

Database and Programming
Project SQLite 8-9
2017.04.25



Lillebaelt Academy of
Professional Higher Education

Author:
Milan Kristof Vince:
mila1025@edu.eal.dk

April, 25, 2017

Table of Contents

Introduction	3
8. How to use SQLite with a Python application	3
9. Update	5
Conclusion.....	6

Introduction

The document contains the eight and ninth exercise of the Project SQLite assignment and their solution. The audience is my lecturer and any of my classmates who is interested in to check. SQLite is a relational database management system. In contrast to many other database management systems, SQLite is not a client–server database engine. It has an embedded SQL database engine. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

8. How to use SQLite with a Python application

- Document how to integrate SQLite3 into a Python program.
- Document and show a working Python example where a customerTable as shown here is created.

customerTable				
idCust	name	email	address	city
1	Per	pda@eal.dk	Mystreet 1	Odense
2	Artur	at@hotmail.com	Allstreet 741	Vilnius
3	Alice	ab@gmail.com	Topstreet 56	London

” Show how to insert and retrieve data into and from the table by means of Python.

Python Code

```
# Import SQLite3.
import sqlite3

# Create a connection to the database.
conn=sqlite3.connect('assignment89.db')

# Create "Cursor" object.
cursor = conn.cursor()

# Create a table and call it 'customerTable'.
cursor.execute("""CREATE TABLE customerTable
    ( idCust integer,
      name varchar(50),
      email varchar(50),
      address varchar(60),
      city varchar(50))
    """)

# "cursor.execute" is a method which calls the object to get used SQL
commands.
```

```

# Insert data into the 'customerTable'.
cursor.execute("""
INSERT INTO customerTable (idCust, name, email,address, city)
VALUES (1, 'Per', 'pda@eal.dk', 'MyStreet 1', 'Odense')
""")

cursor.execute("""
INSERT INTO customerTable (idCust, name, email,address, city)
VALUES (2, 'Giovanni', 'gov@gmail.com', 'AllStreet 2', 'Aalborg')
""")

cursor.execute("""
INSERT INTO customerTable (idCust, name, email,address, city)
VALUES (3, 'Jesper', 'jes@hotmail.com', 'WestStreet 2', 'London')
""")

# Save (commit) the changes.
conn.commit()
print("You have just stored some data in the database")

# Recover data from the table.
print("\n\nYour current data in the database: ")
cursor.execute('SELECT * FROM customerTable')

for row in cursor.fetchall():
    print("\n", end='')

    for field in row:
        print(field, ' ',end='')

# Close the cursor.
cursor.close()

# Close the connection.
conn.close()

```

Output

```

You have just stored some data in the database

Your current data in the database:

1 Per pda@eal.dk MyStreet 1 Odense
2 Giovanni gov@gmail.com AllStreet 2 Aalborg
3 Jesper jes@hotmail.com WestStreet 2 London
>>> |

```

Just for verify if the user wants to check it is saved, just open SQLite3 in command prompt to show the table:

```
C:\>sqlite3 assignment89.db
SQLite version 3.18.0 2017-03-28 18:48:43
Enter ".help" for usage hints.
sqlite> SELECT * FROM customerTable;
1|Per|pda@eal.dk|MyStreet 1|Odense
2|Giovanni|gov@gmail.com|AllStreet 2|Aalborg
3|Jesper|jes@hotmail.com|WestStreet 2|London
```

9. Update

”Show how to do UPDATE of already entered data rows in Python.”

Python Code

```
# Import SQLite3.
import sqlite3

# Create a connection to the database.
conn=sqlite3.connect('assignment89.db')

# Create "Cursor" object.
cursor = conn.cursor()

# Update city for Giovanni.
cursor.execute('UPDATE customerTable set city = ? WHERE idCust = ?',
               ('Kolding', 2))

# Update email for Jesper.
cursor.execute('UPDATE customerTable set email = ? WHERE idCust = ?',
               ('test123@edu.eal.dk', 3))

# Save (commit) the changes.
conn.commit()
print('You have just updated the database.')

# Recover data from the table.
print("\n\nCurrent data in the database: ")
cursor.execute('SELECT * FROM customerTable')

for row in cursor.fetchall():
    print("\n", end='')

    for field in row:
        print(field, ' ', end='')
```

```
# Close the cursor
cursor.close()
# Close the connection.
conn.close()
```

Output

```
You have just updated the database.
```

```
Current data in the database:
```

```
1 Per pda@eal.dk MyStreet 1 Odense
2 Giovanni gov@gmail.com AllStreet 2 Kolding
3 Jesper test123@edu.eal.dk WestStreet 2 London
```

Just to verify that the table was successfully updated the user can use SQLite3 in command prompt to check:

```
C:\>sqlite3 assignment89.db
SQLite version 3.18.0 2017-03-28 18:48:43
Enter ".help" for usage hints.
sqlite> SELECT * FROM customerTable;
1|Per|pda@eal.dk|MyStreet 1|Odense
2|Giovanni|gov@gmail.com|AllStreet 2|Kolding
3|Jesper|test123@edu.eal.dk|WestStreet 2|London
sqlite>
```

Conclusion

This exercise was interesting I like the way how easily can use SQLite 3 with Python. I like to work with databases. ☺