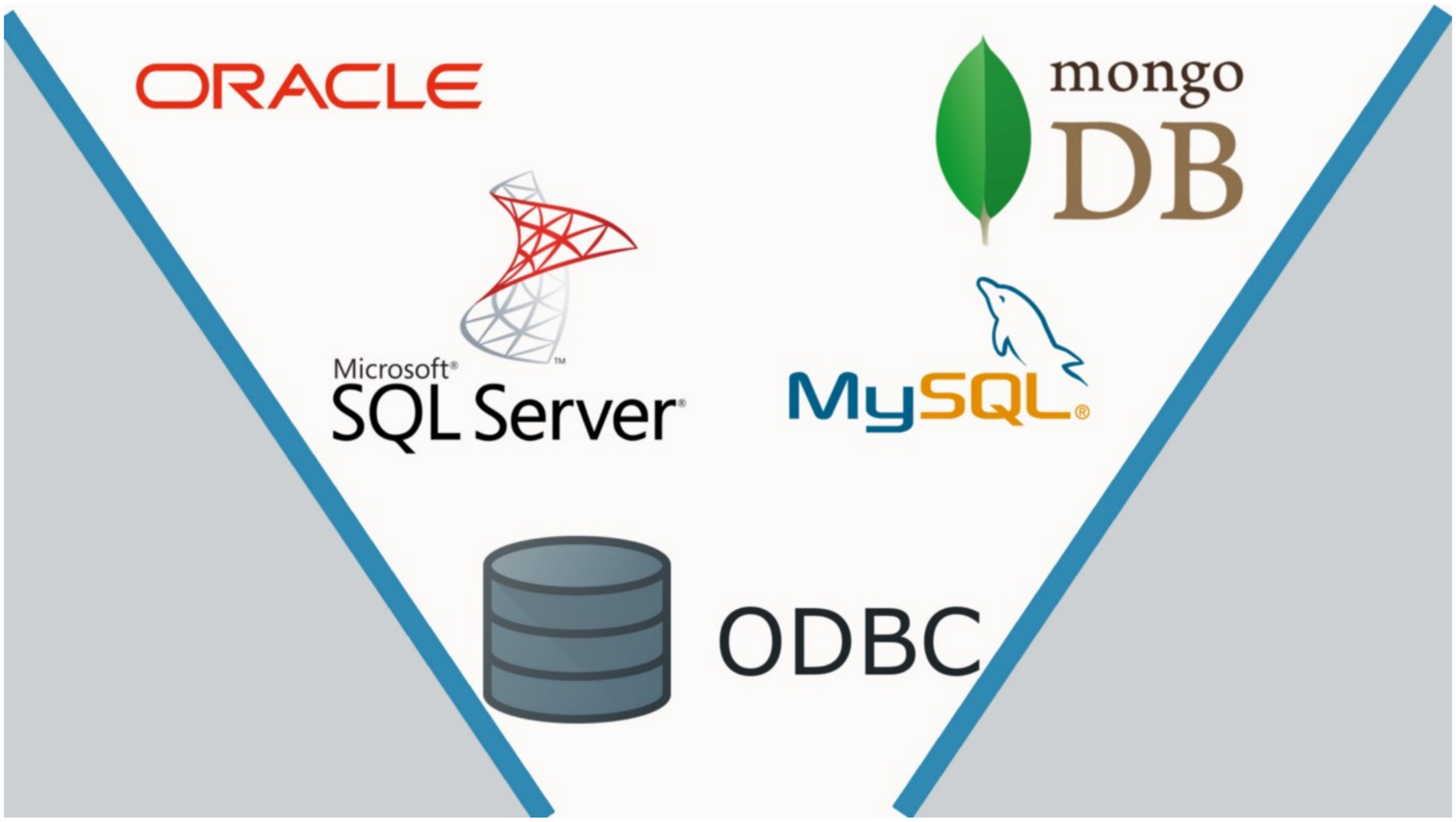
Using ODBC to connect any database directly to Jupyter notebook.

Psst, Also alternatives if by any case you don't like ODBC

Gaurav Chauhan Jan 29 · 5 min read ★



Whether you have stored your data locally or in a remote system, whatever of its type or database management system, ODBC is the solution.

Introduction

ODBC is an open interface that connects with almost any database management system using ODBC drivers. ODBC drivers are been provided by a database management system that helps to connect that particular database system.

So when Data Scientists or Analysts quickly needed to run queries in their favorable Jupyter NB, ODBC comes handy.

Now for connectivity of ODBC in R, you can follow other tutorials, as I will only be demonstrating python implementation.

Note that apart from ODBC, you can utilize other packages for connectivity, but most of these packages can connect only specific database management systems.

python package for ODBC: ***pyodbc***

MS-SQL

> ***Installation of MS-SQL driver — this is only required if data is on remote machine***

Download MS-SQL driver as per your OS

Windows

1. for windows, it's easy as installing within a click. ODBC for MS-SQL is available here.

2. Install the .msi file as per your bit.

Linux — Ubuntu

```
sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add
-

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#Ubuntu 14.04
curl https://packages.microsoft.com/config/ubuntu/14.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Ubuntu 16.04
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Ubuntu 18.04
curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Ubuntu 18.10
curl https://packages.microsoft.com/config/ubuntu/18.10/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Ubuntu 19.04
curl https://packages.microsoft.com/config/ubuntu/19.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list

exit
sudo apt-get update
sudo ACCEPT_EULA=Y apt-get install msodbcsql17
# optional: for bcp and sqlcmd
sudo ACCEPT_EULA=Y apt-get install mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
# optional: for unixODBC development headers
sudo apt-get install unixodbc-dev
```

Linux — Debian

```
sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add
-

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version
```

```
#Debian 8
curl https://packages.microsoft.com/config/debian/8/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Debian 9
curl https://packages.microsoft.com/config/debian/9/prod.list >
/etc/apt/sources.list.d/mssql-release.list

#Debian 10
curl https://packages.microsoft.com/config/debian/10/prod.list >
/etc/apt/sources.list.d/mssql-release.list

exit
sudo apt-get update
sudo ACCEPT_EULA=Y apt-get install msodbcsql17
# optional: for bcp and sqlcmd
sudo ACCEPT_EULA=Y apt-get install mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
# optional: for unixODBC development headers
sudo apt-get install unixodbc-dev
# optional: kerberos library for debian-slim distributions
sudo apt-get install libgssapi-krb5-2
```

Linux — RedHat

```
sudo su

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#RedHat Enterprise Server 6
curl https://packages.microsoft.com/config/rhel/6/prod.repo >
/etc/yum.repos.d/mssql-release.repo

#RedHat Enterprise Server 7
curl https://packages.microsoft.com/config/rhel/7/prod.repo >
/etc/yum.repos.d/mssql-release.repo

#RedHat Enterprise Server 8
curl https://packages.microsoft.com/config/rhel/8/prod.repo >
/etc/yum.repos.d/mssql-release.repo

exit
sudo yum remove unixODBC-utf16 unixODBC-utf16-devel #to avoid
conflicts
sudo ACCEPT_EULA=Y yum install msodbcsql17
# optional: for bcp and sqlcmd
```

```
sudo ACCEPT_EULA=Y yum install mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
# optional: for unixODBC development headers
sudo yum install unixODBC-devel
```

Mac OS

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew tap microsoft/mssql-release
https://github.com/Microsoft/homebrew-mssql-release
brew update
brew install msodbcsql17 mssql-tools
```

Install pyodbc — ODBC package for python

1. install pyodbc package.

```
pip install pyodbc
```

2. To check whether the driver has installed properly, find all the drivers connected to pyodbc.

```
import pyodbc
pyodbc.drivers()
```

for MS-SQL it will result in

```
['ODBC Driver 17 for SQL Server']
```

As more drivers you will add to your system, more drivers will be added in the list.

Connect to database

1. For remote connection.

```
# enter ip address and port number of the system where the database
resides.
server = 'tcp:31.288.186.65,49170'
database = 'database_name' # enter database name
username = 'user_name'
password = 'pass_word'

# add appropriate driver name
cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL
Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD=
'+ password)

cursor = cnxn.cursor()
```

2. For Local connection (if data is in your local computer).

```
server = 'tcp:31.288.186.65,49170'
database = 'database_name' # enter database name

cnxn = pyodbc.connect('DRIVER={SQL
Server};SERVER='+server+';DATABASE='+database+';Trusted_Connection=ye
s;')

cursor = cnxn.cursor()
```

Query the database

1. you can query the database ie, select, insert, update or delete in your notebook.

your query can be directly converted to pandas DataFrame.

```
import pandas as pd

# select command
query = ''' SELECT RecordID FROM tables''';
```

```
data = pd.read_sql(query, cnxn)
data.head()
```

Alternatives

Alternatively, you can use pymssql which works the same but it has been discontinued. Still, if you want to use it, you have to install by

```
pip install "pymssql<3.0"
```

ORACLE

Oracle also supports ODBC databases. You need to install an ODBC driver for oracle as per your OS.

1. Install Oracle ODBC client.

Download instant client basic and odbc package from the oracle website.

2. If the downloaded package is zip, use ***wget*** to extract or ***rpm*** for rpm file.

3. set ORACLE_HOME and LD_LIBRARY_PATH

```
export ORACLE_HOME=/usr/lib/oracle/12.2/client64
export LD_LIBRARY_PATH=/usr/lib/oracle/12.2/client64/lib
```

change path “/12.2/” as per your version.

4. In /etc/odbcinst.ini set:

```
[oracle_db]
Description=Oracle ODBC driver for Oracle 12c
Driver=/usr/lib/oracle/12.2/client64/lib/libsqora.so.12.1
FileUsage=1
```

```
Driver Logging=7
UsageCount=1
```

5. Now open Jupyter NB and you can easily connect to your oracle database.

```
import pyodbc
conn = pyodbc.connect('DRIVER=
{oracle_db};Host=1.1.1.1;Port=1521;Service Name=orcl.local;User
ID=test1;Password=test1')
```

Alternatives

Alternatively, you can use Cx_Oracle which works similarly but only for oracle databases.

MY-SQL

1. For MySQL its very much similar to the above configuration where you need to install appropriate drivers of MySQL.

```
import pyodbc
conn = pyodbc.connect('DRIVER
{MySQL};SERVER=localhost;DATABASE=test;UID=root;PWD=abc;')
```

Alternatives

For Mysql, you have many packages to connect to python like

- MySQLdb
- mysql.connector

MONGO DB

1. For MongoDB, install the MongoDB driver and connect to pyodbc.

Alternatives

Alternatively, you can use pymongo for connectivity with python.

Conclusion

As per connectivity goes, there are more than one packages to connect the databases, but if you want only one generalized package, pyodbc is the package you want which uses the ODBC interface to access data independent of database systems.

> Further, for any clarification, comment, help, appreciation or suggestions just post it in comments and I will help you in that.
>
> If you liked it you can follow me on medium.

Gaurav Chauhan - Medium

Read writing from Gaurav Chauhan on Medium. Data Scientist at hopscotch.health. Every day, Gaurav Chauhan and thousands…

medium.com

> Also, you can connect me on my social media if you want more content based on Data Science, Artificial Intelligence or Data Pipeline.

- Linkedin
- Email
- Twitter
- Medium

Database Data Data Science Machine Learning Odbc