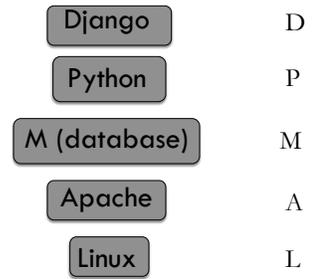
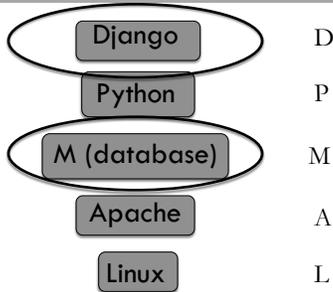


# ECS 89

## Django on top of it all



## Django on top of it all



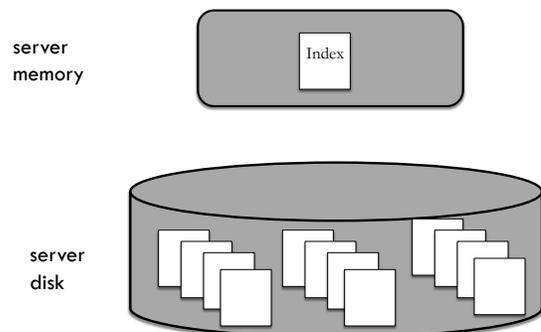
## Database

- A database is a collection of files storing information for an app, usually on the server
- The spreadsheet in the last assignment was a simple database
- Files are stored on disk
- More complicated databases might have several files, either as a way of organizing data or because the dataset is too big to fit into one file.
- What would be wrong with a single gigantic file?

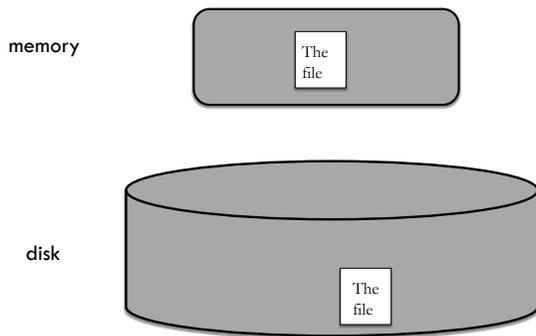
## Wrong with a single gigantic file

- Not that it all has to be loaded into memory at once; files stream (read a bit, use it, throw it away...)
- But you still have to read the whole thing from beginning to end. Better just to go for what you want, somewhere in the middle.

## A big database



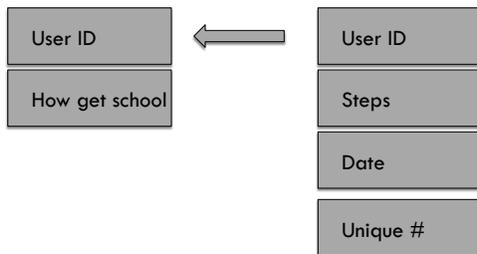
## A small database (our programs)



## Tables

- A database is made up of tables
- Same data organization as a spreadsheet
- Rows (eg. pedometer records)
- Columns (eg. steps, data, userID)
  
- A database may include several tables.
- We'll have a user table, with information about users, and a pedometer table, with pedometer records.

## Model we're after



## Django – a framework

- Say you write 10 Web apps, all with the same pieces:
  - ▣ Database
  - ▣ Data loading via Web forms, file uploads
  - ▣ Output Web pages
  - ▣ Output calculations, tables, graphs
- Framework tries to handle the repetitive parts of this process automatically
- Makes a lot of stub files for you, you fill in specifics
- Final app is a collaboration

## Django

- Developed for building a news site
- Two sets of customers
  - ▣ Internal - reporters and Web site maintainers, who load files and data into the database
  - ▣ External - readers who access the Web site
- Django developer (that's you) makes interfaces for both sets of customers
- Basic decision is database model

## Poll tutorial

- Run through Web tutorial
- We're using Django 1.5; don't try the 1.7 version
  
- Notice as we go along that database tables are represented by special Python objects.
  - ▣ Instance of object = row of table
  - ▣ Attribute of object = column of table
  
- manage.py modifies Django's part of the code

## Python features

- `__init__.py` file in directories created by Django is an empty file. Indicates that the code in the directory is a Python package. This is a group of modules that go together.

- Object inheritance

```
class Poll(models.Model):
```

This means take Django's base `Model` class as a beginning for our new `Poll` class. `Poll` inherits all methods and attributes from `Model`. Objects that inherit from `Model` correspond to database tables.