# Association rules

- Vấn đề
- Tổng quan về khai phá luật kết hợp
- Association Rules
- FP-tree

# Vấn đề: Phân tích bán hàng

Nếu mua BIA, liệu xác suất bao nhiêu % sẽ mua thêm Gà

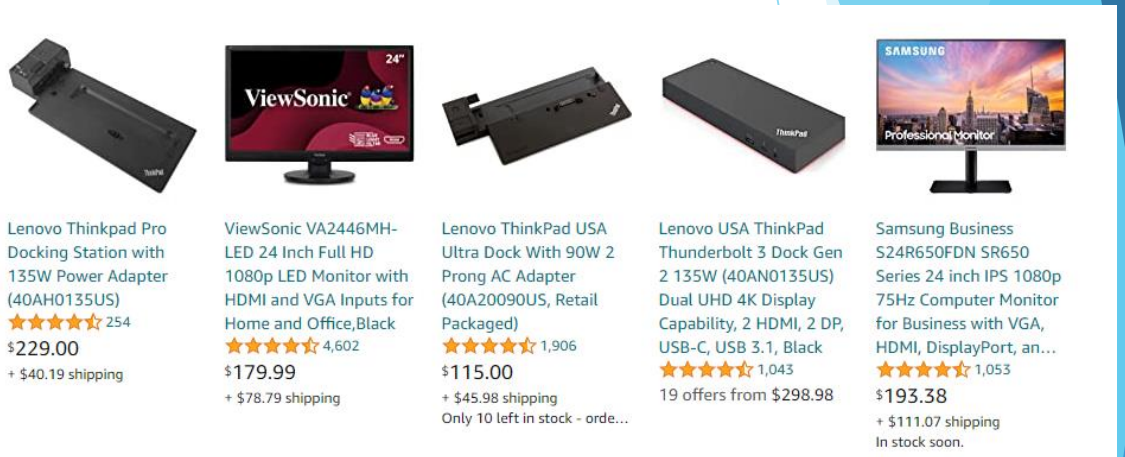


Mua Sữa, Có mua Bỉm không?

Học sinh mua thước kẻ, liệu có mua them bút không

# Vấn đề: Tăng doanh thu

Sẽ Xem




# Khai phá luật kết hợp

- Khai phá luật kế hợp
  - Tìm các mẫu (hàng) xuất hiện nhiều, các liên quan, tương quan, hoặc là các cấu trúc hệ quả trong các tập hợp các mẫu hoặc các đối tượng trong cơ sở dữ liệu (chủ yếu về giao dịch).
- Ứng dụng
  - Phân tích bán hàng (theo gói), tiếp thị chéo (tăng doanh thu), thiết kế các mẫu, phân nhóm, phân cụm.

# Ví dụ về dữ liệu giao dịch (transaction data)

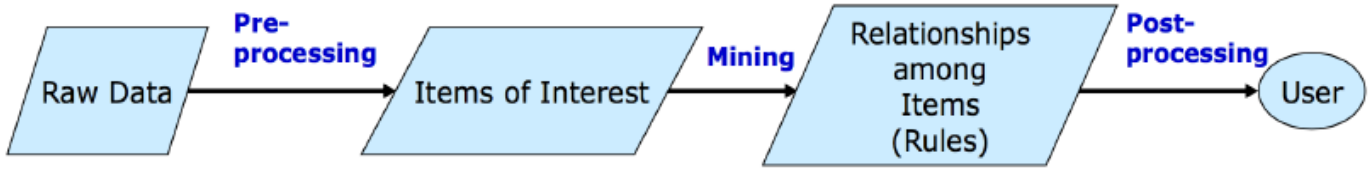
| cardid | cookies | fish | orange juice | lemon tea | red wine | peanuts | beer | chocolate milk | bread | vegetable |
|---|---|---|---|---|---|---|---|---|---|---|
| 39808 | cookies | | | | | | | chocolate milk | bread | |
| 67362 | cookies | | | | | | | | bread | |
| 10872 | | fish | | | red wine | peanuts | beer | | | |
| 26748 | | | | lemon tea | | | | chocolate milk | | |
| 91609 | | | | | | | | | | |
| 26630 | | fish | | lemon tea | | | | | bread | |
| 62995 | | | orange juice | | | | | | | vegetable |
| 38765 | | | | | red wine | | | | | |
| 28935 | | | | | | peanuts | | | | vegetable |
| 41792 | | fish | | | | | | | | vegetable |
| 59480 | | fish | | lemon tea | | | beer | chocolate milk | bread | vegetable |
| 60755 | | fish | | | | | | | | vegetable |
| 70998 | | fish | | | | peanuts | | chocolate milk | | |
| 80617 | | fish | orange juice | | | | | | | |
| 61144 | | fish | orange juice | | | | | chocolate milk | | vegetable |
| 36405 | | | orange juice | | red wine | peanuts | | | | |
| 76567 | | fish | | | | | beer | | | vegetable |
| 85699 | | fish | | | | | | | | |
| 11357 | | fish | | | red wine | | | | | |
| 97761 | cookies | fish | | lemon tea | | | beer | | | vegetable |
| 20362 | | fish | | | | peanuts | | | | |
| 33173 | cookies | | | lemon tea | | | | | | |
| 69934 | | fish | | | | | | | | |
| 14743 | | | | lemon tea | red wine | peanuts | beer | chocolate milk | bread | vegetable |
| 83071 | cookies | | | | red wine | | | | | |
| 17571 | | | | | red wine | peanuts | | chocolate milk | | vegetable |
| 37917 | cookies | | | lemon tea | | | | | bread | |

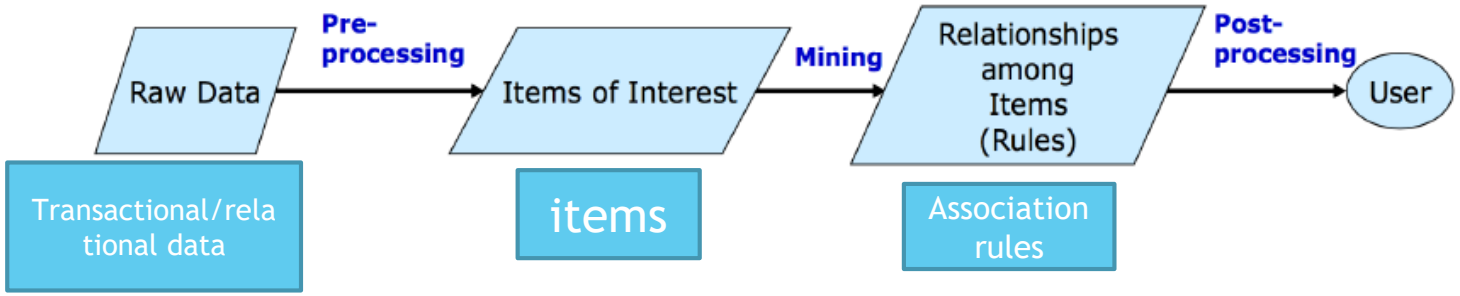
# Mục đích

- Cho một dữ liệu về giao dịch, làm thế nào khám phá được các luật kết hợp mà nó thu hút được về mặt thương mại.
- Các luật kết hợp được sử dụng trong lĩnh vực bán lẻ:
  - Đẩy mảnh quảng cáo Sản phẩm
  - Thay thế sản phẩm
  - Quản lý dữ liệu thực trạng

# Tổng qua về KPLKH

- Quá trình khai phá LKH
  - DỮ liệu gốc (Raw data)
  - Tiền xử lý dữ liệu
  - Tìm được các mẫu, loại, phần tử (được quan tâm nhiều)
  - Khai phá tìm ra các mối quan hệ giữa chúng (gọi là các luật)
  - Sau đấy xử lý để đưa ra các luật, ứng dụng cho người dùng





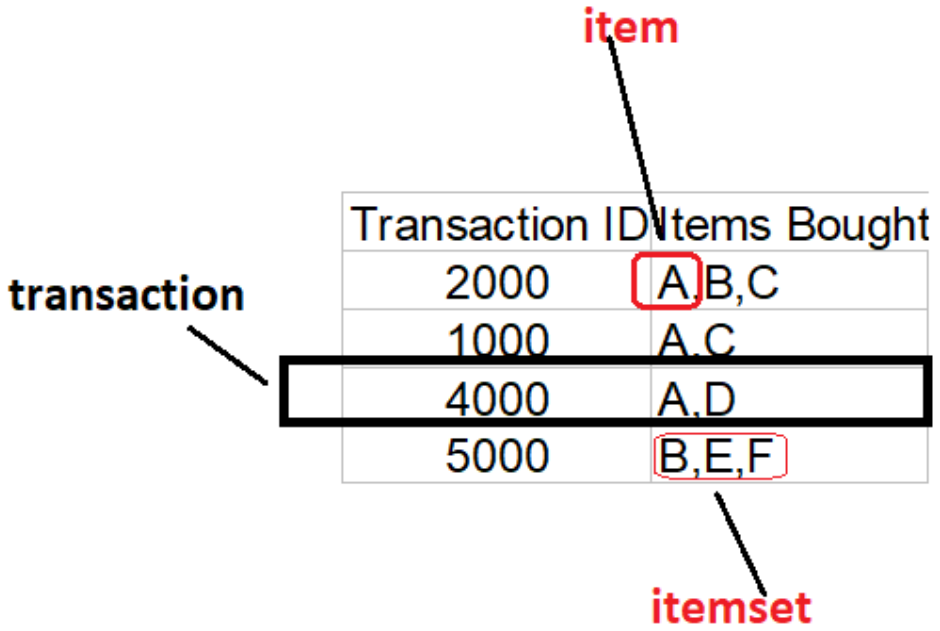

| Transaction ID | Items Bought |
|---|---|
| 2000 | A,B,C |
| 1000 | A,C |
| 4000 | A,D |
| 5000 | B,E,F |

A, B, C, D, E, F

A ->C (50%, 66.6%)

# Một số khái niệm về: Association Rules

- Một số khái niệm cơ bản
  - Item (Phần tử, mẫu, loại)
  - Itemset (tập các phần tử)
  - Transaction (1 giao dịch)
  - Association: Sự kết hợp
  - Association rules: Luật kết hợp
  - Support (độ hỗ trợ)
  - Confidence (độ tin cậy)
  - Frequent itemset (tập phần tử phổ biến – thường xuyên)



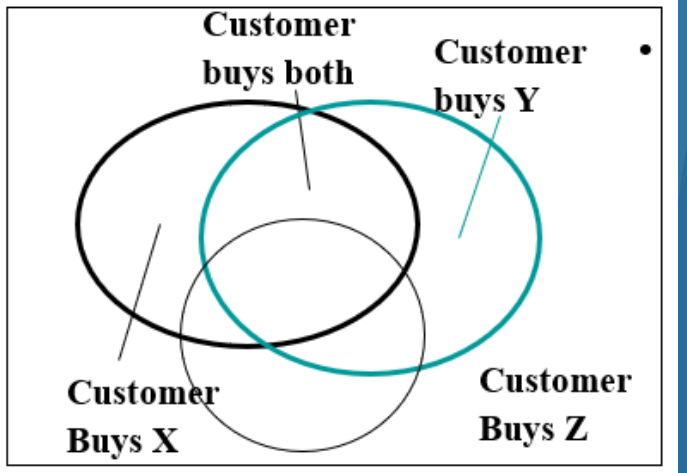
# Một số khái niệm về: Association Rules

- Association (sự kết hợp):
  - Các phần tử cùng xuất hiện với nhau trong một hay nhiều giao dịch.
  - Thể hiện mối liên hệ giữa các phần tử/các tập phần tử
- Luật kết hợp: qui tắc kết hợp có điều kiện giữa các tập phần tử.
  - Thể hiện mối liên hệ (có điều kiện) giữa các tập phần tử
  - Cho A và B là các tập phần tử, luật kết hợp giữa A và B là A → B.
  - Tính khả năng B xuất hiện trong điều kiện A xuất hiện.

Mệnh đề: mua gà

Tiền đề: mua bia

# Một số khái niệm về: Association Rules




- Support, độ hỗ trợ/ủng hộ (“trong bao nhiêu phần trăm dữ liệu thì những điều ở vế trái và vế phải cùng xảy ra")

| Transaction ID | Items Bought |
|---|---|
| 2000 | A,B,C |
| 1000 | A,C |
| 4000 | A,D |
| 5000 | B,E,F |

- $Support\ (I) = \frac{\text{Tổng giao dịch chứa } I}{Tổng\ giao\ dịch\ trong\ dữ\ liệu}$
  - Chú ý: I: có thể là một hoặc nhiều items;
  - Ví dụ **I** = A; hoặc **I** là A và B
- Confidence, độ mạnh (“nếu vế trái xảy ra thì có bao nhiêu khả năng vế phải xảy ra")
- $Confidence\ (X \rightarrow Y) = \frac{Support(X \rightarrow Y)}{Tổng\ giao\ dịch\ chứa\ X}$

# Một số khái niệm về: Association Rules

| Transaction ID | Items Bought |
|---|---|
| 2000 | A,B,C |
| 1000 | A,C |
| 4000 | A,D |
| 5000 | B,E,F |

Min. support 50%
Min. confidence 50%

| Frequent Itemset | Support |
|---|---|
| {A} | 75% |
| {B} | 50% |
| {C} | 50% |
| {A,C} | 50% |

Luật $A \Rightarrow C$:

Suport (A) = ¾ = 0.75 = 75%

Support = support (A ∩*C* ) = *2/4 = ½ = 50%*

confidence = support({*A* ∩ *C*})/support({*A*}) = 66.6%

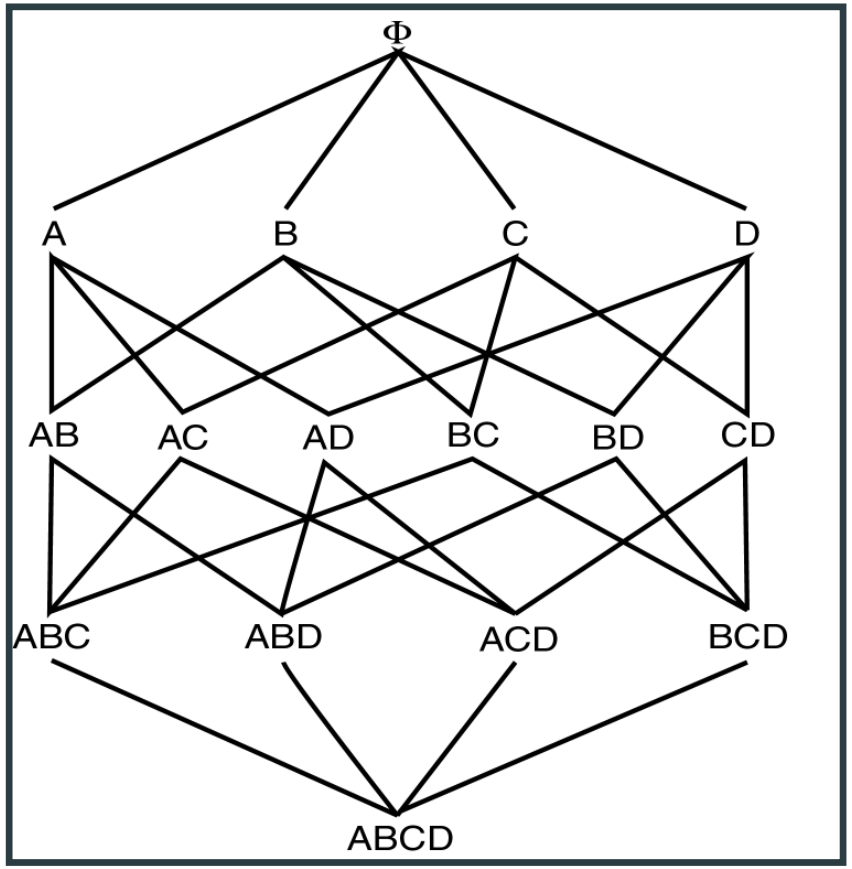
- Vậy luật C -> A; support và confidence?

# Thuật toán Apriori

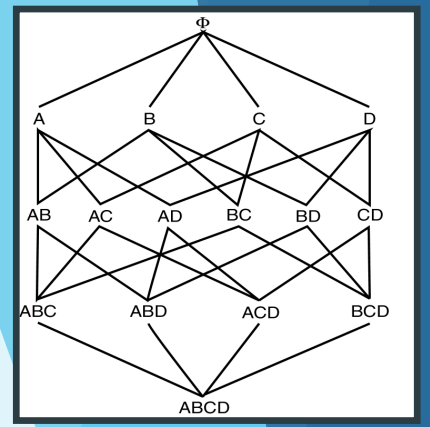
- Giới thiệu lần đầu tiên bởi Agrawal (1993)
- Đây là một trong những thuật toán được triển khai nhiều nhất về khai phá luật kết hợp
- Được cài đặt cùng nhiều phần mềm khai phá dữ liệu
- Code rất sẵn có

# Khai phá: Frequent itemsets

- Tìm một tập các mẫu (loại)  - itemsets: Sao cho chúng có support là bé nhất.
  - Một tập con của một tập mẫu xuất hiện nhiều (frequent itemset) cũng là một tập mẫu xuất hiện nhiều (frequent itemset).
    - Ví dụ: Nếu {A,B} là một tập frequent itemsets thì {A} và {B} cũng là những tập itemsets.
  - Lặp lại tìm kiếm các tập mẫu xuất hiện nhiều lặp lại từ 1->k  (k-itemset)
- Sử dụng tập mẫu này để tạo các luật kết hợp


Φ
A
B
C
D
AB
AC
AD
BC
BD
CD
ABC
ABD
ACD
BCD
ABCD

# Quá trình tạo các itemsets




Database D

| 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

$L_3$

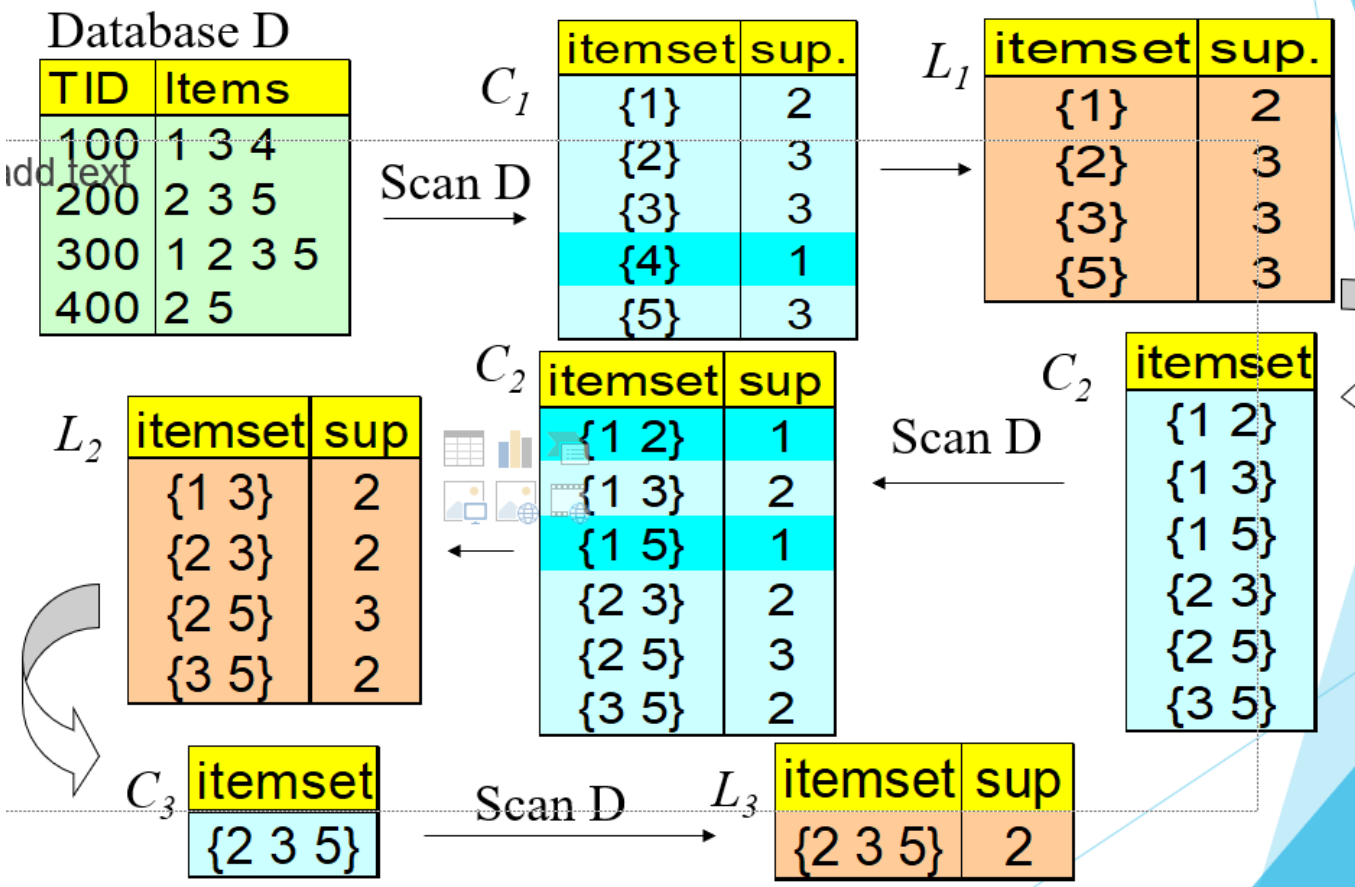
| itemset | sup |
|---|---|
| {2 3 5} | 2 |

# Thuật toán Apriori

- Tạo ra các Ứng viên (itemsets) với kích thước là (i+1) từ một tập các itemsets lớn với size (i)
- Phương pháp sử dụng: Kết hợp các itemsets lớn với kích thước (i+1) nếu chúng cùng với nhau ở trong size (i)
- Đồng thời có thể loại bỏ những ứng viên có các tập con không lớn.

# Thuật toán Apriori: Ý tưởng

1. $C_1$ = Itemsets of size one in I;
2. Determine all large itemsets of size 1, $L_1$;
3. i = 1;
4. Repeat
5. i = i + 1;
6. $C_i$ = Apriori-Gen($L_{i-1}$);
7. Count $C_i$ to determine $L_i$;
8. until no more large itemsets found;



| 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 |

# Example of Apriori

| Transaction | Items |
|---|---|
| $t_1$ | Blouse |
| $t_2$ | Shoes,Skirt,TShirt |
| $t_3$ | Jeans,TShirt |
| $t_4$ | Jeans,Shoes,TShirt |
| $t_5$ | Jeans,Shorts |
| $t_6$ | Shoes,TShirt |
| $t_7$ | Jeans,Skirt |
| $t_8$ | Jeans,Shoes,Shorts,TShirt |
| $t_9$ | Jeans |
| $t_{10}$ | Jeans,Shoes,TShirt |
| $t_{11}$ | TShirt |
| $t_{12}$ | Blouse,Jeans,Shoes,Skirt,TShirt |
| $t_{13}$ | Jeans,Shoes,Shorts,TShirt |
| $t_{14}$ | Shoes,Skirt,TShirt |
| $t_{15}$ | Jeans,TShirt |
| $t_{16}$ | Skirt,TShirt |
| $t_{17}$ | Blouse,Jeans,Skirt |
| $t_{18}$ | Jeans,Shoes,Shorts,TShirt |
| $t_{19}$ | Jeans |
| $t_{20}$ | Jeans,Shoes,Shorts,TShirt |

| Scan | Candidates | Large Itemsets |
|---|---|---|
| 1 | {Blouse},{Jeans},{Shoes},<br>{Shorts},{Skirt},{TShirt} | {Jeans},{Shoes},{Shorts}<br>{Skirt},{Tshirt} |
| 2 | {Jeans,Shoes},{Jeans,Shorts},{Jeans,Skirt},<br>{Jeans,TShirt},{Shoes,Shorts},{Shoes,Skirt},<br>{Shoes,TShirt},{Shorts,Skirt},{Shorts,TShirt},<br>{Skirt,TShirt} | {Jeans,Shoes},{Jeans,Shorts},<br>{Jeans,TShirt},{Shoes,Shorts},<br>{Shoes,TShirt},{Shorts,TShirt},<br>{Skirt,TShirt} |
| 3 | {Jeans,Shoes,Shorts},{Jeans,Shoes,TShirt},<br>{Jeans,Shorts,TShirt},{Jeans,Skirt,TShirt},<br>{Shoes,Shorts,TShirt},{Shoes,Skirt,TShirt},<br>{Shorts,Skirt,TShirt} | {Jeans,Shoes,Shorts},<br>{Jeans,Shoes,TShirt},<br>{Jeans,Shorts,TShirt},<br>{Shoes,Shorts,TShirt} |
| 4 | {Jeans,Shoes,Shorts,TShirt} | {Jeans,Shoes,Shorts,TShirt} |
| 5 | $\emptyset$ | $\emptyset$ |

# Thuật toán Apriori : Pseudo code

**Input:**
$D$ //Database of transactions
$I$ //Items
$L$ //Large itemsets
$s$ //Support
$\alpha$ //Confidence
**Output:**
$R$ //Association Rules satisfying $s$ *and* $\alpha$
**ARGen Algorithm:**
$R = \emptyset$;
**for each** $l \in L$ **do**
  **for each** $x \subset l$ **such that** $x \neq \emptyset$ **and** $x \neq l$ **do**
    **if** $\frac{support(l)}{support(x)} \geq \alpha$ **then**
      $R = R \cup \{x \Rightarrow (l - x)\}$;

- Điểm mạnh
  - Sử dụng các itemsets lớn
  - Dễ triển khai song song
  - Dễ triển khai
- Điểm bất lợi
  - Tốn bộ nhớ lưu trữ
  - Chạy scan lại liên tập tập dữ liệu

# Cải tiến thuật toán Apriori

- Vấn đề của Apriori:
  - Có tới (k-1) itemsets để tạo các ứng viên k-itemsets
  - Sử dụng Dữ liệu tìm và đếm các counts cho các ứng viên
- Vấn đề về tạo các ứng viên
  - 10000 1-itemset (itemset đơn lẻ) sẽ tạo ra $10^7$ các ứng viên 2-itemsets
  - Như vậy để khám phá các mẫu với kích thước 100; ví dụ (a1, …,a100), cần tạo tới $2^{100}$ (xấp xỉ $10^{30}$) ứng viên
  - Scan (quét) database quá nhiều lần.

# Thuật toán FP-Tree (Frequent Pattern Growth Algorithm)

- Nén một dữ liệu lớn thành một tập nhỏ hơn: → FP-Tree structure
- Phương pháp chia để trị: Chia nhỏ ra để thực hiện
- Tránh được việc tạo các ứng viên

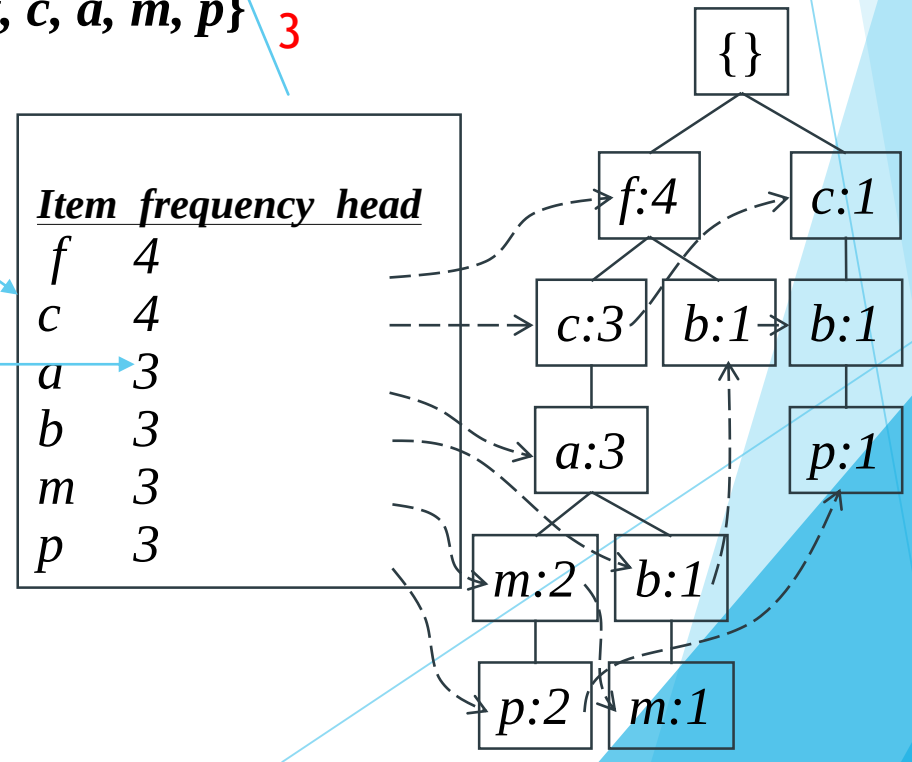
# Thuật toán FP-Tree (Frequent Pattern Growth Algorithm)

| *TID* | *Items* | *(Sắp xếp lại) 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} |

*min_support = 0.5*

**Các bước cơ bản:**

1. Scan DB một lần, tìm ra các 1-itemset (chỉ có một item)
2. Sắp xếp các items này theo thứ tự giảm dần
3. Scan lại DB và xây dựng FP-tree



# Thuật toán FP-Tree (Frequent Pattern Algorithm)

| TID | Items |
|---|---|
| 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} |

Scan DB, đếm các single item (mẫu đơn)

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

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

Với min_sup = 50%.

| Item | Frequency |
|---|---|
| f | 4 |
| c | 4 |
| a | 3 |
| b | 3 |
| m | 3 |
| p | 3 |

- Bước 2:

| Item | Frequency |
|---|---|
| f | 4 |
| c | 4 |
| a | 3 |
| b | 3 |
| m | 3 |
| p | 3 |

| *TID* | *Items* |
|---|---|
| 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*} |

Sắp xếp lại và loại bỏ các item có < sup_min

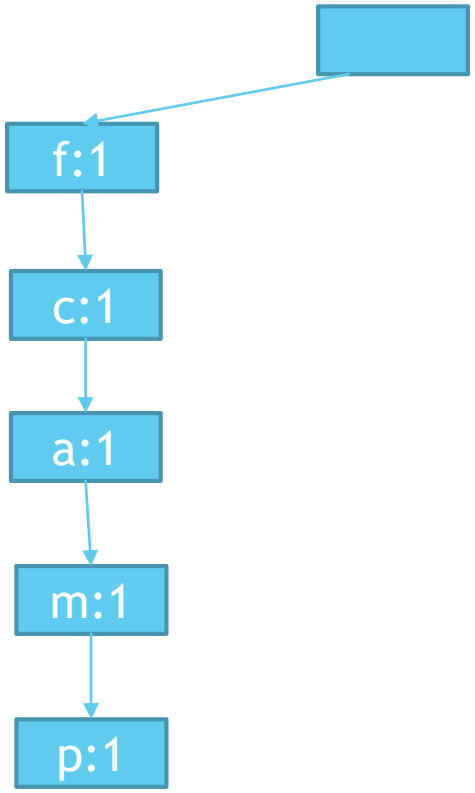
| *(Sắp xếp lại) frequent items* |
|---|
| {*f, c, a, m, p*} |
| {*f, c, a, b, m*} |
| {*f, b*} |
| {*c, b, p*} |
| {*f, c, a, m, p*} |

# Xây dựng FP-Tree

- Tư tưởng: Bắt đầu từ nốt rỗng ∅, xây dựng cây từ DB đã sắp xếp.

***(Sắp xếp lại) frequent items***

~~{f, c, a, m, p}~~
{f, c, a, b, m}
{f, b}
{c, b, p}
{f, c, a, m, p}



***(Sắp xếp lại) frequent items***

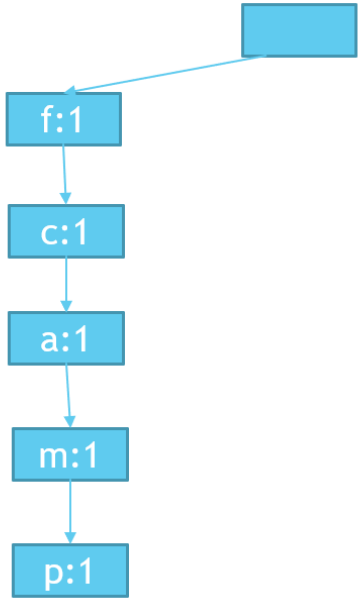
~~{f, c, a, m, p}~~

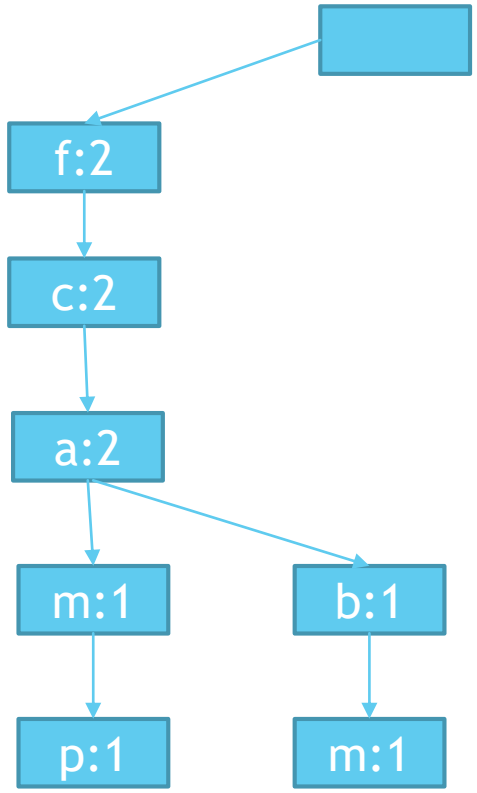
~~{f, c, a, b, m}~~

{f, b}

{c, b, p}

{f, c, a, m, p}



***(Sắp xếp lại) frequent items***

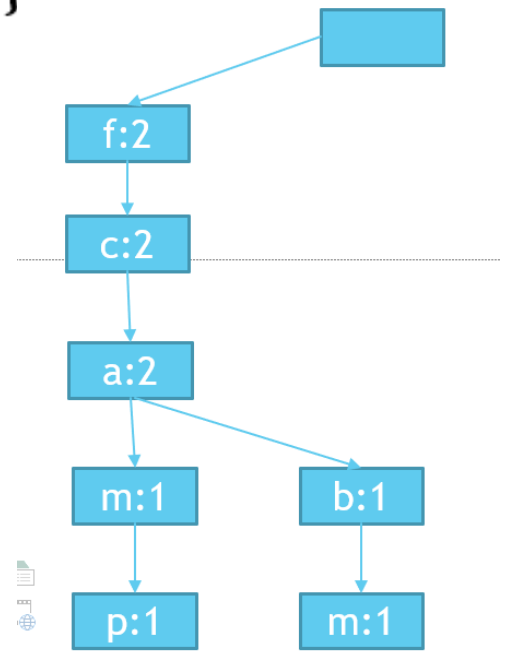
~~{f, c, a, m, p}~~

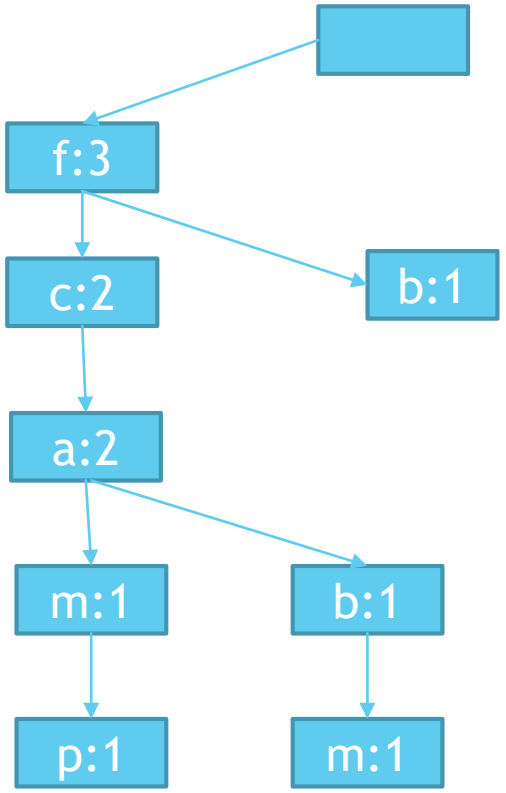
~~{f, c, a, b, m}~~

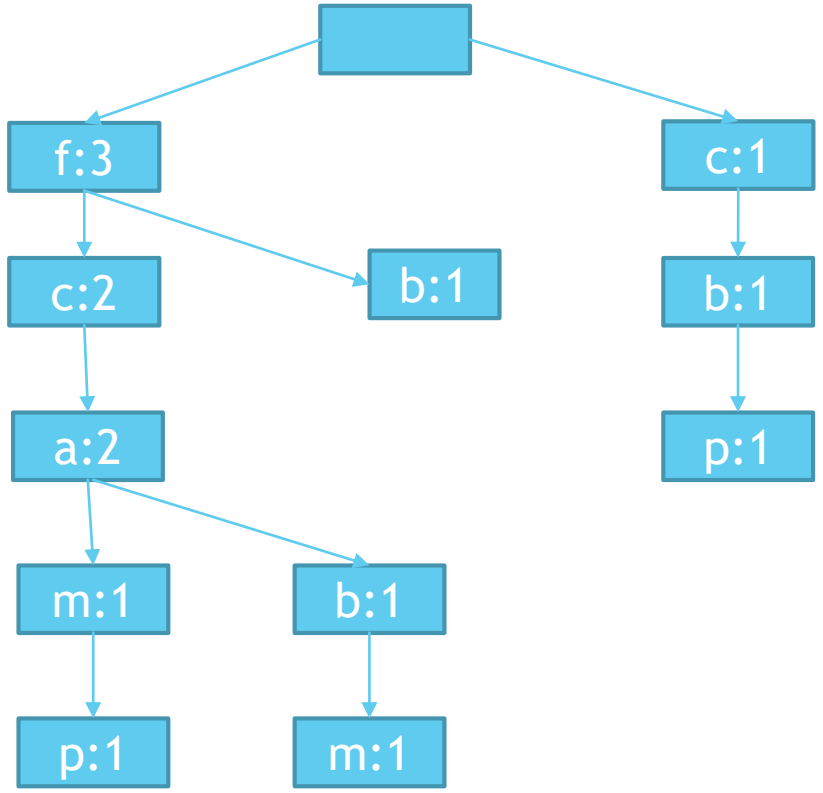
~~{f, b}~~

{c, b, p}

{f, c, a, m, p}






***(Sắp xếp lại) frequent items***

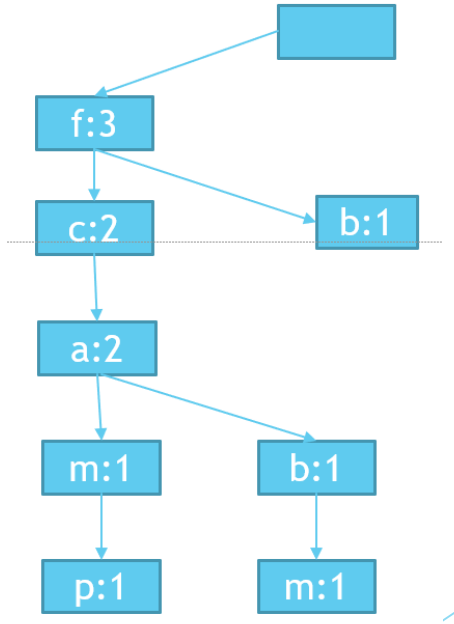
~~{f, c, a, m, p}~~

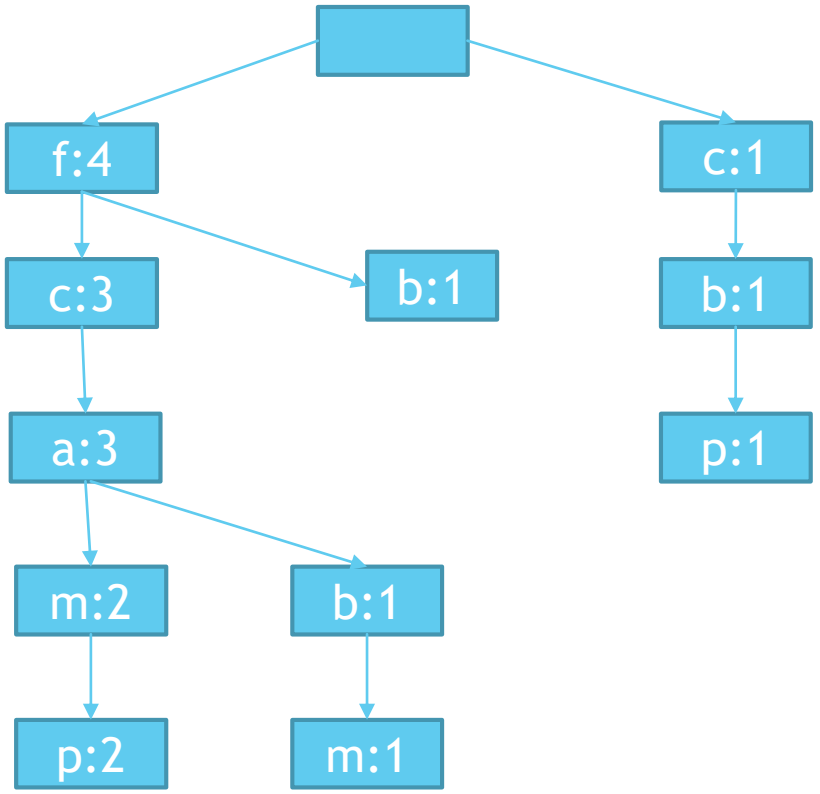
~~{f, c, a, b, m}~~

~~{f, b}~~

~~{c, b, p}~~

~~{f, , , , p}~~






***(Sắp xếp lại) frequent items***

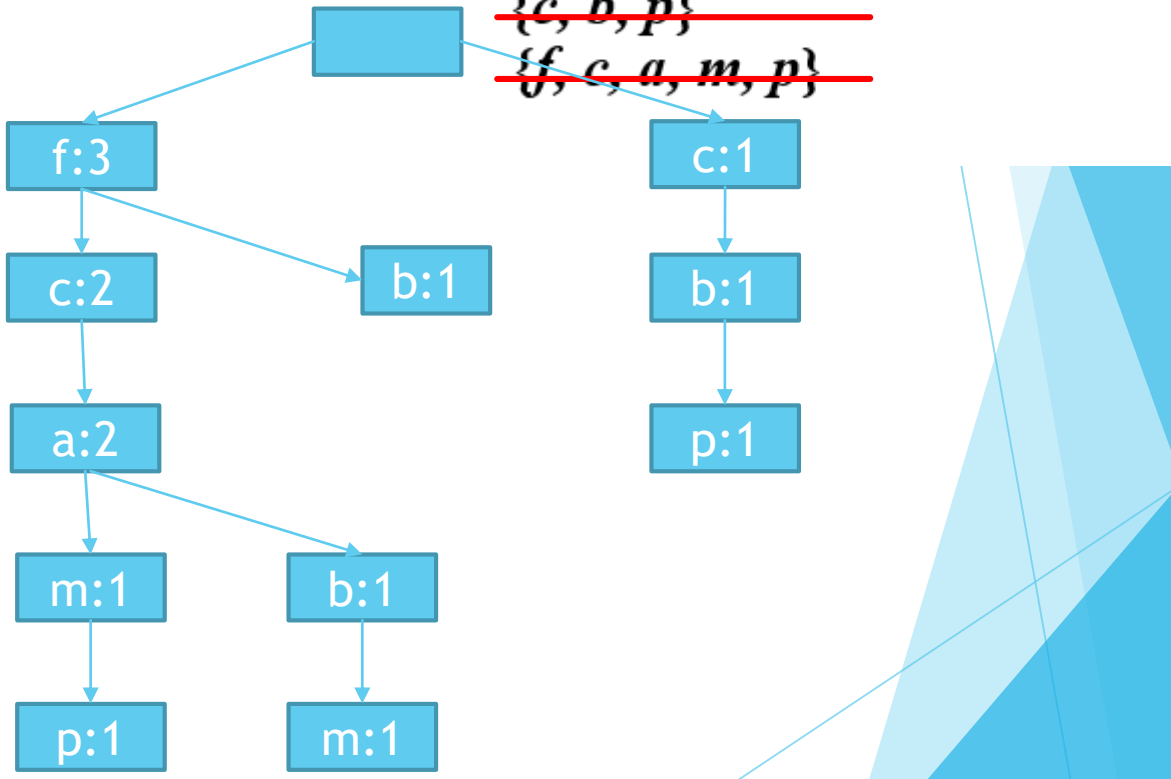
~~{f, c, a, m, p}~~

~~{f, c, a, b, m}~~

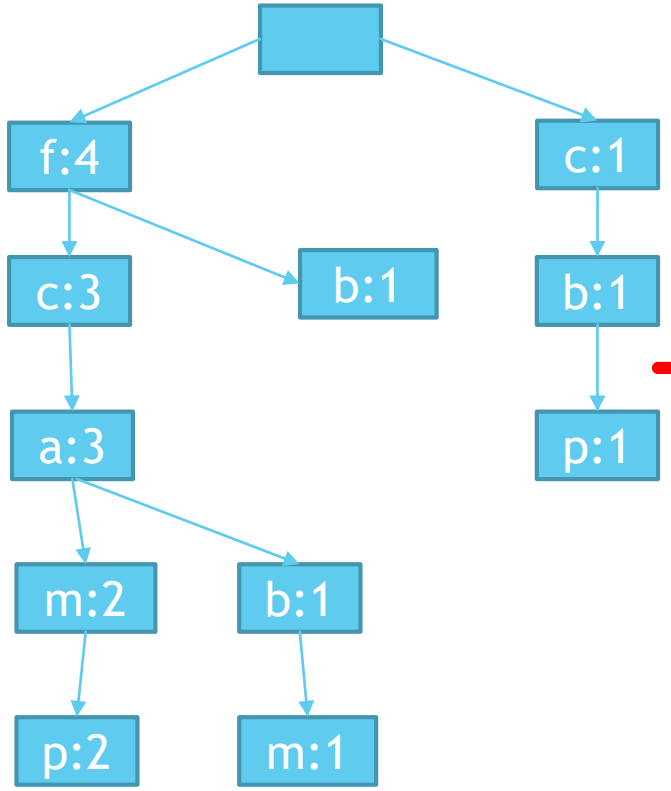
~~{f, b}~~

~~{c, b, p}~~

~~{f, c, a, m, p}~~

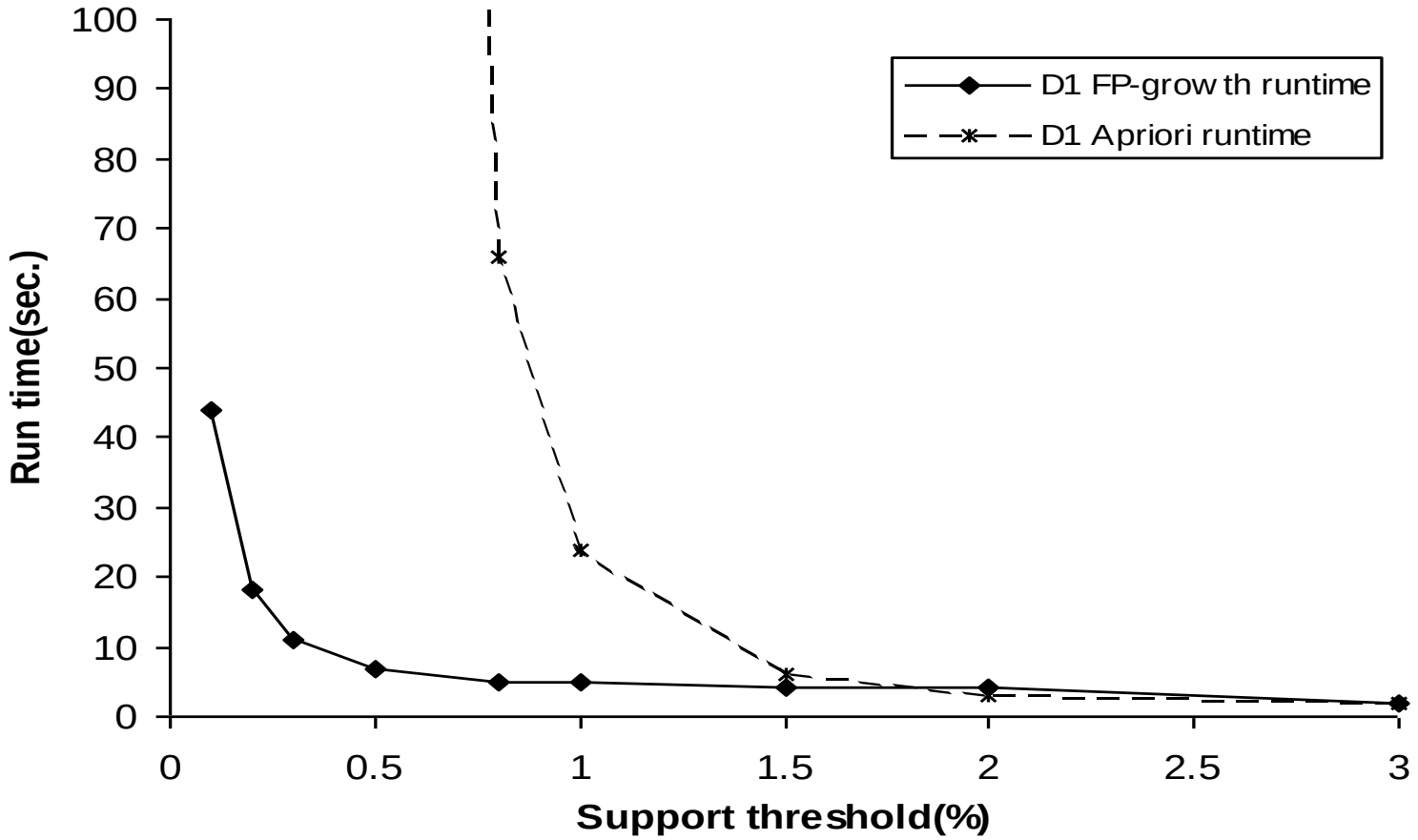


| TID | Items |
|---|---|
| 100 | {f, a, c, d, g, i, m, p} |
| 200 | {a, b, c, f, l, m, o} |
| 300 | {b, f, h, j, o} |
| 400 | {b, c, k, s, p} |
| 500 | {a, f, c, e, l, p, m, n} |




***Itemsets dựa trên các điều kiện***

| | |
|---|---|
| *c* | *f:3* |
| *a* | *fc:3* |
| *b* | *fca:1, f:1, c:1* |
| *m* | *fca:2, fcab:1* |
| *p* | *fcam:2, cb:1* |


100
90
80
70
60
50
40
30
20
10
0
Run time(sec.)
0
0.5
1
1.5
2
2.5
3
Support threshold(%)
D1 FP-grow th runtime
D1 Apriori runtime