In [1]:

```
from pyspark.sql import SparkSession
```

In [2]:

```
spark = SparkSession.builder.appName('miss').getOrCreate()
```

In [3]:

```
df = spark.read.csv('D:/Python/PySpark/Python-and-Spark-for-Big-Data-master/Spark_DataFrames/ContainsNull.csv',inferSchema= True, header= True)
```

In [4]:

```
df.show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp1| John| null|
|emp2| null| null|
|emp3| null|345.0|
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [6]:

```
# 1. you can drop the rows having null's
df.na.drop().show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [8]:

```
## you can also set the threshold values, if null's in a row is greater than the threshold then drop that row
df.na.drop(thresh=2).show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp1| John| null|
|emp3| null|345.0|
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [9]:

```
## you can also drop only rows which contains all the the null's
df.na.drop(how= 'all').show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp1| John| null|
|emp2| null| null|
|emp3| null|345.0|
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [11]:

```
## you can also consider only the columns, if there is any null that column(s) then drop the row
df.na.drop(subset= ['Sales']).show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp3| null|345.0|
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [13]:

```
## fill the null values
df.na.fill('No Name', subset= ['Name']).show()
```

```
+----+-------+-----+
|  Id|   Name|Sales|
+----+-------+-----+
|emp1|   John| null|
|emp2|No Name| null|
|emp3|No Name|345.0|
|emp4|  Cindy|456.0|
+----+-------+-----+
```

In [20]:

```
## fill mean value in Sales columns
from pyspark.sql.functions import mean, avg
```

In [18]:

```
mean_val = df.select(mean(df['Sales'])).collect # or df.select(avg('Sales')).collect()
```

In [33]:

```
mean_sales = mean_val[0][0]
```

In [34]:

```
df.na.fill(mean_sales,subset=['Sales']).show()
```

```
+----+-----+-----+
|  Id| Name|Sales|
+----+-----+-----+
|emp1| John|400.5|
|emp2| null|400.5|
|emp3| null|345.0|
|emp4|Cindy|456.0|
+----+-----+-----+
```

In [ ]:

```
```