IBM Developer
SKILLS NETWORK

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

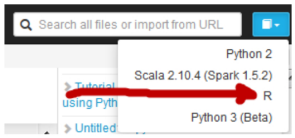
Estimated Time Needed: **15 min**

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. Information about Jupyter notebooks, SN Labs, and Db2 services is provided in the previous labs.

b. Create an R notebook

If necessary, set the notebook kernel to use R (using the dropdown in the top right 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):

[ ]:
```
library(RODBC);
```

d. Connection information

In the following cell enter the connection details for your instance of Db2 and run it. Remember to update the values for hostname, userid, and password.

For instructions on accessing **Db2 Service Credentials**, go to **Hands-on Lab: Setup your database service instance and Access service credentials**.

[ ]:
```
```

▼ Click here to view/hide hint

```
# Fill in the <...>
dsn_driver <- "{...}"
dsn_database <- "..."
dsn_hostname <- "<Enter Hostname>"
dsn_port <- "..."
dsn_protocol <- "..."
dsn_uid <- "<Enter UserID>"
dsn_pwd <- "<Enter Password>"
dsn_security <- "ssl"
```

▼ Click here to view/hide solution

```
dsn_driver <- "{IBM DB2 ODBC Driver}"
dsn_database <- "bludb"            # e.g. "bludb"
dsn_hostname <- "<Enter Hostname>" # e.g "54a2f15b-5c0f-46df-8954-.databases.appdomain.cloud"
```

```
dsn_port <- "<Enter port number"   # e.g. "32733"
dsn_protocol <- "TCPIP"            # i.e. "TCPIP"
dsn_uid <- "<Enter UserID>"        # e.g. "zjh17769"
dsn_pwd <- "<Enter Password>"      # e.g. "zcwd4+8gbq9bm5k4"
dsn_security <- "ssl"
```

e. Create a database connection

Create a connection string and connect:

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
conn_path <- paste("DRIVER=",...
                   ";DATABASE=",...
                   ";HOSTNAME=",...
                   ";PORT=",...
                   ";PROTOCOL=",...
                   ";UID=",...
                   ";PWD=",..."
                   ";SECURITY=",...")
conn <- ...(...)
```

▼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
```

f. Connection Attributes

Let's examine the connection attributes:

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
...(conn)
```

▼Click here to view/hide solution

```
attributes(conn)
```

g. Connection Metadata

And review the connection metadata using the odbcGetInfo() function:

[ ]:

▼Click here to view/hide hint

```
conn.... <- odbc...(conn)
```

▼Click here to view/hide solution

```
conn.info <- odbcGetInfo(conn)
```

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
conn....["DBMS_..."]
conn....["DBMS_..."]
conn....["Driver_..._..."]
```

▼Click here to view/hide solution

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

h. Supported Datatypes

Let's now examine the datatypes supported by the database:

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
sql.... <- sql...(conn)
print(sql....[c(1,...)], row....=FALSE)
```

▼Click here to view/hide solution

```
sql.info <- sqlTypeInfo(conn)
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**. First we will output the

number of tables in the schema, and then display their names.

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
tab.... <- sql...(conn, ...="<Enter Schema>")
nrow(tab....)
tab.frame$...
```

▼Click here to view/hide solution

```
tab.frame <- sqlTables(conn, schema="<Enter Schema>") # e.g. "SYSIBM"
nrow(tab.frame)
tab.frame$TABLE_NAME
```

j. Columns in a Table

Next, let's look at column metadata for columns in the system catalog table **SYSSCHEMATA**:

[ ]:

▼Click here to view/hide hint

```
# Fill in the ...
tab.... <- "...."
col.detail <- sql...(conn, tab....)
print(....detail[c(...,...,7,...,...)], row....=FALSE)
```

▼Click here to view/hide solution

```
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)
```

k. Dis-connect

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

[ ]:

▼Click here to view/hide hint

```
odbc...()
```

▼Click here to view/hide solution

```
odbcCloseAll()
```

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.

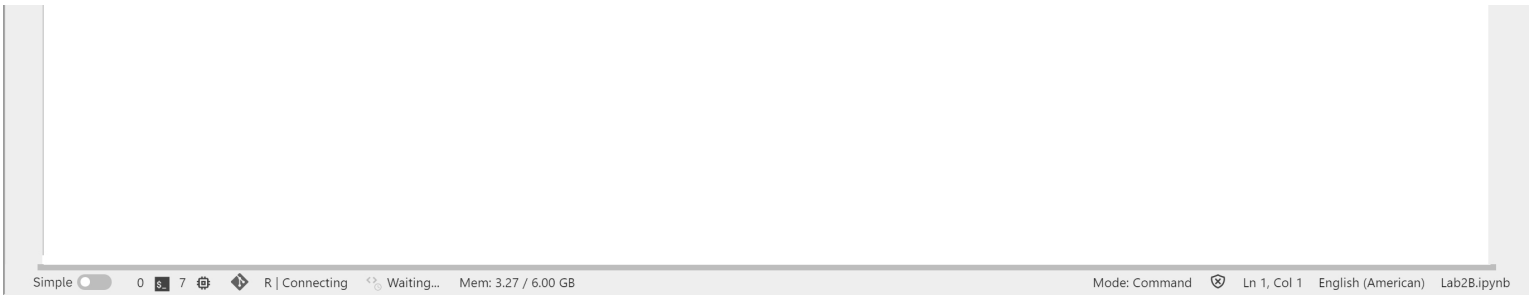
Authors

- Rav Ahuja
- Agatha Colangelo
- Sandip Saha Joy

Changelog

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-07-14	2.1	Lakshmi Holla	Added ssl changes
2021-01-22	2.0	Sandip Saha Joy	Created revised version of the lab
2017	1.0	Rav Ahuja & Agatha Colangelo	Created initial version of the lab

---


Simple
0
7
R | Connecting
Waiting...
Mem: 3.27 / 6.00 GB
Mode: Command
Ln 1, Col 1
English (American)
Lab2B.ipynb