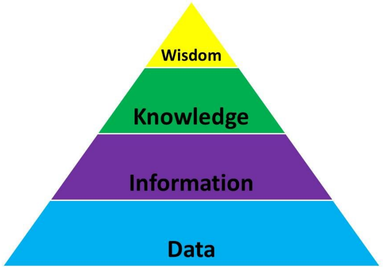
Data, Information, and Knowledge

INFO 8000 Spring 2020



Information

Facts that can be stated about a particular entity in the world.

E.g. I am tall, specifically 6 ft 3 in

E.g. The sky is blue

E.g. The time is 3:30 PM

Data

A quantifiable record, possibly representing information

Can be analog or digital

- Analog - The volume of water in a container, written words
- Digital - Represented by a finite number system, e.g. binary or text

Knowledge

Information that is learned or acquired through experience.

Knowledge is *built* to understand and make decisions concerning the world

Concretely, one can represent knowledge as, for example, a table of information.

Higher order knowledge can be represented as connections between tables, or rules governing the addition of information to tables.

Note: General knowledge can be stated or written down. Tacit knowledge can be used, but is difficult to state or transmit to others. For example, you may “know” how to read a language, but the details of that knowledge are inaccessible to you.

Wisdom

An ability to apply knowledge to achieve goals, and can only be judged by determined successes and failures.

- Training a neural network could be described as making it “wise”

Exercise Part 1

Let’s write down some information we know about this class section

E.g. Section of INFO 8000, a Course at the University of Georgia

Starts at 3:30PM

On Tuesdays and Thursdays

Section Number (CRN) - 12345

Name of Instructor - Kyle Johnsen

Location is Boyd 302

Number of students - 14

Students in the class

Credit hours

Length

Exercise Part 2

Now, let’s consider how we represent that knowledge.

Relational Database Management Systems (RDBMS)

Computer systems formed out of the need to create, retrieve, update, and delete accurate information efficiently and consistently.

Modeled as a Schema of Entities and Relations, which is often represented as as an Entity-Relationship Diagram or ERD.

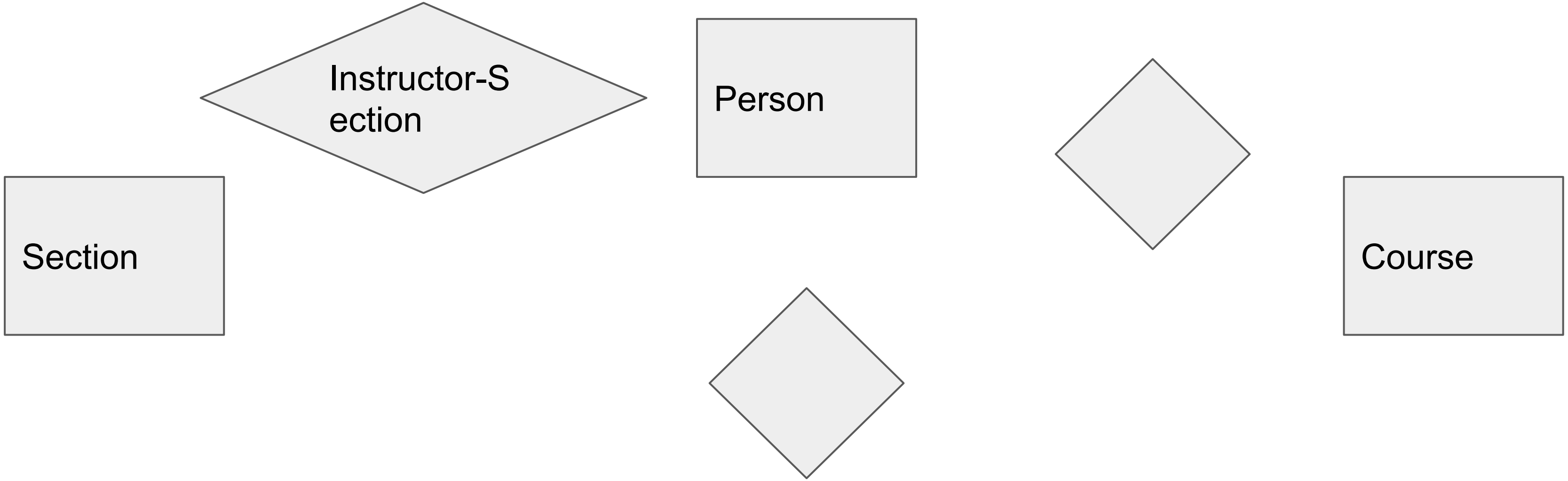
Entities and Relations

- Entity - Description of information about a single thing that can be referred to, i.e. have a relationship with, another entity.
    - Entities have instances, which are the concrete pieces of information that must be stored.
- Relation - How entities reference each other
    - Cardinality determines how many relationships may be present between sets of entities
        - 0 - 1, optional, but only 1 between any 2
        - 0 - M, optional, but any number
        - 1 - 1, required to have a reference between entity instances (also 1-2, etc)
        - 1 - M, required to have at least one reference between entity instances
        - M - N, may have many references, e.g. (A1,B1), (A1,B2), (A2, B1), (A2,B2)

Example ERD for Knowledge about Classes

3 Entities, Section, Person, Course

3 Relationships, Instructor-Section, Student-Section, Section-Course



Turning ERDs into Tables

An Entity is always a table, with columns for each piece of information that is to be encoded.

Cardinality determines how a relationship is mapped to tables

- A relationship that is 1-1 may be a joint table.
- 1-M may be encoded as a link to a unique identifier in another table.
- All other relationships require a separate table to represent (0-M), (0-1), (M-N)

Enforcing cardinality is a different problem (constraints)

SQL

Structured Query Language

A “declarative” programming language used to interact with the features of an RDBMS, which normally means the creation, retrieval, updating, and deletion of:

- Tables
- Instances (records)
- Constraints

SQLite

A self-contained, local, RDBMS that allows the use of SQL queries to manage knowledge

Not designed for high performance, and are tightly associated with application specific code.

Great for learning, because you don’t have to run a separate SQL service. Definitely can be used in practice, especially when you want consistency constraints to be enforced, which are rarely part of other table systems (e.g. Pandas DataFrames).

SQL Create Database and Create Table

CREATE DATABASE DatabaseName]

- E.g. CREATE DATABASE Example;

CREATE TABLE [IF NOT EXISTS] TableName (`ColumnName` DataType);

- E.g. CREATE TABLE student (`id` INTEGER NOT NULL);
- E.g. CREATE TABLE student (`id` INTEGER,`first_name` TEXT);

SQLite DataTypes

- NULL, INTEGER, REAL, TEXT, BLOB (supports various other names common in other database systems, but converts to internal types)

Options and Constraints

Options for a field include

- NOT NULL
- AUTOINCREMENT (for integer type)
- DEFAULT Value

Constraints include

- PRIMARY KEY (`field1`,`field2`)
- FOREIGN KEY (`field1`) REFERENCES table(`field2`) ON DELETE action ON UPDATE action
  - Actions are usually one of: SET NULL, SET DEFAULT, RESTRICT, CASCADE
- CHECK (Expression)
  - Expression is some combination of fields and operations, e.g. price >= 0 AND price >= cost