데이터 수집 및 ETL 실습

김 효 관

Review (Python 기본사용법 및 자료형 익히기)

기억합시다

1. 파이썬 개발환경 (ipython notebook/Jupyter)를 활용하는 방법을 인지한다
2. 자료형: 수, 문자열, 리스트, 튜플, 딕셔너리에 대해서 기억한다
3. 데이터분석을 위한 자료구조인 Dataframe을 생성하는 방법을 인지한다.
4. 데이터베이스 (공유 데이터관리) 를 이해한다.

교육목표: Python 기본 문법 및 데이터를 수집하는 방법을 익힌다.

CONTENTS

1. Pandas 라이브러리를 활용한 데이터 수집&저장

모델링 프로세스

Load → **Processing** → **Unload**

소스 데이터 | **Python** | **출력 데이터**

Sellout Data

Oracle, Postgres, MySQL

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
3	A01	ST0001	201515	810144
4	A01	ST0002	201515	128999

Sellout Data

Oracle, Postgres, MySQL

1. Pandas 라이브러리를 활용한 데이터 수집&저장

주요 목차

1. Pandas 라이브러리를 활용한 데이터 수집&저장

1. 데이터 불러오기 from 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")

# 컬럼해더 재정의
customerData.columns = ['CUSTID','AVGPRICE','EMI','DEVICECOUNT','PRODUCTAGE','CUSTTYPE']

# 데이터 VIEW

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

default =
index = True, header = True
additional info
encoding = 'ms949'
na_rep = '-'

1. Pandas 라이브러리를 활용한 데이터 수집&저장

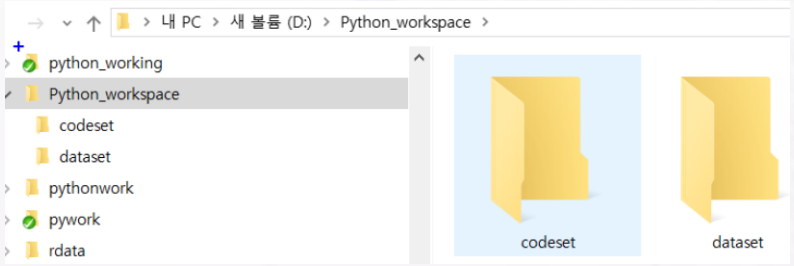
상대경로 / 절대경로의 이해

상대경로 접근

./ → 현재 위치내 파일 접근
../ → 상위 폴더 위치 내 파일 접근

절대경로 접근

D:/Python_workspace/dataset/codeset



D드라이브

Python_workspace

dataset

codeset

dataset 폴더 내
kopo_product_volume.csv
customerdata.csv 有

codeset 폴더 내
파이썬 코드 有

```
import pandas as pd
customerdata = ????
```

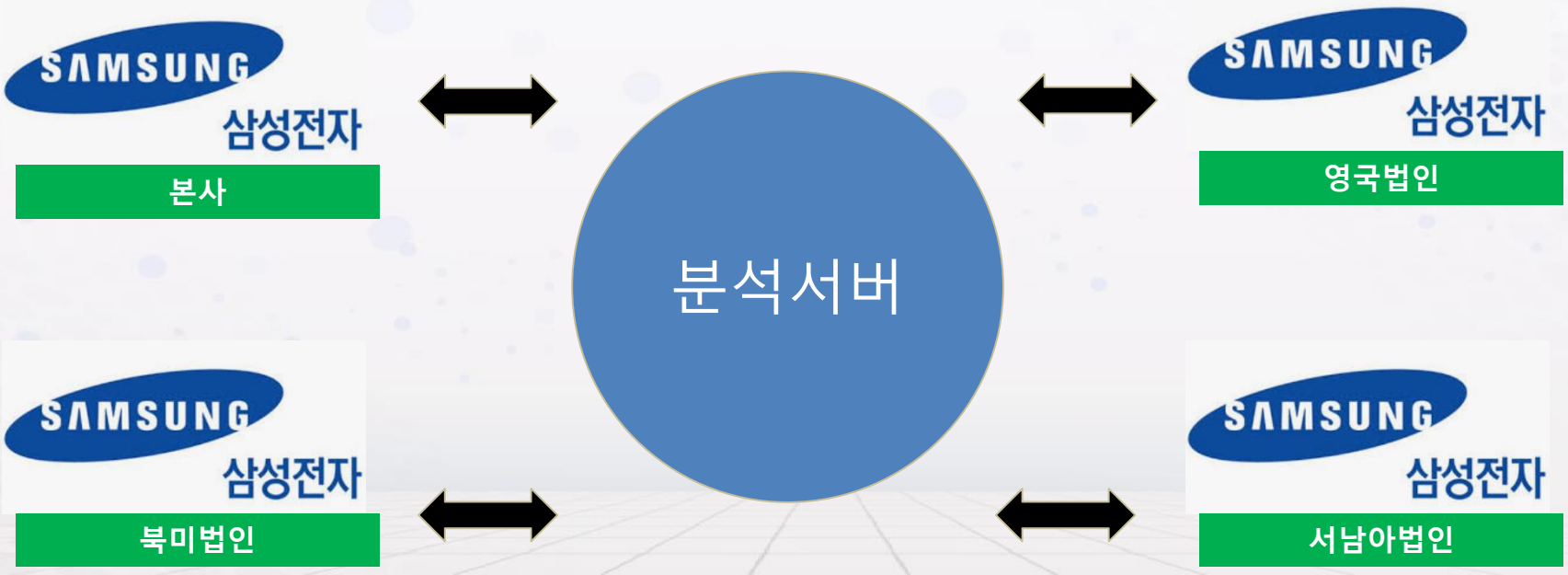
실습

../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

1. Pandas 라이브러리를 활용한 데이터 수집&저장

데이터 이관



1. Pandas 라이브러리를 활용한 데이터 수집&저장

데이터베이스 목록

순번	서버	서버종류	서버주소	포트번호	데이터베이스명	ID	PW
1	로컬 서버	postgresql	127.0.0.1	5432	postgres	postgres	postgres
2	로컬 서버	mariadb	127.0.0.1	3306	mysql	root	mariadb
3	AWS HK서버	postgresql	13.209.112.251	5432	postgres	haiteam	haiteam
4	AWS HK서버	mariadb	13.209.112.251	3306	kopo	kopo	kopo

1. Pandas 라이브러리를 활용한 데이터 수집&저장

2. 데이터 불러오기 from 데이터베이스(Postgres)

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

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

# 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'
selloutData.to_sql(resultname, engine, if_exists='replace')
```

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

```
C:₩Users₩kopo>pip install psycopg2
Collecting psycopg2
```

```
C:₩Users₩kopo>pip show sqlAlchemy
Name: SQLAlchemy
Version: 1.3.5
```

```
C:₩Users₩kopo>pip show psycopg2
Name: psycopg2
Version: 2.8.3
Summary: psycopg2 - Python-PostgreSQL
```

실습

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

1. Pandas 라이브러리를 활용한 데이터 수집&저장

3. 데이터 불러오기 from 데이터베이스(mysql/mariadb)

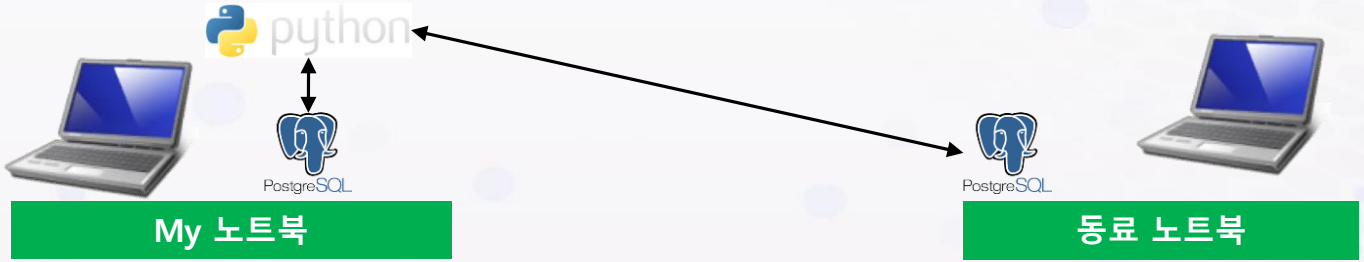
```
# PANDAS 패키지 불러오기
import pandas as pd
import pymysql
from sqlalchemy import create_engine

# DB 커넥션 열기
engine = create_engine('mysql+pymysql://root:mariadb@127.0.0.1:3306/mysql')

# 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']
selloutData.head()

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

```
! DB연동 패키지 설치필요
# python2
$ pip install mysql-python
$ pip install sqlalchemy
# python3
$ pip install pymysql
$ pip install sqlalchemy
```

```
C:₩Users₩kopo>pip install pymysql
Collecting pymysql
```

```
C:₩Users₩kopo>pip show pymysql
Name: PyMySQL
Version: 0.9.3
```

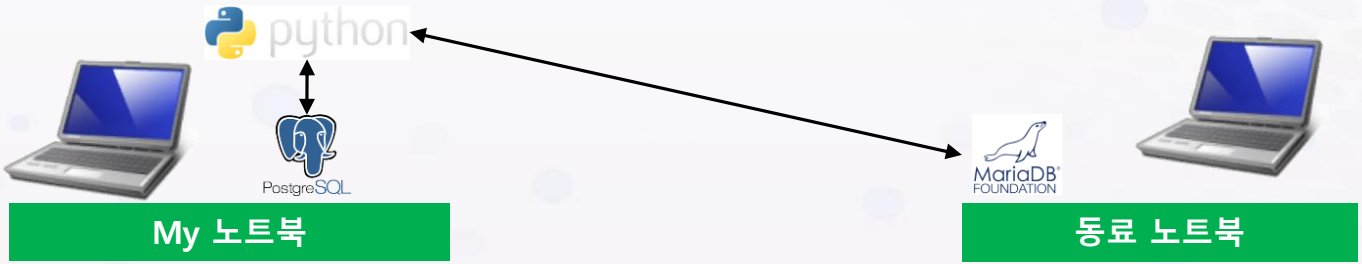
실습



커맨드 창에서 ipconfig 명령 실행 시 ip 확인 가능

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

실습



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

1. Pandas 라이브러리를 활용한 데이터 수집&저장

4. 데이터 불러오기 from 데이터베이스(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
```

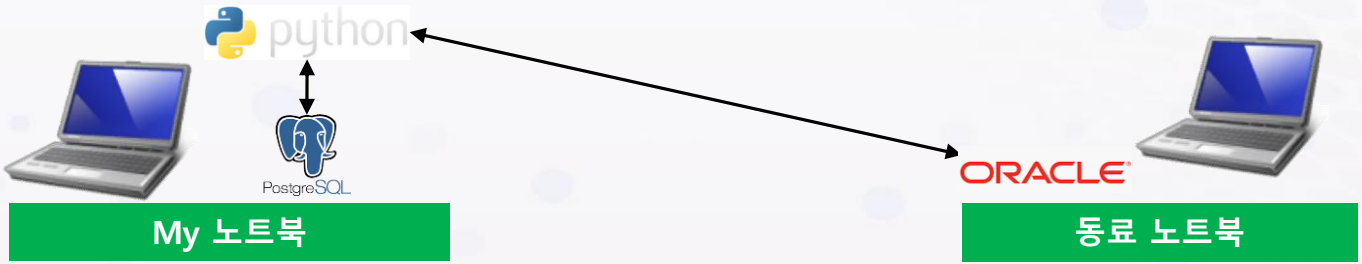
1. Pandas 라이브러리를 활용한 데이터 수집&저장

4. 데이터 불러오기 from 데이터베이스(Oracle) / 데이터 저장 Tip

```
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
from sqlalchemy import types
# 데이터 저장
resultname='oracleresult'

to_varchar = {c:types.VARCHAR(selloutData[c].str.len().max()) ₩
        for c in selloutData.columns[selloutData.dtypes == 'object'].tolist()}

selloutData.to_sql(resultname, engine, if_exists='replace', index=False,
                   dtype=to_varchar)
```

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

실습



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

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.