Write an SQL query to find the second-highest salary from an "Employees" table

SELECT MAX(salary) AS second\_highest\_salary

FROM Employees

WHERE salary < (SELECT MAX(salary) FROM Employees);

Write a MapReduce program to calculate the word count of a given input text file.

from mrjob.job import MRJob

import re

# Regular expression pattern to match words

WORD\_REGEX = re.compile(r"[\w']+")

class WordCount(MRJob):

# Mapper function

def mapper(self, \_, line):

# Extract words using the regular expression pattern

words = WORD\_REGEX.findall(line)

for word in words:

# Emit each word as a key and value 1

yield word.lower(), 1

# Reducer function

def reducer(self, word, counts):

# Sum up the counts for each word and emit the word as key and its count as value

yield word, sum(counts)

if \_\_name\_\_ == '\_\_main\_\_':

# Run the WordCount MapReduce job

WordCount.run()

python word\_count.py file1.txt

Write a Spark program to count the number of occurrences of each word in a given text file.

from pyspark.sql import SparkSession

from pyspark.sql.functions import split, explode

# Create a SparkSession

spark = SparkSession.builder.appName("WordCount").getOrCreate()

# Read the text file into a DataFrame

df = spark.read.text("input.txt")

# Split the lines into words

words = df.select(explode(split(df.value, " ")).alias("word"))

# Count the occurrences of each word

word\_counts = words.groupBy("word").count()

# Show the word counts

word\_counts.show()

# Stop the SparkSession

spark.stop()