CS 327E Lecture 7

Shirley Cohen

October 5, 2016

Announcements

- Reminder: Lab 2 work next week
- Lab 2 specs & grading rubric: Monday
- Lab 2 setup instructions: <u>http://tinyurl.com/hymam9a</u>
- Format of final exam

Homework for Today

• Chapter 3 from the <u>Data Wrangling</u> book

Which of the following data formats is **not** covered in the assigned chapter for today?

- A. CSV
- B. TSV
- C. JSON
- D. YAML
- E. XML

```
"Indicator":"Life expectancy at birth (years)",
"PUBLISH STATES":"Published",
"Year":1990,
"WHO region":"Europe",
"World Bank income group":"High-income",
"Country":"Andorra",
"Sex":"Both sexes",
```

The sample data shown above is in _____ format:

- A. CSV
- B. TSV
- C. JSON
- D. SQL

E. XML

```
<xs:element name="xaxis">
  <xs:complexType>
    <xs:sequence>
        <xs:element ref="property"/>
        <xs:element ref="value"/>
        <xs:element name="unit">
            <xs:element name="unit">
            <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="nm"/>
            <xs:enumeration value="mm"/>
            <xs:enumeration value="mm"/>
            <xs:enumeration value="mm"/>
```

The sample data shown above is in _____ format:

- A. CSV
- B. TSV
- C. JSON
- D. SQL

E. XML

Src,Eqid,Version,Datetime,Lat,Lon,Magnitude, ak,10654594,1,"Tuesday, February 12, 2013 09 ak,10654587,1,"Tuesday, February 12, 2013 08 us,c000f5w2,4,"Tuesday, February 12, 2013 08 ak,10654581,1,"Tuesday, February 12, 2013 08 ak,10654575,1,"Tuesday, February 12, 2013 08 nc,71935890,0,"Tuesday, February 12, 2013 07 nn,00402618,9,"Tuesday, February 12, 2013 07

The sample data shown above is in _____ format.

- A. CSV
- B. TSV
- C. JSON
- D. SQL
- E. XML

Integrating with Python

- Python support for MySQL not build it. To interact with MySQL from Python, use a library called a "connector"
- PyMySQL connector: <u>http://pymysql.readthedocs.io/en/latest/index.html</u>
- Install PyMySQL through pip: pip2 install pymysql
- Assumes existing Python 2.7 installation:

python -V

pip2 -V

Connection Test

• ۲	connect.py ×	
1	import pymysql	
2		
3	try:	
4	<pre>connect = pymysql.connect(host="127.0.0.1", # hostname</pre>	
5	<i>user</i> ="root", # username	
6	<pre>passwd="cs327e!", # password</pre>	
7	db="utexas") # database name	
8	<pre>cur = connect.cursor()</pre>	
9	<pre>cur.execute("select count(*) from dual")</pre>	
10	<pre>print cur.fetchone()</pre>	
11		
12	except pymysql.Error as error:	
13	print "connect error: ", error	
14		
15	finally:	
16	connect.close()	
17		
(1,) [Fini	shed in 0.9s]	

What can go wrong

< > /	connect.py *	
1	import pymysql	
2		
3	try:	
4	<pre>connect = pymysql.connect(host="128.0.0.1", # hostname</pre>	
5	user="root", # username	
6 7	passwa="cs32/e!", # password	
/	ab= utexas) # database name	
0 0	<pre>cur = connect.cursor() cur evecute("select count(*) from dual")</pre>	
10	npint cur fetchone()	
11		
12	except pymysal.Error as error:	
13	print "connect error: ", error	
14		
15	finally:	
16	<pre>connect.close()</pre>	
17		
conne becau becau Fil c NameE [Fini [shel [dir:	<pre>ct error: (2003, "Can't connect to MySQL server on '128.0.0.1' ([Errno 10060] A connection attempt f se the connected party did not properly respond after a period of time, or established connection fai se connected host has failed to respond)") back (most recent call last): e "C:\utcs_work\cs327e_fall_2016\python\connect.py", line 16, in <module> onnect.close() rror: name 'connect' is not defined shed in 21.7s with exit code 1] l_cmd: python -u "C:\utcs_work\cs327e_fall_2016\python\connect.py"] C:\utcs_work\cs327e_fall_2016\python]</module></pre>	ailed .led

Concept Question 1

What caused this connection error?

- A. Bad host or IP address
- B. Bad username and/or password
- C. Bad db name
- D. Bad SQL query
- E. Any of the above

```
connect.py
     import pymysql
         connect = pymysql.connect(host="127.0.0.1",
                                    user="root",
                                                        # username
                                    passwd="cs327e",
                                                       # password
                                    db="utexas")
         cur = connect.cursor()
         cur.execute("select count(*) from dual")
         print cur.fetchone()
 11
 12
     except pymysql.Error as error:
         print "connect error: ", error
    finally:
         connect.close()
connect error:Traceback (most recent call last):
 File "C:\utcs_work\cs327e_fall_2016\python\connect.py", line 16, in <module>
    connect.close()
```

```
NameError: name 'connect' is not defined
```

(1045, u"Access denied for user 'root'@'localhost' (using password: YES)") [Finished in 0.9s with exit code 1]

```
[shell_cmd: python -u "C:\utcs_work\cs327e_fall_2016\python\connect.py"]
[dir: C:\utcs_work\cs327e_fall_2016\python]
```

[path: C:\ProgramData\Oracle\Java\javapath;C:\Python27;C:\oraclexe\app\oracle\
\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPower

Single Insert

<	insert.py ×								
1	import pymysql								
2									
3	<pre>def create_connection():</pre>								
4	try:								
5	<pre>connection = pymysql.connect(host="127.0.0.1", user="root", passwd="cs327e!", db="utexas")</pre>								
6	return connection								
7									
8	except pymysql.Error as error:								
9	print "connection error: ", error								
10									
11	<pre>def insert():</pre>								
12	try:								
13	<pre>conn = create_connection()</pre>								
14	cur = conn.cursor()								
15	<pre>cur.execute("insert into Student (eid, first_name, last_name, age, dob)" +</pre>								
16	" values ('jpa45', 'Jon', 'Patel', 18, '1998-03-01')")								
17	conn.commit()								
18	destroy_connection(conn)								
19									
20	except pymysql.Error as error:								
21	print "insert error: ", error								
22									
23	def destroy_connection(conn):								
24	<pre>conn.close()</pre>								
25									
26	insert()								
[Fini	ished in 0.8s]								

What can go wrong

```
◀ ▶
      insert.py
     import pymysql
     def create connection():
              connection = pymysql.connect(host="127.0.0.1", user="root", passwd="cs327e!", db="utexas")
             return connection
         except pymysql.Error as error:
              print "connection error: ", error
 10
     def insert():
 11
 12
             conn = create connection()
 13
             cur = conn.cursor()
 14
              cur.execute("insert into Student (eid, first name, last name, age, dob)" +
 15
                  " values ('jpa45', 'Jon', 'Patel', 18, '03-01-1998')")
 16
 17
              conn.commit()
             destroy connection(conn)
 18
 19
 20
         except pymysql.Error as error:
              print "insert error: ", error
 21
 22
     def destroy_connection(conn):
 23
 24
         conn.close()
 25
     insert()
insert error: (1292, u"Incorrect date value: '03-01-1998' for column 'dob' at row 1")
[Finished in 0.6s]
```

Concept Question 2

What caused this insert to fail?

4 Þ	insert.py ×
1	import pymysql
2	
3	<pre>def create_connection():</pre>
4	try:
5	<pre>connection = pymysql.connect(host="127.0.0.1", user="root", passwd="cs327e!", db="utexas") return connection</pre>
6 7	return connection
8	except pymysal.Error as error:
9	print "connection error: ", error
10	
11	<pre>def insert():</pre>
12	try:
13	<pre>conn = create_connection()</pre>
14	<pre>cur = conn.cursor()</pre>
15	<pre>cur.execute("insert into Student values ('masm33', 'Mary', 'Smith', 19)")</pre>
16	conn.commit()
17	destroy_connection(conn)
18	
19	except pymysql.Error as error:
20	print "insert error: ", error
21	def destroy connection(conn):
23	
24	
25	insert()
inser	rt error: (1136, u"Column count doesn't match value count at row 1")
[Fini	ished in 0.6s]

A. Duplicate record

C. Invalid connection or cursor object

B. Insufficient values

- D. Internal MySQL error
- E. None of the above

Multiple Inserts

	Α	В	С	D	E	F			
1	First Name	Last Name	Full Name	EID	AGE	DOB			
2	Maria	Reid	Maria Reid	mna34	18	01/01/98			
3	Allison	Chantelle	Allison Chantelle	acr587	18	02/12/98			
4	Francis	Shi	Francis Shi	fos47	18	03/03/98			
5	Oswald	Jia	Oswald Jia	jso3728	17	01/04/99			
6	Jamie	Hitch	Jamie Hitch	jh943	18	01/05/98			
23 24 25 26 27 28	<pre>a def import_csv(): insert_prefix = "insert into Student (first_name, last_name, eid, age, dob) values (" try: csvfile = open("student.csv", "rb")</pre>								
29 30 31 32 33 34 35 36	<pre>reader = csv.reader(csvfile) for i, row in enumerate(reader): if i == 0: continue insert_stmt = insert_prefix for j, val in enumerate(row): if j == 0 or j == 1 or j == 3: insert_stmt = """</pre>								
36 37 38 39 40 41 42 43 44 45	<pre>elif j == 2: continue elif j == 4:</pre>								
46 47	except IO	Error as e: : "IO Error: <u>" +</u>	e.strerror						

What can go wrong

	Α	В	С	D	E	F	
1	First Name Last Name		Full Name	EID	AGE	DOB	
2	Maria Reid		Maria Reid	mna34	18	01/01/98	
3	Allison	Chantelle	Allison Chantelle	acr587	18	02/12/98	
4	Francis	Shi	Francis Shi	fos47	18	03/03/98	
5	Oswald	Jia	Oswald Jia	jso3728	17	01/04/99	
6	Jamie	Hitch	Jamie Hitch	jh943	18	01/05/98	
7	Amy	Krizovensky	Amy Krizovensky	amk466	18	03/06/98	
23 24 25 26 27 28 29 30 31 32 33 34 35 34 35 36 37 38 39 40	<pre>23 v def import_csv(): 24 25 25 27 v try: 28 29 29 29 29 29 29 29 29 29 20 20 20 20 20 20 20 20 20 20</pre>						
41 42 43 44 45 46 47	<pre>ellt j == 4: insert_stmt += val + ", " else: insert_stmt += "str_to_date('" + val + "','%m/%d/%Y')" insert_stmt += ")" run_insert(insert_stmt)</pre>						
inser inser	nsert error: (1062, u Duplicate entry mnas4 for key PRIMARY) nsert error: (1062, u"Duplicate entry 'acr587' for key 'PRIMARY'") nsert error: (1062, u"Duplicate entry 'fos47' for key 'DDIMADY'")						

Delete Statements

Option 1:

DELETE FROM T

e.g. DELETE FROM Student

Option 2:

```
DELETE FROM T WHERE c_0 = v_0
```

e.g. DELETE FROM Student WHERE eid = 'mna34'

Option 3:

DELETE FROM T WHERE (SELECT * FROM T')

e.g. DELETE FROM Current_Student WHERE (SELECT * FROM Archived Student)

Note: T <> T'

Concept Question 3

Suppose we modify the PK in the Student table. Instead of the EID, we use an AUTO_INCREMENT column as the PK. What problem can arise from using a surrogate key?



- A. Surrogate keys are less descriptive
- B. "Hidden" duplicate records

C. Can't reset an AUTO_INCREMENT column D. None of the above

Concept Question 4

Can we make this code run more efficiently? How so?

	A	В	С	D	E	F			
1	First Name	Last Name	Full Name	EID	AGE	DOB			
2	Maria Reid Maria Reid mr		mna34	18	01/01/98				
3	Allison Chantelle Allison Chantelle acr587				18	02/12/98			
4	Francis	Shi	Francis Shi	fos47	18	03/03/98			
22	Occurrent		Onveld the	: 2720	17	01/04/00			
23 24 25 26	3								
27 28 29	<pre>try: csvfile = open("student.csv", "rb") csvfile = open("student.csv", "rb")</pre>								
30	for i	, row in enumera	ate(reader):		aej	<pre>run_insert(insert)</pre>	ert_stmt):		
31 32 33	<pre>if i == 0: continue insert_stmt = insert_prefix for i well in enumerate(reader). insert_stmt = insert_prefix for i well in enumerate(reader). conn = create_connection() cur = conn.cursor() cur.execute(insert_stmt) conn.commit()</pre>								
34	for j, val in enumerate(row): if i == 0 on i == 1 on i == 3: destroy_connection(conn)								
36		insert st	tmt += "'" + val + "'	. "			F		
37		elif j == 2:		·		except pymysql print "inse	ert error: ". error		
38		continue				prine ino.			
39	elif j == 4:								
40	<pre>insert_stmt += val + ", "</pre>								
41		else:							
42		insert_st	tmt += "str_to_date('	" + val + "	','%m/%d/%Y	(')"			
43	1	nsert_stmt += "))"						
44	r	un_insert(insert	_stmt)						

- A. Reuse the connection
- B. Commit inserts in batches

C. Remove print statements D. All of the above

Inserts with FKs

	А	В	С	D	E	F	G	Н
1	First Name	Last Name	Full Name	EID	AGE	DOB	SSN	STATE
2	Maria	Reid	Maria Reid	mna34	18	1/1/1998	666666666	тх
3	Allison	Chantelle	Allison Chantelle	acr587	18	2/12/1998	555555555	ТХ
4	Francis	Shi	Francis Shi	fos47	18	3/3/1998		
5	Oswald	Jia	Oswald Jia	jso3728	17	1/4/1999		
23 24	<i>def</i> import_cs	sv():				. 1= 1	⁴⁴	CA
25 26 27	insert_pr	refix = "insert i	nto Domestic_Studen	t (eid, ssn	, stat	e) values ("		PA
28 29	27 try: 28 csvfile = open("student_detail.csv", "rb") 29 reader = csv.reader(csvfile)							
30 - 31	<pre>0 for i, row in enumerate(reader): 1 if i == 0: continue</pre>							
32	insert_stmt = insert_prefix							
33 34	3 for i val in enumerate(row):							
35	5 is_domestic_student = True 77 WA							
36 37 38	36 37 if j == 0 or j == 1 or j == 2 or j == 4 or j == 5: 38 continue							
39		elif j == 6 a	ind val == "":			12 def run	insert(inser	t stmt).
40 41		is_domest break	:ic_student = False			13 try	:	<u> </u>
42		elif j == 3 c	or j == 6:			14	conn = creat	e_connection()
43	insert_stmt += "'" + val + "', " 15 cur = conn.cursor()							
44		elif j == 7:				16	cur.execute(insert_stmt)
45		insert_st	:mt += "'" + val + "			18	destroy corp) action(conn)
40		f is domestic st	udent is True:			19	descroy_conn	
48		insert stmt	-= ")"			20 <u>exc</u>	ept pymysal.E	rror as error:
49		run_insert(in	sert_stmt)			21	print "inser	t error: ", error

What can go wrong

```
∢ ►
      import_csv_domestic_student.py >>
 23 ▼ def import csv():
 24
         insert_prefix = "insert into Domestic_Student (eid, ssn, state) values ("
 27 🔻
             csvfile = open("student_detail.csv", "rb")
 28
             reader = csv.reader(csvfile)
             for i, row in enumerate(reader):
 30 V
                 if i == 0: continue
                 insert_stmt = insert_prefix
                 for j, val in enumerate(row):
 34 🔻
                      is_domestic_student = True
                     if j == 0 or j == 1 or j == 2 or j == 4 or j == 5:
                     elif i == 6 and val == "":
 39 🔻
                          is domestic student = False
                          break
                     elif j == 3 or j == 6:
                          insert_stmt += "'" + val + "', "
 44
                     elif j == 7:
                          insert_stmt += "'" + val + "'"
                 if is_domestic_student is True:
 47 🔻
                      insert stmt += ")"
                     run insert(insert_stmt)
insert error: (1452, u'Cannot add or update a child row: a foreign key constraint fails
(`utexas`.`domestic student`, CONSTRAINT `domestic_student_ibfk_1` FOREIGN KEY (`eid`) REFERENCES `student` (`eid`)),
[Finished in 0.9s]
```

Final Remarks

- Avoid using surrogate keys. If you have no choice, check for duplicate records by manually inspecting the data. We will learn a more efficient way to do this when we cover the GROUP-BY clause
- Read the API docs for PyMySQL: <u>http://pymysql.readthedocs.io/en/latest/index.html</u>
- PyMySQL sample code available in our snippets repo
- Please setup your environment prior to Monday's class (and if you get stuck, post the issue on Piazza)