

R과 python으로 Oracle DB에 접근

1. R

R에서 DB에 연결

<https://db.rstudio.com/getting-started/connect-to-database>

Ch9 Oracle DB

R 3.6.3

사전작업

#

1. C:드라이브에 OracleTest 폴더 생성

2. OracleTest폴더에 driver ojdbc6 저장

ojdbc6 위치

C:\oracle\exe\app\oracle\product\11.2.0\server\jdbc\lib

3. SQL developer에서 scott계정으로 table 생성

#

create table test_table(

id varchar(50) primary key,

pass varchar(30) not null,

name varchar(25) not null,

age number(2)

);

#

insert into test_table values('hong', '1234', '홍길동', 35);

insert into test_table values('kim', '5678', '김길동', 45);

select * from test_table;

commit;

```
# 아래 package 설치 및 실행
```

```
# package설치
```

```
install.packages("rJava")
```

```
install.packages("DBI")
```

```
install.packages("RJDBC")
```

```
# package로딩
```

```
# Sys.setenv(JAVA_HOME = "C:\\Program Files\\Java\\jdk1.8.0_311")
```

```
# install.packages("rJava")
```

```
library(DBI)
```

```
library(rJava)
```

```
library(RJDBC)
```

```
# remove.packages("rJava", lib=~ /R/win-library/4.0")
```

```
# Drive 설정
```

```
drv <- JDBC("oracle.jdbc.driver.OracleDriver", "C:/OracleTest/ojdbc6.jar")
```

```
# 데이터베이스 연결
```

```
conn <- dbConnect(drv,"jdbc:oracle:thin:@//127.0.0.1:1521/xe", "scott", "tiger")
```

```
# DB 레코드 검색, 추가, 수정, 삭제하기
```

```
# 모든 record
```

```
query = "SELECT * FROM test_table"
```

```
# query = "SELECT * FROM EMP"
```

```
dbGetQuery(conn, query)
```

```
# 나이 기준으로 내림차순 정렬
```

```
query = "SELECT * FROM test_table order by age desc"
```

```
dbGetQuery(conn, query)
```

```
# insert record
```

```
query = "insert into test_table values('kang', '1234', '강감찬', 45)"
```

```
dbSendUpdate(conn, query)
```

```
# 추가 확인
```

```
query = "SELECT * FROM test_table"
```

```
dbGetQuery(conn, query)
```

```
# 나이가 40세 이상인 record
```

```
query = "select * from test_table where age >= 40"
```

```
result <- dbGetQuery(conn, query)
```

```
result
```

```
# name이 '강감찬'인 데이터의 age를 40으로 수정
```

```
query = "update test_table set age = 40 where name = '강감찬'"
```

```
dbSendUpdate(conn, query)
```

```
# 수정된 레코드 조회
```

```
query = "select * from test_table where name = '강감찬'"
```

```
dbGetQuery(conn, query)
```

```
# name이 '홍길동'인 레코드 삭제
```

```
query = "delete from test_table where name = '홍길동'"
```

```
dbSendUpdate(conn, query)
```

```
# 전체 레코드 조회
query = "select * from test_table"
dbGetQuery(conn, query)
```

2. python

python으로 DB에 연결

<https://www.edureka.co/blog/python-database-connection/> (MySQL)

<https://www.geeksforgeeks.org/oracle-database-connection-in-python/> (Oracle)

1) Oracle XE DB

Oracle DB에서 table 생성 (SQL Developer)

CREATE TABLE PY_TABLE(

id varchar(50) primary key,

pass varchar(30) not null,

name varchar(25) not null,

age number(2)

#);

insert into py_table values('hong', '1234', '홍길동', 35);

insert into py_table values('kim', '5678', '김길동', 45);

*# select * from py_table;*

commit;

2) Oracle Instant Client Downloads

<https://www.oracle.com/kr/database/technologies/instant-client/downloads.html>

Instant Client for Microsoft Windows (x64) 선택

다운로드 후 C: 드라이브에 저장

3) Oracle DB 활용

```
import cx_Oracle
import os

# 한글 지원 환경
os.putenv('NLS_LANG', '.UTF8')

# Path 설정
LOCATION = r"C:/instantclient_21_3"
os.environ["PATH"] = LOCATION + ";" + os.environ["PATH"]

# DB 연결
connect = cx_Oracle.connect("scott", "tiger", "localhost:1521/xe")
cs = connect.cursor()

# 데이터 추가(insert)
sql = "insert into py_table values('kang', '1234', '강감찬', 45)"
cs.execute(sql) # SQL 문

cs.execute("select * from py_table") # SQL 문

for i in cs: # data 보기
    print(i)

# 나이가 40 세 이상인 record
sql = "select * from test_table where age >= 40"
cs.execute(sql)

# name 이 '강감찬'인 데이터의 age 를 40 으로 수정
sql = "update test_table set age = 40 where name = '강감찬'"
cs.execute(sql)
```

```
# 수정된 레코드 조회
sql = "select * from test_table where name = '강감찬'"
cs.execute(sql)
```

```
# name 이 '홍길동'인 레코드 삭제
sql = "delete from test_table where name = '홍길동'"
cs.execute(sql)
```

```
# 전체 레코드 조회
sql = "select * from test_table"
cs.execute(sql)
```

```
for i in cs: # data 보기
    print(i)
```

```
# 사용 후 close
cs.close()
connect.commit()
connect.close()
```