

PROMPT

Find all frequent itemsets with minimal support equal to 2 and length 1, so `set_length_1 = {<<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, 6, carnivore, cold, no, true, negative

human, 4, herbivore, cold, no, true, negative

fox, 6, herbivore, cold, no, true, negative

platypus, 6, omnivore, cold, no, false, negative

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

Note: The prompt is same for all lengths except the length demanded. Instead of „length 1“ we can for example have „length 2“

ANSWERS

LENGTH 1

From the CSV, we first ignore the header row and consider each row as a transaction. Then, we count the occurrences of each individual value (i.e., item) across all transactions. The items that appear in at least 2 transactions (minimal support = 2) and that are of length 1 are:

- **From "legs count":** "6" appears 3 times.
- **From "diet":** "herbivore" appears 2 times.
- **From "blood":** "cold" appears 4 times.
- **From "body hair":** "no" appears 4 times.
- **From "eggs":** "true" appears 3 times.
- **From "breastfeeding":** "negative" appears 4 times.

The remaining items (like the names "rabbit", "human", etc., "4", "carnivore", "omnivore", and "false") appear only once and are therefore not frequent.

Thus, formatted as a Python set (with each item as a one-element set), the result is:

ini

ZkopírovatUpravit

```
set_length_1 = {"6", "herbivore", "cold", "no", "true", "negative"}
```

The frequent 1-itemsets (with support at least 2) from the CSV (ignoring the “name” column, as each value there is unique) are:

- **legs count:** "6" (appears 3 times)
- **diet:** "herbivore" (appears 2 times)
- **blood:** "cold" (appears 4 times)
- **body hair:** "no" (appears 4 times)
- **eggs:** "true" (appears 3 times)
- **breastfeeding:** "negative" (appears 4 times)

So, the result formatted as required is:

```
set_length_1 = {"6","herbivore","cold","no","true","negative"}
```

We first list the transactions (ignoring the “name” column since each name is unique):

- **legs count:**
 - “6” appears in rabbit, fox, and platypus (3 times)
 - “4” appears only in human (1 time)
 - Only “6” meets the minimal support of 2.
- **diet:**
 - “carnivore” appears once
 - “herbivore” appears twice (human and fox)
 - “omnivore” appears once
 - Only “herbivore” qualifies.
- **blood:**
 - “cold” appears in every transaction (4 times)
 - “cold” qualifies.
- **body hair:**
 - “no” appears in every transaction (4 times)
 - “no” qualifies.
- **eggs:**
 - “true” appears 3 times (rabbit, human, fox)

- “false” appears once (platypus)
→ “true” qualifies.
- **breastfeeding:**
 - “negative” appears in every transaction (4 times)
→ “negative” qualifies.

Thus, the frequent itemsets of length 1 (using only the item values, not the column names) are:

```
set_length_1={{"6"}, {"herbivore"}, {"cold"}, {"no"}, {"true"}, {"negative"}}
```

To solve the problem by inspection, we first view each row (excluding the header) as a transaction with several items. We then count how many times each item appears across all transactions and retain only those items that appear in at least 2 transactions (minimal support = 2).

Here’s the CSV data with transactions:

- **Row 1:** rabbit, 6, carnivore, cold, no, true, negative
- **Row 2:** human, 4, herbivore, cold, no, true, negative
- **Row 3:** fox, 6, herbivore, cold, no, true, negative
- **Row 4:** platypus, 6, omnivore, cold, no, false, negative

Since the “name” column values are unique to each transaction, none of them are frequent. Next, we review the remaining columns:

- **legs count:**
 - "6" appears in Rows 1, 3, and 4 (support = 3)
 - "4" appears only in Row 2 (support = 1)
- **diet:**
 - "carnivore" appears in Row 1 (support = 1)
 - "herbivore" appears in Rows 2 and 3 (support = 2)
 - "omnivore" appears in Row 4 (support = 1)
- **blood:**
 - "cold" appears in all rows (support = 4)
- **body hair:**
 - "no" appears in all rows (support = 4)

- **eggs:**
 - "true" appears in Rows 1, 2, and 3 (support = 3)
 - "false" appears in Row 4 (support = 1)
- **breastfeeding:**
 - "negative" appears in all rows (support = 4)

Thus, the items with minimal support (≥ 2) are:

- "6"
- "herbivore"
- "cold"
- "no"
- "true"
- "negative"

Following the required output format (a Python set formatted as string), we have:

```
set_length_1={{ "6"}, {"herbivore"}, {"cold"}, {"no"}, {"true"}, {"negative"}}
```

We start by reading the CSV data (using the first row as the header) and then manually counting how many times each value appears (ignoring the "name" column because every value there is unique). For each remaining column, we check for values that occur in at least 2 rows:

- **legs count:**
 - "6" appears in 3 rows (rabbit, fox, platypus).
 - "4" appears only once (human).
- **diet:**
 - "carnivore" appears once (rabbit).
 - "herbivore" appears twice (human, fox).
 - "omnivore" appears once (platypus).
- **blood:**
 - "cold" appears in all 4 rows.
- **body hair:**
 - "no" appears in all 4 rows.

- **eggs:**
 - “true” appears in 3 rows (rabbit, human, fox).
 - “false” appears once (platypus).
- **breastfeeding:**
 - “negative” appears in all 4 rows.

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

- "6"
- "herbivore"
- "cold"
- "no"
- "true"
- "negative"

According to the required format (each item in its own singleton set, and then combined into a Python set as a string), the answer is:

ini

ZkopírovaťUpraviť

```
set_length_1={{"6"}, {"herbivore"}, {"cold"}, {"no"}, {"true"}, {"negative"}}
```

LENGTH 2

```
set_length_2={{"legs count=6", "blood=cold"}, {"legs count=6", "body hair=no"}, {"legs count=6", "eggs=true"}, {"legs count=6", "breastfeeding=negative"}, {"diet=herbivore", "blood=cold"}, {"diet=herbivore", "body hair=no"}, {"diet=herbivore", "eggs=true"}, {"diet=herbivore", "breastfeeding=negative"}, {"blood=cold", "body hair=no"}, {"blood=cold", "eggs=true"}, {"blood=cold", "breastfeeding=negative"}, {"body hair=no", "eggs=true"}, {"body hair=no", "breastfeeding=negative"}, {"eggs=true", "breastfeeding=negative"}}
```

We begin by listing each transaction (each row, ignoring the header row) with the attribute values from the CSV:

- **T1 (rabbit):**
{name=rabbit, legs count=6, diet=carnivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T2 (human):**
{name=human, legs count=4, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T3 (fox):**
{name=fox, legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T4 (platypus):**
{name=platypus, legs count=6, diet=omnivore, blood=cold, body hair=no, eggs=false, breastfeeding=negative}

Since the “name” attribute is unique in every transaction, it does not form any frequent pair (support must be at least 2). For the remaining attributes, we first identify the items that appear in at least 2 transactions (minimal support 2):

- **legs count:**
 - *legs count=6* (appears in T1, T3, T4; support = 3)
 - *legs count=4* (only in T2; support = 1) → **discarded**
- **diet:**
 - *diet=herbivore* (appears in T2, T3; support = 2)
 - *diet=carnivore* (T1 only; support = 1)
 - *diet=omnivore* (T4 only; support = 1)
→ Only *diet=herbivore* qualifies.
- **blood:**
 - *blood=cold* (appears in all; support = 4)
- **body hair:**
 - *body hair=no* (appears in all; support = 4)
- **eggs:**
 - *eggs=true* (appears in T1, T2, T3; support = 3)
 - *eggs=false* (T4 only; support = 1)
- **breastfeeding:**

- *breastfeeding=negative* (appears in all; support = 4)

Next, we form all pairs (itemsets of length 2) using only the frequent items listed above. (When two items appear together in a transaction, their pair has support equal to the number of transactions where both occur.) The pairs that occur in at least 2 transactions are:

1. {"legs count=6", "blood=cold"}
2. {"legs count=6", "body hair=no"}
3. {"legs count=6", "eggs=true"}
4. {"legs count=6", "breastfeeding=negative"}
5. {"blood=cold", "body hair=no"}
6. {"blood=cold", "eggs=true"}
7. {"blood=cold", "breastfeeding=negative"}
8. {"body hair=no", "eggs=true"}
9. {"body hair=no", "breastfeeding=negative"}
10. {"eggs=true", "breastfeeding=negative"}
11. {"diet=herbivore", "blood=cold"}
12. {"diet=herbivore", "body hair=no"}
13. {"diet=herbivore", "eggs=true"}
14. {"diet=herbivore", "breastfeeding=negative"}

Note that although *legs count=6* is frequent and *diet=herbivore* is frequent, their combination appears only in T3 (support = 1) and is therefore not included.

Finally, inserting these pairs into a Python set (formatted as a string) gives:

```
set_length_2 = {"legs count=6", "blood=cold"}, {"legs count=6", "body hair=no"}, {"legs count=6", "eggs=true"}, {"legs count=6", "breastfeeding=negative"}, {"blood=cold", "body hair=no"}, {"blood=cold", "eggs=true"}, {"blood=cold", "breastfeeding=negative"}, {"body hair=no", "eggs=true"}, {"body hair=no", "breastfeeding=negative"}, {"eggs=true", "breastfeeding=negative"}, {"diet=herbivore", "blood=cold"}, {"diet=herbivore", "body hair=no"}, {"diet=herbivore", "eggs=true"}, {"diet=herbivore", "breastfeeding=negative"}
```

This is the set of all frequent itemsets of length 2 with a minimal support of 2, determined manually without using any programming tools.

Below is the answer derived by manually counting the occurrences of the attribute values (ignoring the header row “name” so that only the six feature-columns are used). In each transaction the “items” are the cell values (without their column names). This gives:

- Transaction 1 (rabbit): {"6", "carnivore", "cold", "no", "true", "negative"}
- Transaction 2 (human): {"4", "herbivore", "cold", "no", "true", "negative"}
- Transaction 3 (fox): {"6", "herbivore", "cold", "no", "true", "negative"}
- Transaction 4 (platypus): {"6", "omnivore", "cold", "no", "false", "negative"}

Since the minimum support is 2, we first discard any item that occurs in fewer than 2 transactions. The frequent items (values) are:

- "6" (in T1, T3, T4)
- "herbivore" (in T2, T3)
- "cold" (in all transactions)
- "no" (in all transactions)
- "true" (in T1, T2, T3)
- "negative" (in all transactions)

Next, every pair (of length 2) of these items is examined (by “intersecting” the transactions that contain each value). For example, the pair {"6", "cold"} appears in transactions T1, T3, and T4 (support = 3) and is therefore frequent. One pair, {"6", "herbivore"}, appears only in T3 (support = 1) and is not frequent. Carrying out this analysis for all candidate pairs, the frequent itemsets of length 2 turn out to be:

- {"6", "cold"}
- {"6", "no"}
- {"6", "true"}
- {"6", "negative"}
- {"cold", "no"}
- {"cold", "true"}
- {"cold", "negative"}
- {"no", "true"}
- {"no", "negative"}
- {"true", "negative"}

- {"herbivore", "cold"}
- {"herbivore", "no"}
- {"herbivore", "true"}
- {"herbivore", "negative"}

Thus, formatted as a Python set of sets (with each itemset represented as a set of strings) the answer is:

```
set_length_2 = {"6", "cold"}, {"6", "no"}, {"6", "true"}, {"6", "negative"}, {"cold", "no"}, {"cold", "true"}, {"cold", "negative"}, {"no", "true"}, {"no", "negative"}, {"true", "negative"}, {"herbivore", "cold"}, {"herbivore", "no"}, {"herbivore", "true"}, {"herbivore", "negative"}
```

This is the complete answer according to the given instructions without using any programming language for the calculation.

We start by noting that the CSV has 7 columns (the first “name” is an identifier, so we work with the other 6 attributes). The rows are:

- **rabbit:** legs count=6, diet=carnivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative
- **human:** legs count=4, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative
- **fox:** legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative
- **platypus:** legs count=6, diet=omnivore, blood=cold, body hair=no, eggs=false, breastfeeding=negative

For an item (an attribute-value pair) to be eligible, its support (i.e. its count across rows) must be at least 2. Considering this, the items that pass are:

- **legs count=6** (appears in rabbit, fox, platypus; support 3)
- **diet=herbivore** (appears in human, fox; support 2)
- **blood=cold** (appears in every row; support 4)
- **body hair=no** (appears in every row; support 4)
- **eggs=true** (appears in rabbit, human, fox; support 3)
- **breastfeeding=negative** (appears in every row; support 4)

Now, for each pair of these items we determine the joint support (i.e. the number of rows where both items appear):

1. **{legs count=6, blood=cold}**: rows – rabbit, fox, platypus (support 3)
2. **{legs count=6, body hair=no}**: rows – rabbit, fox, platypus (support 3)
3. **{legs count=6, eggs=true}**: rows – rabbit, fox (support 2)
4. **{legs count=6, breastfeeding=negative}**: rows – rabbit, fox, platypus (support 3)
5. **{diet=herbivore, blood=cold}**: rows – human, fox (support 2)
6. **{diet=herbivore, body hair=no}**: rows – human, fox (support 2)
7. **{diet=herbivore, eggs=true}**: rows – human, fox (support 2)
8. **{diet=herbivore, breastfeeding=negative}**: rows – human, fox (support 2)
9. **{blood=cold, body hair=no}**: rows – all (support 4)
10. **{blood=cold, eggs=true}**: rows – rabbit, human, fox (support 3)
11. **{blood=cold, breastfeeding=negative}**: rows – all (support 4)
12. **{body hair=no, eggs=true}**: rows – rabbit, human, fox (support 3)
13. **{body hair=no, breastfeeding=negative}**: rows – all (support 4)
14. **{eggs=true, breastfeeding=negative}**: rows – rabbit, human, fox (support 3)

All these 14 pairs have a support of at least 2, so they are our frequent itemsets of length 2.

Following the formatting requirement (as a Python set of itemsets represented as strings) the answer is:

ini

ZkopírovatUpravit

```
set_length_2 = {"legs count=6","blood=cold"}, {"legs count=6","body hair=no"}, {"legs count=6","eggs=true"}, {"legs count=6","breastfeeding=negative"}, {"diet=herbivore","blood=cold"}, {"diet=herbivore","body hair=no"}, {"diet=herbivore","eggs=true"}, {"diet=herbivore","breastfeeding=negative"}, {"blood=cold","body hair=no"}, {"blood=cold","eggs=true"}, {"blood=cold","breastfeeding=negative"}, {"body hair=no","eggs=true"}, {"body hair=no","breastfeeding=negative"}, {"eggs=true","breastfeeding=negative"}
```

This string shows the collection of all frequent itemsets of length 2 with minimal support 2

Below is the answer obtained by “manually” working through the CSV without writing any code. In this case the “items” are taken as the cell values from each row (except the “name” column, which serves only as an identifier). (Values that occur only once are not considered because their support is below 2.)

If we ignore the column names, then the “basket” for each row is as follows:

- **Row 1 (rabbit):** 6, carnivore, cold, no, true, negative
- **Row 2 (human):** 4, herbivore, cold, no, true, negative
- **Row 3 (fox):** 6, herbivore, cold, no, true, negative
- **Row 4 (platypus):** 6, omnivore, cold, no, false, negative

Notice that (by support count ≥ 2) only the following values qualify:

- From “legs count”: **6** (because “4” appears only once)
- From “diet”: **herbivore** (carnivore and omnivore appear only once)
- From “blood”: **cold** (appears in every row)
- From “body hair”: **no** (appears in every row)
- From “eggs”: **true** (since “false” appears only once)
- From “breastfeeding”: **negative** (appears in every row)

Thus the candidate frequent items are:

6, herbivore, cold, no, true, negative

We now consider all pairs (itemsets of length 2) that occur together in at least 2 rows. By checking each pair manually, we find the following 14 frequent itemsets:

- {"6", "cold"}
- {"6", "no"}
- {"6", "true"}
- {"6", "negative"}
- {"herbivore", "cold"}
- {"herbivore", "no"}
- {"herbivore", "true"}
- {"herbivore", "negative"}
- {"cold", "no"}

- {"cold", "true"}
- {"cold", "negative"}
- {"no", "true"}
- {"no", "negative"}
- {"true", "negative"}

When formatted as a Python set (using a notation similar to the example provided), the answer is:

```
set_length_2 = {"6", "cold"}, {"6", "no"}, {"6", "true"}, {"6", "negative"}, {"herbivore", "cold"}, {"herbivore", "no"}, {"herbivore", "true"}, {"herbivore", "negative"}, {"cold", "no"}, {"cold", "true"}, {"cold", "negative"}, {"no", "true"}, {"no", "negative"}, {"true", "negative"}
```

LENGTH 3

Below is the solution derived entirely by reasoning:

1. Step 1. Identify Items and Their Supports

Treating each (column, value) pair as an “item” (except “name” since each animal is unique), we have the following attributes (and only keeping those that appear in at least 2 rows):

- **legs count:**
 - “6” appears in *rabbit*, *fox*, and *platypus* (support = 3).
 - “4” appears only in *human* (discarded).
- **diet:**
 - “herbivore” appears in *human* and *fox* (support = 2).
 - “carnivore” (only *rabbit*) and “omnivore” (only *platypus*) are discarded.
- **blood:** “cold” appears in every row (support = 4).
- **body hair:** “no” appears in every row (support = 4).
- **eggs:**
 - “true” appears in *rabbit*, *human*, and *fox* (support = 3).
 - “false” appears only in *platypus* (discarded).
- **breastfeeding:** “negative” appears in every row (support = 4).

2. Step 2. List the Frequent Items

The items that meet the minimum support (2) are:

- legs count=6

- diet=herbivore
- blood=cold
- body hair=no
- eggs=true
- breastfeeding=negative

3. Step 3. Find All 3-Item Combinations with Support ≥ 2

There are 20 possible combinations (choose 3 out of 6), but some occur in only one transaction when checking the intersection across rows. For each transaction, the available items (only considering frequent ones) are:

- **rabbit:** {legs count=6, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **human:** {diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **fox:** {legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **platypus:** {legs count=6, blood=cold, body hair=no, breastfeeding=negative}

By comparing the sets we see that any combination involving “diet=herbivore” together with “legs count=6” appears only in *fox* (since *rabbit* and *platypus* lack “diet=herbivore”) so those combinations are not frequent.

The **frequent 3-item sets** (each found in at least 2 transactions) are:

- {legs count=6, blood=cold, body hair=no} (found in *rabbit*, *fox*, *platypus*)
- {legs count=6, blood=cold, eggs=true} (found in *rabbit*, *fox*)
- {legs count=6, blood=cold, breastfeeding=negative} (found in *rabbit*, *fox*, *platypus*)
- {legs count=6, body hair=no, eggs=true} (found in *rabbit*, *fox*)
- {legs count=6, body hair=no, breastfeeding=negative} (found in *rabbit*, *fox*, *platypus*)
- {legs count=6, eggs=true, breastfeeding=negative} (found in *rabbit*, *fox*)

- {blood=cold, body hair=no, eggs=true} (found in *rabbit*, *human*, *fox*)
- {blood=cold, body hair=no, breastfeeding=negative} (found in all rows)
- {blood=cold, eggs=true, breastfeeding=negative} (found in *rabbit*, *human*, *fox*)
- {body hair=no, eggs=true, breastfeeding=negative} (found in *rabbit*, *human*, *fox*)
- {diet=herbivore, blood=cold, body hair=no} (found in *human*, *fox*)
- {diet=herbivore, blood=cold, eggs=true} (found in *human*, *fox*)
- {diet=herbivore, blood=cold, breastfeeding=negative} (found in *human*, *fox*)
- {diet=herbivore, body hair=no, eggs=true} (found in *human*, *fox*)
- {diet=herbivore, body hair=no, breastfeeding=negative} (found in *human*, *fox*)
- {diet=herbivore, eggs=true, breastfeeding=negative} (found in *human*, *fox*)

4. Step 4. Format the Answer

As requested, the answer is given as a Python set formatted as a string (each frequent 3-itemset is represented as a set of strings):

makefile

```
set_length_3 = {"legs count=6", "blood=cold", "body hair=no"},
               {"legs count=6", "blood=cold", "eggs=true"},
               {"legs count=6", "blood=cold", "breastfeeding=negative"},
               {"legs count=6", "body hair=no", "eggs=true"},
               {"legs count=6", "body hair=no", "breastfeeding=negative"},
               {"legs count=6", "eggs=true", "breastfeeding=negative"},
               {"blood=cold", "body hair=no", "eggs=true"},
               {"blood=cold", "body hair=no", "breastfeeding=negative"},
               {"blood=cold", "eggs=true", "breastfeeding=negative"},
               {"body hair=no", "eggs=true", "breastfeeding=negative"},
               {"diet=herbivore", "blood=cold", "body hair=no"},
```

```
{"diet=herbivore", "blood=cold", "eggs=true"},  
{"diet=herbivore", "blood=cold", "breastfeeding=negative"},  
{"diet=herbivore", "body hair=no", "eggs=true"},  
{"diet=herbivore", "body hair=no", "breastfeeding=negative"},  
{"diet=herbivore", "eggs=true", "breastfeeding=negative"}}
```

This string is your final answer, with all frequent itemsets of length 3 that have support at least 2.

Here's the solution using manual reasoning without any programming:

1. Data Processing:

The CSV (ignoring the "name" column) gives us these transactions (one per row) with six attributes:

- **T1 (rabbit):** {"6", "carnivore", "cold", "no", "true", "negative"}
- **T2 (human):** {"4", "herbivore", "cold", "no", "true", "negative"}
- **T3 (fox):** {"6", "herbivore", "cold", "no", "true", "negative"}
- **T4 (platypus):** {"6", "omnivore", "cold", "no", "false", "negative"}

Since we consider the column names as headers, we only mine the attribute values.

We then focus on items that occur at least twice (minimal support = 2). Examining each attribute:

- **Legs count:** "6" appears in T1, T3, and T4 (support 3); "4" appears only in T2 (support 1).
- **Diet:** "herbivore" is in T2 and T3 (support 2) while "carnivore" and "omnivore" appear only once.
- **Blood:** "cold" is present in every transaction (support 4).
- **Body hair:** "no" is present in every transaction (support 4).
- **Eggs:** "true" appears in T1, T2, and T3 (support 3); "false" appears only in T4.
- **Breastfeeding:** "negative" appears in every transaction (support 4).

So the frequent items are:

- "cold", "no", "negative", "6", "herbivore", and "true".

2. Transactions with Frequent Items Only:

Replace each transaction with only the frequent items:

- **T1:** {"cold", "no", "negative", "6", "true"}
- **T2:** {"cold", "no", "negative", "herbivore", "true"}
- **T3:** {"cold", "no", "negative", "6", "herbivore", "true"}
- **T4:** {"cold", "no", "negative", "6"}

3. Frequent Itemsets of Length 3:

We list all candidate 3-item combinations from the frequent items and count their occurrences (support). For clarity, let's denote:

- A = "cold"
- B = "no"
- C = "negative"
- D = "6"
- E = "herbivore"
- F = "true"

By checking each candidate (by intersecting transactions), the following 3-item sets appear in at least 2 transactions:

- {A, B, C} → {"cold", "no", "negative"} (in all transactions)
- {A, B, D} → {"cold", "no", "6"} (in T1, T3, T4)
- {A, B, E} → {"cold", "no", "herbivore"} (in T2, T3)
- {A, B, F} → {"cold", "no", "true"} (in T1, T2, T3)
- {A, C, D} → {"cold", "negative", "6"} (in T1, T3, T4)
- {A, C, E} → {"cold", "negative", "herbivore"} (in T2, T3)
- {A, C, F} → {"cold", "negative", "true"} (in T1, T2, T3)
- {A, D, F} → {"cold", "6", "true"} (in T1, T3)
- {A, E, F} → {"cold", "herbivore", "true"} (in T2, T3)
- {B, C, D} → {"no", "negative", "6"} (in T1, T3, T4)
- {B, C, E} → {"no", "negative", "herbivore"} (in T2, T3)
- {B, C, F} → {"no", "negative", "true"} (in T1, T2, T3)

- {B, D, F} → {"no", "6", "true"} (in T1, T3)
- {B, E, F} → {"no", "herbivore", "true"} (in T2, T3)
- {C, D, F} → {"negative", "6", "true"} (in T1, T3)
- {C, E, F} → {"negative", "herbivore", "true"} (in T2, T3)

4. Final Output Format:

Following the required Python set format, we insert the frequent itemsets (as string representations) into the placeholder. Thus, the final answer is:

```
set_length_3 = {"cold", "no", "negative"}, {"cold", "no", "6"}, {"cold", "no", "herbivore"}, {"cold", "no", "true"}, {"cold", "negative", "6"}, {"cold", "negative", "herbivore"}, {"cold", "negative", "true"}, {"cold", "6", "true"}, {"cold", "herbivore", "true"}, {"no", "negative", "6"}, {"no", "negative", "herbivore"}, {"no", "negative", "true"}, {"no", "6", "true"}, {"no", "herbivore", "true"}, {"negative", "6", "true"}, {"negative", "herbivore", "true"}
```

Below is the answer with the frequent itemsets (of length 3 and support at least 2) written as a Python-style set of sets. (Remember that the “name” column is not used as an item, and an item is taken to be a “column=value” pair.) In our table only these items are frequent (appearing in at least 2 transactions):

- blood=cold (support 4)
- body hair=no (support 4)
- breastfeeding=negative (support 4)
- legs count=6 (support 3)
- diet=herbivore (support 2)
- eggs=true (support 3)

Then, by considering all combinations of 3 among these frequent items—and discarding any combination that contains an infrequent pair (for example, “legs count=6” and “diet=herbivore” appear together only in the fox row, so they are dropped)—we obtain the following 16 frequent 3-itemsets (each with support ≥2):

- {blood=cold, body hair=no, breastfeeding=negative} (support 4)
- {legs count=6, blood=cold, body hair=no} (support 3)
- {diet=herbivore, blood=cold, body hair=no} (support 2)
- {blood=cold, body hair=no, eggs=true} (support 3)
- {legs count=6, blood=cold, breastfeeding=negative} (support 3)

- {diet=herbivore, blood=cold, breastfeeding=negative} (support 2)
- {blood=cold, eggs=true, breastfeeding=negative} (support 3)
- {legs count=6, blood=cold, eggs=true} (support 2)
- {diet=herbivore, blood=cold, eggs=true} (support 2)
- {legs count=6, body hair=no, breastfeeding=negative} (support 3)
- {diet=herbivore, body hair=no, breastfeeding=negative} (support 2)
- {body hair=no, eggs=true, breastfeeding=negative} (support 3)
- {legs count=6, body hair=no, eggs=true} (support 2)
- {diet=herbivore, body hair=no, eggs=true} (support 2)
- {legs count=6, eggs=true, breastfeeding=negative} (support 2)
- {diet=herbivore, eggs=true, breastfeeding=negative} (support 2)

When formatting the output as a Python set (with each itemset also written as a set of strings) the final answer is:

makefile

```
set_length_3 = {"blood=cold", "body hair=no", "breastfeeding=negative"},
               {"legs count=6", "blood=cold", "body hair=no"},
               {"diet=herbivore", "blood=cold", "body hair=no"},
               {"blood=cold", "body hair=no", "eggs=true"},
               {"legs count=6", "blood=cold", "breastfeeding=negative"},
               {"diet=herbivore", "blood=cold", "breastfeeding=negative"},
               {"blood=cold", "eggs=true", "breastfeeding=negative"},
               {"legs count=6", "blood=cold", "eggs=true"},
               {"diet=herbivore", "blood=cold", "eggs=true"},
               {"legs count=6", "body hair=no", "breastfeeding=negative"},
               {"diet=herbivore", "body hair=no", "breastfeeding=negative"},
               {"body hair=no", "eggs=true", "breastfeeding=negative"},
               {"legs count=6", "body hair=no", "eggs=true"},
               {"diet=herbivore", "body hair=no", "eggs=true"},
```

```
{"legs count=6", "eggs=true", "breastfeeding=negative"},  
{"diet=herbivore", "eggs=true", "breastfeeding=negative"}}
```

This is the set `set_length_3` with all frequent 3-itemsets meeting the criteria.

Below is the answer arrived at by hand-analysis. First, note that the “name” column is just an identifier so we ignore it. We consider the remaining six attributes by pairing each column name with its value. (For example, from the “legs count” column we get items “legs count=6” or “legs count=4”.) When we list the transactions, we keep only those items that occur at least twice overall. In our data the following items are frequent (i.e. with support ≥ 2):

- **legs count=6** (appears in rows 1, 3, 4)
- **diet=herbivore** (rows 2, 3)
- **blood=cold** (all rows)
- **body hair=no** (all rows)
- **eggs=true** (rows 1, 2, 3)
- **breastfeeding=negative** (all rows)

Then the transactions (using only these frequent items) are:

- Row 1: {legs count=6, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Row 2: {diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Row 3: {legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Row 4: {legs count=6, blood=cold, body hair=no, breastfeeding=negative}

Let's denote:

- A = “legs count=6”
- B = “diet=herbivore”
- C = “blood=cold”
- D = “body hair=no”
- E = “eggs=true”
- F = “breastfeeding=negative”

By inspecting the transactions we determine that some 3-itemsets (i.e. sets of 3 items) appear in at least 2 transactions. For example, {C, D, F} appears in every transaction; similarly, {A, C, D} appears in rows 1, 3, and 4; and so on.

After examining all combinations of the 6 items, we find that the following 16 three-itemsets have support ≥ 2 :

- {A, C, D}
- {A, C, E}
- {A, C, F}
- {A, D, E}
- {A, D, F}
- {A, E, F}
- {B, C, D}
- {B, C, E}
- {B, C, F}
- {B, D, E}
- {B, D, F}
- {B, E, F}
- {C, D, E}
- {C, D, F}
- {C, E, F}
- {D, E, F}

Now, replacing back our shorthand with the full items, we have:

- { "legs count=6", "blood=cold", "body hair=no" }
- { "legs count=6", "blood=cold", "eggs=true" }
- { "legs count=6", "blood=cold", "breastfeeding=negative" }
- { "legs count=6", "body hair=no", "eggs=true" }
- { "legs count=6", "body hair=no", "breastfeeding=negative" }
- { "legs count=6", "eggs=true", "breastfeeding=negative" }
- { "diet=herbivore", "blood=cold", "body hair=no" }

- { "diet=herbivore", "blood=cold", "eggs=true" }
- { "diet=herbivore", "blood=cold", "breastfeeding=negative" }
- { "diet=herbivore", "body hair=no", "eggs=true" }
- { "diet=herbivore", "body hair=no", "breastfeeding=negative" }
- { "diet=herbivore", "eggs=true", "breastfeeding=negative" }
- { "blood=cold", "body hair=no", "eggs=true" }
- { "blood=cold", "body hair=no", "breastfeeding=negative" }
- { "blood=cold", "eggs=true", "breastfeeding=negative" }
- { "body hair=no", "eggs=true", "breastfeeding=negative" }

Following the formatting requirement (as a Python set of sets with each inner set in the form {"item1", "item2", "item3"}), the answer is:

```
set_length_3 = {"legs count=6", "blood=cold", "body hair=no"}, {"legs count=6",
"blood=cold", "eggs=true"}, {"legs count=6", "blood=cold", "breastfeeding=negative"},
{"legs count=6", "body hair=no", "eggs=true"}, {"legs count=6", "body hair=no",
"breastfeeding=negative"}, {"legs count=6", "eggs=true", "breastfeeding=negative"},
{"diet=herbivore", "blood=cold", "body hair=no"}, {"diet=herbivore", "blood=cold",
"eggs=true"}, {"diet=herbivore", "blood=cold", "breastfeeding=negative"},
{"diet=herbivore", "body hair=no", "eggs=true"}, {"diet=herbivore", "body hair=no",
"breastfeeding=negative"}, {"diet=herbivore", "eggs=true", "breastfeeding=negative"},
{"blood=cold", "body hair=no", "eggs=true"}, {"blood=cold", "body hair=no",
"breastfeeding=negative"}, {"blood=cold", "eggs=true", "breastfeeding=negative"}, {"body
hair=no", "eggs=true", "breastfeeding=negative"}
```

This is the final result for all frequent 3-itemsets with minimal support equal to 2.

```
set_length_3={"legs count=6", "blood=cold", "body hair=no"}, {"legs count=6",
"blood=cold", "eggs=true"}, {"legs count=6", "blood=cold", "breastfeeding=negative"},
{"legs count=6", "body hair=no", "eggs=true"}, {"legs count=6", "body hair=no",
"breastfeeding=negative"}, {"legs count=6", "eggs=true", "breastfeeding=negative"},
{"diet=herbivore", "blood=cold", "body hair=no"}, {"diet=herbivore", "blood=cold",
"eggs=true"}, {"diet=herbivore", "blood=cold", "breastfeeding=negative"},
{"diet=herbivore", "body hair=no", "eggs=true"}, {"diet=herbivore", "body hair=no",
"breastfeeding=negative"}, {"diet=herbivore", "eggs=true", "breastfeeding=negative"},
{"blood=cold", "body hair=no", "eggs=true"}, {"blood=cold", "body hair=no",
"breastfeeding=negative"}, {"blood=cold", "eggs=true", "breastfeeding=negative"}, {"body
hair=no", "eggs=true", "breastfeeding=negative"}
```

LENGTH 4

Below is the answer. First, note that we treat each row (ignoring the “name” column) as a transaction where each attribute becomes an “item” in the form

attribute=value

For example, for the “rabbit” row the items are:

legs count=6, diet=carnivore, blood=cold, body hair=no, eggs=true,
breastfeeding=negative

Only those attribute–value pairs that appear in at least two transactions are considered “frequent” as singletons. (In our data the only values that appear at least twice are:

- blood=cold (all four rows)
- body hair=no (all four rows)
- breastfeeding=negative (all four rows)
- legs count=6 (in rabbit, fox, and platypus)
- diet=herbivore (in human and fox)
- eggs=true (in rabbit, human, and fox)

)

We then build candidate itemsets of length 4 (that is, sets of 4 such items) and check in which transactions all 4 appear together. (For instance, the set {blood=cold, body hair=no, breastfeeding=negative, eggs=true} is present in the rabbit, human, and fox rows.) After checking all 15 possible combinations (from the 6 frequent items) one finds that the following nine itemsets have support at least 2:

- {blood=cold, body hair=no, breastfeeding=negative, legs count=6}
- {blood=cold, body hair=no, breastfeeding=negative, diet=herbivore}
- {blood=cold, body hair=no, breastfeeding=negative, eggs=true}
- {blood=cold, body hair=no, legs count=6, eggs=true}
- {blood=cold, body hair=no, diet=herbivore, eggs=true}
- {blood=cold, breastfeeding=negative, legs count=6, eggs=true}
- {blood=cold, breastfeeding=negative, diet=herbivore, eggs=true}
- {body hair=no, breastfeeding=negative, legs count=6, eggs=true}
- {body hair=no, breastfeeding=negative, diet=herbivore, eggs=true}

When formatting the answer as a Python set of sets (as a string) with the exact formatting requested, we get:

```
set_length_4 = {"blood=cold", "body hair=no", "breastfeeding=negative", "legs count=6"},  
{"blood=cold", "body hair=no", "breastfeeding=negative", "diet=herbivore"},
```

```
{"blood=cold", "body hair=no", "breastfeeding=negative", "eggs=true"}, {"blood=cold",  
"body hair=no", "legs count=6", "eggs=true"}, {"blood=cold", "body hair=no",  
"diet=herbivore", "eggs=true"}, {"blood=cold", "breastfeeding=negative", "legs count=6",  
"eggs=true"}, {"blood=cold", "breastfeeding=negative", "diet=herbivore", "eggs=true"},  
{"body hair=no", "breastfeeding=negative", "legs count=6", "eggs=true"}, {"body hair=no",  
"breastfeeding=negative", "diet=herbivore", "eggs=true"}}
```

This is the final answer, with each inner set representing one frequent itemset of length 4 that meets the minimal support of 2.

We begin by interpreting the CSV as a transactional database. The first row gives the attribute names, so (except for “name”) each animal (row) is represented by the set of attribute–value pairs. For example, for the first row (rabbit) the transaction is:

- legs count=6
- diet=carnivore
- blood=cold
- body hair=no
- eggs=true
- breastfeeding=negative

Doing the same for the other rows (and ignoring “name”), the transactions become:

- **T1 (rabbit):** {legs count=6, diet=carnivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T2 (human):** {legs count=4, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T3 (fox):** {legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T4 (platypus):** {legs count=6, diet=omnivore, blood=cold, body hair=no, eggs=false, breastfeeding=negative}

Now, before generating candidate 4-itemsets we look at the items that appear at least twice (i.e. support ≥ 2):

- **blood=cold:** appears in all 4
- **body hair=no:** appears in all 4
- **breastfeeding=negative:** appears in all 4

- **eggs=true:** appears in T1, T2, T3 (support 3)
- **legs count=6:** appears in T1, T3, T4 (support 3)
- **diet=herbivore:** appears in T2, T3 (support 2)

The other values (legs count=4, diet=carnivore, diet=omnivore, eggs=false) occur only once, so they are omitted from further consideration.

Let's denote the six frequent items as:

- A = blood=cold
- B = body hair=no
- C = breastfeeding=negative
- D = eggs=true
- E = legs count=6
- F = diet=herbivore

Now we consider all combinations of 4 items (i.e. 6 choose 4 = 15 candidates) and compute their supports by checking in which transactions they occur. (Remember that if an item is not in the candidate set, its presence/absence in a transaction is irrelevant; only transactions that contain all four items count.)

After checking the transactions the following 4-itemsets meet the minimal support (≥ 2):

1. **{A, B, C, D}**
 - Present in T1, T2, T3 (support = 3)
2. **{A, B, C, E}**
 - Present in T1, T3, T4 (support = 3)
3. **{A, B, C, F}**
 - Present in T2, T3 (support = 2)
4. **{A, B, D, E}**
 - Present in T1, T3 (support = 2)
5. **{A, B, D, F}**
 - Present in T2, T3 (support = 2)
6. **{A, C, D, E}**
 - Present in T1, T3 (support = 2)

7. **{A, C, D, F}**

- Present in T2, T3 (support = 2)

8. **{B, C, D, E}**

- Present in T1, T3 (support = 2)

9. **{B, C, D, F}**

- Present in T2, T3 (support = 2)

In other words, the frequent 4-itemsets (with the items expressed in the attribute=value format) are those nine sets.

Finally, following the instructions to output the answer as a python-formatted set (using the style shown in the example, i.e. each inner set is wrapped in curly braces and each item is a string, with no column names added separately) we have:

```
set_length_4 = {"blood=cold", "body hair=no", "breastfeeding=negative", "eggs=true"},  
{"blood=cold", "body hair=no", "breastfeeding=negative", "legs count=6"}, {"blood=cold",  
"body hair=no", "breastfeeding=negative", "diet=herbivore"}, {"blood=cold", "body  
hair=no", "eggs=true", "legs count=6"}, {"blood=cold", "body hair=no", "eggs=true",  
"diet=herbivore"}, {"blood=cold", "breastfeeding=negative", "eggs=true", "legs count=6"},  
{"blood=cold", "breastfeeding=negative", "eggs=true", "diet=herbivore"}, {"body hair=no",  
"breastfeeding=negative", "eggs=true", "legs count=6"}, {"body hair=no",  
"breastfeeding=negative", "eggs=true", "diet=herbivore"}}
```

This is the final answer with the placeholder replaced by the frequent itemsets of length 4.

Below is the answer obtained by manual analysis:

We first list the six non-identifier attributes from the CSV:

- legs count
- diet
- blood
- body hair
- eggs
- breastfeeding

Then we rewrite each transaction as a set of attribute=value pairs (ignoring the “name” field):

- Transaction 1 (rabbit): {legs count=6, diet=carnivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Transaction 2 (human): {legs count=4, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Transaction 3 (fox): {legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- Transaction 4 (platypus): {legs count=6, diet=omnivore, blood=cold, body hair=no, eggs=false, breastfeeding=negative}

Next, we look for itemsets of length 4 that appear in at least 2 transactions. For example, comparing Transactions 1 and 3 we see they share:

- legs count=6
- blood=cold
- body hair=no
- eggs=true
- breastfeeding=negative

Any 4-subset of these 5 common items is frequent (support count = 2 or more). Repeating similar comparisons (for example, between Transactions 2 and 3, or 1 and 2) leads to the following distinct frequent 4-itemsets (each appears in at least 2 transactions):

1. {legs count=6, blood=cold, body hair=no, eggs=true}
2. {legs count=6, blood=cold, body hair=no, breastfeeding=negative}
3. {legs count=6, blood=cold, eggs=true, breastfeeding=negative}
4. {legs count=6, body hair=no, eggs=true, breastfeeding=negative}
5. {blood=cold, body hair=no, eggs=true, breastfeeding=negative}
6. {diet=herbivore, blood=cold, body hair=no, eggs=true}
7. {diet=herbivore, blood=cold, body hair=no, breastfeeding=negative}
8. {diet=herbivore, blood=cold, eggs=true, breastfeeding=negative}
9. {diet=herbivore, body hair=no, eggs=true, breastfeeding=negative}

Thus, replacing the placeholder <<itemsets>> with these itemsets (formatted as a Python set of strings), we get:

```
set_length_4 = {"legs count=6, blood=cold, body hair=no, eggs=true",
```

```

{"legs count=6, blood=cold, body hair=no, breastfeeding=negative"},
{"legs count=6, blood=cold, eggs=true, breastfeeding=negative"},
{"legs count=6, body hair=no, eggs=true, breastfeeding=negative"},
{"blood=cold, body hair=no, eggs=true, breastfeeding=negative"},
{"diet=herbivore, blood=cold, body hair=no, eggs=true"},
{"diet=herbivore, blood=cold, body hair=no, breastfeeding=negative"},
{"diet=herbivore, blood=cold, eggs=true, breastfeeding=negative"},
{"diet=herbivore, body hair=no, eggs=true, breastfeeding=negative"}

```

Each of these 9 itemsets appears in at least 2 transactions and has exactly 4 items.

We start by considering only the columns with actual data (ignoring the “name” column). Each transaction is then represented by its six attribute values. For the CSV

```
pgsql
```

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

```
rabbit,6,carnivore,cold,no,true,negative
```

```
human,4,herbivore,cold,no,true,negative
```

```
fox,6,herbivore,cold,no,true,negative
```

```
platypus,6,omnivore,cold,no,false,negative
```

we list the items in each transaction (ignoring “name”):

- Transaction 1 (rabbit): {6, carnivore, cold, no, true, negative}
- Transaction 2 (human): {4, herbivore, cold, no, true, negative}
- Transaction 3 (fox): {6, herbivore, cold, no, true, negative}
- Transaction 4 (platypus): {6, omnivore, cold, no, false, negative}

A first step is to remove items that are not “frequent” on their own. Since the minimal support is 2, an item must appear in at least 2 transactions to be kept. Checking the counts we have:

- **6** appears in T1, T3, T4 (support = 3)
- **4** appears only in T2 (support = 1)
- **carnivore** appears only in T1 (support = 1)

- **herbivore** appears in T2 and T3 (support = 2)
- **omnivore** appears only in T4 (support = 1)
- **cold** appears in every transaction (support = 4)
- **no** appears in every transaction (support = 4)
- **true** appears in T1, T2, T3 (support = 3)
- **false** appears only in T4 (support = 1)
- **negative** appears in every transaction (support = 4)

Thus the only frequent items are:

{ "6", "herbivore", "cold", "no", "true", "negative" }.

Now, for each transaction we consider only these frequent items:

- T1: { "6", "cold", "no", "true", "negative" }
- T2: { "herbivore", "cold", "no", "true", "negative" }
- T3: { "6", "herbivore", "cold", "no", "true", "negative" }
- T4: { "6", "cold", "no", "negative" }

The task is to find all 4-item (length-4) sets (itemsets) that appear as a subset of at least 2 transactions. After careful manual comparison, the following 9 itemsets have support ≥ 2 :

1. { "6", "cold", "no", "true" } (supported by T1 and T3)
2. { "6", "cold", "no", "negative" } (supported by T1, T3, and T4)
3. { "6", "cold", "true", "negative" } (supported by T1 and T3)
4. { "6", "no", "true", "negative" } (supported by T1 and T3)
5. { "herbivore", "cold", "no", "true" } (supported by T2 and T3)
6. { "herbivore", "cold", "no", "negative" } (supported by T2 and T3)
7. { "herbivore", "cold", "true", "negative" } (supported by T2 and T3)
8. { "herbivore", "no", "true", "negative" } (supported by T2 and T3)
9. { "cold", "no", "true", "negative" } (supported by T1, T2, and T3)

Following the required output format (a Python set of sets expressed as a string), we have:

```
set_length_4 = {{ "6", "cold", "no", "true"}, { "6", "cold", "no", "negative"}, { "6", "cold", "true", "negative"}, { "6", "no", "true", "negative"}, { "herbivore", "cold", "no", "true"}, { "herbivore",
```

"cold", "no", "negative"}, {"herbivore", "cold", "true", "negative"}, {"herbivore", "no", "true", "negative"}, {"cold", "no", "true", "negative"}}

We start by noting that the CSV has four transactions (ignoring the “name” column) with the following attribute–value pairs:

- **T1 (rabbit):** {legs count = 6, diet = carnivore, blood = cold, body hair = no, eggs = true, breastfeeding = negative}
- **T2 (human):** {legs count = 4, diet = herbivore, blood = cold, body hair = no, eggs = true, breastfeeding = negative}
- **T3 (fox):** {legs count = 6, diet = herbivore, blood = cold, body hair = no, eggs = true, breastfeeding = negative}
- **T4 (platypus):** {legs count = 6, diet = omnivore, blood = cold, body hair = no, eggs = false, breastfeeding = negative}

Because our minimal support is 2 we first count each individual item (ignoring “name”):

- **legs count:** “6” appears in T1, T3, T4 (support 3) but “4” appears only once.
- **diet:** “herbivore” appears in T2 and T3 (support 2) while “carnivore” and “omnivore” each appear once.
- **blood:** “cold” appears in every transaction.
- **body hair:** “no” appears in every transaction.
- **eggs:** “true” appears in T1, T2, T3 (support 3) while “false” appears once.
- **breastfeeding:** “negative” appears in every transaction.

Thus, the **frequent 1-itemsets** (support ≥ 2) are:

- legs count = 6
- diet = herbivore
- blood = cold
- body hair = no
- eggs = true
- breastfeeding = negative

Next, we “project” each transaction to only these items:

- **T1:** {legs=6, blood=cold, body hair=no, eggs=true, breastfeeding=negative}

- **T2:** {diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T3:** {legs=6, diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- **T4:** {legs=6, blood=cold, body hair=no, breastfeeding=negative}

We are now interested in the frequent itemsets of **length 4** (i.e. sets of exactly 4 items) that appear in at least 2 transactions. There are 6 candidate items overall. When we list all 4-item combinations that actually occur (i.e. as subsets of the transactions) and then count in how many transactions they appear, we find the following 9 frequent 4-itemsets (each with support ≥ 2):

1. **{legs=6, blood=cold, body hair=no, eggs=true}**
– Appears in T1 and T3.
2. **{legs=6, blood=cold, body hair=no, breastfeeding=negative}**
– Appears in T1, T3, and T4.
3. **{legs=6, blood=cold, eggs=true, breastfeeding=negative}**
– Appears in T1 and T3.
4. **{legs=6, body hair=no, eggs=true, breastfeeding=negative}**
– Appears in T1 and T3.
5. **{diet=herbivore, blood=cold, body hair=no, eggs=true}**
– Appears in T2 and T3.
6. **{diet=herbivore, blood=cold, body hair=no, breastfeeding=negative}**
– Appears in T2 and T3.
7. **{diet=herbivore, blood=cold, eggs=true, breastfeeding=negative}**
– Appears in T2 and T3.
8. **{diet=herbivore, body hair=no, eggs=true, breastfeeding=negative}**
– Appears in T2 and T3.
9. **{blood=cold, body hair=no, eggs=true, breastfeeding=negative}**
– Appears in T1, T2, and T3.

Finally, as requested the answer is to provide these itemsets in a Python set literal of strings (with each string listing the items, separated by commas). Thus, we have:

```
set_length_4={"legs=6, blood=cold, body hair=no, eggs=true"},
              {"legs=6, blood=cold, body hair=no, breastfeeding=negative"},
              {"legs=6, blood=cold, eggs=true, breastfeeding=negative"},
```

```

{"legs=6, body hair=no, eggs=true, breastfeeding=negative"},
{"diet=herbivore, blood=cold, body hair=no, eggs=true"},
{"diet=herbivore, blood=cold, body hair=no, breastfeeding=negative"},
{"diet=herbivore, blood=cold, eggs=true, breastfeeding=negative"},
{"diet=herbivore, body hair=no, eggs=true, breastfeeding=negative"},
{"blood=cold, body hair=no, eggs=true, breastfeeding=negative"}

```

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

LENGTH 5

We can solve this task by reasoning through the dataset manually.

First, note that the first row provides the column names. For each row (ignoring “name” because it is unique), we treat every attribute as an “item” in that transaction. In the four transactions (one per row) the following attribute-value pairs (ignoring “name”) appear:

- **legs count:** “6” appears in three transactions (rabbit, fox, platypus) and “4” appears once (human).
- **diet:** “carnivore” appears once (rabbit), “herbivore” appears twice (human, fox), and “omnivore” appears once (platypus).
- **blood:** “cold” appears in every transaction (4 times).
- **body hair:** “no” appears in every transaction (4 times).
- **eggs:** “true” appears in three transactions (rabbit, human, fox) and “false” appears once (platypus).
- **breastfeeding:** “negative” appears in every transaction (4 times).

Only those attribute–value pairs with support ≥ 2 are considered frequent. (We ignore “name” because all values there have support 1.) Thus the frequent items are:

- *legs count=6*
- *diet=herbivore*
- *blood=cold*
- *body hair=no*
- *eggs=true*

- *breastfeeding=negative*

Next, we are asked for all frequent itemsets of **length 5** (i.e. containing exactly 5 items) that appear in at least 2 transactions.

There are $\binom{6}{5} = 6$ candidate 5-itemsets. By intersecting the transactions that contain each item, we find:

1. **Itemset {legs count=6, diet=herbivore, blood=cold, body hair=no, eggs=true}:**

- *legs count=6* appears in: {rabbit, fox, platypus}
 - *diet=herbivore* appears in: {human, fox}
- Their intersection is only {fox} (support = 1), so it is not frequent.

2. **Itemset {legs count=6, diet=herbivore, blood=cold, body hair=no, breastfeeding=negative}:**

Similarly, the intersection gives {fox} (support = 1), so it is not frequent.

3. **Itemset {legs count=6, diet=herbivore, blood=cold, eggs=true, breastfeeding=negative}:**

The intersection again is {fox} (support = 1).

4. **Itemset {legs count=6, blood=cold, body hair=no, eggs=true, breastfeeding=negative}:**

- *legs count=6* appears in {rabbit, fox, platypus}
 - *eggs=true* appears in {rabbit, human, fox}
- Their intersection is {rabbit, fox} (support = 2). Including the items with universal support (blood=cold, body hair=no, breastfeeding=negative) maintains that support. This itemset is frequent.

5. **Itemset {diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}:**

- *diet=herbivore* appears in {human, fox}
 - *eggs=true* appears in {rabbit, human, fox}
- Their intersection is {human, fox} (support = 2). With the universal items added, this itemset is frequent.

6. **Itemset {legs count=6, diet=herbivore, body hair=no, eggs=true, breastfeeding=negative}:**

Again, the intersection of “legs count=6” ({rabbit, fox, platypus}) and “diet=herbivore” ({human, fox}) is {fox} (support = 1).

Thus, the only two 5-itemsets with minimal support 2 are:

- {legs count=6, blood=cold, body hair=no, eggs=true, breastfeeding=negative}
- {diet=herbivore, blood=cold, body hair=no, eggs=true, breastfeeding=negative}

Finally, as required, we format the answer as a Python set of sets in string form. The final answer is:

```
set_length_5 = {"legs count=6", "blood=cold", "body hair=no", "eggs=true",
"breastfeeding=negative"}, {"diet=herbivore", "blood=cold", "body hair=no", "eggs=true",
"breastfeeding=negative"}}
```

Based on the CSV provided, we first note that the header defines 7 attributes. However, the “name” field is unique in every row and thus never appears at least twice. In contrast, the remaining 6 attributes have the following values (per transaction):

- **legs count:** “6” appears in rows 1, 3, 4 (support 3) while “4” appears only once.
- **diet:** “carnivore” and “omnivore” appear only once each, but “herbivore” appears in rows 2 and 3 (support 2).
- **blood:** “cold” appears in all rows (support 4).
- **body hair:** “no” appears in all rows (support 4).
- **eggs:** “true” appears in rows 1, 2, 3 (support 3) while “false” appears once.
- **breastfeeding:** “negative” appears in all rows (support 4).

Notice that “cold”, “no”, and “negative” appear in every transaction. When looking for frequent itemsets of **length 5** with a minimal support of 2, we can “fix” these three items in every candidate itemset. Then, we need to add two additional items chosen from the remaining attributes such that their joint occurrence appears in at least 2 rows.

Consider the following potential candidates:

- **Candidate A:** {“cold”, “no”, “negative”, “6”, “true”}
 - “6” appears in rows 1, 3, 4
 - “true” appears in rows 1, 2, 3

The intersection is rows 1 and 3 (support 2).
- **Candidate B:** {“cold”, “no”, “negative”, “herbivore”, “true”}
 - “herbivore” appears in rows 2 and 3
 - “true” appears in rows 1, 2, 3

Their intersection is rows 2 and 3 (support 2).

Other combinations, such as using “6” with “herbivore” (which would only occur in row 3), would have a support less than 2.

Thus, the two frequent itemsets (of exactly 5 items each) meeting the criteria are:

- {"cold", "no", "negative", "6", "true"}
- {"cold", "no", "negative", "herbivore", "true"}

According to your required format, we provide the result as a Python set (formatted as a string) without column names:

```
set_length_5 = {"cold","no","negative","6","true"}, {"cold","no","negative","herbivore","true"}}
```

We first list the transactions using the CSV data (ignoring the "name" attribute since its values are unique):

- **Transaction 1 (rabbit):**
{"legs count=6", "diet=carnivore", "blood=cold", "body hair=no", "eggs=true", "breastfeeding=negative"}
- **Transaction 2 (human):**
{"legs count=4", "diet=herbivore", "blood=cold", "body hair=no", "eggs=true", "breastfeeding=negative"}
- **Transaction 3 (fox):**
{"legs count=6", "diet=herbivore", "blood=cold", "body hair=no", "eggs=true", "breastfeeding=negative"}
- **Transaction 4 (platypus):**
{"legs count=6", "diet=omnivore", "blood=cold", "body hair=no", "eggs=false", "breastfeeding=negative"}

Next, we count the support (frequency) for each item (considering only items that occur in at least 2 transactions):

- "legs count=6": appears in transactions 1, 3, and 4 (support = 3).
- "diet=herbivore": appears in transactions 2 and 3 (support = 2).
- "blood=cold": appears in all transactions (support = 4).
- "body hair=no": appears in all transactions (support = 4).
- "eggs=true": appears in transactions 1, 2, and 3 (support = 3).
- "breastfeeding=negative": appears in all transactions (support = 4).

Any item with support 1 (like "legs count=4", "diet=carnivore", "diet=omnivore", "eggs=false") is not frequent.

Now, we consider all combinations of these 6 frequent items taken 5 at a time (there are 6 combinations in total). We then check the support (i.e. the number of transactions that contain all 5 items in the combination).

Upon checking each combination manually:

1. {"legs count=6", "diet=herbivore", "blood=cold", "body hair=no", "eggs=true"}
→ Only transaction 3 contains all five items (support = 1).
2. {"legs count=6", "diet=herbivore", "blood=cold", "body hair=no",
"breastfeeding=negative"}
→ Only transaction 3 qualifies (support = 1).
3. {"legs count=6", "diet=herbivore", "blood=cold", "eggs=true",
"breastfeeding=negative"}
→ Only transaction 3 qualifies (support = 1).
4. {"legs count=6", "blood=cold", "body hair=no", "eggs=true",
"breastfeeding=negative"}
→ Transactions 1 and 3 have these items (support = 2).
5. {"diet=herbivore", "blood=cold", "body hair=no", "eggs=true",
"breastfeeding=negative"}
→ Transactions 2 and 3 have these items (support = 2).
6. {"legs count=6", "diet=herbivore", "body hair=no", "eggs=true",
"breastfeeding=negative"}
→ Only transaction 3 qualifies (support = 1).

Only combinations 4 and 5 have the required minimal support of 2.

Finally, following the required output format (a Python set of sets as a string), we have:

```
set_length_5 = {"legs count=6", "blood=cold", "body hair=no", "eggs=true",  
"breastfeeding=negative"}, {"diet=herbivore", "blood=cold", "body hair=no", "eggs=true",  
"breastfeeding=negative"}
```

The CSV (ignoring the "name" column, since its values are unique) yields these transactions when we look only at the remaining columns:

- T1: {"6", "carnivore", "cold", "no", "true", "negative"}
- T2: {"4", "herbivore", "cold", "no", "true", "negative"}
- T3: {"6", "herbivore", "cold", "no", "true", "negative"}
- T4: {"6", "omnivore", "cold", "no", "false", "negative"}

Only the values that occur in at least 2 transactions (i.e. frequent with support ≥ 2) are:

- "6" (in T1, T3, T4)
- "herbivore" (in T2, T3)
- "cold" (in all transactions)
- "no" (in all transactions)
- "true" (in T1, T2, T3)
- "negative" (in all transactions)

Now, for frequent itemsets of length 5 we need subsets of 5 items (from the frequent ones) that appear in at least 2 transactions. Looking at the transactions' intersections with the frequent items:

- T1 gives: {"6", "cold", "no", "true", "negative"}
- T2 gives: {"herbivore", "cold", "no", "true", "negative"}
- T3 gives: {"6", "herbivore", "cold", "no", "true", "negative"}
- T4 gives only 4 items: {"6", "cold", "no", "negative"} (so it does not yield any 5-itemset)

Notice that T1 and T2 already provide one each:

- The itemset from T1: {"6", "cold", "no", "true", "negative"} appears in T1 and T3.
- The itemset from T2: {"herbivore", "cold", "no", "true", "negative"} appears in T2 and T3.

Any 5-itemset that combines both "6" and "herbivore" (from T3's complete set) appears only in T3. Thus, the only two frequent 5-itemsets (each with support ≥ 2) are:

- {"6", "cold", "no", "true", "negative"}
- {"herbivore", "cold", "no", "true", "negative"}

Following the required format, the final answer is:

```
set_length_5 = {"herbivore","cold","no","true","negative"}, {"6","cold","no","true","negative"}
```

The goal is to find all 5-item sets that appear in at least 2 rows (support ≥ 2) when treating each cell (except the "name" column whose values are all unique) as an item. First, note that the "name" attribute is unique in every transaction so we exclude it. The remaining columns for each row are:

- **legs count:** 6 (or 4)

- **diet:** carnivore, herbivore, or omnivore
- **blood:** cold
- **body hair:** no
- **eggs:** true (or false)
- **breastfeeding:** negative

Looking at the data, the items that occur in at least 2 transactions are:

- From **legs count:** “6” appears in transactions 1, 3, and 4.
- From **diet:** “herbivore” appears in transactions 2 and 3.
- **blood:** “cold” appears in every transaction.
- **body hair:** “no” appears in every transaction.
- From **eggs:** “true” appears in transactions 1, 2, and 3.
- **breastfeeding:** “negative” appears in every transaction.

Thus, the frequent 1-item sets (ignoring the non-frequent ones “4”, “carnivore”, “omnivore”, and “false”) are:

- "6", "herbivore", "cold", "no", "true", "negative"

Now we need to find all 5-item combinations from these six items that are frequent. There are 6 possible combinations (since $6 \text{ choose } 5 = 6$). For each candidate 5-itemset, we check in which transactions it appears:

1. {"6", "herbivore", "cold", "no", "true"}:

- Appears only in transaction 3 (the only row that has both “6” and “herbivore”).
- Support = 1 (not frequent).

2. {"6", "herbivore", "cold", "no", "negative"}:

- Again, only transaction 3 has both “6” and “herbivore”.
- Support = 1.

3. {"6", "herbivore", "cold", "true", "negative"}:

- Only transaction 3 qualifies.
- Support = 1.

4. {"6", "herbivore", "no", "true", "negative"}:

- Only transaction 3 qualifies.
- Support = 1.

5. {"6", "cold", "no", "true", "negative"}:

- Transaction 1: has “6”, “cold”, “no”, “true”, “negative” (its diet is “carnivore” so “herbivore” is not needed here).
- Transaction 3: qualifies as it contains “6”, “herbivore”, “cold”, “no”, “true”, “negative” (and the candidate doesn’t require “herbivore”).
- Support = 2 (frequent).

6. {"herbivore", "cold", "no", "true", "negative"}:

- Transaction 2: qualifies (has “herbivore”, “cold”, “no”, “true”, “negative” with “legs count” 4).
- Transaction 3: qualifies as well.
- Support = 2 (frequent).

Thus, the two frequent 5-itemsets (minimal support ≥ 2) are:

- {"6", "cold", "no", "true", "negative"}
- {"herbivore", "cold", "no", "true", "negative"}

Finally, as requested, here is the answer formatted as a Python set of sets (formatted as a string):

```
set_length_5={{"6", "cold", "no", "true", "negative"}, {"herbivore", "cold", "no", "true", "negative"}}
```