# Fives CortX AWS SparkNotebook / Processing

# Code Notebook

**By :** MOURABIT El Bachir

### This notebook will be mainly used for the project

### This jupyther have been created by EMR cluster in EC2 to Clean, Transform & Consolidate Data that exist in S3

# 1. Create and Explore Dataset

## 1.1- Imports and Initialization

Before we get the data and start exploring it, let's download all the dependencies that we will need.

In [2]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

```
VBox()
Starting Spark application
```

| ID | YARN Application ID | Kind | State | Spark UI | Driver log | Current session? |
|---|---|---|---|---|---|---|
| 4 | application_1643811775736_0005 | pyspark | idle | Link | Link | ✔ |

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',
layout=Layout(height='25px', width='50%'),…
SparkSession available as 'spark'.
FloatProgress(value=0.0, bar_style='info', description='Progress:',
layout=Layout(height='25px', width='50%'),…
```

**Initialize Spark**

spark = SparkSession.builder.appName("Data consolidation").getOrCreate()

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',
layout=Layout(height='25px', width='50%'),…
```

**Initialize sources (input & output)**

In [3]:
```
bucket_path            = 's3a://recrutementdatasciencestorage'
vibration_AB_source    = f'{bucket_path}/vibration_axis_A_axis_B.orc'
vibration_C_source     = f'{bucket_path}/vibration_axis_C.orc'
vibration_D_source     = f'{bucket_path}/vibration_axis_D.orc'
milling_source         = f'{bucket_path}/milling_modes.orc'
output_uri             = f'{bucket_path}/consolidation.orc'
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',
layout=Layout(height='25px', width='50%'),…
```

**Verify if date format is correct / we will use it to verify date columb in every dataframe**

In [4]:
```
import datetime
from pyspark.sql.functions import udf
@udf("boolean")
def isvaliddatetime(date_text, format_date):
    try:
        datetime.datetime.strptime(date_text, format_date)
        return True
    except ValueError:
        return False

format_datetime         = '%Y-%m-%d %H:%M:%S'
format_datetime_decimal = '%Y-%m-%d %H:%M:%S.%f'
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',
layout=Layout(height='25px', width='50%'),…
```

**Test if isvaliddatetime works correctly**

In [5]:
```
isvaliddatetime(col('2018-10-20 03:46:40.000050'), format_datetime)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
Column<b'isvaliddatetime(2018-10-20 03:46:40.000050, %Y-%m-%d %H:%M:%S)'>
```

```
Column<b'isvaliddatetime(2018-10-20 03:46:40.000050, %Y-%m-%d %H:%M:%S)'>
```

## 1.2- Exploring Vibration A & B axis

In [6]:
```
vibration_A_B_df = spark \
                    .read \
                    .option("header", "true") \
                    .orc(vibration_AB_source) \
                    .toDF('date_AB', 'value_A', 'value_B') \
                    .orderBy(asc('date_AB'))
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [7]:
```
vibration_A_B_df_filetered = vibration_A_B_df \
                                .filter(isvaliddatetime(col('date_AB'), lit(format_d
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [8]:
```
vibration_A_B_df_without_null = vibration_A_B_df_filetered \
                                    .na \
                                    .drop()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [9]:
```
vibration_A_B_df_without_null.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+-------------------+-------------------+
|            date_AB|            value_A|            value_B|
+-------------------+-------------------+-------------------+
|2018-10-20 03:46:...|  0.4997962601945021| -0.4232508385579977|
|2018-10-20 03:46:...| -0.3049049175939375|-0.45048938736365657|
|2018-10-20 03:46:...| 0.13483806206868135|-0.22773576150343397|
|2018-10-20 03:46:...|  1.1917179155591455|-0.47666999906759433|
|2018-10-20 03:46:...|-0.05093001961112614|-0.09420548170563174|
+-------------------+-------------------+-------------------+
only showing top 5 rows

104519498
```

+-------------------+-------------------+-------------------+
|            date_AB|            value_A|            value_B|
+-------------------+-------------------+-------------------+
|2018-10-20 03:46:...|  0.4997962601945021| -0.4232508385579977|
|2018-10-20 03:46:...| -0.3049049175939375|-0.45048938736365657|
|2018-10-20 03:46:...| 0.13483806206868135|-0.22773576150343397|
|2018-10-20 03:46:...|  1.1917179155591455|-0.47666999906759433|
|2018-10-20 03:46:...|-0.05093001961112614|-0.09420548170563174|
+-------------------+-------------------+-------------------+
only showing top 5 rows

## 1.3- Exploring Vibration C axis

In [10]:
```
vibration_C_df = spark \
                    .read \
                    .option("header", "true") \
                    .orc(vibration_C_source) \
                    .toDF('date_C', 'value_C') \
                    .orderBy(asc('date_C'))
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [11]:
```
vibration_C_df_filetered = vibration_C_df \
                                .filter(isvaliddatetime(col('date_C'), lit(format_da
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [12]:
```
from pyspark import StorageLevel
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [13]:

```
vibration_C_df_without_null = vibration_C_df_filetered \
                                    .na \
                                    .drop()
vibration_C_df_without_null.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[date_C: string, value_C: string]
```

In [14]:

```
vibration_C_df_without_null.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+------------------+------------------+
|            date_C|           value_C|
+------------------+------------------+
|2018-10-20 03:46:40| 0.5169295745127163|
|2018-10-20 03:47:40|-0.2521033382635548|
|2018-10-20 03:48:40| 0.7167329219560264|
|2018-10-20 03:49:40| 0.5825232860070251|
|2018-10-20 03:50:40|0.27491402631724193|
+------------------+------------------+
only showing top 5 rows
```

Spark Job Progress

```
+------------------+------------------+
|            date_C|           value_C|
+------------------+------------------+
|2018-10-20 03:46:40| 0.5169295745127163|
|2018-10-20 03:47:40|-0.2521033382635548|
|2018-10-20 03:48:40| 0.7167329219560264|
|2018-10-20 03:49:40| 0.5825232860070251|
|2018-10-20 03:50:40|0.27491402631724193|
+------------------+------------------+
only showing top 5 rows
```

## 1.4- Exploring Vibration D axis

In [15]:

```
vibration_D_df = spark \
                    .read \
                    .option("header", "true") \
                    .orc(vibration_D_source) \
                    .toDF('date_D', 'value_D') \
                    .orderBy(asc('date_D'))
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [16]:

```
vibration_D_df_filetered = vibration_D_df \
                                .filter(isvaliddatetime(col('date_D'), lit(format_da
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [17]:

```
vibration_D_df_without_null = vibration_D_df_filetered \
```

```
                                        .na \
                                        .drop()
vibration_D_df_without_null.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[date_D: string, value_D: string]
```

In [18]:

```
vibration_D_df_without_null.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+--------------------+
|             date_D|             value_D|
+-------------------+--------------------+
|2018-10-20 03:46:40|0.001874454035457...|
|2018-10-20 03:46:41|  -11.67241608536065|
|2018-10-20 03:46:42| -0.9856337431497432|
|2018-10-20 03:46:43| -1.2268499578642467|
|2018-10-20 03:46:44|   18.89443445842324|
+-------------------+--------------------+
only showing top 5 rows
```

```
+-------------------+--------------------+
|             date_D|             value_D|
+-------------------+--------------------+
|2018-10-20 03:46:40|0.001874454035457...|
|2018-10-20 03:46:41|  -11.67241608536065|
|2018-10-20 03:46:42| -0.9856337431497432|
|2018-10-20 03:46:43| -1.2268499578642467|
|2018-10-20 03:46:44|   18.89443445842324|
+-------------------+--------------------+
only showing top 5 rows
```

## 1.5- Exploring Milling mode data

In [28]:

```
milling_df = spark \
             .read \
             .option("header", "true") \
             .orc(milling_source) \
             .toDF('debut_prog_mode', 'programme', 'mode') \
             .orderBy(asc('debut_prog_mode'))
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [29]:

```
milling_df_filetered = milling_df \
                                .filter(isvaliddatetime(col('debut_prog_mode'), lit(
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

In [30]:

```
milling_df_without_null = milling_df_filetered \
                                        .na \
```

```
                                                .drop()
milling_df_without_null.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[debut_prog_mode: string, programme: string, mode: string]
```

In [31]:
```
milling_df_without_null.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+---------+-----+
|    debut_prog_mode|programme| mode|
+-------------------+---------+-----+
|2018-10-20 03:53:25|     prg3|mode2|
|2018-10-20 04:02:52|     prg3|mode5|
|2018-10-20 04:03:07|     prg1|mode4|
|2018-10-20 04:18:37|     prg1|mode1|
|2018-10-20 04:25:07|     prg1|mode1|
+-------------------+---------+-----+
only showing top 5 rows
```

```
+-------------------+---------+-----+
|    debut_prog_mode|programme| mode|
+-------------------+---------+-----+
|2018-10-20 03:53:25|     prg3|mode2|
|2018-10-20 04:02:52|     prg3|mode5|
|2018-10-20 04:03:07|     prg1|mode4|
|2018-10-20 04:18:37|     prg1|mode1|
|2018-10-20 04:25:07|     prg1|mode1|
+-------------------+---------+-----+
only showing top 5 rows
```

# 2- Consolidating DATA

## 2.1- Consolidating Axis C dataframe & Milling modes

In [32]:
```
#Join small dataframes / milling mode & vibration_c
vibration_C_milling_df = vibration_C_df_without_null. \
                         join(milling_df_without_null, vibration_C_df_without_null.dat
vibration_C_milling_df.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[date_C: string, value_C: string, debut_prog_mode: string, programme: strin
g, mode: string]
```

In [33]:
```
vibration_C_milling_df.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+-------------------+-------------------+---------+-----+
|             date_C|            value_C|    debut_prog_mode|programme| mode|
+-------------------+-------------------+-------------------+---------+-----+
```

```
|2018-10-20 14:38:40| 1.3677986240703166|2018-10-20 14:38:40|     prg3|mode1|
|2018-10-21 09:38:40|-0.5591151400853271|2018-10-21 09:38:40|     prg1|mode2|
|2018-10-21 10:23:40|  0.834322603398805|2018-10-21 10:23:40|     prg3|mode1|
|2018-10-21 18:02:40| 1.6925290547182141|2018-10-21 18:02:40|     prg1|mode5|
|2018-10-21 19:55:40|  1.534462973971987|2018-10-21 19:55:40|     prg1|mode2|
+-------------------+-------------------+-------------------+---------+-----+
only showing top 5 rows
```

```
+-------------------+-------------------+-------------------+---------
+-----+
|              date_C|            value_C|
debut_prog_mode|programme| mode|
+-------------------+-------------------+-------------------+---------
+-----+
|2018-10-20 14:38:40| 1.3677986240703166|2018-10-20 14:38:40|
prg3|mode1|
|2018-10-21 09:38:40|-0.5591151400853271|2018-10-21 09:38:40|
prg1|mode2|
|2018-10-21 10:23:40|  0.834322603398805|2018-10-21 10:23:40|
prg3|mode1|
|2018-10-21 18:02:40| 1.6925290547182141|2018-10-21 18:02:40|
prg1|mode5|
|2018-10-21 19:55:40|  1.534462973971987|2018-10-21 19:55:40|
prg1|mode2|
+-------------------+-------------------+-------------------+---------
+-----+
only showing top 5 rows
```

## 2.2- Consolidating Axis C dataframe+Milling modes & Axis C dataframe

In [34]:

```
#Join (milling mode, vibration_c) & vibration_d
vibration_C_D_milling_df = vibration_D_df_without_null. \
                       join(vibration_C_milling_df, vibration_D_df_without_null.date
vibration_C_D_milling_df.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[date_D: string, value_D: string, date_C: string, value_C: string, debut_pr
og_mode: string, programme: string, mode: string]
```

In [35]:

```
vibration_C_D_milling_df.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+------------------+-------------------+-------------------+----
--------------+---------+-----+
|             date_D|           value_D|             date_C|            value_C|
debut_prog_mode|programme| mode|
+-------------------+------------------+-------------------+-------------------+----
--------------+---------+-----+
|2018-10-20 14:38:40|-16.91526668832627|2018-10-20 14:38:40| 1.3677986240703166|2018
-10-20 14:38:40|     prg3|mode1|
|2018-10-21 09:38:40|  3.40823511379328|2018-10-21 09:38:40|-0.5591151400853271|2018
-10-21 09:38:40|     prg1|mode2|
|2018-10-21 10:23:40|12.474151699941231|2018-10-21 10:23:40|  0.834322603398805|2018
-10-21 10:23:40|     prg3|mode1|
```

```
|2018-10-21 18:02:40|-39.48241113266405|2018-10-21 18:02:40| 1.6925290547182141|2018
-10-21 18:02:40|     prg1|mode5|
|2018-10-21 19:55:40|-21.89039705668902|2018-10-21 19:55:40|  1.534462973971987|2018
-10-21 19:55:40|     prg1|mode2|
+-------------------+------------------+-------------------+-------------------+----
--------------+---------+-----+
only showing top 5 rows
```

```
+-------------------+-------------------+-------------------+---------
----------+-------------------+---------+-----+
|             date_D|            value_D|             date_C|
value_C|    debut_prog_mode|programme| mode|
+-------------------+-------------------+-------------------+---------
----------+-------------------+---------+-----+
|2018-11-19 15:16:40|-10.205652970535986|2018-11-19
15:16:40|-0.4002595557791937|2018-11-19 15:16:40|     prg3|mode3|
|2019-01-04 21:20:40|  10.47515259186287|2019-01-04 21:20:40|
-0.867314128762798|2019-01-04 21:20:40|     prg1|mode1|
|2019-02-20 03:55:40|  6.749618961061843|2019-02-20
03:55:40|-0.5475838200922781|2019-02-20 03:55:40|     prg1|mode3|
|2019-08-27 12:17:40| -10.55268783115762|2019-08-27 12:17:40|
-0.0512622876769|2019-08-27 12:17:40|     prg3|mode4|
|2019-10-01 03:39:40|-39.105828257085676|2019-10-01 03:39:40|
1.0591544446264256|2019-10-01 03:39:40|     prg3|mode1|
+-------------------+-------------------+-------------------+---------
----------+-------------------+---------+-----+
only showing top 5 rows
```

## 2.2- Consolidating Axis C dataframe+Milling modes+Axis C dataframe & Axis A&B

In [36]:

```
#Join (milling mode, vibration_c, vibration_d ) & vibration_a_b
vibration_A_B_C_D_milling_df = vibration_A_B_df_without_null. \
                       join(vibration_C_D_milling_df, vibration_A_B_df_without_null.
vibration_A_B_C_D_milling_df.persist(StorageLevel.MEMORY_AND_DISK)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
DataFrame[date_AB: string, value_A: string, value_B: string, date_D: string, value_
D: string, date_C: string, value_C: string, debut_prog_mode: string, programme: stri
ng, mode: string]
```

In [37]:

```
vibration_A_B_C_D_milling_df.show(5)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
+-------------------+-------------------+-------------------+-------------------+
------------------+-------------------+-----------------+-------------------+------
---+-----+
|            date_AB|            value_A|            value_B|             date_D|
value_D|             date_C|          value_C|    debut_prog_mode|programme| mode|
+-------------------+-------------------+-------------------+-------------------+
------------------+-------------------+-----------------+-------------------+------
---+-----+
|2018-12-29 16:06:...| 0.13627758136417167| -0.4648627425392277|2018-12-29 16:06:40|
-9.613326548667038|2018-12-29 16:06:40|1.0556886269138857|2018-12-29 16:06:40|     p
rg3|mode3|
```

```
|2018-12-29 16:06:...|-0.21216823394973788| -0.3896096143305258|2018-12-29 16:06:40|
-9.613326548667038|2018-12-29 16:06:40|1.0556886269138857|2018-12-29 16:06:40|     p
rg3|mode3|
|2018-12-29 16:06:...|  0.6511266540000997|-0.09225297692872193|2018-12-29 16:06:40|
-9.613326548667038|2018-12-29 16:06:40|1.0556886269138857|2018-12-29 16:06:40|     p
rg3|mode3|
|2018-12-29 16:06:...| 0.04519463605601759|-0.31353998453994975|2018-12-29 16:06:40|
-9.613326548667038|2018-12-29 16:06:40|1.0556886269138857|2018-12-29 16:06:40|     p
rg3|mode3|
|2018-12-29 16:06:...|  0.9739615081069366| -0.4732913761896134|2018-12-29 16:06:40|
-9.613326548667038|2018-12-29 16:06:40|1.0556886269138857|2018-12-29 16:06:40|     p
rg3|mode3|
+-------------------+-------------------+-------------------+-------------------+
------------------+-------------------+-----------------+-------------------+------
---+-----+
only showing top 5 rows
```

```
+-------------------+-------------------+-------------------+------
------------+------------------+-------------------+----------------
-+-------------------+---------+-----+
|             date_AB|            value_A|             value_B|
date_D|           value_D|             date_C|           value_C|
debut_prog_mode|programme| mode|
+-------------------+-------------------+-------------------+------
------------+------------------+-------------------+----------------
-+-------------------+---------+-----+
|2018-12-29 16:06:...| 0.13627758136417167| -0.4648627425392277|2018-
12-29 16:06:40|-9.613326548667038|2018-12-29
16:06:40|1.0556886269138857|2018-12-29 16:06:40|     prg3|mode3|
|2018-12-29 16:06:...|-0.21216823394973788| -0.3896096143305258|2018-
12-29 16:06:40|-9.613326548667038|2018-12-29
16:06:40|1.0556886269138857|2018-12-29 16:06:40|     prg3|mode3|
|2018-12-29 16:06:...|  0.6511266540000997|-0.09225297692872193|2018-
12-29 16:06:40|-9.613326548667038|2018-12-29
16:06:40|1.0556886269138857|2018-12-29 16:06:40|     prg3|mode3|
|2018-12-29 16:06:...| 0.04519463605601759|-0.31353998453994975|2018-
12-29 16:06:40|-9.613326548667038|2018-12-29
16:06:40|1.0556886269138857|2018-12-29 16:06:40|     prg3|mode3|
|2018-12-29 16:06:...|  0.9739615081069366| -0.4732913761896134|2018-
12-29 16:06:40|-9.613326548667038|2018-12-29
16:06:40|1.0556886269138857|2018-12-29 16:06:40|     prg3|mode3|
+-------------------+-------------------+-------------------+------
------------+------------------+-------------------+----------------
-+-------------------+---------+-----+
only showing top 5 rows
```

## 2.2.3- Save data as ORC

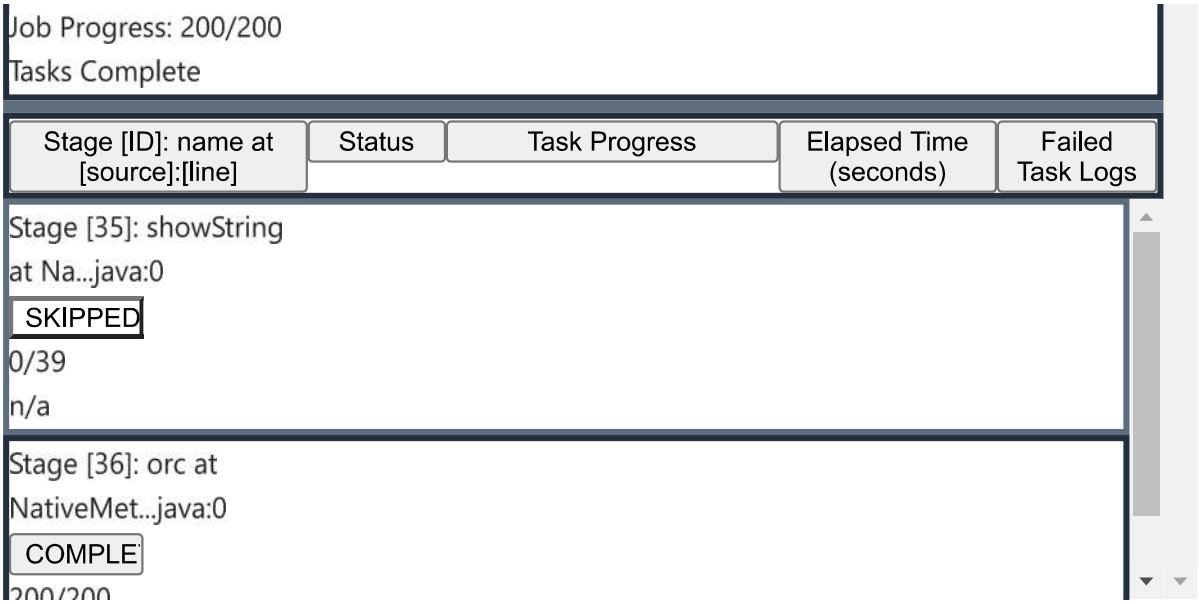
In [40]:

```
#Persist
vibration_A_B_C_D_milling_df.write.mode("overwrite").orc(output_uri)
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
```

Job [20]: orc at NativeMethodAccessorImpl.java:0

Progress for orc at NativeMethodAccessorImpl.java:0




In [43]:
```
print('Consolidation lines : ', vibration_A_B_C_D_milling_df.count())
```

```
VBox()
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(he
ight='25px', width='50%'),…
Consolidation lines :  39999
```