Download this <https://code.visualstudio.com/>

Downloaded

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Beni Ungur | Jason Goertz | Kylie Dillon | Devon Gronquist | William Judge |
| JaeWor | Krissi postor | Matt dubray | Han Ji | Bryan Aleman |
| Alex Inclan | Chris Ellis | Natalia Solar | Alex Kreuch | Mark Ignatovich |
| Tamires B | Mark Dolinar | Matthew Smithers |  | Hannah Geer |
| JaeWook | Christian |  |  | Reuel Mendoza |

import pyodbc

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

with pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password) as conn:

with conn.cursor() as cursor:

cursor.execute("SELECT TOP 3 name, collation\_name FROM sys.databases")

row = cursor.fetchone()

while row:

print (str(row[0]) + " " + str(row[1]))

row = cursor.fetchone()

Python:

<https://www.python.org/downloads/release/python-370/>

Get-Pip

<https://www.liquidweb.com/kb/install-pip-windows/>

ODBC Driver

<https://www.microsoft.com/en-us/download/details.aspx?id=56567>

For Mac

<https://docs.microsoft.com/en-us/sql/connect/odbc/linux-mac/install-microsoft-odbc-driver-sql-server-macos?view=sql-server-ver15>

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"

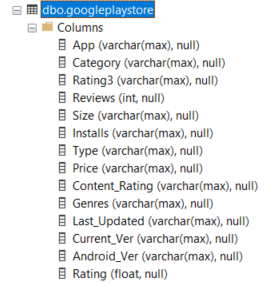
brew tap microsoft/mssql-release https://github.com/Microsoft/homebrew-mssql-release

brew update

HOMEBREW\_NO\_ENV\_FILTERING=1 ACCEPT\_EULA=Y brew install msodbcsql17 mssql-tools

sudo ln -s /usr/local/etc/odbcinst.ini /etc/odbcinst.ini

sudo ln -s /usr/local/etc/odbc.ini /etc/odbc.ini



Problem

**Average Rating (float, null)** NOT Rating3, round to 2 digits

Across all the apps (all the rows).

|  |  |
| --- | --- |
| Group | Average? |
| 1 | 4.19 |
| 2 |  |
| 3 | 4.19 |
| 4 | 4.3 |
| 5 |  |
| 6 |  |
| 7 | 4.19 |

**When doing Data Analysis, if you can do SQL do SQL only for the parts you can’t do SQL do Python.**

**For a linear regression SQL can technically do it with a lot SQL (pick python)**

import pyodbc as db

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

with db.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password) as conn:

with conn.cursor() as cursor:

cursor.execute("SELECT Rating FROM googleplaystore")

total = 0.0

count = 0

row = cursor.fetchone()

while row:

#print (str(row[0]))

if row[0]:

total += float(row[0])

count += 1

row = cursor.fetchone()

print("Average rating: ", round((total/count), 2))

Group 7

import pyodbc

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

with pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password) as conn:

with conn.cursor() as cursor:

cursor.execute("SELECT count(\*), round(avg(Rating),2) FROM dbo.googleplaystore")

row = cursor.fetchone()

while row:

print (str(row[0]) + " " + str(row[1]))

row = cursor.fetchone()

How to turn your SQL result into pandas df

<https://stackoverflow.com/questions/39835770/read-data-from-pyodbc-to-pandas>

data = pandas.read\_sql(sql,cnxn)`

**Break - Back at 7:40**

****

**Problem**

Use a combination of SQL and Python to predict the population of Asia in 2015 based on the data on the table.

|  |  |
| --- | --- |
| Group | Prediction |
| 1 | 4,023,475,003 |
| 2 |  |
| 3 | 1.183375e+08 |
| 4 | 4.023475e+09 |
| 5 |  |
| 6 |  |
| 7 | 4023475003 |

import pandas

import pyodbc

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

cnxn = pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password)

sql = "SELECT Year, sum([Population Total]) as 'Population Total' FROM WorldIndicators WHERE Region = 'Asia' Group By Year"

byyear = pandas.read\_sql(sql,cnxn)

import numpy

from sklearn.linear\_model import LinearRegression

x = numpy.array(byyear["Year"]).reshape(-1, 1) # reshape makes x a 2d array

y = list(byyear["Population Total"])

lm = LinearRegression().fit(x, y)

print(byyear)

print(lm.predict(numpy.array([[2015]])))

Thomas: This code does not group by year 👇

import pyodbc

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

with pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password) as conn:

sql = "SELECT \* FROM WorldIndicators WHERE Region='Asia'"

data = pd.read\_sql(sql, conn)

data = pd.DataFrame(data)

x = pd.DataFrame(data['Year'])

y = pd.DataFrame(data['Population Total'])

model = LinearRegression().fit(x,y)

newdata = np.array([[2015]])

prediction = model.predict(newdata)

print(prediction[0][0])

import pyodbc

import numpy as np

import pandas

from pandas import DataFrame

server = 'ludsampledb.database.windows.net'

database = 'SampleDB'

username = 'sampleadmin'

password = '+U9Ly9/p'

driver= '{ODBC Driver 17 for SQL Server}'

with pyodbc.connect('DRIVER='+driver+';SERVER='+server+';PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password) as conn:

data = pandas.read\_sql("SELECT SUM([Population Total]) as pop, Year FROM dbo.WorldIndicators WHERE Region = 'Asia' GROUP BY Year",conn)

print(data)

model = np.polyfit(data["Year"], data["pop"],1)

predict = np.poly1d(model)

year = 2015

print(predict(year))

Data from SQL, use as much SQL as possible to prepare the data. Then use python to do complex statistics calculations.