Create a new spark job that will process the wikiticker data and filter it based on the following two conditions:

1. cityName is "London"
2. delta edits are greater than 20

Put the appropriate print statements in the Spark job, including the printing of the total count of records post the filtering. Ensure that you write out the data in a separate folder in S3.

Ensure that on running the Spark job the total number of records post the filtering is 10.

Title - Spark Job for Filtering and Processing Wikiticker Data

Objective - The objective of this exercise is to create a Spark job that processes the Wikiticker data and filters it based on specific conditions. The job should filter the data to include only records where the cityName is "London" and the delta edits are greater than 20. Additionally, the job should print the total count of records after filtering and write the filtered data to a separate folder in S3.

Tasks to Perform:

1. Create a new Spark job using the Spark framework.
2. Load the Wikiticker data into a Spark DataFrame.
3. Apply filters to the DataFrame to include records where the cityName is "London" and the delta edits are greater than 20.
4. Print appropriate statements in the Spark job, including the total count of records after filtering.
5. Write the filtered data to a specified folder in S3.

Learning Take Aways:

- Hands-on experience in creating and running Spark jobs.
- Understanding of filtering data using Spark DataFrame operations.
- Knowledge of working with structured data in Spark.
- Experience in printing statements and debugging Spark jobs.
- Familiarity with writing data to S3 using Spark.

Sol)

```
from pyspark.sql import SparkSession

S3_DATA_INPUT_PATH="s3://p2-projectpro-emr-athena/source-folder/wikiticker-
2015-09-12-sampled.json"
S3_DATA_OUTPUT_PATH_AGGREGATED="s3://p2-projectpro-emr-athena/data-
output/filtered_hw"
```

```
def main():
    spark = SparkSession.builder.appName('projectProDemo').getOrCreate()
    df = spark.read.json(S3_DATA_INPUT_PATH)
    print(f'The total number of records in the input data set is
{df.count()}')
    filtered_df = df.filter((df.delta >= 20) & (df.cityName == 'London'))
    print(f'The total number of records in the aggregated data set is
{filtered_df.count()}')
    filtered_df.show(10)
    filtered_df.printSchema()

filtered_df.write.mode('overwrite').parquet(S3_DATA_OUTPUT_PATH_AGGREGATED)
    print('The filtered data has been uploaded successfully')


if __name__ == '__main__':
    main()
```