

노트북: blog
만든 날짜: 2020-10-04 오후 7:20
URL: <https://continuous-development.tistory.com/50?category=793392>

2020. 8. 3. 17:37 수정 삭제 공개

ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_121
NUMBER_OF_PRO...	8
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;C:...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.W...
PROCESSOR_ARC...	AMD64

RJDBC - R에서 자바를 통해 데이터베이스에 접속할 수 있도록 하는 자바 API이다

```

111
112
113 # R과 DBMS 연동을 통한 정형 데이터 처리방법
114
115 install.packages("rJava")
116
117 #인터페이스
118 install.packages("DBI")
119
120 #jdbcs 함수 제공
121 install.packages("RJDBC")
122
123 # 이것 하기전에 jdk가 미리 깔려있어야 한다.
124
125
126 Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jdk1.8.0_121')
127 library(rJava)
128 library(DBI)
129 library(RJDBC)
130
131 |

```

Sys.setenv 를 통해 자바 환경 변수를 설정해 준다.

```

134 #DB 연동을 위한 순서는
135 #1.Driver loading, 2.Connection(hr,hr) , 3.Query 수행 , 4.결과 집합 확인하는 과정
136
137 #DB 가 바뀌는 것에 따라서 경로나 계정등만 달라진다.
138
139 # Driver loading
140 driver <- JDBC(driverClass = "oracle.jdbc.driver.OracleDriver", #라이브러리 경로 / 안에 OracleDriver라는게 있다
141               classPath = "C:\\oraclexe\\app\\oracle\\product\\11.2.0\\server\\jdbc\\lib\\ojdbc6.jar") #드라이브 파일을 도와주는 라이브러리가 있다.
142               #oraclexe의 경로에서 jar 파일을 찾는다. 경로는 다를 수 있다.
143 #드라이브와 연결 할 수 있는 path 가 된다. 내 드라이브 path다
144 conn <- dbConnect(driver, "jdbc:oracle:thin:@localhost:1521:xe", # jdbc:oracle:thin:@여기까지 고정이고 @ 다음 localhost는 내 아이피(127.0.0.1)를 말한다 그리고 xe는 DB 이름이다.
145                  "hr", #계정
146                  "hr") #비밀번호

```

#select 문

#dbGetQuery() : select

```

148 selectEmpQuery <- "select * from employee"
149
150 #dbGetQuery() : select
151 empTable <- dbGetQuery(conn, selectEmpQuery)
152 empTable
153
154 str(empTable) #데이터의 구조를 나타낸다.
155 dim(empTable) #행과 열의 개수만 리턴
156 # *str과 dim의 차이점
157
158
159
156:17 (Top Level) :

```

Console	Terminal	Jobs
~/ ➡		
1	100	한선기 621133-1483658 sg_ahn@vcc.com 0199949999 2090-04-01 13:30:30 J1 9000000 0.20 Y <NA> 90
2	101	강중훈 621136-1006405 jh_park@vcc.com 0193334433 2004-04-30 00:00:00 J2 5500000 NA Y 100 90
3	102	최만석 861011-1940062 ms_choi@vcc.com 0198879908 1995-12-30 00:00:00 J2 3600000 NA Y 101 90
4	103	정도연 631127-2519077 sy_kang@vcc.com 0196654436 1997-06-03 00:00:00 J4 2600000 NA Y 104 60
5	104	안석규 651031-1962810 sg_han@vcc.com 0192347654 1998-07-01 00:00:00 J3 3500000 0.25 Y 100 60
6	107	조재형 721128-1732822 jh_jeon@vcc.com 0193325548 1998-11-23 00:00:00 J3 3800000 NA Y 104 60
7	124	정지현 641231-2269080 jih_jeon@vcc.com 01922976129 2004-07-15 00:00:00 J7 1500000 NA N 141 50

create 문

dbSendUpdate() : DML(insert, update, delete), DDL(create, drop, alter)

```
158
159 # table생성
160 createTempTable <- "create table r_tbl(
161                               id varchar2(20) primary key,
162                               pwd varchar2(20),
163                               username varchar2(50),
164                               upoint number default 1000)"
165
166 # dbSendUpdate() : DML(insert, update, delete), DDL(create, drop, alter)
167
168 dbSendUpdate(conn, createTempTable)
```

#insert 문

```
171
172 # insert(dummy data)
173 insertSQL <- "insert into r_tbl values('jslim','jslim','섭섭이',2000)" #insert구문을 만들고
174 dbSendUpdate(conn,insertSQL) # dbSendUpdate 구문을 통해 conn을 통해 연결된 DB에 insert를 넣는다.
175
176 selectEmpQuery <- "select * from r_tbl" #select으로 확인한다.
177 dummyTbl <- dbGetQuery(conn,selectEmpQuery)
178
```

```
> dummyTbl
  ID   PWD USERNAME UPOINT
1 jslim jslim  섭섭이   2000
> # update
```

#update 문

```
181
182 # update
183 updateSQL <- "update r_tbl
184               set username='관리자'
185               where username='admins' " #update구문을 만들고|
186 dbSendUpdate(conn, updateSQL)
187
188 dummyTbl <- dbGetQuery(conn,selectEmpQuery)
189
190
```

```
> dummyTbl
  ID   PWD USERNAME UPOINT
1 jslim jslim  admins   2000
>
```

#delete 문

```
191 #delete
192
193 deleteSQL <- "delete from r_tbl
194             where username='관리자' " #delete구문을 만들고
195
196 dbSendUpdate(conn, deleteSQL)
197
198 dummyTbl <- dbGetQuery(conn,selectEmpQuery)
```

```
> dummyTbl
[1] ID      PWD      USERNAME UPOINT
<0 행> <또는 row.names의 길이가 0입니다>
```

#drop 문

```
201 #drop
202
203 dropSQL <- "drop table r_tbl"
204
205 dbSendUpdate(conn, dropSQL)
206
207 dummyTbl <- dbGetQuery(conn,selectEmpQuery)
208
```

```
204
205 dbSendUpdate(conn, "drop table r_tbl")
206
```

바로 넣어주는 형태도 가능하다.

```
<0 행> <모든 row.names의 길이가 0입니다>  
> dbSendUpdate(conn, dropSQL)  
> dummyTbl ← dbGetQuery(conn,selectEmpQuery)  
Error in .verify.JDBC.result(r, "Unable to retrieve JDBC result set", :  
  Unable to retrieve JDBC result set  
  JDBC ERROR: ORA-00942: table or view does not exist  
  
Statement: select * from r_tbl
```

'R' 카테고리의 다른 글

[R] R을 통한 텍스트마이닝에서 워드클라우드 까지

[R] R로 하는 비정형 데이터 처리 (facebook 데이터를 통한 긍정/부정 나누기)

[R] R에서 Database 사용하기 / DB 기본적인 구문 사용하기

[R] 예제를 통한 데이터 전처리 작업

[R] R을 통해 시계열 그래프 만들기 (자료 분석을 위한 시각화와 실습 예제)

[R] 같은 형태의 ggplot 과 barplot 만들기 (차이 비교)

R DB

R DB 사용

R 데이터베이스

R과 DB

R에서 DB 사용



꾸까꾸

혼자 끄적끄적하는 블로그 입니다.