

```
1 Lab. Python Database Programming
2
3 1. SQLite3
4 1)Python에서는 SQLite3에 대해 외부 module을 설치하지 않아도 된다.
5 2)Python SQLite3 참조 문서
6 https://goo.gl/8Ghx9r
7 3)https://sqlite.org/index.html
8 -Windows sqlite3
9 -sqlite-tools-win32-x86-xxx.zip
10
11 4)makedb.py
12 import sqlite3
13
14 conn = sqlite3.connect('myaddress.db')
15 cursor = conn.cursor()
16
17 cursor.execute('DROP TABLE IF EXISTS tblAddress')
18 cursor.execute("""CREATE TABLE tblAddress
19                 (name TEXT PRIMARY KEY, phone TEXT, addr TEXT)""")
20 cursor.execute("INSERT INTO tblAddress VALUES ('한지민', '123-4567', '오산')")
21 cursor.execute("INSERT INTO tblAddress VALUES ('이미자', '777-7777', '부산')")
22 cursor.execute("INSERT INTO tblAddress VALUES ('설운도', '888-8888', '목포')")
23
24 conn.commit()
25
26 cursor.close()
27 conn.close()
28
29 5)selectdb.py
30 import sqlite3
31
32 conn = sqlite3.connect('myaddress.db')
33 cursor = conn.cursor()
34
35 cursor.execute('SELECT * FROM tblAddress')
36 table = cursor.fetchall()
37 for record in table:
38     print('이름:%s, 전화:%s, 주소:%s' % record)
39
40 cursor.close()
41 conn.close()
42
43 6)selectdb1.py
44 import sqlite3
45
46 conn = sqlite3.connect('myaddress.db')
47 cursor = conn.cursor()
48
49 cursor.execute('SELECT * FROM tblAddress')
50 while True:
51     record = cursor.fetchone()
```

```
52         if record == None:
53             break
54         print('이름:%s, 전화:%s, 주소:%s' % record)
55
56     cursor.close()
57     conn.close()
58
59 7)DB Browser for SQLite
60   -https://sqlitebrowser.org/
61
62 8)sqllitedemo.py
63     import sqlite3
64
65     conn = sqlite3.connect('test.db')
66     cursor = conn.cursor()
67
68     cursor.execute("""CREATE TABLE IF NOT EXISTS supermarket
69         (Itemno INTEGER, Category TEXT, FoodName TEXT,
70         Company TEXT, Price INTEGER)""")
71
72     sql = "DELETE FROM supermarket"
73     cursor.execute(sql)
74
75     sql = """INSERT INTO supermarket(Itemno, Category, FoodName, Company, Price)
76         VALUES(?,?,?,?,?)"""
77     cursor.execute(sql, (1, '과일', '자몽', '마트', 1500))
78
79     sql = """INSERT INTO supermarket(Itemno, Category, FoodName, Company, Price)
80         VALUES(?,?,?,?,?)"""
81     cursor.execute(sql, (2, '음료수', '망고쥬스', '편의점', 1000))
82
83     conn.commit()
84
85     sql = """SELECT Itemno, Category, FoodName, Company, Price FROM
86         supermarket"""
87     cursor.execute(sql)
88
89     while True:
90         row = cursor.fetchone()
91         if row == None:
92             break
93         print(str(row[0]) + " " + str(row[1]) + " " + str(row[2]) + " " + str(row[3]) + " " +
94             str(row[4]))
95
96     cursor.close()
97     conn.close()
98
99 2. MySQL / MariaDB
100 1)cmd as Administrator
101    -If you're used to using the "Run" box to open apps, you can use that to launch
```

```

102     Command Prompt with admin privileges.
103     -Press Windows+R to open the "Run" box.
104     -Type "cmd" into the box and then press Ctrl+Shift+Enter to run the command as an
105     administrator.
106
107 2)pip install PyMySQL
108
109 3)mariadb.py
110     import pymysql
111
112     # Open database connection
113     db = pymysql.connect(host='localhost', port=3306, user='root', passwd='maria',
114     db='estdb',charset='utf8',autocommit=True)
115
116     # prepare a cursor object using cursor() method
117     cursor = db.cursor()
118
119     # execute SQL query using execute() method.
120     cursor.execute("SELECT VERSION()")
121
122     # Fetch a single row using fetchone() method.
123     data = cursor.fetchone()
124     print ("Database version : %s " % data)
125
126     # disconnect from server
127     db.close()
128
129 4)Create Database mytest;
130 5)use mytest;
131 6)CREATE TABLE supermarket(
132     Itemno INT NULL,
133     Category CHAR(20) NULL,
134     FoodName CHAR(30) NULL,
135     Company CHAR(20) NULL,
136     Price INT NULL);
137 7)INSERT INTO supermarket VALUES(1, '과일', '자몽', '마트', 1500)
138     INSERT INTO supermarket VALUES(2, '음료수', '망고주스', '편의점', 1000)
139     INSERT INTO supermarket VALUES(3, '음료수', '식혜', '시장', 1000)
140     INSERT INTO supermarket VALUES(4, '과자', '머랭', '조각케익가게', 3000)
141 8)SELECT * FROM supermarket;
142
143 9)mariadb1.py
144     import pymysql
145
146     server = 'localhost'
147     user = 'root'
148     password = 'pythonmariadb'
149     dbname = 'mytest'
150
151     conn = pymysql.connect(server, user, password, dbname, charset='utf8')
```

```
150     cursor = conn.cursor()
151
152     cursor.execute('SELECT * FROM supermarket;')
153
154     row = cursor.fetchone()
155
156     while row:
157         print(str(row[0]) + " " + str(row[1]) + " " + str(row[2]) + " " + str(row[3]) + " " +
158               str(row[4]))
159         row = cursor.fetchone()
160
161     conn.close()
162
163 10)mariadb2.py
164     import pymysql
165
166     # Open database connection
167     db = pymysql.connect("localhost","root","pythonmariadb","test" )
168
169     # prepare a cursor object using cursor() method
170     cursor = db.cursor()
171
172     # Drop table if it already exist using execute() method.
173     cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
174
175     # Create table as per requirement
176     sql = """CREATE TABLE EMPLOYEE (
177             FIRST_NAME CHAR(20) NOT NULL,
178             LAST_NAME CHAR(20),
179             AGE INT,
180             SEX CHAR(1),
181             INCOME FLOAT )"""
182
183     cursor.execute(sql)
184
185     # disconnect from server
186     db.close()
187
188 11)mariadb3.py
189     import pymysql
190
191     # Open database connection
192     db = pymysql.connect("localhost","root","pythonmariadb","test" )
193
194     # prepare a cursor object using cursor() method
195     cursor = db.cursor()
196
197     # Prepare SQL query to INSERT a record into the database.
198     sql = """INSERT INTO EMPLOYEE(FIRST_NAME,
199             LAST_NAME, AGE, SEX, INCOME)
200             VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""
```

```
200     try:
201         # Execute the SQL command
202         cursor.execute(sql)
203         # Commit your changes in the database
204         db.commit()
205     except:
206         # Rollback in case there is any error
207         db.rollback()
208
209     # disconnect from server
210     db.close()
211
212 12)mariadb4.py
213     import pymysql
214
215     # Open database connection
216     db = pymysql.connect("localhost","root","pythonmariadb","test" )
217
218     # prepare a cursor object using cursor() method
219     cursor = db.cursor()
220
221     # Prepare SQL query to INSERT a record into the database.
222     sql = "SELECT * FROM EMPLOYEE \
223           WHERE INCOME > '%d'" % (1000)
224     try:
225         # Execute the SQL command
226         cursor.execute(sql)
227         # Fetch all the rows in a list of lists.
228         results = cursor.fetchall()
229         for row in results:
230             fname = row[0]
231             lname = row[1]
232             age = row[2]
233             sex = row[3]
234             income = row[4]
235             # Now print fetched result
236             print ("fname = %s,lname = %s,age = %d,sex = %s,income = %d" % \
237                   (fname, lname, age, sex, income ))
238     except:
239         print ("Error: unable to fetch data")
240
241     # disconnect from server
242     db.close()
243
244 13)mariadb5.py
245     import pymysql
246
247     # Open database connection
248     db = pymysql.connect("localhost","root","pythonmariadb","test" )
249
250     # prepare a cursor object using cursor() method
```

```
251     cursor = db.cursor()
252
253     # Prepare SQL query to UPDATE required records
254     sql = "UPDATE EMPLOYEE SET AGE = AGE + 1
255           WHERE SEX = '%c'" % ('M')
256     try:
257         # Execute the SQL command
258         cursor.execute(sql)
259         # Commit your changes in the database
260         db.commit()
261     except:
262         # Rollback in case there is any error
263         db.rollback()
264
265     # disconnect from server
266     db.close()
267
268 14)mariadb6.py
269     import pymysql
270
271     # Open database connection
272     db = pymysql.connect("localhost","root","pythonmariadb","test" )
273
274     # prepare a cursor object using cursor() method
275     cursor = db.cursor()
276
277     # Prepare SQL query to DELETE required records
278     sql = "DELETE FROM EMPLOYEE WHERE AGE > '%d'" % (20)
279     try:
280         # Execute the SQL command
281         cursor.execute(sql)
282         # Commit your changes in the database
283         db.commit()
284     except:
285         # Rollback in case there is any error
286         db.rollback()
287
288     # disconnect from server
289     db.close()
290
291
292 3. MariaDB World database 이용하기
293 1)World database 다운로드하기
294   -https://dev.mysql.com/doc/index-other.html
295   -Example Databases에서 [World database] 'Zip' link 클릭
296   -다운로드 후 C:/temp에 압축을 풀면 world.sql 파일이 보인다.
297
298 2)MariaDB login한다.
299   mysql -h localhost -u root -p
300
301 3)C:/temp의 world.sql을 실행한다.
```

```
302     MariaDB [(none)]>source C:/temp/world.sql
303
304 4)World database의 table을 확인한다.
305     MariaDB [world]> show tables;
306
307 5)mariadb.py
308     import pymysql
309
310     # Open database connection
311     db = pymysql.connect("localhost","root","pythonmariadb","world" )
312
313     # prepare a cursor object using cursor() method
314     cursor = db.cursor()
315
316     sql = "SELECT ID, Name, CountryCode, District, Population FROM city WHERE
CountryCode='KOR'"
317
318     try:
319         # Execute the SQL command
320         cursor.execute(sql)
321         # Fetch all the rows in a list of lists.
322         results = cursor.fetchall()
323         for row in results:
324             print('ID = %d, Name = %s, CountryCode = %s, District = %s, Popluation = %d' %
(row[0], row[1], row[2], row[3],row[4]))
325     except:
326         print ("Error: unable to fetch data")
327
328     # disconnect from server
329     db.close()
330
331
332 4. mysql-connector-python module 이용하기
333 1)$ install mysql-connector-python
334
335     import mysql.connector as mariadb
336     from mysql.connector import Error
337
338     try:
339         connection = mariadb.connect(user='root', password='pythonmariadb',
host='localhost', database='world')
340         if connection.is_connected() :
341             db_info = connection.get_server_info()
342             print('Connected to MariaDB database. MariaDB Server info is', db_info)
343             cursor = connection.cursor()
344             cursor.execute("SELECT ID, Name, CountryCode, District, Population FROM city
WHERE CountryCode='KOR'")
345             for ID,Name,CountryCode,District,Population in cursor:
346                 print('ID = %d, Name = %s, CountryCode = %s, District = %s, Popluation =
%d' % (ID, Name, CountryCode,District,Population))
347     except Error as e :
```

```
348     print('Error while connecting to MariaDB :', e)
349 finally :
350     #Closing database connection.
351     if connection.is_connected() :
352         cursor.close()
353         connection.close()
354         print("MariaDB connection is closed.")
355
356
357 5. Oracle
358 1)Oracle Database 11g Express Edition Installation
359 2)OracleServiceXE & OracleXETNSListener service start
360 3)HR Schema activation
361     -ALTER USER hr ACCOUNT UNLOCK IDENTIFIED BY hr;
362
363 4)Oracle cx_Oracle 7.2
364 5)cx_Oracle installation
365     $ pip install cx_oracle --upgrade
366
367     -In Anaconda Prompt
368     $ conda install cx_oracle
369
370 6)oracledb.py
371     import cx_Oracle
372
373     conn = cx_Oracle.connect('hr', 'hr', 'localhost:1521/XE')
374     print(conn) #<cx_Oracle.Connection to hr@localhost:1521/XE>
375
376     conn1 = cx_Oracle.connect('hr/hr@localhost:1521/XE')
377     print(conn1) #<cx_Oracle.Connection to hr@localhost:1521/XE>
378
379     dsn_tns = cx_Oracle.makedsn('localhost', 1521, 'XE')
380     print(dsn_tns)
381     #(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNE
CT_DATA=(SID=XE)))
382
383     conn2 = cx_Oracle.connect('hr', 'hr', dsn_tns)
384     print(conn2)
385     #<cx_Oracle.Connection to
hr@\(DESCRIPTION=\(ADDRESS=\(PROTOCOL=TCP\)\(HOST=localhost\)\(PORT=1521\)\)\(CON
NECT\_DATA=\(SID=XE\)\)\)>
386
387     print(conn.version) #11.2.0.2.0
388
389     #Cursor Objects
390     conn3 = cx_Oracle.connect('hr', 'hr', 'localhost:1521/XE')
391     cursor = conn3.cursor()
392     sql = """SELECT employee_id, first_name, salary, to_char(hire_date, 'yyyy-mm-dd'),
department_name, city
393         from employees e inner join departments d on e.department_id = d.department_id
394         inner join locations l on d.location_id = l.location_id"""
```



```
395     cursor.execute(sql)
396
397     for employee_id, first_name, salary, hire_date, department_name, city in cursor :
398         print(employee_id, first_name, salary, hire_date, department_name, city)
399
400     cursor.close()
401
402
403 6. MongoDB
404 1)test 데이터베이스에 supermarket Collection 만들기
405     -Command 창에서 MongoDB Server로 접속
406         $ mongo
407         MongoDB shell version v4.2.0
408         connecting to:
409         mongod://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
410         Implicit session: session { "id" : UUID("b8f7892c-63e2-4491-809b-11a108326b3f") }
411         MongoDB server version: 4.2.0
412         ...
413         >
414 2)test database로 연결
415     > use test
416     switched to db test
417
418     > db    #현재 위치하고 있는 데이터베이스 확인
419     test
420
421     > show dbs    #모든 데이터베이스 보기
422     admin  0.000GB
423     config 0.000GB
424     local  0.000GB
425
426     > db.stats()    #현 test 데이터베이스 상태확인
427     {
428         "db" : "test",
429         "collections" : 0,
430         "views" : 0,
431         "objects" : 0,
432         "avgObjSize" : 0,
433         "dataSize" : 0,
434         "storageSize" : 0,
435         "numExtents" : 0,
436         "indexes" : 0,
437         "indexSize" : 0,
438         "scaleFactor" : 1,
439         "fileSize" : 0,
440         "fsUsedSize" : 0,
441         "fsTotalSize" : 0,
442         "ok" : 1
443     }
444
```

```
445 3)supermarket Collection 생성하기
446 > db.createCollection('supermarket')
447 { "ok" : 1 }
448
449 > show collections #test 데이터베이스의 모든 collection 보기
450 supermarket
451
452 4)데이터 insert
453
454 > db.supermarket.insert([
455 ... {
456 ... Itemno:1,
457 ... Category:'과일',
458 ... FoodName:'자몽',
459 ... Company:'마트',
460 ... Price:1500
461 ... },
462 ... {
463 ... Itemno:2,
464 ... Category:'음료수',
465 ... FoodName:'망고주스',
466 ... Company:'편의점',
467 ... Price:1000
468 ... },
469 ... {
470 ... Itemno:3,
471 ... Category:'음료수',
472 ... FoodName:'식혜',
473 ... Company:'시장',
474 ... Price:1000
475 ... },
476 ... {
477 ... Itemno:4,
478 ... Category:'과자',
479 ... FoodName:'머랭',
480 ... Company:'조각케익가게',
481 ... Price:3000
482 ... }
483 ... ])
484 BulkWriteResult({
485     "writeErrors" : [ ],
486     "writeConcernErrors" : [ ],
487     "nInserted" : 4,
488     "nUpserted" : 0,
489     "nMatched" : 0,
490     "nModified" : 0,
491     "nRemoved" : 0,
492     "upserted" : [ ]
493 })
494
495 5)지금까지 insert한 데이터 모두 select하기
```

```
496
497 > db.supermarket.find().pretty()
498 {
499     "_id" : ObjectId("5d5e3470c8076c145c95fb6f"),
500     "Itemno" : 1,
501     "Category" : "과일",
502     "FoodName" : "자몽",
503     "Company" : "마트",
504     "Price" : 1500
505 }
506 {
507     "_id" : ObjectId("5d5e3470c8076c145c95fb70"),
508     "Itemno" : 2,
509     "Category" : "음료수",
510     "FoodName" : "망고주스",
511     "Company" : "편의점",
512     "Price" : 1000
513 }
514 {
515     "_id" : ObjectId("5d5e3470c8076c145c95fb71"),
516     "Itemno" : 3,
517     "Category" : "음료수",
518     "FoodName" : "식혜",
519     "Company" : "시장",
520     "Price" : 1000
521 }
522 {
523     "_id" : ObjectId("5d5e3470c8076c145c95fb72"),
524     "Itemno" : 4,
525     "Category" : "과자",
526     "FoodName" : "머랭",
527     "Company" : "조각케익가게",
528     "Price" : 3000
529 }
530
531
532 6)mongodbdemo.py
533 import json
534 import pymongo
535 from pymongo import MongoClient
536
537 #Database connection
538 client = MongoClient('localhost', 27017, maxPoolSize = 50)
539
540 #supermarket collection finding
541 db = client.test
542 collection = db['supermarket']
543 cursor = collection.find({}, {'_id':False})
544
545 #cursor로 looping(i에는 0부터 loop의 숫자가 들어간다)
546 for i, document in enumerate(cursor):
```

```
547         rowcontent = "  
548         keycontent = "  
549         #print(i, document)  
550  
551         # #document에서 key와 값을 가져와서 tab으로 구분해서 문자열로 만들  
552         for key, val in document.items():  
553             # print(key, val)  
554             keycontent = keycontent + '\t' + str(key)  
555             rowcontent = rowcontent + '\t' + str(val)  
556  
557         #만약 첫번째 loop라면 key 이름도 출력해서 column 이름 출력한다.  
558         if i == 0 :  
559             print(keycontent)  
560             print(rowcontent)  
561         else:  
562             print(rowcontent)  
563         i += 1
```