```
(defn simplify [g & ks] (map #(if (st/vartest? %) ? :v) ks))
(defmulti get-from-index simplify)


(defmethod get-from-index [ ?  ?  ?]
  [{idx :eav} e a v]
  (for [e (keys idx), a (keys (idx e)), v ((idx e) p)]
    [e a v]))


(defmethod get-from-index [:v  ? :v]
  [{idx :vea} e a v]
  (map vector (get-in idx [v e])))


(defmethod get-from-index [ ? :v  ?]
  [{idx :ave} e a v]
  (let [edx (idx a)] (for [v (keys edx), e (edx v)]
    [e v])))


(defmethod get-from-index [:v :v :v]
  [{idx :eav} e a v]
  (if (get-in idx [e a v]) [[]] []))
```

Query

Recipes containing <= 2 cups of flour

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