Introduction to Database Systems: CS312 SQLite and Python Primer

Oliver Bonham-Carter

2 Oct 2020

Exam1: Wed 7^{th} Oct. Given During Lab (3:00pm)

By Honor code: you cannot talk to your colleagues about exam questions

From the SLIDES

- Handed out: Wednesday, 7^{th} Oct.
- You will have 24 hours to take the exam,
- Submitted to GitHub by Thursday, 8^{th} by 3pm
- Builder files to create SQLite3 databases
- Creating a database from a datasets: Adding tables and data
- Writing queries
- Updating tables and data
- Using Python for simple automation
  - Basic steps to use Python to manage a database
  - Automating queries
  - (and similar conceptual questions)

Big Data

Multiples of bytes							
Decimal			Binary				
Value	Metric		Value	IEC		JEDEC	
1000	kB	kilobyte	1024	KiB	kibibyte	KB	kilobyte
1000^2	MB	megabyte	1024^2	MiB	mebibyte	MB	megabyte
1000^3	GB	gigabyte	1024^3	GiB	gibibyte	GB	gigabyte
1000^4	TB	terabyte	1024^4	TiB	tebibyte	–	
1000^5	PB	petabyte	1024^5	PiB	pebibyte	–	
1000^6	EB	exabyte	1024^6	EiB	exbibyte	–	
1000^7	ZB	**zettabyte**	1024^7	ZiB	zebibyte	–	
1000^8	YB	yottabyte	1024^8	YiB	yobibyte	–	

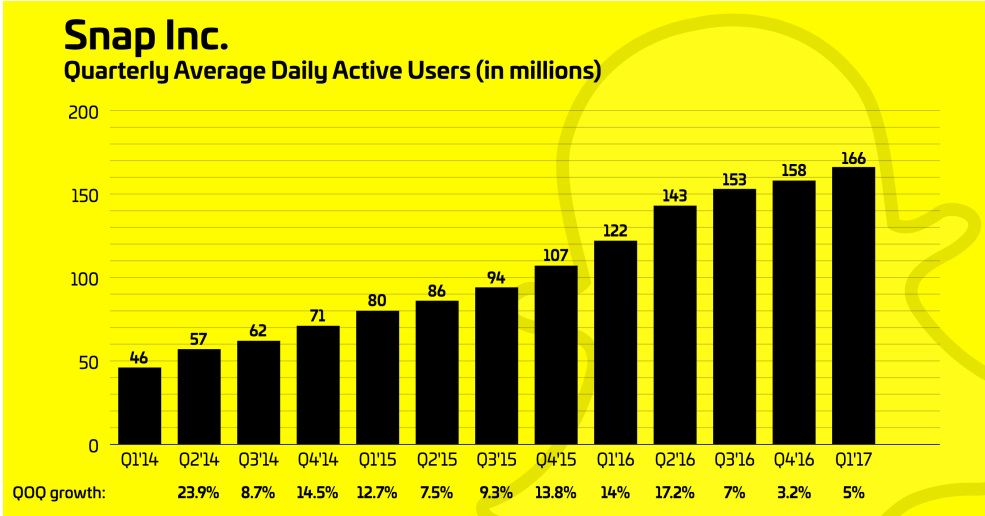
Orders of magnitude of data

- Upwards of 2.7 Zetabytes of data exist in the digital universe
- YouTube users upload 48 hours of new video every minute
- Increase in unstructured data: text, photos, etc.
- `https://www.waterfordtechnologies.com/big-data-interesting-facts/`

Facebook's Daily Data Use

- Facebook processes:
  - 2.5 billion pieces of content
  - upwards of 500 terabytes of data each day from status and location details
  - Processing in 2.7 billion Like actions
  - 300 million photos per day,
  - Scans roughly 105 terabytes of data each half hour
  - 100 petebytes of data are stored in a single Hadoop disk cluster (a distributed system for data management)

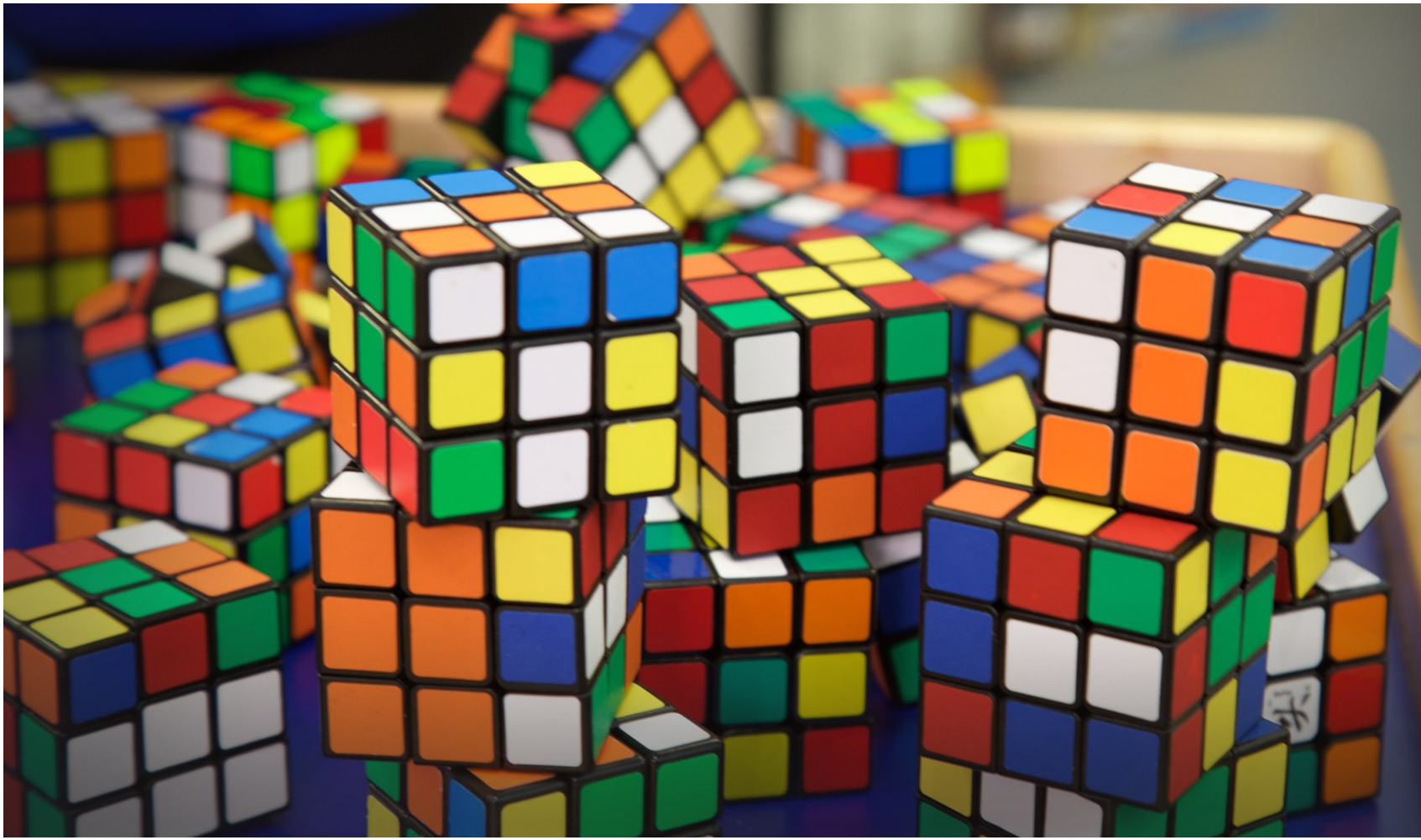
Current Estimates for Users Online



- Facebook: 2.7 Billion Active users
- Amazon: 112 Million (US users)
- SnapChat: 238 million daily active users worldwide
- Google: 4.39 Billion internet users (worldwide)
- Instagram: 1 Billion monthly active users, 500 Million each day.

Lots of names, photos, passwords and posts to record!

How are we to **manage** all this data?



Standardized Database Access with Python

PEP 0249

- Python Database API Specification v2.0
- `https://www.python.org/dev/peps/pep-0249/`
- Specifies a standard API that Python modules that are used to access databases should implement
- Does not provide a library nor a module, just specifications on how to make them
- Third party modules may adhere to these specifications

Steps to run a command in SQL using Python

Five basic steps to using a database according to the Python Database API Specification v2.0

- Step 0: Build automation framework in Python3
- Step 1: Defining the query
- Step 2: Connecting to the database
- Step 3: Execute the query
- Step 4i, (SELECT): Analyze the result
- Step 4ii, or (UPDATE): Commit the change
- Step 5: Cleaning up; close the database connection

Nice tutorial: http://sebastianraschka.com/Articles/2014_sqlite_in_python_tutorial.html

Over all: Using Python3

Making Useful Strings

A concatenated string

Note the 'f' before the quotes for formatting

```
myCollege_str = "Allegheny"
mesg_str = f"I go to {myCollege_str }!!"
print(mesg_str)
```

```
myCollege_str = "Allegheny"
myMajor_str = "CompSci"
mesg_str = f"At {myCollege_str},  my major is {myMajor_str}"

print(mesg_str)
```

Adding quotes

```
iSay_str = "Cool"
mesg_str = f"I say my major is very a \"{iSay_str}\" major"
print(mesg_str)
```

Making Useful Strings

A concatenated string

- Queries are strings of code that can be created by Python.
- These queries can be sent to database management software

Making a Query Statement

```
a1_str = "deptName"
a2_str = "course"
name_str = "Miller"
table_str = "Instructor"
```

```
myQuery_str = f"SELECT {a1_str}, {a2_str} FROM {table_str} WHERE name == \"{name_str}\""
```

```
print(myQuery_str)
```

Making Useful Strings

A concatenated string

Making An Insert Statement

```
myTable = "Instructor"
PersonID = "10101"
name_str = "Miller"
student = "S1"
```

```
insert_str = f"INSERT INTO {myTable} VALUES({PersonID}, \"{name_str}\", \"{student}\")"
```

```
print(insert_str)
```

Python to manage database

Let's Try It Out!

- Locate the sandbox database builder file `sandbox/campusDB_build.txt` and make your DB.
- Find Python source, `sandbox/simpleQueries/simpleQuery1.py`