

데이터 베이스 응용 과제

정보통계학과 20210853 이민희

제 1 부 – 1. MySQL과 파이썬 연동, 테이블 생성

```
In [1]: !pip install pymysql
```

Requirement already satisfied: pymysql in c:\users\minii\anaconda3\lib\site-packages (1.0.2)

```
In [2]: import pymysql
import pandas as pd
import numpy as np
```

```
In [3]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', charset='utf8')
cursor = conn.cursor()

cursor.execute("create database clothingshop")

conn.commit()
conn.close()
```

```
In [4]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql="CREATE TABLE product (pronumber integer not null primary key, proname varchar(20) not null, price integer not null, category
cursor.execute(sql)

conn.commit()
conn.close()
```

```
In [5]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql="INSERT INTO product (pronumber, proname, price, category, stock, manufacturer, textile, country_of_origin, manufacture_year)

val = [ (1, '반팔티 슬림핏', 10000, 1, 3, '난닝팔', '면 함유율: 100%', 1, 2015),
        (2, '반팔티 오버핏', 13000, 1, 6, '난닝팔', '면 함유율: 80%', 1, 2016),
        (3, '반팔티 기본핏', 12000, 1, 8, '난닝팔', '면 함유율: 100%', 1, 2017),

        (54, '반팔티 슬림핏', 11000, 1, 30, '유니클로', '면 함유율: 100%', 2, 2018),
        (4, '반팔티 오버핏', 13000, 1, None, '유니클로', '면 함유율: 100%', 2, 2019),
        (5, '반팔티 기본핏', 12000, 1, 80, '유니클로', '면 함유율: 100%', 2, 2020),

        (6, '반팔티 슬림핏', 90000, 1, 50, '포에버20', '면 함유율: 100%', 3, 2021),
        (7, '반팔티 오버핏', 10000, 1, 48, '포에버20', '면 함유율: 100%', 3, 2022),
        (8, '반팔티 기본핏', 11000, 1, None, '포에버20', '면 함유율: 100%', 5, 2023),

        (9, '긴팔티 슬림핏', 30000, 1, 20, '틀루망고', '폴리에스터 함유율: 100%', 3, 2024),
        (10, '긴팔티 오버핏', 33000, 1, 37, '틀루망고', '폴리에스터 함유율: 100%', 4, 2025),
        (11, '긴팔티 기본핏', 32000, 1, 56, '틀루망고', '폴리에스터 함유율: 100%', 4, 2026),

        (12, '롱스커트 데님', 35000, 2, 23, '포에버20', '면 함유율: 100%', 1, 2015),
        (13, '롱스커트 쉬폰', 36000, 2, 66, '포에버20', '폴리에스터 함유율: 100%', 3, 2016),
        (14, '롱스커트 실크', 36000, 2, 65, '포에버20', '실크 함유율: 100%', 5, 2014),
        (15, '롱스커트 면', 35600, 2, 50, '포에버20', '면 함유율: 100%', 1, 2017),
```

```
        (16, '롱스커트 데님', 35000, 2, 50, '자리', '면 함유율: 80%', 1, 2018),
        (17, '롱스커트 쉬폰', 80000, 2, 10, '자리', '폴리에스터 함유율: 100%', 1, 2019),
        (18, '롱스커트 면', 50000, 2, 50, '자리', '면 함유율: 50%', 1, 2020),

        (19, '미니스커트 데님', 25000, 2, 30, '자리', '면 함유율: 80%', 3, 2021),
        (20, '미니스커트 쉬폰', 40000, 2, 60, '자리', '폴리에스터 함유율: 100%', 4, 2022),
        (21, '미니스커트 실크', 45000, 2, 30, '자리', '실크 함유율: 40%', 1, 2023),
        (22, '미니스커트 면', 30000, 2, 50, '자리', '면 함유율: 50%', 2, 2024),

        (23, '스판 요가 레깅스', 23000, 2, 35, '난닝팔', '스판 함유율: 80%', 6, 2025),
        (24, '스판 헬스 레깅스', 43000, 2, 30, '틀루망고', '스판 함유율: 100%', 4, 2026),
        (25, '스판 레깅스', 33000, 2, 40, '유니클로', '스판 함유율: 100%', 2, 2027),
        (26, '스판 레깅스', 33000, 2, None, '자리', '스판 함유율: 100%', 1, 2020),

        (27, '슬림 조거팬츠', 27500, 2, 100, '난닝팔', '면 함유율: 50%', 5, 2019),
        (28, '오버 조거팬츠', 25500, 2, 35, '포에버20', '면 함유율: 50%', 1, 2018),
        (29, '기본 조거팬츠', 57500, 2, 25, '자리', '면 함유율: 80%', 7, 2022 ),
        (30, '조거팬츠', 27500, 2, 55, '미쵸', '면 함유율: 30%', 1, 2020),
        (31, '러닝용 조거팬츠', 45000, 2, 23, '틀루망고', '면 함유율: 80%', 7, 2020 ),
        (32, '조거팬츠', 30000, 2, 44, '유니클로', '면 함유율: 40%', 1, 2015),

        (33, '청바지 기본핏', 33000, 2, 50, '미쵸', '면 함유율: 100%', 6, 2016),
        (34, '청바지 슬림핏', 29000, 2, 53, '미쵸', '면 함유율: 100%', 6, 2017),
        (35, '청바지 부츠컷', 35000, 2, 49, '미쵸', '면 함유율: 100%', 1, 2018),
        (36, '청바지 기본핏', 30000, 2, 70, '유니클로', '면 함유율: 100%', 2, 2019),
        (37, '청바지 슬림핏', 24500, 2, 73, '유니클로', '면 함유율: 100%', 2, 2020),
        (38, '청바지 부츠컷', 30000, 2, 99, '유니클로', '면 함유율: 100%', 2, 2021),
```

```

(39, '기본핏 스웨터', 50000, 1, 30, '유니클로', '울 함유율: 100%', 5, 2022),
(40, '오버핏 스웨터', 30000, 1, None, '포에버20', '울 함유율: 40%', 5, 2023),
(41, '오버핏 스웨터', 34000, 1, 40, '미즈', '울 함유율: 30%', 1, 2024),
(42, '기본핏 스웨터', 33000, 1, 35, '유니클로', '울 함유율: 50%', 7, 2025),
(43, '기본핏 스웨터', 95000, 1, 33, '자리', '울 함유율: 80%', 7, 2026),

(44, '반팔티 슬림핏', 30000, 1, 33, '플루망고', '면 함유율: 80%', 7, 2020),
(45, '반팔티 오버핏', 35000, 1, 32, '플루망고', '면 함유율: 100%', 7, 2019),
(46, '반팔티 기본핏', 40000, 1, 26, '플루망고', '면 함유율: 70%', 5, 2020),

(47, '반팔티 슬림핏', 10000, 1, 60, '자리', '면 함유율: 80%', 4, 2021),
(48, '반팔티 오버핏', 10000, 1, 70, '자리', '면 함유율: 70%', 7, 2015),
(49, '반팔티 기본핏', 11000, 1, 73, '자리', '면 함유율: 60%', 5, 2016),

(50, '반팔티 슬림핏', 100000, 1, 30, '미즈', '면 함유율: 100%', 6, 2017),
(51, '반팔티 오버핏', 10000, 1, 38, '미즈', '면 함유율: 100%', 1, 2027),
(53, '롱스커트 실크', 90000, 2, None, '자리', '실크 함유율: 40%', 1, 2015),
(52, '반팔티 기본핏', 11000, 1, 33, '미즈', '면 함유율: 100%', 5, 2017)
]

cursor.executemany(sql, val)
print(cursor.rowcount, "개의 레코드가 입력되었습니다.")

```

54 개의 레코드가 입력되었습니다.

product 테이블을 생성하고, 값을 입력하였다.

```

In [6]: conn.commit()
        conn.close()

In [7]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
        cursor = conn.cursor()

        sql = "SELECT * FROM product"

        cursor.execute(sql)
        res = cursor.fetchall()

        for data in res:
            print(data)

        conn.commit()
        conn.close()

(1, '반팔티 슬림핏', 10000, 1, 3, '난닝팔', '면 함유율: 100%', 1, 2015)
(2, '반팔티 오버핏', 13000, 1, 6, '난닝팔', '면 함유율: 80%', 1, 2016)
(3, '반팔티 기본핏', 12000, 1, 8, '난닝팔', '면 함유율: 100%', 1, 2017)
(4, '반팔티 오버핏', 13000, 1, None, '유니클로', '면 함유율: 100%', 2, 2019)
(5, '반팔티 기본핏', 12000, 1, 80, '유니클로', '면 함유율: 100%', 2, 2020)
(6, '반팔티 슬림핏', 90000, 1, 50, '포에버20', '면 함유율: 100%', 3, 2021)
(7, '반팔티 오버핏', 10000, 1, 48, '포에버20', '면 함유율: 100%', 3, 2022)

(8, '반팔티 기본핏', 11000, 1, None, '포에버20', '면 함유율: 100%', 5, 2023)
(9, '긴팔티 슬림핏', 30000, 1, 20, '플루망고', '폴리에스터 함유율: 100%', 3, 2024)
(10, '긴팔티 오버핏', 33000, 1, 37, '플루망고', '폴리에스터 함유율: 100%', 4, 2025)
(11, '긴팔티 기본핏', 32000, 1, 56, '플루망고', '폴리에스터 함유율: 100%', 4, 2026)
(12, '롱스커트 데님', 35000, 2, 23, '포에버20', '면 함유율: 100%', 1, 2015)
(13, '롱스커트 쉬폰', 36000, 2, 66, '포에버20', '폴리에스터 함유율: 100%', 3, 2016)
(14, '롱스커트 실크', 36000, 2, 65, '포에버20', '실크 함유율: 100%', 5, 2014)
(15, '롱스커트 면', 35600, 2, 50, '포에버20', '면 함유율: 100%', 1, 2017)
(16, '롱스커트 데님', 35000, 2, 50, '자리', '면 함유율: 80%', 1, 2018)
(17, '롱스커트 쉬폰', 80000, 2, 10, '자리', '폴리에스터 함유율: 100%', 1, 2019)
(18, '롱스커트 면', 50000, 2, 50, '자리', '면 함유율: 50%', 1, 2020)
(19, '미니스커트 데님', 25000, 2, 30, '자리', '면 함유율: 80%', 3, 2021)
(20, '미니스커트 쉬폰', 40000, 2, 60, '자리', '폴리에스터 함유율: 100%', 4, 2022)
(21, '미니스커트 실크', 45000, 2, 30, '자리', '실크 함유율: 40%', 1, 2023)
(22, '미니스커트 면', 30000, 2, 50, '자리', '면 함유율: 50%', 2, 2024)
(23, '스판 요가 레깅스', 23000, 2, 35, '난닝팔', '스판 함유율: 80%', 6, 2025)
(24, '스판 헬스 레깅스', 43000, 2, 30, '플루망고', '스판 함유율: 100%', 4, 2026)
(25, '스판 레깅스', 33000, 2, 40, '유니클로', '스판 함유율: 100%', 2, 2027)
(26, '스판 레깅스', 33000, 2, None, '자리', '스판 함유율: 100%', 1, 2020)
(27, '슬림 조거팬츠', 27500, 2, 100, '난닝팔', '면 함유율: 50%', 5, 2019)
(28, '오버 조거팬츠', 25500, 2, 35, '포에버20', '면 함유율: 50%', 1, 2018)
(29, '기본 조거팬츠', 57500, 2, 25, '자리', '면 함유율: 80%', 7, 2022)
(30, '조거팬츠', 27500, 2, 55, '미즈', '면 함유율: 30%', 1, 2020)
(31, '러닝용 조거팬츠', 45000, 2, 23, '플루망고', '면 함유율: 80%', 7, 2020)
(32, '조거팬츠', 30000, 2, 44, '유니클로', '면 함유율: 40%', 1, 2015)
(33, '청바지 기본핏', 33000, 2, 50, '미즈', '면 함유율: 100%', 6, 2016)

```

```
(34, '청바지 슬림핏', 29000, 2, 53, '미조', '면 함유율: 100%', 6, 2017)
(35, '청바지 부츠컷', 35000, 2, 49, '미조', '면 함유율: 100%', 1, 2018)
(36, '청바지 기본핏', 30000, 2, 70, '유니클로', '면 함유율: 100%', 2, 2019)
(37, '청바지 슬림핏', 24500, 2, 73, '유니클로', '면 함유율: 100%', 2, 2020)
(38, '청바지 부츠컷', 30000, 2, 99, '유니클로', '면 함유율: 100%', 2, 2021)
(39, '기본핏 스웨터', 50000, 1, 30, '유니클로', '울 함유율: 100%', 5, 2022)
(40, '오버핏 스웨터', 30000, 1, None, '포에버20', '울 함유율: 40%', 5, 2023)
(41, '오버핏 슬림핏', 34000, 1, 40, '미조', '울 함유율: 30%', 1, 2024)
(42, '기본핏 스웨터', 33000, 1, 35, '유니클로', '울 함유율: 50%', 7, 2025)
(43, '기본핏 스웨터', 95000, 1, 33, '자리', '울 함유율: 80%', 7, 2026)
(44, '반팔티 슬림핏', 30000, 1, 33, '틀루망고', '면 함유율: 80%', 7, 2020)
(45, '반팔티 오버핏', 35000, 1, 32, '틀루망고', '면 함유율: 100%', 7, 2019)
(46, '반팔티 기본핏', 40000, 1, 26, '틀루망고', '면 함유율: 70%', 5, 2020)
(47, '반팔티 슬림핏', 10000, 1, 60, '자리', '면 함유율: 80%', 4, 2021)
(48, '반팔티 오버핏', 10000, 1, 70, '자리', '면 함유율: 70%', 7, 2015)
(49, '반팔티 기본핏', 11000, 1, 73, '자리', '면 함유율: 60%', 5, 2016)
(50, '반팔티 슬림핏', 100000, 1, 30, '미조', '면 함유율: 100%', 6, 2017)
(51, '반팔티 오버핏', 10000, 1, 38, '미조', '면 함유율: 100%', 1, 2027)
(52, '반팔티 기본핏', 11000, 1, 33, '미조', '면 함유율: 100%', 5, 2017)
(53, '롱스커트 실크', 90000, 2, None, '자리', '실크 함유율: 40%', 1, 2015)
(54, '반팔티 슬림핏', 11000, 1, 30, '유니클로', '면 함유율: 100%', 2, 2018)
```

```
In [8]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
        cursor = conn.cursor()

sql="CREATE TABLE user( usernumber integer primary key, userid varchar(20) not null, userpassword varchar(20) not null, username v
cursor.execute(sql)

conn.commit()
conn.close()
```

```
In [9]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
        cursor = conn.cursor()

sql= "INSERT INTO user( usernumber, userid, userpassword, username, useraddress, userphone) VALUES (%s, %s, %s, %s, %s, %s)"

val = [ (1, 'jisung1', 'qwerty', '박지성', '영국 맨체스터', '000-5000-0001'),
        (2, 'yuna2', '1235', '김연아', '대한민국 서울', '000-6000-0001'),
        (3, 'miran7', 'pass', '장미란', '대한민국 춘천', '000-7000-0001'),
        (4, 'sinsu5', 'werd', '추신수', '미국 클리블랜드', '000-8000-0001'),
        (5, 'heungmin', 'wordd', '송홍민', '영국 토트넘', '000-1235-1234'),
        (6, 'jisun3', 'qwerty', '박지선', '영국 맨체스터', '000-5000-0001'),
        (7, 'yua2', '12356', '김유아', '대한민국 서울', '000-3000-0001'),
        (8, 'misu7', 'passs', '장미수', '대한민국 강원도', '000-4000-0001'),
        (9, 'sinsung', 'werdd', '추신성', '미국 클리블랜드', '000-5000-0001'),
        (10, 'sejin', 'wordd', '박세진', '대한민국 대전', '000-1225-1234'),
        (11, 'minjoo', 'joo', '김민주', '대한민국 서울', '000-1299-0004'),
```

```
(12, 'junghee', 'hee', '이정희', '대한민국 부산', '000-2222-1234'),
(13, 'soojin', 'jin', '박수진', '캐나다 몬트리올', '000-5555-1234'),
(14, 'kkabii', 'bii', '김신신', '캐나다 퀘벡', '000-1001-1001'),
(15, 'euntak', 'tak', '지은탁', '캐나다 퀘벡', '000-1000-1000'),
(16, 'eunsang', 'sang', '차은상', '대한민국 서울', '000-7777-1234'),
(17, 'kimtan', 'tan', '김탄', '대한민국 대전', '000-4444-1234'),
(18, 'choiyoungdo', 'do', '최영도', '대한민국 서울', '000-1111-1234'),
(19, 'choibona', 'bona', '최보나', '대한민국 부산', '000-2222-1234'),
(20, 'myongsoo', 'soo', '박명수', '대한민국 대전', '000-9999-1234'),
(21, 'kimhana', 'hana', '김하나', '일본 도쿄', '000-1111-1111'),
(22, 'kimdooul', 'dooul', '김두울', '일본 오사카', '000-2222-2222'),
(23, 'kimsaet', 'saet', '김세엣', '중국 베이징', '000-3333-3333'),
(24, 'danaka', 'naka', '다나카', '일본 도쿄', '000-1225-1234'),
(25, 'blare', 'blaree', '블레어', '미국 뉴욕시티', '000-1222-1222'),
(26, 'yonsoo', 'soo', '김연수', '미국 엘에이', '000-1112-1112'),
(27, 'seojin', 'jin', '김서진', '베트남 하노이', '000-1232-1232'),
(28, 'sejoo', 'joo', '박세주', '대한민국 서울', '000-9999-1234'),
(29, 'sejin', 'worddd', '박세진', '대한민국 인천', '000-0000-9999'),
(30, 'seojin', 'worddddd', '박세진', '일본 도쿄', '000-0001-9999')
]

cursor.executemany(sql, val)
print(cursor.rowcount, "개의 레코드가 입력되었습니다.")
```

30 개의 레코드가 입력되었습니다.

user 테이블을 생성하고, 값을 입력하였다.

```
In [10]: conn.commit()
conn.close()
```

```
In [11]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM user"

cursor.execute(sql)
res = cursor.fetchall()

for data in res:
    print(data)

conn.commit()
conn.close()
```

```
(1, 'jisung1', 'qwerty', '박지성', '영국 맨체스터', '000-5000-0001')
(2, 'yuna2', '1235', '김연아', '대한민국 서울', '000-6000-0001')
(3, 'miran7', 'pass', '장미란', '대한민국 춘천', '000-7000-0001')
(4, 'sinsu5', 'werd', '추신수', '미국 클리블랜드', '000-8000-0001')
(5, 'heungmin', 'wordd', '손흥민', '영국 토트넘', '000-1235-1234')
(6, 'jisun3', 'qwerty', '박지선', '영국 맨체스터', '000-5000-0001')
(7, 'yua2', '12356', '김유아', '대한민국 서울', '000-3000-0001')
(8, 'misu7', 'passs', '장미수', '대한민국 강원도', '000-4000-0001')
(9, 'sinsung', 'werdd', '추신성', '미국 클리블랜드', '000-5000-0001')
(10, 'sejin', 'wordd', '박세진', '대한민국 대전', '000-1225-1234')

(11, 'minjoo', 'joo', '김민주', '대한민국 서울', '000-1299-0004')
(12, 'junghee', 'hee', '이정희', '대한민국 부산', '000-2222-1234')
(13, 'soojin', 'jin', '박수진', '캐나다 몬트리올', '000-5555-1234')
(14, 'kkabii', 'bii', '김신신', '캐나다 퀘벡', '000-1001-1001')
(15, 'euntak', 'tak', '지은탁', '캐나다 퀘벡', '000-1000-1000')
(16, 'eunsang', 'sang', '차은상', '대한민국 서울', '000-7777-1234')
(17, 'kintan', 'tan', '김탄', '대한민국 대전', '000-4444-1234')
(18, 'choiyoungdo', 'do', '최영도', '대한민국 서울', '000-1111-1234')
(19, 'choibona', 'bona', '최보나', '대한민국 부산', '000-2222-1234')
(20, 'myongsoo', 'soo', '박명수', '대한민국 대전', '000-9999-1234')
(21, 'kimhana', 'hana', '김하나', '일본 도쿄', '000-1111-1111')
(22, 'kindooul', 'dool', '김두울', '일본 오사카', '000-2222-2222')
(23, 'kimsaet', 'sat', '김세엣', '중국 베이징', '000-3333-3333')
(24, 'danaka', 'naka', '다나카', '일본 도쿄', '000-1225-1234')
(25, 'blare', 'blaree', '블레어', '미국 뉴욕시', '000-1222-1222')
(26, 'yonsoo', 'soo', '김연수', '미국 엘에이', '000-1112-1112')
(27, 'seojin', 'jin', '김서진', '베트남 하노이', '000-1232-1232')
(28, 'sejoo', 'joo', '박세주', '대한민국 서울', '000-9999-1234')
(29, 'sejin', 'worddd', '박세진', '대한민국 인천', '000-0000-9999')
(30, 'seojin', 'worddddd', '박서진', '일본 도쿄', '000-0001-9999')
```

```
In [12]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql="CREATE TABLE orders ( ordernumber integer, usernumber integer not null, pronumber integer not null, delname varchar(20) not null)"
cursor.execute(sql)

conn.commit()
conn.close()
```

```
In [13]: from datetime import datetime
now = datetime.now()
id = 1
formatted_date = now.strftime('%Y-%m-%d')

conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()
sql= "INSERT INTO orders ( ordernumber, usernumber, pronumber, delname, orderdate, payment) VALUES (%s, %s, %s, %s, %s, %s)"

val = [ (1, 1, 36, '한전택배', '2022-06-03', 'card'),
        (2, 1, 40, '한전택배', '2022-06-02', 'card'),
        (3, 2, 44, '한전택배', '2022-05-29', 'card'),
        (4, 3, 10, '대한택배', '2022-05-31', 'card'),
        (5, 4, 30, '대한택배', '2022-05-29', 'cash'),
        (6, 1, 33, '대한택배', '2022-05-29', 'cash'),
        (7, 4, 22, '대한택배', '2022-05-30', 'card'),
```

```
(8, 3, 15, '한전통운', '2022-06-01', 'cash'),
(9, 2, 50, '한전통운', '2022-05-30', 'card'),
(11, 30, 2, '한전통운', '2022-05-29', 'card'),
(12, 29, 22, '한전통운', '2022-05-29', 'card'),
(13, 28, 20, '한전통운', '2022-05-29', 'card'),
(14, 27, 30, '한전통운', '2022-05-29', 'card'),
(15, 26, 31, '한전통운', '2022-05-29', 'card'),
(16, 25, 12, '한전통운', '2022-05-29', 'card'),
(17, 24, 10, '한전통운', '2022-05-29', 'card'),
(18, 23, 9, '한전통운', '2022-05-29', 'card'),
(19, 22, 1, '한전통운', '2022-05-29', 'card'),
(20, 21, 22, '한전통운', '2022-05-29', 'card'),
(21, 20, 22, '한전통운', '2022-05-29', 'card'),
(22, 19, 11, '대한택배', '2022-05-31', 'card'),
(23, 18, 12, '대한택배', '2022-05-31', 'card'),
(24, 17, 13, '대한택배', '2022-05-31', 'card'),
(25, 16, 14, '대한택배', '2022-05-31', 'card'),
(26, 15, 22, '한전통운', '2022-05-29', 'cash'),
(27, 3, 10, '대한택배', '2022-05-31', 'card'),
(28, 14, 50, '대한택배', '2022-05-31', 'card'),
(29, 13, 49, '대한택배', '2022-05-31', 'card'),
(30, 12, 48, '대한택배', '2022-05-31', 'card'),
(31, 10, 47, '대한택배', '2022-05-31', 'card'),
(32, 9, 46, '대한택배', '2022-05-31', 'card'),
(33, 8, 30, '대한택배', '2022-05-31', 'card'),
(34, 7, 20, '대한택배', '2022-05-31', 'card'),
(35, 6, 13, '대한택배', '2022-05-31', 'card'),
```

```
(36, 5, 16, '대한택배', '2022-05-31', 'card'),
(37, 4, 19, '대한택배', '2022-05-31', 'card'),
(38, 3, 23, '대한택배', '2022-05-31', 'card'),
(39, 2, 30, '대한택배', '2022-05-31', 'card'),
(40, 29, 40, '대한택배', '2022-05-31', 'card'),
(41, 29, 41, '대한택배', '2022-05-31', 'card'),
(42, 29, 42, '대한택배', '2022-05-31', 'card'),
(43, 29, 43, '대한택배', '2022-05-31', 'card'),
(44, 29, 44, '대한택배', '2022-05-31', 'card'),
(45, 29, 45, '대한택배', '2022-05-31', 'card'),
(46, 29, 46, '대한택배', '2022-06-03', 'card'),
(47, 21, 22, '한전통운', '2022-05-29', 'card'),
(48, 27, 40, '한전통운', '2022-05-29', 'card'),
(49, 29, 51, '한전통운', '2022-05-29', 'card'),
(50, 29, 52, '한전통운', '2022-05-29', 'card'),
(51, 30, 52, '한전통운', '2022-05-29', 'card')
]
```

```
cursor.executemany(sql, val)
print(cursor.rowcount, "개의 레코드가 입력되었습니다.")
```

50 개의 레코드가 입력되었습니다.

```
In [14]: conn.commit()
conn.close()
```

orders 테이블을 생성하고, 값을 입력하였다.

```
In [15]: conn = pymysql.connect(host='localhost', user='root', password='oracleMini@0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM orders"

cursor.execute(sql)
res = cursor.fetchall()

for data in res:
    print(data)

conn.commit()
conn.close()
```

```
(1, 1, 36, '한전택배', datetime.date(2022, 6, 3), 'card')
(2, 1, 40, '한전택배', datetime.date(2022, 6, 2), 'card')
(3, 2, 44, '한전택배', datetime.date(2022, 5, 29), 'card')
(4, 3, 10, '대한택배', datetime.date(2022, 5, 31), 'card')
(5, 4, 30, '대한택배', datetime.date(2022, 5, 29), 'cash')
(6, 1, 33, '대한택배', datetime.date(2022, 5, 29), 'cash')
(7, 4, 22, '대한택배', datetime.date(2022, 5, 30), 'card')
(8, 3, 15, '한전통운', datetime.date(2022, 6, 1), 'cash')
(9, 2, 50, '한전통운', datetime.date(2022, 5, 30), 'card')
(11, 30, 2, '한전통운', datetime.date(2022, 5, 29), 'card')
(12, 29, 22, '한전통운', datetime.date(2022, 5, 29), 'card')
(13, 28, 20, '한전통운', datetime.date(2022, 5, 29), 'card')
(14, 27, 30, '한전통운', datetime.date(2022, 5, 29), 'card')
```

```
(15, 26, 31, '한전통운', datetime.date(2022, 5, 29), 'card')
(16, 25, 12, '한전통운', datetime.date(2022, 5, 29), 'card')
(17, 24, 10, '한전통운', datetime.date(2022, 5, 29), 'card')
(18, 23, 9, '한전통운', datetime.date(2022, 5, 29), 'card')
(19, 22, 1, '한전통운', datetime.date(2022, 5, 29), 'card')
(20, 21, 22, '한전통운', datetime.date(2022, 5, 29), 'card')
(21, 20, 22, '한전통운', datetime.date(2022, 5, 29), 'card')
(22, 19, 11, '대한택배', datetime.date(2022, 5, 31), 'card')
(23, 18, 12, '대한택배', datetime.date(2022, 5, 31), 'card')
(24, 17, 13, '대한택배', datetime.date(2022, 5, 31), 'card')
(25, 16, 14, '대한택배', datetime.date(2022, 5, 31), 'card')
(26, 15, 22, '한전통운', datetime.date(2022, 5, 29), 'cash')
(27, 3, 10, '대한택배', datetime.date(2022, 5, 31), 'card')
(28, 14, 50, '대한택배', datetime.date(2022, 5, 31), 'card')
(29, 13, 49, '대한택배', datetime.date(2022, 5, 31), 'card')
(30, 12, 48, '대한택배', datetime.date(2022, 5, 31), 'card')
(31, 10, 47, '대한택배', datetime.date(2022, 5, 31), 'card')
(32, 9, 46, '대한택배', datetime.date(2022, 5, 31), 'card')
(33, 8, 30, '대한택배', datetime.date(2022, 5, 31), 'card')
(34, 7, 20, '대한택배', datetime.date(2022, 5, 31), 'card')
(35, 6, 13, '대한택배', datetime.date(2022, 5, 31), 'card')
(36, 5, 16, '대한택배', datetime.date(2022, 5, 31), 'card')
(37, 4, 19, '대한택배', datetime.date(2022, 5, 31), 'card')
(38, 3, 23, '대한택배', datetime.date(2022, 5, 31), 'card')
(39, 2, 30, '대한택배', datetime.date(2022, 5, 31), 'card')
(40, 29, 40, '대한택배', datetime.date(2022, 5, 31), 'card')
(41, 29, 41, '대한택배', datetime.date(2022, 5, 31), 'card')

(42, 29, 42, '대한택배', datetime.date(2022, 5, 31), 'card')
(43, 29, 43, '대한택배', datetime.date(2022, 5, 31), 'card')
(44, 29, 44, '대한택배', datetime.date(2022, 5, 31), 'card')
(45, 29, 45, '대한택배', datetime.date(2022, 5, 31), 'card')
(46, 29, 46, '대한택배', datetime.date(2022, 6, 3), 'card')
(47, 21, 22, '한전통운', datetime.date(2022, 5, 29), 'card')
(48, 27, 40, '한전통운', datetime.date(2022, 5, 29), 'card')
(49, 29, 51, '한전통운', datetime.date(2022, 5, 29), 'card')
(50, 29, 52, '한전통운', datetime.date(2022, 5, 29), 'card')
(51, 30, 52, '한전통운', datetime.date(2022, 5, 29), 'card')
```

MySQL과 파이썬 연동한 후, database clothingshop 을 생성하고, 테이블 product, user, orders를 생성하였다.

제 1부 - 2. CRUD 연산을 파이썬에서 실행

Select문을 활용한 검색 1) 재고가 50인 상품 검색하였다.

```
In [16]: # select문을 활용한 검색 1)
conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM product where stock = %s"
cursor.execute(sql, (50))

res = cursor.fetchall()

for data in res:
    print(data)

conn.commit()
conn.close()

(6, '반팔티 슬림핏', 90000, 1, 50, '포에버20', '면 함유율: 100%', 3, 2021)
(15, '롱스커트 면', 35600, 2, 50, '포에버20', '면 함유율: 100%', 1, 2017)
(16, '롱스커트 데님', 35000, 2, 50, '자리', '면 함유율: 80%', 1, 2018)
(18, '롱스커트 면', 50000, 2, 50, '자리', '면 함유율: 50%', 1, 2020)
(22, '미니스커트 면', 30000, 2, 50, '자리', '면 함유율: 50%', 2, 2024)
(33, '청바지 기본핏', 33000, 2, 50, '미쵸', '면 함유율: 100%', 6, 2016)
```

Select문을 활용한 검색 2) 제조사가 룰루망고인 상품 검색하였다.

```
In [17]: # select문을 활용한 검색 2)
conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM product where manufacturer=%s"
cursor.execute(sql, ("톨루망고"))

res = cursor.fetchall()

for data in res:
    print(data)

conn.commit()
conn.close()

(9, '긴팔티 슬림핏', 30000, 1, 20, '톨루망고', '폴리에스터 함유율: 100%', 3, 2024)
(10, '긴팔티 오버핏', 33000, 1, 37, '톨루망고', '폴리에스터 함유율: 100%', 4, 2025)
(11, '긴팔티 기본핏', 32000, 1, 56, '톨루망고', '폴리에스터 함유율: 100%', 4, 2026)
(24, '스판 헬스 레깅스', 43000, 2, 30, '톨루망고', '스판 함유율: 100%', 4, 2026)
(31, '러닝용 조거팬츠', 45000, 2, 23, '톨루망고', '면 함유율: 80%', 7, 2020)
(44, '반팔티 슬림핏', 30000, 1, 33, '톨루망고', '면 함유율: 80%', 7, 2020)
(45, '반팔티 오버핏', 35000, 1, 32, '톨루망고', '면 함유율: 100%', 7, 2019)
(46, '반팔티 기본핏', 40000, 1, 26, '톨루망고', '면 함유율: 70%', 5, 2020)
```

Select문을 활용한 검색 3) where절 조건으로 2개 이상 매개변수 받는 예제.

제조사가 유니쿨로이고, 카테고리가 1인 상품 검색하였다.

```
In [18]: # select문을 활용한 검색. where절 조건으로 2개 이상 매개변수 받는 예제 3)
conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM product where manufacturer = %s and category = %s"
cursor.execute(sql, ("유니쿨로", 1))
res = cursor.fetchall()
for data in res:
    print(data)

(4, '반팔티 오버핏', 13000, 1, None, '유니쿨로', '면 함유율: 100%', 2, 2019)
(5, '반팔티 기본핏', 12000, 1, 80, '유니쿨로', '면 함유율: 100%', 2, 2020)
(39, '기본핏 스웨터', 50000, 1, 30, '유니쿨로', '울 함유율: 100%', 5, 2022)
(42, '기본핏 스웨터', 33000, 1, 35, '유니쿨로', '울 함유율: 50%', 7, 2025)
(54, '반팔티 슬림핏', 11000, 1, 30, '유니쿨로', '면 함유율: 100%', 2, 2018)
```

```
In [19]: conn.commit()
conn.close()
```

Update문 1)

고객이름이 박명수인 고객의 고객주소를 중국 상하이로 변경하였다.

```
In [20]: #update문
conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "UPDATE user SET useraddress = %s WHERE username = %s "
cursor.execute(sql, ("중국 상하이", "박명수"))

conn.commit()
conn.close()
```

```
In [21]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM user WHERE username = %s "

cursor.execute(sql, ("박명수"))
res = cursor.fetchall()
for data in res:
    print(data)

(20, 'myongsoo', 'soo', '박명수', '중국 상하이', '000-9999-1234')
```

Delete문 1)

상품 테이블에서 상품번호가 54인 상품 삭제하였다.

```
In [22]: #delete문
conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "DELETE FROM product WHERE pronumber = %s"
cursor.execute(sql, (54))
conn.commit()

conn.close()
```

```
In [23]: conn = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**', db='clothingshop', charset='utf8')
cursor = conn.cursor()

sql = "SELECT * FROM product"

cursor.execute(sql)
res = cursor.fetchall()
for data in res:
    print(data)
```

```
(1, '반팔티 슬림핏', 10000, 1, 3, '난닝팔', '면 함유율: 100%', 1, 2015)
(2, '반팔티 오버핏', 13000, 1, 6, '난닝팔', '면 함유율: 80%', 1, 2016)
(3, '반팔티 기본핏', 12000, 1, 8, '난닝팔', '면 함유율: 100%', 1, 2017)
(4, '반팔티 오버핏', 13000, 1, None, '유니클로', '면 함유율: 100%', 2, 2019)
(5, '반팔티 기본핏', 12000, 1, 80, '유니클로', '면 함유율: 100%', 2, 2020)
(6, '반팔티 슬림핏', 90000, 1, 50, '포에버20', '면 함유율: 100%', 3, 2021)
(7, '반팔티 오버핏', 10000, 1, 48, '포에버20', '면 함유율: 100%', 3, 2022)

(8, '반팔티 기본핏', 11000, 1, None, '포에버20', '면 함유율: 100%', 5, 2023)
(9, '긴팔티 슬림핏', 30000, 1, 20, '틀루망고', '폴리에스터 함유율: 100%', 3, 2024)
(10, '긴팔티 오버핏', 33000, 1, 37, '틀루망고', '폴리에스터 함유율: 100%', 4, 2025)
(11, '긴팔티 기본핏', 32000, 1, 56, '틀루망고', '폴리에스터 함유율: 100%', 4, 2026)
(12, '롱스커트 데님', 35000, 2, 23, '포에버20', '면 함유율: 100%', 1, 2015)
(13, '롱스커트 쉬폰', 36000, 2, 66, '포에버20', '폴리에스터 함유율: 100%', 3, 2016)
(14, '롱스커트 실크', 36000, 2, 65, '포에버20', '실크 함유율: 100%', 5, 2014)
(15, '롱스커트 면', 35600, 2, 50, '포에버20', '면 함유율: 100%', 1, 2017)
(16, '롱스커트 데님', 35000, 2, 50, '자리', '면 함유율: 80%', 1, 2018)
(17, '롱스커트 쉬폰', 80000, 2, 10, '자리', '폴리에스터 함유율: 100%', 1, 2019)
(18, '롱스커트 면', 50000, 2, 50, '자리', '면 함유율: 50%', 1, 2020)
(19, '미니스커트 데님', 25000, 2, 30, '자리', '면 함유율: 80%', 3, 2021)
(20, '미니스커트 쉬폰', 40000, 2, 60, '자리', '폴리에스터 함유율: 100%', 4, 2022)
(21, '미니스커트 실크', 45000, 2, 30, '자리', '실크 함유율: 40%', 1, 2023)
(22, '미니스커트 면', 30000, 2, 50, '자리', '면 함유율: 50%', 2, 2024)
(23, '스판 요가 레깅스', 23000, 2, 35, '난닝팔', '스판 함유율: 80%', 6, 2025)
(24, '스판 헬스 레깅스', 43000, 2, 30, '틀루망고', '스판 함유율: 100%', 4, 2026)
(25, '스판 레깅스', 33000, 2, 40, '유니클로', '스판 함유율: 100%', 2, 2027)
(26, '스판 레깅스', 33000, 2, None, '자리', '스판 함유율: 100%', 1, 2020)
(27, '슬림 조거팬츠', 27500, 2, 100, '난닝팔', '면 함유율: 50%', 5, 2019)
(28, '오버 조거팬츠', 25500, 2, 35, '포에버20', '면 함유율: 50%', 1, 2018)
(29, '기본 조거팬츠', 57500, 2, 25, '자리', '면 함유율: 80%', 7, 2022)
(30, '조거팬츠', 27500, 2, 55, '미즈', '면 함유율: 30%', 1, 2020)
(31, '러닝용 조거팬츠', 45000, 2, 23, '틀루망고', '면 함유율: 80%', 7, 2020)
(32, '조거팬츠', 30000, 2, 44, '유니클로', '면 함유율: 40%', 1, 2015)
(33, '청바지 기본핏', 33000, 2, 50, '미즈', '면 함유율: 100%', 6, 2016)
(34, '청바지 슬림핏', 29000, 2, 53, '미즈', '면 함유율: 100%', 6, 2017)

(35, '청바지 부츠컷', 35000, 2, 49, '미즈', '면 함유율: 100%', 1, 2018)
(36, '청바지 기본핏', 30000, 2, 70, '유니클로', '면 함유율: 100%', 2, 2019)
(37, '청바지 슬림핏', 24500, 2, 73, '유니클로', '면 함유율: 100%', 2, 2020)
(38, '청바지 부츠컷', 30000, 2, 99, '유니클로', '면 함유율: 100%', 2, 2021)
(39, '기본핏 스웨터', 50000, 1, 30, '유니클로', '울 함유율: 100%', 5, 2022)
(40, '오버핏 스웨터', 30000, 1, None, '포에버20', '울 함유율: 40%', 5, 2023)
(41, '오버핏 스웨터', 34000, 1, 40, '미즈', '울 함유율: 30%', 1, 2024)
(42, '기본핏 스웨터', 33000, 1, 35, '유니클로', '울 함유율: 50%', 7, 2025)
(43, '기본핏 스웨터', 95000, 1, 33, '자리', '울 함유율: 80%', 7, 2026)
(44, '반팔티 슬림핏', 30000, 1, 33, '틀루망고', '면 함유율: 80%', 7, 2020)
(45, '반팔티 오버핏', 35000, 1, 32, '틀루망고', '면 함유율: 100%', 7, 2019)
(46, '반팔티 기본핏', 40000, 1, 26, '틀루망고', '면 함유율: 70%', 5, 2020)
(47, '반팔티 슬림핏', 10000, 1, 60, '자리', '면 함유율: 80%', 4, 2021)
(48, '반팔티 오버핏', 10000, 1, 70, '자리', '면 함유율: 70%', 7, 2015)
(49, '반팔티 기본핏', 11000, 1, 73, '자리', '면 함유율: 60%', 5, 2016)
(50, '반팔티 슬림핏', 100000, 1, 30, '미즈', '면 함유율: 100%', 6, 2017)
(51, '반팔티 오버핏', 10000, 1, 38, '미즈', '면 함유율: 100%', 1, 2027)
(52, '반팔티 기본핏', 11000, 1, 33, '미즈', '면 함유율: 100%', 5, 2017)
(53, '롱스커트 실크', 90000, 2, None, '자리', '실크 함유율: 40%', 1, 2015)
```

제 2 부 – 1. 데이터 전처리

범주형 자료 처리 1)

product 테이블을 데이터프레임 형태로 변환한 후, category==1은 상의로 category==2은 하의로 변환해 주었다.

```
In [1]: import pandas as pd
import numpy as np
import pymysql

In [2]: con = pymysql.connect(host='localhost', user='root', password='oracleMinirene0522**',
db='clothingshop', charset='utf8', # 한글처리 (charset = 'utf8')
autocommit=True, # 결과 DB 반영 (Insert or update) ##그 셀이 정상적으로 동작하면 자동으로 commit을 해준다.
##autocommit = True가 일반적인 것은 아님. rollback이 불가능하기 때문.
cursorclass=pymysql.cursors.DictCursor # DB조회시 컬럼명을 동시에 보여줌
)
cur = con.cursor()

In [3]: sql = "SELECT * FROM product"
cur.execute(sql)
res = cur.fetchall()

cur.execute(sql)
res = cur.fetchall()
for data in res:
    print(data)
```

```
In [4]: product = pd.DataFrame(res)
product
```

```
Out [4]:
```

	pronumber	proname	price	category	stock	manufacturer	textile	country_of_origin	manufacture_year
0	1	반팔티 슬림핏	10000	1	3.0	난닝팔	면 함유율: 100%	1	2015
1	2	반팔티 오버핏	13000	1	6.0	난닝팔	면 함유율: 80%	1	2016
2	3	반팔티 기본핏	12000	1	8.0	난닝팔	면 함유율: 100%	1	2017

```
In [5]: import matplotlib.pyplot as plt
plt.rc('font', family='Malgun Gothic')
plt.rcParams["font.size"] = 12
plt.rcParams['xtick.labelsize'] = 12.
plt.rcParams['ytick.labelsize'] = 12.
plt.rcParams['axes.unicode_minus'] = False
```

```
In [6]: ## 데이터 전처리: 범주형 자료 처리 1 #####
product.loc[product['category'] == 1, ['category']] = '상의'
product.loc[product['category'] == 2, ['category']] = '하의'
```

```
In [7]: print(product)
```

	pronumber	proname	price	category	stock	manufacturer	#
0	1	반팔티 슬림핏	10000	상의	3.0	난닝팔	
1	2	반팔티 오버핏	13000	상의	6.0	난닝팔	
2	3	반팔티 기본핏	12000	상의	8.0	난닝팔	
3	4	반팔티 오버핏	13000	상의	NaN	유니클로	
4	5	반팔티 기본핏	12000	상의	80.0	유니클로	
5	6	반팔티 슬림핏	90000	상의	50.0	포에버20	
6	7	반팔티 오버핏	10000	상의	48.0	포에버20	
7	8	반팔티 기본핏	11000	상의	NaN	포에버20	
8	9	긴팔티 슬림핏	30000	상의	20.0	틀루랑고	
9	10	긴팔티 오버핏	33000	상의	37.0	틀루랑고	
10	11	긴팔티 기본핏	32000	상의	56.0	틀루랑고	
11	12	롱스커트 데님	35000	하의	23.0	포에버20	
12	13	롱스커트 쉬폰	36000	하의	66.0	포에버20	

범주형 자료 처리 2)

product 테이블을 데이터프레임 형태로 변환한 후, origin_of_country== 1, 2, 3, 4, 5, 6, 7 을 각각 맞는 국가 이름 중국, 일본, 미국, 캐나다, 말레이시아, 대한민국, 홍콩으로 변환해주었다.

```
In [8]: > ### 데이터 전처리: 범주형 자료 처리 2 #####
print(product['country_of_origin'].unique())

[1 2 3 5 4 6 7]
```

```
In [9]: > def country_of_origin_txt( x ):
    if int(x) == 1 :
        return '중국'
    elif int(x) == 2 :
        return '일본'
    elif int(x) == 3 :
        return '미국'
    elif int(x) == 4 :
        return '캐나다'
    elif int(x) == 5 :
        return '말레이시아'
    elif int(x) == 6 :
        return '대한민국'
    elif int(x) == 7 :
        return '홍콩'
```

```
In [10]: > product['country_of_origin'] = product['country_of_origin'].apply(country_of_origin_txt)
```

```
In [11]: > print(product)
```

Null값 대치 1)

재고의 결측치를 재고의 평균값으로 대치하였다.

```
In [12]: > ### 데이터 전처리: Null값 대치 #####
print('\n# 재고 결측치 개수 구하기 before')
print(sum(product['stock'].isnull()))
```

```
# 재고 결측치 개수 구하기 before
5
```

```
In [13]: > product.loc[product['stock'].isnull(), 'stock'] = product['stock'].mean() #null인 행들을 찾아서 평균으로 채워준다.
print('\n# 재고 결측치 개수 구하기 after')
print(sum(product['stock'].isnull()))
```

```
# 재고 결측치 개수 구하기 after
0
```

범주형 변환 1)

```
In [14]: > sql = "SELECT * FROM user"
cur.execute(sql)
res = cur.fetchall()

cur.execute(sql)
res = cur.fetchall()
for data in res:
    print(data)

{'usernumber': 1, 'userid': 'jisung1', 'userpassword': 'qwert', 'username': '박지성', 'useraddress': '영국 맨체스터', 'userphone': '000-5000-0001'}
{'usernumber': 2, 'userid': 'yuna2', 'userpassword': '1235', 'username': '김연아', 'useraddress': '대한민국 서울', 'userphone': '000-6000-0001'}
{'usernumber': 3, 'userid': 'miran7', 'userpassword': 'pass', 'username': '장미란', 'useraddress': '대한민국 춘천', 'userphone': '000-7000-0001'}
{'usernumber': 4, 'userid': 'sinsu5', 'userpassword': 'werd', 'username': '추신수', 'useraddress': '미국 클리블랜드', 'userphone': '000-8000-0001'}
{'usernumber': 5, 'userid': 'heungmin', 'userpassword': 'worrd', 'username': '손흥민', 'useraddress': '영국 토트넘', 'userphone': '000-1235-1234'}
{'usernumber': 6, 'userid': 'jisun3', 'userpassword': 'qwerty', 'username': '박지선', 'useraddress': '영국 맨체스터', 'userphone': '000-5000-0001'}
```

```
In [15]: > user = pd.DataFrame(res)
user
```

```
Out[15]:
```

	usernumber	userid	userpassword	username	useraddress	userphone
0	1	jisung1	qwert	박지성	영국 맨체스터	000-5000-0001
1	2	yuna2	1235	김연아	대한민국 서울	000-6000-0001
2	3	miran7	pass	장미란	대한민국 춘천	000-7000-0001
3	4	sinsu5	werd	추신수	미국 클리블랜드	000-8000-0001
4	5	heungmin	worrd	손흥민	영국 토트넘	000-1235-1234
5	6	jisun3	qwerty	박지선	영국 맨체스터	000-5000-0001

```
In [16]: # 고객들이 어느 국가에 살고 있는지 범주화.
```

```
def country(x):
    if '대한민국' in x:
        return '대한민국'
    elif '중국' in x:
        return '중국'
    elif '미국' in x:
        return '미국'
    elif '일본' in x:
        return '일본'
    elif '캐나다' in x:
        return '캐나다'
    elif '영국' in x:
        return '영국'
    else:
        return '기타'
```

```
In [17]: user['country'] = user['useraddress'].apply(country)
```

```
print(user[['useraddress', 'country']].head())
```

```
useraddress country
0   영국 맨체스터   영국
1  대한민국 서울  대한민국
2  대한민국 춘천  대한민국
3   미국 클리블랜드  미국
4   영국 토트넘   영국
```

고객들의 주소를 활용하여 각 고객들이 어느 국가에 살고 있는지 범주화 하였다.

제 3 부 – 1. 데이터 탐색

각 컬럼 이름 한국어로 바꾸었다.

```
In [18]: col_mapping = {'pronumber':'제품번호', 'praname':'제품이름', 'price':'가격', 'category':'카테고리', 'stock':'재고', 'manufacturer':
product = product.rename(columns = col_mapping)
```

```
In [19]: product
```

```
Out[19]:
```

	제품번호	제품이름	가격	카테고리	재고	제조사	소재	제조국	생산년도
0	1	반팔티 슬림핏	10000	상의	3.000	난닝팔	면 함유율: 100%	중국	2015
1	2	반팔티 오버핏	13000	상의	6.000	난닝팔	면 함유율: 80%	중국	2016
2	3	반팔티 쿨링핏	12000	상의	5.000	난닝팔	면 함유율: 100%	중국	2017

```
In [20]: col_mapping = {'usernumber':'고객번호', 'userid':'아이디', 'userpassword':'비밀번호', 'username':'고객이름', 'useraddress':'고객주
user = user.rename(columns = col_mapping)
```

```
In [21]: user
```

```
Out[21]:
```

	고객번호	아이디	비밀번호	고객이름	고객주소	고객번호	국가
0	1	jsung1	qwert	박지성	영국 맨체스터	000-5000-0001	영국
1	2	yuna2	1235	김연아	대한민국 서울	000-6000-0001	대한민국
2	3	miran7	pass	장미란	대한민국 춘천	000-7000-0001	대한민국
3	4	einer5	ward	츠시즈	미국 클리블랜드	000-8000-0001	미국

```
In [22]: sql = "SELECT * FROM orders"
```

```
cur.execute(sql)
res = cur.fetchall()
```

```
cur.execute(sql)
res = cur.fetchall()
for data in res:
    print(data)
```

```
(('orderid': 1, 'userid': 1, 'productid': 30, 'orderid': '한편재발', 'orderdate': datetime.date(2020, 6, 9), 'amount': 10
```

```
In [23]: orders = pd.DataFrame(res)
orders

Out[23]:
```

	orderid	userid	productid	productname	orderdate	payment
0	1	1	36	한전택배	2022-06-03	card
1	2	1	40	한전택배	2022-06-02	card

```
In [24]: col_mapping = {'orderid': '주문번호', 'userid': '고객번호', 'productid': '제품번호', 'productname': '배송사', 'orderdate': '주문날짜', 'payment': '결제'}
orders = orders.rename(columns=col_mapping)

In [25]: orders

Out[25]:
```

	주문번호	고객번호	제품번호	배송사	주문날짜	결제
0	1	1	36	한전택배	2022-06-03	card
1	2	1	40	한전택배	2022-06-02	card
2	3	2	44	한전택배	2022-05-29	card
3	4	2	40	한전택배	2022-05-31	card

바그래프 1) 제조국 빈도

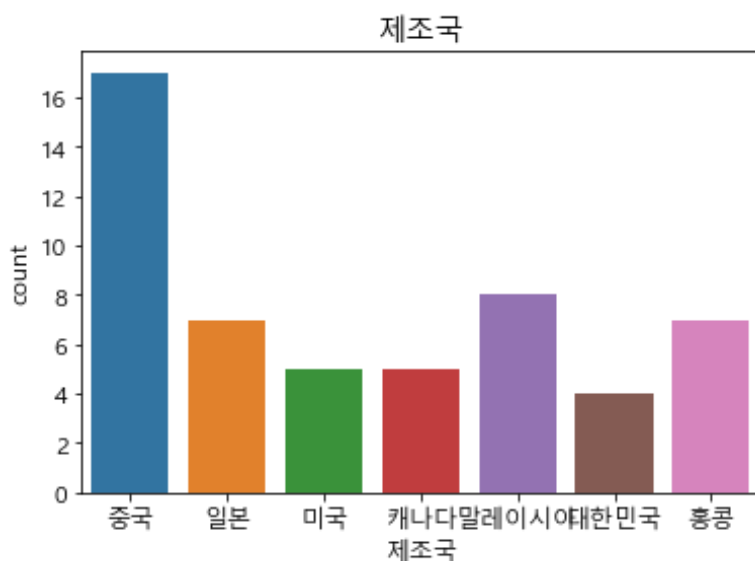
```
In [26]: import matplotlib.pyplot as plt #간단한 통계분석 추가가 시분(matplotlib 베이스). 기본으로 사용할 리소스를 정리해놓음,
plt.rc('font', family='Malgun Gothic')
plt.rcParams["font.size"] = 12
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
plt.rcParams['axes.unicode_minus'] = False

In [27]: #막대그래프
print('제조국 빈도')
result = product.groupby('제조국')['제조국'].count() #sql 그룹바이와 비슷.
print(result)

ORDERING = ['중국', '일본', '미국', '캐나다', '말레이시아', '대한민국', '홍콩']
plt.figure()
plt.title('제조국')

import seaborn as sns
sns.countplot(x='제조국', data=product, order=ORDERING) #order 막대기 배치 순서
savefile="bar_제조국.png"
plt.savefig(savefile, dpi=400)

제조국 빈도
제조국
대한민국    4
말레이시아    8
미국    5
일본    7
중국    17
캐나다    5
홍콩    7
Name: 제조국, dtype: int64
```



분석: 상품의 제조국 중에서는 중국이 가장 많고, 대한민국이 가장 적다.

파이그래프 1) 고객국가 빈도

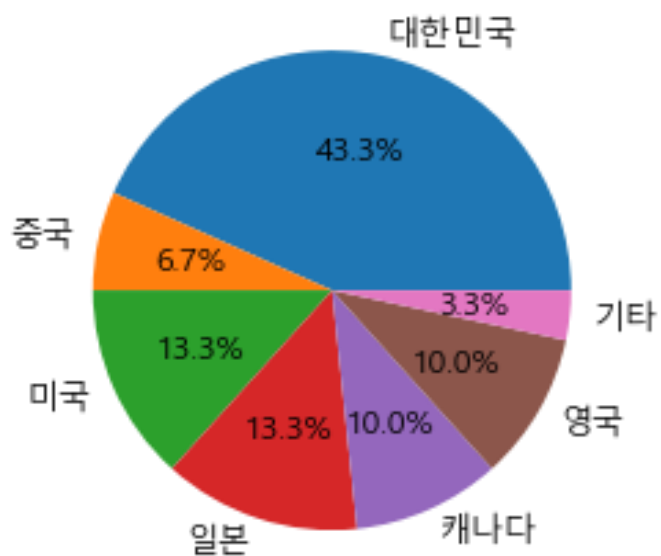
```
In [28]: #파이그래프
import matplotlib

from matplotlib import font_manager, rc
font_path = "C:/windows/Fonts/maibun.ttf"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)
```

```
In [29]: print('고객국가빈도')
result = user.groupby('국가')['국가'].count()
print(result)

plt.figure()
data = pd.Series([13/30,2/30,4/30,4/30,3/30,3/30, 1/30], index=['대한민국', '중국', '미국', '일본', '캐나다', '영국', '기타'])
plt.pie(data, labels=data.index, autopct='%1.1f%%') # 그림출력, autopct는 %를 소수 첫째 자리까지
plt.show()
savefile="pie_고객국가빈도.png"
plt.savefig(savefile, dpi=400)
```

```
고객국가빈도
국가
기타      1
대한민국 13
미국       4
영국       3
일본       4
중국       2
캐나다     3
Name: 국가, dtype: int64
```



분석: 고객들이 거주하는 나라 중에서는 대한민국이 가장 많고, 기타를 제외한 중국이 가장 적다.

산점도 1) 상품 생산년도와 가격 산점도

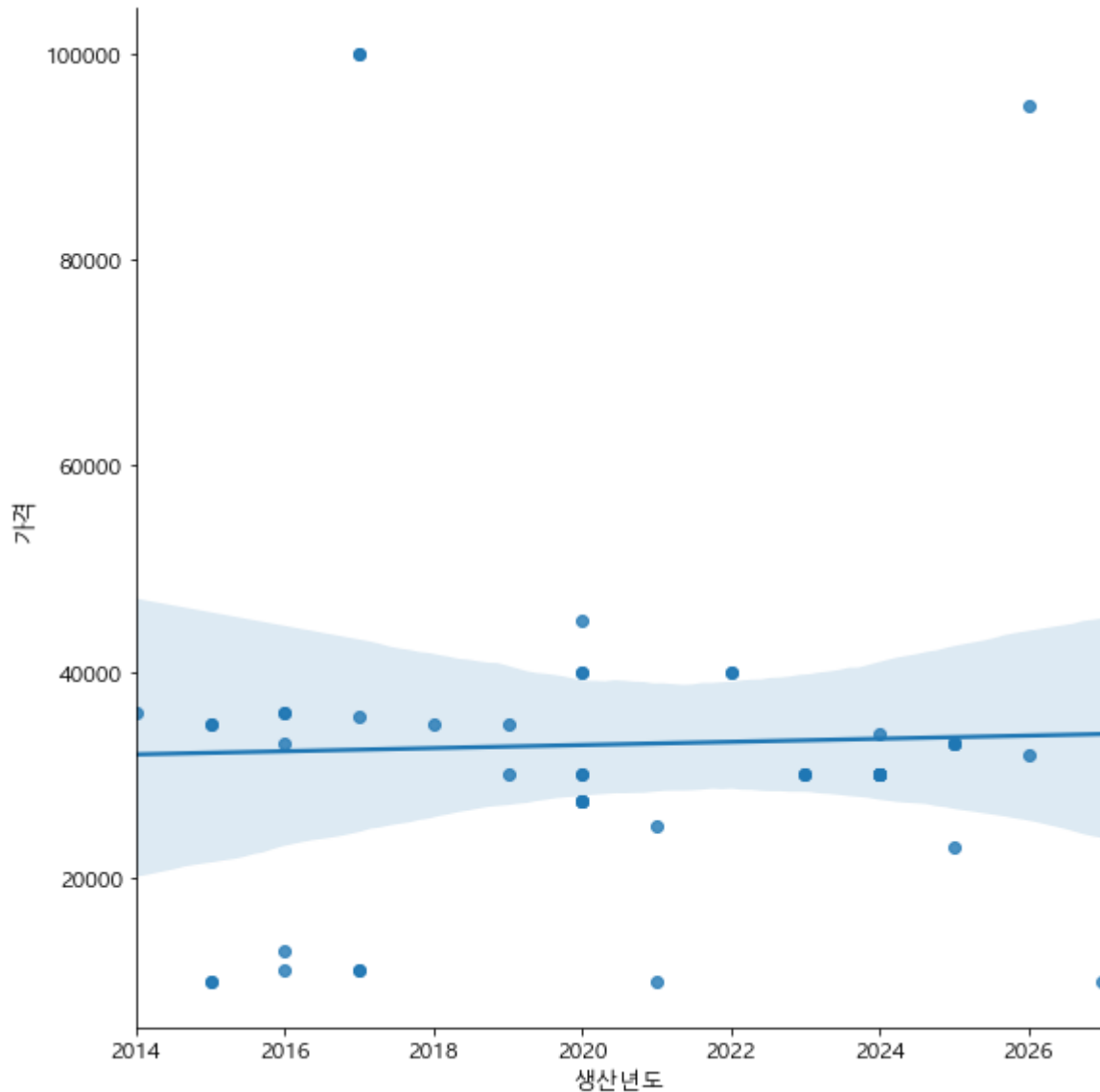
```
In [30]: print('\n# merge() 함수의 left_on과 right_on 사용하기')
orders_detail = orders.merge(product, left_on='제품번호', right_on='제품번호') #pandas에서 조인을 하는 함수.merge(기본, 합칠것,
print(orders_detail)
```

```
# merge() 함수의 left_on과 right_on 사용하기
주문번호  고객번호  제품번호  배송사  주문날짜  결제  제품이름  가격  카테고리  재고
0         1         1       36  한전택배  2022-06-03  card  청바지  기본핏  30000  하의  70.000
```

산점도를 그리기 위한 자료를 얻고자 orders와 product를 merge 한다.

```
In [31]: #산점도
sns.lmplot(x='생산년도', y='가격', height=8, data=orders_detail)

savefile="기본 lmplot.png"
plt.savefig(savefile, dpi=400)
```

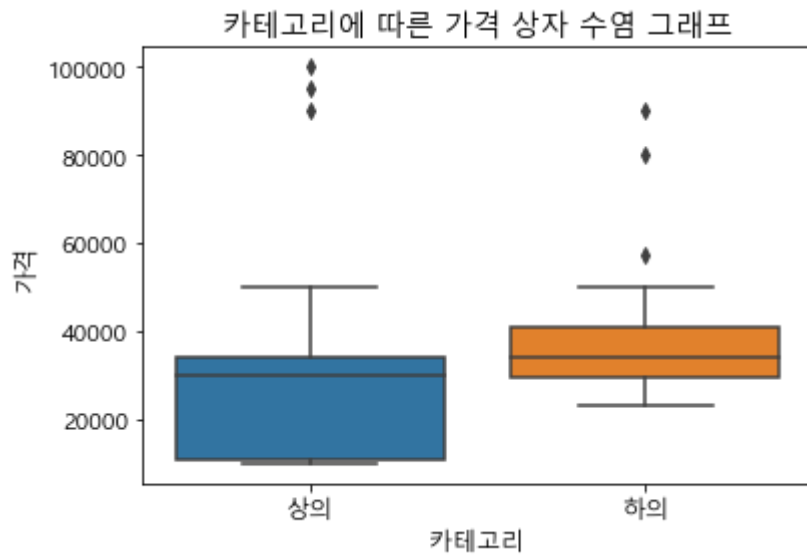


분석: 2020년 부근에 생산된 제품들이 가격이 대체로 낮다.

박스플롯 1) 카테고리에 따른 가격 상자 수염 그래프

```
In [32]: # 박스플롯
ax = plt.subplots()
ax = sns.boxplot(x='카테고리', y='가격', data=product)
ax.set_title('카테고리에 따른 가격 상자 수염 그래프')
ax.set_xlabel('카테고리')
ax.set_ylabel('가격')

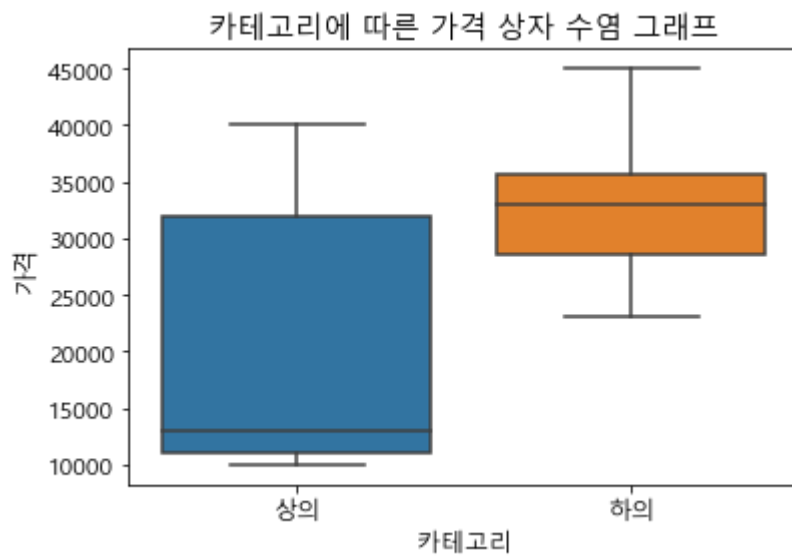
savefile="박스플롯"
plt.savefig(savefile, dpi=400)
```



박스플롯 1-2) 카테고리에 따른 가격 상자 수염 그래프 이상치 제거

```
In [33]: newproduct = product.loc[ product['가격'] <= 49900 ]
ax = plt.subplots()
ax = sns.boxplot(x='카테고리', y='가격', data=newproduct)
ax.set_title('카테고리에 따른 가격 상자 수염 그래프')
ax.set_xlabel('카테고리')
ax.set_ylabel('가격')

savefile="박스플롯_이상치제거"
plt.savefig(savefile, dpi=400)
```

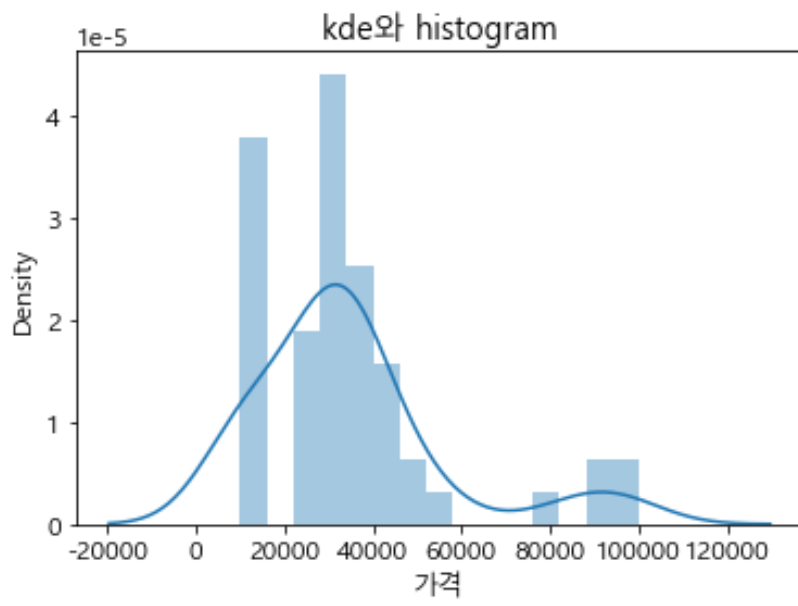


분석: 카테고리가 상위보다 하위인 상품들이 대체로 더 가격이 비싸다.

히스토그램 1) 상품 가격밀도 히스토그램

```
In [35]: plt.figure() # kde(kernel density)
plt.title('kde와 histogram')
sns.distplot(x, rug=False, hist=True, kde=True, label='asdf')

savefile="가격밀도_히스토그램.png"
plt.savefig(savefile, dpi=400)
```



분석: 상품들 중 가격이 30000원대의 상품들이 가장 많다.