

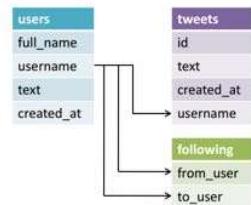


IMPORTING DATA IN PYTHON I

Welcome to the course!

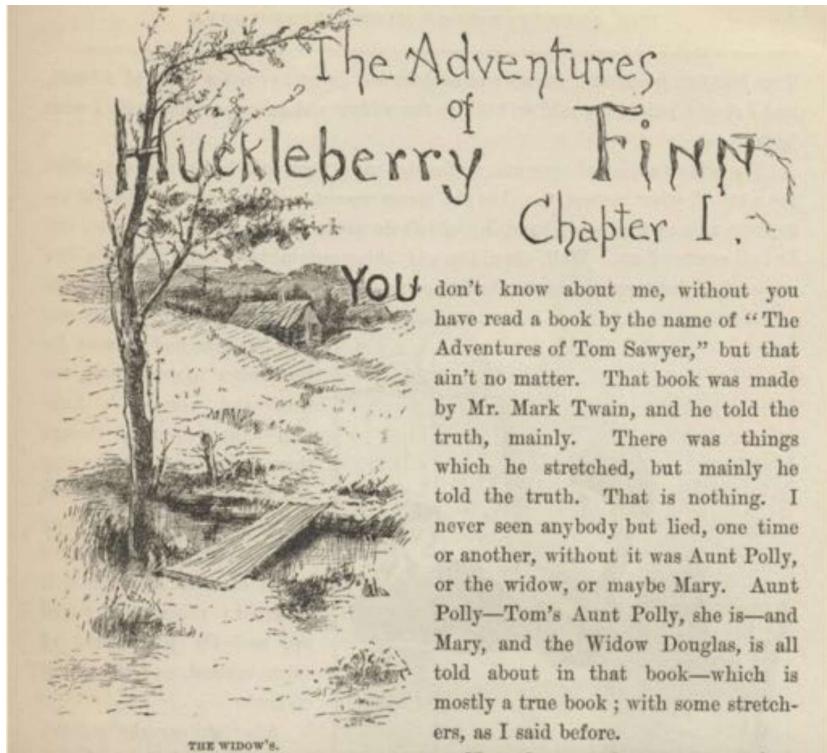
Import data

- Flat files, e.g. .txts, .csvs
- Files from other software
- Relational databases





Plain text files



Source: Project Gutenberg



Table data

titanic.csv

	Name	Sex	Cabin	Survived
Braund, Mr. Owen Harris	male	NaN	0	
Cumings, Mrs. John Bradley	female	C85	1	
Heikkinen, Miss. Laina	female	NaN	1	
Futrelle, Mrs. Jacques Heath	female	C123	1	
Allen, Mr. William Henry	male	NaN	0	

row

column

- Flat file

Reading a text file

```
In [1]: filename = 'huck_finn.txt'

In [2]: file = open(filename, mode='r')    # 'r' is to read

In [3]: text = file.read()

In [4]: file.close()
```



Printing a text file

```
In [5]: print(text)
```

YOU don't know about me without you have read a book by the name of The Adventures of Tom Sawyer; but that ain't no matter. That book was made by Mr. Mark Twain, and he told the truth, mainly. There was things which he stretched, but mainly he told the truth. That is nothing. never seen anybody but lied one time or another, without it was Aunt Polly, or the widow, or maybe Mary. Aunt Polly--Tom's Aunt Polly, she is--and Mary, and the Widow Douglas is all told about in that book, which is mostly a true book, with some stretchers, as I said before.

Writing to a file

```
In [1]: filename = 'huck_finn.txt'  
In [2]: file = open(filename, mode='w')    # 'w' is to write  
In [3]: file.close()
```

Context manager with

```
In [1]: with open('huck_finn.txt', 'r') as file:  
....:     print(file.read())
```

YOU don't know about me without you have read a book by the name of The Adventures of Tom Sawyer; but that ain't no matter. That book was made by Mr. Mark Twain, and he told the truth, mainly. There was things which he stretched, but mainly he told the truth. That is nothing. never seen anybody but lied one time or another, without it was Aunt Polly, or the widow, or maybe Mary. Aunt Polly--Tom's Aunt Polly, she is--and Mary, and the Widow Douglas is all told about in that book, which is mostly a true book, with some stretchers, as I said before.



In the exercises, you'll:

- Print files to the console
- Print specific lines
- Discuss flat files



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

The importance of flat files in data science

Flat files

titanic.csv

column

PassengerId, Survived, Pclass, Name, Gender, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked

1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2.
3101282,7.925,,S

row

	Name	Gender	Cabin	Survived
Braund, Mr. Owen Harris	male	NaN	0	
Cumings, Mrs. John Bradley	female	C85	1	
Heikkinen, Miss. Laina	female	NaN	1	
Futrelle, Mrs. Jacques Heath	female	C123	1	
Allen, Mr. William Henry	male	NaN	0	

Flat files

- Text files containing records
- That is, table data
- Record: row of fields or attributes
- Column: feature or attribute

titanic.csv

```
PassengerId,Survived,Pclass,Name,Gender,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
```

Header

titanic.csv

```
PassengerId,Survived,Pclass,Name,Gender,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC
17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,
35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,
27,0,2,347742,11.1333,,S
```

File extension

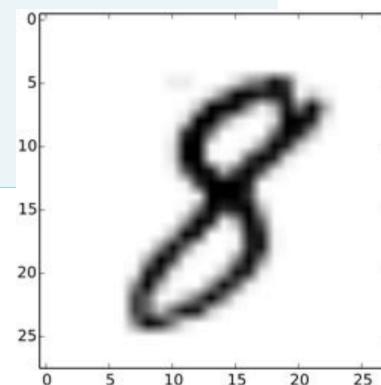
- .csv - Comma separated values
- .txt - Text file
- commas, tabs - Delimiters



Tab-delimited file

MNIST.txt

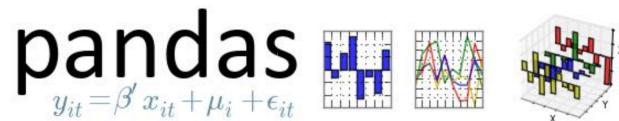
```
pixel149  pixel150  pixel151  pixel152  pixel153  
0          0          0          0          0  
86         250        254        254        254  
0          0          0          9          254  
0          0          0          0          0  
103        253        253        253        253  
0          0          5          165        254  
0          0          0          0          0  
0          0          0          0          0  
0          0          0          0          41  
253        253        253        253        253
```



MNIST
image →

How do you import flat files?

- Two main packages: NumPy, pandas



- Here, you'll learn to import:
 - Flat files with numerical data (MNIST)
 - Flat files with numerical data and strings (titanic.csv)



IMPORTING DATA IN PYTHON I

Let's practice!



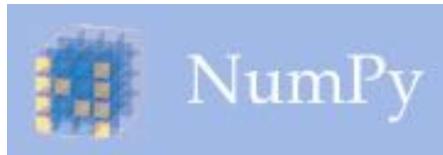
IMPORTING DATA IN PYTHON I

Importing flat files using NumPy



Why NumPy?

- NumPy arrays: standard for storing numerical data
- Essential for other packages: e.g. scikit-learn



- `loadtxt()`
- `genfromtxt()`



Importing flat files using NumPy

```
In [1]: import numpy as np
```

```
In [2]: filename = 'MNIST.txt'
```

```
In [3]: data = np.loadtxt(filename, delimiter=',')
```

```
In [4]: data
```

```
Out[4]:
```

```
[[ 0.    0.    0.    0.    0.]
 [ 86.   250.   254.   254.   254.]
 [ 0.    0.    0.    9.    254.]
 ...,
 [ 0.    0.    0.    0.    0.]
 [ 0.    0.    0.    0.    0.]
 [ 0.    0.    0.    0.    0.]]
```

Customizing your NumPy import

```
In [1]: import numpy as np

In [2]: filename = 'MNIST_header.txt'

In [3]: data = np.loadtxt(filename, delimiter=',',
skiprows=1)

In [4]: print(data)
[[  0.    0.    0.    0.    0.]
 [ 86.   250.   254.   254.   254.]
 [  0.    0.    0.    9.   254.]
 ...,
 [  0.    0.    0.    0.    0.]
 [  0.    0.    0.    0.    0.]
 [  0.    0.    0.    0.    0.]]
```



Customizing your NumPy import

```
In [1]: import numpy as np

In [2]: filename = 'MNIST_header.txt'

In [3]: data = np.loadtxt(filename, delimiter=',', skiprows=1,
usecols=[0, 2])

In [4]: print(data)
[[ 0.    0.]
 [ 86.   254.]
 [ 0.    0.]
 ...,
 [ 0.    0.]
 [ 0.    0.]
 [ 0.    0.]]
```



Customizing your NumPy import

```
In [1]: data = np.loadtxt(filename, delimiter=',',  
dtype=str)
```



Mixed datatypes

titanic.csv

	Name	Gender	Cabin	Fare
Braund, Mr. Owen Harris	male	NaN	7.3	
Cumings, Mrs. John Bradley	female	C85	71.3	
Heikkinen, Miss. Laina	female	NaN	8.0	
Futrelle, Mrs. Jacques Heath	female	C123	53.1	
Allen, Mr. William Henry	male	NaN	8.05	

↑
strings

↑
floats



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Importing flat files using pandas



What a data scientist needs

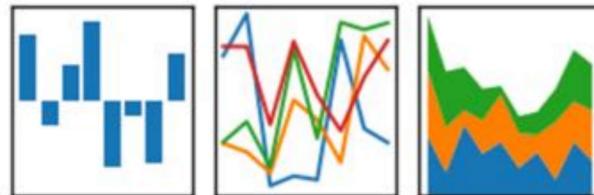
- Two-dimensional labeled data structure(s)
- Columns of potentially different types
- Manipulate, slice, reshape, groupby, join, merge
- Perform statistics
- Work with time series data

Pandas and the DataFrame



pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Wes McKinney



Pandas and the DataFrame

What problem does *pandas* solve?

Python has long been great for data munging and preparation, but less so for data analysis and modeling. *pandas* helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

- DataFrame = pythonic analog of R's data frame



Pandas and the DataFrame

**Hadley Wickham**

@hadleywickham



Following

A matrix has rows and columns. A data frame
has observations and variables. #rstats #tidydata

RETWEETS

128

LIKES

233





Manipulating pandas DataFrames

- Exploratory data analysis
 - Data wrangling
 - Data preprocessing
 - Building models
 - Visualization
-
- Standard and best practice to use pandas

Importing using pandas

```
In [1]: import pandas as pd
```

```
In [2]: filename = 'winequality-red.csv'
```

```
In [3]: data = pd.read_csv(filename)
```

```
In [4]: data.head()
```

```
Out[4]:
```

```
    volatile acidity  citric acid  residual sugar
0              0.70        0.00          1.9
1              0.88        0.00          2.6
2              0.76        0.04          2.3
3              0.28        0.56          1.9
4              0.70        0.00          1.9
```

```
In [5]: data_array = data.values
```



You'll experience:

- Importing flat files in a straightforward manner
- Importing flat files with issues such as comments and missing values



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Final thoughts on data import

Next chapters:

- Import other file types:
 - Excel, SAS, Stata
- Feather



Wes McKinney
@wesmckinn



Following

Announcing Feather: A fast, language-agnostic
data frame file format, by
[@hadleywickham](#) and [@wesmckinn](#)

- Interact with relational databases



Next course:

- Scrape data from the web
- Interact with APIs



IMPORTING DATA IN PYTHON I

Congratulations!



IMPORTING DATA IN PYTHON I

Introduction to other file types



Other file types

- Excel spreadsheets
- MATLAB files
- SAS files
- Stata files
- HDF5 files



Pickled files

- File type native to Python
- Motivation: many datatypes for which it isn't obvious how to store them
- Pickled files are serialized
- Serialize = convert object to bytestream



Pickled files

```
In [1]: import pickle  
  
In [2]: with open('pickled_fruit.pkl', 'rb') as file:  
...:     data = pickle.load(file)  
  
In [3]: print(data)  
{'peaches': 13, 'apples': 4, 'oranges': 11}
```



Importing Excel spreadsheets

```
In [1]: import pandas as pd  
  
In [2]: file = 'urbanpop.xlsx'  
  
In [3]: data = pd.ExcelFile(file)  
  
In [4]: print(data.sheet_names)  
['1960-1966', '1967-1974', '1975-2011']  
  
In [5]: df1 = data.parse('1960-1966') ← sheet name, as a string  
  
In [6]: df2 = data.parse(0) ← sheet index, as a float
```



You'll learn:

- How to customize your import
 - Skip rows
 - Import certain columns
 - Change column names



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Importing SAS/Stata files using pandas



SAS and Stata files

- SAS: Statistical Analysis System
 - Stata: “Statistics” + “data”
-
- SAS: business analytics and biostatistics
 - Stata: academic social sciences research



SAS files

- Used for:
 - Advanced analytics
 - Multivariate analysis
 - Business intelligence
 - Data management
 - Predictive analytics
- Standard for computational analysis



Importing SAS files

```
In [1]: import pandas as pd  
  
In [2]: from sas7bdat import SAS7BDAT  
  
In [3]: with SAS7BDAT('urbanpop.sas7bdat') as file:  
....:     df_sas = file.to_data_frame()
```



Importing Stata files

```
In [1]: import pandas as pd  
In [2]: data = pd.read_stata('urbanpop.dta')
```



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Importing HDF5 files



HDF5 files

- Hierarchical Data Format version 5
- Standard for storing large quantities of numerical data
- Datasets can be hundreds of gigabytes or terabytes
- HDF5 can scale to exabytes



Importing HDF5 files

```
In [1]: import h5py  
  
In [2]: filename = 'H-H1_LOSC_4_V1-815411200-4096.hdf5'  
  
In [3]: data = h5py.File(filename, 'r') # 'r' is to read  
  
In [4]: print(type(data))  
<class 'h5py._hl.files.File'>
```



The structure of HDF5 files

```
In [5]: for key in data.keys():
    ...:     print(key)
meta
quality
strain
```

```
In [6]: print(type(data['meta']))
<class 'h5py._hl.group.Group'>
```

This gives a high level picture of what's contained in a LIGO data file. There are 3 types of information:

- **meta**: Meta-data for the file, this is basic information such as the GPS times covered, which instrument, etc.
- **quality**: Refers to data quality. The main item here is a 1 Hz time series describing the data quality for each second of data. This is an important topic, and we'll devote a whole step of the tutorial to working with data quality information.
- **strain**: Strain data from the interferometer. In some sense, this is "the data", the main measurement performed by LIGO.



The structure of HDF5 files

```
In [7]: for key in data['meta'].keys():
    ...:     print(key)
Description
DescriptionURL
Detector
Duration
GPSstart
Observatory
Type
UTCstart

In [8]: print(data['meta']['Description'].value, data['meta']
['Detector'].value)
b'Strain data time series from LIGO' b'H1'
```



The HDF Project

- Actively maintained by the HDF Group



- Based in Champaign, Illinois



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Importing MATLAB files



MATLAB

- “Matrix Laboratory”
- Industry standard in engineering and science
- Data saved as .mat files





SciPy to the rescue!

- `scipy.io.loadmat()` - read .mat files
- `scipy.io.savemat()` - write .mat files



What is a .mat file?

The screenshot shows the MATLAB R2014b interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is selected, showing the code for 'Turing_two_cells.m'. The workspace on the left contains variables: k, j, m1, n, name, and p1. The command window at the bottom shows initial assignments for m1, p1, and name.

```
% Here we SOLVE the system of ODEs that
% describes Turing pattern formation in (Turing, 1952)
% for the 2 cell, discrete system.
% We use ode15s but, for most parameters, ode45 should work.
% [T,Y] = ode15s(glt,y0,t0,t1,[y100 y200 y300 y400]) numerically
% integrates from time t0 to time t1 with initial condition vector
% [y100 y200 y300 y400].
%
% [T,Y] = ode15s(glt,y0,t0,t1,y0(:,1));
% [T,Y] = ode15s(glt,y0,[0 5],[1.06 1.02 0.94 0.98]);
%
% Here we plot morphogen concentrations as a function of time t.
%
figure
a1 = plot(T,Y(:,1),"-","LineWidth",3)
hold on
%
m1 = zeros(n,j); % initial mRNA 1
p1 = zeros(n,j); % initial protein 1
>> name = "hugobowne";

Name =
```

hugobowne



Importing a .mat file

```
In [1]: import scipy.io  
  
In [2]: filename = 'workspace.mat'  
  
In [3]: mat = scipy.io.loadmat(filename)  
  
In [4]: print(type(mat))  
<class 'dict'>  
  
In [5]: print(type(mat['x']))  
<class 'numpy.ndarray'>
```

- **keys** = MATLAB variable names
- **values** = objects assigned to variables



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Introduction to relational databases



What is a relational database?

- Based on *relational model of data*
- First described by Edgar “Ted” Codd



Example: Northwind database

- Orders table

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	41.34	Victuailles en stock	2, rue du Commerce
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	22.98	Chop-suey Chinese	Hauptstr. 31

- Customers table

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	None	12209	Germany
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	None	WA1 1DP	UK
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	None	68306	Germany
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	None	13008	France

- Employees table

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region
1	Davolio	Nancy	Sales Representative	Ms.	12/8/1948 12:00:00 AM	5/1/1992 12:00:00 AM	507 - 20th Ave. E.\nApt. 2A	Seattle	WA
2	Fuller	Andrew	Vice President, Sales	Dr.	2/19/1952 12:00:00 AM	8/14/1992 12:00:00 AM	908 W. Capital Way	Tacoma	WA
3	Leverling	Janet	Sales Representative	Ms.	8/30/1963 12:00:00 AM	4/1/1992 12:00:00 AM	722 Moss Bay Blvd.	Kirkland	WA



The Orders table

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	41.34	Victuailles en stock	2, rue du Commerce
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	22.98	Chop-suey Chinese	Hauptstr. 31



Tables are linked

- Orders table

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	41.34	Victuailles en stock	2, rue du Commerce
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	22.98	Chop-suey Chinese	Hauptstr. 31

- Customers table

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	None	12209	Germany
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	None	WA1 1DP	UK
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	None	68306	Germany
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	None	13008	France

- Employees table

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region
1	Davolio	Nancy	Sales Representative	Ms.	12/8/1948 12:00:00 AM	5/1/1992 12:00:00 AM	507 - 20th Ave. E.\nApt. 2A	Seattle	WA
2	Fuller	Andrew	Vice President, Sales	Dr.	2/19/1952 12:00:00 AM	8/14/1992 12:00:00 AM	908 W. Capital Way	Tacoma	WA
3	Leverling	Janet	Sales Representative	Ms.	8/30/1963 12:00:00 AM	4/1/1992 12:00:00 AM	722 Moss Bay Blvd.	Kirkland	WA



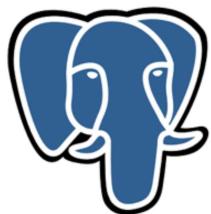
Relational model

- Widely adopted
- Todd's 12 Rules/Commandments
 - Consists of 13 rules (zero-indexed!)
 - Describes what a Relational Database Management System should adhere to to be considered relational



Relational Database Management Systems

- PostgreSQL
- MySQL
- SQLite
- SQL = Structured Query Language



PostgreSQL





IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Creating a database engine in Python



Creating a database engine

- SQLite database
 - Fast and simple
- SQLAlchemy
 - Works with many Relational Database Management Systems

```
In [1]: from sqlalchemy import create_engine
```

```
In [2]: engine = create_engine('sqlite:///Northwind.sqlite')
```



Getting table names

```
In [1]: from sqlalchemy import create_engine
```

```
In [2]: engine = create_engine('sqlite:///Northwind.sqlite')
```

```
In [3]: table_names = engine.table_names()
```

```
In [4]: print(table_names)
['Categories', 'Customers', 'EmployeeTerritories',
'Employees', 'Order Details', 'Orders', 'Products',
'Region', 'Shippers', 'Suppliers', 'Territories']
```



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Querying relational databases in Python



Basic SQL query

```
SELECT * FROM Table_Name
```

- Returns all columns of all rows of the table
- Example:

```
SELECT * FROM Orders
```

- We'll use SQLAlchemy and pandas



Workflow of SQL querying

- Import packages and functions
- Create the database engine
- Connect to the engine
- Query the database
- Save query results to a DataFrame
- Close the connection



Your first SQL query

```
In [1]: from sqlalchemy import create_engine  
  
In [2]: import pandas as pd  
  
In [3]: engine = create_engine('sqlite:///Northwind.sqlite')  
  
In [4]: con = engine.connect()  
  
In [5]: rs = con.execute("SELECT * FROM Orders")  
  
In [6]: df = pd.DataFrame(rs.fetchall())  
  
In [7]: con.close()
```



Printing your query results

```
In [8]: print(df.head())
      0      1      2                      3                      4
0  10248  VINET  5  7/4/1996 12:00:00 AM  8/1/1996 12:00:00 AM
1  10251  VICTE  3  7/8/1996 12:00:00 AM  8/5/1996 12:00:00 AM
2  10254  CHOPS  5  7/11/1996 12:00:00 AM  8/8/1996 12:00:00 AM
3  10256  WELLI  3  7/15/1996 12:00:00 AM  8/12/1996 12:00:00 AM
4  10258  ERNSH  1  7/17/1996 12:00:00 AM  8/14/1996 12:00:00 AM
```



Set the DataFrame column names

```
In [1]: from sqlalchemy import create_engine  
  
In [2]: import pandas as pd  
  
In [3]: engine = create_engine('sqlite:///Northwind.sqlite')  
  
In [4]: con = engine.connect()  
  
In [5]: rs = con.execute("SELECT * FROM Orders")  
  
In [6]: df = pd.DataFrame(rs.fetchall())  
  
In [7]: df.columns = rs.keys()  
  
In [8]: con.close()
```



Set the data frame column names

```
In [9]: print(df.head())
      OrderID CustomerID EmployeeID          OrderDate
0       10248     VINET             5  7/4/1996 12:00:00 AM
1       10251    VICTE             3  7/8/1996 12:00:00 AM
2       10254    CHOPS             5  7/11/1996 12:00:00 AM
3       10256    WELLI             3  7/15/1996 12:00:00 AM
4       10258   ERNSH             1  7/17/1996 12:00:00 AM
```



Using the context manager

```
In [1]: from sqlalchemy import create_engine  
  
In [2]: import pandas as pd  
  
In [3]: engine = create_engine('sqlite:///Northwind.sqlite')  
  
In [4]: with engine.connect() as con:  
...:     rs = con.execute("SELECT OrderID, OrderDate,  
...:                     ShipName FROM Orders")  
...:     df = pd.DataFrame(rs.fetchmany(size=5))  
...:     df.columns = rs.keys()
```



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Querying relational databases directly with pandas



The pandas way to query

```
In [1]: from sqlalchemy import create_engine  
  
In [2]: import pandas as pd  
  
In [3]: engine = create_engine('sqlite:///Northwind.sqlite')  
  
In [4]: with engine.connect() as con:  
...:     rs = con.execute("SELECT * FROM Orders")  
...:     df = pd.DataFrame(rs.fetchall())  
...:     df.columns = rs.keys()
```

```
In [5]: df = pd.read_sql_query("SELECT * FROM Orders", engine)
```



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Advanced querying: exploiting table relationships



Tables are linked

- Orders table

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	41.34	Victuailles en stock	2, rue du Commerce
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	22.98	Chop-suey Chinese	Hauptstr. 31

- Customers table

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	None	12209	Germany
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	None	WA1 1DP	UK
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	None	68306	Germany
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	None	13008	France

- Employees table

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region
1	Davolio	Nancy	Sales Representative	Ms.	12/8/1948 12:00:00 AM	5/1/1992 12:00:00 AM	507 - 20th Ave. E.\nApt. 2A	Seattle	WA
2	Fuller	Andrew	Vice President, Sales	Dr.	2/19/1952 12:00:00 AM	8/14/1992 12:00:00 AM	908 W. Capital Way	Tacoma	WA
3	Leverling	Janet	Sales Representative	Ms.	8/30/1963 12:00:00 AM	4/1/1992 12:00:00 AM	722 Moss Bay Blvd.	Kirkland	WA



JOINing tables

- Orders table

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName	ShipAddress
10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye
10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	41.34	Victuailles en stock	2, rue du Commerce
10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	22.98	Chop-suey Chinese	Hauptstr. 31

- Customers table

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	None	12209	Germany
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	None	WA1 1DP	UK
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	None	68306	Germany
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	None	13008	France



INNER JOIN in Python (pandas)

```
In [1]: from sqlalchemy import create_engine
```

```
In [2]: import pandas as pd
```

```
In [3]: engine = create_engine('sqlite:///Northwind.sqlite')
```

```
In [4]: df = pd.read_sql_query("SELECT OrderID, CompanyName FROM Orders INNER JOIN Customers on Orders.CustomerID = Customers.CustomerID", engine)
```

```
In [5]: print(df.head())
```

	OrderID	CompanyName
0	10248	Vins et alcools Chevalier
1	10251	Victuailles en stock
2	10254	Chop-suey Chinese
3	10256	Wellington Importadora
4	10258	Ernst Handel



IMPORTING DATA IN PYTHON I

Let's practice!



IMPORTING DATA IN PYTHON I

Final Thoughts



What you've learned:

- Relational databases
- Queries
 - SELECT
 - WHERE
 - JOIN



Next course:

- Scrape data from the web
- Interact with APIs



IMPORTING DATA IN PYTHON

Congratulations!