



분석용 데이터

획득 전략

DBMS를 활용한 데이터 저장 기법

학습 목표

+ + +

학습 목표

- DBMS의 개념을 이해하고 주요 명령어를 습득할 수 있다.
- 수집된 데이터를 MySQL DB에 저장하고 조회 및 관리 작업을 수행할 수 있다.

학습 내용

- DBMS의 개념과 주요 명령어의 이해
- 수집된 데이터를 MySQL DB에 저장하고 관리하기

DBMS의 개념

1) DBMS의 의미

(1) 사전적 의미

☆ 데이터베이스란,

동시에 복수의 적용 업무를 지원할 수 있도록 복수 이용자의 요구에 호응해서 데이터를 받아들이고 저장, 공급하기 위하여 일정한 구조에 따라서 편성된 데이터의 집합

➡ 기업이나 조직체의 활동에 필요 불가결한 자원이 되는 정보에 대한 다양한 요구에 응하기 위하여 대량의 정보를 수집, 관리하여 공동으로 이용할 수 있게 한 것

🔍 데이터베이스의 특징

- ㉠ 어느 특정한 적용 업무나 응용 시스템이 아니라 동시에 복수의 적용 업무나 응용 시스템에 대한 데이터의 공급 기지로서 공유할 필요가 있는 데이터를 보관, 관리함. 이 점에서 특정 목적을 위한 데이터를 관리하는 파일(File)과는 근본적으로 다름
- ㉡ 데이터의 특성, 실체 상호간의 의미 관계와 형식 관계를 기술한 개념적인 구조에 따라서 편성된 데이터의 집합
- ㉢ 동일한 내용의 데이터가 중복되어 있지 않아야 하고, 다양한 접근 방식이 마련되어 있어야 하며, 검색이나 갱신이 효율적으로 이루어질 수 있도록 해야 함
- ㉣ 데이터의 **완전성(data integrity)**이 보증되어야 하고 안전 보호 (Security), 동시 접근이나 장애 회복 기능 등이 마련되어야 함

DBMS의 개념

(2) 요약



데이터베이스

대량의 데이터를 여러 사람들이 동시에 쉽게 사용할 수 있도록
여러가지 규칙에 의해서 데이터를 저장하고 관리하는 것



데이터베이스를 관리하는 소프트웨어



DataBase Management System(Software) → DBMS

DBMS: 데이터베이스를 관리하는 시스템이라는 **장르**의 개념

2) DBMS의 종류

[유료]

- Oracle > Oracle
- Microsoft > MS-SQL Server
- IBM > DB2
- Amazon > AWS RDS 서비스

[무료]

- MySQL/MariaDB
- MongoDB
- PostgreSQL

MySQL DB

1) MySQL의 정의



MySQL(마이에스큐엘)

세계에서 가장 많이 쓰이는 오픈 소스의
관계형 데이터베이스 관리 시스템(RDBMS)

- ➡ 다중 스레드, 다중 사용자 형식의 구조 질의어 형식의 데이터베이스 관리 시스템으로써 오라클이 관리 및 지원하고 있으며, Qt 처럼 이중 라이선스가 적용됨
- ➡ 하나의 옵션은 GPL이며, GPL 이외의 라이선스로 적용시키려는 경우 전통적인 지적재산권 라이선스의 적용을 받음
- ➡ 썬 마이크로시스템즈에 10억 달러에 인수되었으나, 썬 마이크로시스템즈가 오라클에 인수되며 같이 넘어감

2) SQL의 정의



SQL이란,

- Structured Query Language(구조적 질의 언어)
- DBMS와 대화하기 위해 구조화된 언어
- DBMS 종류별도 다른 부분도 있고 공통적인 부분도 있음

MySQL DB

3) SQL 주요 명령어 종류

 **DDL**(Data Definition Language)

→ CREATE (생성), ALTER (수정), TRUNCATE (잘라내기), DROP (삭제)

 **DML**(Data Manipulation Language)

→ INSERT (입력), UPDATE (변경), DELETE (삭제), MERGE (병합)

 **DCL**(Data Control Language)

→ GRANT (권한 주기), REVOKE (권한 뺏기)

 **TCL**(Transaction Control Language)






→ COMMIT (확정), ROLLBACK (취소)

 **SELECT**

→ DQL(Data Query Language)

MySQL DB

4) MySQL 서버에 데이터 저장 순서

-  00 MySQL 프로그램이 먼저 설치되어 있어야 함
-  01 데이터 수집 컴퓨터에서 MySQL 서버로 접속하는 커서 생성
-  02 MySQL 서버에 웹 크롤링 저장용 계정 생성
-  03 MySQL 서버에 데이터를 저장할 테이블 생성
-  04 MySQL 서버에 데이터 입력 후 확인

커서(Cursor)

Python과 MySQL을 연결할 때 다리 역할

MySQL DB

 MySQL DB에서 아래 작업을 먼저 수행

[Step1> 윈도우의 명령 프롬프트 창을 실행하여 MySQL 접속하기]

 C:\WINDOWS\system32>mysql -u root -p Enter password: *****

[Step2> 수집한 데이터를 저장할 데이터베이스 생성하기]

 mysql> CREATE DATABASE crawl_data DEFAULT CHARACTER SET utf8;

[Step3> 수집한 데이터를 관리할 계정 생성하기]

 mysql> CREATE USER cuser IDENTIFIED BY 'cuser!@#';

[Step4> 생성한 계정에 권한 설정하기]

 mysql> GRANT ALL ON crawl_data.* TO cuser;

MySQL DB 활용

■ 입력한 검색어

```
21
22 query_txt = input('1. 크롤링할 키워드는 무엇입니까?: ')
23 cnt=int(input('2. 수집할 데이터는 몇 건입니까?: '))
24 page_cnt = math.ceil(cnt / 60)
25
26 f_dir = input("3. 파일을 저장할 폴더명만 쓰세요(기본값:c:/py_temp/):")
27 if f_dir == '' :
28     f_dir="c:\\py_temp\\"
29
30 n = time.localtime()
31 s = '%04d-%02d-%02d-%02d-%02d' %(n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm
32
33 # Step 3. 크롬 드라이버와 폴더와 파일을 설정하기
34 s1 = Service("c:/py_temp/chromedriver.exe")
35 driver = webdriver.Chrome(service=s1)
36
37 #이미지를 저장할 디렉토리 생성
38 img_dir = f_dir+'Naver_View'+ '-' +s+ '-' +query_txt+"\\images"
39 os.makedirs(img_dir)
40 os.chdir(img_dir)
41
42 # csv , xlsx 파일명 지정
43 fc_name = f_dir+'Naver_View'+ '-' +s+ '-' +query_txt+'\\'+ 'Naver_View'+ '-' +s+ '-' +query_txt+'.csv'
44 fx_name = f_dir+'Naver_View'+ '-' +s+ '-' +query_txt+'\\'+ 'Naver_View'+ '-' +s+ '-' +query_txt+'.xlsx'
45 print('=' *100)
```

■ 스톱워치 기능

```
47 #Step 4. 네이버에 접속 후 검색하고 블로그 선택하기
48 url = 'https://www.naver.com'
49 driver.get(url)
50 driver.maximize_window()
51 time.sleep(2)
52
53 s_time = time.time()
54 element = driver.find_element(By.ID, 'query')
55 driver.find_element(By.ID, 'query').click()
56 element.send_keys(query_txt)
57 element.send_keys("\n")
58 time.sleep(2)
59
60 driver.find_element(By.LINK_TEXT, 'VIEW').click()
61 time.sleep(1)
62 driver.find_element(By.LINK_TEXT, '블로그').click()
```

MySQL DB 활용

▪ 매뉴얼대로 MySQL DB 설치

```
1 #####
2 # My SQL DB 관련 작업
3 #####
4 # 아래의 작업을 하기 전에 먼저 MySQL DB에서 아래의 작업을 수행해주세요
5 #[Step 1. 윈도의 명령 프롬프트 창을 실행하여 Mysql에 접속하기]
6 #C:\WINDOWS\system32>mysql -u root -p
7 #Enter password: *****
8 #
9 #[Step 2. 수집한 데이터를 저장할 데이터 베이스 생성하기]
10 #mysql> CREATE DATABASE crawl_data DEFAULT CHARACTER SET utf8;
11 #
12 #[Step 3. 수집한 데이터를 관리할 계정 생성하기]
13 #mysql> CREATE USER cuser IDENTIFIED BY 'cuser!@#';
14 #
15 #[Step 4. 생성한 계정에 권한 설정하기]
16 #mysql> GRANT ALL ON crawl_data.* TO cuser;
17 #####
18
19 #Step 8. MySQL DB에 저장하기
20 # 접속 정보 등록 및 커서 생성하기
21 import pymysql
22 conn = pymysql.connect(host='192.168.150.130', user='cuser', password='cuser!@#', \
23                        db='crawl_data', charset='utf8')
24 cur = conn.cursor()
25
26 # 테이블 생성하기
27 dropsql = "DROP TABLE IF EXISTS naver_blog"
28 cur.execute(dropsql)
29
30 sql = '''CREATE TABLE naver_blog (no char(4),
31                                     title char(50), content char(200), rdate char(20), nick char(30) )
32                                     '''
33 cur.execute(sql)
```

주의사항

host뒤에 본인PC의IP주소를 입력

MySQL DB 활용

■ 저장 확정

```
35 # csv 파일 읽어서 테이블에 데이터 입력하기
36 import csv
37
38 sql = "insert into naver_blog (no, title, content, rdate ,
39 file = open(fc_name, 'r', encoding='utf-8')
40 rdata = csv.reader(file)
41
42 for i in rdata:
43     cur.execute(sql, (i[0], i[1], i[2] , i[3], i[4]))
44
45 conn.commit()
46 conn.close()
47 file.close( )
```

■ 연결을 끊어라

```
35 # csv 파일 읽어서 테이블에 데이터 입력하기
36 import csv
37
38 sql = "insert into naver_blog (no, title, content, rdate ,
39 file = open(fc_name, 'r', encoding='utf-8')
40 rdata = csv.reader(file)
41
42 for i in rdata:
43     cur.execute(sql, (i[0], i[1], i[2] , i[3], i[4]))
44
45 conn.commit()
46 conn.close()
47 file.close( )
```

■ 파일 저장

```
35 # csv 파일 읽어서 테이블에 데이터 입력하기
36 import csv
37
38 sql = "insert into naver_blog (no, title, content, rdate ,
39 file = open(fc_name, 'r', encoding='utf-8')
40 rdata = csv.reader(file)
41
42 for i in rdata:
43     cur.execute(sql, (i[0], i[1], i[2] , i[3], i[4]))
44
45 conn.commit()
46 conn.close()
47 file.close( )
```