Ø«»

0 : Python and Databases

2011-07-20

Python and Databases

Python and Databases

Vimalkumar Velayudhan

July 20, 2011

Python modules for working with databases

- sqlite3 included by default in Python (version > 2.5)
- pyodbc - Any ODBC compatible Database
- psycopg2 - PostgreSQL
- MySQLdb - MySQL
- Others - http://wiki.python.org/moin/DbApiModuleComparison

SQLite

- Embedded relational database management system
- Most widely deployed database engine
- No setup or administration involved
- Database stored as a file in the filesystem
- Useful for testing and embedding in GUI/Web applications
- http://www.sqlite.org/

pyodbc

- Uses ODBC (Open Database Connectivity) to connect to databases
- Cross-platform (Windows, Linux, OS/X..)
- Supports a number of databases - PostgreSQL, MySQL, Microsoft SQL Server, Microsoft Access
- http://code.google.com/p/pyodbc

Client software

To access databases on a database management system

sqlite3
SQLiteBrowser - http://sqlitebrowser.sourceforge.net/
SQLite Manager Firefox Addon - http://sqlite-manager.googlecode.com/

PostgreSQL
pgAdmin - http://www.pgadmin.org/

Microsoft SQL Server
SQL Server Management Studio - http://www.microsoft.com/express

Create and connect to a database

```
#import the builtin sqlite3 module
import sqlite3

#create or connect to an existing database
conn = sqlite3.connect("db.sqlite")

#get a cursor to work with the database
cursor = conn.cursor()

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

example1.py

Create a table

```
import sqlite3
conn = sqlite3.connect("db.sqlite")
cursor = conn.cursor()

#create a table repeats with 4 fields - key, gene, orientation, status
sql = "create table repeats(key text, gene text, orientation text, status text)"
cursor.execute(sql)

#commit the changes
conn.commit()
conn.close()
```

example2.py

Inserting a row

```
import sqlite3
conn = sqlite3.connect("db.sqlite")
cursor = conn.cursor()

#data supplied as tuple. ? - placeholders for data
sql = "insert into repeats values (?, ?, ?, ?)"
data = ("M83", "thrA", "F", "Unfinished")
cursor.execute(sql, data)

#commit the changes
conn.commit()
conn.close()
```

example3.py

Inserting multiple rows

```
import sqlite3
conn = sqlite3.connect("db.sqlite")
cursor = conn.cursor()

#data supplied as tuple of tuples. ? - placeholders for data
sql = "insert into repeats values (?, ?, ?, ?)"
data = (("M83", "thrA", "R", "Unfinished"), ("AU123", "adk", "F", "Unfinished"))
cursor.execute(sql, data)

#commit the changes
conn.commit()
conn.close()
```

example4.py

Retrieving data

```
import sqlite3
conn = sqlite3.connect("db.sqlite")
cursor = conn.cursor()

#SQL SELECT statement
sql = "select * from repeats"
cursor.execute(sql)

#iterate over the results and print them
for row in cursor.fetchall():
    print row

conn.close()
```

example5.py

Updating data

```
import sqlite3
conn = sqlite3.connect("db.sqlite")
cursor = conn.cursor()
```

```
#SQL SELECT statement
sql = "update repeats set status=? where key=? and gene=? and orientation=?"
data = ("Finished", "M83", "thrA", "F")
cursor.execute(sql, data)

#commit the changes
conn.commit()

conn.close()
```

example6.py

Example using pyodbc

```
Create a table, insert and update data

"""
import pyodbc

dsn = 'Driver={PostgreSQL ANSI};Server=localhost;Port=5432;' \
                'Database=test;Uid=test;Pwd=test;'

#connect to the database
conn = pyodbc.connect(dsn)

#get a cursor to work with
cursor = conn.cursor()

#create table
sql = "create table repeats(key text, gene text, orientation text, status text)"
cursor.execute(sql)

#insert a row
sql = "insert into repeats values (?, ?, ?, ?)"
data = ("M83", "thrA", "F", "Unfinished")
cursor.execute(sql, data)

#insert multiple rows
data = (("M83", "thrA", "R", "Unfinished"), ("AU123", "adk", "F", "Unfinished"))
cursor.executemany(sql, data)
```

```
#update data
sql = "update repeats set status=? where key=? and gene=? and orientation=?"
data = ("Finished", "M83", "thrA", "F")
cursor.execute(sql, data)

#commit the changes
conn.commit()

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

example7.py