Discovering Frequent Patterns in Big Data Using ECLATDiffset Algorithm

In this tutorial, we will discuss two approaches to find frequent patterns in big data using ECLATDiffset algorithm.

1. **Basic approach:** Here, we present the steps to discover frequent patterns using a single minimum support value
2. **Advanced approach:** Here, we generalize the basic approach by presenting the steps to discover frequent patterns using multiple minimum support values.

Basic approach: Executing ECLATDiffset on a single dataset at a particular minimum support value

Step 1: Import the ECLATDiffset algorithm

In [1]:
```
from PAMI.frequentPattern.basic import ECLATDiffset  as alg
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'transactional_T10I4D100K.csv'

minimumSupportCount=10000  #Users can also specify this constraint between 0 to 1.

seperator='¥t'
```

Step 3: Execute the ECLATDiffset algorithm

In [ ]:
```
obj = alg.ECLATDiffset(iFile=inputFile, minSup=minimumSupportCount, sep=seperator)
obj.startMine()            #Start the mining process
```

Step 4: Storing the generated patterns

Step 4.1: Storing the generated patterns in a file

In [ ]:
```
obj.savePatterns(outFile='frequentPatternsMinSupCount100.txt')
```

Step 4.2. Storing the generated patterns in a data frame

In [ ]:
```
frequentPatternsDF= obj.getPatternsAsDataFrame()
```

Step 5: Getting the statistics

Step 5.1: Total number of discovered patterns

In [ ]:
```
print('Total No of patterns: ' + str(len(frequentPatternsDF)))
```

Step 5.2: Runtime consumed by the mining algorithm

In [ ]:
```
print('Runtime: ' + str(obj.getRuntime()))
```

In [ ]:
```
##### Step 5.3: Total Memory consumed by the mining algorithm
```

In [ ]:
```
print('Memory (RSS): ' + str(obj.getMemoryRSS()))
print('Memory (USS): ' + str(obj.getMemoryUSS()))
```