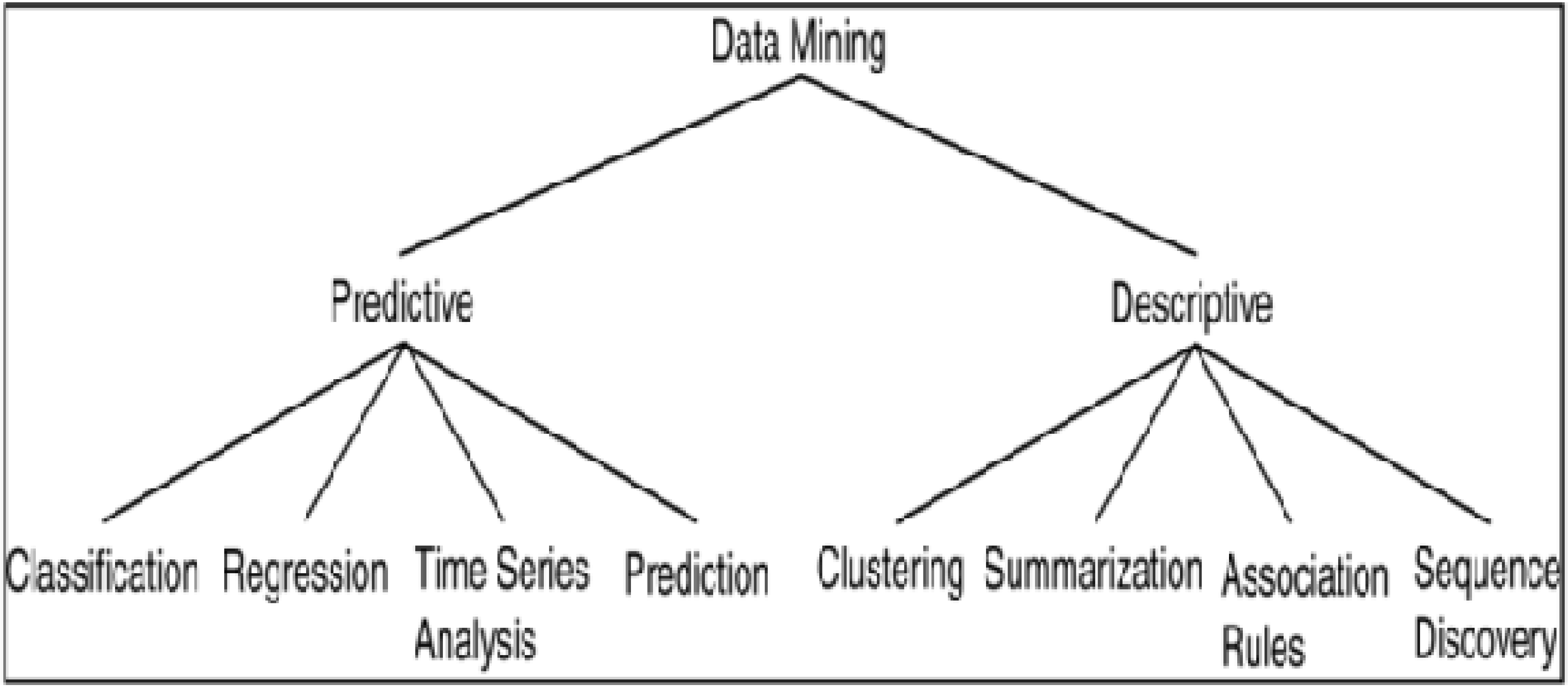

Data Mining
Predictive
Descriptive
Classification
Regression
Time Series
Analysis
Prediction
Clustering
Summarization
Association
Rules
Sequence
Discovery

Association Rule Mining

Market Basket Analysis

Apriori Algorithm

Market Basket Analysis

edureka!

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





LIMITED TIME
OFFER
50% OFF

Association Rule Mining

Association rules can be thought of as an IF-THEN relationship. Suppose item **A** is being bought by the customer, then the chances of item **B** being picked by the customer too under the same **Transaction ID** is found out.

There are two elements of these rules:

Antecedent (IF): This is an item/group of items that are typically found in the Itemsets or Datasets.

Consequent (THEN): This comes along as an item with an Antecedent/group of Antecedents.

D	
Transaction	Itemset
T1	A,B,C
T2	A,C
T3	A,D
T4	B,F,E

Support: It gives the fraction of transactions which contains item A and B. Basically Support tells us about the frequently bought items or the combination of items bought frequently.

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

So with this, we can **filter out** the items that have a **low frequency**.

Confidence: It tells us how often the items A and B occur together, given the number times A occurs.

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

Lift: Lift indicates the strength of a rule over the random occurrence of A and B. It basically tells us the strength of any rule.

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

Measures of Association

- Support
- Confidence
- Lift

$$Rule: \quad X \Rightarrow Y$$

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

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

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*

   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)*

Basket	Product 1	Product 2	Product 3
1	Milk	Cheese	
2	Milk	Apples	Cheese
3	Apples	Banana	
4	Milk	Cheese	
5	Apples	Banana	
6	Milk	Cheese	Banana
7	Milk	Cheese	
8	Cheese	Banana	
9	Cheese	Milk	

$$\textbf{Support} = \frac{(\mathbf{A}+\mathbf{B})}{\textbf{Total}}$$

$$\textbf{Support for Basket 1} = \frac{(\textbf{Milk}+\textbf{Cheese})}{\textbf{Total}} = \frac{6}{9} = .6666667$$

$$\textbf{Confidence} = \frac{(\mathbf{A}+\mathbf{B})}{\mathbf{A}}$$

$$\textbf{Confidence for Basket 1} = \frac{(\textbf{Milk}+\textbf{Cheese})}{\textbf{Milk}} = \frac{6}{6} = 1.000$$

D	
Transaction	Itemset
T1	A,B,C
T2	A,C
T3	A,D
T4	B,F,E

D	
Transaction	Itemset
T1	A,B,C
T2	A,C
T3	A,D
T4	B,F,E

C1	
Itemset	Support
{A}	3
{B}	2
{C}	2
{D}	1
{E}	1
{F}	1

L1	
Itemset	Support
{A}	3
{B}	2
{C}	2

C2	
Itemset	Support
{A,B}	1
{A,C}	2
{B,C}	1

L2	
Itemset	Support
{A,C}	2

Association Rule	Support	Confidence	Confidence %
A $\longrightarrow$ C	2	2/3 = 0.66	66%
C $\longrightarrow$ A	2	2/2 = 1	100%

APRIORI ALGORITHM EXAMPLE

Database D
Minsup = 0.5

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

← Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

←

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D →

L_3

itemset	sup
{2 3 5}	2

Database

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

C_1

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

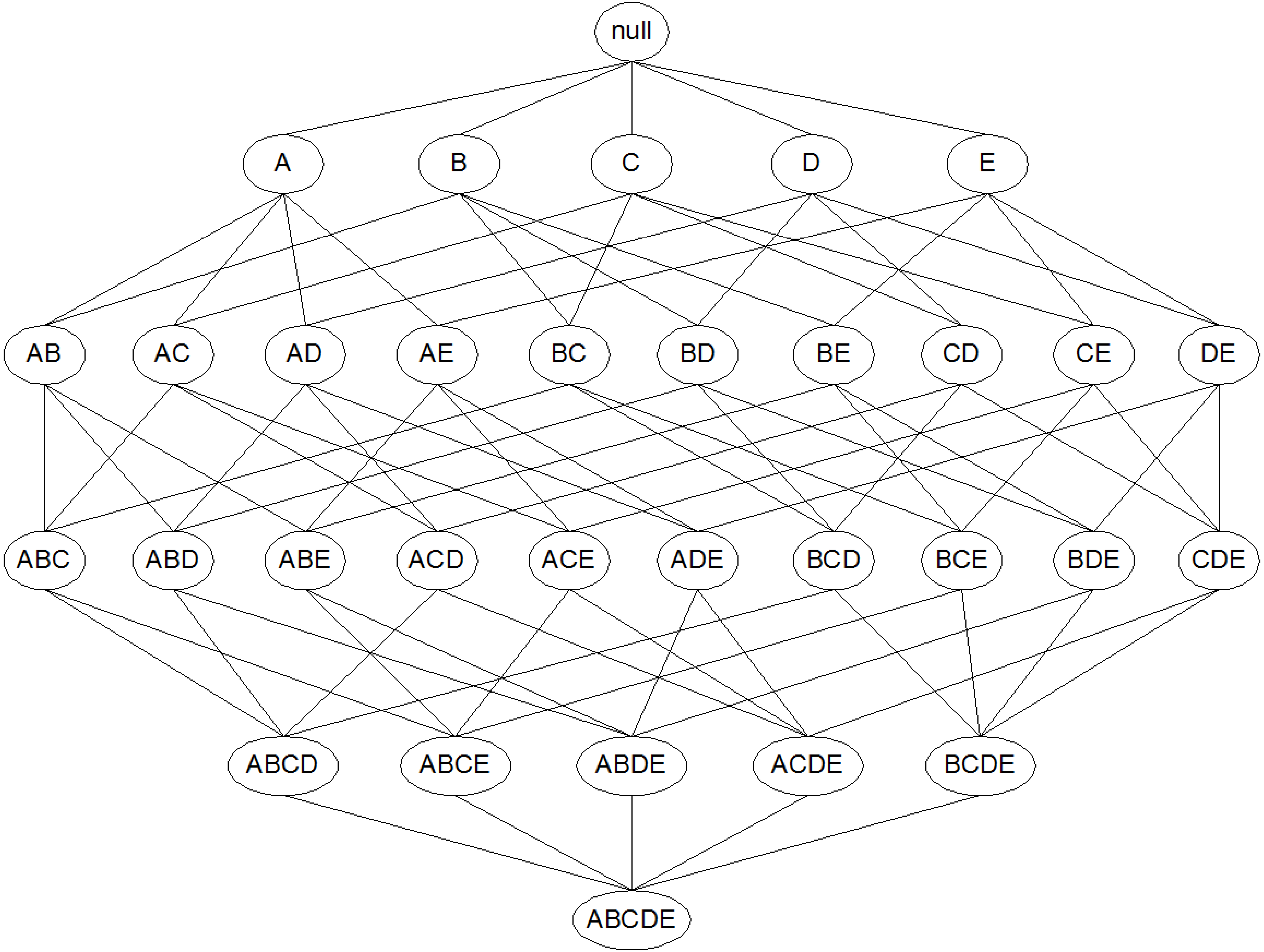
C_2

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

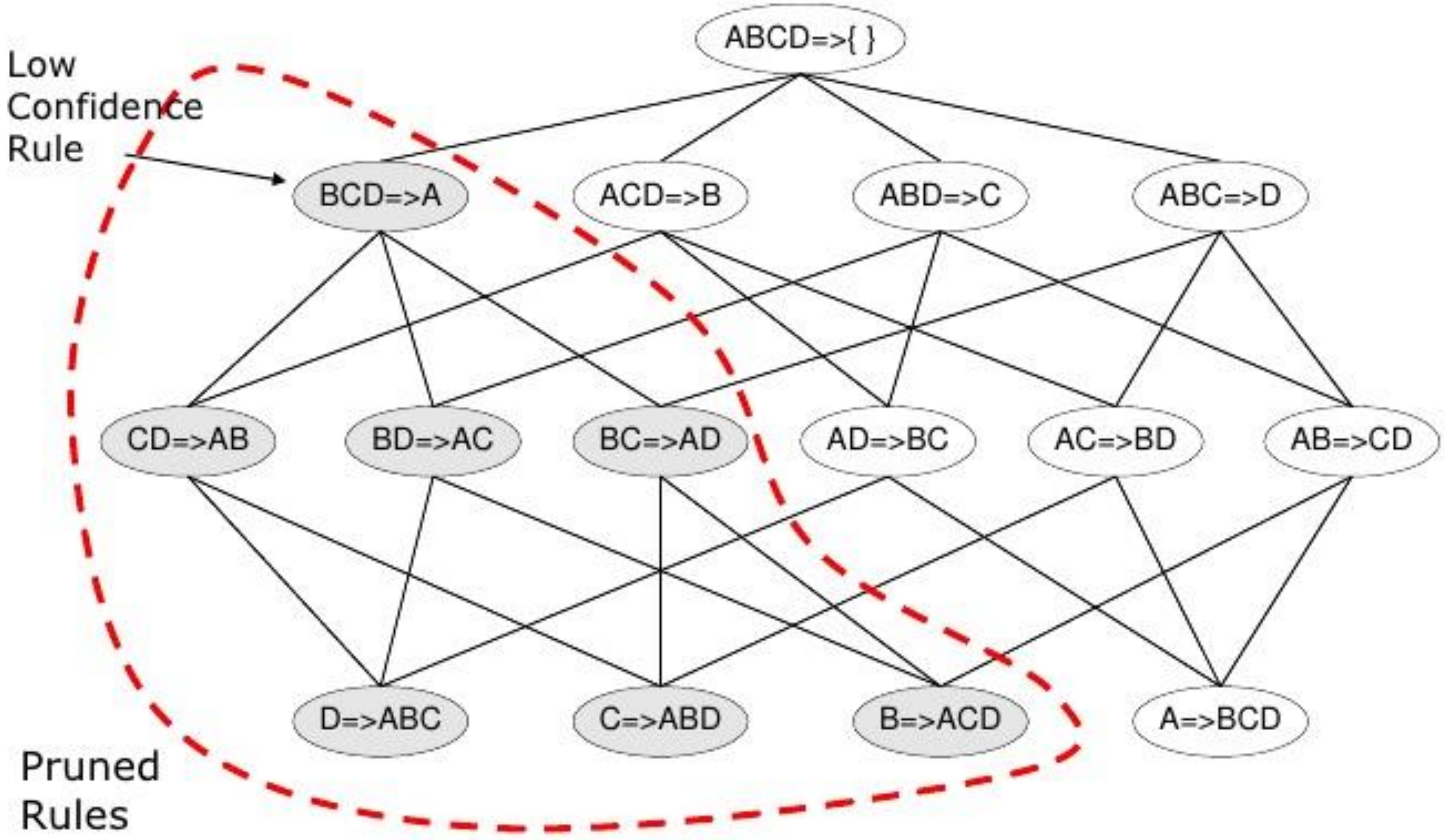
C_3

Itemset	Support
{2 3 5}*	2

Every subset of a frequent itemset is also frequent.


null
A
B
C
D
E
AB
AC
AD
AE
BC
BD
BE
CD
CE
DE
ABC
ABD
ABE
ACD
ACE
ADE
BCD
BCE
BDE
CDE
ABCD
ABCE
ABDE
ACDE
BCDE
ABCDE

Rule Generation for Apriori Algorithm



Example On finding Frequent Itemsets –

Consider the given dataset with given transactions.

transaction ID	items
1	{A,C,D}
2	{B,C,E}
3	{A,B,C,E}
4	{B,E}
5	{A,B,C,E}

- Lets say minimum support count is 3
- Relation hold is maximal frequent => closed => frequent

Ques.) For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider the Values as SUPPORT = 50% and CONFIDENCE = 75%.

Transaction ID.	Items Purchased
1	Bread, cheese, Egg, Juice
2	Bread, cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	cheese, Juice, Milk

Frequent Item Set

Items	Frequency	Support
Bread	4	4/5 = 80%
cheese	3	3/5 = 60%
$\boxed{\text{Egg}}$	1	1/5 = 20%
Juice	4	4/5 = 80%
Milk	3	3/5 = 60%
$\boxed{\text{Yogurt}}$	1	1/5 = 20%

Remove these ∵ there Support is less than 50%

Ques.) For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider the Values as SUPPORT = 50% and CONFIDENCE = 75%.

Transaction ID.	Items Purchased
1	Bread, cheese, Egg, Juice
2	Bread, cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	cheese, Juice, Milk

Ques.) For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider the Values as SUPPORT = 50% and CONFIDENCE = 75%.

Transaction ID.	Items Purchased
1	Bread, cheese, Egg, Juice
2	Bread, cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	cheese, Juice, Milk

Item Pairs	Frequency	Support
(Bread, cheese) →	2 →	2/5 = 40%
(Bread, Juice) →	3 →	3/5 = 60%
(Bread, Milk) →	2 →	2/5 = 40%
(cheese, Juice) →	3 →	3/5 = 60%
(cheese, Milk) →	1 →	1/5 = 20%
(Juice, Milk) →	2 →	2/5 = 40%

There Support ≥ 50%.

(ii) (Bread, Juice)

(Bread $\rightarrow$ Juice) (Juice $\rightarrow$ Bread)

upport

%.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

i.) $(\text{Bread} \rightarrow \text{juice}) = \frac{S(B \cup J)}{S(B)} = \frac{3.5}{5.4} = \frac{3}{4} = 75\%$

ii.) $(\text{Juice} \rightarrow \text{Bread}) = \frac{3.5}{5.4} = 75\%$

$\rightarrow (\text{cheese} \rightarrow \text{juice}) = \frac{3.5}{5.3} = 100\%$

$\rightarrow (\text{juice} \rightarrow \text{cheese}) = \frac{3}{5.}$

Transaction	item sets
I_1	A, B, C
I_2	A, C
I_3	A, D
I_4	B, E, F

minimum support : 50%
minimum confidence : 50%

$$\frac{5\cancel{0}}{10\cancel{0}} \times 4 = 2$$

C_1 =

items	Support
{A}	3
{B}	2
{C}	2
{D}	1
{E}	1
{F}	1

L_1 =

items	Support
{A}	3
{B}	2
{C}	2

$C_2 =$

	items	Support
×	{A, B}	1
×	{B, C}	1
	{A, C}	2

$L_2 =$

Items	Support
{A, C}	2

Transaction	Suppost
{A, C}	2

50 %

Transaction	Itemsets
I_1	A, B, C
I_2	A, C
I_3	A, D
I_4	B, E, F

A, C, D

$A \wedge C \rightarrow D$

$A \wedge D \rightarrow C$

Association rule	Suppost	Confidence	Confidence %
$A \rightarrow C$	2	$2/3 = 0.66$	66 %
$C \rightarrow A$	2	$2/2 = 1$	100 %

Final Rule :

$A \rightarrow C$

$C \rightarrow A$

Apriori Algorithm

Apriori algorithm uses frequent itemsets 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(support).

Let's say we have the following data of a store.

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

Apriori Algorithm

L_1 = {frequent 1-itemsets}; k = 2;

while ($L_{k-1} \neq \emptyset$) **do**

    C_k = candidate itemsets from L_{k-1}

    forall transactions $t \in$ DBASE **do**

        forall candidate itemsets $c \in t$ **do**

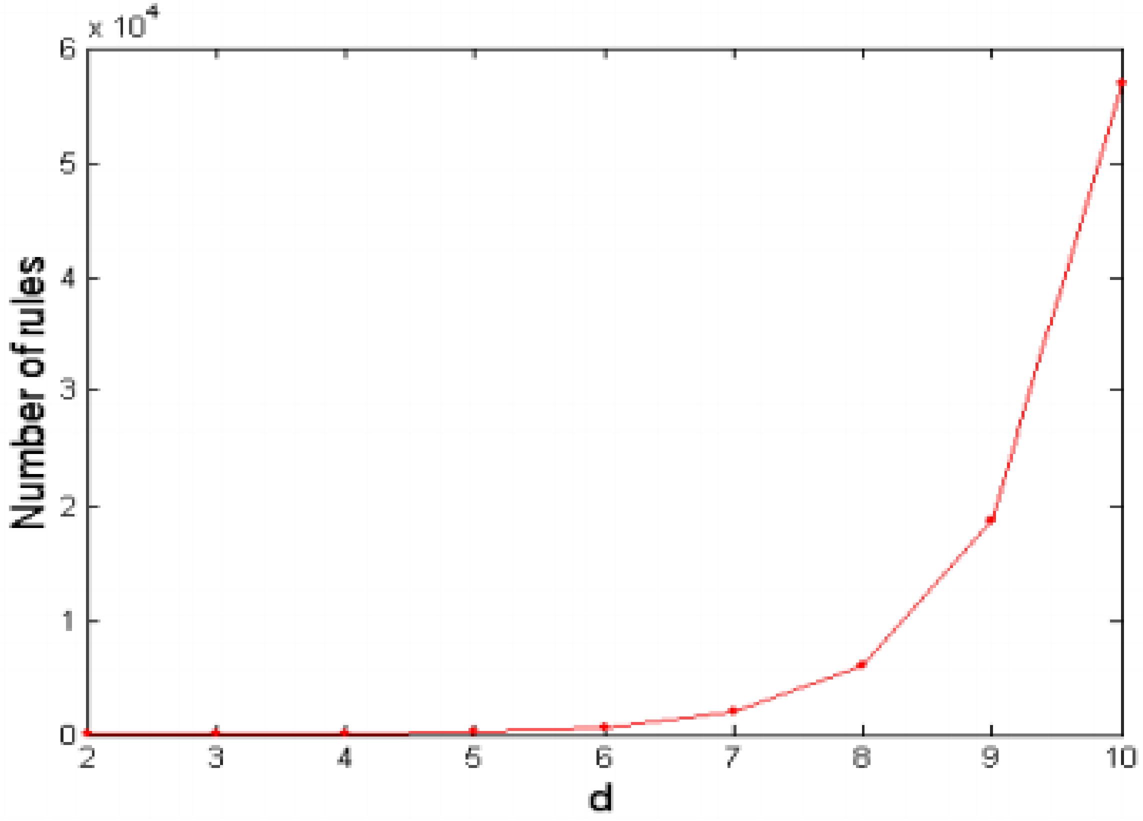
            count[c] = count[c] + 1

    L_k = All $c \in C_k$ with minimum support

    k++

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right] = 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Egg	3
Ketchup	3
Cookies	5

Frequent 1-item Sets	Frequency
Milk	9
Bread	10
Butter	10
Cookies	5

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Milk, Cookies	3
Bread, Butter	9
Butter, Cookies	3
Bread, Cookies	4

Frequent 2-item Sets	Frequency
Milk, Bread	7
Milk, Butter	7
Bread, Butter	9
Bread, Cookies	4

Transactions List

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

Milk, Bread, Butter, Cookies

3-item Sets	**Frequency**
Milk, Bread, Butter	6
Milk, Bread, Cookies	1
Bread, Butter, Cookies	3
Milk, Butter, Cookies	2

Frequent 3-item Sets	Frequency
Milk, Bread, Butter	6

Association Rule Mining - Subset Creation

- Frequent 3-Item Set = I => {Milk, Bread, Butter}

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}

- How to form Association Rule...?
  - For every non-empty subset S of I, the association rule is,
    - **S → (I-S)**
    - **If support(I) / support(S) >= min_confidence**

Association Rule Mining - Subset Creation

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 1: {Milk} → {Bread, Butter} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk) = $\frac{6/12}{9/12}$ = 6/9 = 66.67% > 60%
  - Valid

- Rule 2: {Bread} → {Milk, Butter} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread) = 6/10 = 60% >= 60%
  - Valid

Association Rule Mining - Subset Creation

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 3: {Butter} → {Milk, Bread} {S=50%, C=60%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Butter) = 6/10 = 60%>=60
  - Valid

- Rule 4: {Milk, Bread} → {Butter} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Bread) = 6/7 = 85.7% > 60%
  - Valid

Association Rule Mining - Subset Creation

- Non-Empty subset are
  - {{Milk}, {Bread}, {Butter}, {Milk, Bread}, {Milk, Butter}, {Bread, Butter}}
  - Min_Support = 30% and Min_Confidence = 60%

- Rule 5: {Milk, Butter} → {Bread} {S=50%, C=85.7%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Milk, Butter) = 6/7 = 85.7% >= 60%
  - Valid

- Rule 6: {Bread, Butter} → {Milk} {S=50%, C=66.67%}
  - Support = 6/12 = 50%
  - Confidence = Support (Milk, Bread, Butter)/Support(Bread, Butter) = 6/9 = 66.67%>=60
  - Valid

FP-Growth

- To avoid scanning multiple database
  - the cost of database is too high !!
- To avoid making lots of candidates
  - in apriori algorithm, the bottleneck is generation of candidate
- How can solve these problems?

FP-Growth

- Algorithm was too simple

TID	*Items bought*	*(ordered) frequent items*	
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}	
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}	
300	{*b, f, h, j, o, w*}	{*f, b*}	*min_support = 3*
400	{*b, c, k, s, p*}	{*c, b, p*}	
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}	

1. Scan the database once, find frequent 1-itemsets (single item patterns)
2. Sort the frequent items in frequency descending order, f-list(F-list = f-c-a-b-m-p)
3. Scan the DB again, construct the FP-tree

Frequent-pattern growth(FP-growth):

Finding frequent itemsets without candidate generation.

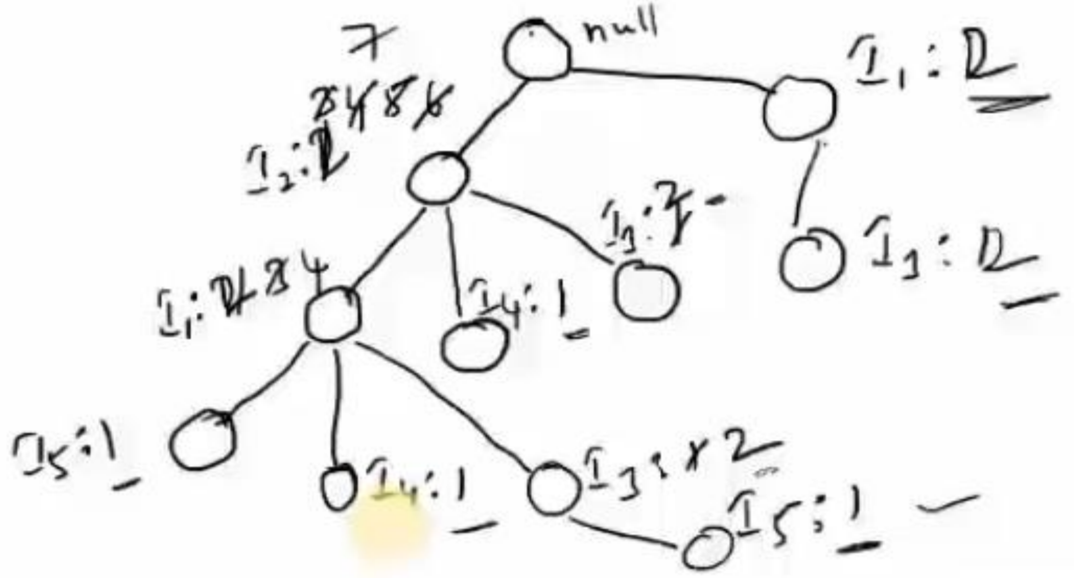
- First, compress the database representing frequent items into a **frequent-pattern tree, or FP-tree**,which retains the itemset association information.
- Then divide the compressed database into a set of **conditional databases** (a special kind of projected database), each associated with one frequent item or "pattern fragment," and mines each such database separately.

Frequent-pattern growth(FP-growth): Example

sup = 2

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

I2 I1 I5
I2 I4
I2 I3
I2 I1 I4
I2 I1 I3 I5
I2 I1 I3

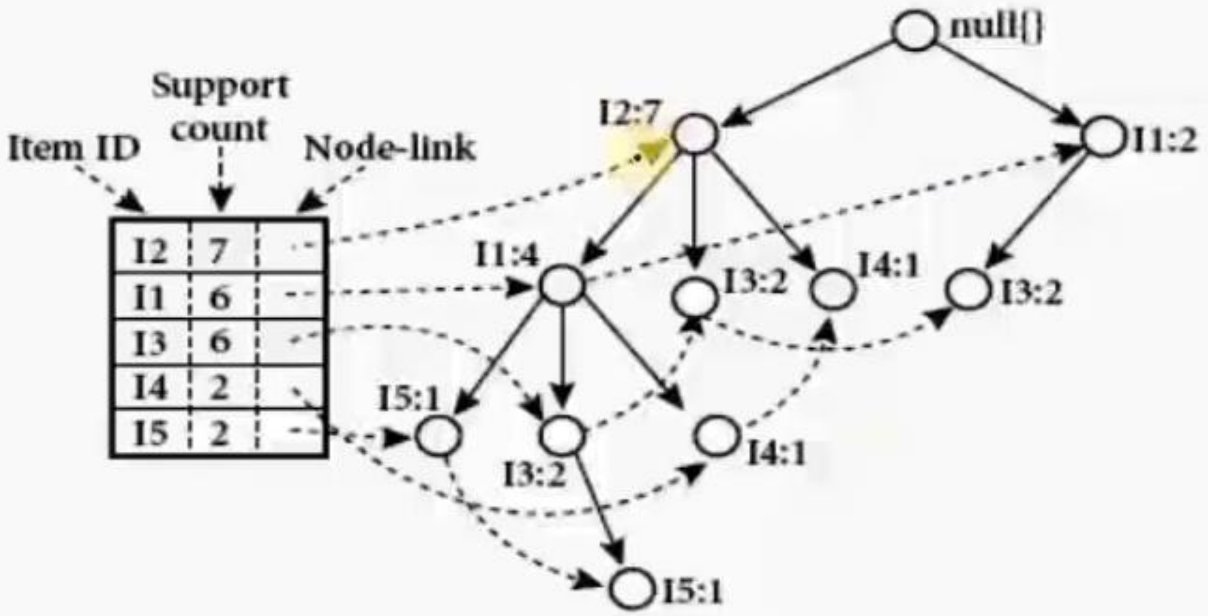


@ M3

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Items	Sup.count
I2	7
I1	6
I3	6
I4	2
I5	2

Frequent-pattern growth(FP-growth): Example



CSE GURUS @ M3

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

FP-Growth

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}

Let the minimum support be 3.

Frequent Pattern Growth Algorithm Solved Example - 1

The frequency of each individual

item is computed:-

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

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

A **Frequent Pattern set (L)** is built which will contain all the elements whose frequency is greater than or equal to the minimum support.

As minimum support be 3.

These elements are stored in descending order of their respective frequencies.

After insertion of the relevant items, the set L looks like this:- **L = {K : 5, E : 4, M : 3, O : 3, Y : 3}**

Now, for each transaction, the respective **Ordered-Item set** is built.

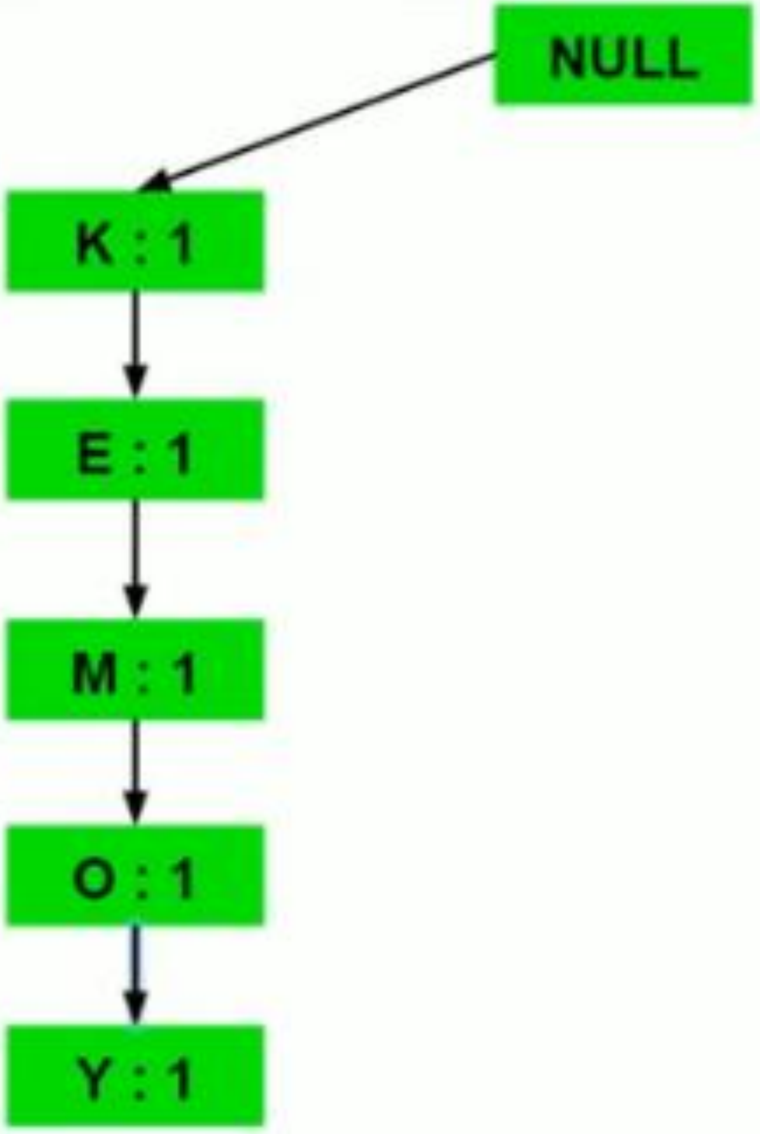
Frequent Pattern set L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

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}

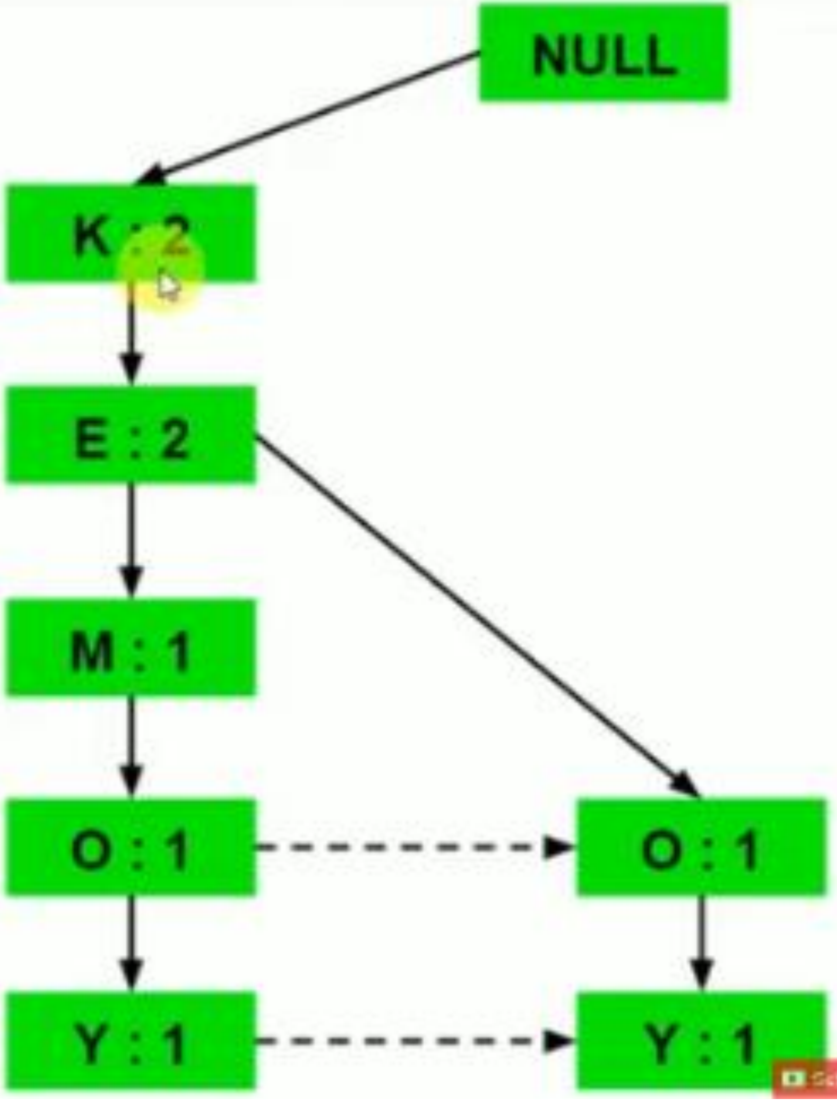
a) Inserting the set {K, E, M, O, Y}:



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}

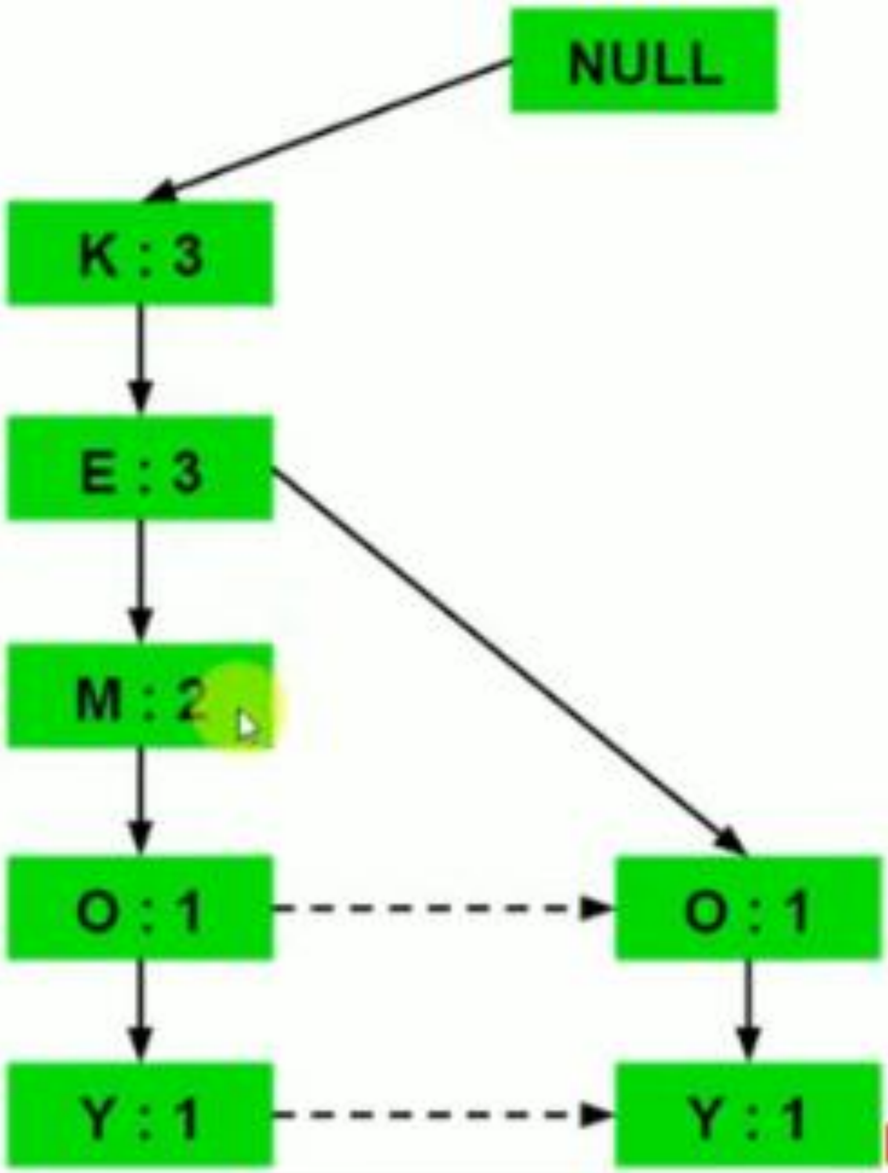
b) Inserting the set {K, E, O, Y}:



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}

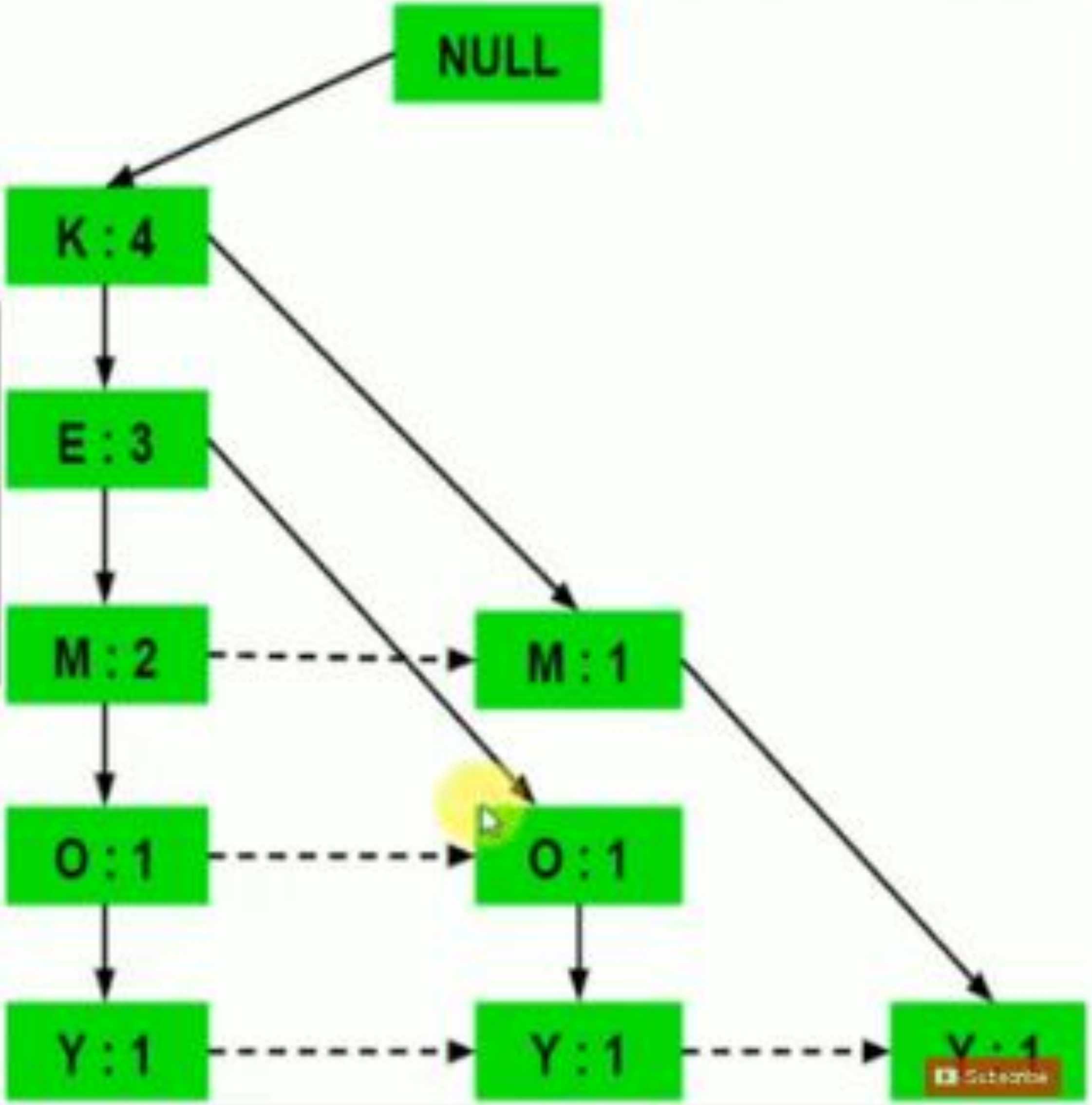
c) Inserting the set {K, E, M}:



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}

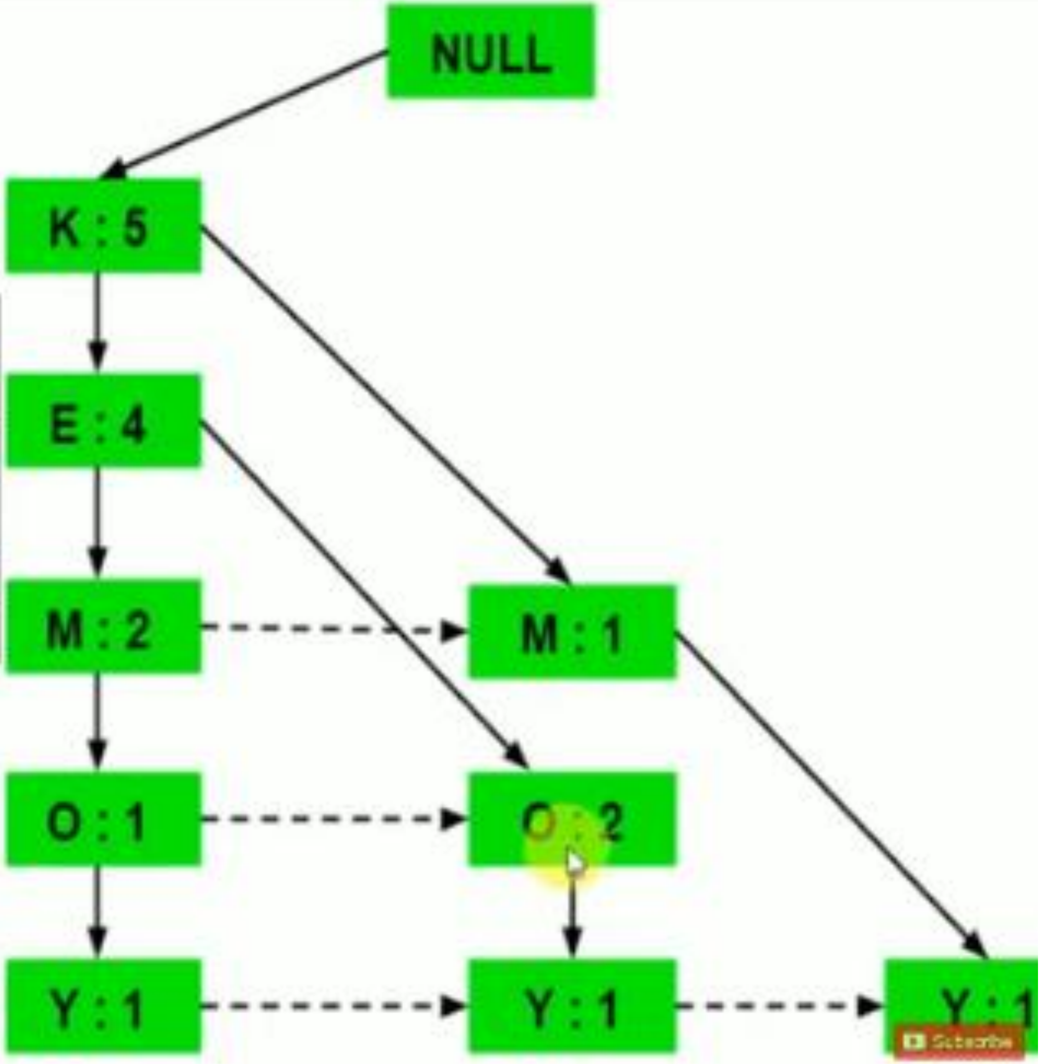
d) Inserting the set {K, M, Y}:



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

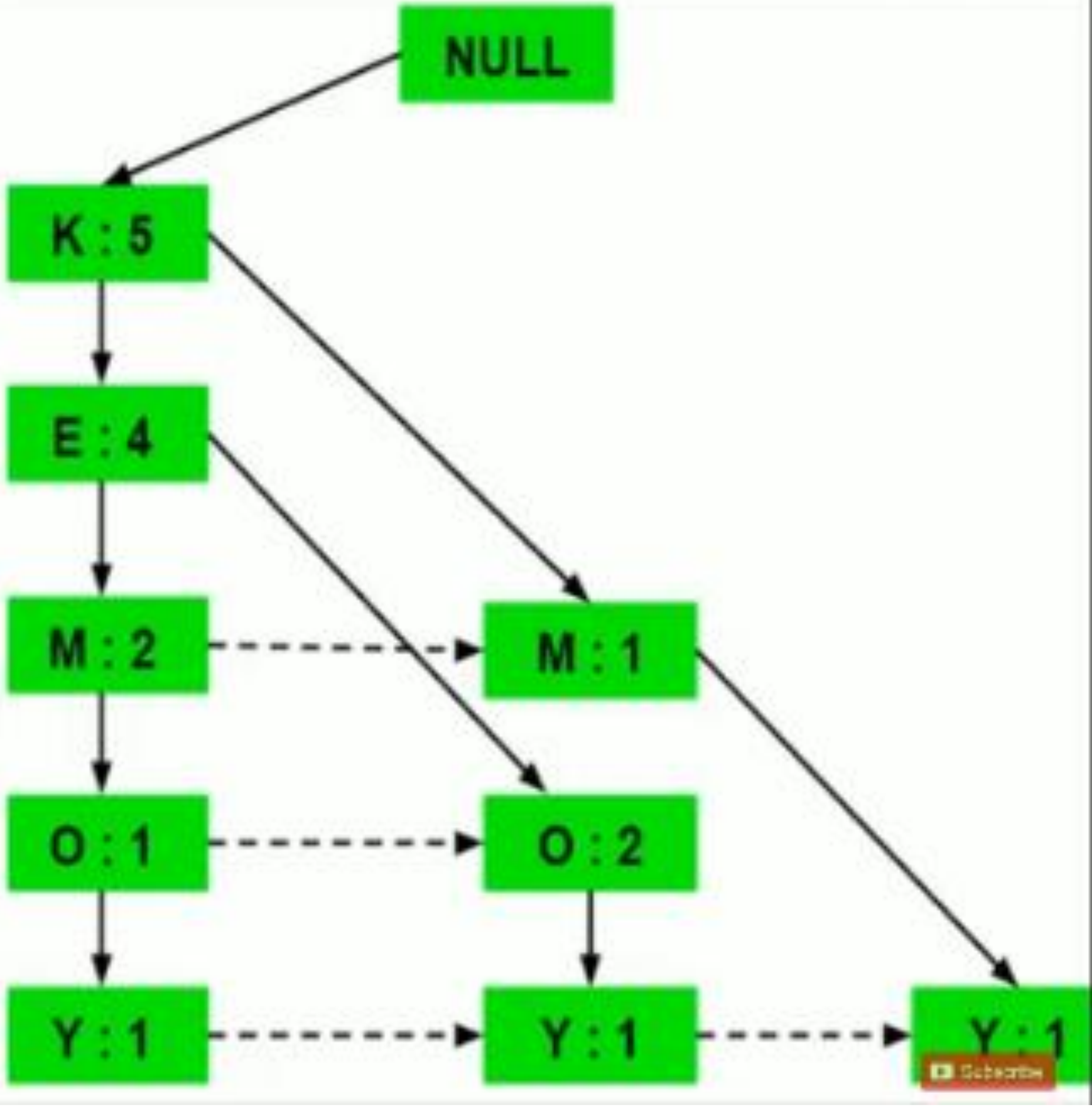
Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

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	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		

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	

Frequent Pattern Growth Algorithm Solved Example - 2

TID	Items Bought
100	*f, a, c, d, g, i, m, p*
200	*a, b, c, f, l, m, o*
300	*b, f, h, j, o*
400	*b, c, k, s, p*
500	*a, f, c, e, l, p, m, n*

Minimum Support - 3

Item	Frequency	Item	Frequency
a	3	j	1
b	3	k	1
c	4	l	2
d	1	m	3
e	1	n	1
f	4	o	2
g	1	p	3
h	1	s	1
i	1		

Frequent Pattern Growth Algorithm Solved Example - 2

Now, for each transaction, the respective **Ordered-Item set** is built.

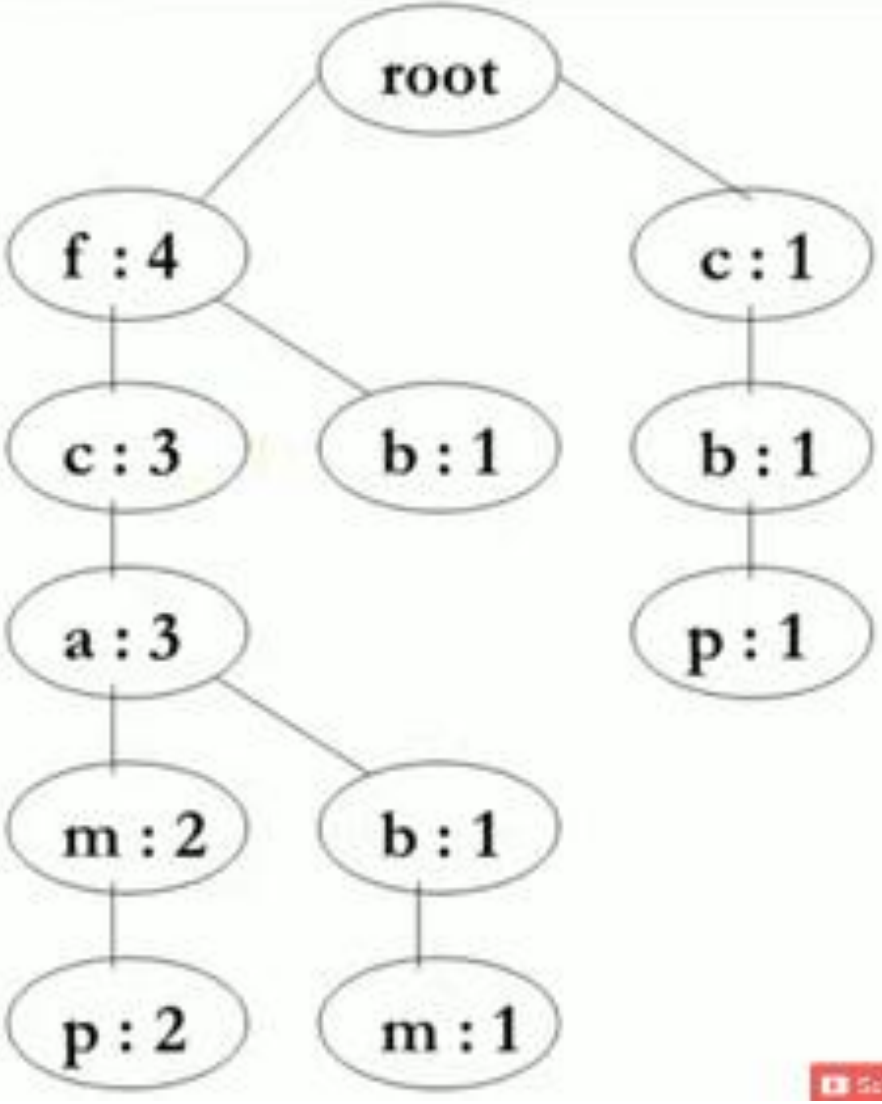
Frequent Pattern set L = { (f:4), (c:4), (a:3), (b:3), (m:3), (p:3) }

TID	Items Bought	(Ordered) Frequent Items
100	*f, a, c, d, g, i, m, p*	*f, c, a, m, p*
200	*a, b, c, f, l, m, o*	*f, c, a, b, m*
300	*b, f, h, j, o*	*f, b*
400	*b, c, k, s, p*	*c, b, p*
500	*a, f, c, e, l, p, m, n*	*f, c, a, m, p*

Frequent Pattern Growth Algorithm Solved Example - 2

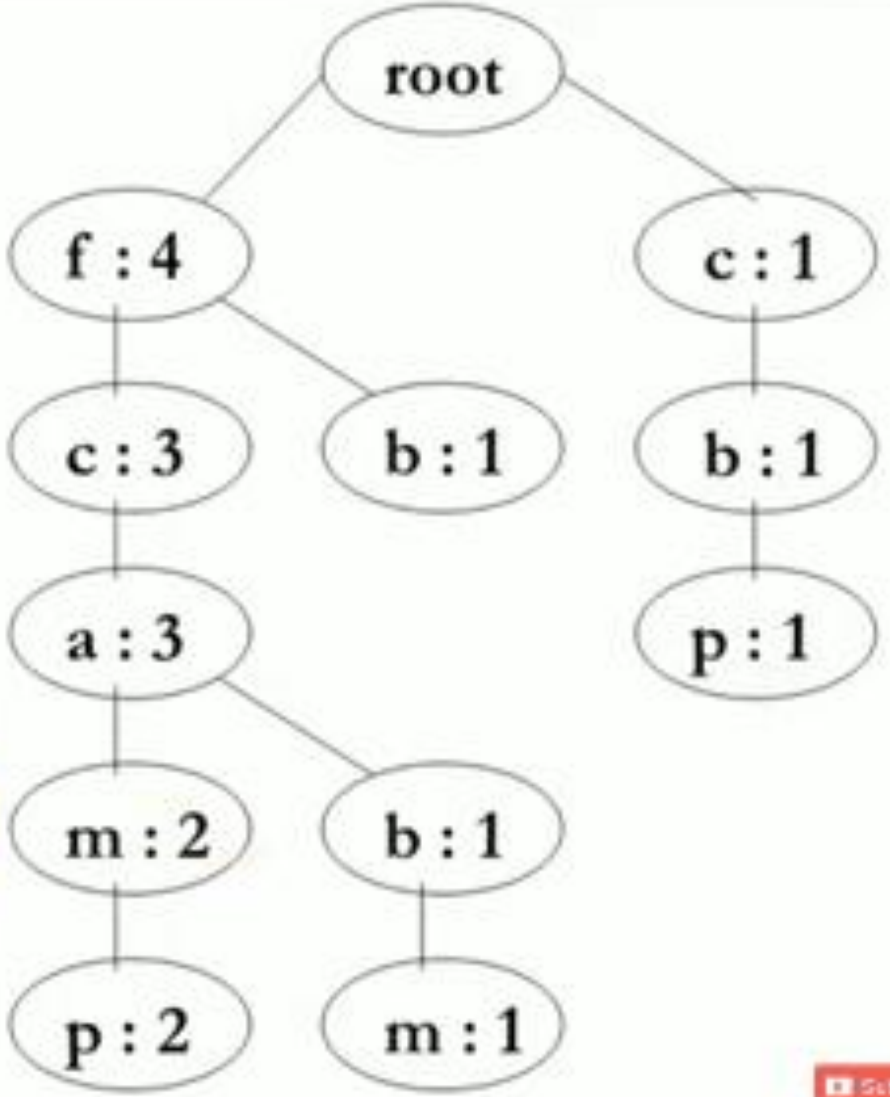
Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

TID	Items Bought	(Ordered) Frequent Items
100	*f, a, c, d, g, i, m, p*	*f, c, a, m, p*
200	*a, b, c, f, l, m, o*	*f, c, a, b, m*
300	*b, f, h, j, o*	*f, b*
400	*b, c, k, s, p*	*c, b, p*
500	*a, f, c, e, l, p, m, n*	*f, c, a, m, p*



Frequent Pattern Growth Algorithm Solved Example - 2

Item	Conditional Pattern Base
p	{{f, c, a, m : 2}, {c, b : 1}}
m	{{f, c, a : 2}, {f, c, a, b : 1}}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}
a	{{f, c : 3}}
c	{{f : 3}}
f	Φ



Frequent Pattern Growth Algorithm Solved Example - 2

Item	Conditional Pattern Base	Conditional FP-Tree
p	{{f, c, a, m : 2}, {c, b : 1}}	{c : 3}
m	{{f, c, a : 2}, {f, c, a, b : 1}}	{f, c, a :3}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}	Φ
a	{{f, c : 3}}	{f, c : 3}
c	{{f : 3}}	{f : 3}
f	Φ	Φ

- 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 item.

Frequent Pattern Growth Algorithm Solved Example - 2

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Patterns Generated
p	{{f, c, a, m : 2}, {c, b : 1}}	{c : 3}	{<c, p : 3>}
m	{{f, c, a : 2}, {f, c, a, b : 1}}	{f, c, a :3}	{ <f, m : 3>, <c, m : 3> <a, m : 3>, <f, c, m : 3> <f, a, m : 3>, <c, a, m :3>}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}	Φ	{}
a	{{f, c : 3}}	{f, c : 3}	{<f, a : 3>, <c, a : 3>, <f, c, a:3>}
c	{{f : 3}}	{f : 3}	{ <f, c : 3>}
f	Φ	Φ	{}

Data Warehouse and [Mumbai Univ, Pune Univ, GTU, PTU,]
Data Mining Lecture Series [UPTU, GGSIPU and other Univ.]

Solved Question on FP-GROWTH Algorithm

Ques.) Generate FP-Tree for the following Transaction Data Set. [Minimum Support = 30%]

Tr.Id.	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

min. no. of Trⁿ = 2.4 ⇒ 3

items	Frequency	Priority
A →	5 →	3
B →	6 →	1
C →	3 →	5
D →	6 →	2
E →	4 →	4

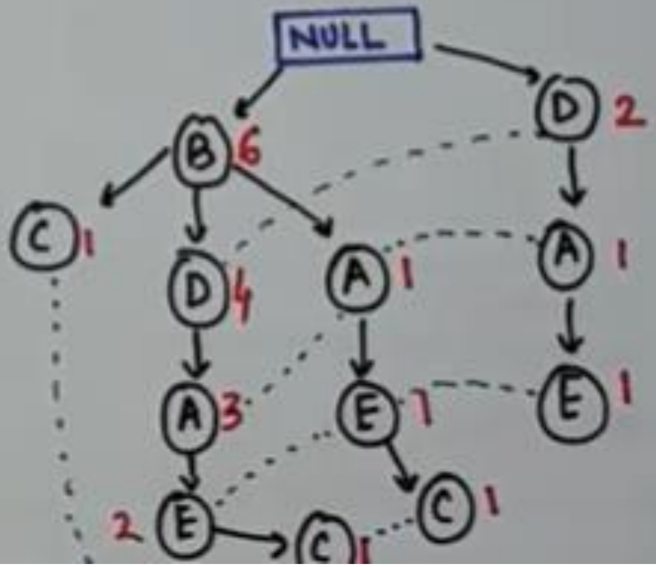
Lower Priority no means High Priority.

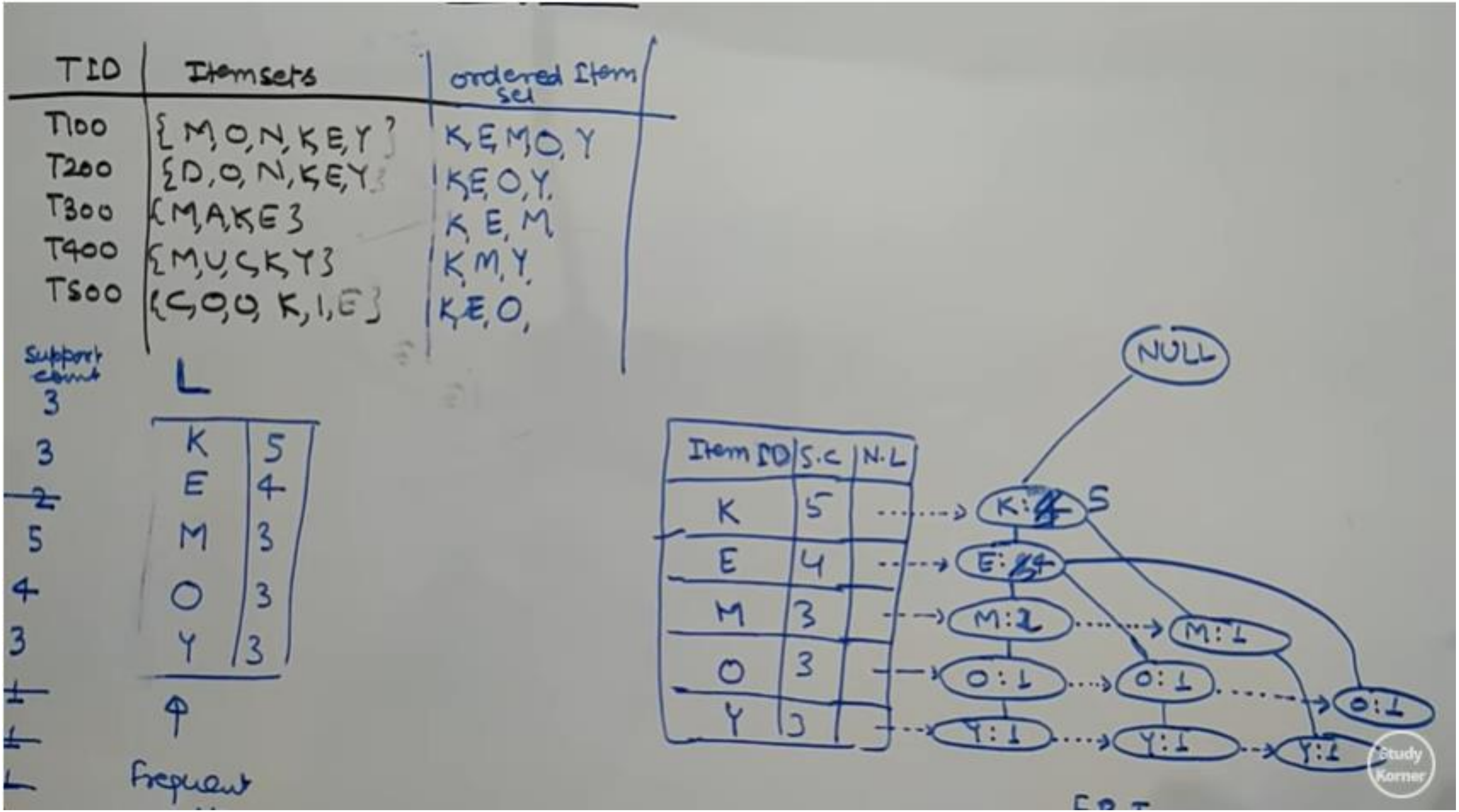
Order the items According to the priority.

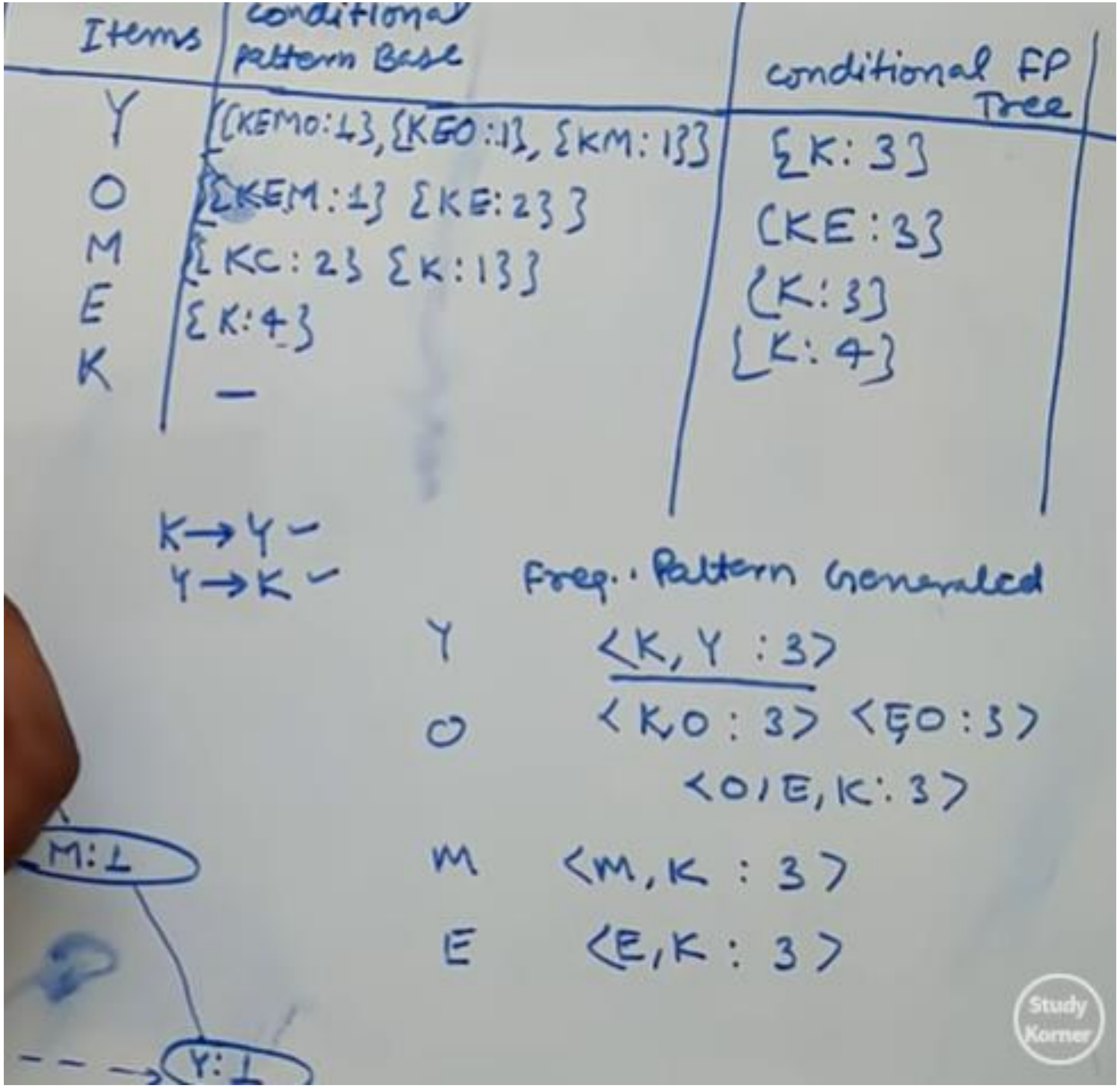
Tr.Id	Items	Ordered Items
1	E, A, D, B	B, D, A, E ✓
2	D, A, C, E, B	B, D, A, E, C ✓
3	C, A, B, E	B, A, E, C ✓
4	B, A, D	B, D, A ✓
5	D	D ✓
6	D, B	B, D ✓
7	A, D, E	D, A, E ✓
8	B, C	B, C

B: 1, 2, 3, 4, 5, 6
D: 1, 2, 3, 4
A: 1, 2, 3
E: 1, 2
C: 1
A: 1 → 3
E: 1
C: 1

D: 1, 2
A: 1
E: 1
C: 1




TID | Itemsets | ordered Item set
T100 {M, O, N, K, E, Y} K, E, M, O, Y
T200 {D, O, N, K, E, Y} K, E, O, Y.
T300 {M, A, K, E} K, E, M
T400 {M, U, C, K, Y} K, M, Y.
T500 {C, O, O, K, I, E} K, E, O,
support count
L
K 5
E 4
M 3
O 3
Y 3
frequent
NULL
Item ID | S.C | N.L
K 5
E 4
M 3
O 3
Y 3
K: 5
E: 4
M: 2
M: 1
O: 1
O: 1
O: 1
Y: 1
Y: 1
Y: 1
Study Korner


Items
conditional pattern Base
conditional FP Tree
Y {{KEMO:1}, {KEO:1}, {KM:1}} {K:3}
O {{KEM:1} {KE:2}} {KE:3}
M {{KC:2} {K:1}} {K:3}
E {K:4} {K:4}
K —
K→Y
Y→K
Freq. Pattern Generated
Y <K, Y : 3>
O <K,O : 3> <E,O : 3>
<O, E, K : 3>
M <M, K : 3>
E <E, K : 3>
M:1
Y:1
Study Korner

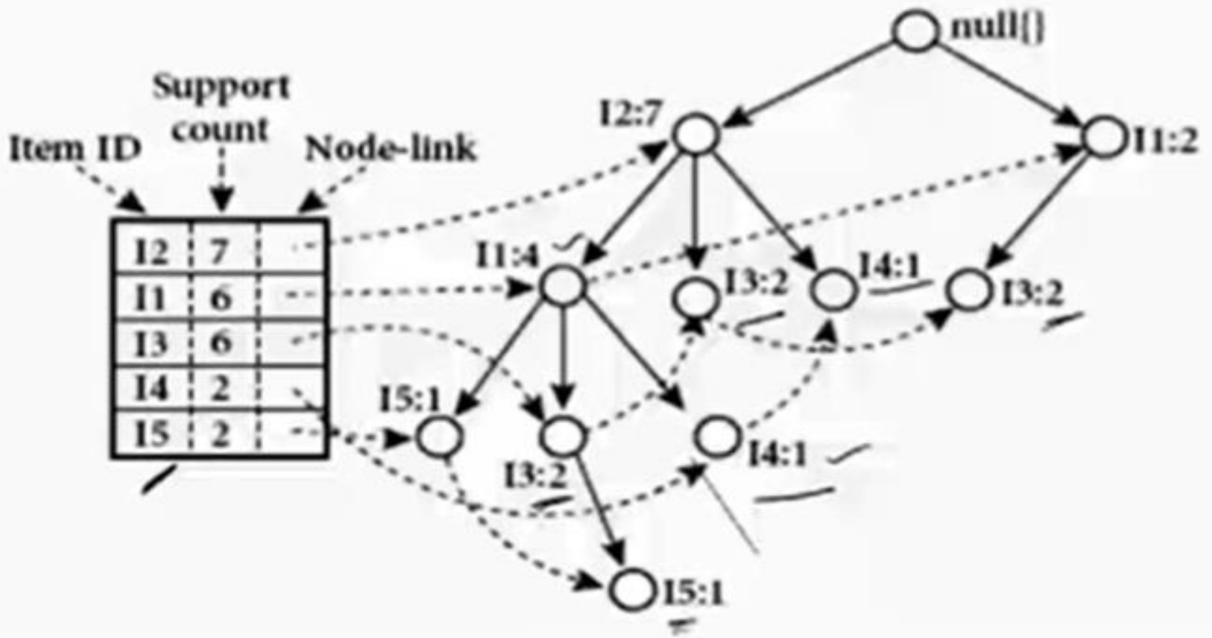
Association rule mining :

Association rule mining can be viewed as a two-step process:

1. Find all frequent item sets
    1. Apriori method
    2. FP-Growth

2. Generate strong association rules from the frequent item sets:
-By definition, these rules must satisfy minimum support and minimum confidence.

Frequent-pattern growth(FP-growth): Example



min sup = 2

CSE GURUS @ M3

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Association rules can be generated as follows:

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "$s \Rightarrow (l - s)$" if $\frac{support\_count(l)}{support\_count(s)} \geq min\_conf$, where min_conf is the minimum confidence threshold.

$l = \{I_1, I_2, I_5\}$

1) $I_1 \rightarrow I_2 \wedge I_5$ Conf = 2/6 = 33.3%
2) $I_2 \rightarrow I_1 \wedge I_5$ Conf = 2/7 = 30%
3) $I_5 \rightarrow I_2 \wedge I_5$ Conf = 2/2 = 100%
4) $I_1 \wedge I_2 \rightarrow I_5$ Conf = 2/4 = 50%
5) $I_1 \wedge I_5 \rightarrow I_2$ Conf = 2/2 = 100%
6) $I_2 \wedge I_5 \rightarrow I_1$ Conf = 2/2 = 100%

$\cancel{I_1 \wedge I_2 \wedge I_5} \rightarrow \phi$

min. Conf = 60%

3, 5, 6 are Strong

Eg:

Tid	List of items
T_{100}	I_1, I_2, I_5
T_{200}	I_2, I_4
T_{300}	I_2, I_3
T_{400}	I_1, I_2, I_4
T_{500}	I_1, I_3
T_{600}	I_2, I_3
T_{700}	I_1, I_3
T_{800}	I_1, I_2, I_3, I_5
T_{900}	I_1, I_2, I_3

$\boxed{\text{Min\_Sup} = 2}$

① List the Items and its count

I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

FP Growth Alg — Mining frequent Itemsets without Candidate Generation

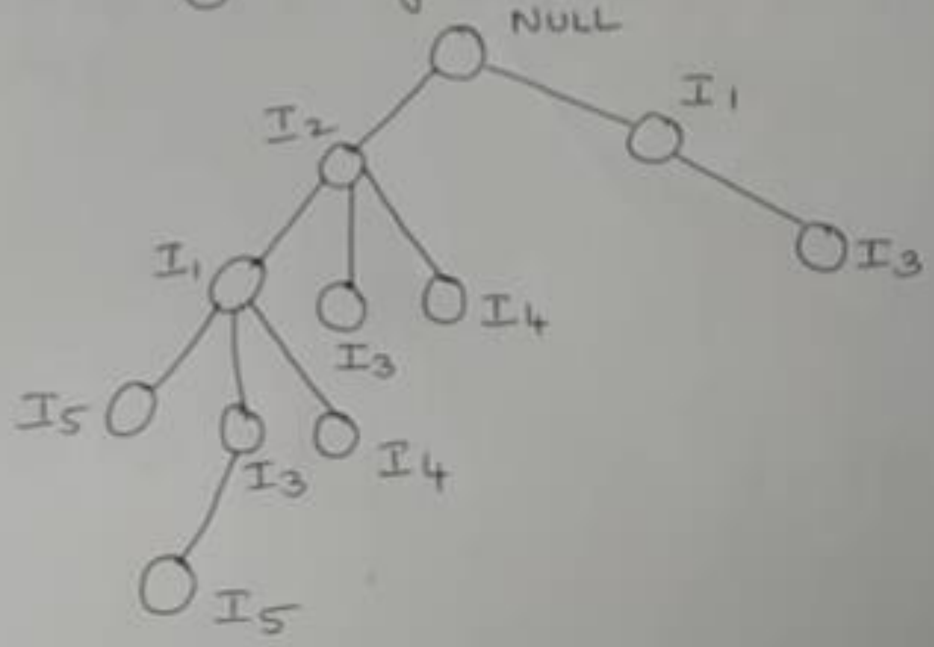
Disadvantage of Apriori Alg

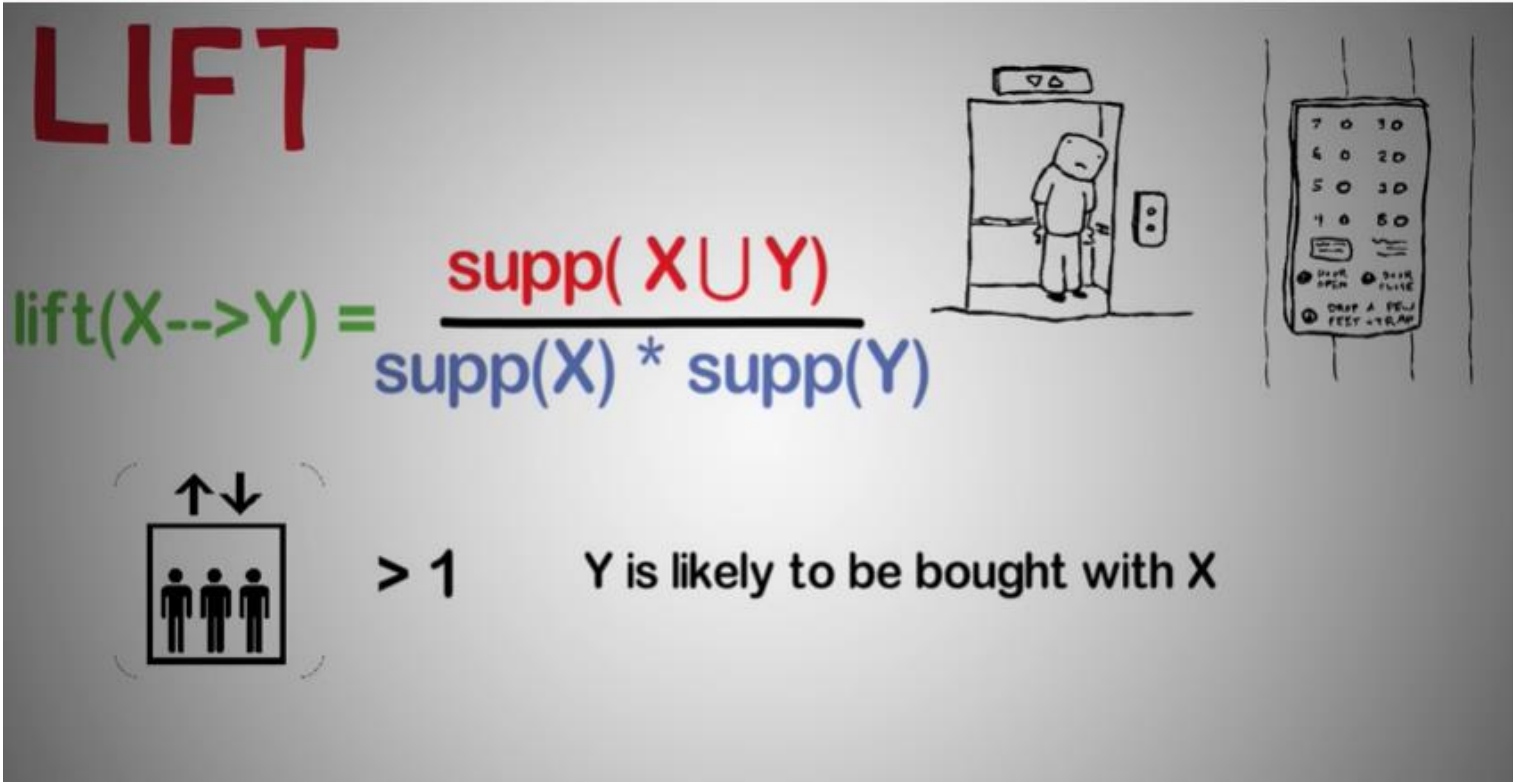
1. It May need to generate a huge number of candidate sets.
2. It may need to repeatedly scan the database and check a large set of candidates by pattern matching

② Sort it in descending order

I_2	7
I_1	6
I_3	6
I_4	2
I_5	2

③ Root of the tree is null




LIFT
lift(X-->Y) = supp(X∪Y) / supp(X) * supp(Y)
> 1
Y is likely to be bought with X

The third measure called the lift or lift ratio is the ratio of confidence to expected confidence. Expected confidence is the confidence divided by the frequency of B. The Lift tells us how much better a rule is at predicting the result than just assuming the result in the first place. Greater lift values indicate stronger associations.

$$\textbf{Lift} = \left(\frac{\left(\frac{(\textbf{A} + \textbf{B})}{\textbf{A}} \right)}{\left(\frac{\textbf{B}}{\textbf{Total}} \right)} \right)$$

$$\textbf{Lift for Basket 1} = \left(\frac{\left(\frac{(\textbf{Milk} + \textbf{Cheese})}{\textbf{Milk}} \right)}{\left(\frac{\textbf{Cheese}}{\textbf{Total}} \right)} \right) = \left(\frac{\left(\frac{6}{6} \right)}{\left(\frac{7}{9} \right)} \right) = \left(\frac{1}{.7777778} \right) = 1.2857$$

Interpreted as: How much our confidence has increased that B will be purchased *given* that A was purchased.