**PYSPARK :-**

**Read S3 data in pyspark by using boto3 with pycharm**

from pyspark.sql import SparkSession  
import boto3  
  
# Create a SparkSession  
spark = SparkSession.builder \  
 .appName("Read S3 Data") \  
 .getOrCreate()  
  
# Set your AWS access key and secret key  
access\_key = "AKIAYT4VWTLNSK7LNTZW"  
secret\_key = "JFEVk80UMFVE4DEit3NUvjwBaBexq/CPMwJ6jXDH"  
  
# Create a Boto3 S3 client  
s3\_client = boto3.client('s3', aws\_access\_key\_id=access\_key, aws\_secret\_access\_key=secret\_key)  
  
# Specify the S3 bucket and file path  
bucket\_name = "kushilkumars3"  
file\_path = "emptable.csv"  
  
# Download the file locally  
local\_file\_path = "s3\_file.csv"  
s3\_client.download\_file(bucket\_name, file\_path, local\_file\_path)  
  
# Read the local file into a DataFrame  
df = spark.read.csv(local\_file\_path, header=True, inferSchema=True)  
  
# Display the DataFrame  
df.show()  
  
# Stop the SparkSession  
spark.stop()

**Read S3 data in pyspark without using boto3 with pycharm and some increments and some filters are applied here.**

from pyspark.sql import SparkSession  
from pyspark.sql.functions import col  
  
# Create a SparkSession  
spark = SparkSession.builder \  
 .appName("Read S3 Data and increment salary") \  
 .config("spark.hadoop.fs.s3a.access.key", "AKIAYT4VWTLNSK7LNTZW") \  
 .config("spark.hadoop.fs.s3a.secret.key", "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") \  
 .config("spark.hadoop.fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem") \  
 .getOrCreate()  
  
# Specify the S3 bucket and file path  
bucket\_name = "kushilkumars3"  
file\_path = "emptable.csv"  
  
# Read the S3 file into a DataFrame  
df = spark.read.csv("s3a://{}/{}".format(bucket\_name, file\_path), header=True, inferSchema=True)  
# df\_with\_increased\_sal = df.withColumn('sal', col('sal') + 2000)  
# df\_with\_increased\_sal.show()  
  
# df.show()  
df.createOrReplaceTempView("my\_table")  
filtered\_df = spark.sql("SELECT distinct empno,ename,mgr,hiredate,sal,comm,deptno from my\_table WHERE deptno=20")  
#  
# # Display the DataFrame  
filtered\_df.show()  
  
# Stop the SparkSession  
spark.stop()

**Read and convert local CSV file in to excel in pyspark with pychharm.**

from pyspark.sql import SparkSession  
  
  
spark = SparkSession.builder.getOrCreate()  
  
df = spark.read.csv("C:/Users/SSS2022259/OneDrive/Desktop/lll/emptable.csv", header=True, inferSchema=True)  
df.printSchema()  
df.show()  
  
  
df.toPandas().to\_excel("C:/Users/SSS2022259/Downloads/archive/ex.xlsx", index=False)

**To convert EXCEL file to JSON file :-**

from pyspark.sql import SparkSession  
import pandas as pd  
import json  
  
spark = SparkSession.builder.getOrCreate()  
  
df = pd.read\_excel('C:/Users/SSS2020222/Desktop/IPL\_CAP.xlsx', sheet\_name='Sheet1')  
  
json\_string = df.to\_json(orient='records')  
  
with open('C:/Users/SSS2020222/Desktop/IPL\_CAP12.json', 'w') as f:

f.write(json\_string)

**To convert CSV to JSON file :-**

from pyspark.sql import SparkSession  
import pandas as pd  
import json  
  
spark = SparkSession.builder.getOrCreate()  
  
df = pd.read\_csv('C:/Users/SSS2022259/Downloads/archive/webscra.csv',encoding="latin-1")  
  
json\_string = df.to\_json(orient='records')  
  
with open('C:/Users/SSS2022259/Downloads/archive/exjsoncsv.json', 'w') as f:  
 f.write(json\_string)

**To convert JSON file to CSV file :-**

from pyspark.sql import SparkSession  
from pyspark.sql import functions as f  
  
# Set up the SparkSession  
spark = SparkSession.builder \  
 .appName("YourAppName") \  
 .getOrCreate()  
  
# Read the JSON file into a DataFrame  
df = spark.read.option("multiline", "true").json("C:/Users/SSS2022259/Downloads/archive/monthly\_sales.json")  
  
# Flatten the nested structure  
flattened\_df = df.withColumn("contents", f.explode(f.col("contents"))).withColumn("monthlySales", f.explode(f.col("contents.monthlySales"))).select("contents.category", "contents.region", "monthlySales.month", "monthlySales.sales")  
  
flattened\_df.write.csv("C:/Users/SSS2022259/OneDrive/Desktop/lll/Pyspark/json2csv2.csv",header='true')

**To Connect Pyspark to Oracle database:-**

from pyspark.sql import SparkSession  
# Step 1: Create a SparkSession  
spark = SparkSession.builder \  
 .appName("PySpark Oracle Connection") \  
 .getOrCreate()  
# Step 2: Define Oracle connection properties  
connection\_properties = {  
 "user": "C##DUMMY",  
 "password": "DUMMY",  
 "driver": "oracle.jdbc.driver.OracleDriver",  
 "url": "jdbc:oracle:thin:@//127.0.0.1"  
}  
# Step 3: Read data from Oracle into a DataFrame  
df = spark.read \  
 .format("jdbc") \  
 .option("query", "SELECT \* FROM kushil") \  
 .options(\*\*connection\_properties) \  
 .load()  
# Step 4: Perform operations on the DataFrame  
df.show()  
# Additional operations on the DataFrame...  
# df.select(...)  
# Step 5: Stop the SparkSession  
spark.stop()

**How to create a table in pyspark and that table stored or updated in Oracle SQL.**

### create table from pyspark and stored in oracle  
  
  
from pyspark.sql import SparkSession  
spark = SparkSession.builder \  
 .appName("CreateTableExample") \  
 .getOrCreate()  
# jdbc\_url = "jdbc:oracle:thin:@<host>:<port>/<service\_name>"  
jdbc\_url = "jdbc:oracle:thin:@localhost:1521:xe"  
  
connection\_properties = {  
 "user": "C##DUMMY",  
 "password": "DUMMY",  
 "driver": "oracle.jdbc.driver.OracleDriver"  
}  
  
data = [("kushil kumar", 25,6300624859), ("srikar", 30,9898989898), ("manoj", 35,8989898989),("nikhil", 26,7878787878),("sudheer", 25,8787878787)]  
columns = ["name", "age","mbno"]  
df = spark.createDataFrame(data, columns)  
df.write \  
 .format("jdbc") \  
 .option("url", jdbc\_url) \  
 .option("dbtable", "kings") \  
 .mode("overwrite") \  
 .options(\*\*connection\_properties) \  
 .save()