



data.world

A Platform for Global-Scale Semantic Publishing

<http://ow.ly/zwF030g2ryo>

Bryon Jacob[0000-0003-0470-9300] bryon@data.world,

Dave Griffith[0000-0001-9700-0012] dave.griffith@data.world,

Triet Le[0000-0001-5619-5802] triet.le@data.world

WHAT

is data.world?



data.world

Is a collaborative web platform with

- a user base that are (primarily) not Semantic Web experts
- datasets that are (generally) not published as linked data (mostly tabular/JSON data)

We use web standards to translate that data into RDF



The screenshot displays the Data.world interface for a dataset named "U.S. Zipcodes". On the left, a "Data dictionary" section shows metadata for the dataset, including a total of 604,674 triples and 1,000+ distinct entities. Below this, namespaces and distinct classes/predicates are listed. The main area shows a "Workspace" for the dataset, where a SPARQL query is being edited. The query is highlighted in pink and is as follows:

```
1 # Using the linked zipcode DB
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
4 PREFIX : <https://bryon.linked.data.world/d/us-zipcodes/DRAFT#>
5
6 SELECT ?zipcode ?zip ?city ?state ?timezone ?lat ?long
7 WHERE {
8   ?zipcode a      :ZipCode;
9            :city  ?city;
10           :state ?state;
11           :timezone ?timezone;
12           :zip   ?zip;
13           geo:lat ?lat;
14           geo:long ?long
15
16   FILTER (?zip = "78731")
17 }
18
```

The query results are displayed in a table with 7 rows and 5 columns: zipcode, zip, city, state, and timezone. The results are as follows:

zipcode	zip	city	state	timezone
:DRAFT#78731	78731	Austin	http://ontologi.es/place/US-TX	http://www.w3.org/2006/timezone-us#AKST
:DRAFT#78731	78731	Austin	http://ontologi.es/place/US-TX	http://www.w3.org/2006/timezone-us#AST
:DRAFT#78731	78731	Austin	http://ontologi.es/place/US-TX	http://www.w3.org/2006/timezone-us#CST
:DRAFT#78731	78731	Austin	http://ontologi.es/place/US-TX	http://www.w3.org/2006/timezone-us#EST
:DRAFT#78731	78731	Austin	http://ontologi.es/place/US-TX	http://www.w3.org/2006/timezone-us#HST

RDF Formats are supported natively, and gets you a SPARQL endpoint to query the data

The screenshot displays the Data.world workspace interface. On the left, there are two data views: 'products.tsv' showing drug ingredients and routes, and 'reference.xlsx' showing exclusion codes. The main workspace area contains a 'TABULAR FILES (4)' list with 'exclusivity.tsv', 'patents.tsv', 'products.tsv', and 'reference.xlsx'. Below this is a 'QUERIES (7)' section with a 'New query' button. The central pane shows a SPARQL query titled 'Schema by File' with the following code:

```

1 # Schema by File
2 # For each file, show the columns and their associate types
3
4 PREFIX csvw: <http://www.w3.org/ns/csvw#>
5 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
6 PREFIX dw: <http://data.world#>
7
8 SELECT ?FileName ?Table ?Column ?Type ?ColumnIRI
9 WHERE {
10   ?tg a csvw:TableGroup .
11   ?tg csvw:base ?baseUrl .
12   ?tg csvw:table ?tableObject .
13   ?tableObject csvw:tableSchema ?s .
14   ?tableObject csvw:url ?url .
15   ?tableObject dw:name ?TableName .
16   OPTIONAL {
17     ?tableObject dw:file ?FileName .
18   }
19   ?tableObject dw:file ?FileName .

```

The query results table shows the following data:

FileName	Table	Column	Type
exclusivity.tsv	exclusivity	appl_type	xsd:string
exclusivity.tsv	exclusivity	appl_no	xsd:string
exclusivity.tsv	exclusivity	product_no	xsd:string
exclusivity.tsv	exclusivity	exclusivity_code	xsd:string
exclusivity.tsv	exclusivity	exclusivity_date	xsd:string
patents.tsv	patents	appl_type	xsd:string

On the right, the 'Dataset schema' panel shows a hierarchical view of the schema for 'exclusivity.tsv', 'patents.tsv', and 'products.tsv', including fields like 'appl_type', 'appl_no', 'product_no', 'patent_no', 'patent_expire_date_text', 'drug_substance_flag', 'drug_product_flag', 'patent_use_code', 'delist_flag', 'ingredient', 'df_route', 'trade_name', 'applicant', 'strength', and 'appl_type'.

Tabular formats are also ingested – an RDF model of the data is built using CSVW.

The screenshot shows the data.world interface with a SPARQL query editor and a results table. The query is as follows:

```

1 PREFIX : <https://markmarkoh.linked.data.world/d/us-state-table/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3
4 SELECT ?abbreviation ?area ?circuit_court
5 FROM NAMED <https://data.world/markmarkoh/us-state-table>
6 WHERE {
7   GRAPH <https://data.world/markmarkoh/us-state-table> {
8     ?state :col-state_table-abbreviation ?abbreviation;
9           :col-state_table-circuit_court ?circuit_court.
10  }
11  ?state :wikidataEntity ?wd_state.
12  SERVICE <https://query.wikidata.org/sparql> {
13    ?wd_state wdt:P2046 ?area.
14  }
15 }
16 ORDER BY DESC(?area)

```

The results table shows the following data:

abbreviation	area	circuit_court
AK	1717856	9
TX	696241	5
CA	423970	9
VT	381154	8

- Each dataset in data.world is accessible as a *named graph* (provided you have access permissions)
- Remote SPARQL endpoints can be federated naturally via SERVICE patterns

WHY

does it exist?

We think more people should care about the Semantic Web.

The real question is *"Why don't they?"*

Semantic Web Products – Solutions in search of problems?

In order to make the Semantic Web appeal to mainstream data workers, we need products that focus on their needs.

*We need to find the real-world **problems** that are searching for a **solution** that the Semantic Web can provide.*

N% of Data Scientists' time is spent finding and prepping data.

Is an unsubstantiated claim that is thrown around a lot.

Nonetheless – it is a real problem, data workers do spend a lot of time in discovery and exploratory analysis of data – and the real crime is that work is done redundantly.

Semantic Web offers a powerful set of tools to deal with this:

- a universal structure for data
- an open-world model that adapts to heterogeneous data
- metadata can be added to the data
- SPARQL for federated query



The set of people working on data projects is diverse

- knowledge engineers who work directly in the creation of ontologies and knowledge bases
- data scientists, machine learning experts, and statisticians who produce models, derivative data works, and visualizations
- analysts, scientists, and students who are accustomed to using spreadsheets or visually-driven analytics platforms
- stakeholders and end-users consuming the high-level conclusions of the work.

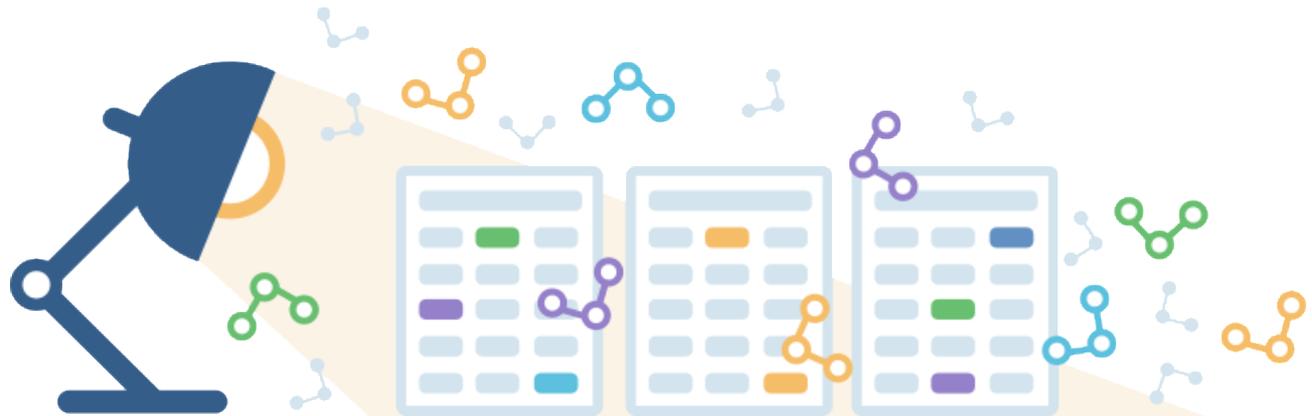


Each of these personas has a process and a tool chain they prefer – we do a lot of user-centered design work that focuses on the elements that support collaboration between them. The unifying factor is the standard data model.

Working on data is not a one-way conversation

Rather than a data “portal” where there are a small, fixed number of data producers publishing data to data consumers, data.world focuses on collaborations where each actor can play both the producer and consumer roles.

Data can be worked on completely in the open, completely privately, or in groups of collaborators with read/write ACL.



The screenshot shows a web browser window with the URL <https://data.world/bryon/usda-ld>. The page displays a dataset with 16 files. A modal window titled "227 file warnings" for the file "DATA_SRC.csv" is open. The modal is divided into two sections: "Structural (5)" and "Numeric (11)".

Structural (5)

5 blank cells detected

- Cell at row 100, column authors appears blank.
- Cell at row 171, column authors appears blank.
- Cell at row 43, column year appears blank.
- Cell at row 100, column year appears blank.

+ 1 similar issue

Numeric (11)

11 numeric values outside standard deviation detected

- Value 11807.0 at row 118, column end_page is more than 4 standard deviations of 1748.49 from the mean of 802.39.
- Value 9445.0 at row 150, column end_page is more than 4 standard deviations of 1748.49 from the mean of 802.39.

At the bottom of the modal, there are two buttons: "Upload new file" and "Done".

Data inspector powered by RDF & SPARQL

Most data is tabular, many more people know SQL than SPARQL

The screenshot shows a web interface for a data.world workspace. The main area displays a SQL query:

```
1 SELECT *
2 FROM shootingscitystate
3 JOIN gmoney.us_city_temp_ranges.difference_in_temps delta
4 ON shootingscitystate.citystate = delta.city
```

Red boxes highlight the table names in the query. A red arrow points from the `gmoney.us_city_temp_ranges` table name to a URL: https://data.world/gmoney/us_city_temp_ranges. Another red arrow points from the `shootingscitystate` table name to a file named `shootingscitystate.csv` in the 'TABULAR FILES' section.

The 'Dataset schema' panel on the right shows the structure of the `shootingscitystate` dataset:

- `shootingscitystate`
 - # id
 - name

The 'Results table' shows 401 query results:

state	id	name	date	mann
LA	1252	Eric Harris	2016-02-08	shot
AZ	1781	Ruben Horacio Strand Alvear	2016-08-13	shot
LA	417	Jared Johnson	2015-04-28	shot
LA	1085	Calvin McKinnis	2015-12-14	shot
LA	2254	Arties Manning	2017-01-24	shot
LA	308	Richard White	2015-03-20	shot

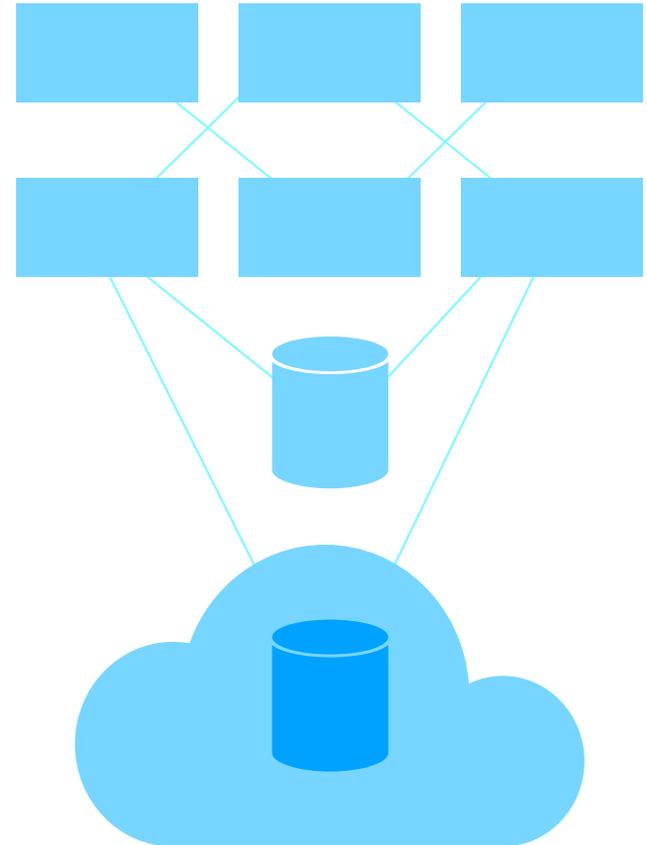
The interface includes a sidebar with 'TABULAR FILES (2)', 'OTHER FILES (2)', 'DOCUMENTS (2)', and 'QUERIES (7)'. The 'QUERIES' section shows two recent queries, with the top one being 'Join City Temp Ranges with Sh...' by @bryon, 2 minutes ago.

HOW

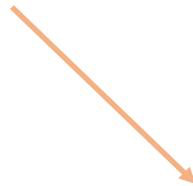
does it work?

Our query architecture prioritizes scale and query responsiveness over raw performance and update flexibility.

Updates as bulk ingest, with the output of the ingest pipeline an immutable HDT file.



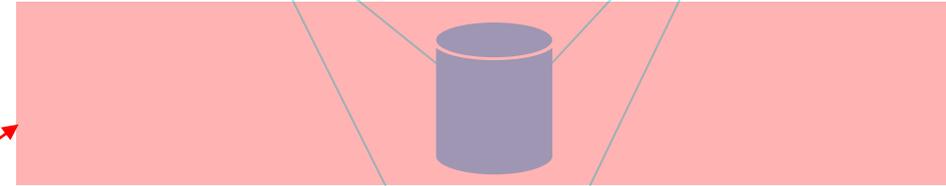
Front-end query servers parse user queries and route them (this is where SQL is transpiled)



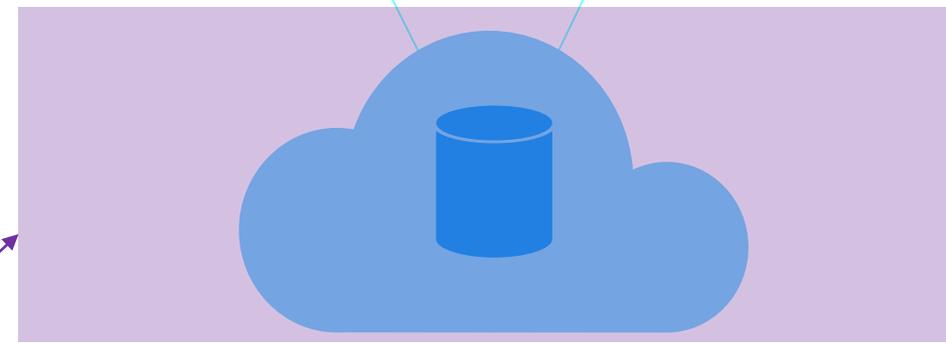
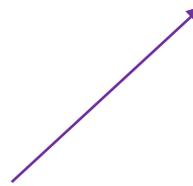
Query heads load HDT data on-demand from deep storage and execute SPARQL



HDT and indices are cached in a fast storage layer, leveraging HDT's query-in-place



Data is stored at rest in inexpensive, scalable cloud storage



WHAT

are we working on now?

More tools to enrich data with schema, metadata, and reconciliation against standard taxonomies/entities...

The screenshot shows a web browser window with the URL <https://data.world/bryon/usda-ld>. The page displays a dataset named 'DATA_SRC.csv' (80.94 KB) with 16 files. A modal window is open for editing the schema, showing the 'Column details' tab. The columns and their data types are:

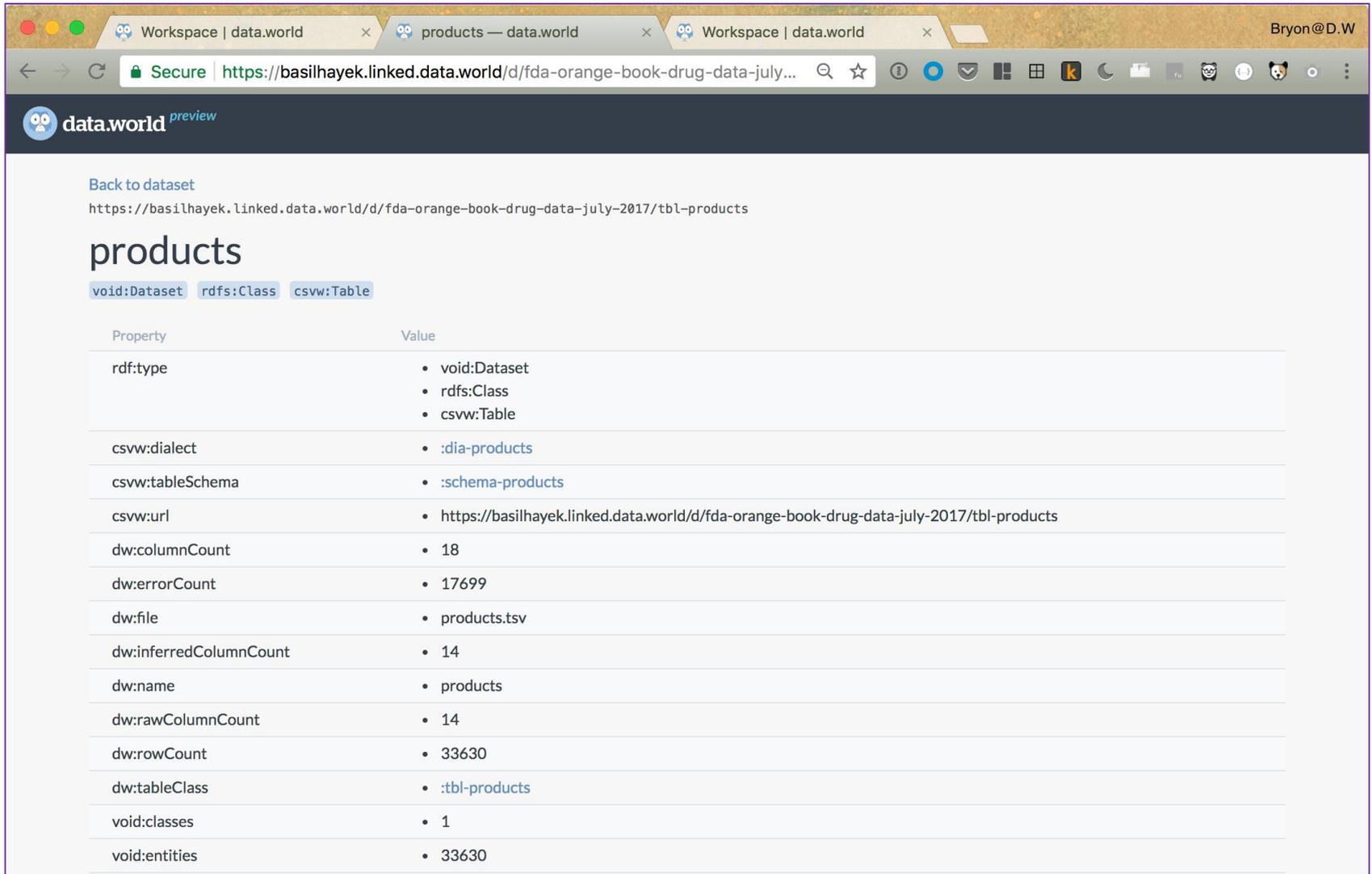
- datasrc_id**: string (dropdown menu is open showing options: String, Integer, Decimal, Boolean, URL)
- authors**: (no data type specified)
- title**: string
- year**: year

Sample data values are shown in preview boxes:

- datasrc_id**: D1066, S986
- authors**: , BJ Ector, n/a
- title**: 2011 U.S. Pulse Quality ... kimchi
- year**: (empty)

The modal includes 'Cancel' and 'Save' buttons at the bottom.

Entity browser enhancements – HTML templating for browse interfaces...



The screenshot shows a web browser window with three tabs. The active tab is 'products — data.world'. The address bar shows the URL: <https://basilhayek.linked.data.world/d/fda-orange-book-drug-data-july-2017/tbl-products>. The page header includes the 'data.world preview' logo and the user 'Bryon@D.W'. Below the header, there is a 'Back to dataset' link and the URL. The main content is titled 'products' and includes three tabs: 'void:Dataset', 'rdfs:Class', and 'csvw:Table'. A table below lists various properties and their values.

Property	Value
rdf:type	<ul style="list-style-type: none">• void:Dataset• rdfs:Class• csvw:Table
csvw:dialect	<ul style="list-style-type: none">• :dia-products
csvw:tableSchema	<ul style="list-style-type: none">• :schema-products
csvw:url	<ul style="list-style-type: none">• https://basilhayek.linked.data.world/d/fda-orange-book-drug-data-july-2017/tbl-products
dw:columnCount	<ul style="list-style-type: none">• 18
dw:errorCount	<ul style="list-style-type: none">• 17699
dw:file	<ul style="list-style-type: none">• products.tsv
dw:inferredColumnCount	<ul style="list-style-type: none">• 14
dw:name	<ul style="list-style-type: none">• products
dw:rawColumnCount	<ul style="list-style-type: none">• 14
dw:rowCount	<ul style="list-style-type: none">• 33630
dw:tableClass	<ul style="list-style-type: none">• :tbl-products
void:classes	<ul style="list-style-type: none">• 1
void:entities	<ul style="list-style-type: none">• 33630

More problems to help users solve in: dataset versioning, auditing, governance...

The screenshot shows a web browser window displaying a dataset page on data.world. The browser's address bar shows the URL: <https://data.world/basilhayek/fda-orange-book-drug-data-july-2017>. The page title is "basilhayek/FDA Orange Book Drug Data July 2017" with a green "OPEN" badge. The page is updated on Oct 20 and is under a Public Domain License. The user "Bryon@D.W" is logged in.

The page has tabs for "Overview", "Contributors", "Discussion", and "Activity". The "Activity" tab is selected, showing a list of updates:

- @basilhayek updated the column type in patents.tsv.** 2 days ago · [Show changes](#)
- @basilhayek updated the column descriptions in reference.xlsx.** 22 days ago · [Show changes](#)
- @basilhayek updated the labels of reference.xlsx.** 22 days ago · [Show changes](#)
- @basilhayek updated the description of reference.xlsx.** 22 days ago · [Show changes](#)
- @basilhayek uploaded reference.xlsx.** 22 days ago
- @basilhayek uploaded reference.xlsx.** 22 days ago
- @basilhayek updated the tags.** 23 days ago · [Show changes](#)
- @basilhayek updated the summary.**

On the right side, there are sections for "Details" and "Activity". The "Activity" section shows "QUERIES (9)" with the following items:

- Distinct Predicates** @bryon · 7 hours ago [SPARQL](#)
- Products by patent expiration** @bryon · 7 hours ago [SPARQL](#)
- exclusivity (exclusivity.tsv)** [SQL](#)
- patents (patents.tsv)** [SQL](#)
- products (products.tsv)** [SQL](#)

Below the queries, there is a "TAGS (2)" section with tags for "health" and "pharmaceuticals". At the bottom, the "CONTRIBUTORS (1)" section lists "Basil Hayek @basilhayek" with a question mark icon.

Query Architecture

Our HDT-based query architecture works well for scaling out many exploratory queries, but it's not optimal for large analytical (non-selective) queries.

We're pushing the limits of HDT here, and also contemplating a hybrid architecture where large analytical queries are run against columnar formats such as parquet files.

Our hypothesis...

To increase the size and connectivity of the web of Linked Data, we need to increase the connectivity of the network of people working with data.

By using Semantic Web technology to facilitate data work, we can leverage that work to grow the web of Linked Data.

Create an account:

<https://data.world/semweb>

Use it, and send me feedback:

<https://data.world/bryon>
bryon@data.world

Thank You!!