In [ ]:
```
%pip install spark
%pip install pyspark

from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("example") \
    .getOrCreate()
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-w
heels/public/simple/
Requirement already satisfied: spark in /usr/local/lib/python3.9/dist-packages
(0.2.1)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-w
heels/public/simple/
Requirement already satisfied: pyspark in /usr/local/lib/python3.9/dist-packag
es (3.3.2)
Requirement already satisfied: py4j==0.10.9.5 in /usr/local/lib/python3.9/dist
-packages (from pyspark) (0.10.9.5)
```

In [ ]:
```
# DataFrames
# Use the DataFrames in Apache Spark to calculate the following:
# 1. Fatalities per month (12.5 pts): Calculate the fatalities per month. Sho
# with the most fatalities, in descending order.
# 2. Fatalities per year (12.5 pts): Calculate the fatalities per year. Show
# the most fatalities, in descending order.

from pyspark.sql.functions import year, month, countDistinct

fatalities_df = spark.read.format("csv").option("header", True).load("combine

# Create new columns 'fatality_month' from 'FAT_YEARMONTH' column
fatalities_df = fatalities_df.withColumn('fatality_month', fatalities_df['FAT_

# Perform aggregation by month and count the distinct fatality IDs
fatalities_by_month = fatalities_df.groupby('fatality_month').agg(countDistin

# Create new columns 'fatality_year'
fatalities_df = fatalities_df.withColumn('fatality_year', substring('FAT_YEAR

# Perform aggregation by year and count the distinct fatality IDs
fatalities_by_year = fatalities_df.groupby('fatality_year') \
                                   .agg(countDistinct('FATALITY_ID').alias('f
                                   .orderBy('fatalities', ascending=False)

# Show the results
fatalities_by_year.show(10)
fatalities_by_month.show()
```

```
+-------------+----------+
|fatality_year|fatalities|
+-------------+----------+
|         2005|      1453|
|         2011|      1335|
|         2018|      1050|
|         2021|       984|
|         1999|       906|
|         2020|       901|
|         2008|       825|
|         2017|       775|
|         2019|       732|
|         2007|       712|
+-------------+----------+
only showing top 10 rows

+--------------+----------+
|fatality_month|fatalities|
+--------------+----------+
|            07|      3381|
|            08|      3248|
|            06|      1955|
|            04|      1697|
|            05|      1634|
+--------------+----------+
```

In [ ]:

```
# SparkSQL
# Use SparkSQL to calculate the following:
# 1. Fatalities per month (12.5 pts): Calculate the fatalities per month. Sho
# with the most fatalities, in descending order.
# 2. Fatalities per year (12.5 pts): Calculate the fatalities per year. Show
# the most fatalities, in descending order.


from pyspark.sql import SparkSession

# Create SparkSession
spark = SparkSession.builder.appName("fatalities_analysis").getOrCreate()

# Load fatalities data from CSV file
fatalities_df = spark.read.format("csv").option("header", True).load("combine

# Register DataFrame as temporary view
fatalities_df.createOrReplaceTempView("fatalities")

# Perform SQL query to calculate fatalities by year
fatalities_by_year = spark.sql("SELECT SUBSTR(FAT_YEARMONTH, 1, 4) as fatalit
                                FROM fatalities \
                                GROUP BY fatality_year \
                                ORDER BY fatalities DESC")

# Create a temporary view for SparkSQL queries
fatalities_df.createOrReplaceTempView("fatalities")

# Perform the SQL query to calculate fatalities by month
fatalities_by_month = spark.sql("SELECT SUBSTRING(FAT_YEARMONTH, 5, 2) AS fat

# Show the result
fatalities_by_year.show(10)
fatalities_by_month.show()
```

```
+-------------+----------+
|fatality_year|fatalities|
+-------------+----------+
|         2005|      1453|
|         2011|      1335|
|         2018|      1050|
|         2021|       984|
|         1999|       906|
|         2020|       901|
|         2008|       825|
|         2017|       775|
|         2019|       732|
|         2007|       712|
+-------------+----------+
only showing top 10 rows

+--------------+----------+
|fatality_month|fatalities|
+--------------+----------+
|            07|      3381|
|            08|      3248|
|            06|      1955|
|            04|      1697|
|            05|      1634|
+--------------+----------+
```