

[Python]



Python으로 배우는

# 소프트웨어 원리

## Chapter 14. 데이터베이스 연동

# 목차

1. 데이터베이스
2. Python에서 Oracle DB 연동

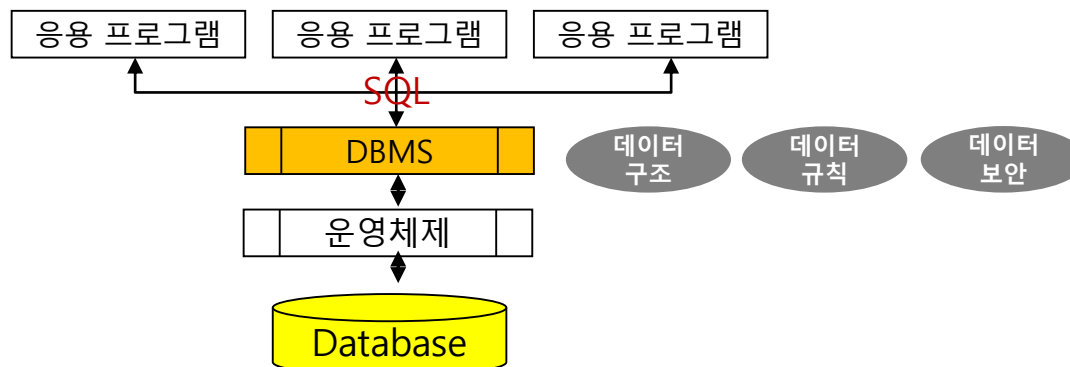
01

데이터베이스

# 01. 데이터베이스

## I. 데이터베이스

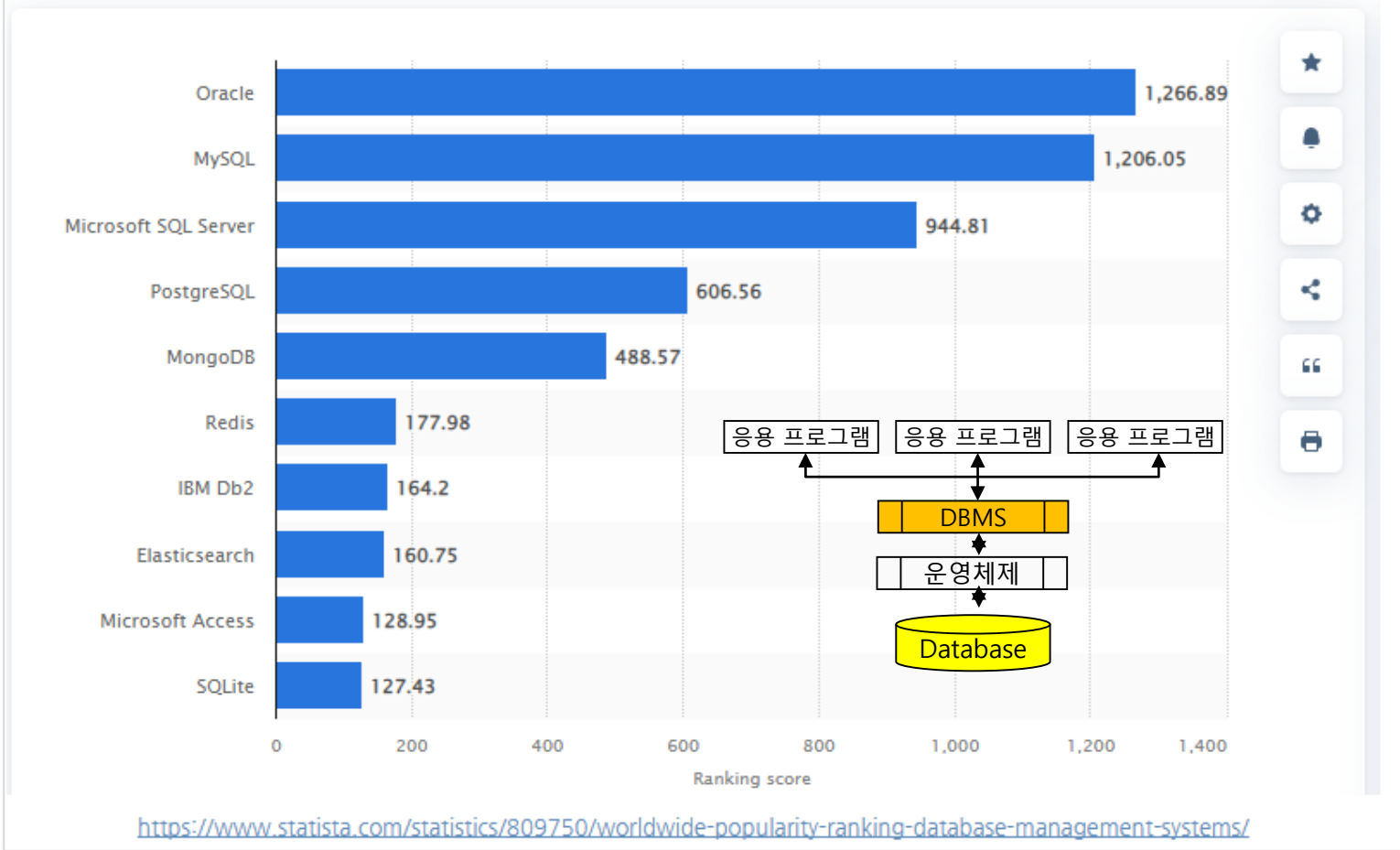
- 데이터베이스(DB: Database)
  - 조직에서 필요한 전체 데이터들을 체계적으로 구조화 하여 저장하고, 필요에 따라 조작(검색, 갱신, 추가, 삭제)할 수 있는 데이터 집합체
- 데이터베이스 관리 시스템(DBMS: Database Management System)
  - [구조] 조직에서 필요한 전체 데이터들을 체계적으로 구조화 하여 저장하고
  - [규칙] 필요에 따라 데이터에 대한 무결성을 유지하여 조작할 수 있도록 하며
  - [보안] 데이터에 대한 보안, 백업/복구 등을 관리하는 시스템
- 질의 언어(SQL: Structured Query Language)
  - 데이터베이스의 데이터를 조작하기 위해 DBMS에 작업을 요청하는 언어



# 01. 데이터베이스

## I. 상용 DBMS 세계 Ranking

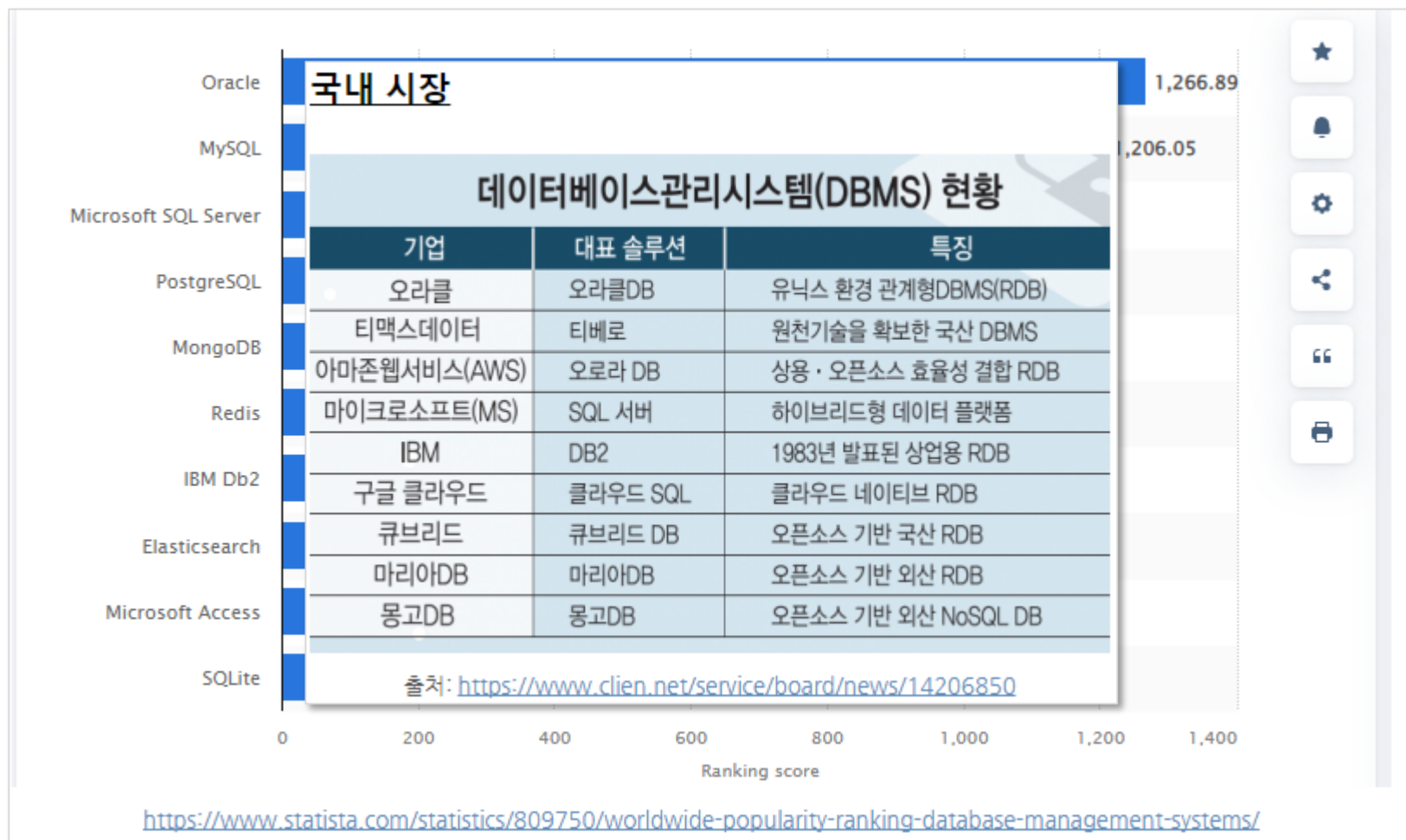
Ranking of the most popular database management systems worldwide, as of January 2022



# 01. 데이터베이스

## I. 상용 DBMS 국내 현황

- Oracle이 국내 시장의 60%이상을 점유하고 있음.



02

# Python에서 Oracle DB 연동

## 02. Python에서 Oracle DB 연동

### I. Oracle DB 연동

- Oracle 관련 모듈 설치

- pip install cx\_Oracle 실행

- Oracle 연결 객체 생성

- # Oracle 서버 연결 정보  
dsn = cx\_Oracle.makedsn('호스트', 포트번호, service\_name='서비스명')  
username = '사용자명'  
password = '비밀번호'

- # Oracle 서버에 연결 객체 생성  
con = cx\_Oracle.connect(username, password, dsn)

- # Oracle 서버 연결 정보  
dsn = cx\_Oracle.makedsn('192.168.142.72', 1521, service\_name='xe')  
username = 'scott'  
password = 'tiger'

- # Oracle 서버에 연결 객체 생성  
con = cx\_Oracle.connect(username, password, dsn)

- # Oracle 서버에 연결 객체 생성 (한 줄로 해결)  
con = cx\_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe", encoding="UTF-8")



## 02. Python에서 Oracle DB 연동

### I. Oracle DB 연동

- Oracle 관련 모듈 설치
  - pip install cx\_Oracle 실행
- Oracle 연결 객체 생성
  - import cx\_Oracle
  - con = cx\_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")
- Query 구성 및 실행
  - sql="Select \* from tab" #Query 구성(존재하는 테이블 검색)
  - cur = con.cursor() #커서 생성: 여러 행에 순차적으로 접근하기 위한 지시자
  - rows = cur.execute(sql) #Query 실행: 결과 행들을 리스트 형태로 받아냄
- 실행 결과 사용
  - for row in rows: #검색 결과 리스트를 탐색
  - print(row)

## 02. Python에서 Oracle DB 연동

### I. Oracle DB 연동

```
import cx_Oracle
#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

#Query 실행
sql="Select * from tab" #존재 테이블 검색
#sql="Select * from dept" #DEPT 테이블 검색
#sql = "select column_name, data_type, data_length from USER_TAB_COLUMNS where
table_name = 'DEPT'" #테이블의 컬럼 구조 검색
cur = con.cursor() #커서 생성: 여러 행에 순차적으로 접근하기 위한 지시자
rows = cur.execute(sql)

#검색 결과 리스트 탐색
for row in rows:
    print(row)

con.close()
```

```
('BONUS', 'TABLE', None)
('DEPT', 'TABLE', None)
('EMP', 'TABLE', None)
('PROJECT', 'TABLE', None)
('SALGRADE', 'TABLE', None)
```

## 02. Python에서 Oracle DB 연동

### I. Oracle DB 연동 : sel\_query()

Ex-OracleDB00.py

```
import cx_Oracle

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    cur = con.cursor() #커서 생성: 여러 행에 순차적으로 접근하기 위한 지시자
    cur.execute(sql)
    rows = [] #검색 결과 행 반환용 리스트
    seq = 0
    for row in cur:
        rows.append(row)
    return rows

### Main ###
sql="Select * from tab" #테이블 검색
rows = sel_query(con, sql) #Query 실행 요청
for row in rows: #검색 결과 리스트 탐색
    print(row)

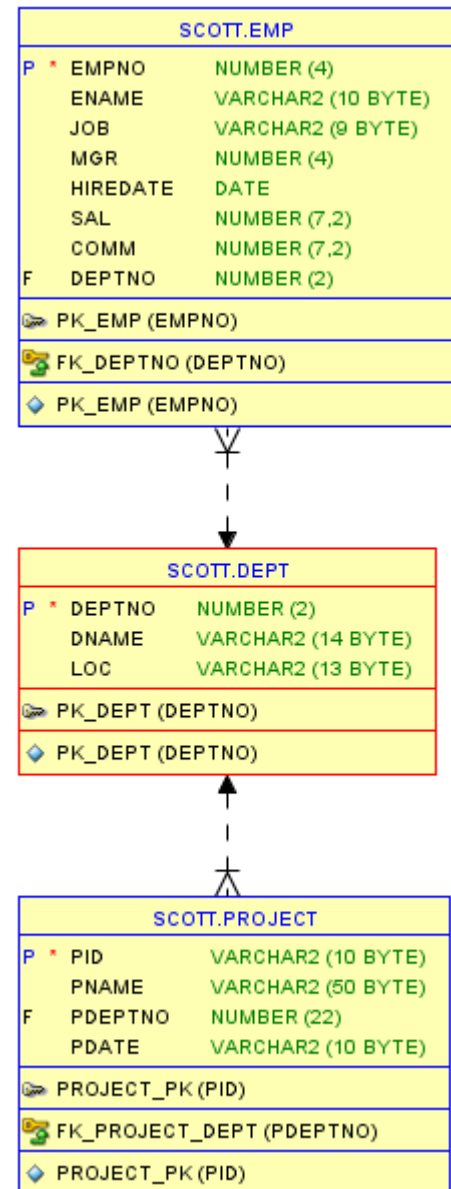
con.close()
```

```
('BONUS', 'TABLE', None)
('DEPT', 'TABLE', None)
('EMP', 'TABLE', None)
('PROJECT', 'TABLE', None)
('SALGRADE', 'TABLE', None)
```

## 02. Python에서 Oracle DB 연동

### II. scott/tiger 데이터베이스 사용

- Oracle에서 제공하는 실습용 DB
- scott/tiger 데이터베이스
  - host : 192.168.142.72
  - port no. : 1521
  - service name(SID) : xe
  - user name : scott
  - user password : tiger



## 02. Python에서 Oracle DB 연동

### II. scott/tiger 데이터베이스 사용

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	80/12/17	800	(null)	20
2	7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
3	7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
4	7566	JONES	MANAGER	7839	81/04/02	2975	(null)	20
5	7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
6	7698	BLAKE	MANAGER	7839	81/05/01	2850	(null)	30
7	7782	CLARK	MANAGER	7839	81/06/09	2450	(null)	10
8	7839	KING	PRESIDENT	(null)	81/11/17	5000	(null)	10
9	7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
10	7900	JAMES	CLERK	7698	81/12/03	950	(null)	30
11	7902	FORD	ANALYST	7566	81/12/03	3000	(null)	20
12	7934	MILLER	CLERK	7782	82/01/23	1300	(null)	10

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

	PID	PNAME	PDEPTNO	PDATE
1	P20230001	Pythone Programmin Buield up team	10	2023/05/21
2	P20230002	Oracle DB team	30	2023/06/10

SCOTT.EMP	
P *	EMPNO NUMBER (4)
	ENAME VARCHAR2 (10 BYTE)
	JOB VARCHAR2 (9 BYTE)
	MGR NUMBER (4)
	HIREDATE DATE
	SAL NUMBER (7,2)
	COMM NUMBER (7,2)
F	DEPTNO NUMBER (2)
PK_EMP (EMPNO)	
FK_DEPTNO (DEPTNO)	
PK_EMP (EMPNO)	

SCOTT.DEPT	
P *	DEPTNO NUMBER (2)
	DNAME VARCHAR2 (14 BYTE)
	LOC VARCHAR2 (13 BYTE)
PK_DEPT (DEPTNO)	
PK_DEPT (DEPTNO)	

SCOTT.PROJECT	
P *	PID VARCHAR2 (10 BYTE)
	PNAME VARCHAR2 (50 BYTE)
F	PDEPTNO NUMBER (22)
	PDATE VARCHAR2 (10 BYTE)
PROJECT_PK (PID)	
FK_PROJECT_DEPT (PDEPTNO)	
PROJECT_PK (PID)	

## 02. Python에서 Oracle DB 연동

### II. Oracle DB 연동 : DEPT 테이블 검색

Ex-OracleDB00.py

```
import cx_Oracle

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    cur = con.cursor() #커서 생성: 여러 행에 순차적으로 접근하기 위한 지시자
    cur.execute(sql)
    rows = [] #검색 결과 행 반환용 리스트
    seq = 0
    for row in cur:
        rows.append(row)
    return rows

### Main ###
sql="Select * from DEPT" #테이블 검색
rows = sel_query(con, sql) #Query 실행 요청
for row in rows: #검색 결과 리스트 탐색
    print(row)

con.close()
```

```
(10, 'ACCOUNTING', 'NEW YORK')
(20, 'RESEARCH', 'DALLAS')
(30, 'SALES', 'CHICAGO')
(40, 'OPERATIONS', 'BOSTON')
```

## 02. Python에서 Oracle DB 연동

### II. Oracle DB 연동 : DEPT 테이블 구조 검색

Ex-OracleDB00.py

```
import cx_Oracle

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    cur = con.cursor() #커서 생성: 여러 행에 순차적으로 접근하기 위한 지시자
    cur.execute(sql)
    rows = [] #검색 결과 행 반환용 리스트
    seq = 0
    for row in cur:
        rows.append(row)
    return rows

### Main ###
sql = "select column_name, data_type, data_length from USER_TAB_COLUMNS where table_name = 'DEPT'"
rows = sel_query(con, sql) #Query 실행 요청
for row in rows: #검색 결과 리스트 탐색
    print(row)

con.close()
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

```
('DEPTNO', 'NUMBER', 22)
('DNAME', 'VARCHAR2', 14)
('LOC', 'VARCHAR2', 13)
```

## 02. Python에서 Oracle DB 연동

### II. [실습] DEPT 테이블 검색 함수 prt\_dept()

Ex-OracleDB01.py

```
import cx_Oracle

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

def prt_dept():
    sql="Select * from dept" #DEPT 테이블 검색
    rows = sel_query(con, sql)
    for row in rows:
        deptno, dname, loc = row #검색 행 unpacking
        print("%5s %25s %15s" %(deptno, dname, loc))

### Main ###
prt_dept() #DEPT 테이블 검색, 출력
con.close()
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



## 02. Python에서 Oracle DB 연동

### II. [실습] EMP 테이블 검색 함수 prt\_emp()

Ex-OracleDB02.py

```
import cx_Oracle

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

def prt_emp():
    sql="select empno, ename, job, mgr, to_char(hiredate, 'YYYY/MM/DD'), sal,
    NVL(comm, 0), deptno from EMP" #DEPT 테이블 검색
    rows = sel_query(con, sql)
    for row in rows:
        empno, ename, job, mgr, hiredate, sal, comm, deptno = row
        print("%10s %15s %10s %10s %10s" %(empno, ename, job, mgr, hiredate),
        end="")
        print("%10.1f %10d %5s" %(sal, comm, deptno))

### Main ###
prt_emp() #EMP 테이블 검색, 출력

con.close()
```

7369	SMITH	CLERK	7902	1980/12/17	800.0	0	20
7499	ALLEN	SALESMAN	7698	1981/02/20	1600.0	300	30
7521	WARD	SALESMAN	7698	1981/02/22	1250.0	500	30
7566	JONES	MANAGER	7839	1981/04/02	2975.0	0	20
7654	MARTIN	SALESMAN	7698	1981/09/28	1250.0	1400	30
7698	BLAKE	MANAGER	7839	1981/05/01	2850.0	0	30
7782	CLARK	MANAGER	7839	1981/06/09	2450.0	0	10
7839	KING	PRESIDENT	None	1981/11/17	5000.0	0	10
7844	TURNER	SALESMAN	7698	1981/09/08	1500.0	0	30
7900	JAMES	CLERK	7698	1981/12/03	950.0	0	30
7902	FORD	ANALYST	7566	1981/12/03	3000.0	0	20
7934	MILLER	CLERK	7782	1982/01/23	1300.0	0	10

## 02. Python에서 Oracle DB 연동

### II. [실습] EMP, DEPT 테이블 Join 검색 함수 prt\_empdept()

```
import cx_OracleEx-OracleDB03.py

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

def prt_empdept():
    sql="select empno, ename, job, dname from EMP e, DEPT d where e.deptno = d.deptno" #DEPT 테이블 검색
    rows = sel_query(con, sql)
    for row in rows:
        empno, ename, job, dname = row
        print("%10s %15s %10s %15s" %(empno, ename, job, dname))

### Main ###
prt_empdept() #EMP, DEPT 테이블 Join 검색, 출력

con.close()
```

7369	SMITH	CLERK	RESEARCH
7499	ALLEN	SALESMAN	SALES
7521	WARD	SALESMAN	SALES
7566	JONES	MANAGER	RESEARCH
7654	MARTIN	SALESMAN	SALES
7698	BLAKE	MANAGER	SALES
7782	CLARK	MANAGER	ACCOUNTING
7839	KING	PRESIDENT	ACCOUNTING
7844	TURNER	SALESMAN	SALES
7900	JAMES	CLERK	SALES
7902	FORD	ANALYST	RESEARCH
7934	MILLER	CLERK	ACCOUNTING

## 02. Python에서 Oracle DB 연동

### II. [실습] EMP, EMP 테이블 Join 검색 함수 prt\_empemp()

```
import cx_OracleEx-OracleDB04.py

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

def prt_empemp():
    sql="select e1.empno, e1.ename, e1.job, e2.ename, e2.job from EMP e1, EMP e2
    where e1.mgr = e2.empno"
    rows = sel_query(con, sql)
    for row in rows:
        empno, ename, ejob, mname, mjob = row
        print("%10s %15s %10s %15s %15s" %(empno, ename, ejob, mname, mjob))

### Main ###
prt_empemp() #EMP, EMP 테이블 Join
con.close()
```

7902	FORD	ANALYST	JONES	MANAGER
7900	JAMES	CLERK	BLAKE	MANAGER
7844	TURNER	SALESMAN	BLAKE	MANAGER
7654	MARTIN	SALESMAN	BLAKE	MANAGER
7521	WARD	SALESMAN	BLAKE	MANAGER
7499	ALLEN	SALESMAN	BLAKE	MANAGER
7934	MILLER	CLERK	CLARK	MANAGER
7782	CLARK	MANAGER	KING	PRESIDENT
7698	BLAKE	MANAGER	KING	PRESIDENT
7566	JONES	MANAGER	KING	PRESIDENT
7369	SMITH	CLERK	FORD	ANALYST

## 02. Python에서 Oracle DB 연동

### II. [실습] PROJECT 테이블에 행 삽입 함수 insert\_project()

## 중간 생략 ##

Ex-OracleDB05.py

```
def in_query(con, sql=None, indata=None):
```

```
    cur = con.cursor()
```

```
    cur.execute(sql, indata)
```

```
    con.commit()
```

```
def insert_project() :
```

```
    sql = "insert into PROJECT(pid, pname, pdeptno, pdate) " \
          " values (:0, :1, :2, :3)"
```

```
    indata = []
```

```
    print(">> Project 컬럼 값 입력<<")
```

```
    indata.append(input("Project ID >"))
```

```
    indata.append(input("Project Name >"))
```

```
    indata.append(int(input("Management Deptno >")))
```

```
    indata.append(input("Start Date(yyyy/mm/dd) >"))
```

```
    print(indata)
```

```
    in_query(con, sql, indata)
```

```
### Main ###
```

```
insert_project() #PROJECT 테이블에 행 추가
```

```
con.close()
```

```
>> Project 컬럼 값 입력<<
Project ID > P20230601
Project Name > Test project
Management Deptno > 10
Start Date(yyyy/mm/dd) > 2023/06/07
['P20230601', 'Test project', 10, '2023/06/07']
```

## 02. Python에서 Oracle DB 연동

### II. [실습] PROJECT 테이블에 행 삽입 함수 insert\_project()

## 중간 생략 ##

Ex-OracleDB05.py

```
def in_query(con, sql=None, indata=None):
```

```
    cur = con.cursor()
    cur.execute(sql, indata)
    con.commit()
```

```
def insert_project() :
```

```
    sql = "insert into PROJECT(pid, pname, pdeptno, pdate) " \
          " values (:0, :1, :2, :3)"
```

```
    indata = []
```

```
    print(">> Project 컬럼 값 입력<<")
```

```
    indata.append(input("Project ID >"))
```

```
    indata.append(input("Project Name >"))
```

```
    indata.append(int(input("Management Deptno >")))
```

```
    indata.append(input("Start Date(yyyy/mm/dd) >"))
```

```
    print(indata)
```

```
    in_query(con, sql, indata)
```

```
### Main ###
```

```
insert_project() #PROJECT 테이블에 행 추가
```

```
con.close()
```

```
>> Project 컬럼 값 입력<<
Project ID > P20230601
Project Name > Test project
Management Deptno > 10
Start Date(yyyy/mm/dd) > 2023/06/07
['P20230601', 'Test project', 10, '2023/06/07']
```

## 02. Python에서 Oracle DB 연동

### II. [실습] PROJECT 테이블 검색 함수 prt\_project()

```
import cx_OracleEx-OracleDB05.py

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

## 중간 생략 ##

def prt_project():
    pass ## 작성할 부분

### Main ###
prt_project() #PROJECT 테이블 검색, 출력

con.close()
```

P20230001  
P20230002

Pythone Programmin Buield up team  
Oracle DB team

10 2023/05/21  
20 2023/06/10

## 02. Python에서 Oracle DB 연동

### II. [실습] PROJECT 테이블 행 변경 함수 update\_project()

```
import cx_OracleEx-OracleDB06.py

#Oracle 연결
con = cx_Oracle.connect("scott", "tiger", "192.168.142.72:1521/xe")

def sel_query(con, sql=None):
    ## 생략 ##

    ## 중간 생략 ##

### Main ###
update_project() #PROJECT 테이블 행 변경

con.close()
```

```
>> 수정할 Project ID 입력(없으면 Enter) > P20230002
>>Queried by row count: 1
>> 수정할 컬럼 값 입력(없으면 Enter)<<
Oracle DB team >
20 > 30
2023/06/10 >
update PROJECT set pname='Oracle DB team', pdeptno=30, pdate='2023/06/10' where pid = 'P20230002'
```

## 02. Python에서 Oracle DB 연동

### II. [실습] PROJECT 테이블 행 변경 함수 update\_project()

```
def update_project():
```

```
    pid = input(">> 수정할 Project ID 입력(없으면 Enter) >")
```

```
    if len(pid) == 0:
```

```
        return
```

```
    else:
```

```
        sql=f"Select * from project where pid = '{pid}'"
```

```
        rows = sel_query(con, sql)
```

```
        for row in rows:
```

```
            pid, pname, pdeptno, pdate = row
```

```
        print(">> 수정할 컬럼 값 입력(없으면 Enter)<<")
```

```
        indata = input(f"{pname} >")
```

```
        if len(indata) != 0:
```

```
            pname = indata
```

```
        indata = input(f"{pdeptno} >")
```

```
        if len(indata) != 0:
```

```
            pdeptno = int(indata)
```

```
        indata = input(f"{pdate} >")
```

```
        if len(indata) != 0:
```

```
            pdeptno = indata
```

```
        sql = f"update PROJECT set pname='{pname}', pdeptno={pdeptno},
```

```
pdate='{pdate}' "
```

```
        f" where pid = '{pid}'"
```

```
        print(sql)
```

```
        up_query(con, sql, indata)
```

Ex-OracleDB06.py



# Thank You !

[Python]