

The logo for LINDA Query Designer features a stylized graphic on the left consisting of three overlapping teardrop shapes: a large grey one at the top, a smaller orange one at the bottom left, and a smaller grey one at the bottom right. To the right of this graphic, the word "LINDA" is written in a bold, orange, sans-serif font, followed by the words "Query Designer" in a bold, black, sans-serif font.

LINDA Query Designer

In a nutshell

Create simple or complex linked data queries in a drag-n-drop manner, similar to SQL Query Designers of relational database management systems.

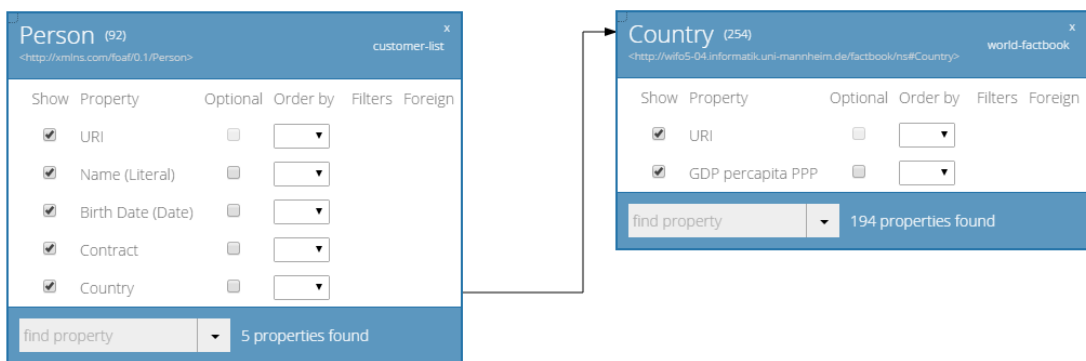
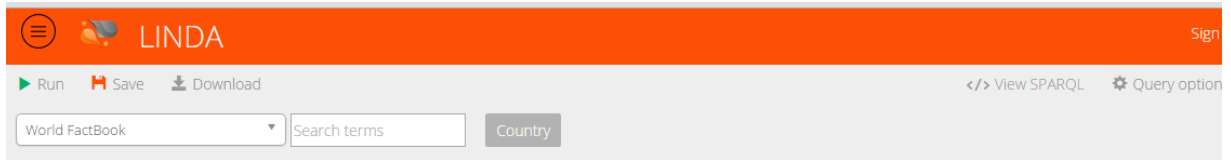
Description

The LinDA Query Designer is a graphic tool that facilitates the needs of users who want to query linked data. With LinDA Query Designer you can create complex queries, join multiple data sources and apply advanced filters with a few clicks. The user selects one (or more) sparql endpoints / stored rdf and the LINDA Query Designer auto-detects the available classes and object properties. The items are presented with pagination, and they can be filtered via the "Search terms" input box. The user selects the classes that he desires and drags them to the Query Designer Canvas. The system auto-detects the available properties of the classes and the user selects the properties that he/she wishes to include in the query. The Query Designer prompts the number of instances of each property / class as an indication for the user for the popularity of the class. For each property the user is able to add filters and order by clauses. The user can then click the run button and get the results of the query. No prior knowledge of the SPARQL language is needed, although the user can see in real-time the SPARQL query that is being constructed.

Moreover the user can link any number of classes together in order to create more complicated queries. The classes can reside in different SPARQL endpoints through the use of the Federated Query of SPARQL 1.1.

Indicative Screenshots

Screenshot #1



Screenshot #2

The screenshot shows the LINDA web interface. At the top, there's a navigation bar with a menu icon, the LINDA logo, and a 'Sign in' link. Below that, there are buttons for 'Run', 'Save', and 'Download', along with a 'View SPARQL' button and 'Query options'. A search bar contains 'DBPEDIA' and 'Search terms'. Below the search bar are several buttons for different output formats: 'Quad Map Format', 'Quad Storage', 'Array of Quad Map', 'Quad Map', 'Array of Quad Map Format', 'Quad Map Value', 'Array of Quad Map ATable', 'Array of Quad Map Column', 'Quad Map Column', and 'Quad Map FText'. A status bar indicates 'Showing 1-10/24.424 terms'. On the right, there's a 'Feedback' button. The main content area shows a search for 'Person (2.137.101)' in 'dbpedia'. A modal window is open for 'Person (2.137.101)', showing a search for properties with a dropdown menu listing various properties like 'Person Fullname', 'Person Alt Publication Name', 'Has Authored Web Page', 'Has Publication List RDF', and 'Has Authored Email'. The modal also shows '296 properties found' and 'results 26 - 50'. On the right side of the modal, there's a list of results with expandable arrows: 'Statement (64.000.750)', 'Article (43.411.837)', 'Item (17.725.000)', 'Document (11.539.262)', 'Thing (4.860.277)', 'Agent (2.371.260)', 'Physical Entity100001930 (2.250.653)', 'Person (2.137.101)', 'Person (2.135.040)', 'Natural Person (2.135.040)', 'Globe Coordinates Value (2.131.903)', 'Yago Legal Actor Geo (1.815.297)', 'Yago Permanently Located Entity (1.2...', 'Concept (1.213.462)', and 'Abstraction100002137 (726.957)'. There's also a 'Feedback' button on the right side of the modal.

Innovations

- By adopting the approach of SQL wizards of popular database management systems, the Query Designer lowers the learning curve of linked data querying. With simple drag n drop functionality, a user is able to perform a simple query without any previous knowledge of either linked data or SPARQL.
- In contrast to current linked data visual, graph-based exploration tools (e.g LodLive, Facete, OntoWiki browser, etc.) the Query Designer is not limited to the unfolding of one specific class but has the ability for matching triple patterns and conjunctions between any numbers of classes.
- The Query Designer takes a different approach from existing visual query builders for the construction of the query. Existing visual query builders prompt users to put nodes and links for the construction of a query (e.g Virtuoso iSPARQL). In this context, visual query builders are considered more as a visual aid for constructing the query for experts rather than a query tool for non-linked-data-savvy users. The basic usage workflow of the Query Designer on the other hand is directly targeted to non-expert users.
- Even for users who do know SPARQL Queries really well, the Query Designer eliminates the need to browse / reference the available classes and properties of an endpoint or triple store. Usually you have to know the exact structure of the stored RDF in an SPARQL

endpoint before you can create a query. With LinDA Query Designer you don't have to know anything about the SPARQL endpoint(s). You just reference the data source and the system automatically prompts you of what is inside. This is an invaluable functionality, for both experts and novice users.

- The Query Designer can significantly lower the required speed and manual input of a SPARQL query. For instance a simple drag and drop of two classes, a selection of 3-4 properties and a quick set of order by at the desired property can easily substitute 10-15 lines of SPARQL code. For more complicated queries, this benefit becomes even more apparent. Larger queries are much easier managed through the Query Designer.
- The Query Designer allows out-of-the-box, the ability to drag and drop classes even if they belong to different SPARQL endpoints. This functionality promotes the interlinking power of linked data and reveals a substantial advantage in comparison to working with other data models in isolated data silos. This feature is completely transparent to the user, while under the hood, the query designer takes advantage of the Federated Query syntax and auto-generates all the necessary SPARQL code.
- Real-time, dynamic exploration of public SPARQL endpoints in the web relies on the ability of underlying tools to efficiently interact with and extract useful information from datasets containing hundreds of millions of RDF triples. A straightforward approach of constructing greedy queries, executing them and waiting for results that will be presented to the user is not realistic in the case of a web based tool for linked data queries. Internally, the Query Designer addresses a number of major performance challenges in order to provide its services to the user with speed and reliability. Identifying all categories of objects and object-to-object relationships in real world data sources can't be entirely based on data source metadata (like the ontology been added alongside with the actual data), as such metadata are often out of date or even non-existent. As a result, the queries used to identify classes and properties must be based on the actual data while still returning all or at least the most probably relevant entities in tolerable waiting times. Moreover, filtering and (Ajax-based) pagination techniques are utilized in the front-end, as even modern browsers' performance drops dramatically for documents containing more than a few thousands DOM elements. While entities are loading, the tool remains usable and allows the exploitation of the information already fetched from the endpoint

Current list of features

- Ability to load datasets live from an endpoint or a stored in a triple store
- With simple drag n drop functionality users can perform SPARQL queries
- On-the-fly and ultra-fast loading of all available classes and properties found in a dataset
- On-the-fly loading of classes linked through a predicate (e.g. drag-n-drop of dbpedia author will bring Work and Person classes linked with the Author property)
- Display the short name of a class but also the full URI to avoid conflicts

- Ability to see all available owl:classes and subclasses in hierarchy (right panel)
- Ability to see the number of instances of available owl classes and subclasses. Order classes based on number of available instances
- Display number of instances of a selected class
- Autocomplete search on available classes,properties
- Scalable performance and working on the background to handle loading of thousands of properties and classes
- Pagination of classes / properties search results
- Ability to drag-n-drop multiple classes on the designer canvas
- Browse, find and select Properties contained within selected classes
- Display number of instances of properties
- Ability to show / hide a particular property
- Ability to set a particular property as optional
- Sort results by selecting "order by" clause of a property (ASC / DESC in case of number / date)
- Create multiple filters per property (String, Number, Date, URL)
- Conditional operands on multiple filters ("AND" or "OR" operands). Ability to create a custom boolean expression
- Autocomplete search on available URI resources when selecting a URI filter
- Link properties belonging to different classes (add a connection between two or more classes similar to INNER JOIN of database tables)
- Auto-generate the equivalent SPARQL Query
- Display and modify if required the equivalent SPARQL Query
- Run the SPARQL Query and display the results
- Enable pagination for the results
- Download the results using the RDF2Any (JSON, CSV, PDF, SQL script, custom output)
- Ability to save / load a query created with the Query Designer
- List the queries created with the Query Designer. Right-click to load them in order to run or modify them
- Perform federated queries between classes of different endpoints (endpoints need to support SPARQL 1.1)
- Perform UNION and MINUS queries between classes
- Ability to set a custom pattern formula (UNION, MINUS) between classes
- Set DISTINCT, LIMIT and OFFSET Clauses for triple patterns

Applicability

The LinDA Query designer can be used, installed or integrated easily for any project that needs linked data querying in ANY domain.

Demo

You can play with LinDA Query Designer at:

<http://linda.epu.ntua.gr/query-designer/>

You can add more resources through the Data Sources section:

<http://linda.epu.ntua.gr/datasources/>

Tutorial examples

Follow this example to obtain a basic understanding of how the LinDA Query Designer works. You can follow a live version of this tutorial (<http://linda.epu.ntua.gr:8585/query-designer/tutorials/overview/>)

1. In the LinDA Workbench, on the left-side main menu of the website click the "Queries" link.
2. On the top right of the page click the "Query Designer" green button.
3. In the gray toolbar on top, a select list contains the datasources connected with LinDA. Choose the IMDB data source.
4. Next to the datasource select, a list of the item types (*Classes*) appears. You can use the search box to filter them.
5. Drag and drop the "Film" button in the white canvas area below (workspace) to find films in IMDB.
6. You can see a new instance named "Film" has been added to the workspace. By default it only contains the film's URI, which works like an ID. From the select in the lower part of the Film instance choose the "label" and "initial release date". You can also type in the input to find properties faster.
7. After adding the properties, click the "Run" button in the toolbar (the one with the green arrow). A list of IMDB films with their URIs, labels and initial release dates will appear.
8. Uncheck the "show" checkbox of the URI property of the film and run the query again. You can also press "F9" to run the query.
9. By default you see the first 100 results, click on the "Fetch results 101 to 200" on the top right of the results to see the next page.
10. From the toolbar, drag and drop the "Actor" class to the workspace.
11. In the Film instance, add the "Actor" property. Then, move the arrow over the actor property row of the Film, click the "add" link in the foreign key column, and move the mouse over the new "Actor" instance.
12. As you move the mouse over the Actor instance, an arrow appears connecting the two instances. Click the Actor instance to make this arrow permanent and create a connection between the two instances.
13. The arrow that was created specifies a relationship like a foreign key would in a relational database: The actor property of the Film is now a "foreign key" to the Actor instance.
14. Add the label property of the Actor, uncheck the "hide" checkbox next to its URI and run the query again.
15. Each line of the result now contains a film, the film's release date and an actor that played in this film. To order the results, in the Film instance, in the order by column of the "initial release date" property, select the "DESC" option and run the query again. In the results now, the films are ordered from the most recent to the older.

Filters and Aggregates

Follow this example to obtain learn how to add various filters and aggregate functions in your queries. You can follow a live version of this tutorial here: http://linda.epu.ntua.gr:8585/query-designer/tutorials/filters_and_aggregates/.

1. In the LinDA Workbench, on the left-side main menu of the website click the “Queries” link.
2. On the top right of the page click the “Query Designer” green button.
3. In the gray toolbar on top, a select list contains the datasources connected with LinDA. Choose the Eurostat data source.
4. Drag and drop the Countries button in the workspace. In the new instance, add the “Level of internet access” and “Geocode” properties.
5. Click the run button or press “F9” to see the results. The first two results (http://wifo5-04.informatik.uni-mannheim.de/eurostat/resource/countries/European_Union_Eu-27) and (http://wifo5-04.informatik.uni-mannheim.de/eurostat/resource/countries/European_Union_Eu-25) do not represent actual countries, but different subdivisions of the European Union. In order to filter them out do the following:
 - a. On the Filter column, click the edit link of the Geocode property.
 - b. In the filter dialog you can specify constraints to allow or exclude results. Select the String filter type for the Geocode property to filter countries based on their code.
 - c. In order to exclude all countries with a geocode starting with “eu”, follow these steps:
 - i. From the list after Values, choose the option “starts with”.
 - ii. Type “eu” in the input on the right, and click the +Add filter button.
 - iii. Type “ea” in the input on the right, and click the +Add filter button.
 - iv. You must specify how these filters will be linked together. In the top of the Filters dialog, choose the “All filters must be false” option.
 - v. Click “Save & Close” to save the new filters.
6. After adding the necessary filters, run the query again. Only actual countries should now be contained to the result set. Our next target is to calculate the average level of internet access for these countries.
 - a. First of all we have to hide all properties that are not common for the average that we'll calculate. Uncheck the “Show” checkbox of the URI and the geocode. Notice that while we're hiding the geocodes from the results they're not actually removed, so all the filters we set still apply.
 - b. Right click on the “Level of internet access” property and click “Options”.
 - c. From the aggregate functions list select the “Average” function and click OK.
7. Run the query. The result is the average level of internet access for the countries defined in the Eurostat datasource.

Open Source

LinDA Query Designer is completely open-source. You can find the Github repository at: <https://github.com/linda-tools>

Contact

National Technical University of Athens, Decision Support Systems Laboratory
Spiros Mouzakitis (smouzakitis@epu.ntua.gr)
Dimitris Papaspyros (dpap@epu.ntua.gr)