U5ZAJ15P0 : ooh, makes sense!
U5ZAJ15P0 : Hi! I have the following in a Datalog query:```
'[:find ... :where [(wef-backend.app.identity/past-expiry? ?expiry)]]]
```

U5ZAJ15P0 : Due to how Datomic/Datalog works I must fully qualify the past-expiry? function, even though it is available in the namespace where I wrote this query
U5ZAJ15P0 : is there some clever trick that would let me omit the full qualifier?
U5ZAJ15P0 : I tried syntax quoting on the query but that doesn't work, as it fully qualifies the `?expiry, etc` variables too
U5ZAJ15P0 : Essentially I would like to fully qualify some symbols in that quoted expression
U5ZAJ15P0 : but not others
U2H58V4P2 : I don't recall any problems with namespaced keywords, but I seem to remember I had problems with namespaced maps. As these were limited to my test classes, I added these namespaces to the ignore list I think. Also, I think there were some fixes on the master branch that remain unreleased. A bit vague I realise, but if you're still stuck tomorrow (unlikely) then I'll check when I'm back in front of a computer...
U051SS2EU : <@U5ZAJ15P0> you can use ` for this, and selectively use ~ for the symbols you want to namespace qualify
U5ZAJ15P0 : <@U051SS2EU> how so? I tried variations on your suggestion but didn't manage to get it to work
U051SS2EU : ```peregrine.circle=&amp;gt; `[:find ... :where [identity ~'?foo]][:find ... :where [clojure.core/identity ?foo]]```

U051SS2EU : I just picked a random thing with identity in the name, but I hope it translates
U5ZAJ15P0 : ah
U051SS2EU : I guess I should have explicitly mentioned the `~'` idiom to prevent namespacing
U5ZAJ15P0 : is there a way to do the opposite? selectively pick the symbols I want to namespace qualify
U051SS2EU : that's harder
U5ZAJ15P0 : instead of the ones I do not want to namespace qualify
U5ZAJ15P0 : mmh ok
U5ZAJ15P0 : nevermind then
U5ZAJ15P0 : thanks :slightly_smiling_face:
U051SS2EU : this is close, but no cigar ```peregrine.circle=&amp;gt; '[:find ... :where [`identity ?foo]][:find ... :where [(quote clojure.core/identity) ?foo]]```

U5ZAJ15P0 : yep, not quite :confused:
U5ZAJ15P0 : why wouldn't this work:```
'[:find ... :where [~`identity ?foo]]
```
?

U051SS2EU : ~ doesn't make sense outside `
U5ZAJ15P0 : why does it make sense here:```
`[:find ... :where [identity ~'?foo]]
```
?

U051SS2EU : it's inside the ` at the beginning of the vector
U051HUZLD : thanks
U5ZAJ15P0 : right, but why does swapping out the use of syntax quoting vs normal quoting makes it different in use?
U5ZAJ15P0 : (I'm not familiar enough with quoting yet)
U051SS2EU : because ~ is only valid in syntax quote
```