An **attribute** or **list of attributes** may be declared PRIMARY KEY or UNIQUE
  - There is a very subtle difference
- Either form indicates that no two tuples of the relation may agree in all attributes on the list
- This is an example of a constraint in the data model.
- Example: **single-attribute key**

```
CREATE TABLE Beers(
    name    CHAR(20) UNIQUE,
    brewery VARCHAR(20),
    abv     REAL
);
```