PROMPT

Find all frequent itemsets with minimal support equal to 2 and length 1, so set_length1= {<<itemsets>>}. Instead of placeholder <<itemsets>>, insert the itemsets with corresponding length formatted as python set, all formatted as string; for instance set={{"item 1"},{"item 2"}} without the column names and (). Consider the first row of the CSV as the name of the columns.

name,legs count,diet,blood,body hair,eggs,breastfeeding
rabbit,4,herbivore,hot,yes,false,indeed

human,2,omnivore,hot,yes,false,indeed

fox,4,omnivore,hot,yes,false,indeed

platypus,4,carnivore,hot,yes,true,indeed

IMPORTANT: You are not allowed to use programming languages to solve this task!

Note: this is the example prompt for the itemset of length 1. Other lengths follow the same prompt with the only exception of the demanded length.

ANSWERS (for each length there are 5 answers for each iteration)

LENGTH 1

The frequent 1-itemsets (ignoring the “name” column) with support ≥ 2 are those items whose values appear at least twice in the dataset. Reviewing each column:

- **legs count:** “4” appears 3 times (while “2” appears once)
- **diet:** “omnivore” appears 2 times (others appear only once)
- **blood:** “hot” appears in every row (4 times)
- **body hair:** “yes” appears in every row (4 times)
- **eggs:** “false” appears 3 times (while “true” appears once)
- **breastfeeding:** “indeed” appears in every row (4 times)

Since we are instructed to format the output as a Python set of strings (and exclude the column names), the result is:

set_length1={{"4"},{"omnivore"},{"hot"},{"yes"},{"false"},{"indeed"}}

We start by treating the first row as header labels and the remaining rows as transactions. Each transaction is a set of items (the cell values). With four transactions, we list the items in each column (ignoring “name” since each is unique):

- **legs count:**
    - "4" appears in 3 transactions (rabbit, fox, platypus)
    - "2" appears only once
- **diet:**
    - "herbivore" appears once
    - "omnivore" appears twice (human, fox)
    - "carnivore" appears once
- **blood:**
    - "hot" appears in all 4 transactions
- **body hair:**
    - "yes" appears in all 4 transactions
- **eggs:**
    - "false" appears in 3 transactions
    - "true" appears once
- **breastfeeding:**
    - "indeed" appears in all 4 transactions

Now, the frequent itemsets of length 1 (with support ≥ 2) are the items that occur in at least two transactions:

- "4"
- "omnivore"
- "hot"
- "yes"
- "false"
- "indeed"

Formatted as a Python set of single-item sets (as strings), the answer is:

set_length1 = {{"4"}, {"omnivore"}, {"hot"}, {"yes"}, {"false"}, {"indeed"}}

We begin by treating each row (ignoring the header) as a transaction where each cell value (except the unique “name” values) is an item. We then count the number of transactions in which each item appears. With minimal support set to 2, an item is “frequent” if it occurs in at least 2 transactions.

Let’s inspect the columns (other than “name” which has unique values):

- **legs count:**
  • “4” appears in 3 rows (rabbit, fox, platypus)
  • “2” appears only once
  → Frequent: **"4"**
- **diet:**
  • “herbivore” appears once
  • “omnivore” appears in 2 rows (human, fox)
  • “carnivore” appears once
  → Frequent: **"omnivore"**
- **blood:**
  • “hot” appears in all 4 rows
  → Frequent: **"hot"**
- **body hair:**
  • “yes” appears in all 4 rows
  → Frequent: **"yes"**
- **eggs:**
  • “false” appears in 3 rows
  • “true” appears once
  → Frequent: **"false"**
- **breastfeeding:**
  • “indeed” appears in all 4 rows
  → Frequent: **"indeed"**

Thus, the frequent itemsets of length 1 (items with minimal support ≥ 2) are:

set_length1 = {{"4"}, {"omnivore"}, {"hot"}, {"yes"}, {"false"}, {"indeed"}}

We start by noting that the CSV’s first row contains the column names, so each subsequent row represents one transaction. The items are simply the attribute values (without the column names). For an itemset of length 1, we need to list all unique items that occur in at least 2 transactions (i.e. have a minimal support of 2).

Let’s examine each column:

- **name:**
  - rabbit, human, fox, platypus
  - *None* of these are repeated, so no frequent item here.
- **legs count:**
  - Values: 4, 2, 4, 4
  - "4" appears 3 times (≥ 2), so it is frequent. "2" appears once.
- **diet:**
  - Values: herbivore, omnivore, omnivore, carnivore
  - "omnivore" appears 2 times (≥ 2), so it is frequent. The other values appear only once.
- **blood:**
  - All rows have "hot"
  - "hot" appears in all 4 transactions, so it is frequent.
- **body hair:**
  - All rows have "yes"
  - "yes" is frequent.
- **eggs:**
  - Values: false, false, false, true
  - "false" appears 3 times (≥ 2), so it is frequent. "true" is not.
- **breastfeeding:**
  - All rows have "indeed"
  - "indeed" is frequent.

Thus, the frequent itemsets of length 1 (items with support at least 2) are:

- "4"
- "omnivore"
- "hot"
- "yes"

- "false"
- "indeed"

Following the requested format (formatted as a Python set of items, for instance set={{"item 1"},{"item 2"}}), we have:

set_length1 = {{"4"}, {"omnivore"}, {"hot"}, {"yes"}, {"false"}, {"indeed"}}

Based on the provided CSV data (ignoring the “name” column), we first list each attribute with its corresponding value for every animal:

- **legs count:**
    - 4: rabbit, fox, platypus (support = 3)
    - 2: human (support = 1)
- **diet:**
    - herbivore: rabbit (support = 1)
    - omnivore: human, fox (support = 2)
    - carnivore: platypus (support = 1)
- **blood:**
    - hot: all rows (support = 4)
- **body hair:**
    - yes: all rows (support = 4)
- **eggs:**
    - false: rabbit, human, fox (support = 3)
    - true: platypus (support = 1)
- **breastfeeding:**
    - indeed: all rows (support = 4)

We then select only those itemsets (of length 1) that have a support of at least 2. These are:

- "legs count=4"
- "diet=omnivore"
- "blood=hot"

- "body hair=yes"
- "eggs=false"
- "breastfeeding=indeed"

Finally, formatted as a Python set of singleton sets, the answer is:

set={{"legs count=4"}, {"diet=omnivore"}, {"blood=hot"}, {"body hair=yes"}, {"eggs=false"}, {"breastfeeding=indeed"}}

LENGTH 2

We start by treating each row (ignoring the “name” column, which merely identifies the row) as a transaction where every other column gives an attribute-value pair. For example, the “rabbit” row gives the following items:

- legs count=4
- diet=herbivore
- blood=hot
- body hair=yes
- eggs=false
- breastfeeding=indeed

Doing this for all four rows, we note that some attribute–value pairs occur only once (for instance, “diet=herbivore”, “diet=carnivore”, “legs count=2”, and “eggs=true”). These are not frequent when we require a minimal support of 2. The items that appear in at least 2 transactions are:

- legs count=4 (support 3: rabbit, fox, platypus)
- diet=omnivore (support 2: human, fox)
- blood=hot (support 4: all rows)
- body hair=yes (support 4: all rows)
- eggs=false (support 3: rabbit, human, fox)
- breastfeeding=indeed (support 4: all rows)

Now, we look for all pairs (itemsets of length 2) that occur together in at least 2 transactions. (Remember that if one of the two items appears only in a single row, the pair cannot reach the support threshold.) By examining the transactions we find the following frequent pairs:

1. {legs count=4, blood=hot}
2. {legs count=4, body hair=yes}
3. {legs count=4, eggs=false}
4. {legs count=4, breastfeeding=indeed}
5. {diet=omnivore, blood=hot}
6. {diet=omnivore, body hair=yes}
7. {diet=omnivore, eggs=false}
8. {diet=omnivore, breastfeeding=indeed}
9. {blood=hot, body hair=yes}
10. {blood=hot, eggs=false}
11. {blood=hot, breastfeeding=indeed}
12. {body hair=yes, eggs=false}
13. {body hair=yes, breastfeeding=indeed}
14. {eggs=false, breastfeeding=indeed}

Each of these pairs appears in at least 2 of the four transactions.

Following the required output format (a python-set formatted as a string, for example: set={{"item 1"},{"item 2"}}), we substitute the placeholder with the frequent itemsets. The final answer is:

ini

ZkopírovatUpravit

set_length2={{"legs count=4", "blood=hot"}, {"legs count=4", "body hair=yes"}, {"legs count=4", "eggs=false"}, {"legs count=4", "breastfeeding=indeed"}, {"diet=omnivore", "blood=hot"}, {"diet=omnivore", "body hair=yes"}, {"diet=omnivore", "eggs=false"}, {"diet=omnivore", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes"}, {"blood=hot", "eggs=false"}, {"blood=hot", "breastfeeding=indeed"}, {"body hair=yes", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed"}, {"eggs=false", "breastfeeding=indeed"}}

This is the complete set of all frequent itemsets of length 2 with a minimal support of 2.

Based on the CSV data, we first list all transactions (ignoring the “name” column because each value there is unique and would never reach the support threshold):

- **Row 1 (rabbit):**
  {legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 2 (human):**
  {legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 3 (fox):**
  {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 4 (platypus):**
  {legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

Since the minimal support is 2, we first identify the **frequent 1-itemsets** (only those appearing in at least 2 transactions). Notice that in the "legs count" column only the value 4 is frequent (appearing in rows 1, 3, and 4) while in "diet" only omnivore is frequent (rows 2 and 3). All other columns with constant or mostly constant values give:

- **legs count=4** (support 3)
- **diet=omnivore** (support 2)
- **blood=hot** (support 4)
- **body hair=yes** (support 4)
- **eggs=false** (support 3)
- **breastfeeding=indeed** (support 4)

Next, we find all pairs of these frequent items (length-2 itemsets) that occur together in at least 2 transactions. Here's a breakdown (using A, B, C, D, E, F as shorthand):

- **A:** legs count=4
- **B:** diet=omnivore
- **C:** blood=hot
- **D:** body hair=yes
- **E:** eggs=false
- **F:** breastfeeding=indeed

Transaction membership (only listing frequent items):

- **Transaction 1:** {A, C, D, E, F}

- **Transaction 2:** {B, C, D, E, F}
- **Transaction 3:** {A, B, C, D, E, F}
- **Transaction 4:** {A, C, D, F}

Counting co-occurrences gives the following frequent 2-itemsets (each with support ≥ 2):

1. {A, C} → {legs count=4, blood=hot}
2. {A, D} → {legs count=4, body hair=yes}
3. {A, E} → {legs count=4, eggs=false}
4. {A, F} → {legs count=4, breastfeeding=indeed}
5. {B, C} → {diet=omnivore, blood=hot}
6. {B, D} → {diet=omnivore, body hair=yes}
7. {B, E} → {diet=omnivore, eggs=false}
8. {B, F} → {diet=omnivore, breastfeeding=indeed}
9. {C, D} → {blood=hot, body hair=yes}
10. {C, E} → {blood=hot, eggs=false}
11. {C, F} → {blood=hot, breastfeeding=indeed}
12. {D, E} → {body hair=yes, eggs=false}
13. {D, F} → {body hair=yes, breastfeeding=indeed}
14. {E, F} → {eggs=false, breastfeeding=indeed}

Formatting the answer as a Python set of sets (each inner set represented with the two items as strings) yields:

makefile

ZkopírovatUpravit

```
set_length2 = {{"legs count=4", "blood=hot"},
        {"legs count=4", "body hair=yes"},
        {"legs count=4", "eggs=false"},
        {"legs count=4", "breastfeeding=indeed"},
        {"diet=omnivore", "blood=hot"},
        {"diet=omnivore", "body hair=yes"},
```

{"diet=omnivore", "eggs=false"},

{"diet=omnivore", "breastfeeding=indeed"},

{"blood=hot", "body hair=yes"},

{"blood=hot", "eggs=false"},

{"blood=hot", "breastfeeding=indeed"},

{"body hair=yes", "eggs=false"},

{"body hair=yes", "breastfeeding=indeed"},

{"eggs=false", "breastfeeding=indeed"}}

This is the final answer with all length-2 frequent itemsets having a minimal support of 2.

Below is the answer with all length-2 frequent itemsets (with support ≥ 2) inserted as a Python-set formatted string:

ini

ZkopírovatUpravit

set_length2={{"legs count=4", "blood=hot"}, {"legs count=4", "body hair=yes"}, {"legs count=4", "eggs=false"}, {"legs count=4", "breastfeeding=indeed"}, {"diet=omnivore", "blood=hot"}, {"diet=omnivore", "body hair=yes"}, {"diet=omnivore", "eggs=false"}, {"diet=omnivore", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes"}, {"blood=hot", "eggs=false"}, {"blood=hot", "breastfeeding=indeed"}, {"body hair=yes", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed"}, {"eggs=false", "breastfeeding=indeed"}}

Explanation

1. **Dataset and Transactions:**

The CSV has 4 rows (after the header). Each row is considered a transaction whose items are the attribute-value pairs. For example, the first transaction (rabbit) contains:

    - legs count=4
    - diet=herbivore
    - blood=hot
    - body hair=yes
    - eggs=false
    - breastfeeding=indeed

2. **Frequent Items (Support ≥ 2):**

Items that appear in at least 2 transactions (ignoring the unique name attribute) are:

  - "legs count=4" (appears in 3 transactions)
  - "diet=omnivore" (appears in 2 transactions)
  - "blood=hot" (appears in all 4 transactions)
  - "body hair=yes" (appears in all 4 transactions)
  - "eggs=false" (appears in 3 transactions)
  - "breastfeeding=indeed" (appears in all 4 transactions)

3. **Finding Frequent Itemsets of Length 2:**

All pairs (2-item combinations) of the frequent items were examined. Only those pairs that co-occur in at least 2 transactions are kept. For example:

  - The pair {"legs count=4", "blood=hot"} appears in the transactions for rabbit, fox, and platypus (3 times).
  - The pair {"diet=omnivore", "blood=hot"} appears in the transactions for human and fox (2 times).
  - All other pairs were computed similarly.

The final set of 14 itemsets is shown above, formatted exactly as a Python set of sets.

We start by treating each row (after the header) as a “transaction” where each item is given by the column name together with its value. For example, the first row gives the items

- **name=rabbit**
- **legs count=4**
- **diet=herbivore**
- **blood=hot**
- **body hair=yes**
- **eggs=false**
- **breastfeeding=indeed**

Because “name” is just an identifier and appears uniquely in every row, only the other attribute-value pairs can appear in more than one transaction. In our CSV the items that appear in at least 2 transactions are:

- **legs count=4** (support = 3)
- **diet=omnivore** (support = 2)
- **blood=hot** (support = 4)
- **body hair=yes** (support = 4)
- **eggs=false** (support = 3)
- **breastfeeding=indeed** (support = 4)

Next, we consider all pairs (itemsets of length 2) from these frequent items and count in how many transactions they co-occur. For instance, “legs count=4” occurs in the transactions for rabbit, fox, and platypus; “blood=hot” appears in every transaction. Checking each candidate pair we find that the following 14 itemsets appear together in at least 2 transactions:

- {legs count=4, blood=hot} (support = 3)
- {legs count=4, body hair=yes} (support = 3)
- {legs count=4, eggs=false} (support = 2)
- {legs count=4, breastfeeding=indeed} (support = 3)
- {diet=omnivore, blood=hot} (support = 2)
- {diet=omnivore, body hair=yes} (support = 2)
- {diet=omnivore, eggs=false} (support = 2)
- {diet=omnivore, breastfeeding=indeed} (support = 2)
- {blood=hot, body hair=yes} (support = 4)
- {blood=hot, eggs=false} (support = 3)
- {blood=hot, breastfeeding=indeed} (support = 4)
- {body hair=yes, eggs=false} (support = 3)
- {body hair=yes, breastfeeding=indeed} (support = 4)
- {eggs=false, breastfeeding=indeed} (support = 3)

Following the formatting instructions (“formatted as python set, all formatted as string”) the final result is:

ini

ZkopírovatUpravit

set_length2 = {{"legs count=4", "blood=hot"}, {"legs count=4", "body hair=yes"}, {"legs count=4", "eggs=false"}, {"legs count=4", "breastfeeding=indeed"}, {"diet=omnivore", "blood=hot"}, {"diet=omnivore", "body hair=yes"}, {"diet=omnivore", "eggs=false"}, {"diet=omnivore", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes"}, {"blood=hot", "eggs=false"}, {"blood=hot", "breastfeeding=indeed"}, {"body hair=yes", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed"}, {"eggs=false", "breastfeeding=indeed"}}

This is the set of all frequent itemsets (of length 2) with minimal support equal to 2.

set_length2={{"legs count=4", "blood=hot"}, {"legs count=4", "body hair=yes"}, {"legs count=4", "eggs=false"}, {"legs count=4", "breastfeeding=indeed"}, {"diet=omnivore", "blood=hot"}, {"diet=omnivore", "body hair=yes"}, {"diet=omnivore", "eggs=false"}, {"diet=omnivore", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes"}, {"blood=hot", "eggs=false"}, {"blood=hot", "breastfeeding=indeed"}, {"body hair=yes", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed"}, {"eggs=false", "breastfeeding=indeed"}}

LENGTH 3

Below is the result after manually “mining” the dataset for 3-item combinations that appear in at least 2 rows. In this context, each “item” is an attribute–value pair (ignoring the name column), and we only consider those items that occur in at least 2 transactions. (For example, “blood=hot”, “body hair=yes” and “breastfeeding=indeed” appear in every row; “legs count=4” appears in three rows; “diet=omnivore” appears twice; and “eggs=false” appears three times.)

The four transactions (ignoring the name) are:

- Transaction 1 (rabbit): {legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Transaction 2 (human): {legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Transaction 3 (fox): {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Transaction 4 (platypus): {legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

After discarding the items that occur only once (such as “legs count=2”, “diet=herbivore”, “diet=carnivore”, “eggs=true”), the candidate frequent items become:

- blood=hot (support 4)
- body hair=yes (support 4)

- breastfeeding=indeed (support 4)
- legs count=4 (support 3)
- diet=omnivore (support 2)
- eggs=false (support 3)

Next, we consider every 3-item combination (ensuring that each combination contains items from different attributes so that they can actually appear together in a row) and count in how many transactions that particular triple occurs. For example, the set {blood=hot, body hair=yes, breastfeeding=indeed} appears in all 4 transactions. By manually counting the support for every candidate combination, we find that the following 16 itemsets (each of length 3) meet the minimum support threshold (≥2):

1. {blood=hot, body hair=yes, breastfeeding=indeed}
2. {blood=hot, body hair=yes, legs count=4}
3. {blood=hot, body hair=yes, diet=omnivore}
4. {blood=hot, body hair=yes, eggs=false}
5. {blood=hot, breastfeeding=indeed, legs count=4}
6. {blood=hot, breastfeeding=indeed, diet=omnivore}
7. {blood=hot, breastfeeding=indeed, eggs=false}
8. {blood=hot, legs count=4, eggs=false}
9. {blood=hot, diet=omnivore, eggs=false}
10. {body hair=yes, breastfeeding=indeed, legs count=4}
11. {body hair=yes, breastfeeding=indeed, diet=omnivore}
12. {body hair=yes, breastfeeding=indeed, eggs=false}
13. {body hair=yes, legs count=4, eggs=false}
14. {body hair=yes, diet=omnivore, eggs=false}
15. {breastfeeding=indeed, legs count=4, eggs=false}
16. {breastfeeding=indeed, diet=omnivore, eggs=false}

Finally, following the requested format (a python-like set of sets represented as a string, with each itemset enclosed in braces and all enclosed within an outer set), we have:

makefile

ZkopírovatUpravit

```
set_length3={{"blood=hot","body hair=yes","breastfeeding=indeed"},
```

{"blood=hot","body hair=yes","legs count=4"},

{"blood=hot","body hair=yes","diet=omnivore"},

{"blood=hot","body hair=yes","eggs=false"},

{"blood=hot","breastfeeding=indeed","legs count=4"},

{"blood=hot","breastfeeding=indeed","diet=omnivore"},

{"blood=hot","breastfeeding=indeed","eggs=false"},

{"blood=hot","legs count=4","eggs=false"},

{"blood=hot","diet=omnivore","eggs=false"},

{"body hair=yes","breastfeeding=indeed","legs count=4"},

{"body hair=yes","breastfeeding=indeed","diet=omnivore"},

{"body hair=yes","breastfeeding=indeed","eggs=false"},

{"body hair=yes","legs count=4","eggs=false"},

{"body hair=yes","diet=omnivore","eggs=false"},

{"breastfeeding=indeed","legs count=4","eggs=false"},

{"breastfeeding=indeed","diet=omnivore","eggs=false"}}

This string represents the set of all frequent 3-itemsets from the dataset (with minimum support 2) as required.

Below is the answer obtained by “manually” (i.e. without any programming language) scanning the CSV table. (Recall that each transaction is given by a row with the header defining its items; we do not consider the “name” column.) We first note the following from the CSV:

- **Transactions (ignoring “name”):**
  - T_1: {legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
  - T_2: {legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
  - T_3: {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
  - T_4: {legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

- **Items that are individually "frequent" (support ≥ 2):**
  - • blood=hot (support 4)
  - • body hair=yes (support 4)
  - • breastfeeding=indeed (support 4)
  - • legs count=4 (support 3)
  - • diet=omnivore (support 2)
  - • eggs=false (support 3)

Any item that is not frequent (e.g. legs count=2, diet=herbivore, diet=carnivore, eggs=true) cannot form part of a 3-item combination that appears in at least two transactions.

When considering all possible 3-item combinations from the six "frequent" items, we check in how many transactions they appear together (note that a transaction carries exactly one value per attribute). For example, the combination
{blood=hot, body hair=yes, breastfeeding=indeed}
appears in every transaction (support 4) because those items are common to all.

After "examining" all 20 possible 3-combinations (from the set {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, diet=omnivore, eggs=false}), we found that the following 16 itemsets occur in at least two transactions:

- {blood=hot, body hair=yes, breastfeeding=indeed} (support 4)
- {blood=hot, body hair=yes, legs count=4} (support 3)
- {blood=hot, body hair=yes, diet=omnivore} (support 2)
- {blood=hot, body hair=yes, eggs=false} (support 3)
- {blood=hot, breastfeeding=indeed, legs count=4} (support 3)
- {blood=hot, breastfeeding=indeed, diet=omnivore} (support 2)
- {blood=hot, breastfeeding=indeed, eggs=false} (support 3)
- {blood=hot, legs count=4, eggs=false} (support 2)
- {blood=hot, diet=omnivore, eggs=false} (support 2)
- {body hair=yes, breastfeeding=indeed, legs count=4} (support 3)
- {body hair=yes, breastfeeding=indeed, diet=omnivore} (support 2)
- {body hair=yes, breastfeeding=indeed, eggs=false} (support 3)
- {body hair=yes, legs count=4, eggs=false} (support 2)
- {body hair=yes, diet=omnivore, eggs=false} (support 2)

- {breastfeeding=indeed, legs count=4, eggs=false} (support 2)
- {breastfeeding=indeed, diet=omnivore, eggs=false} (support 2)

In the requested format (a python set of sets, each inner set represented as a set of strings), we replace the placeholder <<itemsets>> with the following string:

ini

ZkopírovatUpravit

set_length3={{"blood=hot", "body hair=yes", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes", "legs count=4"}, {"blood=hot", "body hair=yes", "diet=omnivore"}, {"blood=hot", "body hair=yes", "eggs=false"}, {"blood=hot", "breastfeeding=indeed", "legs count=4"}, {"blood=hot", "breastfeeding=indeed", "diet=omnivore"}, {"blood=hot", "breastfeeding=indeed", "eggs=false"}, {"blood=hot", "legs count=4", "eggs=false"}, {"blood=hot", "diet=omnivore", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed", "legs count=4"}, {"body hair=yes", "breastfeeding=indeed", "diet=omnivore"}, {"body hair=yes", "breastfeeding=indeed", "eggs=false"}, {"body hair=yes", "legs count=4", "eggs=false"}, {"body hair=yes", "diet=omnivore", "eggs=false"}, {"breastfeeding=indeed", "legs count=4", "eggs=false"}, {"breastfeeding=indeed", "diet=omnivore", "eggs=false"}}

This is the final answer with the itemsets formatted exactly as required.

Let's first list the transactions (ignoring the “name” column) for each row using the header as attribute names:

- **rabbit:** {"4", "herbivore", "hot", "yes", "false", "indeed"}
- **human:** {"2", "omnivore", "hot", "yes", "false", "indeed"}
- **fox:** {"4", "omnivore", "hot", "yes", "false", "indeed"}
- **platypus:** {"4", "carnivore", "hot", "yes", "true", "indeed"}

We then note which items are *frequent* on their own (appear in at least 2 transactions):

- **legs count:** "4" (appears in rabbit, fox, platypus; support = 3) while "2" (human) is infrequent.
- **diet:** "omnivore" (human, fox; support = 2) but “herbivore” and “carnivore” each appear only once.
- **blood:** "hot" (all 4 rows; support = 4)
- **body hair:** "yes" (all 4 rows; support = 4)

- **eggs:** "false" (rabbit, human, fox; support = 3) while "true" (platypus) is infrequent.
- **breastfeeding:** "indeed" (all 4 rows; support = 4)

When building *itemsets* of length 3 we must choose items that together appear in at least 2 transactions. Although “omnivore” is frequent (support = 2), combining it with any other item might lower support because it only appears in human and fox. In fact, if we try to build a three-itemset containing “omnivore” and “4”, the support would be only 1 (since “4” does not appear in human). Thus, we focus on items that are common to more transactions.

The best choice is to use the items that occur in almost every transaction:

- "4" (appears in T1, T3, T4)
- "hot" (appears in all transactions)
- "yes" (appears in all transactions)
- "false" (appears in T1, T2, T3)
- "indeed" (appears in all transactions)

Now, list the transactions using only these items:

- **T1 (rabbit):** {"4", "hot", "yes", "false", "indeed"}
- **T2 (human):** {"hot", "yes", "false", "indeed"}
- **T3 (fox):** {"4", "hot", "yes", "false", "indeed"}
- **T4 (platypus):** {"4", "hot", "yes", "indeed"}

Every 3-item combination (from these 5 items) is checked by intersecting across transactions:

1. **{"4", "hot", "yes"}:** Appears in T1, T3, and T4 (support = 3)
2. **{"4", "hot", "false"}:** Appears in T1 and T3 (support = 2)
3. **{"4", "hot", "indeed"}:** Appears in T1, T3, and T4 (support = 3)
4. **{"4", "yes", "false"}:** Appears in T1 and T3 (support = 2)
5. **{"4", "yes", "indeed"}:** Appears in T1, T3, and T4 (support = 3)
6. **{"4", "false", "indeed"}:** Appears in T1 and T3 (support = 2)
7. **{"hot", "yes", "false"}:** Appears in T1, T2, and T3 (support = 3)
8. **{"hot", "yes", "indeed"}:** Appears in all transactions (support = 4)

9. **{"hot", "false", "indeed"}:** Appears in T1, T2, and T3 (support = 3)
10. **{"yes", "false", "indeed"}:** Appears in T1, T2, and T3 (support = 3)

Each of these 10 candidate 3-itemsets has support ≥ 2.

Per the problem's instructions, we need to present the answer as a python set of sets (formatted as string) like this:

ini

ZkopírovatUpravit

set={{"item 1"},{"item 2"}}

Thus, replacing <<itemsets>> with our computed itemsets, we have:

ini

ZkopírovatUpravit

set={{"4", "hot", "yes"}, {"4", "hot", "false"}, {"4", "hot", "indeed"}, {"4", "yes", "false"}, {"4", "yes", "indeed"}, {"4", "false", "indeed"}, {"hot", "yes", "false"}, {"hot", "yes", "indeed"}, {"hot", "false", "indeed"}, {"yes", "false", "indeed"}}

We start by "translating" the CSV rows into transactions, where each transaction contains the attribute–value pairs (ignoring the "name" attribute):

- **Transaction 1 (rabbit):**
  {"legs count=4", "diet=herbivore", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}
- **Transaction 2 (human):**
  {"legs count=2", "diet=omnivore", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}
- **Transaction 3 (fox):**
  {"legs count=4", "diet=omnivore", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}
- **Transaction 4 (platypus):**
  {"legs count=4", "diet=carnivore", "blood=hot", "body hair=yes", "eggs=true", "breastfeeding=indeed"}

The first step is to count the support (the number of transactions in which an item appears) for each attribute–value pair. Items that appear in at least 2 transactions (i.e. minimal support of 2) are:

- **legs count:**
    - “legs count=4” appears in transactions 1, 3, and 4 (support = 3)
    - “legs count=2” appears only in transaction 2 (support = 1) → *discarded*
- **diet:**
    - “diet=omnivore” appears in transactions 2 and 3 (support = 2)
    - “diet=herbivore” (only transaction 1) and “diet=carnivore” (only transaction 4) → *discarded*
- **blood:** “blood=hot” appears in all transactions (support = 4)
- **body hair:** “body hair=yes” appears in all transactions (support = 4)
- **eggs:**
    - “eggs=false” appears in transactions 1, 2, and 3 (support = 3)
    - “eggs=true” appears only in transaction 4 (support = 1) → *discarded*
- **breastfeeding:** “breastfeeding=indeed” appears in all transactions (support = 4)

Thus, the only frequent single items are:
{ "legs count=4", "diet=omnivore", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed" }.

Next, we want **frequent itemsets of length 3** (i.e. 3-item combinations) that appear in at least 2 transactions. For each candidate 3-itemset we “intersect” the transactions in which its items appear. For example, the itemset
{"legs count=4", "blood=hot", "body hair=yes"}
has support equal to the number of transactions where all three occur. Since “blood=hot” and “body hair=yes” are in every transaction and “legs count=4” is in transactions 1, 3, and 4, their intersection is {1, 3, 4} (support = 3), so it is frequent.

After checking all C(6,3) = 20 possible combinations, the only 3-itemsets that are *not* frequent are those that include both “legs count=4” and “diet=omnivore” because “diet=omnivore” appears only in transactions 2 and 3. When intersected with “legs count=4” (which appears in 1, 3, 4) the only common transaction is transaction 3, yielding support = 1. This issue occurs for the following four combinations:

- {"legs count=4", "blood=hot", "diet=omnivore"}
- {"legs count=4", "body hair=yes", "diet=omnivore"}
- {"legs count=4", "breastfeeding=indeed", "diet=omnivore"}
- {"legs count=4", "eggs=false", "diet=omnivore"}

All the remaining 16 three-itemsets are frequent (each appears in at least 2 transactions). These are:

1. {"legs count=4", "blood=hot", "body hair=yes"}
2. {"legs count=4", "blood=hot", "breastfeeding=indeed"}
3. {"legs count=4", "blood=hot", "eggs=false"}
4. {"legs count=4", "body hair=yes", "breastfeeding=indeed"}
5. {"legs count=4", "body hair=yes", "eggs=false"}
6. {"legs count=4", "breastfeeding=indeed", "eggs=false"}
7. {"blood=hot", "body hair=yes", "breastfeeding=indeed"}
8. {"blood=hot", "body hair=yes", "eggs=false"}
9. {"blood=hot", "body hair=yes", "diet=omnivore"}
10. {"blood=hot", "breastfeeding=indeed", "eggs=false"}
11. {"blood=hot", "breastfeeding=indeed", "diet=omnivore"}
12. {"blood=hot", "eggs=false", "diet=omnivore"}
13. {"body hair=yes", "breastfeeding=indeed", "eggs=false"}
14. {"body hair=yes", "breastfeeding=indeed", "diet=omnivore"}
15. {"body hair=yes", "eggs=false", "diet=omnivore"}
16. {"breastfeeding=indeed", "eggs=false", "diet=omnivore"}

Finally, we need to return the result as a Python set (formatted as a string) using the example formatting provided. Replacing the placeholder <<itemsets>> with the above frequent itemsets, the final output is:

ini

ZkopírovatUpravit

set_length3 = {{"legs count=4", "blood=hot", "body hair=yes"}, {"legs count=4", "blood=hot", "breastfeeding=indeed"}, {"legs count=4", "blood=hot", "eggs=false"}, {"legs count=4", "body hair=yes", "breastfeeding=indeed"}, {"legs count=4", "body hair=yes", "eggs=false"}, {"legs count=4", "breastfeeding=indeed", "eggs=false"}, {"blood=hot", "body hair=yes", "breastfeeding=indeed"}, {"blood=hot", "body hair=yes", "eggs=false"}, {"blood=hot", "body hair=yes", "diet=omnivore"}, {"blood=hot", "breastfeeding=indeed", "eggs=false"}, {"blood=hot", "breastfeeding=indeed", "diet=omnivore"}, {"blood=hot", "eggs=false", "diet=omnivore"}, {"body hair=yes", "breastfeeding=indeed", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed", "diet=omnivore"}, {"body hair=yes", "eggs=false", "diet=omnivore"}, {"breastfeeding=indeed", "eggs=false", "diet=omnivore"}}

This is the final answer, with each 3-itemset represented as a set (and the whole collection as a Python set literal).

Below is the answer reached by reasoning "by hand" (without using a programming language) on the given CSV. Recall that each item is interpreted as a "column=value" pair (except for the name), and we only include those pairs that appear in at least 2 rows. In the CSV the following pairs are frequent (support ≥ 2):

- **legs count:** only "legs count=4" appears at least twice (rows 1, 3, and 4).
- **diet:** only "diet=omnivore" appears at least twice (rows 2 and 3).
- **blood:** "blood=hot" appears in every row.
- **body hair:** "body hair=yes" appears in every row.
- **eggs:** only "eggs=false" appears in rows 1, 2, and 3.
- **breastfeeding:** "breastfeeding=indeed" appears in every row.

Each transaction (ignoring the animal name) then is:

- Row1: {legs count=4, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Row2: {diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Row3: {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- Row4: {legs count=4, blood=hot, body hair=yes, breastfeeding=indeed}

A "length 3" itemset is any set of three items that appears (as a subset of a row's items) in at least 2 rows. By examining the transactions, we find these 16 frequent triples:

1. {"blood=hot", "body hair=yes", "breastfeeding=indeed"}
2. {"legs count=4", "blood=hot", "body hair=yes"}
3. {"legs count=4", "blood=hot", "breastfeeding=indeed"}
4. {"legs count=4", "blood=hot", "eggs=false"}
5. {"legs count=4", "body hair=yes", "breastfeeding=indeed"}
6. {"legs count=4", "body hair=yes", "eggs=false"}

7. {“legs count=4”, “breastfeeding=indeed”, “eggs=false”}
8. {“blood=hot”, “body hair=yes”, “eggs=false”}
9. {“blood=hot”, “breastfeeding=indeed”, “eggs=false”}
10. {“body hair=yes”, “breastfeeding=indeed”, “eggs=false”}
11. {“diet=omnivore”, “blood=hot”, “body hair=yes”}
12. {“diet=omnivore”, “blood=hot”, “breastfeeding=indeed”}
13. {“diet=omnivore”, “blood=hot”, “eggs=false”}
14. {“diet=omnivore”, “body hair=yes”, “breastfeeding=indeed”}
15. {“diet=omnivore”, “body hair=yes”, “eggs=false”}
16. {“diet=omnivore”, “breastfeeding=indeed”, “eggs=false”}

(For example, the triple {“legs count=4”, “blood=hot”, “eggs=false”} appears in row1 and row3; the triple {“diet=omnivore”, “blood=hot”, “body hair=yes”} appears in rows 2 and 3; and so on.)

The final answer (with the itemsets formatted as a Python set of sets, each inner set shown with double-quoted strings) is:

makefile

ZkopírovatUpravit

```
set_length3 = {{"blood=hot", "body hair=yes", "breastfeeding=indeed"},
        {"legs count=4", "blood=hot", "body hair=yes"},
        {"legs count=4", "blood=hot", "breastfeeding=indeed"},
        {"legs count=4", "blood=hot", "eggs=false"},
        {"legs count=4", "body hair=yes", "breastfeeding=indeed"},
        {"legs count=4", "body hair=yes", "eggs=false"},
        {"legs count=4", "breastfeeding=indeed", "eggs=false"},
        {"blood=hot", "body hair=yes", "eggs=false"},
        {"blood=hot", "breastfeeding=indeed", "eggs=false"},
        {"body hair=yes", "breastfeeding=indeed", "eggs=false"},
        {"diet=omnivore", "blood=hot", "body hair=yes"},
        {"diet=omnivore", "blood=hot", "breastfeeding=indeed"},
        {"diet=omnivore", "blood=hot", "eggs=false"},
        {"diet=omnivore", "body hair=yes", "breastfeeding=indeed"},
        {"diet=omnivore", "body hair=yes", "eggs=false"},
        {"diet=omnivore", "breastfeeding=indeed", "eggs=false"}}
```

This string is your final answer with the placeholder <<itemsets>> replaced by the computed frequent itemsets of length 3.

LENGTH 4

Below is the final answer. Notice that we first identified the columns (ignoring the "name" attribute) and then looked for those 4-item combinations (using only items that occur in at least 2 rows) whose common appearance (support) is at least 2. In our CSV, the following items are frequent as singletons:

- **legs count=4** (present in rabbit, fox, platypus),
- **diet=omnivore** (human, fox),
- **blood=hot** (all rows),
- **body hair=yes** (all rows),
- **eggs=false** (rabbit, human, fox),
- **breastfeeding=indeed** (all rows).

Any 4-item combination from the above that appears in at least 2 rows is considered frequent. After checking the intersections manually, we obtain the following nine frequent itemsets of length 4:

makefile

ZkopírovatUpravit

```
set_length4 = {{"legs count=4", "blood=hot", "body hair=yes", "eggs=false"},
        {"legs count=4", "blood=hot", "body hair=yes", "breastfeeding=indeed"},
        {"legs count=4", "blood=hot", "eggs=false", "breastfeeding=indeed"},
        {"legs count=4", "body hair=yes", "eggs=false", "breastfeeding=indeed"},
        {"diet=omnivore", "blood=hot", "body hair=yes", "eggs=false"},
        {"diet=omnivore", "blood=hot", "body hair=yes", "breastfeeding=indeed"},
        {"diet=omnivore", "blood=hot", "eggs=false", "breastfeeding=indeed"},
```

{"diet=omnivore", "body hair=yes", "eggs=false", "breastfeeding=indeed"},

{"blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}}

Each inner set represents a frequent itemset (with minimal support equal to 2) and all are formatted as a python set of strings.

Below is the reasoning without any code.

Step 1. Interpret the CSV as transactions:
Treating the header row as the names of the columns, we use each subsequent row as a transaction. We ignore the first column (the animal "name") and work with the remaining six attribute values. Thus, the transactions become:

- **Transaction 1 (rabbit):** {4, herbivore, hot, yes, false, indeed}
- **Transaction 2 (human):** {2, omnivore, hot, yes, false, indeed}
- **Transaction 3 (fox):** {4, omnivore, hot, yes, false, indeed}
- **Transaction 4 (platypus):** {4, carnivore, hot, yes, true, indeed}

Step 2. Identify frequent items (minimum support = 2):
We check each attribute value (ignoring the animal name):

- **"hot"** appears in all 4 transactions.
- **"yes"** appears in all 4.
- **"indeed"** appears in all 4.
- **"false"** appears in Transactions 1, 2, and 3.
- **"4"** appears in Transactions 1, 3, and 4.
- **"omnivore"** appears in Transactions 2 and 3.

(The values "herbivore", "2", "carnivore", and "true" occur only once, so they are not frequent.)

So the frequent items are:
{ "4", "hot", "yes", "indeed", "false", "omnivore" }.

Step 3. Determine all 4-itemsets from the frequent items:
We now list every possible combination (without considering order) of 4 items drawn from the set above. There are 15 potential combinations (since 6 choose 4 = 15). We then check in which transactions each combination occurs (by seeing if all its items are present in the transaction's intersection with our frequent items).

After checking each candidate manually, the ones that occur in at least 2 transactions (support ≥ 2) are:

1. **{"4", "hot", "yes", "indeed"}**
   - Found in Transactions 1, 3, and 4.
2. **{"4", "hot", "yes", "false"}**
   - Found in Transactions 1 and 3.
3. **{"4", "hot", "indeed", "false"}**
   - Found in Transactions 1 and 3.
4. **{"4", "yes", "indeed", "false"}**
   - Found in Transactions 1 and 3.
5. **{"hot", "yes", "indeed", "false"}**
   - Found in Transactions 1, 2, and 3.
6. **{"hot", "yes", "indeed", "omnivore"}**
   - Found in Transactions 2 and 3.
7. **{"hot", "yes", "false", "omnivore"}**
   - Found in Transactions 2 and 3.
8. **{"hot", "indeed", "false", "omnivore"}**
   - Found in Transactions 2 and 3.
9. **{"yes", "indeed", "false", "omnivore"}**
   - Found in Transactions 2 and 3.

Step 4. Format the answer as a Python set of sets (as a string):

Following the provided format example (set={{"item 1"},{"item 2"}}) and not using column names or extra parentheses, the answer is:

ini

ZkopírovatUpravit

```
set_length4={{"4", "hot", "yes", "indeed"}, {"4", "hot", "yes", "false"}, {"4", "hot", "indeed", "false"}, {"4", "yes", "indeed", "false"}, {"hot", "yes", "indeed", "false"}, {"hot", "yes", "indeed", "omnivore"}, {"hot", "yes", "false", "omnivore"}, {"hot", "indeed", "false", "omnivore"}, {"yes", "indeed", "false", "omnivore"}}
```

This is the final answer with all frequent 4-itemsets (minimal support = 2) formatted as requested.

We start by treating each row (except for the “name” column) as a transaction where each cell is an item (with its attribute and value combined, e.g. "legs count=4"). Then we only consider items that are “frequent” by themselves (i.e. occur in at least 2 transactions). In our CSV, the transactions are:

- **Row 1 (rabbit):** {legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 2 (human):** {legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 3 (fox):** {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Row 4 (platypus):** {legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

Counting each item over the rows shows that only the following items appear in at least 2 transactions:

- **"blood=hot"** (in all 4 transactions)
- **"body hair=yes"** (in all 4 transactions)
- **"breastfeeding=indeed"** (in all 4 transactions)
- **"legs count=4"** (in rows 1, 3, and 4)
- **"eggs=false"** (in rows 1, 2, and 3)
- **"diet=omnivore"** (in rows 2 and 3)

(Items such as “legs count=2”, “diet=herbivore”, “diet=carnivore”, and “eggs=true” are not frequent, so they are ignored.)

Let’s assign a short label for clarity:

- A = "blood=hot"
- B = "body hair=yes"
- C = "breastfeeding=indeed"
- D = "legs count=4"
- E = "eggs=false"
- F = "diet=omnivore"

Rewriting the transactions with only frequent items:

- **T1 (rabbit):** {A, B, C, D, E}

- **T2 (human):** {A, B, C, E, F}
- **T3 (fox):** {A, B, C, D, E, F}
- **T4 (platypus):** {A, B, C, D}

Now, we need to find all itemsets of exactly 4 items that occur in at least 2 transactions (minimal support = 2). By checking all 4-combinations from the 6 frequent items, we find that the following 9 itemsets meet the condition:

1. {A, B, C, D}
2. {A, B, C, E}
3. {A, B, C, F}
4. {A, B, D, E}
5. {A, B, E, F}
6. {A, C, D, E}
7. {A, C, E, F}
8. {B, C, D, E}
9. {B, C, E, F}

Mapping the labels back to their original items, the frequent 4-itemsets are:

- {"blood=hot", "body hair=yes", "breastfeeding=indeed", "legs count=4"}
- {"blood=hot", "body hair=yes", "breastfeeding=indeed", "eggs=false"}
- {"blood=hot", "body hair=yes", "breastfeeding=indeed", "diet=omnivore"}
- {"blood=hot", "body hair=yes", "legs count=4", "eggs=false"}
- {"blood=hot", "body hair=yes", "eggs=false", "diet=omnivore"}
- {"blood=hot", "breastfeeding=indeed", "legs count=4", "eggs=false"}
- {"blood=hot", "breastfeeding=indeed", "eggs=false", "diet=omnivore"}
- {"body hair=yes", "breastfeeding=indeed", "legs count=4", "eggs=false"}
- {"body hair=yes", "breastfeeding=indeed", "eggs=false", "diet=omnivore"}

Finally, following the formatting instructions, we insert these itemsets (formatted as python sets) into our answer string. The final answer is:

ini

ZkopírovatUpravit

set_length4={{"blood=hot", "body hair=yes", "breastfeeding=indeed", "legs count=4"}, {"blood=hot", "body hair=yes", "breastfeeding=indeed", "eggs=false"}, {"blood=hot", "body hair=yes", "breastfeeding=indeed", "diet=omnivore"}, {"blood=hot", "body hair=yes", "legs count=4", "eggs=false"}, {"blood=hot", "body hair=yes", "eggs=false", "diet=omnivore"}, {"blood=hot", "breastfeeding=indeed", "legs count=4", "eggs=false"}, {"blood=hot", "breastfeeding=indeed", "eggs=false", "diet=omnivore"}, {"body hair=yes", "breastfeeding=indeed", "legs count=4", "eggs=false"}, {"body hair=yes", "breastfeeding=indeed", "eggs=false", "diet=omnivore"}}

Below is the solution worked out by hand. We first note that the CSV's first row gives the column names:

css

ZkopírovatUpravit

name,legs count,diet,blood,body hair,eggs,breastfeeding

Since the "name" attribute identifies the object (rabbit, human, etc.), we ignore it for our itemset mining. That leaves these six items (each expressed as "column=value"):

- legs count: can be "4" (appears in rabbit, fox, platypus) or "2" (only human). Only "4" is frequent (support ≥ 2).
- diet: "herbivore" (rabbit; support 1), "omnivore" (human and fox; support 2), "carnivore" (platypus; support 1). Only "omnivore" is frequent.
- blood: "hot" (in all 4 rows; support 4).
- body hair: "yes" (in all 4; support 4).
- eggs: "false" (in rabbit, human, fox; support 3) and "true" (only platypus; support 1). Only "false" is frequent.
- breastfeeding: "indeed" (all 4; support 4).

Thus the only frequent items (support ≥2) are:
A = legs count=4
B = diet=omnivore
C = blood=hot
D = body hair=yes
E = eggs=false
F = breastfeeding=indeed

There are 6 frequent items. In a set of 6, there are 15 possible 4-item combinations. However, note that any combination that contains both A (legs count=4) and B (diet=omnivore) will appear only in the one transaction (human or fox) that carries the one or the other – not both – except in the unique case of fox. (In fact, fox is the only one that has legs count=4 together with diet=omnivore, while human has omnivore but not legs count=4 and rabbit/platypus have legs count=4 but not omnivore.) Thus every 4–item candidate including both A and B has support 1, and is not frequent.

That leaves the 4–itemsets that do not contain both A and B. There are exactly 9 such itemsets, and each of them appears in at least 2 transactions. In detail, the 9 frequent 4–itemsets (each written as a set of “column=value” items) are:

- {legs count=4, blood=hot, body hair=yes, eggs=false}
- {legs count=4, blood=hot, body hair=yes, breastfeeding=indeed}
- {legs count=4, blood=hot, eggs=false, breastfeeding=indeed}
- {legs count=4, body hair=yes, eggs=false, breastfeeding=indeed}
- {diet=omnivore, blood=hot, body hair=yes, eggs=false}
- {diet=omnivore, blood=hot, body hair=yes, breastfeeding=indeed}
- {diet=omnivore, blood=hot, eggs=false, breastfeeding=indeed}
- {diet=omnivore, body hair=yes, eggs=false, breastfeeding=indeed}
- {blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}

Following the required format (a Python set whose elements are themselves sets written as strings without additional punctuation), we substitute <<itemsets>> with the above itemsets. The final answer is:

ini

ZkopírovatUpravit

```
set_length4 = {{"legs count=4, blood=hot, body hair=yes, eggs=false"}, {"legs count=4, blood=hot, body hair=yes, breastfeeding=indeed"}, {"legs count=4, blood=hot, eggs=false, breastfeeding=indeed"}, {"legs count=4, body hair=yes, eggs=false, breastfeeding=indeed"}, {"diet=omnivore, blood=hot, body hair=yes, eggs=false"}, {"diet=omnivore, blood=hot, body hair=yes, breastfeeding=indeed"}, {"diet=omnivore, blood=hot, eggs=false, breastfeeding=indeed"}, {"diet=omnivore, body hair=yes, eggs=false, breastfeeding=indeed"}, {"blood=hot, body hair=yes, eggs=false, breastfeeding=indeed"}}
```

This answer has been determined entirely by manual reasoning without using any programming language.

Based on the given CSV data (with the first row as headers) and by considering only the non-name attributes as the items in each transaction, the transactions are as follows:

- T_1 (rabbit): {4, herbivore, hot, yes, false, indeed}
- T_2 (human): {2, omnivore, hot, yes, false, indeed}
- T_3 (fox): {4, omnivore, hot, yes, false, indeed}
- T_4 (platypus): {4, carnivore, hot, yes, true, indeed}

Since the “name” field is unique per row, it is not useful for finding frequent patterns. We therefore consider only the other attributes. Notice that many attributes have the same value in every row (for example, blood is always “hot”, body hair is “yes”, and breastfeeding is always “indeed”), so these items automatically appear in every transaction.

After counting the frequency of each single item among the remaining attributes, only the following items are frequent (appear in at least 2 transactions):

- **"4"** (appears in T_1, T_3, T_4)
- **"omnivore"** (appears in T_2, T_3)
- **"hot"** (appears in all transactions)
- **"yes"** (appears in all transactions)
- **"false"** (appears in T_1, T_2, T_3)
- **"indeed"** (appears in all transactions)

We now form candidate itemsets of length 4 using only these frequent singletons. (Notice that items like “2”, “herbivore”, “carnivore”, and “true” are not included because they appear in only one transaction.) By checking the support (i.e. the number of transactions in which a candidate appears), we find that the following 4-item combinations appear in at least 2 transactions:

1. { "4", "hot", "yes", "false" }
    - Appears in T_1 and T_3.
2. { "4", "hot", "yes", "indeed" }
    - Appears in T_1, T_3, and T_4.
3. { "4", "hot", "false", "indeed" }
    - Appears in T_1 and T_3.
4. { "4", "yes", "false", "indeed" }

    - Appears in T_1 and T_3.

5. { "omnivore", "hot", "yes", "false" }
    - Appears in T_2 and T_3.
6. { "omnivore", "hot", "yes", "indeed" }
    - Appears in T_2 and T_3.
7. { "omnivore", "hot", "false", "indeed" }
    - Appears in T_2 and T_3.
8. { "omnivore", "yes", "false", "indeed" }
    - Appears in T_2 and T_3.
9. { "hot", "yes", "false", "indeed" }
    - Appears in T_1, T_2, and T_3.

Thus, formatted as a Python set (with each frequent 4-itemset enclosed in curly braces, as illustrated in the example), the answer is:

ini

ZkopírovatUpravit

```
set_length4 = {{"4", "hot", "yes", "false"}, {"4", "hot", "yes", "indeed"}, {"4", "hot", "false",
"indeed"}, {"4", "yes", "false", "indeed"}, {"omnivore", "hot", "yes", "false"}, {"omnivore",
"hot", "yes", "indeed"}, {"omnivore", "hot", "false", "indeed"}, {"omnivore", "yes", "false",
"indeed"}, {"hot", "yes", "false", "indeed"}}
```

LENGTH 5

We first “translate” each row (ignoring the “name” column, which is unique to each transaction) into a set of attribute–value items. For example, the “rabbit” row becomes:

- legs count=4
- diet=herbivore
- blood=hot
- body hair=yes
- eggs=false
- breastfeeding=indeed

Looking at all rows (rabbit, human, fox, platypus), you can see that the items “blood=hot”, “body hair=yes”, and “breastfeeding=indeed” appear in every row. This means any candidate itemset that includes these three will be supported in all transactions, and then we only need to add two additional items chosen from the remaining three attributes (legs count, diet, eggs) that have a common value in at least two transactions.

By comparing the transactions manually:

- **Rabbit and Fox:**
  • They both have **legs count=4** and **eggs=false** (though their diets differ: herbivore vs omnivore).
  • Together with the three common items, they share the 5-itemset:
  {“legs count=4”, “eggs=false”, “blood=hot”, “body hair=yes”, “breastfeeding=indeed”}
  This itemset appears in at least two transactions (rabbit and fox).
- **Human and Fox:**
  • They both have **diet=omnivore** and **eggs=false** (human has legs count=2 and fox has legs count=4, so “legs count” does not match).
  • With the three common items the resulting 5-itemset is:
  {“diet=omnivore”, “eggs=false”, “blood=hot”, “body hair=yes”, “breastfeeding=indeed”}
  This itemset is contained in both the human and fox rows.

Checking other pairs does not yield any additional 5–item combinations that are common in at least two transactions. (For example, comparing rabbit with human or fox with platypus, the intersection of attribute–value pairs is smaller than 5.)

Thus, the two frequent itemsets of length 5 with minimal support 2 are:

ini

ZkopírovatUpravit

```
set_length5 = {{"legs count=4", "eggs=false", "blood=hot", "body hair=yes", "breastfeeding=indeed"}, {"diet=omnivore", "eggs=false", "blood=hot", "body hair=yes", "breastfeeding=indeed"}}
```

Based on the CSV file, we first note that the header gives the attribute names:

css

ZkopírovatUpravit

```
name, legs count, diet, blood, body hair, eggs, breastfeeding
```

Ignoring the “name” (since it identifies the animal) and treating each remaining column as an item (using “column=value”), we have for each row:

- **rabbit**
  Items:
  • legs count=4
  • diet=herbivore
  • blood=hot
  • body hair=yes
  • eggs=false
  • breastfeeding=indeed
- **human**
  Items:
  • legs count=2
  • diet=omnivore
  • blood=hot
  • body hair=yes
  • eggs=false
  • breastfeeding=indeed
- **fox**
  Items:
  • legs count=4
  • diet=omnivore
  • blood=hot
  • body hair=yes
  • eggs=false
  • breastfeeding=indeed
- **platypus**
  Items:
  • legs count=4
  • diet=carnivore
  • blood=hot
  • body hair=yes
  • eggs=true
  • breastfeeding=indeed

To have an individual item be frequent (i.e. appear in at least 2 transactions), we check the counts:

- **blood=hot** appears in all 4 rows.
- **body hair=yes** appears in all 4 rows.

- **breastfeeding=indeed** appears in all 4 rows.
- **legs count=4** appears in 3 rows (rabbit, fox, platypus) while **legs count=2** appears only once.
- **diet=omnivore** appears in 2 rows (human, fox) while **diet=herbivore** and **diet=carnivore** appear only once.
- **eggs=false** appears in 3 rows (rabbit, human, fox) while **eggs=true** appears only once.

Thus, the items that occur in at least 2 transactions (frequent 1-itemsets) are:

- blood=hot
- body hair=yes
- breastfeeding=indeed
- legs count=4
- diet=omnivore
- eggs=false

Since we need frequent itemsets of **length 5** (i.e. containing exactly 5 items) with a minimal support of 2, we must choose 5 items from the above list and count in which transactions they appear together.

Examining the transactions with respect to these frequent items:

- **Transaction (rabbit):** Contains: legs count=4, blood=hot, body hair=yes, eggs=false
  (It does not have diet=omnivore, because its diet is herbivore.)
- **Transaction (human):** Contains: diet=omnivore, blood=hot, body hair=yes, eggs=false
  (It does not have legs count=4, because its legs count is 2.)
- **Transaction (fox):** Contains: legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false
  (It contains all 5 of these items.)
- **Transaction (platypus):** Contains: legs count=4, blood=hot, body hair=yes, breastfeeding=indeed
  (It does not have eggs=false nor diet=omnivore.)

Also, note that **breastfeeding=indeed** is present in every row, so we can include it without affecting support. (In fact, the other three items – blood=hot, body hair=yes, and breastfeeding=indeed – appear in every transaction.)

Now, consider which 5-item combinations (subsets of our 6 eligible items) can appear in at least 2 transactions. Two promising combinations are:

1. **Combination 1:**
   {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, eggs=false}
   - Appears in: rabbit and fox (support = 2)
2. **Combination 2:**
   {blood=hot, body hair=yes, breastfeeding=indeed, diet=omnivore, eggs=false}
   - Appears in: human and fox (support = 2)

Other 5-item combinations either require an item that is missing in a second transaction or would appear only in one row.

Thus, the frequent itemsets of length 5 with minimal support equal to 2 are the two combinations listed above.

Following the exact formatting requested (as a python set of sets represented as a string), the answer is:

ini

ZkopírovatUpravit

set_length5 = {{"blood=hot", "body hair=yes", "breastfeeding=indeed", "legs count=4", "eggs=false"}, {"blood=hot", "body hair=yes", "breastfeeding=indeed", "diet=omnivore", "eggs=false"}}

The only two 5-item combinations that appear in at least 2 rows (ignoring the “name” column and using only the cell values) are the following:

- The itemset that comes from the “legs count” value 4 combined with “blood” = hot, “body hair” = yes, “eggs” = false, and “breastfeeding” = indeed.
- The itemset that comes from the “diet” value omnivore combined with “blood” = hot, “body hair” = yes, “eggs” = false, and “breastfeeding” = indeed.

Thus, replacing <<itemsets>> with the corresponding Python set of sets (using only the values), we have:

set_length5 = {{"4", "hot", "yes", "false", "indeed"}, {"omnivore", "hot", "yes", "false", "indeed"}}

This answer was derived by identifying the frequent single items (with support ≥ 2) among the columns (excluding “name”) and then checking which 5-item combinations appear together in at least 2 transactions.

We start by transforming the CSV into transactions (ignoring the “name” column). Each transaction is then the set of attribute-value pairs from these six attributes:

- **Transaction 1 (rabbit):**
  {legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Transaction 2 (human):**
  {legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Transaction 3 (fox):**
  {legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
- **Transaction 4 (platypus):**
  {legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

Step 1. Identify frequent single items (support ≥2):

Looking at each attribute:

- **blood=hot, body hair=yes, breastfeeding=indeed** appear in all 4 transactions.
- **legs count=4** appears in transactions 1, 3, and 4.
- **eggs=false** appears in transactions 1, 2, and 3.
- **diet=omnivore** appears in transactions 2 and 3.

The remaining attribute–values (legs count=2, diet=herbivore, diet=carnivore, eggs=true) occur only once, so they are not frequent. Thus the only frequent items are: {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, eggs=false, diet=omnivore}.

Step 2. Construct candidate itemsets of length 5:

Since there are 6 frequent items, we form all 6 subsets of length 5 (each is obtained by dropping one item):

1. **Drop blood=hot:**
   {body hair=yes, breastfeeding=indeed, legs count=4, eggs=false, diet=omnivore}
   Support check: Only transaction 3 has legs count=4, eggs=false, diet=omnivore (transactions 1 and 4 fail on diet, and transaction 2 fails on legs count). Support = 1. Not frequent.

2. **Drop body hair=yes:**
   {blood=hot, breastfeeding=indeed, legs count=4, eggs=false, diet=omnivore}
   Support: Only transaction 3 qualifies. Support = 1. Not frequent.

3. **Drop breastfeeding=indeed:**
   {blood=hot, body hair=yes, legs count=4, eggs=false, diet=omnivore}
   Support: Again only transaction 3 qualifies. Support = 1. Not frequent.

4. **Drop legs count=4:**
   {blood=hot, body hair=yes, breastfeeding=indeed, eggs=false, diet=omnivore}
   Support:

   - Transaction 2: has blood=hot, body hair=yes, breastfeeding=indeed, eggs=false, diet=omnivore
   - Transaction 3: has blood=hot, body hair=yes, breastfeeding=indeed, eggs=false, diet=omnivore
     Support = 2. **Frequent.**

5. **Drop eggs=false:**
   {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, diet=omnivore}
   Support:

   - Transaction 3: qualifies (it has legs count=4 and diet=omnivore), but transaction 1 fails on diet and transaction 4 fails on diet.
     Support = 1. Not frequent.

6. **Drop diet=omnivore:**
   {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, eggs=false}
   Support:

   - Transaction 1: qualifies
   - Transaction 3: qualifies
     (Transactions 2 and 4 do not have legs count=4 or eggs=false, respectively.)
     Support = 2. **Frequent.**

Step 3. Final answer

The two frequent itemsets of length 5 with minimal support 2 are:

- {blood=hot, body hair=yes, breastfeeding=indeed, eggs=false, diet=omnivore}
- {blood=hot, body hair=yes, breastfeeding=indeed, legs count=4, eggs=false}

Following the formatting example (a python set of sets formatted as a string), we have:

ini

ZkopírovatUpravit

set={{"blood=hot","body hair=yes","breastfeeding=indeed","eggs=false","diet=omnivore"}, {"blood=hot","body hair=yes","breastfeeding=indeed","legs count=4","eggs=false"}}

We first view each row (after the header) as a transaction whose items are "column=value". For example, the row for the rabbit is represented as:

- legs count=4
- diet=herbivore
- blood=hot
- body hair=yes
- eggs=false
- breastfeeding=indeed

Likewise, the other three rows give us:

Transaction 1 (rabbit):
{legs count=4, diet=herbivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}

Transaction 2 (human):
{legs count=2, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}

Transaction 3 (fox):
{legs count=4, diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}

Transaction 4 (platypus):
{legs count=4, diet=carnivore, blood=hot, body hair=yes, eggs=true, breastfeeding=indeed}

Since we need itemsets to be "frequent" (i.e. appear in at least 2 transactions) and have a length of 5, we first note that an item must occur at least twice to even be considered. Ignoring the "name" attribute (which uniquely identifies each row), we count the frequency of each attribute-value pair:

- **legs count=4:** in transactions 1, 3, and 4 (support = 3)
- **diet=omnivore:** in transactions 2 and 3 (support = 2)
- **blood=hot:** in every transaction (support = 4)

- **body hair=yes:** in every transaction (support = 4)
- **eggs=false:** in transactions 1, 2, and 3 (support = 3)
- (Other values appear only once and are not frequent.)

Thus, the only items that occur with support ≥ 2 are:

- legs count=4
- diet=omnivore
- blood=hot
- body hair=yes
- eggs=false
- breastfeeding=indeed (Note: “breastfeeding=indeed” appears in all transactions.)

Now, since we want itemsets of length 5, we must pick 5 items from these 6 and then check in which transactions they occur together. When we examine all 6 choose 5 (=6) combinations, only two combinations appear together in at least 2 transactions:

1. **Combination 1:** {legs count=4, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
    - Found in Transaction 1 (rabbit) and Transaction 3 (fox).
2. **Combination 2:** {diet=omnivore, blood=hot, body hair=yes, eggs=false, breastfeeding=indeed}
    - Found in Transaction 2 (human) and Transaction 3 (fox).

No other 5-item combination reaches a support of 2.

Thus, formatted exactly as required (using python set notation with each itemset inside curly braces and no extra characters), the answer is:

ini

ZkopírovatUpravit

```
set_length5 = {{"legs count=4", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}, {"diet=omnivore", "blood=hot", "body hair=yes", "eggs=false", "breastfeeding=indeed"}}
```