























recipe?ri?q



```
: find ?recipe
: where [?recipe :ingredient ?i]
        [?i :unit :cups]
        [?i :quantity ?q]
        [?i :type :flour]
        [( <= ?q 2 )]
```













The first of these is the fact that the system is not in a steady state. The system is in a state of flux, and the variables are changing. This is a dynamic system, and the variables are interdependent. The second is the fact that the system is not linear. The relationships between the variables are non-linear, and the system is subject to feedback loops. The third is the fact that the system is not deterministic. The system is subject to random fluctuations, and the outcomes are uncertain. The fourth is the fact that the system is not isolated. The system is open to the environment, and the environment can influence the system. The fifth is the fact that the system is not homogeneous. The system is composed of different parts, and the parts are not identical. The sixth is the fact that the system is not static. The system is in a state of constant change, and the variables are always moving. The seventh is the fact that the system is not simple. The system is complex, and the relationships between the variables are intricate. The eighth is the fact that the system is not predictable. The system is subject to uncertainty, and the outcomes are not certain. The ninth is the fact that the system is not controllable. The system is not under human control, and the outcomes are not predictable. The tenth is the fact that the system is not measurable. The system is not quantifiable, and the outcomes are not measurable.

?recipe

: cake



**?recipe**  
:cake

```
:find ?recipe  
:where [?recipe :ingredient ?i]  
       [?i :unit :cups]  
       [?i :quantity ?q]  
       [?i :type :flour]  
       [(<= ?q 2)]
```



```
(defn pattern-left-join
  [graph partial-result pattern]
  (let [cols (:cols (meta partial-result))
        total-cols (calc-new-columns cols pattern)
        pattern->left (matching-vars pattern cols)]
```

```
;; iterate over partial-result, lookup pattern
(with-meta
  (for [left-row partial-result
        ;; convert bindings in left-row into the
        ;; pattern to lookup in the graph
        :let [lookup (modify-pattern left-row
                                      pattern->left
                                      pattern)]]
    [right-row (gr/resolve-pattern graph lookup)]
    (concat left-row right-row))
  {:cols total-cols})))
```