Lisp in Summer Projects Submission

Submission Date	2013-10-15 07:42:25
Full Name	Jonas Enlund
Country	Finland
Project Name	Learn Datalog Today!
Type of software	other
General category	documentation/web-page
LISP dialect	Clojure
GitHub URL	https://github.com/jonase/learndatalogtoday/tree/lisp
Did you start this project?	Yes, all the code is written by me
Project Description	I want to upload a free-form 3-4 page PDF composition.
Upload 3-4 page detailed PDF	learndatalogtoday.pdf
Build Instructions	Follow the instructions at https://github.com/jonase/learndatalogtoday/tree/lisp#bootstrapping
Test Instructions	There are no tests
Execution Instructions	Follow the instructions at https://github.com/jonase/learndatalogtoday/tree/lisp#running-the-webapp
Describe any bugs or caveats	There's a bug in the lisp branch which prevents the exercises to work with Firefox. This has been fixed in the master branch and the running program at learndatalogtoday.org
Screen shots	learndatalogtoday.png
Official	I have read rules and have abided by them. I am 18 years of age or older. I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.

Learn Datalog Today!

www.learndatalogtoday.org

Purpose

Learn Datalog Today! is a website dedicated to teaching the basics of the datomic[1] flavour of datalog[2]. Datalog is a declarative database query language with roots in logic programming. The goal is that a competent developer interested in exploring these particular technologies should be able to grasp the basics after one day of study.

Function

The website is structured much like a textbook where the learning material is divided into chapters with accompanying exercises. The exercises are interactive in the sense that the student will interact with a “real” database when writing datalog queries. Each chapter introduces a new concept of datalog and the exercises are a set of questions. The answer to each question is a user written query which will run against the live database in order check if it yields the expected result. This way the reader will get immediate feedback on whether he has understood the material covered in the chapter.

An example exercise might look like this:

> *“Given a collection of actors and (the now familiar) ratings data. Find the average rating for each actor. The query should return the actor name and the average rating.”*

To which the student should write a query equivalent to

```
[:find ?name (avg ?rating)
 :in $ [?name ...] [[?title ?rating]]
 :where
 [?p :person/name ?name]
 [?m :movie/cast ?p]
 [?m :movie/title ?title]]
```

[1] http://www.datomic.com/
[2] http://en.wikipedia.org/wiki/Datalog

Motivation

The motivating factors for creating *Learn Datalog Today!* were

- Datalog is a beautiful query language that deserves the recognition of a much wider audience.
- There has been a resurgence in the interest of datalog and other logic programming concepts in the Clojure community lately due to the release of Datomic and the logic library core.logic. Therefor, there is a need for basic training material.
- Writing an extensive tutorial in a foreign language seemed to be an interesting personal challenge.

Audience

The target audience is anyone who is interested in these technologies or are evaluating different databases systems or query languages. I'm also hoping it will be used by people new to Datomic who are faced with the task of using or maintaining systems using it.

Methodology

The web server is written in Clojure and hosted on Heroku. The chapter texts are written in markdown, compiled to html on startup and served to users upon request using the Compojure routing library.

Part of the design of datomic is that storage is an abstraction. Different storage engines can be used depending on system needs. The requirements for a query tutorial are very modest. The system runs on heroku and uses in-memory storage only. When the process starts it loads the example database and immediately grabs an immutable database value. The contents of the database is large enough to be able to write interesting queries but not so large as to be able to bring down the system by writing exponentially slow queries.

Datalog queries in datomic are ordinary data structures so it's easy to send them from the client over the wire to the server. When the server receives a query it validates and sanitizes it so that no queries which might compromise the server will to run. Two queries are run against the database: The user supplied query and a correct query stored on the server. The results are compared and sent back to the client along with a message whether the answer was correct or not.

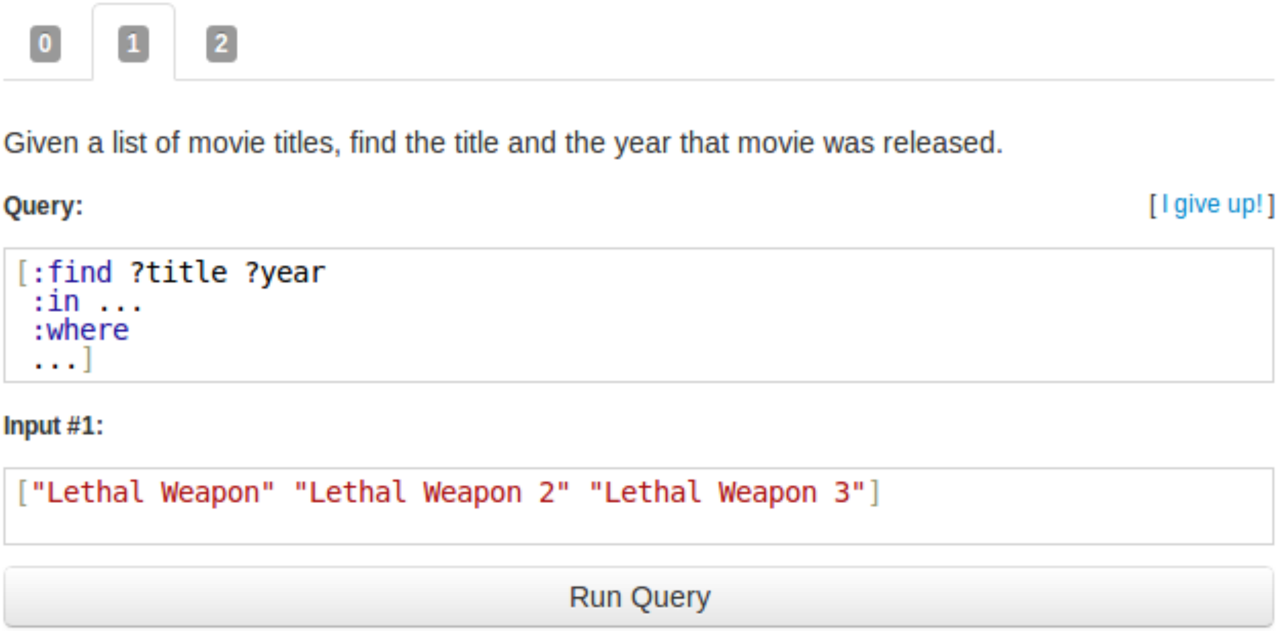
The client side is written in ClojureScript and uses CodeMirror as text editor which provides nice syntax coloring. When results from a query arrives a message is shown indicating whether the query he wrote was a correct or invalid query. If the query was invalid the user can try again or request to be shown the correct query.

Most of the work spent on this project was not actually writing the server (or client) code but instead time was spent on writing chapters and figuring out good exercises.

Conclusion

The web site has been running continuously since early august and has had daily users ever since. I'm not planning any big changes or additions to the tutorial. If Datomic releases new datalog features I will probably add chapters covering them. I also occasionally receive valuable feedback from users whose suggestions I take into account and add small clarifications or additional explanations.

Screenshot:



Given a list of movie titles, find the title and the year that movie was released.

Query:

```
[:find ?title ?year
 :in ...
 :where
 ...]
```

Input #1:

```
["Lethal Weapon" "Lethal Weapon 2" "Lethal Weapon 3"]
```