CS 250
sqlite3 in python

Benjamin Dicken

sqlite3

- **sqlite3** is a python module which we can use to connect to a SQLite database
- To use the module, you must first create a Connection object that represents the database

sqlite3

- To connect to a database named **dbname** with bash

```
$ sqlite sqlite3
```

- To connect to a database named **dbname** with sqlite3 in python

```
import sqlite3
conn = sqlite3.connect('dbname')
```

sqlite3

- Once you have a Connection, you can create a Cursor object and call its execute() method to perform SQL commands

```
c = conn.cursor()
# Create table
c.execute('''CREATE TABLE character(
   name TEXT, cid INT, pid INT)''')
# Save (commit) the changes
conn.commit()
# Close the database connection
conn.close()
```

sqlite3

- Once you have a Connection, you can create a Cursor object and call its execute() method to perform SQL commands

```
c = conn.cursor()
c.execute('''CREATE TABLE character(
  name TEXT, description TEXT, is_good BOOLEAN,
  appearances INT, cid INT)''')
conn.commit()
conn.close()
```

sqlite3

- Once you have a Connection, you can create a Cursor object and call its execute() method to perform SQL commands

```
c = conn.cursor()
c.execute('''INSERT INTO character
  (name, description, is_good, appearances, cid)
  VALUES (‘Batman’, ‘Rich Hero Dude’, 1, 137, 2)''')
conn.commit()
conn.close()
```

sqlite3

- Can do multiple commands with a single connection cursor

```
c = conn.cursor()
c.execute('''INSERT INTO character
  VALUES ('Batman', 'Rich Hero Dude', 1, 137, 2)''')
c.execute('''INSERT INTO character
  VALUES ('Robin', 'Sidekick to Batman', 1, 137, 2)''')
conn.commit()
conn.close()
```

sqlite3

- The changes you've made are persistent and is available in subsequent sqlite3 sessions:

```
import sqlite3
conn = sqlite3.connect('heromovies')
c = conn.cursor()
query_res = c.execute('SELECT * FROM character'):
for row in q_res:
    print(row)
```

sqlite3

- We often want to use values from within our python program in a query to the SQL database
- This can be done using parameter substitution

sqlite3

```
conn = sqlite3.connect('heromovies')
c = conn.cursor()
params = ('Batman',)
query_res = c.execute(
  'SELECT * FROM character WHERE name = ?', params):
for row in q_res:
    print(row)
```

sqlite3

```
params = ('Batman', 'Hero from another planet')
query_res = c.execute(
  '''SELECT * FROM character
     WHERE name = ?
     OR description = ?''', params):
for row in q_res:
    print(row)
```

sqlite3

```
params = ('Batman', 'Hero from another planet', 3)
query_res = c.execute(
  '''SELECT * FROM character
     WHERE name = ?,
     OR description = ?,
     OR cid = ?''', params):
for row in q_res:
    print(row)
```

In-Class Programming

- Re-Implement role.py and addrole.py using sqlite3
- Re-Implement the thesaurus python program using sqlite3 instead of a text-file database

sqlite3

- **Required Materials**
  - docs.python.org/3/library/sqlite3.html