

데이터 크롤링과 정제

6장. 데이터 저장

목차

- 미디어 파일
- 데이터를 CSV로 저장
- MySQL
 - MySQL 설치
 - 기본 명령어
 - 파이썬과 통합

미디어 파일

- urllib.request 라이브러리
 - urlretrieve() 함수 원형

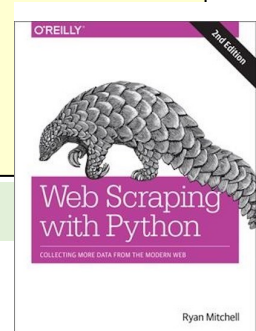
```
urlretrieve(url, filename=None, reporthook=None, data=None)
```

- url로 표시된 네트워크 객체를 로컬 파일로 복사
- filename: 복사할 파일 위치 및 이름 지정

```
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com')
bs = BeautifulSoup(html, 'html.parser')
home_image = bs.find('img', {'class': 'pagelayer-img'})
image_location = home_image['src']
print(image_location)
urlretrieve(image_location, 'logo.jpg')
```

```
https://pythonscraping.com/wp-content/uploads/2021/08/home1.jpg
```



미디어 파일

- src 속성이 있는 태그에 연결된 내부 파일 출력

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com')
bs = BeautifulSoup(html, 'html.parser')
downloadList = bs.find_all(src=True) # 특정 속성이 있는 태그를 찾을 때: True

for down_url in downloadList:
    print(down_url['src'])
```

src 속성에 있는 링크 출력

```
https://pythonscraping.com/wp-includes/js/jquery/jquery.min.js?ver=3.6.0
https://pythonscraping.com/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2
https://pythonscraping.com/wp-content/plugins/pagelayer/js/givejs.php?give=pagelayer-
frontend.js%2Cnivo-lightbox.min.js%2Cwow.min.js%2Cjquery-
numerator.js%2CsimpleParallax.min.js%2Cowl.carousel.min.js&premium&ver=1.5.9
https://pythonscraping.com/wp-content/uploads/2021/08/home1.jpg
https://pythonscraping.com/wp-content/uploads/2021/08/logo01.png
https://pythonscraping.com/wp-content/themes/popularfx/js/navigation.js?ver=1.2.0
```

6.1 예제 소스: 내부 파일 저장 #1

```
import os
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

downloadDirectory = 'downloaded'
baseUrl = 'https://pythonscraping.com'
def getAbsoluteURL(baseUrl, source):
    if source.startswith('http://www.'):
        url = 'http://{0}'.format(source[11:]) # 인덱스 11부터 끝까지
    elif source.startswith('http://'):
        url = source
    elif source.startswith('www.'):
        url = source[4:]
        url = 'http://{0}'.format(source)
    else:
        url = '{0}'.format(source)
    return url

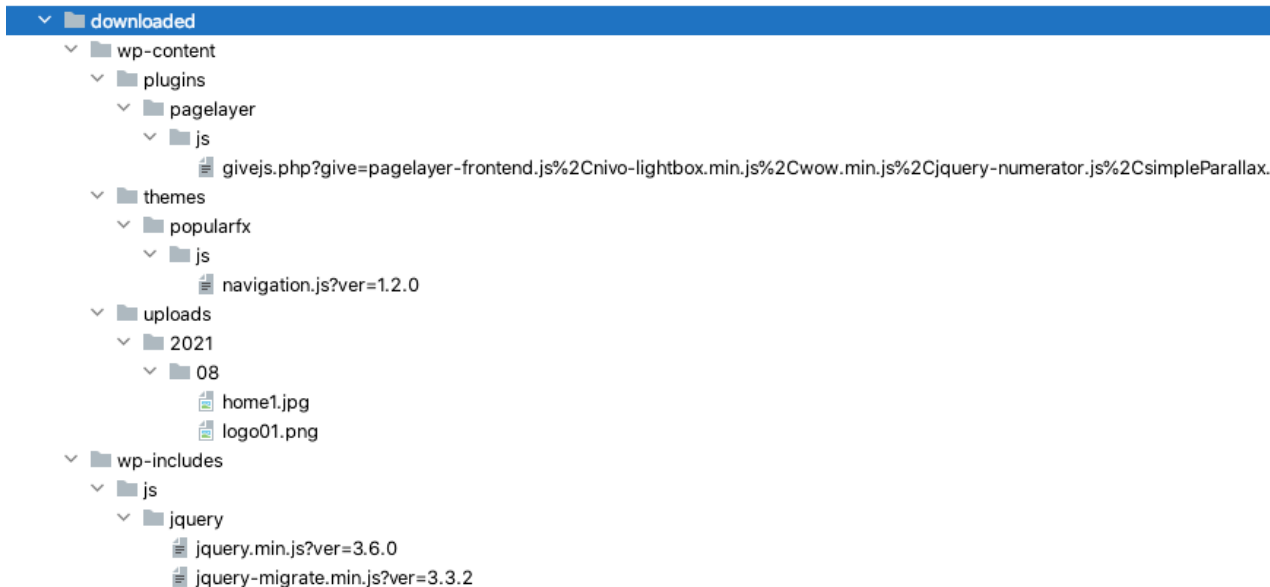
def getDownloadPath(baseUrl, absoluteUrl, downloadDirectory):
    path = absoluteUrl.replace('www.', '')
    path = path.replace(baseUrl, '')
    path = downloadDirectory + path
    directory = os.path.dirname(path)

    if not os.path.exists(directory):
        os.makedirs(directory)
    return path
```

6.1 예제 소스: 내부 파일 저장 #2

```
html = urlopen('http://www.pythonscraping.com')
bs = BeautifulSoup(html, 'html.parser')
downloadList = bs.find_all(src=True)

for download in downloadList:
    fileUrl = getAbsoluteURL(baseUrl, download['src'])
    if fileUrl is not None:
        print(fileUrl)
        urlretrieve(fileUrl, getDownloadPath(baseUrl, fileUrl,
                                                downloadDirectory))
```



downloaded 폴더
아래에 저장된 파일

데이터를 CSV 파일로 저장

- CSV (Comma Separated Values)
 - 쉼표로 구분된 값
 - python에 기본 내장된 라이브러리 사용
 - `import csv`
 - 파일 열기

```
f = open (csvfile, dialect='excel', **fmtparams)
```

- 파라미터
 - `csvfile`: csv 파일 이름
 - `'w'`: 쓰기 모드로 열기, `'w+'`: 읽기 또는 쓰기 모드
 - 파일이 없으면 새롭게 생성함
 - `encoding='utf-8'`: UTF-8로 인코딩

```
writer = csv.writer(f, dialect='excel', **fmtparams)
```

- csv writer 객체 리턴
- 파라미터
 - `f`: 파일 객체
 - `newline=''`: Windows 환경에서 빈 라인이 추가되는 것을 막기 위함
- 파일 저장
 - `writer.writerow(row)`: 한 라인씩 csv파일에 저장
 - `writer.writerows(rows)`

데이터를 CSV파일로 저장

```
import csv
csvFile = open('test.csv', 'w', encoding='UTF-8')
try:
    writer = csv.writer(csvFile)
    writer.writerow(('number', 'number plus 2', 'number times 2'))
    for i in range(10):
        writer.writerow((i, i+2, i*2))
finally:
    csvFile.close()
```

	A	B	C
1	number	number plus 2	number times 2
2	0	2	0
3	1	3	2
4	2	4	4
5	3	5	6
6	4	6	8
7	5	7	10
8	6	8	12
9	7	9	14
10	8	10	16
11	9	11	18

Excel 사용

1	number,number plus 2,number times 2
2	0,2,0
3	1,3,2
4	2,4,4
5	3,5,6
6	4,6,8
7	5,7,10
8	6,8,12
9	7,9,14
10	8,10,16
11	9,11,18

일반 Text 에디터 사용

6.2 데이터를 CSV로 저장

- 위키피디아에서 텍스트 에디터 비교 테이블을 가져와서 저장하기
 - https://en.wikipedia.org/wiki/Comparison_of_text_editors
 - 해당 URL의 첫 번째 테이블

List of text editors

Name	Developer	Initial release	Latest release		Programming language	Cost (US\$)	License	GUI	TUI or CLI
			Version	Date					
Acme	Rob Pike	1993	Plan 9 and Inferno		C	No cost	MIT GPL-2.0-only LPL-1.02	✓	✗
AkelPad	Alexey Kuznetsov Alexander Shengalts	2003	4.9.8^[1]	2016-07-18	C	No cost	BSD-2-Clause	✓	✗
Alphatk	Vince Darley	1999	8.3.3^[2]	2004-12-10		\$40	Proprietary, with BSD components	✓	✗
Aquamacs	David Reitter	2005	3.6^[3]	2021-12-28	C, Emacs Lisp	No cost	GPL-3.0-or-later	✓	✓
Atom	GitHub	2014	1.60.0^[4]	2022-03-08	HTML, CSS, JavaScript, C++	No cost	MIT	✓	✗
BBEdit	Rich Siegel	1992	14.1.1^[5]	2022-03-16	Objective-C , Objective-C++	No cost for most features, \$49.99 for full version	Proprietary	✓	✗
Bluefish	Bluefish Development Team	1999	2.2.12^[6]	2020-11-05	C	No cost	GPL-3.0-or-later	✓	✗
Brackets	Adobe Systems	2012	2.0.1^[7]	2021-12-01	HTML, CSS, JavaScript, C++	No cost	MIT	✓	✗
Coda	Panic	2007	2.7.7^[8]	2020-11-05	Objective-C	\$99	Proprietary	✓	
ConTEXT	ConTEXT Project Ltd	1999	0.98.6^[9]	2009-08-14	Object Pascal (Delphi)	No cost	BSD-3-Clause	✓	
Crimson Editor	Ingyu Kang	1999	3.72-r286m^[10]	2011-10-01	C++	No cost	Proprietary	✓	
CudaText	UVViewSoft^[a]	2015	1.165.0^[11]^[12]	2022-05-24	Object Pascal (Lazarus)	No cost	MPL-2.0	✓	✗
ed	Ken Thompson	1970	unchanged from original		C	No cost	?	✗	✓

6.2 데이터를 CSV로 저장 예제 #1

```
import csv
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://en.wikipedia.org/wiki/Comparison_of_text_editors')
bs = BeautifulSoup(html, 'html.parser')
# 두 개의 테이블 중에 첫 번째 테이블 사용
table = bs.find_all('table', {'class': 'wikitable'})[0]
rows = table.find_all('tr')

csvFile = open('editors.csv', 'wt', encoding='utf-8') # t: text mode
writer = csv.writer(csvFile)

try:
    for row in rows:
        csvRow = []
        for cell in row.find_all(['td', 'th']):
            csvRow.append(cell.get_text())

        writer.writerow(csvRow)
finally:
    csvFile.close()
```

교재와 다른 부분

6.2 데이터를 CSV로 저장 예제 #1

■ 실행 결과 확인: editors.csv

	A	B	C	D	E	F	G	H	I
1	Name	Developer	Initial release	Latest release	Programming language	Cost (US\$)	License	GUI	TUI or CLI
2	Version	Date							
3	Acme	Rob Pike	1993	Plan 9 and Inimob			No cost	MITGPL-2.0-onlyLPL-1.02	
4	AkelPad	Alexey KuznetsovAlexander Shengalts	2003	4.9.8[1]호	2016-07-18	C	No cost	BSD-2-Clause	
5	Alphatk	Vince Darley	1999	8.3.3[2]호	2004-12-10		\$40	Proprietary, with BSD components	
6	Aquamacs	David Reitter	2005	3.6[3]호	2021-12-28	C, Emacs Lisp	No cost	GPL-3.0-or-later	
7	Atom	GitHub	2014	1.60.0[4]호	2022-03-08	HTML, CSS, JavaScript, C++	No cost	MIT	
8	BBEdit	Rich Siegel	1992	14.1.1[5]호	2022-03-16	Objective-C, Objective-C++	No cost for most features, \$49.99 for full version	Proprietary	

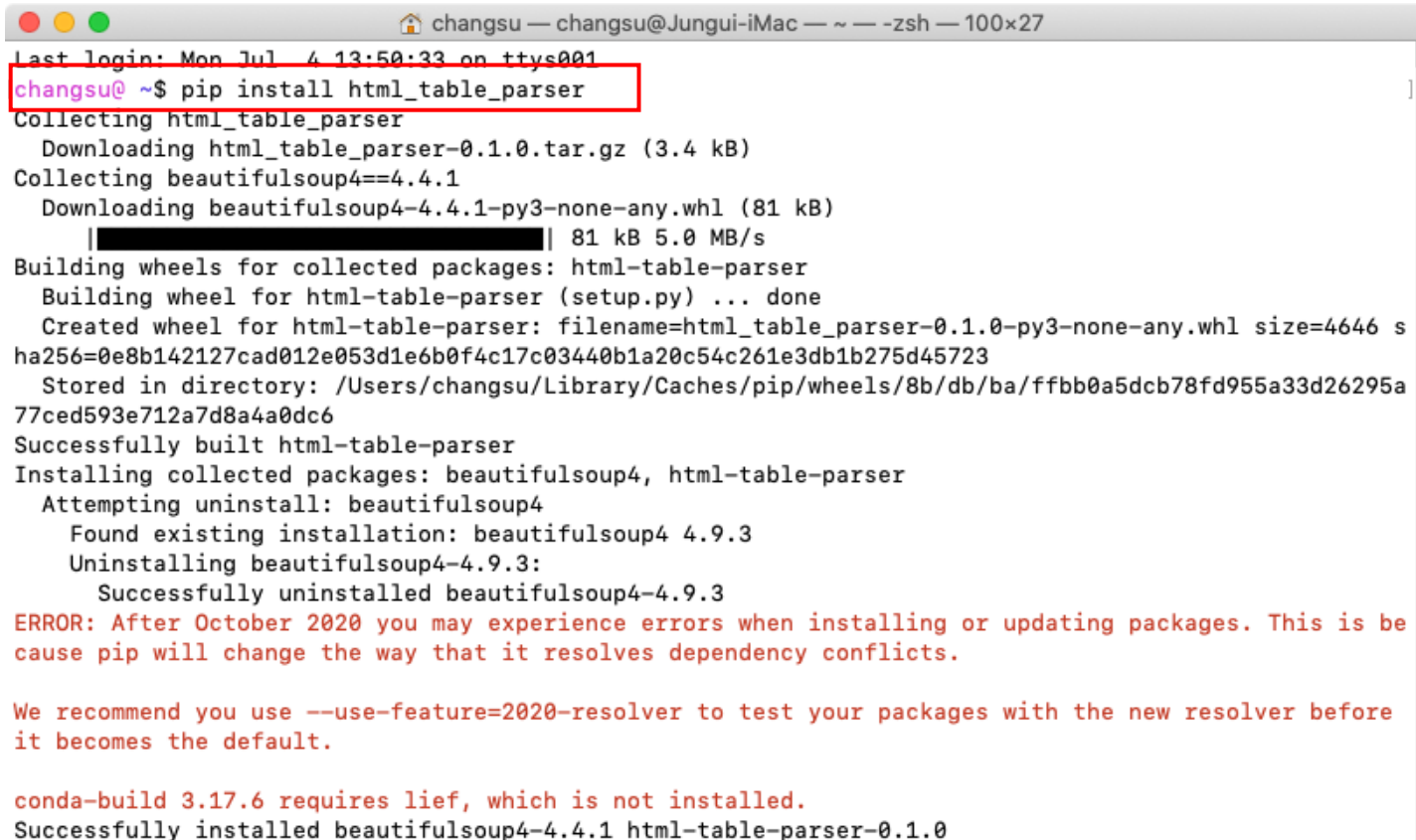
colspan=2로 설정

Latest release 아래에 Version과 Date 항목이 있어서 저장 위치가 원본과 다름

웹페이지 테이블 크롤링하기

■ html_table_parser 패키지 설치

```
$ pip install html_table_parser
```



```
changsus — changsu@Jungui-iMac — ~ — zsh — 100x27
Last login: Mon Jul 4 13:50:33 on ttys001
changsus@ ~$ pip install html_table_parser
Collecting html_table_parser
  Downloading html_table_parser-0.1.0.tar.gz (3.4 kB)
Collecting beautifulsoup4==4.4.1
  Downloading beautifulsoup4-4.4.1-py3-none-any.whl (81 kB)
    |████████████████████| 81 kB 5.0 MB/s
Building wheels for collected packages: html-table-parser
  Building wheel for html-table-parser (setup.py) ... done
  Created wheel for html-table-parser: filename=html_table_parser-0.1.0-py3-none-any.whl size=4646 s
ha256=0e8b142127cad012e053d1e6b0f4c17c03440b1a20c54c261e3db1b275d45723
  Stored in directory: /Users/changsus/Library/Caches/pip/wheels/8b/db/ba/ffbb0a5dcb78fd955a33d26295a
77ced593e712a7d8a4a0dc6
Successfully built html-table-parser
Installing collected packages: beautifulsoup4, html-table-parser
  Attempting uninstall: beautifulsoup4
    Found existing installation: beautifulsoup4 4.9.3
    Uninstalling beautifulsoup4-4.9.3:
      Successfully uninstalled beautifulsoup4-4.9.3
ERROR: After October 2020 you may experience errors when installing or updating packages. This is be
cause pip will change the way that it resolves dependency conflicts.

We recommend you use --use-feature=2020-resolver to test your packages with the new resolver before
it becomes the default.

conda-build 3.17.6 requires lief, which is not installed.
Successfully installed beautifulsoup4-4.4.1 html-table-parser-0.1.0
```

6.2 데이터를 CSV로 저장 예제 개선 내용

- colspan=2로 설정된 부분 처리
 - html_table_parser 라이브러리 사용
 - make2d() 함수: 2차원 리스트 형태로 변환
 - Pandas의 DataFrame으로 변환
 - csv 파일 저장

생성된 DataFrame

	Name	Developer	Initial release	Version	Date	Programming language	Cost (US\$)
0	Acme	Rob Pike	1993	Plan 9 and Inferno		C	
1	AkelPad	Alexey KuznetsovAlexander Shengalts	2003	4.9.8[1]	2016-07-18	C	
2	AlphatK	Vince Darley	1999	8.3.3[2]	2004-12-10		
3	Aquamacs	David Reitter	2005	3.6[3]	2021-12-28	C, Emacs Lisp	
4	Atom	GitHub	2014	1.60.0[4]	2022-03-08	HTML, CSS, JavaScript, C++	
5	BBEdit	Rich Siegel	1992	14.1.1[5]	2022-03-16	Objective-C, Objective-C++	No cost for most features
6	Bluefish	Bluefish Development Team	1999	2.2.12[6]	2020-11-05	C	
7	Brackets	Adobe Systems	2012	2.0.1[7]	2021-12-01	HTML, CSS, JavaScript, C++	
8	Coda	Panic	2007	2.7.7[8]	2020-11-05	Objective-C	
9	ConTEXT	ConTEXT Project Ltd	1999	0.98.6[9]	2009-08-14	Object Pascal (Delphi)	

6.2 데이터를 CSV로 저장 예제 개선 소스

```
import csv
from urllib.request import urlopen
from bs4 import BeautifulSoup
from html_table_parser import parser_functions as parse
import pandas as pd

html = urlopen('http://en.wikipedia.org/wiki/Comparison_of_text_editors')
bs = BeautifulSoup(html, 'html.parser')
# 두 개의 테이블 중에 첫 번째 테이블 사용
table = bs.find_all('table', {'class': 'wikitable'})[0]
rows = table.find_all('tr')

table_data = parse.make2d(table) # 2차원 배열

# Pandas DataFrame으로 저장
df = pd.DataFrame(table_data[2:], columns=table_data[1])
print(df)

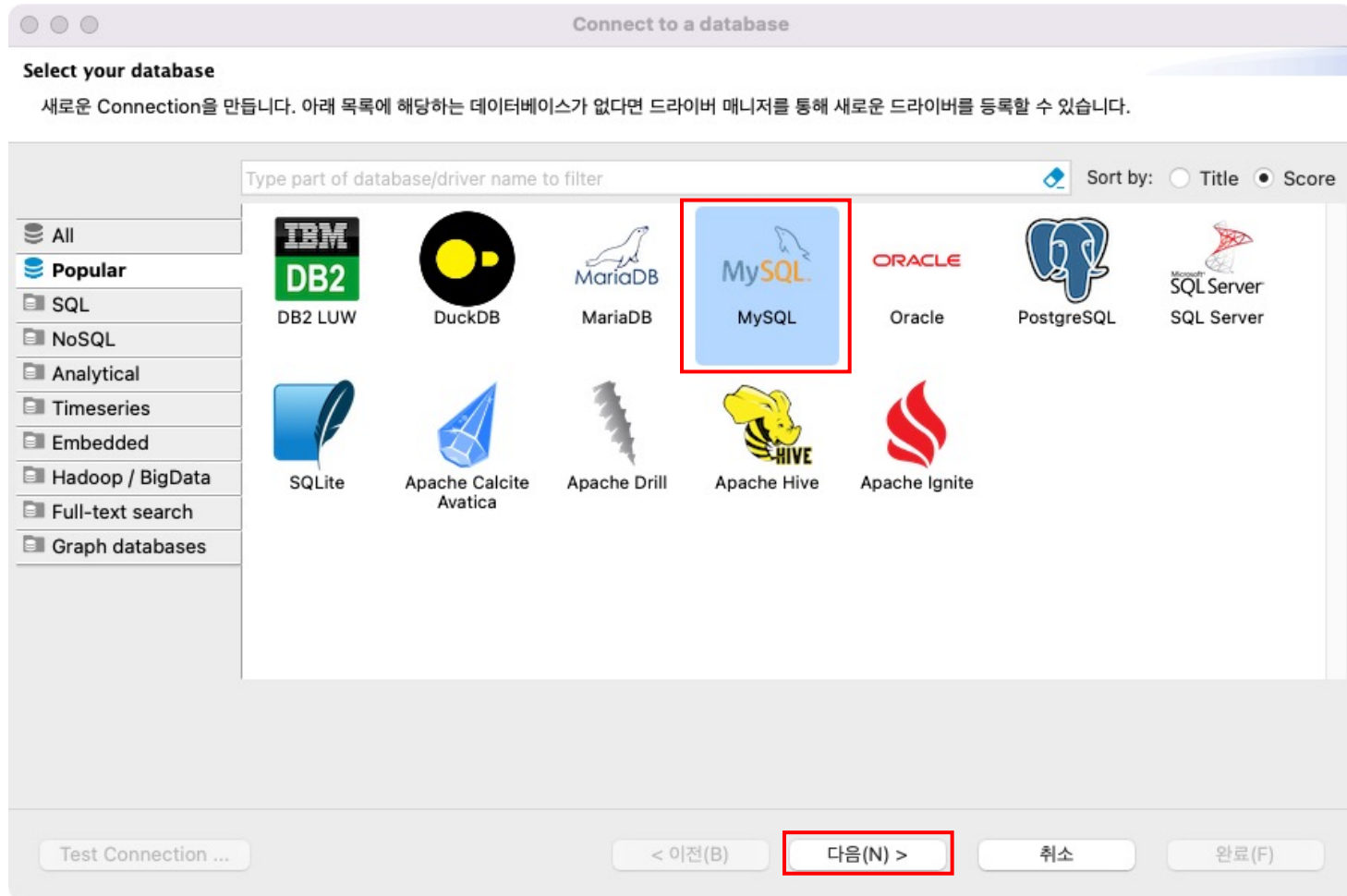
# csv 파일로 저장
csvFile = open('editors1.csv', 'w', encoding='utf-8') # t: text mode
writer = csv.writer(csvFile)

for row in table_data:
    writer.writerow(row)

csvFile.close()
```

DBeaver 실행

■ MySQL 연결



MySQL에 연결

■ Server Host

- 본인의 컴퓨터에 연결: localhost
- 다른 컴퓨터에 설치된 MySQL 연결: 해당 컴퓨터의 IP 주소 입력

Connect to a database

Connection Settings
MySQL connection settings

MySQL

Main Driver properties SSH Proxy SSL

Server

Server Host: localhost Port: 3306

Database:

Authentication (Database Native)

Username: root

Password: ☒ Save password locally

Advanced

Server Time Zone: Auto-detect

Local Client: <not present>

MySQL 설치시 입력한 Password 입력

① You can use variables in connection parameters.

Connection details (name, type, ...)

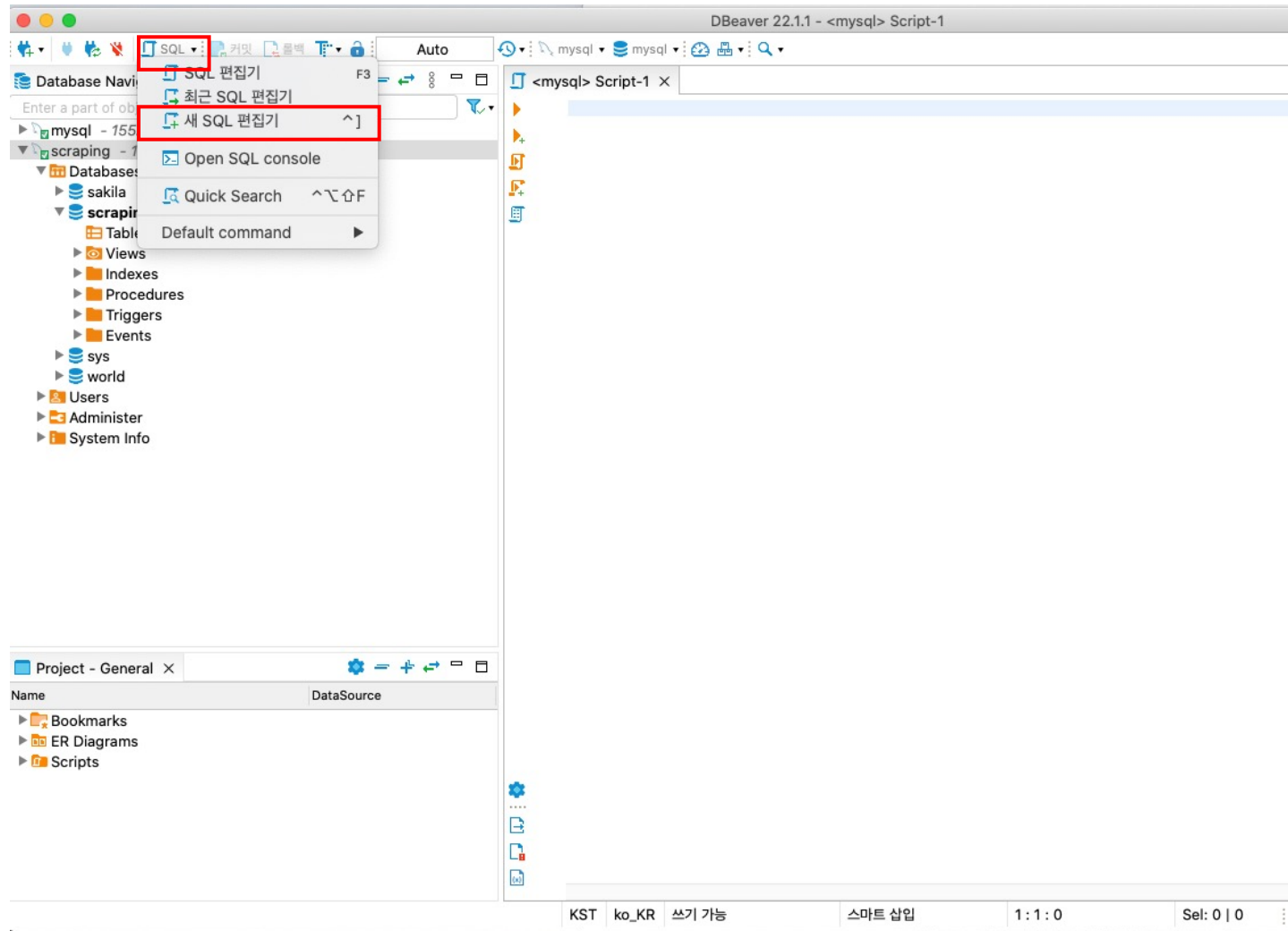
Driver name: MySQL Edit Driver Settings

Test Connection ...

< 이전(B) 다음(N) > 취소 완료(F)

SQL > 새 SQL 편집기 실행

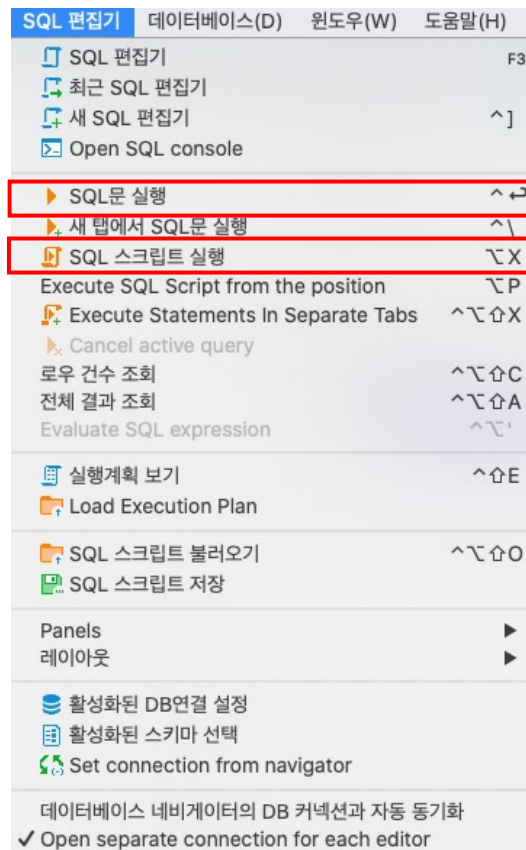
■ 새 SQL 편집기 선택



DBeaver 실행 화면

■ SQL 편집기 메뉴

- 한 줄의 쿼리 실행: SQL문 실행 메뉴(**Ctrl + Enter**)
 - 블록 설정 후 Ctrl + Enter
- 여러 라인의 쿼리 실행: SQL 스크립트 실행 메뉴(**Alt + X**)



새로운 데이터 베이스 및 테이블 생성

- 새로운 데이터 베이스 생성
 - 모든 명령어는 **세미콜론(;)으로 끝남**

```
create database scraping;
```

- 생성된 데이터 베이스 삭제

```
drop database scraping;
```

- 생성한 데이터 베이스 선택

```
use scraping;
```

DBeaver 실행 화면

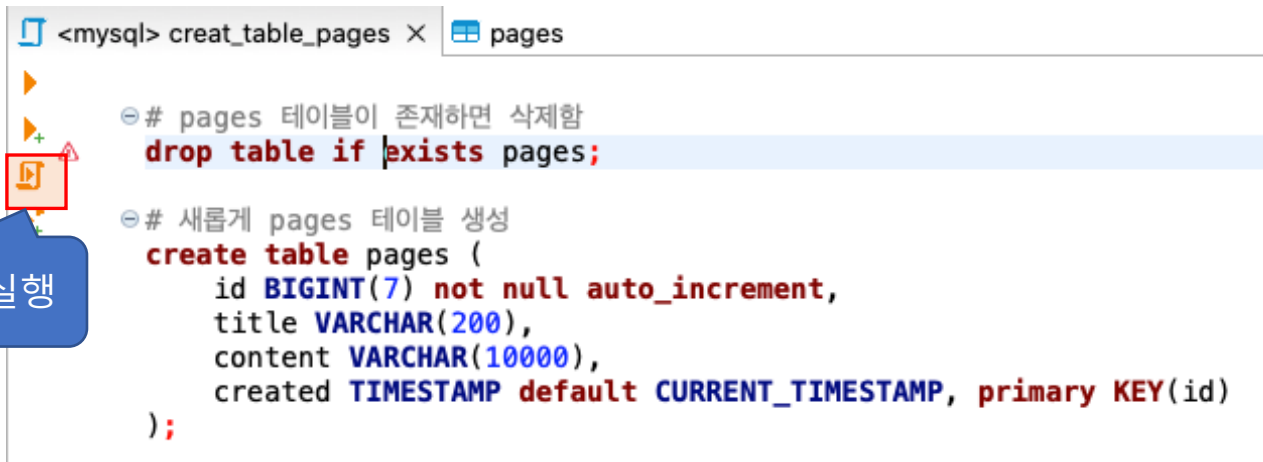
- 데이터 베이스에 새로운 테이블(pages) 생성
 - 테이블 생성시 기존 테이블이 있으면 삭제함

```
# pages 테이블이 존재하면 삭제함
drop table if exists pages;
```

```
create table pages (
  id BIGINT(7) not null auto_increment,
  title VARCHAR(200),
  content VARCHAR(10000),
  created TIMESTAMP default CURRENT_TIMESTAMP, PRIMARY KEY(id)
);
```

하나의 테이블에는
반드시 PRIMARY KEY가
존재해야 됨

– default CURRENT_TIMESTAMP: 자동으로 현재 시간이 저장됨



스크립트 실행

생성된 테이블 확인

- 오른쪽 트리창에서 생성한 테이블(pages) 더블 클릭
 - **Properties** 탭에서 테이블의 컬럼 확인

Database Navigator: 155.230.120.235 - 155.230.120.235:3306, mysql - 155.230.120.235:3306, scraping - 155.230.120.235:3306

Database: scraping, Tables: pages (16K)

Properties for 'pages':

- 테이블명: pages
- Engine: InnoDB
- Auto Increment: 0
- Charset: utf8mb4
- Collation: utf8mb4_0900_ai_ci
- Description:

컬럼명	#	Data Type	Not Null	Auto Increment	Key	더플트	Extra
id	1	bigint	[v]	[v]	PRI		auto_incr
title	2	varchar(200)	[]	[]			
content	3	varchar(10000)	[]	[]			
created	4	timestamp	[]	[]		CURRENT_TIMESTAMP	DEFAULT

테이블에 포함된 컬럼 표시

pages 테이블 내용 확인

- 각 컬럼은 3가지 부분으로 구성됨
 - 컬럼명
 - id, title, content, created
 - Data Type
 - bigint, varchar, timestamp 등
 - 추가 옵션
 - NOT NULL, AUTO_INCREMENT

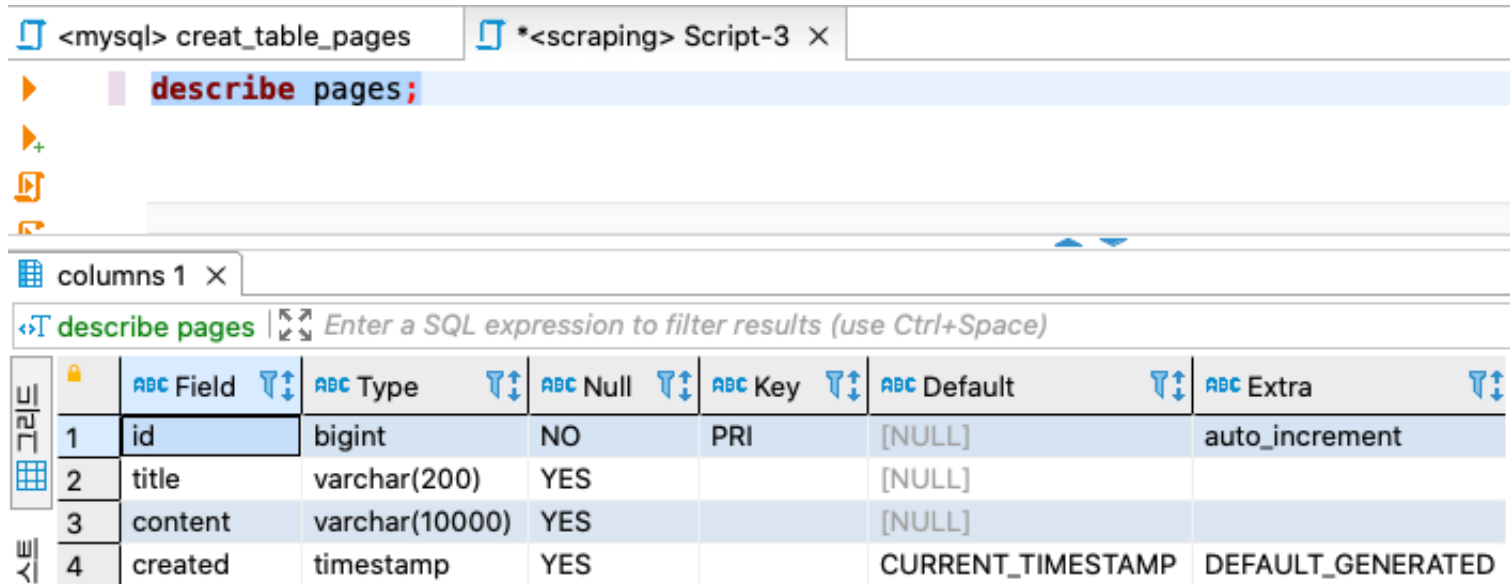
	컬럼명	#	Data Type	Not Null	Auto Increment	Key	디폴트	Extra
Columns	id	1	bigint	[v]	[v]	PRI		auto_incr
Constraints	title	2	varchar(200)	[]	[]			
Foreign Keys	content	3	varchar(10000)	[]	[]			
References	created	4	timestamp	[]	[]		CURRENT_TIMESTAMP	DEFAULT
Triggers								
Indexes								
Partitions								
Statistics								
DDL								
Virtual								

테이블 내용 확인 명령어

- 테이블 구조 확인

```
describe pages;
```

- DBeaver 실행 화면



The screenshot shows the DBeaver SQL editor interface. The top toolbar includes icons for connecting to a database, running a query, and saving. The SQL editor contains the command `describe pages;`. Below the editor, the 'columns' tab is active, displaying the results of the command in a table. The table has columns for Field, Type, Null, Key, Default, and Extra. The results show four columns: id (bigint, primary key, auto-increment), title (varchar(200)), content (varchar(10000)), and created (timestamp, default generated).

	Field	Type	Null	Key	Default	Extra
1	id	bigint	NO	PRI	[NULL]	auto_increment
2	title	varchar(200)	YES		[NULL]	
3	content	varchar(10000)	YES		[NULL]	
4	created	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

pages 테이블에 데이터 추가

- 데이터 추가: **INSERT** 명령어
 - title, content 컬럼에 데이터 추가

```
insert into pages(title, content)
values(
  "Test page title",
  "This is some test page content. It can be up to 10,000 characters long.");
```

- 해당 테이블 선택 후 **Data** 탭 확인

<mysql> creat_table_pages <scraping> chap06_script pages X				
Properties Data 엔티티 관계도				
pages Enter a SQL expression to filter results (use Ctrl+Space)				
	id	title	content	created
1	1	Test page title	This is some test page content. It can be up to 10,000	2022-07-05 11:48:44

id 컬럼은 자동 증가

created 열
- 현재 시간 자동 입력

데이터 추가 및 데이터 가져오기

■ 두 번째 데이터 추가

```
insert into pages(title, content)
values(
    "Second page title",
    "This is the second test page content"
);
```

	id	title	content	created
1	1	Test page title	This is some test page content. It can be up to 10,000 characters long.	2022-07-05 11:48:44
2	2	Second page title	This is the second test page content	2022-07-05 11:57:18

■ pages 테이블에서 id값이 2인 모든 데이터 가져오기

```
select * from pages where id = 2;
```

pages 1

데이터 가져오기 예제

- title 필드(컬럼)에 "test"를 포함하는 행 반환

```
select * from pages where title like "%test%";
```

	id	title	content	created
1	1	Test page title	This is some test page content. It can be up to 10,000 c	2022-07-05 11:48:44

- 조건에 맞는 id, title, created 필드만 가져오기
 - content 필드에서 "second" 포함

```
select id, title, created from pages where content like "%second%";
```

```
id|title|created|
--+-+-----+
2|Second page title|2022-07-05 11:57:18|
```

데이터 삭제 및 업데이트

■ 데이터 삭제: **DELETE** 명령어

- id값이 1인 행 삭제

```
delete from pages where id=1;
```

- 전체 pages 테이블 데이터 가져오기

```
select * from pages;
```

id	title	content	created
2	Second page title	This is the second test page content	2022-07-05 11:57:18

■ 자료 수정: **UPDATE** 명령어

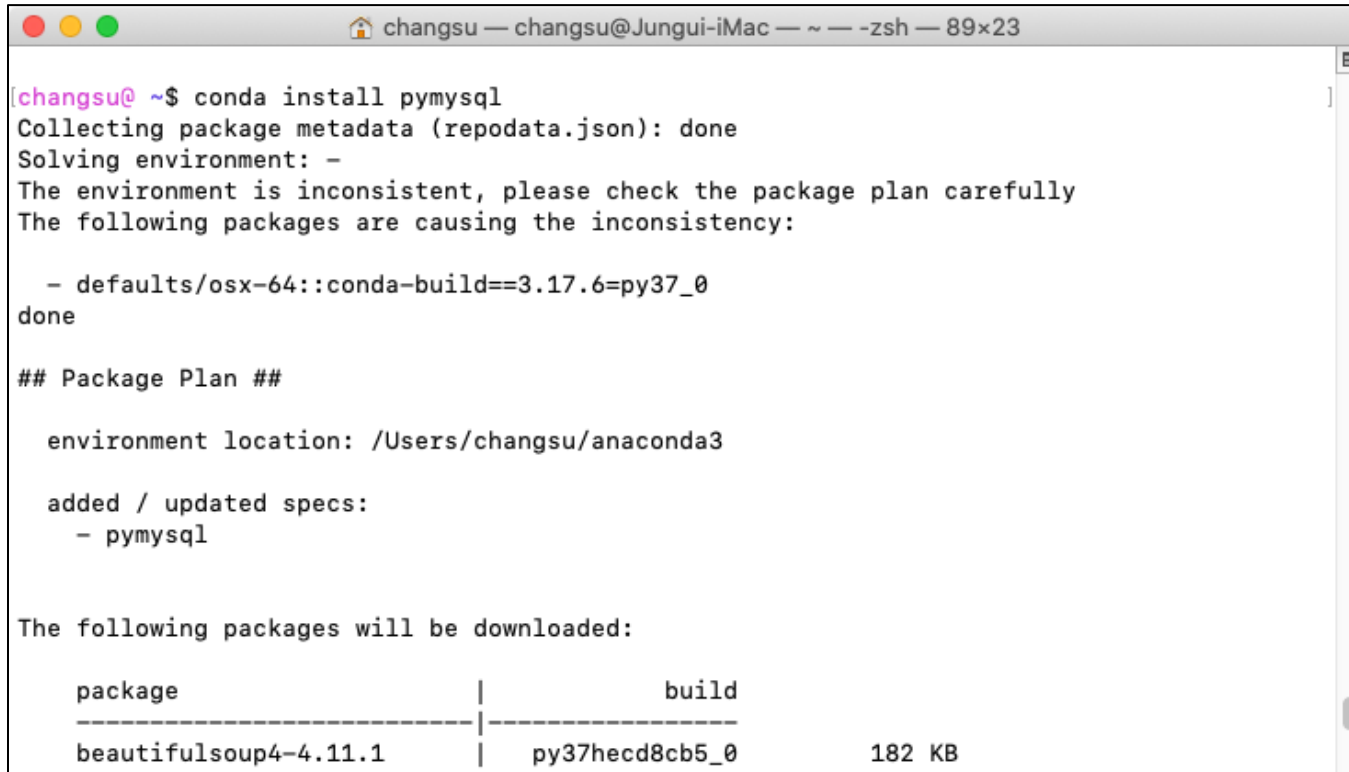
```
update pages set title="A new title",  
content="Some new content" where id=2;
```

id	title	content	created
2	Second page title	This is the second test page content	2022-07-05 11:57:18

6.3.3 파이썬과 통합

- MySQL과 Python을 연동하기 위해 [PyMySQL](#) 라이브러리 설치

```
$ conda install pymysql
```



```
changsu — changsu@Jungui-iMac — ~ — zsh — 89x23

[changsu@ ~]$ conda install pymysql
Collecting package metadata (repodata.json): done
Solving environment: -
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

- defaults/osx-64::conda-build==3.17.6=py37_0
done

## Package Plan ##

environment location: /Users/changsu/anaconda3

added / updated specs:
- pymysql

The following packages will be downloaded:

package | build | size
-----|-----|-----
beautifulsoup4-4.11.1 | py37hecd8cb5_0 | 182 KB
```

6.3.3 파이썬과 통합: PyMySQL 연동

▪ connect() 함수

- host: DB가 존재하는 서버의 주소(`localhost` 또는 `IP주소`)
- user: 사용자 ID
- db: 연결할 데이터베이스 이름

```
import pymysql

conn = pymysql.connect(host= 'localhost',
                       user='root', passwd= '비밀번호',
                       db='scraping', charset='utf8')

cur = conn.cursor()
cur.execute("SELECT * FROM pages WHERE id=2")
print(cur.fetchone())
cur.close()
conn.close()
```

```
(2, 'A new title', 'Some new content', datetime.datetime(2022, 7, 5, 11, 57, 18))
```

6.3.3 파이썬과 통합: PyMySQL

- `cursor()` 함수
 - `cursor` 객체 생성
 - `cursor`: 쿼리문에 의해 반환되는 결과값을 저장하는 공간
- `execute('쿼리문장')` 함수
 - 작성한 쿼리를 실행
- `fetch` 관련 함수
 - `fetchall()`: 모든 데이터를 한 번에 가져옴
 - `fetchone()`: 한 번 호출에 하나의 행만 가져옴
 - `fetchmany(n)`: `n`개 만큼의 데이터를 가져옴
- `commit()` 함수
 - 데이터를 추가, 수정, 삭제 등의 작업을 수행한 다음에 실행
- `close()` 함수
 - MySQL과의 연결 종료

6.3.3. 파이썬과 통합: 예제 #1

- 위키피디아의 자료를 MySQL에 저장
 - Kevin Bacon에 연결된 모든 링크 가져오기

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import datetime
import random
import pymysql
import re

conn = pymysql.connect(host='localhost',
                       user='사용자ID', passwd='비밀번호', db='scraping', charset='utf8')

cur = conn.cursor()
random.seed(datetime.datetime.now())

def store(title, content):
    cur.execute('INSERT INTO pages (title, content) VALUES ("%s", "%s")',
                (title, content))
    cur.connection.commit()
```

title, content 필드에
데이터 추가

6.3.3. 파이썬과 통합: 예제 #2

```
def getLinks(articleUrl):
    html = urlopen('http://en.wikipedia.org'+articleUrl)
    bs = BeautifulSoup(html, 'html.parser')
    title = bs.find('h1').get_text()
    content = bs.find('div', {'id':'mw-content-text'}).find('p').get_text()
    store(title, content)
    return bs.find('div', {'id':'bodyContent'}).find_all('a', href=re.compile('^(/wiki/)((?!:).)*$'))

links = getLinks('/wiki/Kevin_Bacon')
try:
    while len(links) > 0:
        newArticle = links[random.randint(0, len(links)-1)].attrs['href']
        print(newArticle)
        links = getLinks(newArticle)
finally:
    cur.close()
    conn.close()
```


실행 결과

MySQL에 저장된 결과 확인

- pages 테이블에 저장된 개수 확인: count(*)

```
select count(*) from pages;
```

```
count(*) |  
-----+  
      249 |
```

- page 테이블의 전체 데이터 보기

```
select * from pages;
```

	id	title	content	created
1	1	Test page title	This is some test page content. It can be up to 10,000 characters long.	2022-07-05 15:14:46
2	2	A new title	Some new content	2022-07-05 15:14:46
3	3	'Kevin Bacon'	'I'	2022-07-05 15:19:31
4	4	'Denver Film Festival'	'The Denver Film Festival is held in November, primarily at the Denver Film (2022-07-05 15:19:31
5	5	'Tippi Hedren'	'I'	2022-07-05 15:19:32
6	6	'Michael Jackson'	'I'	2022-07-05 15:19:34
7	7	'Kitty Wells'	'I'	2022-07-05 15:19:35
8	8	'Willie Nelson'	'I'	2022-07-05 15:19:37
9	9	'Shelly West'	'Shelly West (born May 23, 1958)[1] is an American country music singer. H	2022-07-05 15:19:38
10	10	'Grand Ole Opry'	'I'	2022-07-05 15:19:41
11	11	'Jim Ed Brown'	'I'	2022-07-05 15:19:42
12	12	'Billy Grammer'	'I'	2022-07-05 15:19:43
13	13	'Bobby Osborne'	'Bobby Osborne (born December 7, 1931) is an American bluegrass musica	2022-07-05 15:19:45
14	14	'Red Foley'	'I'	2022-07-05 15:19:45
15	15	'Rockabilly'	'I'	2022-07-05 15:19:46
16	16	'Medieval metal'	'I'	2022-07-05 15:19:47
17	17	'Music of the Cook Islands'	'The music of the Cook Islands is diverse. Christian music is extremely pop	2022-07-05 15:19:48
18	18	'Samoan literature'	'Samoan literature can be divided into oral (pre-colonial and post-colonial)	2022-07-05 15:19:49

6.3.5 여섯 다리와 MySQL

- wikipedia 데이터베이스 생성
 - pages, links 테이블 생성

```
# 새로운 wikipedia 데이터베이스 생성
```

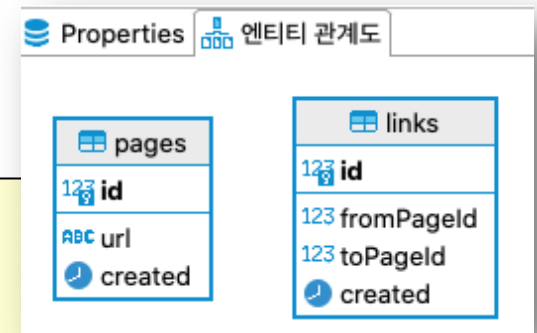
```
create database wikipedia;  
use wikipedia;
```

```
# pages 테이블 생성 (backtick `기호 생략 가능)
```

```
CREATE TABLE pages (  
    id INT NOT NULL AUTO_INCREMENT,  
    url VARCHAR(255) NOT NULL,  
    created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (id));
```

```
# links 테이블 생성
```

```
CREATE TABLE links (  
    id INT NOT NULL AUTO_INCREMENT,  
    fromPageId INT NULL,  
    toPageId INT NULL,  
    created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (id));
```



6.3.5 여섯 다리와 MySQL: 예제 #1

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import pymysql
from random import shuffle

conn = pymysql.connect(host='localhost',
                       user='사용자ID', passwd='비밀번호', db='wikipedia', charset='utf8')

cur = conn.cursor()

def insertPageIfNotExists(url):
    cur.execute('SELECT * FROM pages WHERE url = %s', (url))

    if cur.rowcount == 0:
        cur.execute('INSERT INTO pages (url) VALUES (%s)', (url))
        conn.commit()
        return cur.lastrowid
    else:
        return cur.fetchone()[0]
```

cur.rowcount
- select 명령어를
수행후 결과가
없으면 0 리턴

6.3.5 여섯 다리와 MySQL: 예제 #2

```
# pages 테이블에 있는 모든 데이터를 리스트에 담아 리턴
def loadPages():
    cur.execute('SELECT * FROM pages')
    pages = [row[1] for row in cur.fetchall()]
    return pages

# fromPageId와 toPageId로 검색이 되지 않으면 links 테이블에 링크 추가
def insertLink(fromPageId, toPageId):
    cur.execute('SELECT * FROM links WHERE fromPageId = %s AND toPageId = %s',
                (int(fromPageId), int(toPageId)))
    if cur.rowcount == 0:
        cur.execute('INSERT INTO links (fromPageId, toPageId) VALUES (%s, %s)',
                    (int(fromPageId), int(toPageId)))
        conn.commit()

# links 테이블에서 frompageId 필드에 pageId가 있는지 확인
def pageHasLinks(pageId):
    cur.execute('SELECT * FROM links WHERE fromPageId = %s', (int(pageId)))
    rowcount = cur.rowcount
    if rowcount == 0:
        return False
    return True
```

6.3.5 여섯 다리와 MySQL: 예제 #3

```
def getLinks(pageUrl, recursionLevel, pages):
```

```
    if recursionLevel > 4:  
        return
```

재귀호출의 단계를
5단계로 제한

```
    pageId = insertPageIfNotExists(pageUrl)  
    html = urlopen('http://en.wikipedia.org{}'.format(pageUrl))  
    bs = BeautifulSoup(html, 'html.parser')  
    links = bs.find_all('a', href=re.compile('^(/wiki/)((?!:).)*$'))  
    links = [link.attrs['href'] for link in links]
```

```
    for link in links:  
        linkId = insertPageIfNotExists(link)  
        insertLink(pageId, linkId)  
        if not pageHasLinks(linkId):  
            print("PAGE HAS NO LINKS: {}".format(link))  
            pages.append(link)  
            getLinks(link, recursionLevel + 1, pages)
```

getLinks()를 다시 호출:
재귀호출

```
getLinks('/wiki/Kevin_Bacon', 0, loadPages())  
cur.close()  
conn.close()
```

6.3.5 여섯 다리와 MySQL: 예제 #4

■ 실행 결과

- pages, links 테이블에 저장된 데이터 개수 계산

```
select count(*) from pages;
```

```
count(*)|  
-----+  
45812|
```

```
select count(*) from links;
```

```
count(*)|  
-----+  
64664|
```



Questions?