

0.1 Itemset Mining

Student-said

Sort items in each individual transaction lexicographically: 1 - {Apple}, {Carrot}, {Flour} 2 - {Apple}, {Carrot}, {Detergent}, {Eggs} 3 - {Apple}, {Carrot}, {Eggs} 4 - {{Apple}, {Detergent}} 5 - {Apple}, {Banana}, {Carrot}, {Flour} 6 - {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

Tutor said:

Here are the transactions with items sorted lexicographically within each transaction:

- 1 - {Apple}, {Carrot}, {Flour}
- 2 - {Apple}, {Carrot}, {Detergent}, {Eggs}
- 3 - {Apple}, {Carrot}, {Eggs}
- 4 - {Apple}, {Detergent}
- 5 - {Apple}, {Banana}, {Carrot}, {Flour}
- 6 - {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

I also corrected the misplaced curly braces in transaction 4. Let me know if you need further adjustments!

Student-said-1

list all unique items in the transactions

Tutor said:

Here are all the unique items from the transactions:

- Apple
- Banana
- Carrot
- Detergent
- Eggs
- Flour

These are the distinct items across all transactions. Student-said-2

give count of each item in all transactions

Tutor said:

Here is the count of each item across all the transactions:

- Apple: 6
- Banana: 1
- Carrot: 6
- Detergent: 3
- Eggs: 3
- Flour: 3

Let me know if you need anything else! Student-said-3

remove all items with a count less than 3 and return the remaining items with the count, as L1

Tutor said:

Here are the remaining items with a count of 3 or more, labeled as L1:

- Apple: 6
- Carrot: 6
- Detergent: 3
- Eggs: 3
- Flour: 3

These are the items that meet the count condition. Student-said-4

Give sets of 2-items with {Apple} from L1 such that "Apple" < X lexicographically

Tutor said:

Here are the 2-item sets with {Apple} from L1, where "Apple" < X lexicographically:

- {Apple, Carrot}
- {Apple, Detergent}
- {Apple, Eggs}
- {Apple, Flour}

Let me know if you'd like me to remember anything from this!

Student-said-5

Give sets of 2-items with {Carrot} from L1 such that "Carrot" < X lexicographically

Tutor said:

Here are the 2-item sets with {Carrot} from L1, where "Carrot" < X lexicographically:

- {Carrot, Detergent}
- {Carrot, Eggs}
- {Carrot, Flour}

Let me know if you need further assistance! Student-said-6

Give sets of 2-items with {Detergent} from L1 such that "Detergent" < X lexicographically

Tutor said:

Here are the 2-item sets with {Detergent} from L1, where "Detergent" < X lexicographically:

- {Detergent, Eggs}
- {Detergent, Flour}

Let me know if you need anything else! Student-said-7

Give sets of 2-items with {Eggs} from L1 such that "Eggs" < X lexicographically

Tutor said:

Here is the 2-item set with {Eggs} from L1, where "Eggs" < X lexicographically:

- {Eggs, Flour}

Let me know if you'd like further assistance! Student-said-8

Give sets of 2-items with {Flour} from L1 such that "Flour" < X lexicographically

Tutor said:

There are no 2-item sets with {Flour} from L1 where "Flour" < X lexicographically, since "Flour" is the last item in L1 alphabetically.

Let me know if you'd like to explore something else! Student-said-9

Combine all these 2-itemsets as C2

Tutor said:

Here are all the 2-item sets combined as C2:

- {Apple, Carrot}
- {Apple, Detergent}
- {Apple, Eggs}
- {Apple, Flour}
- {Carrot, Detergent}
- {Carrot, Eggs}
- {Carrot, Flour}
- {Detergent, Eggs}
- {Detergent, Flour}
- {Eggs, Flour}

This is the combined set C2. Let me know if you'd like further modifications! Student-said-10

give a count of transactions in which each item in a single 2-itemset appears in same transaction

Tutor said:

To determine the count of transactions in which both items of each 2-itemset appear together, I'll go through each set in **C2** and count how many transactions contain both items.

Here are the counts:

- {Apple, Carrot} → 6
- {Apple, Detergent} → 3
- {Apple, Eggs} → 3
- {Apple, Flour} → 3
- {Carrot, Detergent} → 3

- {Carrot, Eggs} \rightarrow 3
- {Carrot, Flour} \rightarrow 3
- {Detergent, Eggs} \rightarrow 3
- {Detergent, Flour} \rightarrow 1
- {Eggs, Flour} \rightarrow 1

Let me know if you need any adjustments!
Student-said-11

{Apple, Carrot} appears in 5 transactions {Detergent, Eggs} appears in 2 transactions
{Carrot, Detergent} appears in 2 transactions

Tutor said:

Thanks for the correction! Here is the revised count of transactions in which each **C2** 2-itemset appears together:

- {Apple, Carrot} \rightarrow 5
- {Apple, Detergent} \rightarrow 3
- {Apple, Eggs} \rightarrow 3
- {Apple, Flour} \rightarrow 3
- {Carrot, Detergent} \rightarrow 2
- {Carrot, Eggs} \rightarrow 3
- {Carrot, Flour} \rightarrow 3
- {Detergent, Eggs} \rightarrow 2
- {Detergent, Flour} \rightarrow 1
- {Eggs, Flour} \rightarrow 1

Let me know if you'd like any further refinements! Student-said-12

remove all items with a count less than 3 and return the remaining items with the count, as L2

Tutor said:

Here are the remaining 2-item sets with a count of **3 or more**, labeled as **L2**:

- {Apple, Carrot} → 5
- {Apple, Detergent} → 3
- {Apple, Eggs} → 3
- {Apple, Flour} → 3
- {Carrot, Eggs} → 3
- {Carrot, Flour} → 3

These are the frequent 2-itemsets that meet the threshold. Let me know if you need any further modifications! Student-said-13

Create C3 from L2 such that $C_k = \{x \mid a \text{ } b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C3** from **L2**, we follow the given rule:

- Combine two sets **a** and **b** from **L2** if their first (k-2) items match and the last item of **a** is lexicographically smaller than the last item of **b**.

Now, applying this rule to **L2**:

12

L2:L2:

1. {Apple, Carrot}
2. {Apple, Detergent}
3. {Apple, Eggs}
4. {Apple, Flour}
5. {Carrot, Eggs}
6. {Carrot, Flour}

c3-generated-by-merging-compatible-l2-sets

C3 (Generated by merging compatible L2 sets):

- **{Apple, Carrot, Eggs}** (from {Apple, Carrot} and {Apple, Eggs})
- **{Apple, Carrot, Flour}** (from {Apple, Carrot} and {Apple, Flour})

These are the valid 3-itemsets in **C3** based on the given conditions. Let me know if you need further refinements! Student-said-14

Why some combinations like {Apple, Carrot, Detergent} were not added?

Tutor said:

You're absolutely right to question that! Let's go step by step to ensure we correctly generate **C3**.

c3-construction-rule-recap

C3 Construction Rule Recap

We generate **C3** by combining two sets **a** and **b** from **L2** if:

1. Their first **(k-2)** items match.
2. The last item of **a** is lexicographically smaller than the last item of **b**.

l2-frequent-2-itemsets

L2 (Frequent 2-itemsets)

We have:

1. {Apple, Carrot}
2. {Apple, Detergent}
3. {Apple, Eggs}
4. {Apple, Flour}

5. {Carrot, Eggs}
6. {Carrot, Flour}

step-by-step-combination-to-form-c3

Step-by-Step Combination to Form C3

combinations-with-apple-x-as-common-items

- {Apple, Carrot} + {Apple, Detergent} → {Apple, Carrot, Detergent}
- {Apple, Carrot} + {Apple, Eggs} → {Apple, Carrot, Eggs}
- {Apple, Carrot} + {Apple, Flour} → {Apple, Carrot, Flour}
- {Apple, Eggs} + {Apple, Flour} → {Apple, Eggs, Flour}

combinations-with-carrot-x-as-common-items

- {Carrot, Eggs} + {Carrot, Flour} → {Carrot, Eggs, Flour}

final-c3-set

Final C3 Set:

- {Apple, Carrot, Detergent}
- {Apple, Carrot, Eggs}
- {Apple, Carrot, Flour}
- {Apple, Eggs, Flour}
- {Carrot, Eggs, Flour}

why-was-apple-carrot-detergent-missing-earlier

Why was {Apple, Carrot, Detergent} missing earlier? Why was {Apple, Carrot, Detergent} missing earlier?

It **should** have been included because {Apple, Carrot} and {Apple, Detergent} exist in **L2**, fulfilling the conditions for merging.

Good catch! Let me know if you need any more clarifications. Student-said-15
remove sets from C3 the subset of which were not present in L2

Tutor said:

Now, let's filter **C3** to remove sets where any **2-item subset** was **not present** in **L2**.

c3-before-filtering

C3 (Before Filtering)

1. {Apple, Carrot, Detergent}
2. {Apple, Carrot, Eggs}
3. {Apple, Carrot, Flour}
4. {Apple, Eggs, Flour}
5. {Carrot, Eggs, Flour}

l2-frequent-2-itemsets-1

L2 (Frequent 2-itemsets)

1. {Apple, Carrot}
2. {Apple, Detergent}
3. {Apple, Eggs}
4. {Apple, Flour}
5. {Carrot, Eggs}

6. {Carrot, Flour}

checking-each-c3-sets-2-item-subsets-against-l2

Checking Each C3 Set's 2-Item Subsets Against L2

1. {Apple, Carrot, Detergent}

- Subsets: {Apple, Carrot}, {Apple, Detergent}, {Carrot, Detergent}
- {Carrot, Detergent} is NOT in L2 → REMOVE

2. {Apple, Carrot, Eggs}

- Subsets: {Apple, Carrot}, {Apple, Eggs}, {Carrot, Eggs}
- All subsets are in L2 → KEEP

3. {Apple, Carrot, Flour}

- Subsets: {Apple, Carrot}, {Apple, Flour}, {Carrot, Flour}
- All subsets are in L2 → KEEP

4. {Apple, Eggs, Flour}

- Subsets: {Apple, Eggs}, {Apple, Flour}, {Eggs, Flour}
- {Eggs, Flour} is NOT in L2 → REMOVE

5. {Carrot, Eggs, Flour}

- Subsets: {Carrot, Eggs}, {Carrot, Flour}, {Eggs, Flour}
- {Eggs, Flour} is NOT in L2 → REMOVE

final-filtered-c3

Final Filtered C3:Final Filtered C3:

- {Apple, Carrot, Eggs}
- {Apple, Carrot, Flour}

These are the 3-itemsets that have all their **2-item subsets present in L2**.

Let me know if you need further refinements! Student-said-16

Find L3 as the count transactions in which each set appears

Tutor said:

To determine **L3**, we need to count the number of transactions in which each set from **C3** appears.

filtered-c3-final-from-last-step

Filtered C3 (Final from Last Step)Filtered C3 (Final from Last Step)

1. {Apple, Carrot, Eggs}
2. {Apple, Carrot, Flour}

counting-transactions-containing-each-set

Counting Transactions Containing Each SetCounting Transactions Containing Each Set

||@||@

() Itemset Transaction Count

() {Apple, Carrot, Eggs} 3

{Apple, Carrot, Flour} 3

()

l3-frequent-3-itemsets-with-count-3

L3 (Frequent 3-itemsets with Count 3) L3 (Frequent 3-itemsets with Count 3)

Since both itemsets appear in at least 3 transactions, **L3** remains the same as the filtered **C3** with their counts:

- {Apple, Carrot, Eggs} → 3
- {Apple, Carrot, Flour} → 3

This is **L3**, the frequent **3-itemsets** that meet the count threshold. Let me know if you need anything else! Student-said-17

Create C4 from L3 such that $C_k = \{x \mid a \subseteq b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C4** from **L3**, we follow the rule:

Combine two sets **a** and **b** from **L3** if:

1. Their first **(k-2)** items match.
2. The last item of **a** is lexicographically smaller than the last item of **b**.

l3-frequent-3-itemsets-with-count-3-1

L3 (Frequent 3-itemsets with Count 3) L3 (Frequent 3-itemsets with Count 3)

1. {Apple, Carrot, Eggs}
2. {Apple, Carrot, Flour}

step-by-step-combination-to-form-c4

Step-by-Step Combination to Form C4

- $\{\text{Apple, Carrot, Eggs}\}$ and $\{\text{Apple, Carrot, Flour}\}$ have **Apple, Carrot** in common.
- Since **Eggs** < **Flour**, they satisfy the condition (**Eggs** < **Flour**).

Thus, we generate:

$\{\text{Apple, Carrot, Eggs, Flour}\}$

final-c4-set

Final C4 Set:

- $\{\text{Apple, Carrot, Eggs, Flour}\}$

This is the candidate **4-itemset (C4)**. Let me know if you need further refinements!

Find L4 as the count of transactions in which each set appears

Tutor said:

Now, let's determine **L4**, which consists of **C4 itemsets that appear in at least 3 transactions**.

c4-candidate-4-itemset

C4 (Candidate 4-itemset)

- $\{\text{Apple, Carrot, Eggs, Flour}\}$

counting-transactions-containing-this-set

Counting Transactions Containing This Set

We check how many transactions contain **{Apple, Carrot, Eggs, Flour}**:

①

() **Itemset Transaction Count**

() **{Apple, Carrot, Eggs, Flour} 1**

()

filtering-for-l4

Filtering for L4

Since **{Apple, Carrot, Eggs, Flour}** appears in **only 1 transaction (<3)**, it does **not** qualify for **L4**.

Thus, **L4 is empty**.

final-l4

Final L4:

No frequent 4-itemsets ($L4 = \emptyset$)

This means the frequent itemset mining process **ends here**, as no larger itemsets meet the minimum transaction threshold.

Let me know if you need further analysis! Student-said-19

Give all frequent itemsets as L1 union L2 union L3 union L4

Tutor said:

final-frequent-itemsets-l1-l2-l3-l4

Final Frequent Itemsets (L1 L2 L3 L4)

Since **L4 is empty**, the union consists of **L1, L2, and L3**.

l1-frequent-1-itemsets

L1 (Frequent 1-itemsets)L1 (Frequent 1-itemsets)

- **Apple** $\rightarrow 6$
- **Carrot** $\rightarrow 6$
- **Detergent** $\rightarrow 3$
- **Eggs** $\rightarrow 3$
- **Flour** $\rightarrow 3$

l2-frequent-2-itemsets-2

L2 (Frequent 2-itemsets)L2 (Frequent 2-itemsets)

- **{Apple, Carrot}** $\rightarrow 5$
- **{Apple, Detergent}** $\rightarrow 3$
- **{Apple, Eggs}** $\rightarrow 3$
- **{Apple, Flour}** $\rightarrow 3$
- **{Carrot, Eggs}** $\rightarrow 3$
- **{Carrot, Flour}** $\rightarrow 3$

l3-frequent-3-itemsets

L3 (Frequent 3-itemsets)L3 (Frequent 3-itemsets)

- **{Apple, Carrot, Eggs}** $\rightarrow 3$
- **{Apple, Carrot, Flour}** $\rightarrow 3$

l4-frequent-4-itemsets

L4 (Frequent 4-itemsets)

No frequent 4-itemsets ($L4 = \emptyset$)

final-frequent-itemsets-l1-l2-l3-l4-1

Final Frequent Itemsets (L1 L2 L3 L4)

- Apple, Carrot, Detergent, Eggs, Flour
- {Apple, Carrot}, {Apple, Detergent}, {Apple, Eggs}, {Apple, Flour}, {Carrot, Eggs}, {Carrot, Flour}
- {Apple, Carrot, Eggs}, {Apple, Carrot, Flour}

This represents all **frequent itemsets** extracted using the Apriori algorithm! Let me know if you need any modifications or further insights. Student-said-20

will the iterations stop at L4?

Tutor said:

Yes, the **iterations will stop at L4** because there are no frequent 4-itemsets ($L4 = \emptyset$).

why-do-the-iterations-stop

Why Do the Iterations Stop?

1. Apriori Property:

- If **L4 is empty**, then **L5, L6, ... will also be empty**, since larger itemsets are formed only from frequent (L4) itemsets.

2. No Valid C5 Candidates:

- The **C5 candidate set cannot be generated** since there are no frequent 4-itemsets in L4.

3. Algorithm Termination Condition:

- The algorithm stops when **Lk is empty**, meaning no frequent k-itemsets exist.

final-conclusion

Final ConclusionFinal Conclusion

Since **L4 = \emptyset** , no **C5** can be generated, and **the iterations stop at L4**.

Thus, the **final set of frequent itemsets is L1 L2 L3**.

prompt-textarea

Let me know if you need more clarifications! composer-background

O

Search

ChatGPT can make mistakes. Check important info.

?

radix-R1jf69t9jaklj5:

0.2 Class Apriori Frequent Itemset

Student-said

Sort items in each individual transaction lexicographically: 1 - {Butter}, {Milk}, {Juice}, {Bread} 2 - {Diapers}, {Bread}, {Milk} 3 - {Milk}, {Juice}, {Bread} 4 - {{Butter}, {Diapers} 5 - {Milk}, {Butter}, {Bread} 6 - {Diapers}, {Butter}, {Milk}, {Juice}, {Bread}

Tutor said:

Here are the items in each transaction sorted lexicographically:

1 - {Bread}, {Butter}, {Juice}, {Milk}

2 - {Bread}, {Diapers}, {Milk}

3 - {Bread}, {Juice}, {Milk}

4 - {Butter}, {Diapers}

5 - {Bread}, {Butter}, {Milk}

6 - {Bread}, {Butter}, {Diapers}, {Juice}, {Milk}

2/2Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-1

list unique items

Tutor said:

The unique items across all transactions are:

{Bread, Butter, Diapers, Juice, Milk}

2/2Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-2

give the number of transactions in which each item appears

2/2

Tutor said:

Here is the number of transactions in which each item appears:

- **Bread** → 5 transactions (1, 2, 3, 5, 6)
- **Butter** → 4 transactions (1, 4, 5, 6)
- **Diapers** → 3 transactions (2, 4, 6)
- **Juice** → 3 transactions (1, 3, 6)
- **Milk** → 5 transactions (1, 2, 3, 5, 6)

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-3

remove any item that has count less than 3 and return the remaining with the count as L1

Tutor said:

After removing items that appear in less than 3 transactions, the remaining items and their counts (L1) are:

- **Bread** → 5
- **Butter** → 4
- **Diapers** → 3
- **Juice** → 3
- **Milk** → 5

So, $L1 = \{ (\text{Bread}, 5), (\text{Butter}, 4), (\text{Diapers}, 3), (\text{Juice}, 3), (\text{Milk}, 5) \}$.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-4

Create C2 from L1 such that $C_k = \{x \mid a \cup b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L(k-1) \text{ and } b \in L(k-1)\}$

Tutor said:

To generate **C2** from **L1**, we follow the given condition:

$$C_k = \{x \mid a \cup b \text{ where } a_1 = b_1, a_2 = b_2, \dots, a_{(k-2)} = b_{(k-1)}, a_{(k-1)} < b_{(k-1)}\}$$

$$C_k = \{x \mid a \cup b \text{ where } a_1=b_1, a_2=b_2, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)}\}$$

Since **L1** contains single items, we create **C2** by forming **all possible pairs** (two-item sets) from L1 in lexicographical order:

l1

L1:L1:

$$\{(Bread, 5), (Butter, 4), (Diapers, 3), (Juice, 3), (Milk, 5)\}$$

$$\{(Bread, 5), (Butter, 4), (Diapers, 3), (Juice, 3), (Milk, 5)\}$$

generating-c2

Generating C2:Generating C2:

- (Bread, Butter)
- (Bread, Diapers)
- (Bread, Juice)
- (Bread, Milk)
- (Butter, Diapers)
- (Butter, Juice)
- (Butter, Milk)
- (Diapers, Juice)
- (Diapers, Milk)
- (Juice, Milk)

final-c2

Final C2:Final C2:

$C2 = \{(Bread, Butter), (Bread, Diapers), (Bread, Juice), (Bread, Milk), (Butter, Diapers), (Butter, Juice),$

$C2 = \{(Bread, Butter), (Bread, Diapers), (Bread, Juice), (Bread, Milk), (Butter, Diapers), (Butter, Juice), (Butter, Milk)\}$
 said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-5

give the transactions in which all items of the 2-itemset appear

2/2

Tutor said:

Now, let's determine the transactions in which both items of each **C2** pair appear:

1. **(Bread, Butter)** → Transactions: **1, 5, 6**
2. **(Bread, Diapers)** → Transactions: **2, 6**
3. **(Bread, Juice)** → Transactions: **1, 3, 6**
4. **(Bread, Milk)** → Transactions: **1, 2, 3, 5, 6**
5. **(Butter, Diapers)** → Transactions: **4, 6**
6. **(Butter, Juice)** → Transactions: **1, 6**
7. **(Butter, Milk)** → Transactions: **1, 5, 6**
8. **(Diapers, Juice)** → Transactions: **6**
9. **(Diapers, Milk)** → Transactions: **2, 6**
10. **(Juice, Milk)** → Transactions: **1, 3, 6**

Now we have each 2-itemset with its corresponding transactions. Let me know if you need further processing!

Student-said-6

give a count of the transactions in which each 2-itemset appear

Tutor said:

Here is the count of transactions in which each **2-itemset** appears:

1. **(Bread, Butter)** → **3** transactions (1, 5, 6)

2. **(Bread, Diapers)** → **2** transactions (2, 6)
3. **(Bread, Juice)** → **3** transactions (1, 3, 6)
4. **(Bread, Milk)** → **5** transactions (1, 2, 3, 5, 6)
5. **(Butter, Diapers)** → **2** transactions (4, 6)
6. **(Butter, Juice)** → **2** transactions (1, 6)
7. **(Butter, Milk)** → **3** transactions (1, 5, 6)
8. **(Diapers, Juice)** → **1** transaction (6)
9. **(Diapers, Milk)** → **2** transactions (2, 6)
10. **(Juice, Milk)** → **3** transactions (1, 3, 6)

Now, you can filter out any itemsets based on a minimum threshold if needed! Let me know what to do next. Tutor said: Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-7

remove the 2-itemsets that have count less than 3 and return the remaining with the count as L2

Tutor said:

After removing the 2-itemsets that appear in less than **3** transactions, the remaining **L2** itemsets with their counts are:

1. **(Bread, Butter)** → **3**
2. **(Bread, Juice)** → **3**
3. **(Bread, Milk)** → **5**
4. **(Butter, Milk)** → **3**
5. **(Juice, Milk)** → **3**

final-l2

Final L2:Final L2:

$$L2 = \{(Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3)\}$$

$$L2 = \{(Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3)\}$$

Let me know if you need the next step!Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-8

Create C3 from L2 such that $C_k = \{x \mid a \cup b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)}<b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C3** from **L2**, we follow the rule:

$$C_k = \{x \mid a \cup b \text{ where } a_1 = b_1, a_2 = b_2, \dots, a_{(k-2)} = b_{(k-1)}, a_{(k-1)} < b_{(k-1)}\}$$

$$C_k = \{x \mid a \cup b \text{ where } a_1=b_1, a_2=b_2, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)}<b_{(k-1)}\}$$

Since **L2** consists of **2-itemsets**, we now generate **3-itemsets** by merging pairs from **L2** that share the first item and maintain lexicographical order.

l2

L2:L2:

$$L2 = \{(Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3)\}$$

$$L2 = \{(Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3)\}$$

generating-c3

Generating C3:Generating C3:

- **(Bread, Butter, Juice)** → Possible from (Bread, Butter) and (Bread, Juice)
- **(Bread, Butter, Milk)** → Possible from (Bread, Butter) and (Bread, Milk)
- **(Bread, Juice, Milk)** → Possible from (Bread, Juice) and (Bread, Milk)

final-c3

Final C3:Final C3:

$$C3 = \{(Bread, Butter, Juice), (Bread, Butter, Milk), (Bread, Juice, Milk)\}$$

$$C3 = \{(Bread, Butter, Juice), (Bread, Butter, Milk), (Bread, Juice, Milk)\}$$

Now, we can proceed to find in how many transactions each **3-itemset** appears. Let me know if you want that next!

Tutor said: Tutor said: Tutor said: Tutor said: Tutor said: Student-said-9

give a count of the transactions in which all items of the 3-itemset appear

Tutor said:

Now, let's determine the count of transactions in which all items of each **C3** 3-itemset appear:

1. **(Bread, Butter, Juice)** → Transactions: **1, 6** → **Count = 2**
2. **(Bread, Butter, Milk)** → Transactions: **1, 5, 6** → **Count = 3**
3. **(Bread, Juice, Milk)** → Transactions: **1, 3, 6** → **Count = 3**

final-count-for-c3

Final Count for C3:Final Count for C3:

$$\{(Bread, Butter, Juice, 2), (Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

$$\{(Bread, Butter, Juice, 2), (Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

Would you like to filter out the 3-itemsets that appear in fewer than 3 transactions and return **L3**?

Tutor said: Tutor said: Tutor said: Tutor said: Tutor said: Student-said-10

remove the 3-itemsets that have count less than 3 and return the remaining with the count as L3

Tutor said:

After removing the **3-itemsets** that appear in fewer than **3** transactions, the remaining **L3** itemsets with their counts are:

1. **(Bread, Butter, Milk)** → **3** transactions
2. **(Bread, Juice, Milk)** → **3** transactions

final-l3

Final L3:Final L3:

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

Let me know if you want to proceed to generate **C4**! Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-11

Create C4 from L3 such that $C_k = \{x \mid a \cup b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L(k-1) \text{ and } b \in L(k-1) \text{ and } L(k-1) \text{ is lexicographically sorted}\}$

2/2

Tutor said:

To generate **C4** from **L3**, we follow the rule:

$$C_k = \{x \mid a \cup b \text{ where } a_1 = b_1, a_2 = b_2, \dots, a_{(k-2)} = b_{(k-1)}, a_{(k-1)} < b_{(k-1)}\}$$

$$C_k = \{x \mid a \cup b \text{ where } a_1=b_1, a_2=b_2, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)}\}$$

Since **L3** consists of **3-itemsets**, we now generate **4-itemsets** by merging pairs from **L3** that share the first **(k-2)** items and maintain lexicographical order.

l3-sorted-lexicographically

L3 (Sorted Lexicographically):L3 (Sorted Lexicographically):

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$$

generating-c4

Generating C4:Generating C4:

- **(Bread, Butter, Juice, Milk)** → Possible from (Bread, Butter, Milk) and (Bread, Juice, Milk), since the first two items (Bread) match, and the last differing items (Butter < Juice) are in lexicographical order.

final-c4

Final C4:Final C4:

$$C4 = \{(Bread, Butter, Juice, Milk)\}$$

$$C4 = \{(Bread, Butter, Juice, Milk)\}$$

Would you like me to find the count of transactions in which this **4-itemset** appears?
 Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-12

Butter != Juice

Tutor said:

You're right! **Butter Juice**, so we cannot merge those **L3** itemsets into a valid **C4** itemset.

Since there are no valid pairs in **L3** that satisfy the given conditions for **C4**, the result is:

final-c4-1

Final C4:Final C4:

$C4 =$

$C4 =$

This means there are no **4-itemsets** that can be formed from **L3** while following the given rules.

Let me know if you need any modifications or further steps! Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-13

Combine all L1, L2 and L3

Tutor said:

Here is the combined **L1, L2, and L3** with their counts:

l1-1-itemsets

L1 (1-itemsets):L1 (1-itemsets):

- (Bread, 5)
- (Butter, 4)
- (Diapers, 3)
- (Juice, 3)
- (Milk, 5)

l2-2-itemsets

L2 (2-itemsets):L2 (2-itemsets):

- (Bread, Butter, 3)
- (Bread, Juice, 3)
- (Bread, Milk, 5)
- (Butter, Milk, 3)
- (Juice, Milk, 3)

l3-3-itemsets

L3 (3-itemsets):L3 (3-itemsets):

- (Bread, Butter, Milk, 3)
- (Bread, Juice, Milk, 3)

final-combined-set-l1-l2-l3

Final Combined Set (L1 L2 L3):Final Combined Set (L1 L2 L3):

$L = \{(Bread, 5), (Butter, 4), (Diapers, 3), (Juice, 3), (Milk, 5), (Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3), (Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$

$L = \{(Bread, 5), (Butter, 4), (Diapers, 3), (Juice, 3), (Milk, 5), (Bread, Butter, 3), (Bread, Juice, 3), (Bread, Milk, 5), (Butter, Milk, 3), (Juice, Milk, 3), (Bread, Butter, Milk, 3), (Bread, Juice, Milk, 3)\}$

This represents the **frequent itemsets** at different levels. Let me know if you need any further processing! Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
 prompt-textarea
 composer-background

0.3 Class Itemset Mining

Student-said

Sort items in each individual transaction lexicographically: 1 - {Milk}, {Butter}, {Bread}, {Eggs} 2 - {Diapers}, {Bread} 3 - {Butter}, {Milk}, {Eggs}, {Flour} 4 - {Flour}, {Eggs}, {Butter} 5 - {Bread}, {Milk}, {Eggs} 6 - {Butter}, {Bread}

Tutor said:

Here are the transactions with items sorted lexicographically within each transaction:

1. **Transaction 1:** {Bread}, {Butter}, {Eggs}, {Milk}
2. **Transaction 2:** {Bread}, {Diapers}
3. **Transaction 3:** {Butter}, {Eggs}, {Flour}, {Milk}
4. **Transaction 4:** {Butter}, {Eggs}, {Flour}
5. **Transaction 5:** {Bread}, {Eggs}, {Milk}
6. **Transaction 6:** {Bread}, {Butter}

Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-1

list all unique items in the transactions

Tutor said:

The unique items appearing across all transactions are:

{Bread, Butter, Diapers, Eggs, Flour, Milk}Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-2

give the count of each time in all transactions

2/2

Tutor said:

Here is the count of each item across all transactions:

- **Bread:** 4 times
- **Butter:** 4 times
- **Diapers:** 1 time
- **Eggs:** 4 times
- **Flour:** 2 times

- **Milk:** 3 times

Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-3

remove all item with count less than 2 and return the remaining item with count as L1
Tutor said:

After removing items with a count less than 2, the remaining items and their counts (L1) are:

L1 = {Bread: 4, Butter: 4, Eggs: 4, Flour: 2, Milk: 3}Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-4

Create C2 from L1 such that $C_k = \{x \mid a \mid b \ \& \ a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C** from **L**, we form all possible **2-item** combinations of items from **L** while maintaining lexicographic order.

lux2081-bread-4-butter-4-eggs-4-flour-2-milk-3

L = {Bread: 4, Butter: 4, Eggs: 4, Flour: 2, Milk: 3}**L = {Bread: 4, Butter: 4, Eggs: 4, Flour: 2, Milk: 3}**

cux2082-candidate-2-itemsets

- {Bread, Butter}
- {Bread, Eggs}
- {Bread, Flour}
- {Bread, Milk}
- {Butter, Eggs}
- {Butter, Flour}
- {Butter, Milk}
- {Eggs, Flour}
- {Eggs, Milk}
- {Flour, Milk}

Thus, $C =$
 $\{\{\text{Bread, Butter}\}, \{\text{Bread, Eggs}\}, \{\text{Bread, Flour}\}, \{\text{Bread, Milk}\}, \{\text{Butter, Eggs}\}, \{\text{Butter, Flour}\}, \{\text{Butter, Milk}\}, \{\text{Eggs, Flour}\}, \{\text{Eggs, Milk}\}, \{\text{Flour, Milk}\}\}$ Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-5

give a count of transactions in which each item in a single set appears

Tutor said:

To count the number of transactions in which each **2-itemset** from C appears, let's check their occurrences in the given transactions:

transactions-after-lexicographic-sorting

Transactions (after lexicographic sorting) Transactions (after lexicographic sorting)

1. {Bread, Butter, Eggs, Milk}
2. {Bread, Diapers}
3. {Butter, Eggs, Flour, Milk}
4. {Butter, Eggs, Flour}
5. {Bread, Eggs, Milk}
6. {Bread, Butter}

count-of-each-2-itemset-in-transactions

Count of each 2-itemset in transactions Count of each 2-itemset in transactions

@ @	
() 2-Itemset	Count
() {Bread, Butter}	2
{Bread, Eggs}	2
{Bread, Flour}	0
{Bread, Milk}	2
{Butter, Eggs}	3
{Butter, Flour}	2
{Butter, Milk}	2
{Eggs, Flour}	2

{Eggs, Milk} 2
 {Flour, Milk} 1

()

final-counted-set-cux2082-with-counts

Final Counted Set (C with counts)Final Counted Set (C with counts)

C = { (Bread, Butter): 2, (Bread, Eggs): 2, (Bread, Flour): 0, (Bread, Milk): 2, (Butter, Eggs): 3, (Butter, Flour): 2, (Butter, Milk): 2, (Eggs, Flour): 2, (Eggs, Milk): 2, (Flour, Milk): 1 }Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-6

remove all with a count less than 2 and return the remaining 2-itemset with a count as

L2

Tutor said:

After removing 2-itemsets with a count less than 2, the remaining **2-itemsets with counts (L)** are:

L = { (Bread, Butter): 2, (Bread, Eggs): 2, (Bread, Milk): 2, (Butter, Eggs): 3, (Butter, Flour): 2, (Butter, Milk): 2, (Eggs, Flour): 2, (Eggs, Milk): 2 }Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-7

Create C3 from L2 such that $C_k = \{x | a \leq b \text{ \& } a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C** from **L**, we form all possible **3-item** combinations where the first **k-2** items are the same, and the last item is in lexicographic order.

lux2082

L = {**L** = {

- (Bread, Butter): 2
- (Bread, Eggs): 2
- (Bread, Milk): 2
- (Butter, Eggs): 3
- (Butter, Flour): 2
- (Butter, Milk): 2

- (Eggs, Flour): 2
- (Eggs, Milk): 2
- }

generating-cux2083-candidate-3-itemsets

Generating C (Candidate 3-itemsets)Generating C (Candidate 3-itemsets)

We combine pairs from **L** that share the first (**k-2**) = **1** elements and maintain lexicographic order:

1. {Bread, Butter, Eggs} → (From {Bread, Butter} {Bread, Eggs})
2. {Bread, Butter, Milk} → (From {Bread, Butter} {Bread, Milk})
3. {Bread, Eggs, Milk} → (From {Bread, Eggs} {Bread, Milk})
4. {Butter, Eggs, Flour} → (From {Butter, Eggs} {Butter, Flour})
5. {Butter, Eggs, Milk} → (From {Butter, Eggs} {Butter, Milk})
6. {Eggs, Flour, Milk} → (From {Eggs, Flour} {Eggs, Milk})

final-candidate-3-itemsets-cux2083

Final Candidate 3-itemsets (C)Final Candidate 3-itemsets (C)

C = { (Bread, Butter, Eggs), (Bread, Butter, Milk), (Bread, Eggs, Milk), (Butter, Eggs, Flour), (Butter, Eggs, Milk), (Eggs, Flour, Milk) }
Tutor said:Tutor said:Tutor said:

Student-said-8

give a count of transactions in which each item in a single 3-itemset appears

Tutor said:

To count the number of transactions in which each **3-itemset** from **C** appears, let's check their occurrences in the given transactions.

transactions-after-lexicographic-sorting-1

Transactions (after lexicographic sorting)Transactions (after lexicographic sorting)

1. {Bread, Butter, Eggs, Milk}
2. {Bread, Diapers}

3. {Butter, Eggs, Flour, Milk}
4. {Butter, Eggs, Flour}
5. {Bread, Eggs, Milk}
6. {Bread, Butter}

count-of-each-3-itemset-in-transactions

Count of each 3-itemset in transactions

@ @	
() 3-Itemset	Count
() {Bread, Butter, Eggs}	1
{Bread, Butter, Milk}	1
{Bread, Eggs, Milk}	2
{Butter, Eggs, Flour}	2
{Butter, Eggs, Milk}	2
{Eggs, Flour, Milk}	1

final-counted-set-cux2083-with-counts

Final Counted Set (C with counts)

$C = \{ (\text{Bread, Butter, Eggs}): 1, (\text{Bread, Butter, Milk}): 1, (\text{Bread, Eggs, Milk}): 2, (\text{Butter, Eggs, Flour}): 2, (\text{Butter, Eggs, Milk}): 2, (\text{Eggs, Flour, Milk}): 1 \}$ Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-9

remove all with a count less than 2 and return the remaining 3-itemset with a count as L3

Tutor said:

After removing 3-itemsets with a count less than 2, the remaining **3-itemsets with counts (L)** are:

$L = \{ (\text{Bread, Eggs, Milk}): 2, (\text{Butter, Eggs, Flour}): 2, (\text{Butter, Eggs, Milk}): 2 \}$ Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-10

Create C4 from L3 such that $C_k = \{x \mid a \in b \ \& \ a_1=b_1, a_2=b_2, a_3=b_3, \dots, a_{(k-2)}=b_{(k-1)}, a_{(k-1)} < b_{(k-1)} \text{ where } a \in L_{(k-1)} \text{ and } b \in L_{(k-1)}\}$

Tutor said:

To generate **C** from **L**, we form all possible **4-item** combinations where the first **(k-2) = 2** elements are the same, and the last item is in lexicographic order.

lux2083

L = {**L** = {

- (**Bread, Eggs, Milk**): **2**
- (**Butter, Eggs, Flour**): **2**
- (**Butter, Eggs, Milk**): **2**
- }

generating-cux2084-candidate-4-itemsets

Generating C (Candidate 4-itemsets)Generating C (Candidate 4-itemsets)

We combine pairs from **L** that share the first **(k-2) = 2** elements and maintain lexicographic order:

1. {**Butter, Eggs, Flour, Milk**} → (From {**Butter, Eggs, Flour**} {**Butter, Eggs, Milk**})

final-candidate-4-itemsets-cux2084

Final Candidate 4-itemsets (C)Final Candidate 4-itemsets (C)

C = { (**Butter, Eggs, Flour, Milk**) }Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-11

give a count of transactions in which each item in a single 4-itemset appears

Tutor said:

To count the number of transactions in which the **4-itemset** from **C** appears, let's check its occurrences in the given transactions.

transactions-after-lexicographic-sorting-2

Transactions (after lexicographic sorting)Transactions (after lexicographic sorting)

1. {**Bread, Butter, Eggs, Milk**}
2. {**Bread, Diapers**}
3. {**Butter, Eggs, Flour, Milk**}

4. {Butter, Eggs, Flour}

5. {Bread, Eggs, Milk}

6. {Bread, Butter}

checking-butter-eggs-flour-milk-in-transactions

Checking {Butter, Eggs, Flour, Milk} in transactions
Checking {Butter, Eggs, Flour, Milk} in transactions

- Appears in **Transaction 3**
- Does NOT appear in **Transaction 1, 2, 4, 5, 6**

final-counted-set-cux2084-with-counts

Final Counted Set (C with counts)Final Counted Set (C with counts)

C = { (Butter, Eggs, Flour, Milk): 1 }Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-12

remove all with a count less than 2 and return the remaining 4-itemset with a count as
L4

Tutor said:

Since the only **4-itemset (Butter, Eggs, Flour, Milk)** has a count of **1**, which is less than **2**, it is removed.

Thus, **L is empty**:

L = Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-13

Give all frequent itemsets as a union of L1, L2, L3, andL4

Tutor said:

The **frequent itemsets** are the union of **L, L, L, and L**. Since **L is empty**, the final frequent itemsets are:

frequent-itemsets-lux2081-lux2082-lux2083

Frequent Itemsets = L L LFrequent Itemsets = L L L

lux2081-1-itemsets

- {Bread}: 4

- {Butter}: 4
- {Eggs}: 4
- {Flour}: 2
- {Milk}: 3

lux2082-2-itemsets

- {Bread, Butter}: 2
- {Bread, Eggs}: 2
- {Bread, Milk}: 2
- {Butter, Eggs}: 3
- {Butter, Flour}: 2
- {Butter, Milk}: 2
- {Eggs, Flour}: 2
- {Eggs, Milk}: 2

lux2083-3-itemsets

- {Bread, Eggs, Milk}: 2
- {Butter, Eggs, Flour}: 2
- {Butter, Eggs, Milk}: 2

lux2084-4-itemsets

- **Empty** (no frequent 4-itemsets)

final-frequent-itemsets

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

Unique Items:

Beer, Bread, Butter, Diapers, MilkTutor said:Tutor said:Tutor said:Tutor said:
Student-said-2

In given transactions, give the number of transactions in which each item appears. It should be in the form of a set of items with their count lexicographically sorted by item name.

Tutor said:

Let's calculate the count of transactions for each item:

Transactions:

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

Item Counts:

- **Beer:** Appears in transactions 4, 5 → **2 transactions**
- **Bread:** Appears in transactions 1, 2, 3, 4, 5, 6 → **6 transactions**
- **Butter:** Appears in transactions 1, 2, 4, 5, 6 → **5 transactions**
- **Diapers:** Appears in transactions 3, 5, 6 → **3 transactions**

- **Milk:** Appears in transactions 1, 3, 5, 6 → **4 transactions**

Result (sorted lexicographically):

{(Beer, 2), (Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)} Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-3

So {(Beer, 2), (Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)} is C1. Give $L1 = C1 - \{\text{item count} < \text{minsup}\}$ Here $\text{minsup} = 3$

Tutor said:

To compute $L1L1$, we remove items from $C1C1$ where the count is less than the minimum support ($\text{minsup} = 3$).

Given $C1C1$:

$\{(Beer, 2), (Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$

{(Beer,2),(Bread,6),(Butter,5),(Diapers,3),(Milk,4)}

Items with count $\geq \text{minsup}$:

- Bread (6)
- Butter (5)
- Diapers (3)
- Milk (4)

Result ($L1L1$):

$L1 = \{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$

$L1 = \{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$ Tutor said: Tutor said: Tutor said: Tutor said: Student-said-4

Using $L1$ give $C2$ using this formula: Insert into C_k SELECT p.item_1, p.item_2, ..., p.item_{k-1}, p.item_k FROM L_{k-1} p, L_{k-1} q WHERE p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}

Tutor said:

To compute $C2C2$ from $L1L1$, we use the formula provided:
given-l1l1

Given $L1$:Given $L1$:

$$L1 = \{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$$

$$L1 = \{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$$

This means the items in $L1$ are:

$$\{Bread, Butter, Diapers, Milk\}$$

$$\{Bread, Butter, Diapers, Milk\}$$

steps-to-generate-c2c2

Steps to Generate $C2$:Steps to Generate $C2$:

We pair all items from $L1$ such that the first $k - 2$ items are equal (for $k = 2$, there are no common prefixes), and the last item in pp is less than the last item in qq .

item-pairs-for-c2c2

Item Pairs for $C2$:Item Pairs for $C2$:

- Bread < Butter $\rightarrow (Bread, Butter)(Bread, Butter)$
- Bread < Diapers $\rightarrow (Bread, Diapers)(Bread, Diapers)$
- Bread < Milk $\rightarrow (Bread, Milk)(Bread, Milk)$
- Butter < Diapers $\rightarrow (Butter, Diapers)(Butter, Diapers)$
- Butter < Milk $\rightarrow (Butter, Milk)(Butter, Milk)$
- Diapers < Milk $\rightarrow (Diapers, Milk)(Diapers, Milk)$

result-c2c2

Result $C2$:Result $C2$:

$$C2 = \{(Bread, Butter), (Bread, Diapers), (Bread, Milk), (Butter, Diapers), (Butter, Milk), (Diapers, Milk)\}$$

$$C2 = \{(Bread, Butter), (Bread, Diapers), (Bread, Milk), (Butter, Diapers), (Butter, Milk), (Diapers, Milk)\}$$

Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-5

Update $C2$ with the count of each item from transactions

Tutor said:

To update $C2$ with the count of each pair, we check each pair's occurrences in the given transactions.

given-transactions-sorted

Given Transactions (sorted):

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

pairs-in-c2c2

Pairs in C2C2:

1. **(Bread, Butter):**
Appears in transactions 1, 2, 4, 5, 6 → **5 transactions**
2. **(Bread, Diapers):**
Appears in transactions 3, 5, 6 → **3 transactions**
3. **(Bread, Milk):**
Appears in transactions 1, 3, 5, 6 → **4 transactions**
4. **(Butter, Diapers):**
Appears in transactions 5, 6 → **2 transactions**
5. **(Butter, Milk):**
Appears in transactions 1, 5, 6 → **3 transactions**
6. **(Diapers, Milk):**
Appears in transactions 3, 5, 6 → **3 transactions**

updated-c2c2

Updated C2C2:

$$C2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Diapers, 2), (Butter, Milk, 3), (Diapers, Milk, 3)\}$$

$C2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Diapers, 2), (Butter, Milk, 3), (Diapers, Milk, 3)\}$
 said:Tutor said:Tutor said:Tutor said:

Student-said-6

Using same minsup = 3 Give $L2 = C2 - \{item\ count < minsup\}$

Tutor said:

To compute $L2$, we remove pairs from $C2$ where the count is less than the minimum support (minsup = 3).
 given-c2

Given $C2$:

$C2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Diapers, 2), (Butter, Milk, 3), (Diapers, Milk, 3)\}$

$C2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Diapers, 2), (Butter, Milk, 3), (Diapers, Milk, 3)\}$
 filter-pairs-with-count-geq-textminsup

Filter Pairs with Count \geq minsup:

1. **(Bread, Butter, 5):** Count 5 \geq 3 \rightarrow **Include**
2. **(Bread, Diapers, 3):** Count 3 \geq 3 \rightarrow **Include**
3. **(Bread, Milk, 4):** Count 4 \geq 3 \rightarrow **Include**
4. **(Butter, Diapers, 2):** Count 2 $<$ 3 \rightarrow **Exclude**
5. **(Butter, Milk, 3):** Count 3 \geq 3 \rightarrow **Include**
6. **(Diapers, Milk, 3):** Count 3 \geq 3 \rightarrow **Include**

result-l2

Result $L2$:

$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$

$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$
 Tutor said:Tutor said:Tutor said:

Student-said-7

Using $L2$ give $C3$ using this formula: Insert into C_k SELECT p.item₁, p.item₂, ..., p.item_{k-1}, p.item_k FROM L_{k-1} p, L_{k-1} q WHERE p.item₁ = q.item₁, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}

Tutor said:

To generate $C3C3$ from $L2L2$, we use the formula provided:
 given-l2l2

Given $L2L2$:Given $L2L2$:

$$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$$

$$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$$

The items for pairing are:

$$\{Bread, Butter, Diapers, Milk\}$$

$$\{Bread, Butter, Diapers, Milk\}$$

steps-to-generate-c3c3

Steps to Generate $C3C3$:Steps to Generate $C3C3$:

We combine pairs from $L2L2$ where the first $k - 2 = 1k2=1$ items (prefix) match, and
 $p.item_{k-1} < q.item_{k-1}$ to form itemsets of size 3.

formed-triples-for-c3c3

Formed Triples for $C3C3$:Formed Triples for $C3C3$:**1. From (Bread, Butter):**

- Matches with (Bread, Diapers) $\rightarrow (Bread, Butter, Diapers)$ (Bread,Butter,Diapers)
- Matches with (Bread, Milk) $\rightarrow (Bread, Butter, Milk)$ (Bread,Butter,Milk)

2. From (Bread, Diapers):

- Matches with (Bread, Milk) $\rightarrow (Bread, Diapers, Milk)$ (Bread,Diapers,Milk)

3. From (Butter, Milk):

- Does not match with any other pair.

4. From (Diapers, Milk):

- Does not match with any other pair.

result-c3c3

Result C3C3:Result C3C3:

$$C3 = \{(Bread, Butter, Diapers), (Bread, Butter, Milk), (Bread, Diapers, Milk)\}$$

C3={ (Bread,Butter,Diapers),(Bread,Butter,Milk),(Bread,Diapers,Milk) } Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-8

In C3, check if all possible 2-item combination of each entry exists in L2. Eliminate the entry that has some 2-item combination that does not exist in L2. It is called pruning

Tutor said:

To perform pruning on C3C3, we ensure that all possible 2-item subsets of each 3-item combination in C3C3 exist in L2L2. If any 2-item subset of a 3-item combination does not exist in L2L2, we eliminate that 3-item combination.

given-c3c3

Given C3C3:Given C3C3:

$$C3 = \{(Bread, Butter, Diapers), (Bread, Butter, Milk), (Bread, Diapers, Milk)\}$$

$$C3=\{(Bread,Butter,Diapers),(Bread,Butter,Milk),(Bread,Diapers,Milk)\}$$

given-l2l2-1

Given L2L2:Given L2L2:

$$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$$

$$L2=\{(Bread,Butter,5),(Bread,Diapers,3),(Bread,Milk,4),(Butter,Milk,3),(Diapers,Milk,3)\}$$

pruning-steps

Pruning Steps: Pruning Steps:

1. (Bread, Butter, Diapers):

- 2-item subsets:
 - $(Bread, Butter)(Bread, Butter)$: Exists in $L2L2$.
 - $(Bread, Diapers)(Bread, Diapers)$: Exists in $L2L2$.
 - $(Butter, Diapers)(Butter, Diapers)$: **Does not exist** in $L2L2$.
- **Eliminate** $(Bread, Butter, Diapers)(Bread, Butter, Diapers)$.

2. (Bread, Butter, Milk):

- 2-item subsets:
 - $(Bread, Butter)(Bread, Butter)$: Exists in $L2L2$.
 - $(Bread, Milk)(Bread, Milk)$: Exists in $L2L2$.
 - $(Butter, Milk)(Butter, Milk)$: Exists in $L2L2$.
- **Keep** $(Bread, Butter, Milk)(Bread, Butter, Milk)$.

3. (Bread, Diapers, Milk):

- 2-item subsets:
 - $(Bread, Diapers)(Bread, Diapers)$: Exists in $L2L2$.
 - $(Bread, Milk)(Bread, Milk)$: Exists in $L2L2$.
 - $(Diapers, Milk)(Diapers, Milk)$: Exists in $L2L2$.
- **Keep** $(Bread, Diapers, Milk)(Bread, Diapers, Milk)$.

pruned-c3c3

Pruned C3C3:Pruned C3C3:

$$C3 = \{(Bread, Butter, Milk), (Bread, Diapers, Milk)\}$$

C3={ (Bread,Butter,Milk),(Bread,Diapers,Milk)} Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-9

Update C3 with the count of each entry from transactions

Tutor said:

To update C3C3 with the count of each 3-item combination, we check the number of transactions in which each combination appears.

pruned-c3c3-1

Pruned C3C3:Pruned C3C3:

$$C3 = \{(Bread, Butter, Milk), (Bread, Diapers, Milk)\}$$

C3={ (Bread,Butter,Milk),(Bread,Diapers,Milk)}
transactions-sorted

Transactions (sorted):Transactions (sorted):

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

count-for-each-combination

Count for Each Combination:Count for Each Combination:1. **(Bread, Butter, Milk):**

- Appears in transactions 1, 5, 6 → **3 transactions**

2. **(Bread, Diapers, Milk):**

- Appears in transactions 3, 5, 6 → **3 transactions**

updated-c3c3

Updated C3C3:Updated C3C3:

$$C3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$$

$C3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$ Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-10

Using same minsup = 3 Give $L3 = C3 - \{\text{item count} < \text{minsup}\}$

Tutor said:

To compute $L3L3$, we remove pairs from $C3C3$ where the count is less than the minimum support ($\text{minsup} = 3$).

given-c3c3-1

Given C3C3:Given C3C3:

$$C3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$$

$C3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$

filter-pairs-with-count-geq-textminsupminsup-1

Filter Pairs with Count $\geq \text{minsup}$:Filter Pairs with Count $\geq \text{minsup}$:

1. **(Bread, Butter, Milk, 3):** Count 3 ≥ 3 → **Include**
2. **(Bread, Diapers, Milk, 3):** Count 3 ≥ 3 → **Include**

result-l3l3

Result L3L3:Result L3L3:

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$$

L3={ (Bread,Butter,Milk,3),(Bread,Diapers,Milk,3)} Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-11

Since these cannot be combined using the formula, all the L_k are frequent item sets.

Give item sets as Union of L1, L2, L3

2/2

Tutor said:

To obtain the final set of frequent item sets as the union of L1L1, L2L2, and L3L3, we combine all the items from these sets.

given-l1l1-l2l2-and-l3l3

Given L1L1, L2L2, and L3L3:Given L1L1, L2L2, and L3L3:

- **L1L1:**

$$L1 = \{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4)\}$$

L1={ (Bread,6),(Butter,5),(Diapers,3),(Milk,4)}

- **L2L2:**

$$L2 = \{(Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3), (Diapers, Milk, 3)\}$$

L2={ (Bread,Butter,5),(Bread,Diapers,3),(Bread,Milk,4),(Butter,Milk,3),(Diapers,Milk,3)}

- **L3L3:**

$$L3 = \{(Bread, Butter, Milk, 3), (Bread, Diapers, Milk, 3)\}$$

L3={ (Bread,Butter,Milk,3),(Bread,Diapers,Milk,3)}

union-of-l1l1-l2l2-and-l3l3

Union of $L1L1$, $L2L2$, and $L3L3$: Union of $L1L1$, $L2L2$, and $L3L3$:

The union of these sets includes all the item sets from $L1L1$, $L2L2$, and $L3L3$.

$$\text{Frequent Item Sets} = L1 \cup L2 \cup L3$$

Frequent Item Sets= $L1L2L3$

final-result

Final Result:Final Result:

$\{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4), (Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Bu$

$\{(Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4), (Bread, Butter, 5), (Bread, Diapers, 3), (Bread, Milk, 4), (Butter, Milk, 3)$

said:Tutor said:Tutor said:Tutor said:
prompt-textarea

composer-background

Tutor said:

0.5 Itemset Mining 2

Student-said

sort the items in these transactions lexicographically $\{Milk, Bread, Butter\}$, $\{Bread, Butter\}$, $\{Milk, Bread, Diapers\}$, $\{Bread, Butter, Beer\}$, $\{Milk, Bread, Butter, Beer, Diapers\}$, $\{Milk, Bread, Butter, Diapers\}$

Tutor said:

To sort the items in each transaction lexicographically, we arrange the items alphabetically:

1. $\{Milk, Bread, Butter\} \rightarrow \{Bread, Butter, Milk\}$
2. $\{Bread, Butter\} \rightarrow \{Bread, Butter\}$
3. $\{Milk, Bread, Diapers\} \rightarrow \{Bread, Diapers, Milk\}$
4. $\{Bread, Butter, Beer\} \rightarrow \{Beer, Bread, Butter\}$
5. $\{Milk, Bread, Butter, Beer, Diapers\} \rightarrow \{Beer, Bread, Butter, Diapers, Milk\}$
6. $\{Milk, Bread, Butter, Diapers\} \rightarrow \{Bread, Butter, Diapers, Milk\}$

final-result

Final Result:

- {Bread, Butter, Milk}
- {Bread, Butter}
- {Bread, Diapers, Milk}
- {Beer, Bread, Butter}
- {Beer, Bread, Butter, Diapers, Milk}
- {Bread, Butter, Diapers, Milk}

Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-1

Count the occurrences of each item in these transactions as C1

Tutor said:

To count the occurrences of each item across all transactions (C1):
transactions

Transactions:

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

count-of-each-item

Count of each item:

- **Bread:** Appears in **6** transactions.
- **Butter:** Appears in **5** transactions.
- **Milk:** Appears in **4** transactions.
- **Diapers:** Appears in **3** transactions.
- **Beer:** Appears in **2** transactions.

result

Result:

[]@ll@	
() Item	Count
() Bread	6
Butter	5
Milk	4
Diapers	3
Beer	2
() Tutor said:Tutor said:Tutor said:Tutor said:	
Student-said-2	
remove any item that has count < minsup from C1. minsup = 3. return the remaining lexicographically sorted result as L1	
2/2	
Tutor said:	
To generate L1 , we remove items from C1 where the count is less than the minimum support (minsup = 3) and sort the remaining items lexicographically.	
c1	

C1:

[]@ll@	
() Item	Count
() Bread	6
Butter	5
Milk	4

Diapers 3

Beer 2

()

filtering

Filtering:

- Remove **Beer** because its count (**2**) is less than **minsup (3)**.

remaining-items

Remaining Items:

- Bread, Butter, Milk, Diapers.

lexicographically-sorted-l1

Lexicographically Sorted L1:

L1 = {Bread, Butter, Diapers, Milk}Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-3

L1 also contains the count of occurrences of the items

Tutor said:

Including the counts with the lexicographically sorted items in **L1**:

filtered-items-with-counts

Filtered Items with Counts:

||@||@

() Item Count

() Bread 6

Butter 5

Diapers 3

Milk 4

()

lexicographically-sorted-l1-1

Lexicographically Sorted L1:Lexicographically Sorted L1:

||@||@

() Item Count

() Bread 6
 Butter 5
 Diapers 3
 Milk 4

() Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-4
 Return L1 as a set containing the count as well
Tutor said:
 The **L1** set with items and their counts can be represented as:
L1 = { (Bread, 6), (Butter, 5), (Diapers, 3), (Milk, 4) }Tutor said:Tutor
 said:Tutor said:Tutor said:
 Student-said-5

For $L1 = \{ \{Bread: 6\}, \{Butter: 5\}, \{Diapers: 3\}, \{Milk: 4\} \}$ make C2 such that
 Insert into C.k SELECT p.item_1, p.item_2, ..., p.item_{k-1}, p.item_k FROM L_{k-2} p,
 L_{k-1} q WHERE p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} <
 q.item_{k-1}

Tutor said:

To construct **C2** (candidate 2-itemsets) from **L1**, the process is as follows:
 steps

Steps:

1. **L1** = { {Bread: 6}, {Butter: 5}, {Diapers: 3}, {Milk: 4} }
2. Generate all possible 2-item combinations from the items in **L1** that meet the conditions:

- The items share the first $k - 2 = 0$ items (trivially true in this case).
- $p.item_{k-1} < q.item_{k-1}$, i.e., lexicographic order.

candidate-2-itemsets-c2

Candidate 2-Itemsets (C2):

Combine items from **L1** lexicographically:

- (Bread, Butter)
- (Bread, Diapers)
- (Bread, Milk)
- (Butter, Diapers)
- (Butter, Milk)
- (Diapers, Milk)

C2:

||@l@

() Itemset

() {Bread, Butter}
 {Bread, Diapers}
 {Bread, Milk}
 {Butter, Diapers}
 {Butter, Milk}
 {Diapers, Milk}

() Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-6

For each itemset, count the transactions in which all items of the itemset appear and give it as a set L2

Tutor said:

To generate **L2**, we count the occurrences of each 2-itemset in the transactions where all items in the set appear. The original transactions are:

transactions-1

Transactions:

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}

6. {Bread, Butter, Diapers, Milk}

candidate-2-itemsets-c2-1

Candidate 2-Itemsets (C2):Candidate 2-Itemsets (C2):

- {Bread, Butter}
- {Bread, Diapers}
- {Bread, Milk}
- {Butter, Diapers}
- {Butter, Milk}
- {Diapers, Milk}

counting-transactions-for-each-itemset

Counting Transactions for Each Itemset:

- {Bread, Butter}: Appears in transactions 1, 2, 4, 5, 6 → **5 occurrences**
- {Bread, Diapers}: Appears in transactions 3, 5, 6 → **3 occurrences**
- {Bread, Milk}: Appears in transactions 1, 3, 5, 6 → **4 occurrences**
- {Butter, Diapers}: Appears in transactions 5, 6 → **2 occurrences**
- {Butter, Milk}: Appears in transactions 1, 5, 6 → **3 occurrences**
- {Diapers, Milk}: Appears in transactions 3, 5, 6 → **3 occurrences**

applying-minimum-support-minsup-3

Applying Minimum Support (minsup = 3):Applying Minimum Support (minsup = 3):

- Remove {Butter, Diapers} (count = 2).

L2:L2:

[]@[]@	
() Itemset	Count
<hr/>	
() {Bread, Butter}	5
{Bread, Diapers}	3
{Bread, Milk}	4
{Butter, Milk}	3
{Diapers, Milk}	3
<hr/>	
()	
resulting-set	

Resulting Set:Resulting Set:

L2 = { {Bread, Butter: 5}, {Bread, Diapers: 3}, {Bread, Milk: 4}, {Butter, Milk: 3}, {Diapers, Milk: 3} }Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-7

from this make C3 such that Insert into C k SELECT p.item_1, p.item_2, ..., p.item_{k-1}, p.item_k FROM L_{k-2} p, L_{k-1} q WHERE p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}

Tutor said:

To construct **C3** (candidate 3-itemsets) from **L2**, we combine the itemsets of **L2** according to the rule:

rule-for-joining

Rule for Joining:

- **Condition:** $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$
 $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$.
- $k - 2 = 1$: This means the first item in the itemsets must be the same.

l2-1

L2:L2:

[]@[]@	
() Itemset	Count
<hr/>	
() {Bread, Butter}	5
{Bread, Diapers}	3

{Bread, Milk} 4
 {Butter, Milk} 3
 {Diapers, Milk} 3

()
 step-1-combine-itemsets

Step 1: Combine Itemsets

We combine 2-itemsets from **L2** to form 3-itemsets:

1. {Bread, Butter} and {Bread, Diapers} \rightarrow {Bread, Butter, Diapers}
2. {Bread, Butter} and {Bread, Milk} \rightarrow {Bread, Butter, Milk}
3. {Bread, Diapers} and {Bread, Milk} \rightarrow {Bread, Diapers, Milk}
4. {Butter, Milk} and {Diapers, Milk} \rightarrow {Butter, Diapers, Milk}

step-2-verify-lexicographical-order

Step 2: Verify Lexicographical Order

The generated 3-itemsets are:

- {Bread, Butter, Diapers}
- {Bread, Butter, Milk}
- {Bread, Diapers, Milk}
- {Butter, Diapers, Milk}

c3

C3:C3:

||@l@

() Itemset

() {Bread, Butter, Diapers}
 {Bread, Butter, Milk}
 {Bread, Diapers, Milk}
 {Butter, Diapers, Milk}

() Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-8

validate all itemset in C3

Tutor said:

To validate the 3-itemsets in **C3**, we count the transactions in which all items of the 3-itemset appear. The transactions are:

transactions-2

Transactions:

1. {Bread, Butter, Milk}
2. {Bread, Butter}
3. {Bread, Diapers, Milk}
4. {Beer, Bread, Butter}
5. {Beer, Bread, Butter, Diapers, Milk}
6. {Bread, Butter, Diapers, Milk}

c3-candidates

C3 Candidates:

1. {Bread, Butter, Diapers}
2. {Bread, Butter, Milk}
3. {Bread, Diapers, Milk}
4. {Butter, Diapers, Milk}

count-for-each-itemset

Count for Each Itemset:

1. {Bread, Butter, Diapers}:

- Appears in transactions: 5, 6 → **2 occurrences**

2. **{Bread, Butter, Milk}**:

- Appears in transactions: 1, 5, 6 → **3 occurrences**

3. **{Bread, Diapers, Milk}**:

- Appears in transactions: 3, 5, 6 → **3 occurrences**

4. **{Butter, Diapers, Milk}**:

- Appears in transactions: 5, 6 → **2 occurrences**

apply-minimum-support-minsup-3

Apply Minimum Support (minsup = 3):Apply Minimum Support (minsup = 3):

- Remove **{Bread, Butter, Diapers}** (count = 2).
- Remove **{Butter, Diapers, Milk}** (count = 2).

validated-l3

Validated L3:Validated L3:

[]@[]@	
() Itemset	Count
() {Bread, Butter, Milk}	3
{Bread, Diapers, Milk}	3
()	
l3	

L3:L3:

L3 = { {Bread, Butter, Milk: 3}, {Bread, Diapers, Milk: 3} }Tutor said:Tutor said:Tutor said:
prompt-textarea
composer-background

0.6 Itemset Mining 3

Student-said

Sort items in each individual transaction lexicographically: 1 - {Carrot}, {Apple}, {Flour} 2 - {Apple}, {Detergent}, {Eggs}, {Carrot} 3 - {Carrot}, {Eggs}, {Apple} 4 - {{Apple}, {Detergent} 5 - {Banana}, {Flour}, {Apple}, {Carrot} 6 - {Carrot}, {Flour}, {Detergent}, {Apple}, {Eggs}

Tutor said:

Here are the transactions with items sorted lexicographically:

1 - {Apple}, {Carrot}, {Flour}
 2 - {Apple}, {Carrot}, {Detergent}, {Eggs}
 3 - {Apple}, {Carrot}, {Eggs}
 4 - {Apple}, {Detergent}
 5 - {Apple}, {Banana}, {Carrot}, {Flour}
 6 - {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour} Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-1

list all unique items in the transactions

Tutor said:

The unique items present in all the transactions are:

Apple, Banana, Carrot, Detergent, Eggs, Flour Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-2

give the count of each item in all transactions

Tutor said:

Here is the count of each item across all transactions:

- **Apple:** 6
- **Banana:** 1
- **Carrot:** 5
- **Detergent:** 4
- **Eggs:** 3
- **Flour:** 4

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-3

in how many transactions does {Detergent} appears?

Tutor said:

The item **{Detergent}** appears in **4** transactions (Transactions 2, 4, and 6).Tutor
 said:Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-4

That makes 3 transactions

Tutor said:

You're right! Let's recount the transactions where **{Detergent}** appears:

- **Transaction 2:** {Apple}, {Carrot}, {Detergent}, {Eggs}
- **Transaction 4:** {Apple}, {Detergent}
- **Transaction 6:** {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

So, **{Detergent}** appears in **3** transactions. Thanks for catching that! Tutor said:Tutor
 said:Tutor said:Tutor said:Tutor said:
 Student-said-5

in how many transactions {Flour} appears?

Tutor said:

Let's count the transactions where **{Flour}** appears:

- **Transaction 1:** {Apple}, {Carrot}, {Flour}
- **Transaction 5:** {Apple}, {Banana}, {Carrot}, {Flour}
- **Transaction 6:** {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

So, **{Flour}** appears in **3** transactions. Tutor said:Tutor said:Tutor said:Tutor said:Tutor
 said:
 Student-said-6

give all items with the counts

Tutor said:

Here is the count of each item across all transactions:

- **Apple: 6** transactions
- **Banana: 1** transaction
- **Carrot: 5** transactions
- **Detergent: 3** transactions
- **Eggs: 3** transactions
- **Flour: 3** transactions

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-7

remove the items with less than 3 count and return the remaining with count as L1

Tutor said:

Here is **L1**, containing only the items that appear in at least **3** transactions:

- **Apple: 6**
- **Carrot: 5**
- **Detergent: 3**
- **Eggs: 3**
- **Flour: 3**

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-8

Give it as a set with count

2/2

Tutor said:

Here is **L1** represented as a set with the counts:

L1 = { (Apple: 6), (Carrot: 5), (Detergent: 3), (Eggs: 3), (Flour: 3) }

2/2Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-9

Give sets of 2-items with {Apple} from L1 such that "Apple" < X lexicographically

Tutor said:

Here are the valid **2-item sets** with {Apple}, where "Apple" < X lexicographically, using items from **L1**:

L2 = { (Apple, Carrot), (Apple, Detergent), (Apple, Eggs), (Apple, Flour) }Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-10

Give sets of 2-items with {Carrot} from L1 such that "Carrot" < X lexicographically

Tutor said:

Here are the valid **2-item sets** with {Carrot}, where "Carrot" < X lexicographically, using items from **L1**:

L2 = { (Carrot, Detergent), (Carrot, Eggs), (Carrot, Flour) }Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-11

Give sets of 2-items with {Detergent} from L1 such that "Detergent" < X lexicographically

Tutor said:

Here are the valid **2-item sets** with {Detergent}, where "Detergent" < X lexicographically, using items from **L1**:

L2 = { (Detergent, Eggs), (Detergent, Flour) }Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-12

Give sets of 2-items with {Eggs} from L1 such that "Eggs" < X lexicographically

Tutor said:

Here is the valid **2-item set** with {Eggs}, where "Eggs" < X lexicographically, using items from **L1**:

L2 = { (Eggs, Flour) }Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-13

Give sets of 2-items with {Flour} from L1 such that "Flour" < X lexicographically

Tutor said:

There are **no** valid **2-item sets** with {Flour}, since there is no item in **L1** that comes after "Flour" lexicographically.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-14

Combine all these 2-itemsets as C2

Tutor said:

Here is **C2**, the combined set of all valid **2-itemsets** from **L1**:

C2 = { (Apple, Carrot), (Apple, Detergent), (Apple, Eggs), (Apple, Flour), (Carrot, Detergent), (Carrot, Eggs), (Carrot, Flour), (Detergent, Eggs), (Detergent, Flour), (Eggs, Flour) }Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-15

count transactions in which (Apple, Carrot) appear together

Tutor said:

Let's count the transactions where **(Apple, Carrot)** appear together:

Given transactions:

1. {Apple}, {Carrot}, {Flour}
2. {Apple}, {Carrot}, {Detergent}, {Eggs}
3. {Apple}, {Carrot}, {Eggs}
4. {Apple}, {Detergent}
5. {Apple}, {Banana}, {Carrot}, {Flour}
6. {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

Total count = 5 transactions.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-16

count transactions in which (Apple, Detergent) appear together

Tutor said:

Let's count the transactions where **(Apple, Detergent)** appear together:

Given transactions:

1. {Apple}, {Carrot}, {Flour}
2. {Apple}, {Carrot}, {Detergent}, {Eggs}
3. {Apple}, {Carrot}, {Eggs}
4. {Apple}, {Detergent}
5. {Apple}, {Banana}, {Carrot}, {Flour}

6. {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

Total count = 3 transactions.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-17

count transactions in which (Apple, Eggs) appear together

Tutor said:

Let's count the transactions where (**Apple, Eggs**) appear together:

Given transactions:

1. {Apple}, {Carrot}, {Flour}
2. {Apple}, {Carrot}, {Detergent}, {Eggs}
3. {Apple}, {Carrot}, {Eggs}
4. {Apple}, {Detergent}
5. {Apple}, {Banana}, {Carrot}, {Flour}
6. {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

Total count = 3 transactions.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-18

count transactions in which (Apple, Flour) appear together

Tutor said:

Let's count the transactions where (**Apple, Flour**) appear together:

Given transactions:

1. {Apple}, {Carrot}, {Flour}
2. {Apple}, {Carrot}, {Detergent}, {Eggs}
3. {Apple}, {Carrot}, {Eggs}
4. {Apple}, {Detergent}
5. {Apple}, {Banana}, {Carrot}, {Flour}
6. {Apple}, {Carrot}, {Detergent}, {Eggs}, {Flour}

Total count = 3 transactions.Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-19

count transactions in which (Carrot, Detergent) appear together count transactions in which (Carrot, Eggs) appear together count transactions in which (Carrot, Flour) appear together

Tutor said:

Here are the counts for each **2-itemset** in transactions:

1. **(Carrot, Detergent)**

- Appears in Transactions: **2, 6**
- **Total count = 2** transactions

2. **(Carrot, Eggs)**

- Appears in Transactions: **2, 3, 6**
- **Total count = 3** transactions

3. **(Carrot, Flour)**

- Appears in Transactions: **1, 5, 6**
- **Total count = 3** transactions

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-20

count transactions in which (Detergent, Eggs) appear together count transactions in which (Detergent, Flour) appear together count transactions in which (Eggs, Flour) appear together

Tutor said:

Here are the counts for each **2-itemset** in transactions:

1. **(Detergent, Eggs)**

- Appears in Transactions: **2, 6**
- **Total count = 2** transactions

2. (Detergent, Flour)

- Appears in Transactions: **6**
- **Total count** = 1 transaction

3. (Eggs, Flour)

- Appears in Transactions: **6**
- **Total count** = 1 transaction

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-21

combine all the counts of 2-itemsets

Tutor said:

Here is the combined count of all **2-itemsets** from **C2**:

L2 = {
 (Apple, Carrot): 5,
 (Apple, Detergent): 3,
 (Apple, Eggs): 3,
 (Apple, Flour): 3,
 (Carrot, Detergent): 2,
 (Carrot, Eggs): 3,
 (Carrot, Flour): 3,
 (Detergent, Eggs): 2,
 (Detergent, Flour): 1,
 (Eggs, Flour): 1
 }Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-22

remove the 2-itemsets with a count less than 3 and return the set of remaining with the count as L2

Tutor said:

Here is **L2**, containing only the **2-itemsets** that appear in at least **3** transactions:

L2 = {
 (Apple, Carrot): 5,
 (Apple, Detergent): 3,
 (Apple, Eggs): 3,
 (Apple, Flour): 3,

(Carrot, Eggs): 3,
 (Carrot, Flour): 3
 }Tutor said:Tutor said:Tutor said:Tutor said:
 Student-said-23

give all combinations of $X = (\text{Apple}, \text{Carrot})$ with all other 2-itemsets Y such that
 $X[-1] = Y[-1] \ \& \ X[-1] < Y[-1]$

2/2

Tutor said:

We are given $X = (\text{Apple}, \text{Carrot})$ and need to find all valid Y from **L2** such that:

1. $X[-1] = Y[-1] \rightarrow$ The first item of X and Y must be the same.
2. $X[-1] < Y[-1] \rightarrow$ The second item of X must be lexicographically smaller than the second item of Y .

From **L2**, we have the following **2-itemsets**:

- (Apple, Carrot)
- (Apple, Detergent)
- (Apple, Eggs)
- (Apple, Flour)
- (Carrot, Eggs)
- (Carrot, Flour)

step-1-identify-valid-y

Step 1: Identify valid Y

Since $X = (\text{Apple}, \text{Carrot})$, we need Y where:

- The first item is **Apple** (same as $X[-1]$).
- The second item is lexicographically greater than **Carrot**.

From **L2**, the valid Y are:

- (Apple, Detergent)

- (Apple, Eggs)
- (Apple, Flour)

step-2-generate-combinations-x-y

Step 2: Generate combinations (X, Y)

- ((Apple, Carrot), (Apple, Detergent))
- ((Apple, Carrot), (Apple, Eggs))
- ((Apple, Carrot), (Apple, Flour))

final-result

Final Result:

$C3 = \{ ((\text{Apple}, \text{Carrot}), (\text{Apple}, \text{Detergent})), ((\text{Apple}, \text{Carrot}), (\text{Apple}, \text{Eggs})), ((\text{Apple}, \text{Carrot}), (\text{Apple}, \text{Flour})) \}$

Student-said-24

take union of each combination of X & Y

Tutor said:

Taking the **union** of each valid (X, Y) pair from **C3**, we get the following **3-itemsets**:

1. (Apple, Carrot) (Apple, Detergent) = (Apple, Carrot, Detergent)
2. (Apple, Carrot) (Apple, Eggs) = (Apple, Carrot, Eggs)
3. (Apple, Carrot) (Apple, Flour) = (Apple, Carrot, Flour)

final-result-1

Final Result:

$C3 = \{ (\text{Apple}, \text{Carrot}, \text{Detergent}), (\text{Apple}, \text{Carrot}, \text{Eggs}), (\text{Apple}, \text{Carrot}, \text{Flour}) \}$

Student-said-25

give all combinations of X = (Apple, Detergent) with all other 2-itemsets Y such that $X[-1] = Y[-1]$ & $X[-1] < Y[-1]$ and return the union of each valid combination

Tutor said:

We are given **X** = (Apple, Detergent) and need to find all valid **Y** from **L2** such that:

conditions

Conditions:

1. $X[:1] = Y[:1] \rightarrow$ The first item of **X** and **Y** must be the same.
2. $X[-1] < Y[-1] \rightarrow$ The second item of **X** must be lexicographically smaller than the second item of **Y**.

step-1-identify-valid-y-1

Step 1: Identify valid Y

Since **X** = (**Apple**, **Detergent**), we need **Y** where:

- The first item is **Apple** (same as $X[:1]$).
- The second item is lexicographically greater than **Detergent**.

From **L2**, the valid **Y** are:

- (**Apple**, **Eggs**)
- (**Apple**, **Flour**)

step-2-take-the-union-of-each-valid-combination

Step 2: Take the union of each valid combination

1. (**Apple**, **Detergent**) (**Apple**, **Eggs**) = (**Apple**, **Detergent**, **Eggs**)
2. (**Apple**, **Detergent**) (**Apple**, **Flour**) = (**Apple**, **Detergent**, **Flour**)

final-result-2

Final Result:

C3 = { (**Apple**, **Detergent**, **Eggs**), (**Apple**, **Detergent**, **Flour**) } Tutor said: Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-26

give all combinations of $X = (\text{Apple}, \text{Eggs})$ with all other 2-itemsets **Y** such that $X[:1] = Y[:1]$ & $X[-1] < Y[-1]$ and return the union of each valid combination

Tutor said:

We are given **X** = (**Apple**, **Eggs**) and need to find all valid **Y** from **L2** such that: conditions-1

Conditions:

1. $\mathbf{X}[:1] = \mathbf{Y}[:1] \rightarrow$ The first item of \mathbf{X} and \mathbf{Y} must be the same.
2. $\mathbf{X}[-1] < \mathbf{Y}[-1] \rightarrow$ The second item of \mathbf{X} must be lexicographically smaller than the second item of \mathbf{Y} .

step-1-identify-valid-y-2

Step 1: Identify valid Y

Since $\mathbf{X} = (\mathbf{Apple}, \mathbf{Eggs})$, we need \mathbf{Y} where:

- The first item is **Apple** (same as $\mathbf{X}[:1]$).
- The second item is lexicographically greater than **Eggs**.

From **L2**, the valid \mathbf{Y} is:

- **(Apple, Flour)**

step-2-take-the-union-of-each-valid-combination-1

Step 2: Take the union of each valid combination

1. **(Apple, Eggs) (Apple, Flour) = (Apple, Eggs, Flour)**

final-result-3

Final Result:

C3 = { (Apple, Eggs, Flour) } Tutor said: Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-27

give all combinations of $\mathbf{X} = (\mathbf{Apple}, \mathbf{Flour})$ with all other 2-itemsets \mathbf{Y} such that $\mathbf{X}[:1] = \mathbf{Y}[:1]$ & $\mathbf{X}[-1] < \mathbf{Y}[-1]$ and return the union of each valid combination

Tutor said:

We are given $\mathbf{X} = (\mathbf{Apple}, \mathbf{Flour})$ and need to find all valid \mathbf{Y} from **L2** such that:
conditions-2

Conditions:

1. $\mathbf{X}[:1] = \mathbf{Y}[:1] \rightarrow$ The first item of \mathbf{X} and \mathbf{Y} must be the same.
2. $\mathbf{X}[-1] < \mathbf{Y}[-1] \rightarrow$ The second item of \mathbf{X} must be lexicographically smaller than the second item of \mathbf{Y} .

step-1-identify-valid-y-3

Step 1: Identify valid Y

Since $\mathbf{X} = (\mathbf{Apple}, \mathbf{Flour})$, we need \mathbf{Y} where:

- The first item is **Apple** (same as $\mathbf{X}[:1]$).
- The second item is lexicographically greater than **Flour**.

From **L2**, there are no **2-itemsets** where the second item comes after **Flour** lexicographically.

step-2-result

Step 2: Result

Since there are no valid \mathbf{Y} for $(\mathbf{Apple}, \mathbf{Flour})$, there are no valid **3-itemsets**.

final-result-4

Final Result:

C3 = { } (Empty set, no valid combinations)Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-28

give all combinations of $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Eggs})$ with all other 2-itemsets \mathbf{Y} such that $\mathbf{X}[:1] = \mathbf{Y}[:1]$ & $\mathbf{X}[-1] < \mathbf{Y}[-1]$ and return the union of each valid combination

Tutor said:

We are given $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Eggs})$ and need to find all valid \mathbf{Y} from **L2** such that:

conditions-3

Conditions:

1. $\mathbf{X}[:1] = \mathbf{Y}[:1] \rightarrow$ The first item of \mathbf{X} and \mathbf{Y} must be the same.
2. $\mathbf{X}[-1] < \mathbf{Y}[-1] \rightarrow$ The second item of \mathbf{X} must be lexicographically smaller than the second item of \mathbf{Y} .

step-1-identify-valid-y-4

Step 1: Identify valid Y

Since $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Eggs})$, we need \mathbf{Y} where:

- The first item is **Carrot** (same as $\mathbf{X}[-1]$).
- The second item is lexicographically greater than **Eggs**.

From **L2**, the valid \mathbf{Y} is:

- **(Carrot, Flour)**

step-2-take-the-union-of-each-valid-combination-2

Step 2: Take the union of each valid combination

1. **(Carrot, Eggs) (Carrot, Flour) = (Carrot, Eggs, Flour)**

final-result-5

Final Result:

$\mathbf{C3} = \{ (\mathbf{Carrot}, \mathbf{Eggs}, \mathbf{Flour}) \}$ Tutor said: Tutor said: Tutor said: Tutor said: Tutor said:

Student-said-29

give all combinations of $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Flour})$ with all other 2-itemsets \mathbf{Y} such that $\mathbf{X}[-1] = \mathbf{Y}[-1]$ & $\mathbf{X}[-1] < \mathbf{Y}[-1]$ and return the union of each valid combination

Tutor said:

We are given $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Flour})$ and need to find all valid \mathbf{Y} from **L2** such that:
conditions-4

Conditions:

1. $\mathbf{X}[-1] = \mathbf{Y}[-1] \rightarrow$ The first item of \mathbf{X} and \mathbf{Y} must be the same.
2. $\mathbf{X}[-1] < \mathbf{Y}[-1] \rightarrow$ The second item of \mathbf{X} must be lexicographically smaller than the second item of \mathbf{Y} .

step-1-identify-valid-y-5

Step 1: Identify valid Y

Since $\mathbf{X} = (\mathbf{Carrot}, \mathbf{Flour})$, we need \mathbf{Y} where:

- The first item is **Carrot** (same as $\mathbf{X}[:-1]$).
- The second item is lexicographically greater than **Flour**.

From **L2**, there are no **2-itemsets** where the second item comes after **Flour** lexicographically.

step-2-result-1

Step 2: Result

Since there are no valid \mathbf{Y} for $(\mathbf{Carrot}, \mathbf{Flour})$, there are no valid **3-itemsets**.

final-result-6

Final Result:

C3 = { } (Empty set, no valid combinations)

Student-said-30

combine all 3-itemsets in C3

Tutor said:

Here is the combined set of all **3-itemsets** in **C3**:

C3 = {
(Apple, Carrot, Detergent),
(Apple, Carrot, Eggs),
(Apple, Carrot, Flour),
(Apple, Detergent, Eggs),
(Apple, Detergent, Flour),
(Apple, Eggs, Flour),
(Carrot, Eggs, Flour)
}

Student-said-31

remove all 3-itemsets in which 2-itemsets of the last 2 items do not appear in **L2**

Tutor said:

To filter **C3**, we need to remove **3-itemsets** where the **2-itemset of the last two items** does not exist in **L2**.

given

Given:

```

C3 = {
(Apple, Carrot, Detergent),
(Apple, Carrot, Eggs),
(Apple, Carrot, Flour),
(Apple, Detergent, Eggs),
(Apple, Detergent, Flour),
(Apple, Eggs, Flour),
(Carrot, Eggs, Flour)
}

```

l2-valid-2-itemsets

L2 (Valid 2-itemsets):

```

L2 = {
(Apple, Carrot),
(Apple, Detergent),
(Apple, Eggs),
(Apple, Flour),
(Carrot, Eggs),
(Carrot, Flour)
}

```

step-1-check-last-two-items-of-each-3-itemset

Step 1: Check last two items of each 3-itemset

1. (Apple, Carrot, Detergent) → (Carrot, Detergent) (Not in L2) → **Remove**
2. (Apple, Carrot, Eggs) → (Carrot, Eggs) (In L2) → **Keep**
3. (Apple, Carrot, Flour) → (Carrot, Flour) (In L2) → **Keep**
4. (Apple, Detergent, Eggs) → (Detergent, Eggs) (Not in L2) → **Remove**
5. (Apple, Detergent, Flour) → (Detergent, Flour) (Not in L2) → **Remove**
6. (Apple, Eggs, Flour) → (Eggs, Flour) (Not in L2) → **Remove**
7. (Carrot, Eggs, Flour) → (Eggs, Flour) (Not in L2) → **Remove**

final-c3-after-filtering

Final C3 after filtering:

C3 = {
(Apple, Carrot, Eggs),
(Apple, Carrot, Flour)
} Tutor said: Tutor said: Tutor said: Tutor said: Tutor said:
 Student-said-32
 count transactions in which (Apple, Carrot, Eggs) appear together count transactions
 in which (Apple, Carrot, Flour) appear together
Tutor said:
 Let's count the transactions where the **3-itemsets** appear together:
 given-transactions

Given Transactions:

1. {Apple, Carrot, Flour}
 2. {Apple, Carrot, Detergent, Eggs}
 3. {Apple, Carrot, Eggs}
 4. {Apple, Detergent}
 5. {Apple, Banana, Carrot, Flour}
 6. {Apple, Carrot, Detergent, Eggs, Flour}
- count-for-apple-carrot-eggs

Count for (Apple, Carrot, Eggs)

- Appears in **Transaction 2**
- Appears in **Transaction 3**
- Appears in **Transaction 6**

Total count = 3 transactions.
 count-for-apple-carrot-flour

Count for (Apple, Carrot, Flour)Count for (Apple, Carrot, Flour)

- Appears in **Transaction 1**
- Appears in **Transaction 5**
- Appears in **Transaction 6**

Total count = 3 transactions.
final-result-7

Final Result:

- (Apple, Carrot, Eggs) → 3 transactions
- (Apple, Carrot, Flour) → 3 transactions

Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
Student-said-33

remove the 3-itemsets with a count less than 3 and return the remaining 3-itemsets with the count as L3

Tutor said:

Since both (Apple, Carrot, Eggs) and (Apple, Carrot, Flour) appear in **3 transactions**, they qualify for **L3**.

final-l3

Final L3:Final L3:

L3 = {
(Apple, Carrot, Eggs): 3,
(Apple, Carrot, Flour): 3
}Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-34

give all combinations of $X = (\text{Apple}, \text{Carrot}, \text{Eggs})$ with all other 3-itemsets Y such that $X[-1] = Y[-1] \ \& \ X[-1] < Y[-1]$ and return the union of each valid combination

Tutor said:

We are given $X = (\text{Apple}, \text{Carrot}, \text{Eggs})$ and need to find all valid Y from **L3** such that:

conditions-5

Conditions:Conditions:

1. $X[:-1] = Y[:-1] \rightarrow$ The first two items of \mathbf{X} and \mathbf{Y} must be the same.
2. $X[-1] < Y[-1] \rightarrow$ The third item of \mathbf{X} must be lexicographically smaller than the third item of \mathbf{Y} .

step-1-identify-valid-y-6

Step 1: Identify valid YStep 1: Identify valid Y

Since $\mathbf{X} = (\mathbf{Apple}, \mathbf{Carrot}, \mathbf{Eggs})$, we need \mathbf{Y} where:

- The first two items are **(Apple, Carrot)** (same as $X[:-1]$).
- The third item is lexicographically greater than **Eggs**.

From **L3**, the only other 3-itemset is **(Apple, Carrot, Flour)**.

- **(Apple, Carrot, Flour)** is valid because:
 - The first two items **(Apple, Carrot)** match $X[:-1]$.
 - **Eggs** < **Flour** lexicographically.

step-2-take-the-union-of-each-valid-combination-3

Step 2: Take the union of each valid combinationStep 2: Take the union of each valid combination

1. **(Apple, Carrot, Eggs) (Apple, Carrot, Flour) = (Apple, Carrot, Eggs, Flour)**

final-result-8

Final Result:Final Result:

$\mathbf{C4} = \{ (\mathbf{Apple}, \mathbf{Carrot}, \mathbf{Eggs}, \mathbf{Flour}) \}$ Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-35

give all combinations of $\mathbf{X} = (\mathbf{Apple}, \mathbf{Carrot}, \mathbf{Flour})$ with all other 3-itemsets \mathbf{Y} such that $X[:-1] = Y[:-1]$ & $X[-1] < Y[-1]$ and return the union of each valid combination

Tutor said:

We are given $\mathbf{X} = (\text{Apple}, \text{Carrot}, \text{Flour})$ and need to find all valid \mathbf{Y} from $\mathbf{L3}$ such that:

conditions-6

Conditions:Conditions:

1. $\mathbf{X}[:-1] = \mathbf{Y}[:-1] \rightarrow$ The first two items of \mathbf{X} and \mathbf{Y} must be the same.
2. $\mathbf{X}[-1] < \mathbf{Y}[-1] \rightarrow$ The third item of \mathbf{X} must be lexicographically smaller than the third item of \mathbf{Y} .

step-1-identify-valid-y-7

Step 1: Identify valid YStep 1: Identify valid Y

Since $\mathbf{X} = (\text{Apple}, \text{Carrot}, \text{Flour})$, we need \mathbf{Y} where:

- The first two items are **(Apple, Carrot)** (same as $\mathbf{X}[:-1]$).
- The third item is lexicographically greater than **Flour**.

From $\mathbf{L3}$, the only other 3-itemset is **(Apple, Carrot, Eggs)**.

- **(Apple, Carrot, Eggs)** is **not valid** because:
 - While the first two items **(Apple, Carrot)** match $\mathbf{X}[:-1]$,
 - **Flour** is **not** lexicographically smaller than **Eggs** (**Flour** > **Eggs**), which violates the condition $\mathbf{X}[-1] < \mathbf{Y}[-1]$.

step-2-result-2

Step 2: ResultStep 2: Result

Since there are no valid \mathbf{Y} for **(Apple, Carrot, Flour)**, there are no valid **4-itemsets**.

final-result-9

Final Result:Final Result:

$\mathbf{C4} = \{ \}$ (**Empty set, no valid combinations**)Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:

Student-said-36

remove all 4-itemsets in which the last 3 items do not appear in $\mathbf{L3}$

Tutor said:

The only **4-itemset** in **C4** is:

C4 = { (Apple, Carrot, Eggs, Flour) }

step-1-check-if-the-last-3-items-exist-in-l3

Step 1: Check if the last 3 items exist in L3

- The last 3 items of (Apple, Carrot, Eggs, Flour) are (Carrot, Eggs, Flour).

- Checking **L3**:

L3 = {
 (Apple, Carrot, Eggs): 3,
 (Apple, Carrot, Flour): 3
 }

- (Carrot, Eggs, Flour) is NOT in **L3**, so we remove this 4-itemset.

final-result-10

Final Result:

Since all **4-itemsets** were removed, the final **C4** is:

C4 = { } (Empty set, no valid 4-itemsets).
 Tutor said:
 Tutor said:
 Tutor said:
 Tutor said:

Student-said-37

combine all L1, L2 and L3 along with the counts

Tutor said:

Here is the combined set of **L1, L2, and L3** along with their counts:

l1-1-itemsets

L1 (1-itemsets):

- (Apple): 6
- (Carrot): 5
- (Detergent): 3
- (Eggs): 3
- (Flour): 3

l2-2-itemsets

L2 (2-itemsets):L2 (2-itemsets):

- (Apple, Carrot): 5
- (Apple, Detergent): 3
- (Apple, Eggs): 3
- (Apple, Flour): 3
- (Carrot, Eggs): 3
- (Carrot, Flour): 3

l3-3-itemsets

L3 (3-itemsets):L3 (3-itemsets):

- (Apple, Carrot, Eggs): 3
- (Apple, Carrot, Flour): 3

final-combined-set

Final Combined Set:Final Combined Set:

$L = \{$
 (Apple): 6, (Carrot): 5, (Detergent): 3, (Eggs): 3, (Flour): 3,
 (Apple, Carrot): 5, (Apple, Detergent): 3, (Apple, Eggs): 3, (Apple, Flour):
 3, (Carrot, Eggs): 3, (Carrot, Flour): 3,
 (Apple, Carrot, Eggs): 3, (Apple, Carrot, Flour): 3
 $\}$ Tutor said:Tutor said:Tutor said:Tutor said:Tutor said:
 prompt-textarea
 composer-background