Laboratory Tutorial 7: DB-API

In this laboratory tutorial you will use Python to connect to a database; create tables, use SELECT, UPDATE, DELETE AND INSERT statements.

Preamble

In previous tutorials you gained experience in using some DML commands. **We will continue to use DB Browser for SQLite.**

In this lab tutorial you will use Python to connect to a database to create some tables and perform some tasks.

We will refer to a small scenario – Driving Lessons. Based on this scenario, we will create a database which consists of three tables along with primary keys and foreign keys.

An instructor provides driving lessons to one or many learners. Each learner is taught by one instructor.

A learner takes one or many lessons. Each lesson is taken by one learner.

Exercise 7.1: SQLite Connection and Cursor objects

Before closing a connection, where applicable, make sure to commit, otherwise changes made will not be saved.

Open Python Jupyter notebook and create a new file called labws7.

Next, type the following:

import sqlite3

con=sqlite3.connect(r"C:\Users\user\APISQL\labws7.db")

Note: you can create your own path, the above is mine on my laptop.

Click Run.

Labws7.db does not exist, but it will be created in the directory where you want it to reside.

Now, open DB Browser for SQLite, and open the labws7.db.

****

Next, we will obtain a Cursor object to execute transactions against the database.

Type the following in jupyter:

cursor=con.cursor()

cursor.execute('''CREATE TABLE Instructor(id INTEGER PRIMARY KEY NOT NULL,name TEXT,address TEXT)''')

con.commit()

Click Run. A table named Instructor is now created in SQLite. Go and check. It might take a while.

TASK:
In Jupiter write the commands to insert some values (refer below) in the Instructor Table.

Make sure you commit when you are done.

Instructor

id	name	address
1001	JS	Sheffield
1002	MN	Leeds
1003	SK	Derby

Your table should now have some records. Check in SQLite.

TASK:

You are required to write the code to create the table for Learner with some values (refer to the table below). Don't forget to include the foreign keys.

Note: as part of your commands, you need to include the following - UPDATE CASCADE, DELETE CASCADE, INSERT CASCADE - in case you need to make some adjustments to your table.

Learner

learnerNo	name	address	age	id
1	TK	York	19	1001
2	KS	Newcastle	20	1002
3	FB	Derby	19	1001
4	MD	York	18	1003

TASK:

You need to write the code to create the third and final table – Lesson, with some values (refer to the table below). Here also, you will need - UPDATE CASCADE, DELETE CASCADE, INSERT CASCADE – and foreign keys.

Lesson

lessonID	date	learnerNo
111	23-10-20	1
222	25-10-20	2
333	05-11-20	1
444	27-10-20	3
555	29-10-20	4

You should now have three tables in SQLite.

Exercise 7.2: UPDATE, DELETE and INSERT Statements

Make sure that your Python Jupyter notebook file (labws7) is still open.

Before closing a connection, where applicable, make sure to commit, otherwise changes made will not be saved.

We are going to an insert a new instructor in the Instructor table, with ID=1004,name=SG, and address = Hull.

TASK:
Write the commands to accomplish the above.

Now click Run. Your table should be updated.

****
Next, we are going to change the id of one of the instructors. Current id = 1001, and needs to be changed to 999. To do this, we will type the following commands.

```
con.execute("PRAGMA foreign_ keys=ON")
cursor.execute("UPDATE Instructor SET id=999 WHERE id=1001")
con.commit()
```

Click Run. All corresponding tables should be updated.
****

Another method:

```
VALUES=(999,1001)
cursor.execute("UPDATE Instructor SET id=? WHERE id=?", VALUES)
con.commit()
```

****

TASK:
You need to delete a learner, with learnerNo = 3. Write your commands to perform this change and check if your table has been updated in the DB browser.

Exercise 7.3: Use of fetch

Make sure that your Python Jupyter notebook file (labws7) is still open.

Before closing a connection, where applicable, make sure to commit, otherwise changes made will not be saved.

We are now going to use Python's cursor class methods to retrieve rows from a table.

TASK:
How would you use the fetchall() method to retrieve all attributes and records from Instructor table?

You should get a result like this:

```
888 SG Hull
999 JS Sheffield
1002 MN Leeds
1003 SK Derby
```

****

TASK:
How would you use the fetchmany() method to display the first two records from the Learner table?

You should get a result like this:

1 TK York 20 999
2 KS Newcastle 19 1002

Summary and future work

In this tutorial you have learned how to use python to connect to a database and perform some tasks. Specifically, you have used the connection object, cursor object, execute() method, parameterised SQL, fetch method, and close method.

Further Reading

Databases Illuminated (Chapter 5, Section 5.8).