AA - Python to Hive (ODBC)

This page serves as a platform for connecting to hive directly from the Python environment through an ODBC Connection. This pattern is tailored for the purpose of exporting data from Hive and being able to manipulate it. It is recommended that all data management (as much as possible) is done on Hue as this process can be slower.

Pre-requisites for Hive

The correct version of the hive ODBC driver needs to be installed locally. Please see details below:

Hive ODBC driver version	Bit	Link location
2.6.9 (or later)	32 or 64 depending on your operating system	https://www.cloudera.com/downloads/connectors/hive/odbc/2-6-9.html

NB:

This Link takes takes you through to the landing page. The user must select the appropriate version i.e. 2.6.9

Once the driver has been downloaded :

- Install ClouderaHiveODBCxx.msi (with xx replaced with 32 or 64 depending on your operating system).

The following Pre-requisites are required on the Python front once the ODBC driver has been set up:

Software requirements

Requirement	Description	version
Base Python	The Code engine	latest version
(Optional) Jupyter Notebook or any prefrered IDE	IDE for Python	latest version

Python Package requirements:

Ensure that these packages are installed before the execution of the R script below. This will ensure that the connection to Hive is successful

Package Name
pyodbc
getPass
pandas

The following code will prompt you to enter your e-code and password as inputs.

Python Script: R ODBC

```
################################## Script to Create a Connection to Hive from Python
#############################################

import pyodbc
import pandas as pd
import getpass

user = input('Username:\n')
pw = getpass.getpass(prompt='Password:\n')

def connect_to_hive(dsn="Cloudera Andromedia Hive",
                    host="cloudera.sanlam.co.za",
                    port=10000,
                    username=user,
                    password=pw):
    """
    Connect to a Hive database using a DSN, host, port & credentials.
    """
    try:
        conn = pyodbc.connect(f"DSN={dsn};HOST={host};PORT={port};UID={username};PWD={password};
DATABASE=information_schema", autocommit=True)
        return conn
    except pyodbc.Error as e:
        print("Error connecting to Hive database:", e)
        return None


# Example usage:

conn = connect_to_hive(password=pw)

# Use conn object to read data to pandas dataframe

df = pd.read_sql_query("SELECT DISTINCT(businessentityname) FROM groupbi_lablive_spflab.par", conn) ##########
change the query if you want to view something else

conn.close()
```