**Pandas DataFrame Example**

In order to use Pandas library in Python, you need to import it using import pandas as pd.

The below example creates a Pandas DataFrame from the list.

import pandas as pd

data = [["James","","Smith",30,"M",60000],

["Michael","Rose","",50,"M",70000],

["Robert","","Williams",42,"",400000],

["Maria","Anne","Jones",38,"F",500000],

["Jen","Mary","Brown",45,None,0]]

columns=['First Name','Middle Name','Last Name','Age','Gender','Salary']

# Create the pandas DataFrame

pandasDF=pd.DataFrame(data=data, columns=columns)

# print dataframe.

print(pandasDF)

Outputs below data on the console. Note that Pandas add an index sequence number to every data frame.

## Pandas Transformations

Below are some transformations you can perform on Pandas DataFrame. Note that statistical functions calculate at each column by default. you don’t have to explicitly specify on what column you wanted to apply the statistical functions. Even count() function returns count of each column (by ignoring null/None values).

* df.count() – Returns the count of each column (the count includes only non-null values).
* df.corr() – Returns the correlation between columns in a data frame
* df.head(n) – Returns first n rows from the top.
* df.max() – Returns the maximum of each column.
* df.mean() – Returns the mean of each column.
* df.median() – Returns the median of each column.
* df.min() – Returns the minimum value in each column.
* df.std() – Returns the standard deviation of each column
* df.tail(n) – Returns last n rows.
* print(pandasDF.count())
* First Name 5
* Middle Name 5
* Last Name 5
* Age 5
* Gender 4
* Salary 5
* print(pandasDF.max())
* First Name Robert
* Middle Name Rose
* Last Name Williams
* Age 50
* Salary 500000
* print(pandasDF.mean())
* Age 41.0
* Salary 206000.0

# PySpark createOrReplaceTempView() Explained

A Temporary view in PySpark is similar to a real SQL table that contains rows and columns but the view is not materialized into files. In this article, we will be discussing what is createOrReplaceTempView() and how to use it to create a temporary view and run PySpark SQL queries.

**. Syntax of createOrReplaceTempView()**

Following is the syntax of the createOrReplaceTempView() method. This method is from the Dataset class. So you can use this method from PySpark DataFrame and Dataset objects.

#Syntax of createOrReplaceTempView()

createOrReplaceTempView(viewName)

**How Does createOrReplaceTempView() workin PySpark?**

createOrReplaceTempView() in PySpark creates a view only if not exist, if it exits it replaces the existing view with the new one. PySpark SQL views are lazily evaluated meaning it does not persist in memory unless you cache the dataset by using the cache() method.

Some KeyPoints to note:

* createOrReplaceTempView() is used when you wanted to store the table for a specific SparkSession.

Use saveAsTable() to materialize the contents of the DataFrame and create a pointer to the data in the metastore.

## Create a Temporary View

The createOrReplaceTempView() is used to create a temporary view/table from the PySpark DataFrame or Dataset objects. Since it is a temporary view, the lifetime of the table/view is tied to the current SparkSession. Hence, It will be automatically removed when your SparkSession ends.

If you are using an older version prior to PySpark 2.0, you can use registerTempTable() to create a temporary table.

Following are the steps to create a temporary view in PySpark and access it.

* Step 1: Create a PySpark DataFrame
* Step 2: Convert it to an SQL table (a.k.a view)
* Step 3: Access view using SQL query

### 3.1 Create a DataFrame

First, let’s create a PySpark DataFrame with columns firstname, lastname, country and state columns.

from pyspark.sql import SparkSession

# Create spark session

spark = SparkSession \

.builder \

.appName("SparkByExamples.com") \

.enableHiveSupport() \

.getOrCreate()

data = [("James","Smith","USA","CA"),

("Michael","Rose","USA","NY"),

("Robert","Williams","USA","CA"),

("Maria","Jones","USA","FL")

]

columns = ["firstname","lastname","country","state"]

# Create dataframe

sampleDF = spark.sparkContext.parallelize(data).toDF(columns)

sampleDF.show()

### 3.2 Create Table/View using PySpark createOrReplaceTempView()

Let’s create a temporary view on top of the DataFrame object by using df.createOrReplaceTempView(). PySpark SQL temporary views are session-scoped and will not be available if the session that creates it terminates. The following examples create a Person table.

# Create Temporary View/Table

sampleDF.createOrReplaceTempView("Person")

If you want to have a temporary view that is shared among all sessions and keep alive until the PySpark application terminates, you can create a global temporary view using createGlobalTempView()Ezoic

### 3.3 Access View using PySpark SQL Query

Using SparkSession you can access PySpark SQL capabilities in Apache PySpark. Once you have a temporary view you can run any ANSI SQL queries using spark.sql() method.

# Run SQL Query

spark.sql("select firstname, lastname from Person").show()

Yields below output.

## Use createOrReplaceTempView() on AzureDatabricks

Below is a simple snippet on how to use createOrReplaceTempView() on Azure Databricks and how to access it using PySpark SQL query.