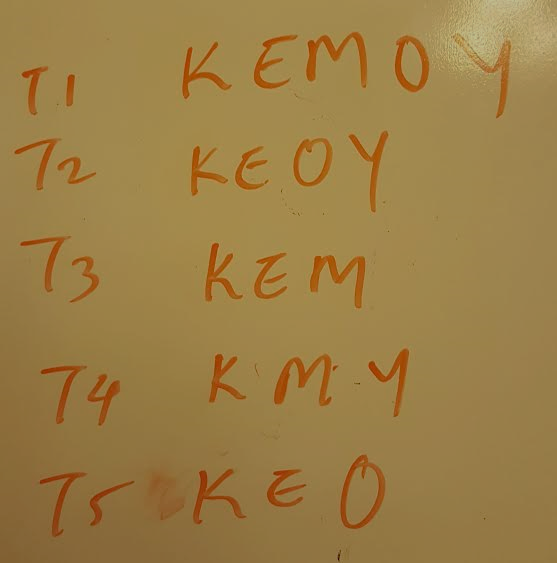


a)

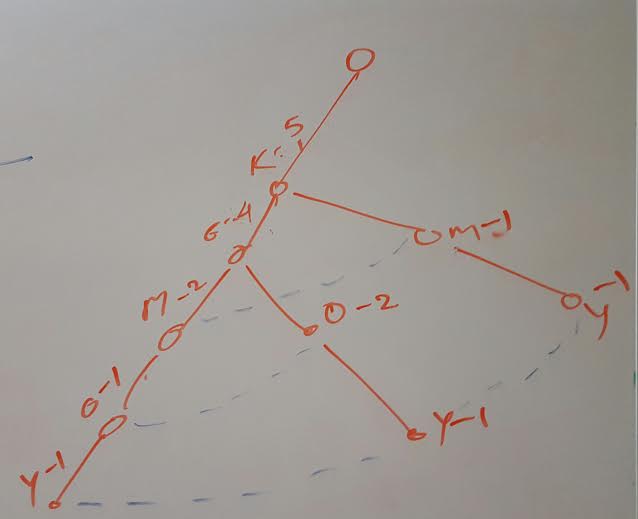
|  |  |
| --- | --- |
| **Itemset** | **Sup. Count** |
| K | 5 |
| E | 4 |
| O | 4 |
| M | 3 |
| Y | 3 |
| C | 2 |
| N | 2 |
| A | 1 |
| D | 1 |
| I | 1 |
| U | 1 |

Keeping only the frequent set of items :

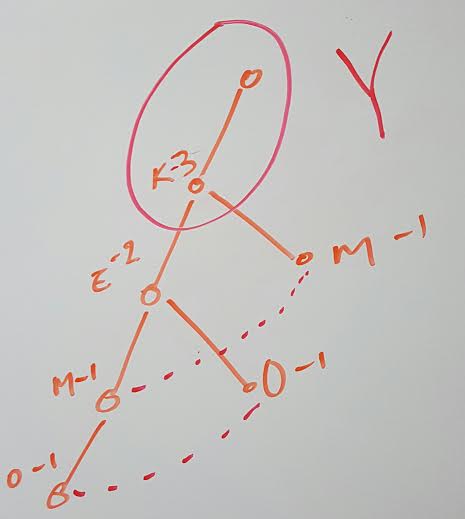
|  |  |
| --- | --- |
| **Itemset** | **Sup. Count** |
| K | 5 |
| E | 4 |
| O | 4 |
| M | 3 |
| Y | 3 |

Arranging the transactions in descending order and removing duplicate entries:  
Removed infrequent items:  
  


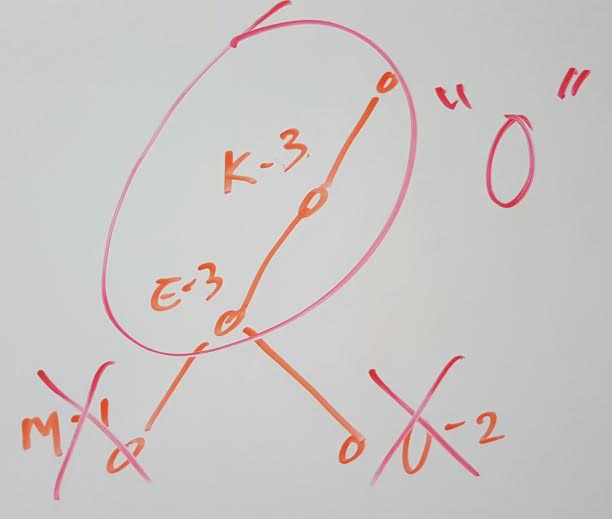
So the FP tree can be built as below from the above transactions as below:



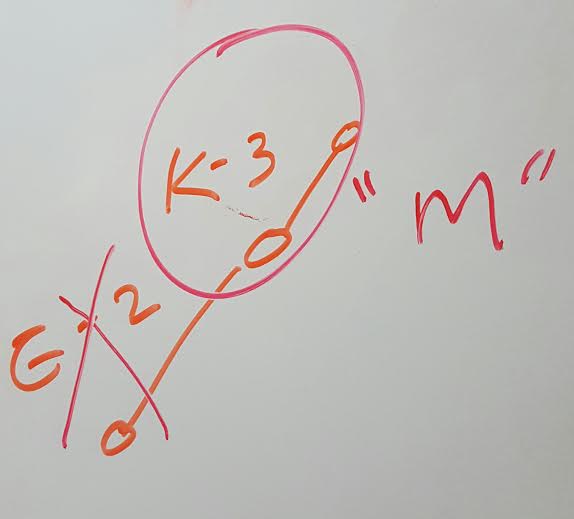
Conditional FP tree for Y is as below:



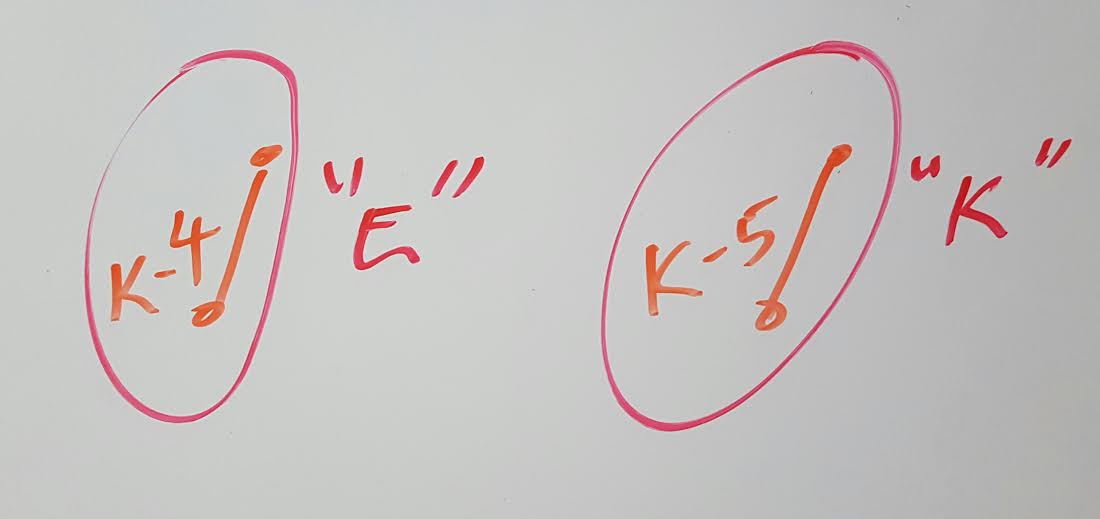
Conditional FP tree for O is as below:



Conditional FP tree for M is as below:



Conditional FP tree for E is as below:



|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Conditional Pattern Base** | **Conditional FP-Tree** | **Frequent Patterns Generated** |
| Y | {K,E,M,O:1}, {K,E,O:1}, {K, M:1} | {K:3} | {K, Y:3} |
| O | {K,E,M:1}, {K,E:2} | {E:3, K:3} | {E,O:3}, {K,O:3}, {E,K,O:3} |
| M | {K,E:2}, {K:1} | {K:3} | {K, M:3} |
| E | {K:4} | {K:4} | {K, E:4} |
| K | NULL | NULL | NULL |

The strong association rules are:

**{E, O} => K and   
{K, O} => E**

A study of the FP-growth method performance shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm.

FP Growth algorithm does only 2 passes over the data set.

It compresses the data-set

There is no candidate generation in FP growth algorithm compared to apriori algorithm.