쇼핑몰 크롤링 DB에 저장

- 데이터베이스 생성

- 'create database beauty_shop'

In [2]:
```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
import pymysql
```

모든 제품을 크롤링해서 DB에 저장

- 저장할 DB부터 테이블 생성해서 형식에 맞게 data 변환 후 DB로 insert

1. DB 연결 코드
2. 필요한 DB 및 table 생성
3. 크롤링
   - 데이터를 table 형식에 맞게 정제
4. insert (데이터 생성)
5. commit() 해서 db에 반영
6. db 닫기

- db 테이블을 읽어와서 df 에 저장

1. DB연결 및 필요 객체 생성

In [52]:
```
def conn(d_name) :
    import pymysql
    host_name = 'localhost'
    host_port = 3306
    username = 'root'
    password = 'toor'
    database_name = d_name
    db = pymysql.connect(
        host=host_name,     # MySQL Server Address
        port=host_port,          # MySQL Server Port
        user=username,      # MySQL username
        passwd=password,    # password for MySQL username
        db=database_name,   # Database name
        charset='utf8'
    )
    return db
```

In [53]:
```
db = conn('beauty_shop')
cursor = db.cursor()
```

In [54]:
```
sql = "show databases"
cursor.execute(sql)
result = cursor.fetchall()
result
```

Out[54]:
```
(('beauty_shop',),
 ('classicmodels',),
 ('ecommerce',),
 ('employees',),
 ('information_schema',),
 ('mysql',),
 ('performance_schema',),
 ('sakila',),
 ('shopdb',),
 ('sqldb',),
 ('student_mgmt',),
 ('sys',),
 ('world',))
```

In [55]:
```
sql = "use beauty_shop"
cursor.execute(sql)
```

Out[55]:
```
0
```

In [56]:
```
sql = "select database()"
cursor.execute(sql)
result = cursor.fetchone()
result
```

Out[56]:
```
('beauty_shop',)
```

In [58]:
```
# table 생성
sql = '''
    CREATE TABLE product (
        PRODUCT_CODE int AUTO_INCREMENT NOT NULL,
        TITLE VARCHAR(200) NOT NULL,
        ORI_PRICE FLOAT,
        DISCOUNT_PRICE FLOAT,
        link VARCHAR(200),
        PRIMARY KEY(PRODUCT_CODE)
    );
'''

cursor.execute(sql)
db.commit()
```

In [59]:
```
sql = "show tables"
cursor.execute(sql)
result = cursor.fetchall()
result
```

Out[59]:
```
(('product',),)
```

In [60]:
```
sql = 'desc product'
cursor.execute(sql)
result = cursor.fetchall()
result
```

Out[60]:
```
(('PRODUCT_CODE', 'int', 'NO', 'PRI', None, 'auto_increment'),
 ('TITLE', 'varchar(200)', 'NO', '', None, ''),
 ('ORI_PRICE', 'float', 'YES', '', None, ''),
 ('DISCOUNT_PRICE', 'float', 'YES', '', None, ''),
 ('link', 'varchar(200)', 'YES', '', None, ''))
```

2. 크롤링 코드 - insert 구문을 추가해서 변경

In [62]:
```
# 크롤링 문서 요청해서 응답객체 반환(첫번째 크롤링 문서 요청 후 응답)
url="http://jolse.com/category/toners-mists/1019/"
html = urlopen(url)
htmls = html.read()
# print(htmls)
bs_obj = BeautifulSoup(htmls,"html.parser")
```

In [63]:
```
# box안에 들어 있는 1개의 상품에서 정보를 추출해서 dict형태로  반환하는 함수
# 데이터 전처리 : 제품명에 ' 제거/ 가격 USD 제거 / 세일가격 없는 경우 처리
def get_product_info(box) :
    p_tag = box.find("p",{"class":"name"})
    # 품목 추출
    span = p_tag.find("span")

    # 세부페이지링크 추출
    a = p_tag.find("a")
    sub_link = 'https://jolse.com' + a["href"]
    # 가격 추출 코드
    price_ul = box.find("ul")
    price_span = price_ul.findAll("span")

    # 데이터 전처리
    title = span.text.replace("'","''") # ' 처리
    ord_price = price_span[1].text.split(' ')[1] # USD 제거
    dis_price = price_span[-1].text.split(' ')[1]
    # 세일 가격이 없는 경우
    if dis_price =='' :
        dis_price = '0.0'


    # 최종 data 추출 후 반환
    return{"prd_name":title,"price":ord_price,"sale_price":dis_price,"sub_link":sub_link}
```

In [65]:
```
def save_data(prd_info) :
#     print(prd_info)

    # insert 구문
    sql = "INSERT INTO product (title, ori_price, discount_price,link) values('" ₩
        + prd_info["prd_name"] ₩
        +"'," ₩
        + prd_info['price']₩
        +"," ₩
        + prd_info['sale_price']₩
        +",'"₩
        + prd_info['sub_link']₩
        + "')"
    print(sql)
    cursor.execute(sql)
```

In [66]:
```
# 전달된 url 페이지에 접근해서 해당페지의 전체 상품 데이터를 추출 한 후 각 상품마다 get_product_info()함수를 호출해서
# 각 상품에대한 추출 정보를 받아옴 - 들어온 각 상품 정보를 리스트에 저장 한 수 해당 반환
def get_page_products(url) :
    url=url
    html = urlopen(url)
    htmls = html.read()
    # print(htmls)
    bs_obj = BeautifulSoup(htmls,"html.parser")

    ## 한 페이지에 모든 상품이 들어있는 ul 태그 추출
    # ul class:prdList grid4
    ul=bs_obj.find("ul",{"class":"prdList grid4"})
    ## 품목 1개를 담고 있는 div 태그 추출
    ## div class:box
    prd_boxes = ul.findAll("div", {"class":"box"}) #1개 페이지의 전체 상품
    # 반환되는 품목 데이터를 db에 insert  : 함수호출해서 진행
    for box in prd_boxes :
        prd = get_product_info(box)
#         print(prd)
        save_data(prd) #사용자 정의 함수(생성해야 함)
```

main 코드(프로그램 시작점)

In [67]:
```
from tqdm import tqdm_notebook # 상태바 표시

#여러 페이지의 화장품 정보를 추출해서 df 에 저장 후 csv에 저장하는 코드
url = "http://jolse.com/category/toners-mists/1019/?page=" #페이지 번호를 제외한 공통 url 문자열

last= int(bs_obj.find("p",{"class":"last"}).find("a")['href'].split("=")[1])

for i in tqdm_notebook(range(2,last+1)) : # 2페이지부터 수집
# for i in range(1,2) : # 1페이지 insert
    # url 완성 :page번호를 추가 i 변수 값을 활용
    urlfin =url + str(i)
    get_page_products(urlfin)
```

In [68]:
```
db.commit()
```

In [69]:
```
sql = "select * from product"
cursor.execute(sql)
result = cursor.fetchall()
result
```

```
 'WellDerma Sapphire Collagen Impact Mini Edition',
  28.0,
  19.6,
  'https://jolse.com/product/detail.html?product_no=43412&cate_no=1019&display_group=1'),
 (7,
  'Troiareuke ACSEN TOC Toner 100ml (Renewal)',
  43.0,
  32.25,
  'https://jolse.com/product/detail.html?product_no=43410&cate_no=1019&display_group=1'),
 (8,
  'MISSHA New Artemisia Calming Mist 120ml',
  34.0,
  27.17,
  'https://jolse.com/product/detail.html?product_no=43373&cate_no=1019&display_group=1'),
 (9,
  'so natural Yellow Cica Toner 260ml',
  24.5,
  0.0,
  'https://jolse.com/product/detail.html?product_no=43058&cate_no=1019&display_group=1'),
 (10,
```

In [70]:
```
db.close()
```

db 테이블에 저장된 데이터 df로 가져오기

In [71]:
```
db = conn('beauty_shop')
```

In [72]:
```
sql = "select * from product"
df = pd.read_sql(sql,db)
df
```

Out[72]:

	PRODUCT_CODE	TITLE	ORI_PRICE	DISCOUNT_PRICE	link
0	1	Abib Rebalancing Toner Skin Booster 200ml	28.50	17.10	https://jolse.com/product/detail.html?product_...
1	2	Rovectin Clean Lotus Water Calming Toner 200ml	22.00	18.70	https://jolse.com/product/detail.html?product_...
2	3	SWANICOCO More Mild Lotus Toner 200ml	24.00	20.40	https://jolse.com/product/detail.html?product_...
3	4	Acropass Trouble Cure Allanto-Cica Essence Ton...	18.00	11.70	https://jolse.com/product/detail.html?product_...
4	5	KTW Pure Rice Water Milky Toner Pad 70ea	24.00	20.40	https://jolse.com/product/detail.html?product_...
...	...	...	...	...	...
530	531	It's skin Aloe Relaxing Toner 150ml	19.00	16.15	https://jolse.com/product/detail.html?product_...
531	532	It's skin Hyaluronic Acid Moisture Toner 150ml	19.00	16.15	https://jolse.com/product/detail.html?product_...
532	533	Ciracle Base Toner pH5.6 105.5ml	23.48	17.59	https://jolse.com/product/detail.html?product_...
533	534	[Ciracle] Anti-Blemish Toner 105.5ml	22.25	16.67	https://jolse.com/product/detail.html?product_...
534	535	Ciracle Pore Control Tightening Toner 105.5ml	21.48	16.09	https://jolse.com/product/detail.html?product_...

535 rows × 5 columns

In [73]:
```
sql = "select title,ori_price,discount_price from product where discount_price > 15.00"
df = pd.read_sql(sql,db)
df
```

Out[73]:

	title	ori_price	discount_price
0	Abib Rebalancing Toner Skin Booster 200ml	28.50	17.10
1	Rovectin Clean Lotus Water Calming Toner 200ml	22.00	18.70
2	SWANICOCO More Mild Lotus Toner 200ml	24.00	20.40
3	KTW Pure Rice Water Milky Toner Pad 70ea	24.00	20.40
4	WellDerma Sapphire Collagen Impact Mini Edition	28.00	19.60
...	...	...	...
347	It's skin Aloe Relaxing Toner 150ml	19.00	16.15
348	It's skin Hyaluronic Acid Moisture Toner 150ml	19.00	16.15
349	Ciracle Base Toner pH5.6 105.5ml	23.48	17.59
350	[Ciracle] Anti-Blemish Toner 105.5ml	22.25	16.67
351	Ciracle Pore Control Tightening Toner 105.5ml	21.48	16.09

352 rows × 3 columns

In [ ]:

In [ ]: