3. 데이터베이스 이해 및 ETL 실습 (01. 데이터베이스 이해)

김 효 관

3. 데이터베이스 이해 및 ETL 실습

단원 개요

[단원명]

데이터베이스 이해 및 ETL 실습

[단원 소개]

- 파이썬 기초기술을 익힌 후 데이터 분석을 위한 첫번째 단계인 데이터 수집부분을 다룬다. 일반적인 스프레드 형식이나 웹 상의 데이터 수집 방법 외에 기업에서 데이터 저장소로 많이 사용하는 데이터베이스를 직접 구축하고 자료를 수집하는 방법을 다룬다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
데이터베이스 이해	- 데이터베이스 개념을 이해한다. - 데이터베이스 구성요소 및 장점을 이해한다.
데이터베이스 개발환경 구축	- 오픈소스 데이터베이스 개발환경을 구축한다.
데이터 수집 및 저장	- 파일 데이터 수집/저장 방법을 실습한다. - 데이터베이스 데이터 수집/저장 방법을 실습한다.

교육목표: 데이터베이스의 개념을 정립할 수 있다.

CONTENTS

교육목표: Python 기본 문법 및 데이터를 수집하는 방법을 익힌다.

CONTENTS

1. Pandas 활용한 데이터 수집&저장

모델링 프로세스

Load

소스 데이터

Sellout Data

PostgreSQL, MySQL, Oracle, MongoDB

Processing

Python

Transfer

```
# PANDAS 패키지 불러오기
import pandas as pd
import pymysql
from sqlalchemy import create_engine

# DB 커넥션 열기
mysql_engine = create_engine('mysql+pymysql://kopo:kopo@192.168.110.111:3306/kopo')

# DB 테이블을 읽어 Data Frame 변수에 저장하기
storeData = pd.read_sql_query('SELECT * FROM kopo_product_volume', mysql_engine)
storeData.head()
```

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0002	201512	151750
1	A01	ST0001	201520	645626
2	A01	ST0002	201520	125863
	A01	ST0001	201515	810144
	A01	ST0002	201515	128999

Unload

출력 데이터

Sellout Data

PostgreSQL, MySQL, Oracle, MongoDB

파이썬의 역할

파이썬의 역할

다양한 형태의 데이터를 파이썬에서 분석 후 원하는 곳에 저장 가능함

1. Pandas 활용한 데이터 수집&저장

주요 목차

다양한 데이터 소스를 불러오는 방법을 알면 다양한 각도의 분석이 가능하다

파트1. 파일 데이터 수집 및 저장 (파일)

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장

	CUSTID	AVGPRICE	EMI	DEVICECOUNT	PRODUCTAGE	CUSTTYPE
0	A13566	4273.900000	3	6	1.679181	Big-Screen-lover
1	A14219	3642.441950	2	4	2.682023	Sleeping-dog
2	A15312	3653.884565	2	5	3.208202	Sleeping-dog
3	A16605	3713.211107	2	6	0.900000	Early-bird

Load (수집)

Python

Unload(저장)

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장

기업 고객
구매 데이터

	CUSTID	AVGPRICE	EMI	DEVICECOUNT	PRODUCTAGE	CUSTTYPE
0	A13566	4273.900000	3	6	1.679181	Big-Screen-lover
1	A14219	3642.441950	2	4	2.682023	Sleeping-dog
2	A15312	3653.884565	2	5	3.208202	Sleeping-dog
3	A16605	3713.211107	2	6	0.900000	Early-bird

Column Name	Column Type	Column description	키	길이	NULL 허용	비고
CUSTID	문자	고객 아이디	1		N	
AVGPRICE	숫자	평균 구매 가격				
EMI	숫자	무이자 할부 이용 건수				
DEVICECOUNT	숫자	장비 보유 수				
PRODUCTAGE	숫자	평균 장비 이용기간				
CUSTTYPE	문자	고객 타입				

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장

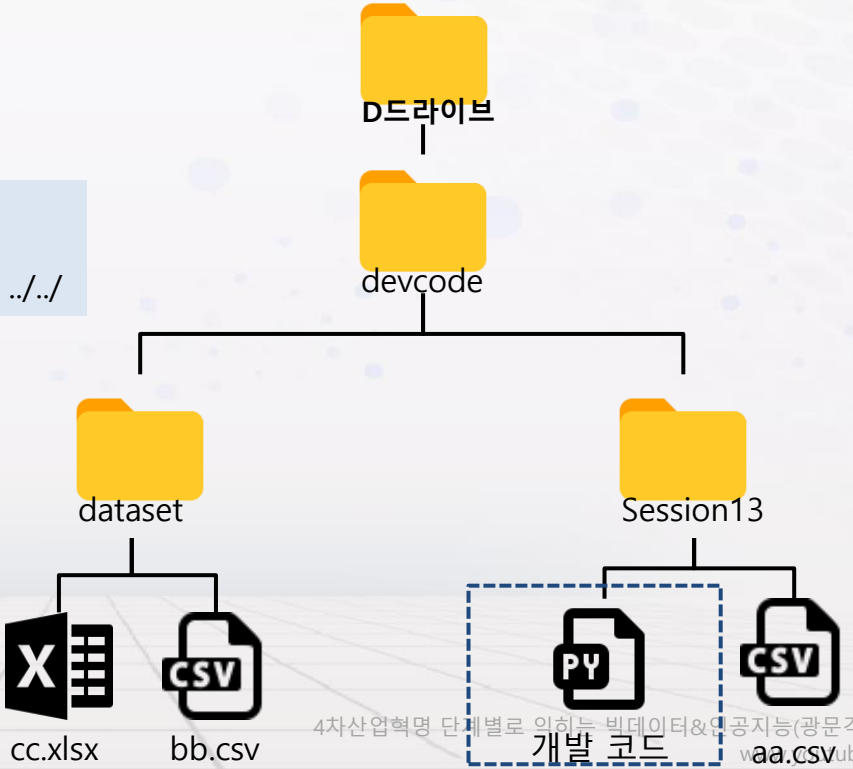
상대경로 접근

./aa.csv
../dataset/bb.csv
../dataset/cc.xlsx

현재 폴더 -> ./
상위 폴더 이동 시 -> ../
상위 폴더의 상위 폴더 이동 시 -> ../../

절대경로 접근

D:/devcode/Session13/aa.csv
D:/devcode/dataset/bb.csv
D:/devcode/dataset/cc.xlsx



1. Pandas 활용한 데이터 수집&저장 (파일)

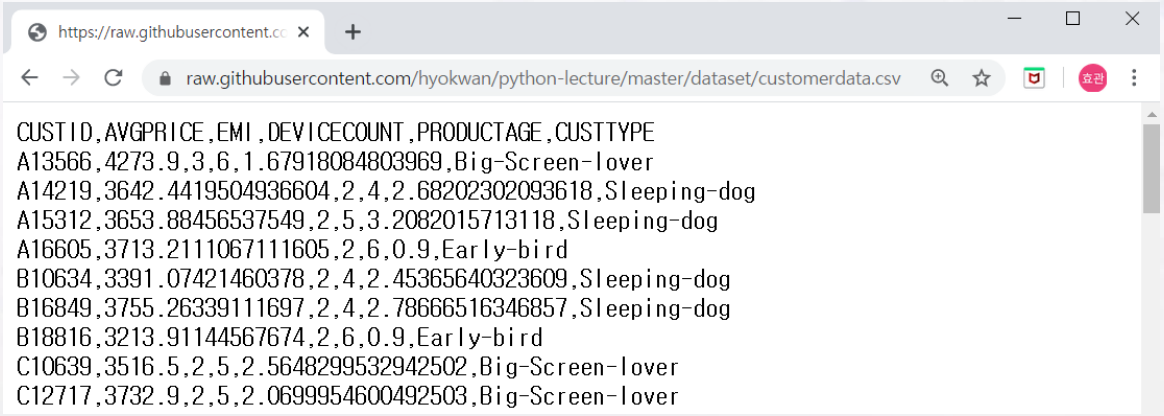
1. 파일 데이터 수집 및 저장

웹 접근

raw.githubusercontent.com/aa.csv

or

파일 속성 내 url 복사



1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장

```
# 활용 라이브러리 (pandas)
import pandas as pd
```

데이터 불러오기

```
pd.read_csv( " 파일경로 ", 추가 옵션)
pd.read_excel(" 파일경로 ", 추가 옵션
```

순서	주요 옵션	내용	비고
1	sep	구분자	콤마: "," 탭: "₩t"
2	encoding	문자 -> bit 변환 방식	한글은 대부분 "ms949"
3	skiprows	무시할 행 개수	

* kopo_region_mst_hangul.csv 한글파일 불러오기

데이터 저장하기

```
pd.to_csv(" 저장경로 ", 추가 옵션)
pd.to_excel(" 저장경로", 추가 옵션)
```

순서	주요옵션	내용	비고
1	index	인덱스 컬럼 포함	일반적으로 False

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장 (csv)

한글깨짐 ms949는 매직 charaterset
kopo_region_mst_hangul을 불러와보자

```
# Pandas 패키지 불러오기
import pandas as pd

# CSV 파일을 읽어 Data Frame 변수에 저장하기
customerData = pd.read_csv("../dataset/customerdata.csv")

# 컬럼명 변경
customerData.columns = ["custid","avgprice","emi",₩
                        "devicecount","productage","cstype"]

# CSV 파일로 저장
customerData.to_csv("../dataset/customerdata_out.csv", index=False)
customerData.head()
```

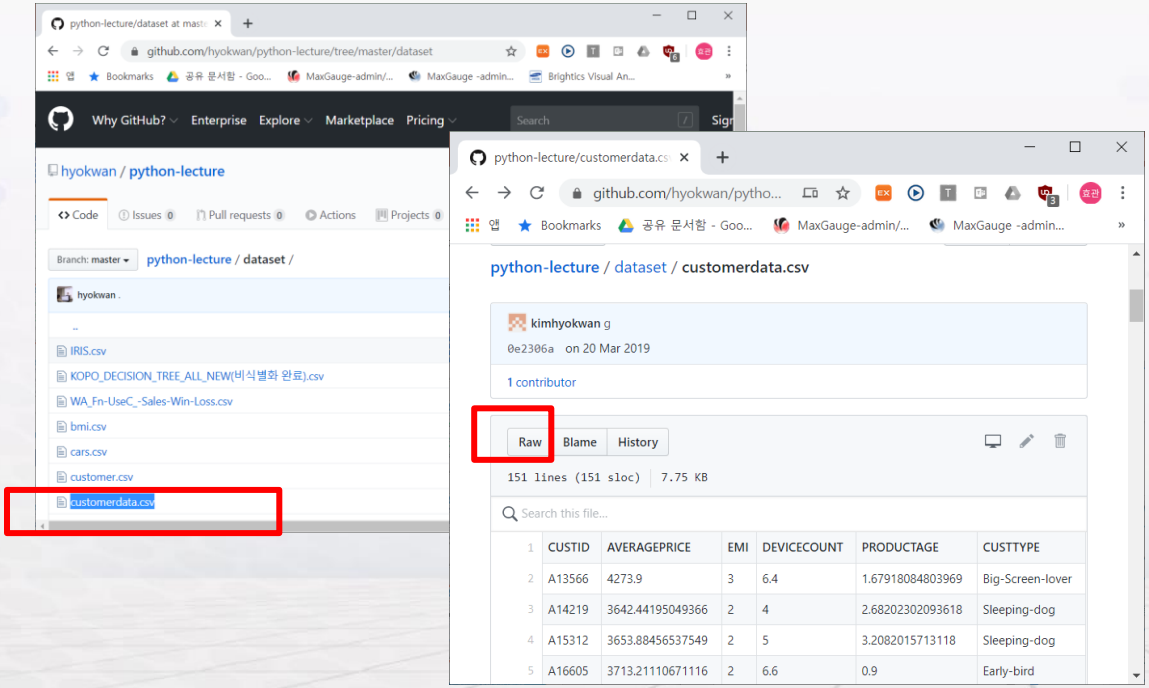
1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장 (한글)

```
# Pandas 패키지 불러오기
import pandas as pd

# CSV 파일을 읽어올 때 encoding = “ms949” 옵션 정의
regionData = pd.read_csv("../dataset/kopo_region_mst_hangul.csv", ₩
                         encoding="ms949")
regionData.head()
```

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장 (웹)



csv url 확인 깃허브 경우 RAW 버튼 클릭 시 URL 필요

1. Pandas 활용한 데이터 수집&저장 (파일)

1. 파일 데이터 수집 및 저장 (웹)

```
# Pandas 패키지 불러오기
import pandas as pd

# CSV 파일을 읽어 Data Frame 변수에 저장하기
customerData = pd.read_csv("https://raw.githubusercontent.com/hyokwan/python-
lecture/master/dataset/customerdata.csv")

# 컬럼명 변경
customerData.columns = ["custid","avgprice","emi",₩
                        "devicecount","productage","cstype"]

customerData.head()
```

실습

../dataset 폴더 내 kopo_product_volume.csv 파일을 selloutdata 변수에 담으세요

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

파트2. Pandas 활용한 데이터 수집 및 저장 (MySQL)

데이터베이스 접속정보

데이터베이스 접속정보

타입	DB 유형	아이피		포트	아이디	비번	디비명/서비스명
학교망	postgres	192.168.110.111		5432	postgres	postgres	postgres
학교망	MySQL	192.168.110.111		3306	kopo	kopo	kopo
학교망	oracle	192.168.110.112		1521	kopo	kopo	orcl
학교망	oracle	192.168.110.111		1521	kopo	kopo	orcl
외부망	postgres	139.150.80.119	16.2	5432	hkcode	hkcodepostgres	hkcodedb
외부망	mariadb	139.150.80.119	15.1	3306	hkcode	hkcodeMySQL	hkcodedb
외부망	MongoDB	hkcluster.ponmqv7.mongodb.net		27017	haiteamkopo		

외부망은 외부 PC에서도 어디서든 접속 가능함!

데이터 정의

데이터 정의

지역/상품 연주차별 판매실적

	regionid	productgroup	yearweek	volume
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

Column Name	Column Type	Column description	키	길이	NULL 허용	비고
regionid	문자	지역정보	1		N	미국, 호주 등
productgroup	문자	상품군 정보	1		N	노트북, 펜 등
yearweek	문자	연주차 정보	1		N	2020년 42주차-> 202042
volume	숫자	판매량 정보				판매량

2. Pandas 활용한 데이터 수집&저장 (MySQL)

1. 파일 데이터 불러오기

```
# Pandas 패키지 불러오기
import pandas as pd
from sqlalchemy import create_engine, inspect

# CSV 파일을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_csv("../dataset/kopo_product_volume.csv")

selloutData.head()
```

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

2. Pandas 활용한 데이터 수집&저장 (MySQL)

2. MySQL에 데이터프레임 저장

pip install pandas sqlalchemy mysql-connector-python

```
# MySQL 데이터베이스 연결 정보 설정
user = '아이디'
password = '패스워드'
host = '127.0.0.1'
port = '3306'
database = 'kopodb'

# SQLAlchemy 엔진 생성
engine = create_engine(f'mysql+mysqlconnector://{user}:{password}@{host}:{port}/{database}')

# 데이터프레임을 MySQL 데이터베이스의 테이블에 저장
table_name = 'kopo_product_volume'
selloutData.to_sql(name=table_name, con=engine, if_exists='replace', index=False)

# 테이블 조회
inspector = inspect(engine)
tables = inspector.get_table_names()
tables
```

! DB연동 패키지 설치필요
# python3
$ pip install mysql-connector-python
$ pip install sqlalchemy

```
Installing collected packages: mysql-connector-python, greenlet, sqlalchemy
Successfully installed greenlet-3.0.3 mysql-connector-python-9.0.0 sqlalchemy-2.0.31

[notice  A new release of pip is available: 24.0 -> 24.1.2
[notice  To update, run: python.exe -m pip install --upgrade pip

C:¥Users¥fintech>
```

2. Pandas 활용한 데이터 수집&저장 (MySQL)

3. 데이터 조회

```
# 테이블 목록 조회
inspector = inspect(engine)

tables = inspector.get_table_names()
tables

# 타겟 테이블 조회
sqlSelect = """
SELECT *
FROM KOPO_PRODUCT_VOLUME"""

selloutDf = pd.read_sql_query(sqlSelect, con=engine)
selloutDf.head()
```

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

실습

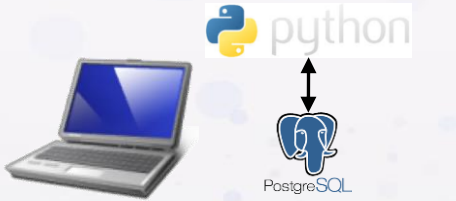
커맨드 창에서 ipconfig 명령 실행 시 ip 확인 가능

local mysql 데이터를 클라우드 mysql에 저장하세요
customerdata -> customerdata_홍길동

퀴즈문제 (스텝업)

타입	DB 유형	아이피	포트	아이디	비번	디비명/서비스명
입력	postgres	192.168.110.111	5432	postgres	postgres	postgres
출력	MySQL	139.150.80.119	3306	hkcode	hkcodeMySQL	hkcodedb

테이블명:csdata_kopo



서버

09:30분까지~!!

테이블명:csdata_홍길동

139.150.80.119클라우드서버

postfres -> mysql

postgresql

```
# 라이브러리 선언
import psycopg2
import pandas as pd
from sqlalchemy import create_engine

dbPrefix = "postgresql"
dbId = "hkcode"
dbPw = "hkcodepostgres"
dbIp = "139.150.80.119"
dbPort = "5432"
dbName = "hkcodedb"
pgEngine = create_engine(f"{dbPrefix}://{dbId}:{dbPw}@{dbIp}:{dbPort}/{dbName}")

# 데이터 불러오기
fileName = "kopo_product_volume"
selloutData = pd.read_csv( f"../dataset/{fileName}.csv" )

# 데이터 저장하기 (자동 세션 종료)
selloutData.to_sql(name=fileName, con= pgEngine, if_exists="replace", index=False)
```

265

MySQL 또는 MySQL

```
# 라이브러리 선언
import pymysql
import pandas as pd
from sqlalchemy import create_engine

dbPrefix = "mysql+pymysql"
dbId = "hkcode"
dbPw = "hkcodemariadb"
dbIp = "139.150.80.119"
dbPort = "3306"
dbName = "hkcodedb"
maEngine = create_engine(f"{dbPrefix}://{dbId}:{dbPw}@{dbIp}:{dbPort}/{dbName}")

# 데이터 불러오기
fileName = "kopo_product_volume"
selloutData = pd.read_csv( f"../dataset/{fileName}.csv" )

# 데이터 저장하기 (자동 세션 종료)
selloutData.to_sql(name=fileName, con= maEngine, if_exists="replace", index=False)
```

265

예외처리는 알아서!! try except 처리 (접속 저장 모두)

파트3. Pandas 활용한 데이터 수집 및 저장 (MongoDB)

데이터베이스 접속정보

데이터베이스 접속정보

타입	DB 유형	아이피		포트	아이디	비번	디비명/서비스명
학교망	postgres	192.168.110.111		5432	postgres	postgres	postgres
학교망	MySQL	192.168.110.111		3306	kopo	kopo	kopo
학교망	oracle	192.168.110.112		1521	kopo	kopo	orcl
학교망	oracle	192.168.110.111		1521	kopo	kopo	orcl
외부망	postgres	139.150.80.119	16.2	5432	hkcode	hkcodepostgres	hkcodedb
외부망	mariadb	139.150.80.119	15.1	3306	hkcode	hkcodeMySQL	hkcodedb
외부망	MongoDB	hkcluster.ponmqv7.mongodb.net		27017	haiteamkopo		

외부망은 외부 PC에서도 어디서든 접속 가능함!

데이터 정의

데이터 정의

지역/상품 연주차별 판매실적

	regionid	productgroup	yearweek	volume
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

Column Name	Column Type	Column description	키	길이	NULL 허용	비고
regionid	문자	지역정보	1		N	미국, 호주 등
productgroup	문자	상품군 정보	1		N	노트북, 펜 등
yearweek	문자	연주차 정보	1		N	2020년 42주차-> 202042
volume	숫자	판매량 정보				판매량

3. Pandas 활용한 데이터 수집&저장 (MongoDB)

1. 파일 데이터 불러오기

```
# Pandas 패키지 불러오기
import pandas as pd
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi

# CSV 파일을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_csv("../dataset/kopo_product_volume.csv")

selloutData.head()
```

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

3. Pandas 활용한 데이터 수집&저장 (MongoDB)

2. MongoDB 연결 확인

```
# MongoDB 연결 정보 설정
user = "hkcode"
password = "1234"
host = '127.0.0.1'
port = 27017
database = 'hdb'

# MongoClient 생성
uri = f"mongodb://{user}:{password}@{host}:{port}/{database}?authSource=admin"
client = MongoClient(uri)

try:
    # 연결 확인 (서버 정보 가져오기)
    server_info = client.server_info()
    print("Connected to MongoDB server:", server_info)
except ConnectionFailure as e:
    print("Could not connect to MongoDB server:", e)
```

```
! DB연동 패키지 설치필요
# python3
$ pip install pymongo

사용자목록 확인 mongosh에서
use admin
db.getUsers()
```

```
Connected to MongoDB server: {'version': '7.0.12', 'gitVersion': 'b6513ce0781db68
008 R2', 'modules': [], 'allocator': 'tcmalloc', 'javascriptEngine': 'mozjs', 'sys
ing': 'Windows SChannel'}, 'buildEnvironment': {'distmod': 'windows', 'distarch':
19.31.31107 for x64', 'ccflags': '/nologo /WX /FImongo/platform/basic.h /fp:strict
d4800 /wd4251 /wd4291 /we4013 /we4099 /we4930 /errorReport:none /MD /O2 /Oy- /bigo
/diagnostics:caret /std:c++20 /Gw /Gy /Zc:inline', 'cxx': 'cl: Microsoft (R) C/C++
'linkflags': '/nologo /DEBUG /INCREMENTAL:NO /LARGEADDRESSAWARE /OPT:REF', 'target
INTRINSICS 0 PCRE2_STATIC NDEBUG BOOST_ALL_NO_LIB _UNICODE UNICODE _SILENCE_CXX17_
RS_DEPRECATION_WARNING _SILENCE_CXX17_CODECVT_HEADER_DEPRECATION_WARNING _SILENCE_
```

3. Pandas 활용한 데이터 수집&저장 (MongoDB)

2. MongoDB 연결 확인

```
# MongoDB 연결 정보 설정
user = "hkcode"
password = "1234"
host = '127.0.0.1'
port = 27017
database = 'hdb'

# MongoClient 생성
uri = f"mongodb://{user}:{password}@{host}:{port}/{database}?authSource=admin"
client = MongoClient(uri)

try:
    # 연결 확인 (서버 정보 가져오기)
    server_info = client.server_info()
    print("Connected to MongoDB server:", server_info)
except ConnectionFailure as e:
    print("Could not connect to MongoDB server:", e)
```

```
! DB연동 패키지 설치필요
# python3
$ pip install pymongo

사용자목록 확인 mongosh에서
use admin
db.getUsers()
```

```
Connected to MongoDB server: {'version': '7.0.12', 'gitVersion': 'b6513ce0781db68
008 R2', 'modules': [], 'allocator': 'tcmalloc', 'javascriptEngine': 'mozjs', 'sys
ing': 'Windows SChannel'}, 'buildEnvironment': {'distmod': 'windows', 'distarch':
19.31.31107 for x64', 'ccflags': '/nologo /WX /FImongo/platform/basic.h /fp:strict
d4800 /wd4251 /wd4291 /we4013 /we4099 /we4930 /errorReport:none /MD /O2 /Oy- /bigo
/diagnostics:caret /std:c++20 /Gw /Gy /Zc:inline', 'cxx': 'cl: Microsoft (R) C/C++
'linkflags': '/nologo /DEBUG /INCREMENTAL:NO /LARGEADDRESSAWARE /OPT:REF', 'target
INTRINSICS 0 PCRE2_STATIC NDEBUG BOOST_ALL_NO_LIB _UNICODE UNICODE _SILENCE_CXX17_
RS_DEPRECATION_WARNING _SILENCE_CXX17_CODECVT_HEADER_DEPRECATION_WARNING _SILENCE_
```

참고 mongodb 사용자 생성

접속을 위한 계정생성

ADMIN 계정 생성

```
# admin database 진입
use admin


# admin user 생성 및 권한 부여
db.createUser({
  user: "hkadmin2",
  pwd: "1234",
  roles: [{ role: "userAdminAnyDatabase", db: "admin" }]
});
```

개발자용 계정 생성

```
# dev user 생성
db.createUser({
  user: "hkuser",
  pwd: "1234",
  roles: []
});

# dev user 권한 부여
db.grantRolesToUser("hkuser", [
  "readWriteAnyDatabase",
  "dbAdminAnyDatabase",
  "userAdminAnyDatabase"
])

# 접속 확인
mongodb://hkuser:1234@{127.0.0.1:27017}
```

3. Pandas 활용한 데이터 수집&저장 (MongoDB)

3. MongoDB 데이터프레임 저장

```
collection_name = "kopo_product_collect"

# 데이터베이스와 컬렉션 객체 생성
db = client[database]
collection = db[collection_name]

# 데이터프레임을 MongoDB에 저장
records = selloutData.to_dict(orient='records')
collection.insert_many(records)
print("Data inserted successfully into MongoDB collection:", collection_name)
```

```
! DB연동 패키지 설치필요
# python3
$ pip install pymongo

사용자목록 확인 mongosh에서
use admin
db.getUsers()
```

```
Connected to MongoDB server: {'version': '7.0.12', 'gitVersion': 'b6513ce0781db681
008 R2', 'modules': [], 'allocator': 'tcmalloc', 'javascriptEngine': 'mozjs', 'sys
ing': 'Windows SChannel'}, 'buildEnvironment': {'distmod': 'windows', 'distarch':
19.31.31107 for x64', 'ccflags': '/nologo /WX /FImongo/platform/basic.h /fp:strict
d4800 /wd4251 /wd4291 /we4013 /we4099 /we4930 /errorReport:none /MD /O2 /Oy- /bigo
/diagnostics:caret /std:c++20 /Gw /Gy /Zc:inline', 'cxx': 'cl: Microsoft (R) C/C++
'linkflags': '/nologo /DEBUG /INCREMENTAL:NO /LARGEADDRESSAWARE /OPT:REF', 'target
INTRINSICS 0 PCRE2_STATIC NDEBUG BOOST_ALL_NO_LIB _UNICODE UNICODE _SILENCE_CXX17_
RS_DEPRECATION_WARNING _SILENCE_CXX17_CODECVT_HEADER_DEPRECATION_WARNING _SILENCE_
```

3. Pandas 활용한 데이터 수집&저장 (MongoDB)

4. 데이터 조회

```
db = client[database]
collection = db[collection_name]

# 컬렉션에서 데이터 조회
cursor = collection.find()

# 조회된 데이터를 데이터프레임으로 변환
selloutDf = pd.DataFrame(list(cursor))
```

selloutDf

	_id	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	66962199e40f814a45521294	A01	ST0001	201415	810144
1	66962199e40f814a45521295	A01	ST0002	201415	128999
2	66962199e40f814a45521296	A01	ST0001	201418	671464
3	66962199e40f814a45521297	A01	ST0002	201418	134467
4	66962199e40f814a45521298	A01	ST0001	201413	470040

핵심정리 및 Q&A

요약

1. MongoDB 데이터 저장하는 방법 습득

파트4. Pandas 활용한 데이터 수집 및 저장 (postgreSQL)

데이터베이스 접속정보

데이터베이스 접속정보

타입	DB 유형	아이피		포트	아이디	비번	디비명/서비스명
학교망	postgres	192.168.110.111		5432	postgres	postgres	postgres
학교망	MySQL	192.168.110.111		3306	kopo	kopo	kopo
학교망	oracle	192.168.110.112		1521	kopo	kopo	orcl
학교망	oracle	192.168.110.111		1521	kopo	kopo	orcl
외부망	postgres	139.150.80.119	16.2	5432	hkcode	hkcodepostgres	hkcodedb
외부망	mariadb	139.150.80.119	15.1	3306	hkcode	hkcodeMySQL	hkcodedb
외부망	MongoDB	hkcluster.ponmqv7.mongodb.net		27017	haiteamkopo		

외부망은 외부 PC에서도 어디서든 접속 가능함!

데이터 정의

데이터 정의

지역/상품 연주차별 판매실적

	regionid	productgroup	yearweek	volume
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

Column Name	Column Type	Column description	키	길이	NULL 허용	비고
regionid	문자	지역정보	1		N	미국, 호주 등
productgroup	문자	상품군 정보	1		N	노트북, 펜 등
yearweek	문자	연주차 정보	1		N	2020년 42주차-> 202042
volume	숫자	판매량 정보				판매량

4. Pandas 활용한 데이터 수집&저장 (PostgreSQL)

1. 파일 데이터 불러오기

```
# Pandas 패키지 불러오기
import pandas as pd
from sqlalchemy import create_engine

# CSV 파일을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_csv("../dataset/kopo_product_volume.csv")

selloutData.head()
```

	REGIONID	PRODUCTGROUP	YEARWEEK	VOLUME
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

4. Pandas 활용한 데이터 수집&저장 (PostgreSQL)

2. 데이터베이스 데이터 수집 및 저장 (PostgreSQL)

! DB연동 패키지 설치필요
$ pip install psycopg2
$ pip install sqlAlchemy

```
import psycopg2
import pandas as pd
from sqlalchemy import create_engine

# csv 데이터 로딩 후 컬럼 소문자로 변환
selloutData = pd.read_csv("../dataset/kopo_product_volume.csv")
selloutData.columns = ["regionid","productgroup","yearweek","volume"]

print(selloutData.head())

# 데이터베이스 접속 엔진 생성
engine = create_engine('postgresql://postgres:postgres@127.0.0.1:5432/postgres')

selloutData.columns = selloutData.columns.str.lower()
# 데이터 저장
resultname='kopo_product_volume'
selloutData.to_sql(name=resultname, con=engine, index = False, if_exists='replace')
# 데이터 불러오기
indata = pd.read_sql_query("select * from kopo_product_volume", engine)
indata.head()
```

fast_executemany=True,

4. Pandas 활용한 데이터 수집&저장 (PostgreSQL)

3. 데이터 조회

```
# 테이블 목록 조회
inspector = inspect(engine)

tables = inspector.get_table_names()
tables

# 타겟 테이블 조회
sqlSelect = """
SELECT *
FROM KOPO_PRODUCT_VOLUME"""

selloutDf = pd.read_sql_query(sqlSelect, con=engine)
selloutDf.head()
```

	regionid	productgroup	yearweek	volume
0	A01	ST0001	201415	810144
1	A01	ST0002	201415	128999
2	A01	ST0001	201418	671464
3	A01	ST0002	201418	134467
4	A01	ST0001	201413	470040

4. Pandas 활용한 데이터 수집&저장 (PostgreSQL)

참고. 성능향상

! DB연동 패키지 설치필요
$ pip install d6tstack

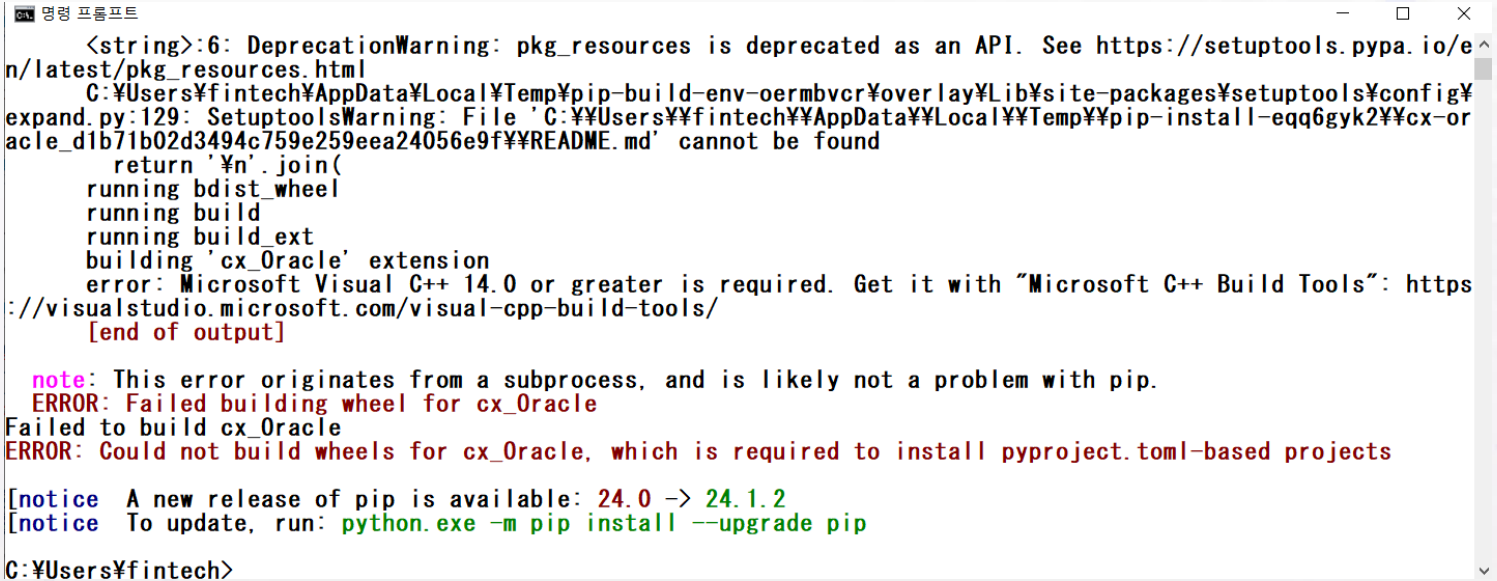
```
import psycopg2
import pandas as pd
from sqlalchemy import create_engine
import d6tstack

# DB 커넥션 열기
purl = 'postgresql+psycopg2://postgres:postgres@127.0.0.1:5432/postgres'
engine = create_engine(purl)

# DB 테이블을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_sql_query('SELECT * FROM kopo_product_volume', engine)

selloutData.head()

# 컬럼해더 재정의
selloutData.columns = ['regionid','pg','yearweek','volume']

# 데이터 저장
resultname='pgresult'
d6tstack.utils.pd_to_psql(df=selloutData, uri=purl, table_name=resultname, if_exists='replace')
```

2018년 6월 출시 Pandas 성능 향상 패키지

실습

postgreSQL에 접속하여
kopo_product_volume 데이터를 불러온 후
regionid 컬럼명을 salesid로 변경 후
dataset 폴더 내 kopo_product_volume_out.csv
파일로 저장하세요

참고. Pandas 활용한 데이터 수집 및 저장 (Oracle)



https://visualsudio.microsoft.com/visual-cpp-build-tools

참고. Pandas 활용한 데이터 수집&저장 (oracle)

2. 데이터베이스 데이터 수집 및 저장 (oracle)

```
import pandas as pd
from sqlalchemy import create_engine

# DB 커넥션 열기
engine = create_engine('oracle+cx_oracle://kopo:kopo@127.0.0.1:1521/xe')

# DB 테이블을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_sql_query('SELECT * FROM kopo_product_volume', engine)

# 컬럼해더 재정의
selloutData.columns = ['REGIONID','PRODUCTGROUP','YEARWEEK','VOLUME']

# 데이터 VIEW
print(selloutData.head())

# 데이터 저장
resultname='oracleresult'
selloutData.to_sql(resultname, engine, if_exists='replace', index=False)
```

! DB연동 패키지 설치필요
$ pip install cx_Oracle

! Oracle client 설치 필요
Toad 설치후 tnsname 설정 후
재시작 필요

```
C:₩Windows₩system32>pip install cx_Oracle==7.1.3
Collecting cx_Oracle==7.1.3
```

```
C:₩Windows₩system32>pip install sqlAlchemy==1.2.15
Collecting sqlAlchemy==1.2.15
```

참고. Pandas 활용한 데이터 수집&저장 (oracle)

2. 데이터베이스 데이터 수집 및 저장 (oracle)

```
import pandas as pd
from sqlalchemy import create_engine
# DB 커넥션 열기
engine = create_engine('oracle+cx_oracle://kopo:kopo@127.0.0.1:1521/xe')
# DB 테이블을 읽어 Data Frame 변수에 저장하기
selloutData = pd.read_sql_query('SELECT * FROM kopo_product_volume', engine
# 컬럼해더 재정의
selloutData.columns = ['REGIONID','PRODUCTGROUP','YEARWEEK','VOLUME']
# Oracle 데이터 저장 속도 향상방법
#### 문자컬럼에 대해서 varchar (100) 적용 _ 사용시 속도 SpeedUp 50배
objectColumns = list(selloutData.columns[selloutData.dtypes == 'object'])
typeDict={}
maxLen = 100

for i in range(0, len(objectColumns)):
    # selloutData[i].str.len().max() 최대치 사용할 경우
    typeDict[ objectColumns[i] ] = types.VARCHAR(100)
#### 문자컬럼에 대해서 varchar (100) 적용 _ 사용시 속도 SpeedUp 50배
selloutData.to_sql(name=tableName.lower(), if_exists="replace", con=engine,dtype=typeDict, index=False)
```

for문 구현 뒤페이지 참고

오라클은 저장시 selloutData.dtypes를 확인하면 object 속성이 추후 clob 형태로 저장된다. 따라서 문자열형태로 변환을 한후 저장시 더 빠른 속도를 낼 수 있다.

참고. Pandas 활용한 데이터 수집&저장 (oracle)

2. 데이터베이스 데이터 수집 및 저장 (oracle)

```
# 필수! 성능향상을 위해 데이터베이스의 타입 정의
for i in range(0, len(objectColumns)):
    # selloutData[i].str.len().max() 최대치 사용할 경우
    typeDict[ objectColumns[i] ] = types.VARCHAR(100)
#### 문자컬럼에 대해서 varchar (100) 적용 _ 사용시 속도 SpeedUp 50배

selloutData.to_sql(name=tableName, if_exists="replace",
con=engine,dtype=typeDict, index=False)
```

실습

옆에 있는 동료의 oracle 서버에 접속 한후
“kopo_product_volume” 자료를 불러와서
컬럼명을 변경 후 자신의 postgreSQL에 저장하세요

실습

옆에 있는 동료의 postgreSQL 서버에 접속 한후
“kopo_product_volume” 자료를 불러와서
컬럼명을 변경 후 자신의 MySQL 에 저장하세요

실습

본인pc의 kopo_channel_seasonality_new.csv 파일을
본인 PC의 ORACLE에 저장한다.
단 dtype을 정의한것과 안한것을 비교한다.

2. 핵심정리 및 Q&A

요약

1. 파이썬에서 파일데이터를 불러오는 방법을 이해해야 한다.
2. 파이썬에서 데이터베이스 저장 데이터를 불러오는 방법을 이해해야 한다.
3. 데이터를 자유롭게 이관 및 저장하는 방법을 이해해야 한다.

2. 핵심정리 및 Q&A

감사합니다.