Hands-on Lab: Accessing Your Database with RODBC

Welcome!

In this hands-on lab, we will learn how to connect and discover metadata from database servers with R using RODBC.

Tasks

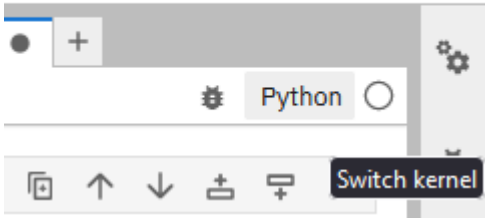
Estimated Time Needed: **15 min**

a. Pre-requisites

In this lab we will use Jupyter Notebooks within SN Labs to access data in a Db2 on Cloud database using RODBC.

b. Create an R notebook

If required, set the notebook kernel to R by clicking on the kernel on the top right hand corner:



c. Load RODBC

The RODBC package and the ODBC driver for Db2 are pre-installed on your workbench. Let's load the RODBC package by clicking on the following cell and executing it (Shift+Enter):

In [1]:
```
#install.packages("RODBC")
library(RODBC);
```

d. Connection information

To connect to your Db2 instance, you require the following details:

- Driver class
- Database name

- Hostname
- Port number
- Protocol
- Username
- Password

We will be using different variables to store this information, so that we can use these values at a later point of time when required.

Replace the values for **hostname, port number, username and password** by copying them from Service Credentials in your DB2 instance.

For instructions on accessing **Db2 Service Credentials**, go to Hands-on Lab: Create Db2 Service Credentials.

> Note:This is just an example screenshot of service credentials. However these values will vary with respect to the DB2 instance which you create.

▶ Click here to view/hide hint

In [2]:
```
#Enter the values for your database connection
dsn_driver = "com.ibm.db2.jcc.DB2Driver"
dsn_database = "bludb"
dsn_hostname = "19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"   #replace <yourhostname>
dsn_port = "30699"                    #replace with the port number from Service Credentials
dsn_protocol = "TCPIP"
dsn_uid = "ltw17738"            #replace <username> with your username from Service Credentials
dsn_pwd = "zOkHANl9GIFBAZak"            #replace <password> with your password from Service Credentials
dsn_security = "ssl"
```

e. Create a database connection

The next step is to create a connection string and connect to Db2 using odbcDriverConnect() function. **odbcDriverConnect()** takes this connection string as its parameter and returns a connection object.

In [3]:
```
conn_path <- paste("DRIVER=",dsn_driver,
                   ";DATABASE=",dsn_database,
```

```
                    ";HOSTNAME=",dsn_hostname,
                    ";PORT=",dsn_port,
                    ";PROTOCOL=",dsn_protocol,
                    ";UID=",dsn_uid,
                    ";PWD=",dsn_pwd,
                    ";SECURITY=",dsn_security,
                      sep="")
conn_path

conn <- odbcDriverConnect(conn_path)
conn
```

'DRIVER=com.ibm.db2.jcc.DB2Driver;DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;PROTOCOL=TCPIP;UID=ltw17738;PWD=zOkHANI9GIFBAZak;SECURITY

```
Warning message in odbcDriverConnect(conn_path):
“[RODBC] ERROR: state 01000, code 0, message [unixODBC][Driver Manager]Can't open lib 'com.ibm.db2.jcc.DB2Driver' : file not fo
und”Warning message in odbcDriverConnect(conn_path):
“ODBC connection failed”
```

-1

f. Connection Attributes

Let’s examine the connection attributes using the attributes() function:

In [4]:
```
attributes(conn)
```

NULL

g. Connection Metadata

And review the connection metadata using the odbcGetInfo() function. This function will provide details about the database name, version and the version of the ODBC driver:

In [5]:
```
conn.info = odbcGetInfo(conn)
conn.info["DBMS_Name"]
```

```
conn.info["DBMS_Ver"]
conn.info["Driver_ODBC_Ver"]
```

```
Error in odbcGetInfo(conn): argument is not an open RODBC channel
Traceback:

1. odbcGetInfo(conn)
2. stop("argument is not an open RODBC channel")
```

h. Supported Datatypes

Let's now examine the datatypes supported by the database using sqlTypeInfo() function. This function will return a dataframe having information about the supported datatypes. The dataframe will have 4 columns such as Type_Name, Data Type and Column Size.

In [6]:
```
sql.info = sqlTypeInfo(conn)
print(sql.info)
```

```
Error in sqlTypeInfo(conn): first argument is not an open RODBC channel
Traceback:

1. sqlTypeInfo(conn)
2. stop("first argument is not an open RODBC channel")
```

Let's print only the first and third column from the dataframe:

In [7]:
```
print(sql.info[c(1,3)], row.names=FALSE)
```

```
Error in print(sql.info[c(1, 3)], row.names = FALSE): object 'sql.info' not found
Traceback:

1. print(sql.info[c(1, 3)], row.names = FALSE)
```

i. List of Tables

We will use the sqlTables() function to return a dataframe with information about table-like objects (i.e. TABLEs, VIEWs, ALIASes, etc.) in the Db2 system Schema **SYSIBM** and save it in a variable called tab.frame. We will get the count of the tables in the schema using **nrow()**

function. We can then display their names using the TABLE_NAME column of the dataframe.

In [8]:
```
tab.frame = sqlTables(conn, schema="<Enter Schema>") # e.g. "SYSIBM"
nrow(tab.frame)
tab.frame$TABLE_NAME
```

```
Error in sqlTables(conn, schema = "<Enter Schema>"): first argument is not an open RODBC channel
Traceback:

1. sqlTables(conn, schema = "<Enter Schema>")
2. stop("first argument is not an open RODBC channel")
```

j. Columns in a Table

Next, let's look at column metadata for columns in the system catalog table **SYSSCHEMATA**. We will use **sqlColumns()** function which describes the column structure of tables on an ODBC database connection.

In [9]:
```
tab.name <- "<Enter Table>" # e.g. "SYSSCHEMATA"
col.detail <- sqlColumns(conn, tab.name)
print(col.detail[c(2,3,4,6,7,9,18)], row.names=FALSE)
```

```
Error in sqlColumns(conn, tab.name): first argument is not an open RODBC channel
Traceback:

1. sqlColumns(conn, tab.name)
2. stop("first argument is not an open RODBC channel")
```

k. Dis-connect

Finally, as a best practice we should close the database connection once we're done with it.

In [10]:
```
odbcCloseAll()
```

Practice exercises

1. Provide the database credentials for your instance of Db2

In [ ]:
```
#write your code here
```

▶ Click here to view/hide hint

▼ Click here to view/hide solution

```
#Enter the values for you database connection
dsn_driver = "com.ibm.db2.jcc.DB2Driver"
dsn_database = "bludb"            # e.g. "bludb"
dsn_hostname = "<yourhostname>"   # e.g. replace <yourhostname> with your hostname
dsn_port = ""                 # e.g. "3273"
dsn_protocol = "TCPIP"            # i.e. "TCPIP"
dsn_uid = "<username>"              # e.g. replace <username> with your userid
dsn_pwd = "<password>"            # e.g. replace <password> with your password
```

2. Create a connection string and connect to Db2.

In [ ]:
```
#write your code here
```

▶ Click here to view/hide hint

▼ Click here to view/hide solution

```
conn_path <- paste("DRIVER=",dsn_driver,
                   ";DATABASE=",dsn_database,
                   ";HOSTNAME=",dsn_hostname,
                   ";PORT=",dsn_port,
                   ";PROTOCOL=",dsn_protocol,
                   ";UID=",dsn_uid,
                   ";PWD=",dsn_pwd,
                   ";SECURITY=",dsn_security,
                     sep="")
conn <- odbcDriverConnect(conn_path)
conn
```

3. List of tables: Use the sqlTables() function to return a dataframe with information about table-like objects (i.e. TABLEs, VIEWs, ALIASes, etc.) in the Db2 system Schema SYSIBM and save it in a variable called tab. Display the count of the tables in the schema using nrow() function and their names using the TABLE_NAME column of the dataframe.

In [ ]:
```
#write your code here
```

▶ Click here to view/hide hint

▶ Click here to view/hide solution

4. Display the column metadata for columns in the IBM system catalog table SYSSTRINGS

In [ ]:
```
#write your code here
```

▶ Click here to view/hide hint

In [ ]:
```
<details>
<summary>Click here to view/hide solution</summary>
<p>

```
query = "SELECT * FROM SYSIBM.SYSSTRINGS";
rs = dbSendQuery(conn,query);
df = fetch(rs,20);
```
</p>
</details>
```

Summary

In this lab you accessed data in a Db2 on Cloud database using RODBC connection from a R notebook in Jupyter, and discovered different metadata.

Thank you for completing this lab on getting connected and querying databases using RODBC.

Authors

- Rav Ahuja
- Agatha Colangelo
- Sandip Saha Joy
- Shreya Khurana

In [ ]: