Chapter. 04

데이터 베이스 SQL

I 데이터베이스 기본과 SQLite 및 Mysql

FASTCAMPUS
ONLINE
금융공학/퀀트I

강사. 서찬웅

I 이번 시간에 배울 내용

강의내용

1. 데이터베이스와 SQLITE란?
2. sqlite 사용법
3. mysql 접속방법

FAST CAMPUS
ONLINE
서찬웅 강사.

I 데이터베이스(db)에 대해서 간단히 알아보겠습니다.

- 데이터베이스는 체계화된 데이터의 모임이다. 작성된 목록으로써 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음이다. 즉, 여러 사람이 공유하여 사용할 목적으로 통합, 관리하는 데이터의 집합이다. - 위키백과
- Oracle, MS-SQL, Mysql같은 프로그램은 데이터베이스를 관리하기 위한 프로그램의 집합으로 DBMS(DataBase Management System)이라고 합니다.
- DBMS 프로그램에서 정보에 접근하기 위해서 사용하는 언어가 바로 SQL입니다.
- SQL(Structured Query Language)는 구조화된 질의 언어이며, 철자 그대로 에스큐엘이라고 부르거나 '시퀄'이라고 부르는 사람도 있습니다.
    - SQL은 SELECT, JOIN, INSERT, UPDATE등의 명령어를 통해서 데이터를 삽입 및 조회를 할 수 있습니다.
- SQLite는 MySQL나 PostgreSQL와 같은 DBMS이지만, 따로 설치가 필요 없는 비교적 가벼운 데이터베이스입니다.
- 파이썬에서는 sqlite3 모듈이 내장되어 있어, sqlite를 사용할 수 있습니다.

I sqlite3에 대해서 알아보겠습니다.

- sqlite3는 별도의 설치가 없이 디스크 기반의 가벼운 데이터베이스를 제공합니다. 가벼운 데이터베이스지만, 일반적인 데이터베이스 명령어인 CRUD(Create, Read, Update, Delete)도 당연히 동작합니다.
- SQL은 별도의 하나의 과정이므로 이 과정에서 설명을 다 할 수 없습니다. 예제를 통해서 파이썬에서 sqlite3를 사용하는 방법을 알아보도록 하겠습니다.
- 같이 제공된 csv파일을 읽어 sqlite DB에 넣는 예제를 같이 해보겠습니다.

```
import sqlite3
import csv
```

- fastcampus_stock.csv는 삼성전자, 셀트리온, 네이버의 2019년부터 데이터가 들어있습니다.
- 해당 데이터를 sqlite3에 테이블을 생성하고 insert 하는 예제를 통해 학습을 해보겠습니다.

```
with open("./Data/fastcampus_stock.csv", 'r', encoding='utf8') as f:
    filereader = csv.reader(f)
    for content in filereader:
        print (content)
```

```
['st_id', 'st_day', 'st_start', 'st_high', 'st_low', 'st_close', 'st_before', 'st_trade', 'st_money']
['A005930', '2019-01-02', '39400', '39400', '38550', '38750', '50', '7847664', '3046830000000']
['A005930', '2019-01-03', '38300', '38550', '37450', '37600', '-1150', '12471493', '4729670000000']
['A005930', '2019-01-04', '37450', '37600', '36850', '37450', '-150', '14108958', '5254030000000']
['A005930', '2019-01-07', '38000', '38900', '37800', '38750', '1300', '12748997', '4908330000000']
['A005930', '2019-01-08', '38000', '39200', '37950', '38100', '-650', '12756554', '4915780000000']
```

I sqlite3에 테이블을 생성해 봅시다.

- sqlite3에 테이블을 생성하겠습니다. 테이블 생성처럼 데이터를 정의하는 것을 DDL(Data Definition Language)라고 부르며, (생성, 변경, 삭제, 이름변경)에 관련된 명령어들을 의미합니다.
  - CREATE, ALTER, DROP, RENAME, TRUNCATE
  - 문법 : CREATE DATABASE *databasename*;

- connect() 함수를 사용하여 데이터베이스를 나타내는 con이라는 부르는 접속 객체 생성
- 내부 메모리에 생성한다면 :memory: 를 사용
- 외부에 파일로 저장한다면 stock.db 처럼 파일 이름으로 객체 생성

```
con = sqlite3.connect('stock.db')
```

- DDL 명령문을 실행할 Query문을 생성합니다.

```
query = """CREATE TABLE stock_day_table
           (st_id VARCHAR(30) NOT NULL,
            st_day DATE NOT NULL,
            st_start INT(11) NULL DEFAULT NULL,
            st_high INT(11) NULL DEFAULT NULL,
            st_low INT(11) NULL DEFAULT NULL,
            st_close INT(11) NULL DEFAULT NULL,
            st_before INT(11) NULL DEFAULT NULL,
            st_trade BIGINT(20) NULL DEFAULT NULL,
            st_money BIGINT(20) NULL DEFAULT NULL,
            PRIMARY KEY (st_id, st_day)
)"""
```

- execute() 함수는 query를 실행하는 함수입니다. execute()에 위에서 만든 query 문자열 객체를 전달합니다.

I sqlite3에 데이터를 삽입해 봅시다.

- execute() 함수는 query를 실행하는 함수입니다. execute()에 위에서 만든 query 문자열 객체를 전달합니다.

```
con.execute(query)
```

```
<sqlite3.Cursor at 0x7f2ca87c52d0>
```

- 데이터베이스는 데이터가 commit이라는 작업을 최종 진행해야지 반영이 됩니다.

```
con.commit()
```

- 데이터를 삽입해 보겠습니다. 데이터를 삽입 하는 문법은 아래와 같습니다.
- INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...) VALUES (*value1*, *value2*, *value3*, ...);

```
statement = "INSERT INTO stock_day_table VALUES(?, ?, ?, ?, ?,?, ?, ?, ?)"
```

```
with open("./Data/fastcampus_stock.csv", 'r', encoding='utf8') as f:
    filereader = csv.reader(f)
    for idx, content in enumerate(filereader):
        con.execute(statement, content)
```

```
con.commit()
```

I sqlite3에 테이블을 조회해봅시다.

- SELECT는 데이터를 조회하는 문법입니다. 아래와 같습니다.
- SELECT *column1*, *column2, ...* FROM *table_name*;
- DML(Data Manipulation Language) : SELECT, INSERT, UPDATE, DELETE
- execute() 메소드를 실행하여 생성된 객체를 cursor라는 식별자를 가진 변수에 저장합니다.
- cursor란 레코드를 탐색하고 조작할 수 있게 해주는 요소를 의미합니다.
- fetchall() 메소드는 모든 결과를 반환해주는 기능을 합니다. 모든 결과를 rows라는 변수에 저장합니다.

```
cursor = con.execute("SELECT * FROM stock_day_table")
```

```
rows = cursor.fetchall()
```

```
rows
```

```
[('st_id',
  'st_day',
  'st_start',
  'st_high',
  'st_low',
  'st_close',
  'st_before',
  'st_trade',
  'st_money'),
 ('A005930',
  '2019-01-02',
  39400,
  39400,
  38550,
  38750,
  50,
  7847664,
  304683000000),
 ('A005930',
```

I sqlite3에 테이블을 조건을 걸어 조회해봅시다.

- where 구문은 SQL문에서 조건을 걸어 데이터를 조회하게 합니다.
- SELECT *column1*, *column2, ...* FROM *table_name* WHERE *condition*;
- 삼성전자의 1월 2일 데이터만 조회해보겠습니다.

```
cursor = con.execute("SELECT * FROM stock_day_table where st_id = 'A005930' AND st_day = '2019-01-02'")
```

```
cursor.fetchall()
```

```
[('A005930',
  '2019-01-02',
  39400,
  39400,
  38550,
  38750,
  50,
  7847664,
  304683000000)]
```

- 여기서 다루는 SQL 문법은 아주 기초적인 수준입니다. 시간을 투자해서 SQL를 공부하는 것을 권장합니다.

I mysql 설치하여 접속하기

- sqlite를 사용하여 별도의 DB를 설치하지 않고 사용할 수 있지만, 기존 DB가 있다면 파이썬에서는 해당 DB의 접속할 수 있는 모듈을 설치한다면 접속하여 모든 작업을 진행할 수 있습니다.
- 매직 명령어를 실행하여 !pip install pymysql를 실행합니다.
- pymysql은 파이썬에서 mysql를 연결하여 사용할수 있게 만들어주는 외부 라이브러리입니다.

```
import pymysql

# db 연결
try:
    con = pymysql.connect(host="주소", port = 3306, user = '사용자', password='암호', db='Database이름',
                          charset='utf8')
    cur = con.cursor()

except:
    print ("error")
```

I 정리

- sqlite3
- DDL, DML
- INSERT, SELECT, WHERE
- SQL은 따로 더 공부하기

FAST CAMPUS
ONLINE
서찬웅 강사.

감사합니다

FAST CAMPUS
ONLINE
서찬웅 강사.