# ASSOCIATION RULE MINING

BEIYU LIN

# ASSOCIATION RULE MINING

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items

$X \longrightarrow Y$

$X = \{\text{Beer}\}$ $Y = \{\text{eggs}\}$

Market transactions

| *TID* | *Items* |
|---|---|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

Example of Association Rules

{Beer} → {Eggs},
{Milk, Bread} → {Diaper, Beer},

$\sigma(X) = 3 \geq 2$

$s(X) = \frac{\sigma(X)}{5} = \frac{3}{5}$

$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} = \frac{1}{3}$

$\sigma(Y) = 1 \ngeq 2$

$s(Y) = \frac{1}{5}$

$\min \sigma = 2$ $\min s = \frac{\min \sigma}{\boxed{\phantom{5}}}$

$\sigma(X \cup Y) = 1$

sup cnt $\sigma$ : sup $s$ , conf $c$ , freq $\geq \min \sigma$ or $\min s$

# ASSOCIATION RULE MINING

- **Itemset (set / subset)**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2

| *TID* | *Items* |
|---|---|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

**Support**

Fraction of transactions that contain an itemset

E.g. s({Milk, Bread, Diaper}) = 2/5

**Frequent Itemset**

An itemset whose support is greater than or equal to a *minsup* threshold

# DEFINITION: ASSOCIATION RULE

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}

| *TID* | *Items* |
|---|---|
| **1** | **Bread, Milk** |
| **2** | **Bread, Diaper, Beer, Eggs** |
| **3** | **Milk, Diaper, Beer, Coke** |
| **4** | **Bread, Milk, Diaper, Beer** |
| **5** | **Bread, Milk, Diaper, Coke** |

- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \{\text{Beer}\}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

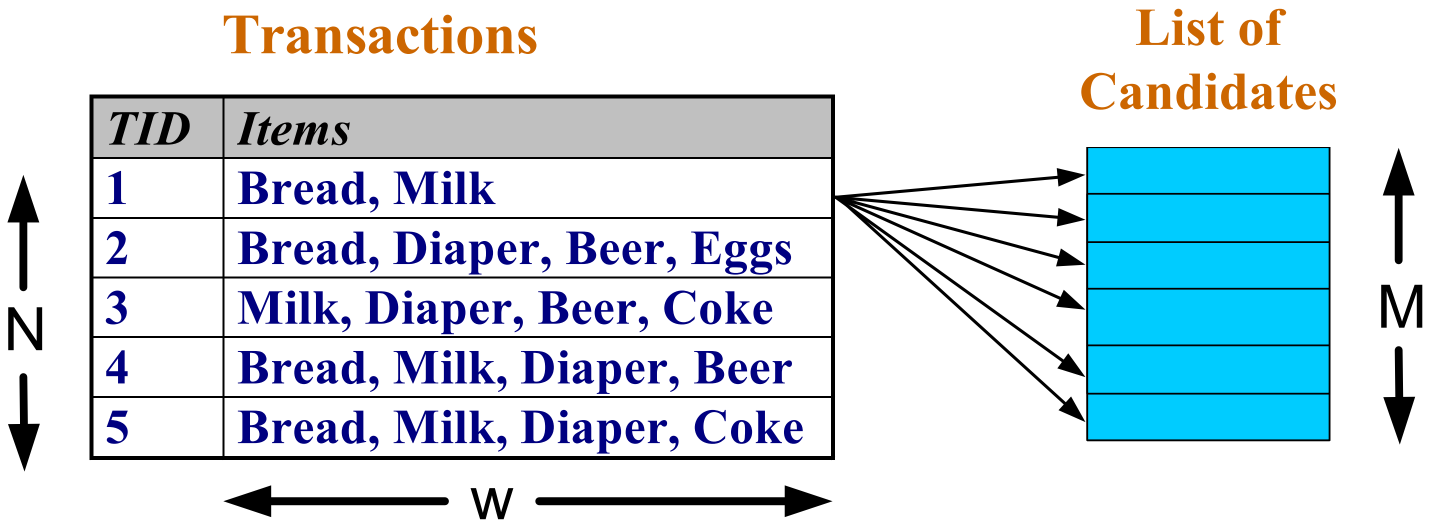
=# of itemset / total # transac

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

= # of itemset of XandY/ # of

# FREQUENT ITEMSET GENERATION

- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database



Transactions

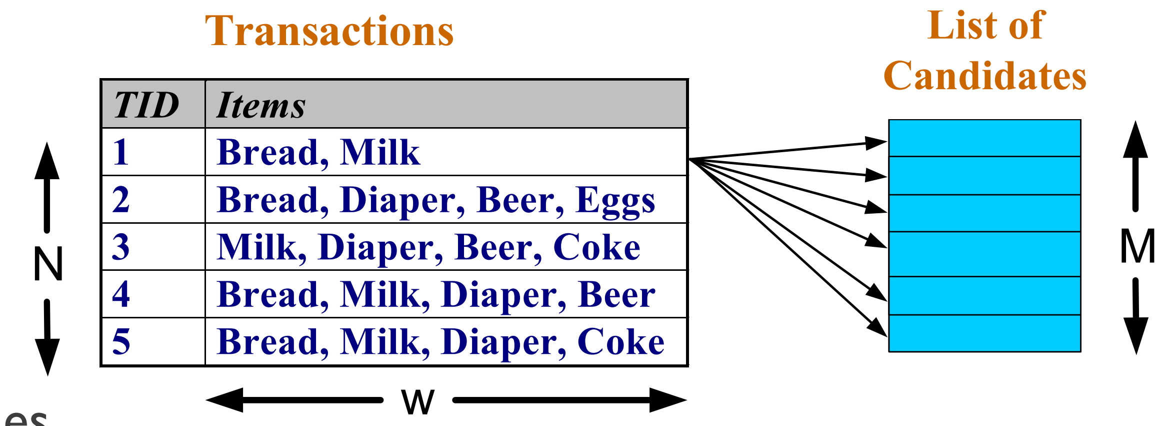
| *TID* | *Items* |
|---|---|
| **1** | **Bread, Milk** |
| **2** | **Bread, Diaper, Beer, Eggs** |
| **3** | **Milk, Diaper, Beer, Coke** |
| **4** | **Bread, Milk, Diaper, Beer** |
| **5** | **Bread, Milk, Diaper, Coke** |

List of Candidates

  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$ !!!

# FREQUENT ITEMSET GENERATION STRATEGIES

- Reduce the number of candidates (M)
  - Complete search: M=$\underline{2^d}$ $d=6$ $2^6$
  - Use pruning techniques to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by DHP and vertical-based mining algorithms
- Reduce the number of comparisons (N$\underline{M}$)
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction



**Transactions**

| *TID* | *Items* |
|---|---|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

$I = \{A, B, C\}$ $2^3$

$I = \{B, M, D, B, E, C\}$

# REDUCING NUMBER OF CANDIDATES

| *TID* | *Items* |
|---|---|
| **1** | **Bread, Milk** ✓ |
| **2** | **Bread, Diaper, Beer, Eggs** |
| **3** | **~~Milk, Diaper~~, Beer, Coke** |
| **4** | **~~Bread, Milk, Diaper~~, Beer** |
| **5** | **~~Bread, Milk, Diaper~~, Coke** |

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
- Apriori principle holds due to the following property of the support measure:

$$\forall X,Y:(X\subseteq Y)\Rightarrow s(X)\geq s(Y)$$

for any $X, Y$: $\sigma(X) \geq \sigma(Y)$

  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

**Support**

$\nearrow 2^3$

$f$: $\underbrace{s(\{\text{Milk, Bread, Diaper}\})}_{X_1} = 2/5$

$\sigma(X_1) = 2 \geq \min\sigma = 2$

$X \rightarrow Y$

$X = \{\text{Beers}\}$ $f$ $\sigma(X) = 3$

$X_{11} = \{\text{milk}\} \subseteq X_1$ $\sigma(X_{11}) = 4 \geq \min\sigma$

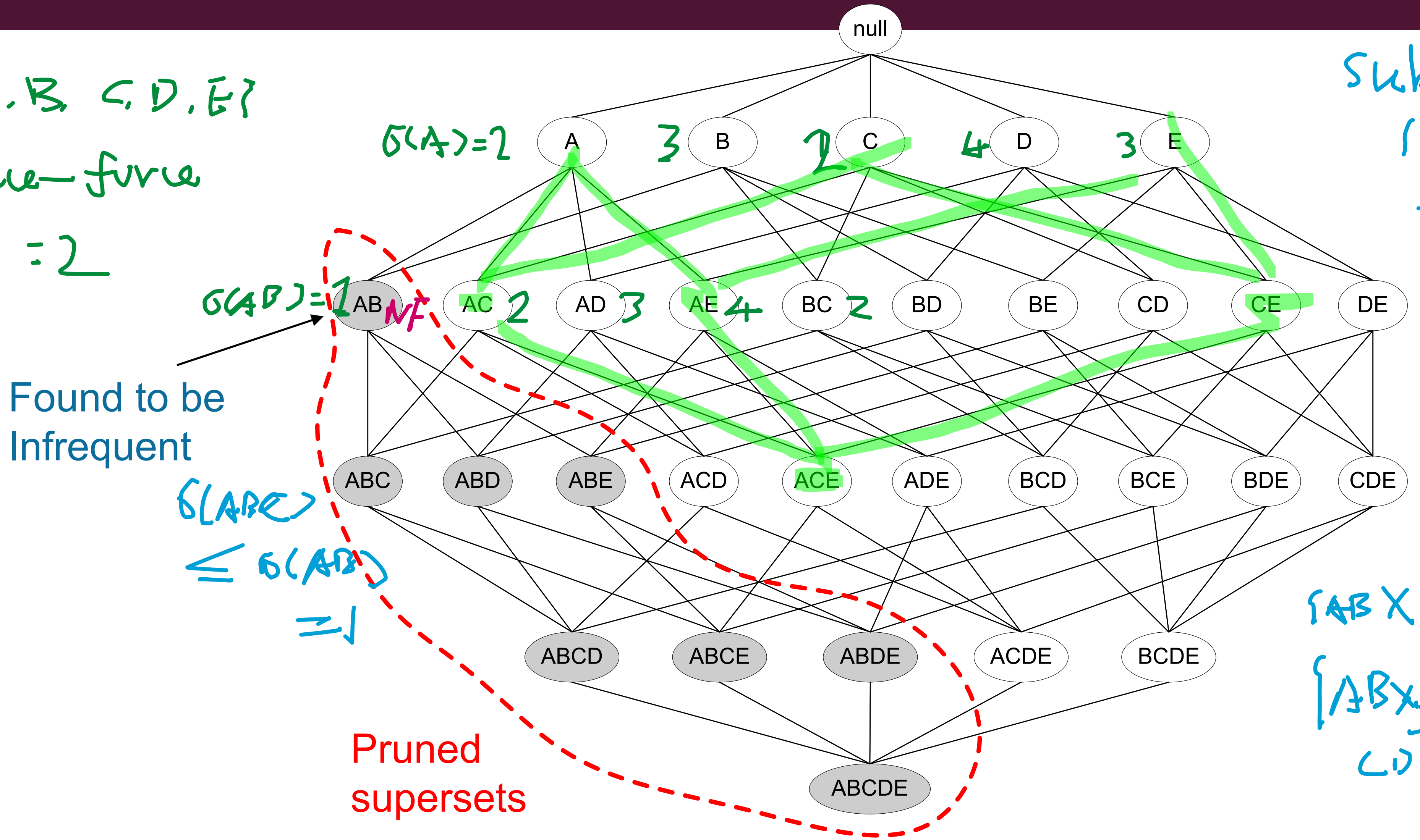
$X_{11}$ $f$ $\Leftarrow f$ $\sigma(X_{11}) \geq \sigma(X_1)$

# ILLUSTRATING APRIORI PRINCIPLE

E={A,B,C,D,E}

Brute-force

min σ =2



σ(ABE) ≤ σ(AB) =1

Subset:

{}, {A}...{E}

---- {A...E}

superset:

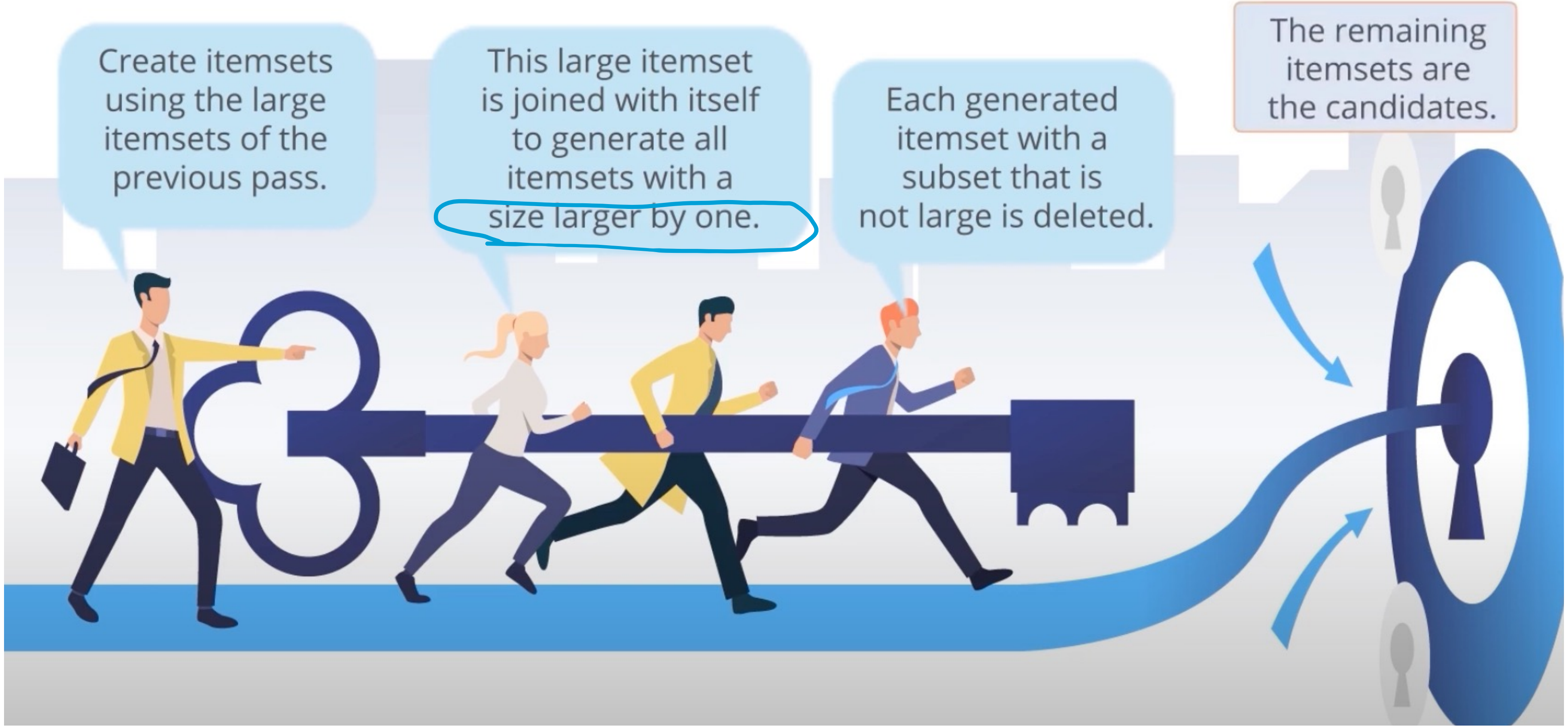
{AB} → ?

{ABX} {ABXXX}

{ABXX} CDE

CD, CE, DE

# APRIORI PRINCIPLE



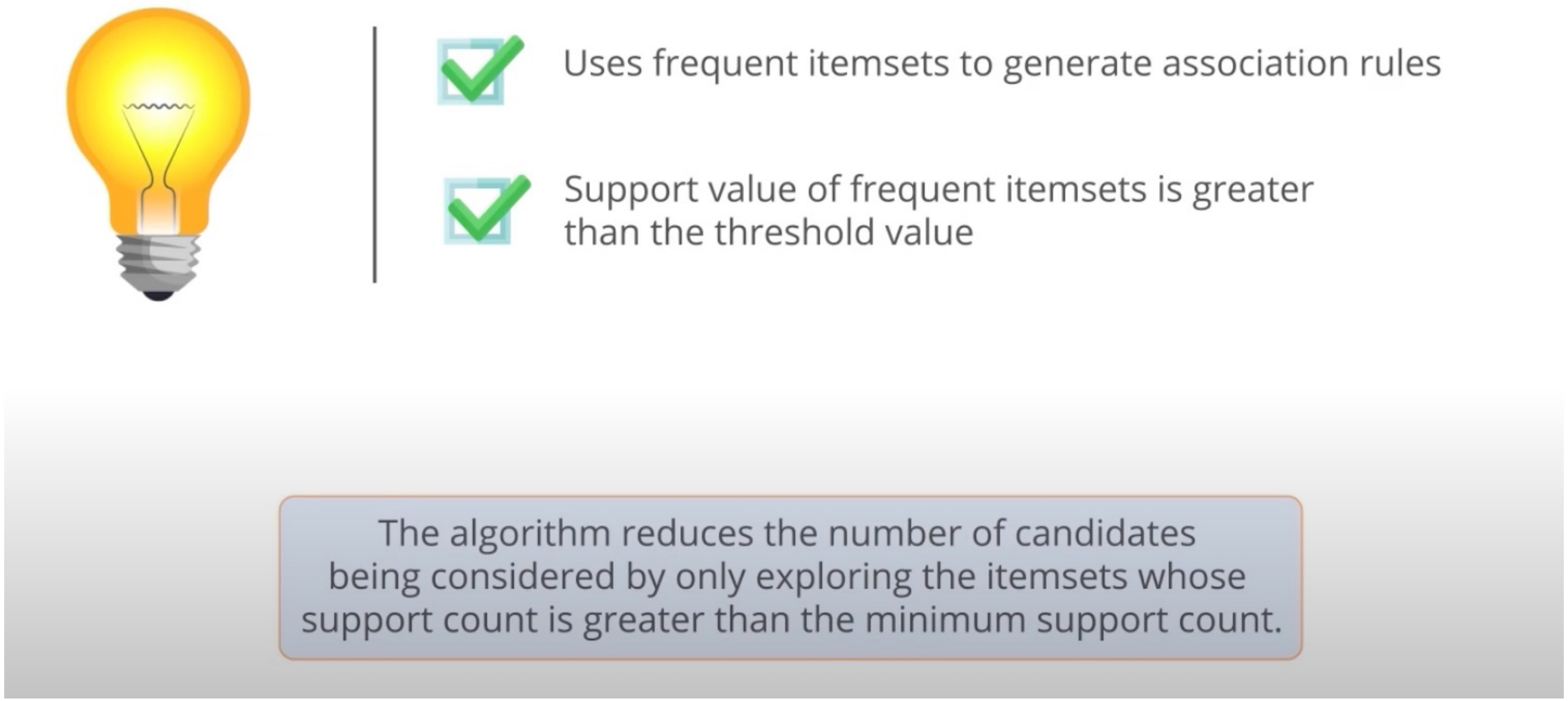
Create itemsets using the large itemsets of the previous pass.

This large itemset is joined with itself to generate all itemsets with a size larger by one.

Each generated itemset with a subset that is not large is deleted.

The remaining itemsets are the candidates.

# APRIORI PRINCIPLE

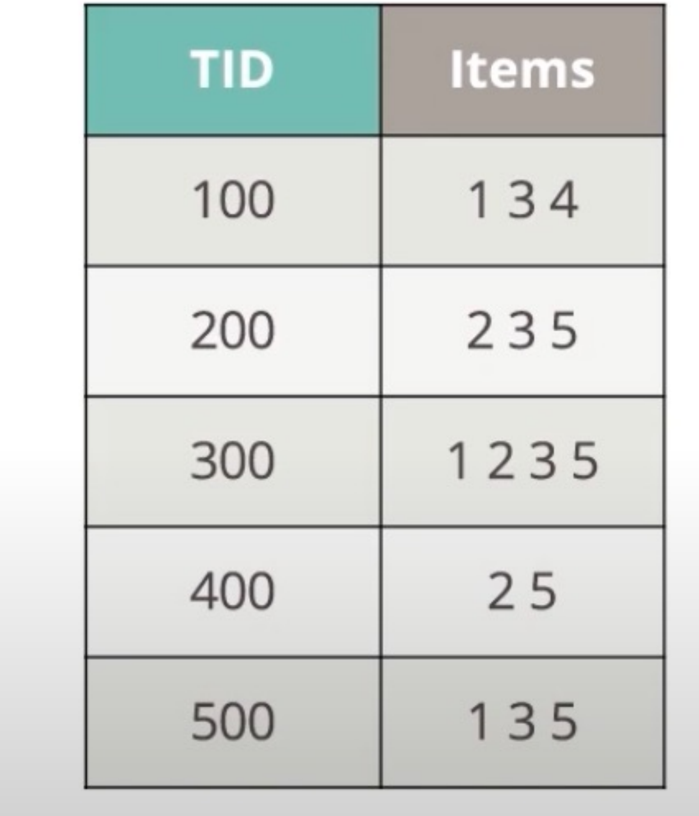


☑ Uses frequent itemsets to generate association rules

☑ Support value of frequent itemsets is greater than the threshold value

The algorithm reduces the number of candidates being considered by only exploring the itemsets whose support count is greater than the minimum support count.

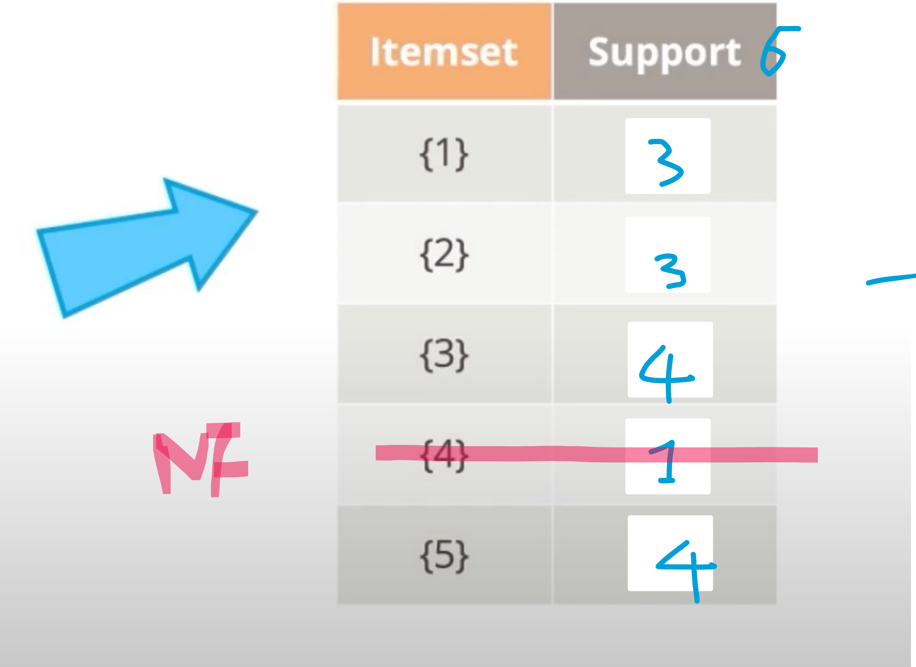
# APRIORI PRINCIPLE EXAMPLE



| TID | Items |
|---|---|
| 100 | 1 3 4 |
| 200 | 2 3 5 |
| 300 | 1 2 3 5 |
| 400 | 2 5 |
| 500 | 1 3 5 |

item=1

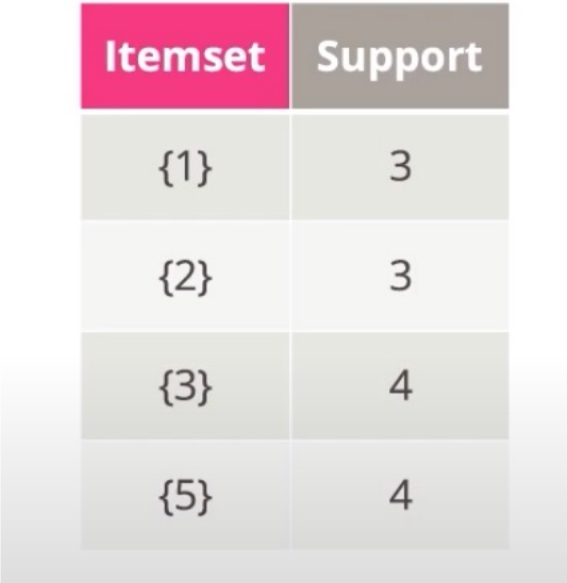
CI1



| Itemset | Support σ |
|---|---|
| {1} | 3 |
| {2} | 3 |
| {3} | 4 |
| ~~{4}~~ | ~~1~~ |
| {5} | 4 |

NF

F11

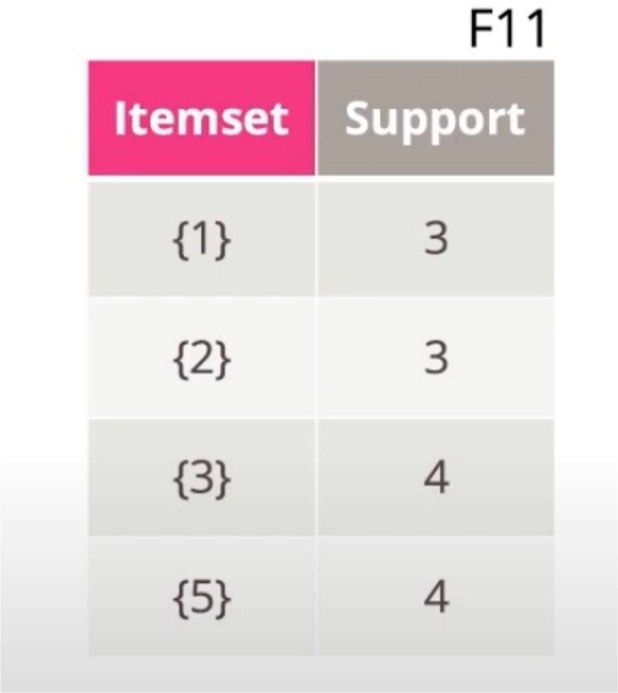


| Itemset | Support |
|---|---|
| {1} | 3 |
| {2} | 3 |
| {3} | 4 |
| {5} | 4 |

$I = \{1, 2, 3, 4, 5\}$ $\min \sigma = 2$

# APRIORI PRINCIPLE EXAMPLE

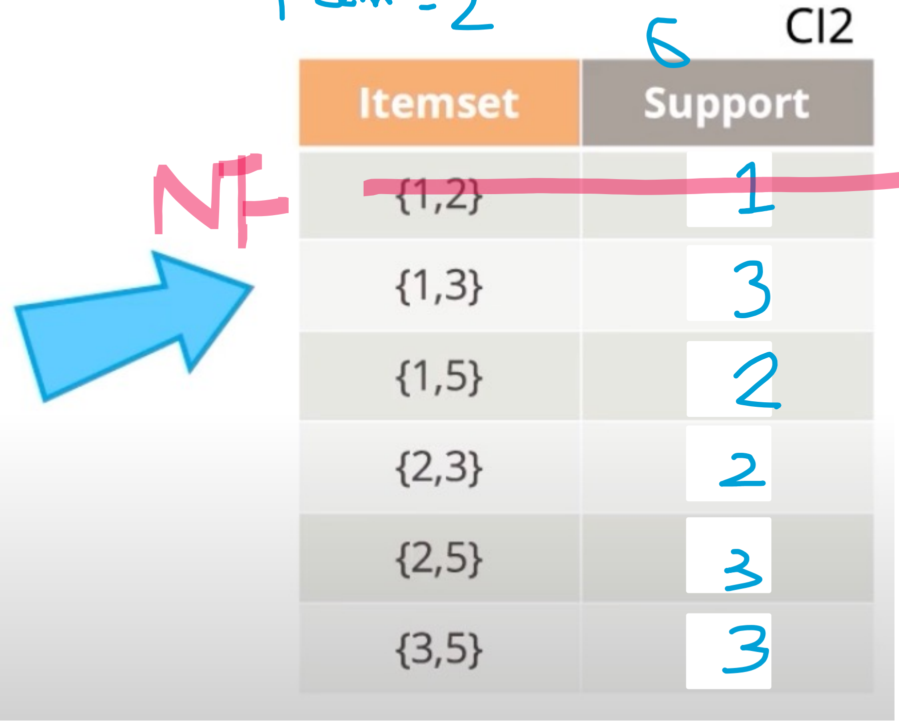
The length of the itemset is extended with 1 (k = k+1).



F11

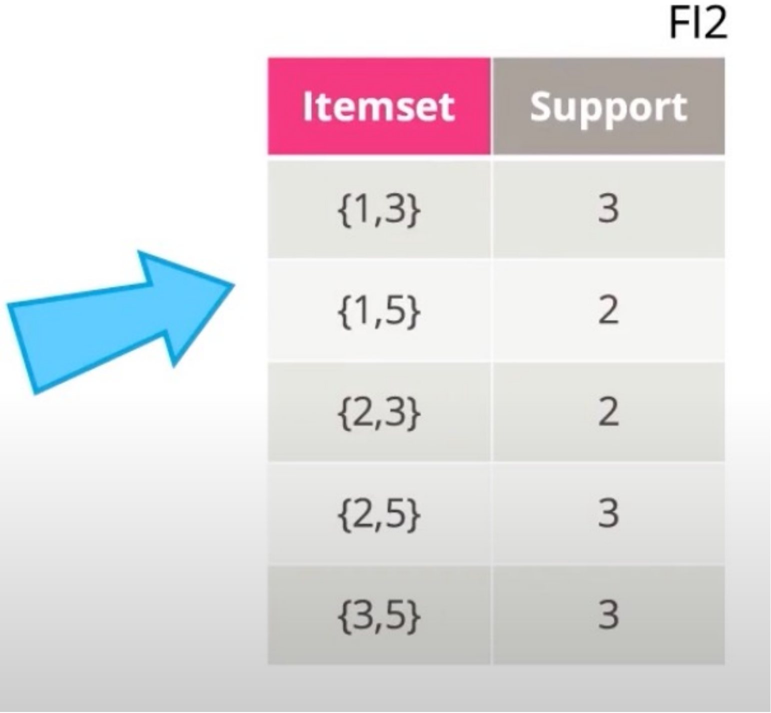
| Itemset | Support |
|---|---|
| {1} | 3 |
| {2} | 3 |
| {3} | 4 |
| {5} | 4 |

item = 2



6 CI2

| Itemset | Support |
|---|---|
| NF ~~{1,2}~~ | ~~1~~ |
| {1,3} | 3 |
| {1,5} | 2 |
| {2,3} | 2 |
| {2,5} | 3 |
| {3,5} | 3 |

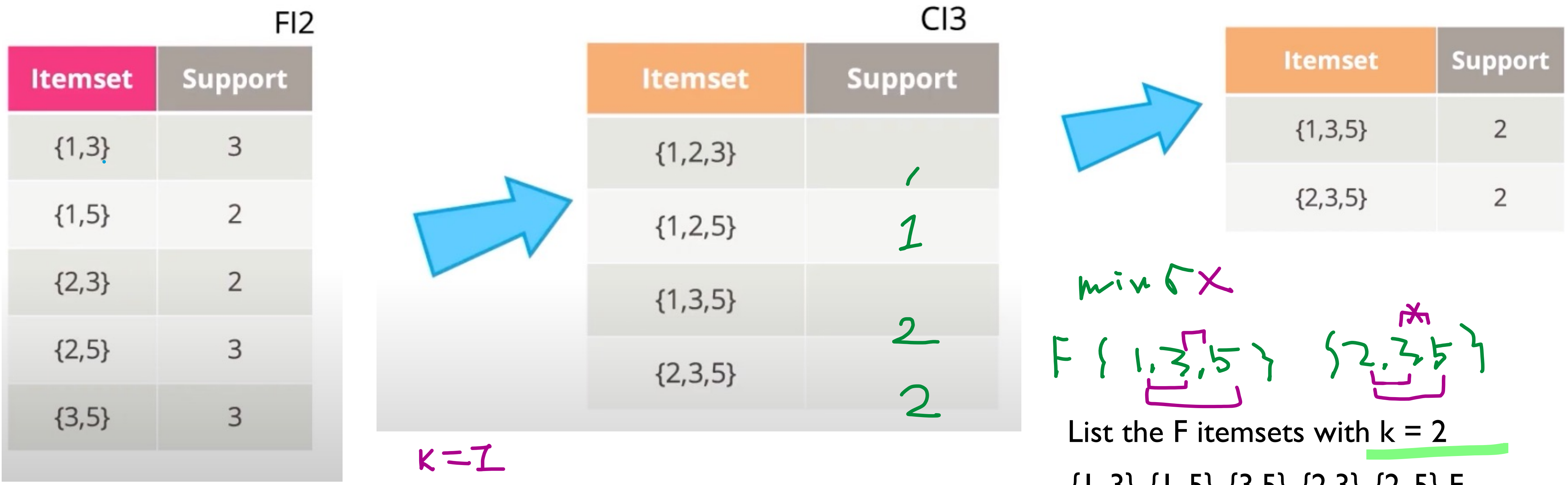


FI2

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

# APRIORI PRINCIPLE EXAMPLE

The length of the itemset is extended with 1 (k = k+1).



FI2

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

CI3

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

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

min σ x

F {1,3,5} {2,3,5}

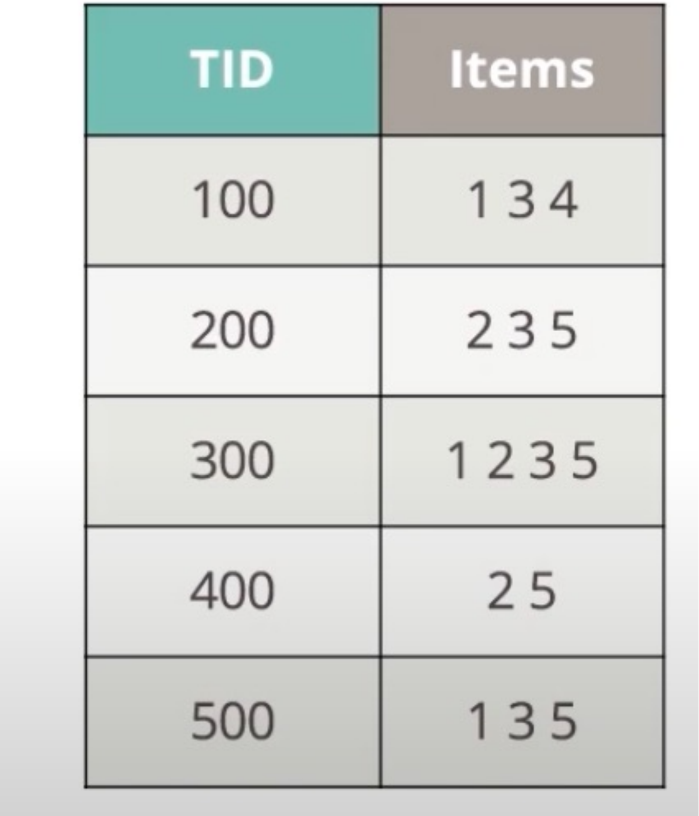
List the F itemsets with k = 2

{1, 3}, {1, 5}, {3,5}, {2,3}, {2, 5} F

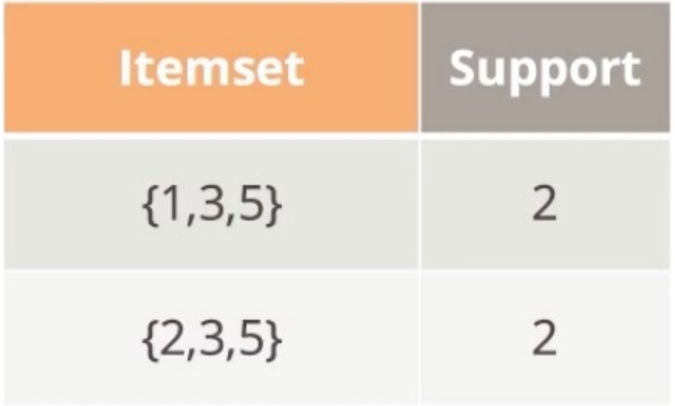
k=1

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

# APRIORI PRINCIPLE EXAMPLE



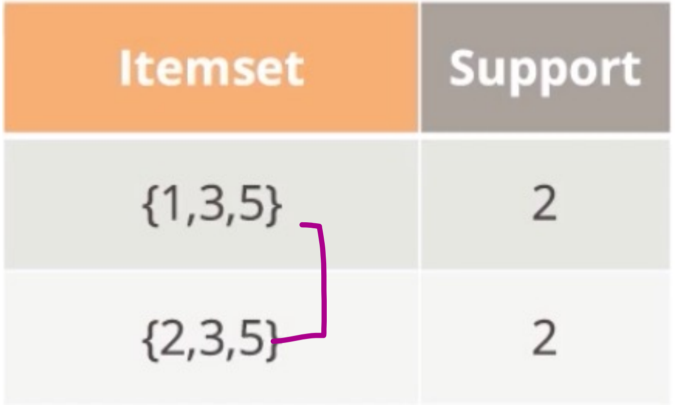
| TID | Items |
|---|---|
| 100 | 1 3 4 |
| 200 | 2 3 5 |
| 300 | 1 2 3 5 |
| 400 | 2 5 |
| 500 | 1 3 5 |



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

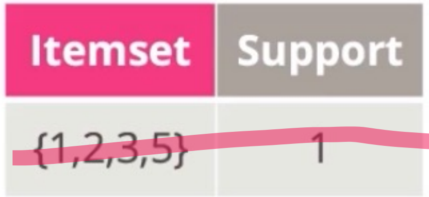
# APRIORI PRINCIPLE EXAMPLE

The length of the itemset is extended with 1 (k = k+1).



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

C4

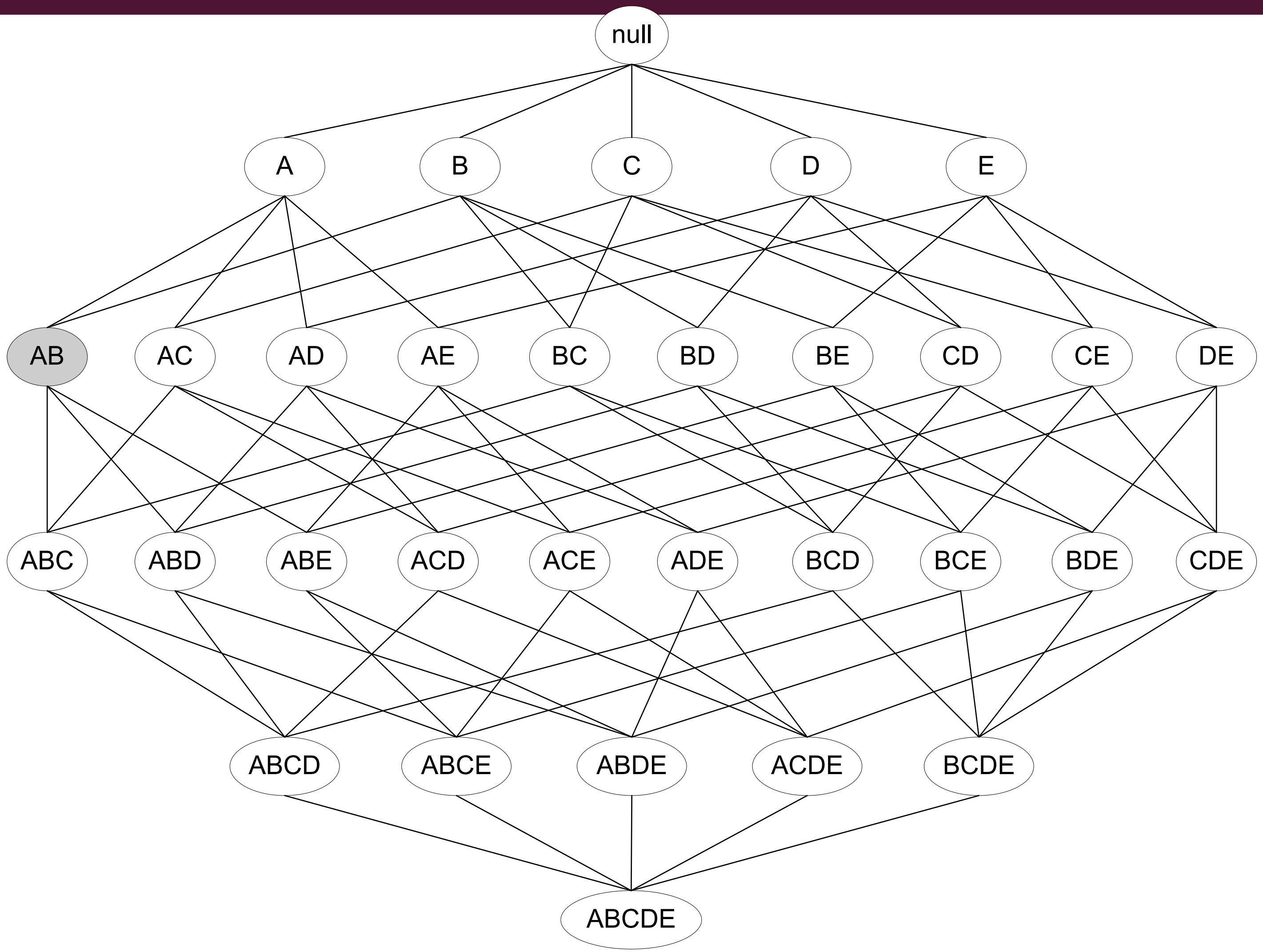


| Itemset | Support |
|---|---|
| ~~{1,2,3,5}~~ | ~~1~~ |

NF

# ILLUSTRATING APRIORI PRINCIPLE

| TID | ITEMS |
|---|---|
| 1 | A, B |
| 2 | A, C, D, E |
| 3 | B, C, D |
| 4 | B, C, E |
| 5 | C, B, D |



# WEKA

- Dataset: https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/contact-lenses.arff