Apriori Algorithm

Market Basket Analysis

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items.

Bread and Jam

Laptop and Bag


Bread and Butter
OFFER
50% OFF

Association Rule Mining

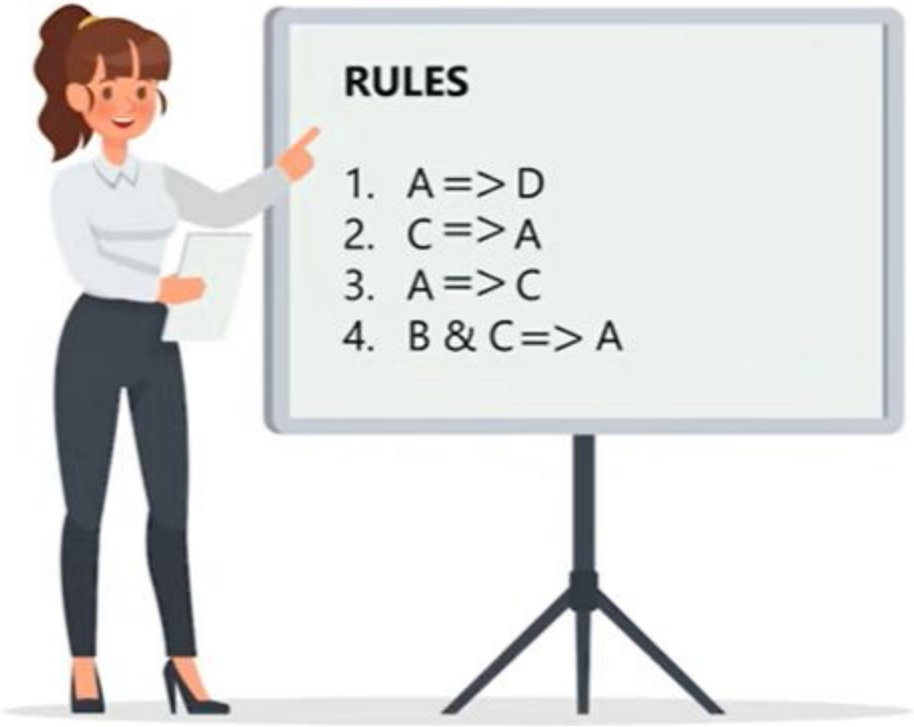
A ⇒ B

IF **THEN**

Measure Association

Transaction at a Local Market

T1	A	B	C
T2	A	C	D
T3	B	C	D
T4	A	D	E
T5	B	C	E


RULES
1. A => D
2. C => A
3. A => C
4. B & C => A

Apriori Algorithm

Apriori algorithm uses frequent item sets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.

Frequent Itemset is an itemset whose support value is greater than a threshold value.

Apriori Algorithm

Example

Apriori Algorithm

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Min. Support count = 2

Apriori Algorithm - 1st Iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Apriori Algorithm – 2nd Iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Only Items present in F1

C2

Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

C3 ?

Itemset	Support
{1,2,3}	
{1,2,5}	
{1,3,5}	
{2,3,5}	

Apriori Algorithm – Pruning

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

→

C3

Itemset	In F2?
{1,2,3}, {1,2}, {1,3}, {2,3}	NO
{1,2,5}, {1,2}, {1,5}, {2,5}	NO
{1,3,5},{1,5}, {1,3}, {3,5}	YES
{2,3,5}, {2,3}, {2,5}, {3,5}	YES

Apriori Algorithm – 4^{th} Iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

C3

Itemset	Support
{1,2,3,5}	1

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

For I = {1,3,5}, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}

For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}

- For every subsets S of I, output the rule:

S → (I-S) (S recommends I-S)

if ***support(I)/support(S) >= min_conf value***

Assume minimum confidence is 60%

Applying Rules to Item set F3

1. **{1,3,5}**

- ✓ Rule 1: **{1,3} → ({1,3,5} - {1,3})** means 1 & 3 → 5
  Confidence = support(1,3,5)/support(1,3) = 2/3 = 66.66% > 60%
  Rule 1 is selected

- ✓ Rule 2: **{1,5} → ({1,3,5} - {1,5})** means 1 & 5 → 3
  Confidence = support(1,3,5)/support(1,5) = 2/2 = 100% > 60%
  Rule 2 is selected

- ✓ Rule 3: **{3,5} → ({1,3,5} - {3,5})** means 3 & 5 → 1
  Confidence = support(1,3,5)/support(3,5) = 2/3 = 66.66% > 60%
  Rule 3 is selected

Applying Rules to Item set F3

1. {1,3,5}

✓ Rule 4: **{1} → ({1,3,5} - {1})** means 1 → 3 & 5
Confidence = support(1,3,5)/support(1) = 2/3 = 66.66% > 60%
Rule 4 is selected

✓ Rule 5: **{3} → ({1,3,5} - {3})** means 3 → 1 & 5
Confidence = support(1,3,5)/support(3) = 2/4 = 50% <60%
Rule 5 is rejected

✓ Rule 6: **{5} → ({1,3,5} - {5})** means 5 → 1 & 3
Confidence = support(1,3,5)/support(3) = 2/4 = 50% < 60%
Rule 6 is rejected

MEASURES OF PREDICTIVE ABILITY OF THE RULES

$$X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

1. *Support* refers to the percentage of baskets where the rule was true (both left and right side products were present).
   - ✓ *Frequency of items bought over all transactions*
2. Confidence measures what percentage of baskets that contained the left-hand product also contained the right.
   - ✓ *How often items X and Y occured together based on number of X occur(left item)*
   - ✓ *Support (X and Y) / Support (X)*
3. *Lift/Correlation* measures how much more frequently the left-hand item is found with the right than without the right.
   - ✓ *Confidence of X and Y over number of Y occur(right item)*
   - ✓ *Confidence(X and Y) / Support(Y)*

Association Rule Mining

$$A \Rightarrow B$$

$$Support = \frac{freq(A,B)}{N}$$

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

$$Lift = \frac{Support}{Supp(A)\ x\ Supp(B)}$$

EXAMPLE OF ASSOCIATION RULES

Assume there are 5 customers

3 of them bought milk, 2 bought potato chip and 2 bought both of them

Transaction 1:	Frozen pizza, cola, milk
Transaction 2:	Milk, potato chips
Transaction 3:	Cola, frozen pizza
Transaction 4:	Milk, potato chips
Transaction 5:	Cola, pretzels

milk ➔ potato chip

support milk = P(milk) = 3/5 = 0.6

support potato chip = P(potato chip) = 2/5 = 0.4

support = P(milk & potato chip) = 2/5 = 0.4

confidence

= support (milk & potato chip)/support(milk)

= 0.4/0.6

= 0.67

CONFIDENCE= P(Milk & potato chip) /P(Milk)

lift = confidence/support(potato chip) = 0.67/0.40 = 1.67

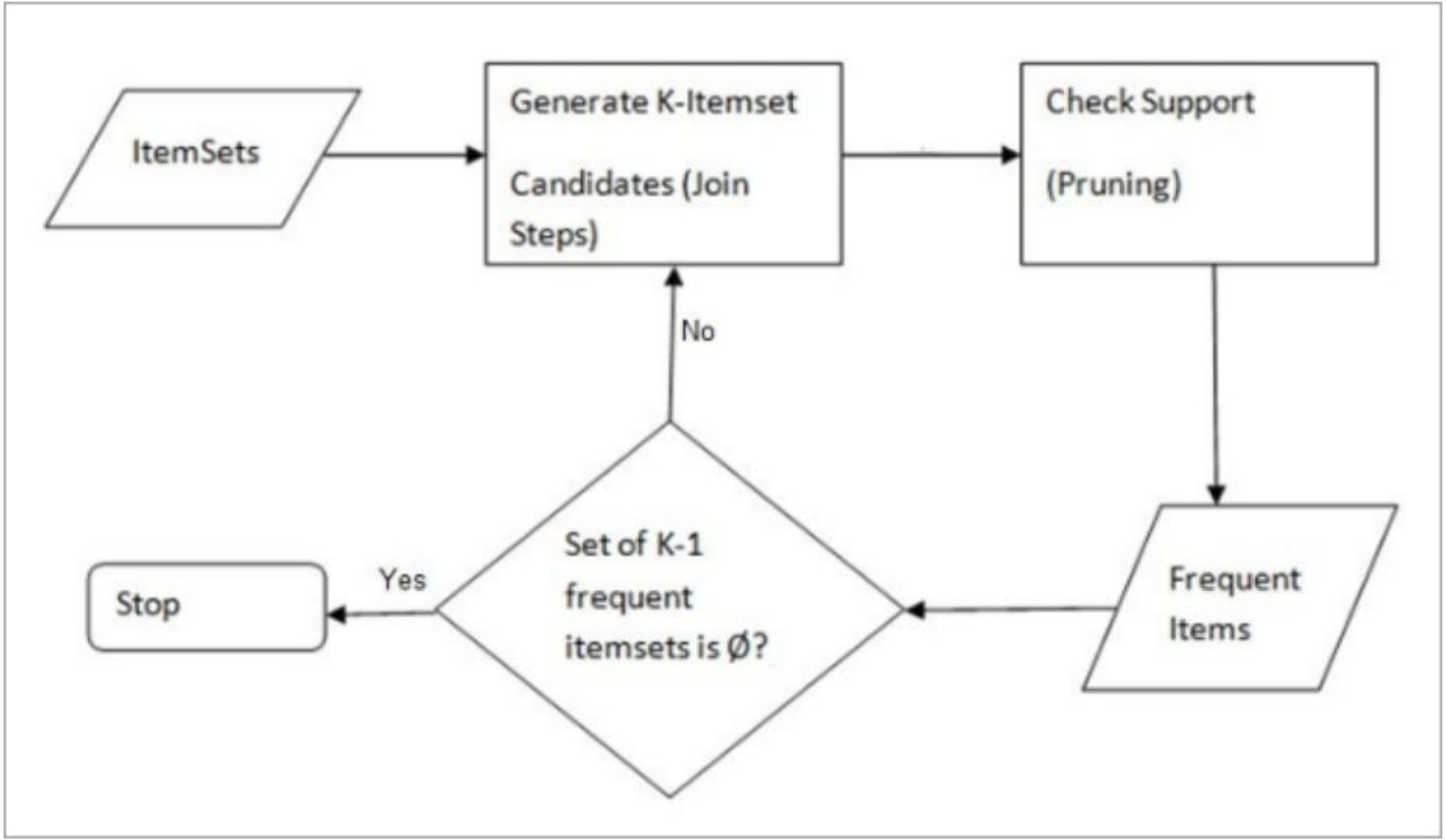
LIFT =

[P(Milk & Potato chip)/P(milk)] /P(Potato chip)

Any rule with a lift < 1 does not indicate a cross-selling opportunity


How about
Potato chip
→ Milk ?

Flowchart - Apriori Algorithm



Disadvantages

1. It requires high computation if the itemsets are very large and the minimum support is kept very low.

2. The entire database needs to be scanned.

Methods To Improve Apriori Efficiency

1. **Partitioning:** This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
2. **Sampling:** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.
3. **Hash-Based Technique:** This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.

Frequent Pattern Growth Algorithm

FP Growth Algorithm

Transaction ID	Items
T1	{E,K,M,N,O,Y}
T2	{D,E,K,N,O,Y}
T3	{A,E,K,M}
T4	{C,K,M,U,Y}
T5	{C,E,I,K,O,O}

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 3>}
K	

Consider minimum threshold =3

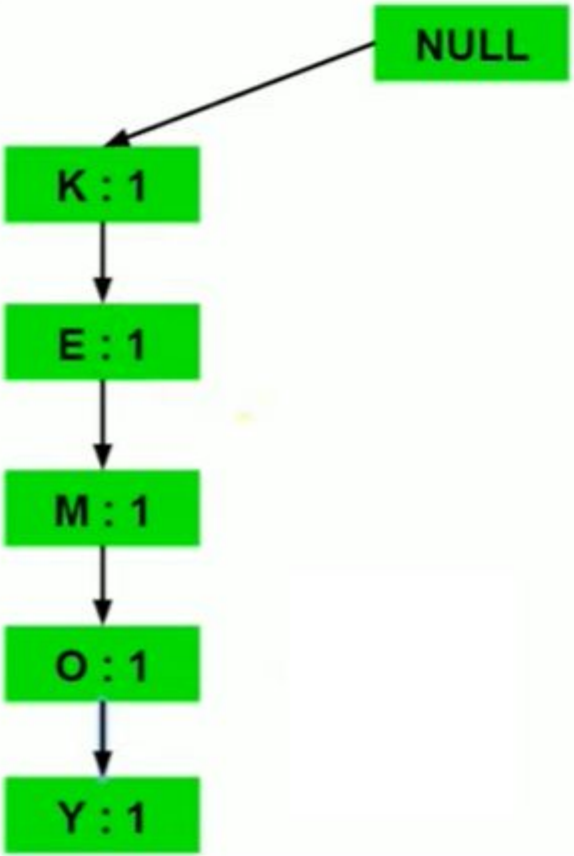
Frequent Pattern set L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

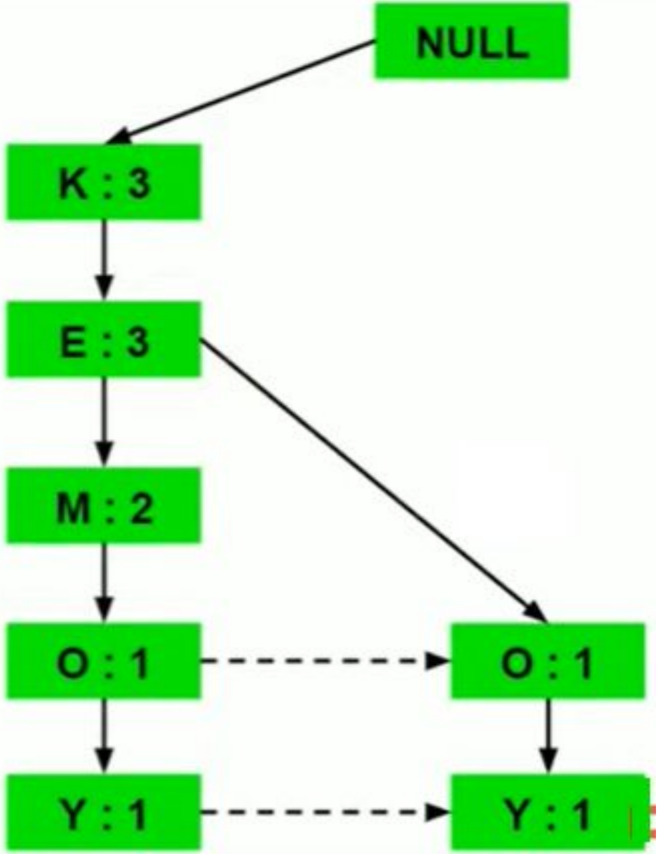
a) Inserting the set {K, E, M, O, Y}:



Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

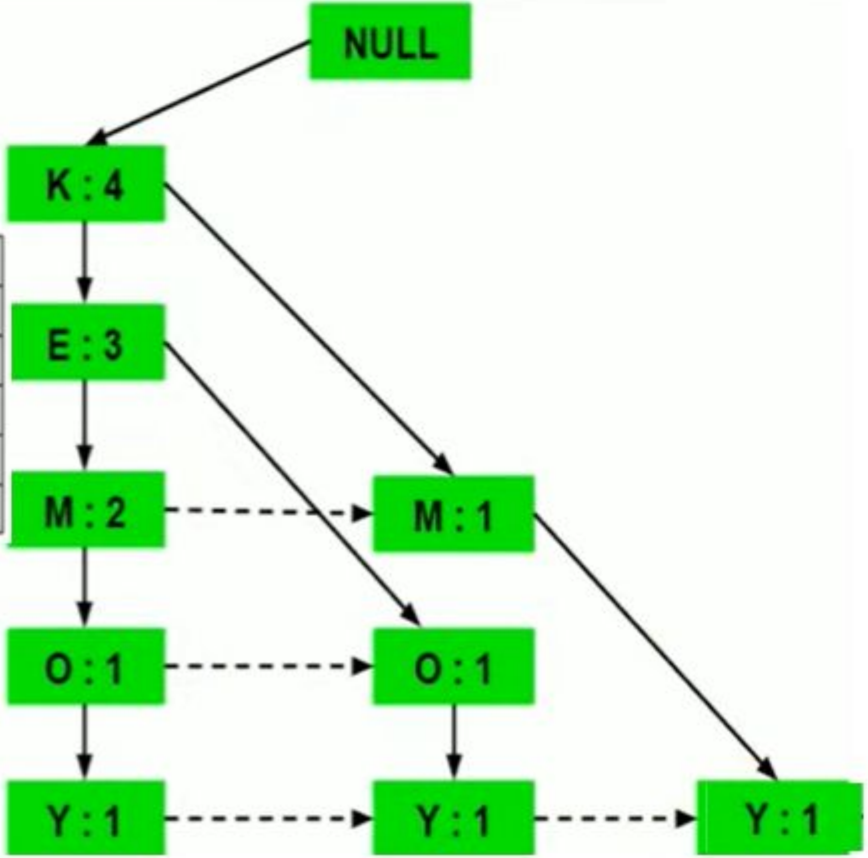
c) Inserting the set {K, E, M}:



Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

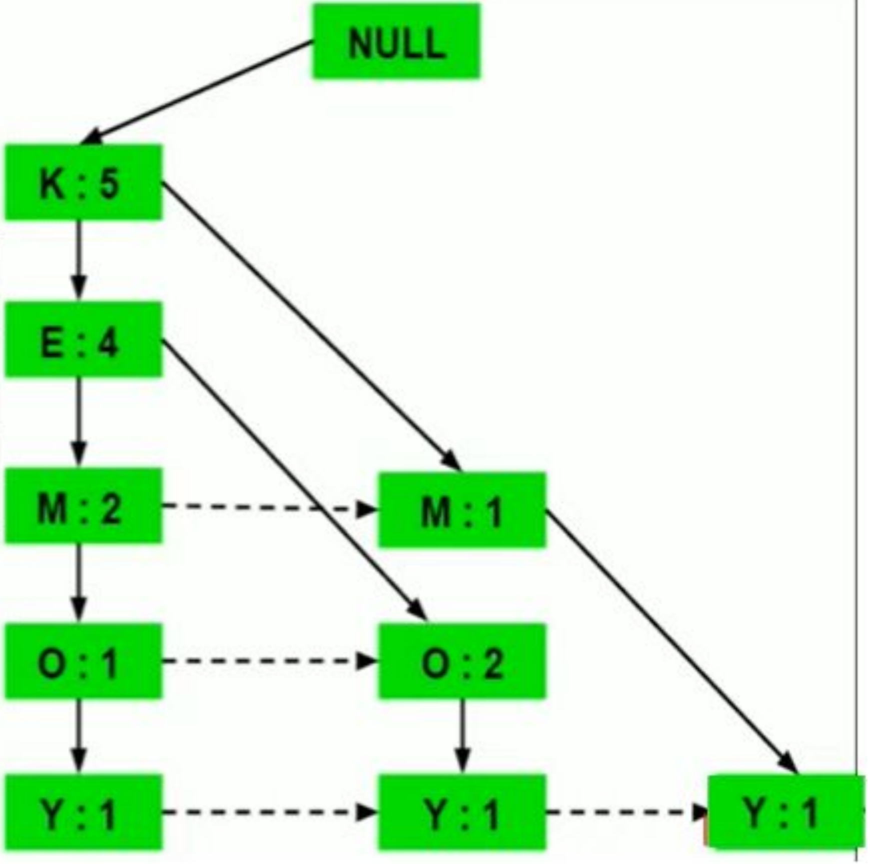
d) Inserting the set {K, M, Y}:



Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

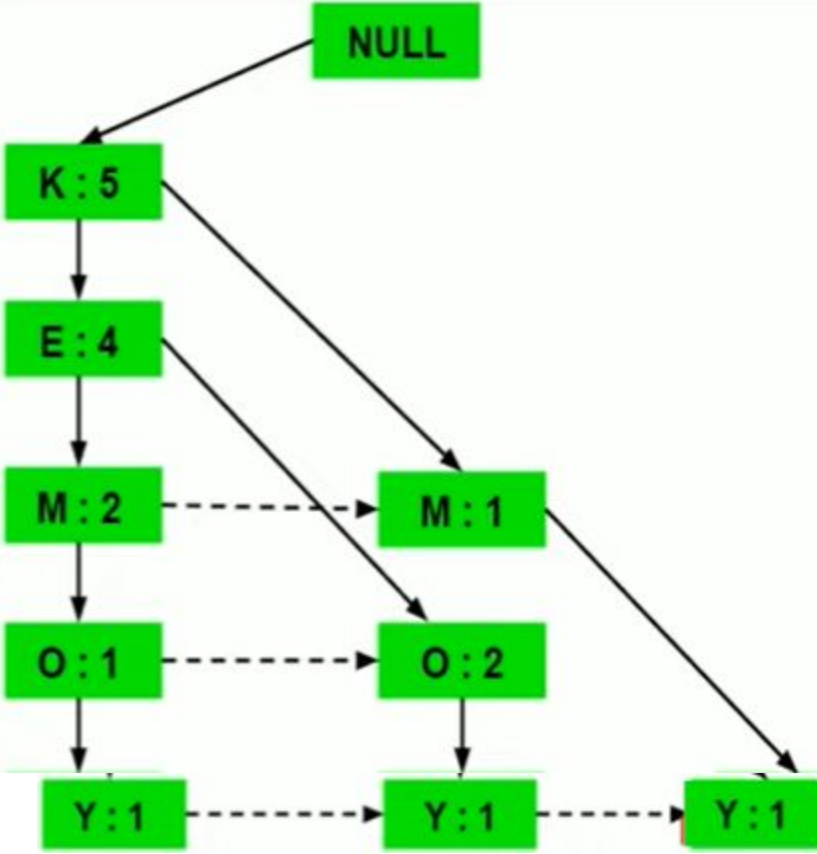
Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

e) Inserting the set {K, E, O}:



Now, for each item, the **Conditional Pattern Base** is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree.

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	



Now for each item the **Conditional Frequent Pattern Tree is built.** It is done by taking the set of elements which is common in all the paths in the Conditional Pattern Base of that item and calculating it's support count by summing the support counts of all the paths in the Conditional Pattern Base.

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K :4>}
K	