Introduction to Database Systems: CS312 SQLite and Python Primer

Oliver Bonham-Carter

20 Feb 2019

Exam1: Friday 22^{nd} during lab (2:30pm)

By Honor code: you must be in Alden hall to take the exam

Some of what to know

- Quiz 1 material
- Queries and code
- Given an output and table, determine the query code
- Given a table and a query, determine the output
- Code to perform types of updates
- Python's `sqlite3` library
  - Querying a database using Python: Concepts
  - Why are automated queries necessary?
  - (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	–

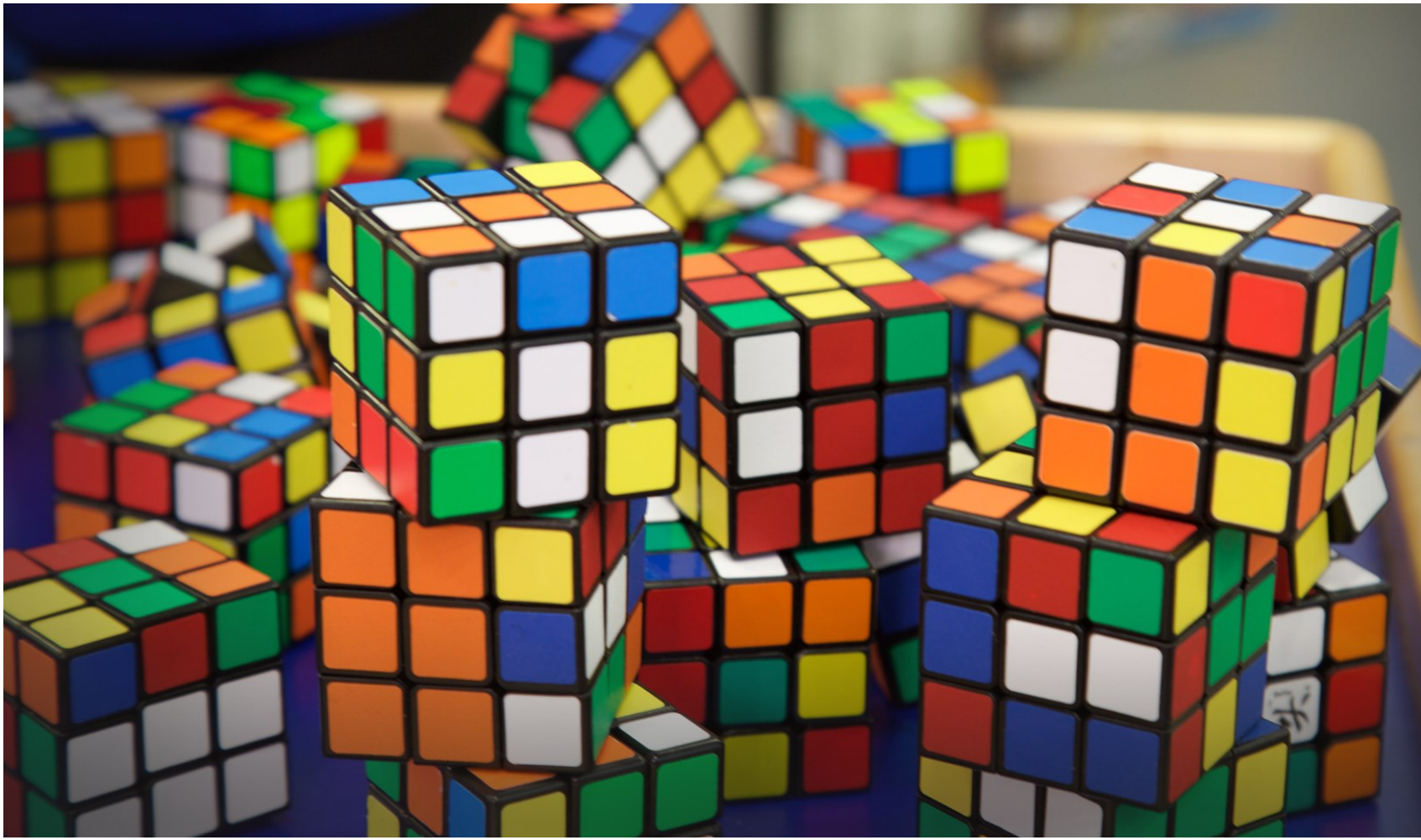
Orders of magnitude of data

- 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
- `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)

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 Python2

Making Useful Strings
Python3

- Creating queries as strings
- Write a correct SQL statement, stored as a Python string, include no semicolon
- ex: sqlCommand = "SELECT attrib1 FROM table"

Making Strings

```
quote = "'"
myString = "this " + quote + "is" + quote + " cool"
print(myString)
```

Query Strings: Note the added spaces and quotes

```
quote = "'"
myQuery = "SELECT * FROM Instructor
WHERE name ==" + quote + "Miller" + quote
print(myQuery)
```

Concatenating Strings

A concatenated string with substituted values and added quotes

UGH! Too much to concatenate!

```
PersonID = "10101"
name = "Miller"
student = "S1"
deptName = "CompSci"
salary =  95000.00
quote = "'"

myInsert1 = "INSERT INTO " +myTable+ " VALUES("
+ quote  + PersonID  + quote +','
+ quote + name      + quote +','
+ quote + student   + quote +','
+ quote + deptName  + quote +','
+ str(salary)
+")"

print(myInsert1) #gives
INSERT INTO Instructor VALUES
('10101','Milder','S1','CompSci',95000.0)
```

Making Useful Strings

A concatenated string with substituted values and added quotes

```
myCollege_str = "Allegheny"
mesg_str = "I go to {A}!!".format(A = myCollege_str)
print(mesg_str)
```

```
myCollege_str = "Allegheny"
myMajor_str = "Computer Science"
mesg_str = "I go to {A} and my major is {B}".
format(A = myCollege_str, B = myMajor_str)
print(mesg_str)
```

Adding quotes

```
theySay_str = "Cool"
mesg_str = "They Say it is a \"{A}\" major".
format(A = theySay_str)
print(mesg_str)
```

Making Useful Strings

Make a less-complicated INSERT string

A string with formatted substitutions

```
PersonID = "10101"
name = "Milder"
student = "S1"
deptName = "CompSci"
salary =  95000.00

#all on one line
myInsert2 = "INSERT INTO instructor VALUES
(\"{A}\",\"{B}\",\"{C}\",\"{D}\",{E})"
.format(A = PersonID, B = name,
C = student, D = deptName, E = salary)

print(myInsert2)
```

Python to manage database

Let's Try It Out!

- Locate the sandbox database builder file `sandbox/campusDB_build.txt` and make your DB.
- Test-out writing Python code to perform Sqlite functions