



Workshop #3

Introduction to Wikidata Query Service

Workshop #3



REFRESHER

Workshops #1 & #2

Workshop #3

REFRESHER

Workshop #1 Introduction to Wikidata

A *triple* (also called *statement* in Wikidata) is the basic unit of a knowledge base.

A triple consists of:

Element

Property

Value

Example:

The Nutcracker

was composed by

Tchaikovsky



Workshop #3

REFRESHER

Workshop#2 Contributing to Wikidata

- Creating an item and avoiding duplicates, part 1
- An item and its page, part 1
- Main statement = `instance of (P31)`, part 2
- Properties & values, part 2
- Referencing data, part 2
- Qualifiers, part 2
- Other Wikimedia projects, part 2
- Discussion pages, part 2
- Use [Bistro](#) for general questions, part 4
- Ask specific questions and find answers in [Wikidata projects](#), part 4



Workshop #3



Workshop Scope

Workshop #1: Exploring Wikidata

Workshop #2: Creating a Wikidata item (André Laliberté)

Workshop #3 : How do you use Wikidata? With SparQL queries.

The best ways to use data is by searching and extracting it with SparQL. It's the tool you use to check that the data you added to Wikidata is producing the results you want.

You don't need to know everything about SparQL syntax to add data to Wikidata. However, it can help you understand how others use data and help you see your data input from different points of view.

Workshop #3

Introduction to SparQL

Content

Part 1

- Searching directly in Wikidata
- Wiki page history

Part 2

- Wikidata Query Service (WDQS)
- SparQL language
- Comments, variables, syntax

Part 3

- Extracting the list of "performing artist" subclasses
- Displaying query results as a bubble graph
- Displaying query results on a map graph

End of workshop

Workshop #3



Part 1

Search and History in Wikidata

Workshop #3 Part 1

Searching for Wikidata Entities

Before creating a new item (an entity), **make sure you are not creating a duplicate.**

- With a name in another language
- Under another name
- Under an artist's name (or if you search for an artist's name)

There are several ways to search:

- Wikidata Search Engine
- Wikidata Query Service (WDQS)

Workshop #3 Part 1

Searching in Wikidata: Header Field

Begin the search in the header field of the Wikidata page.

A menu appears when you type in this field. Pay attention to **the last menu item**, the one that will take you to a search page.

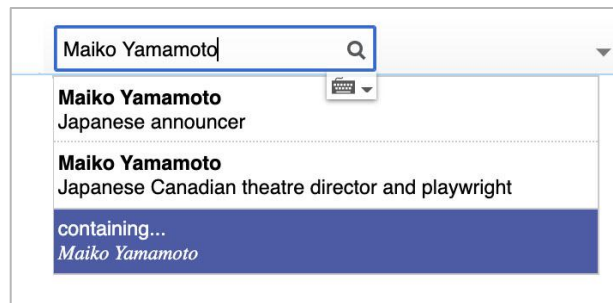
Notice how important a good description is in guiding your choice when items have the same label.



Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

Search Wikidata

Search Wikidata [ctrl-option-f]



Maiko Yamamoto

Maiko Yamamoto
Japanese announcer

Maiko Yamamoto
Japanese Canadian theatre director and playwright

containing...
Maiko Yamamoto

Workshop #3 Part 1

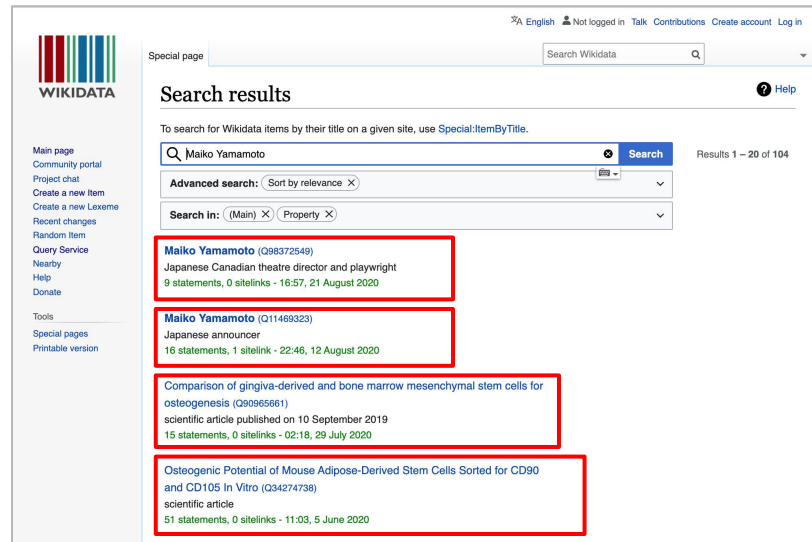
Searching in Wikidata: Results Page

This page displays the results of a search.

Each result has three lines of data:

1. Item label
2. Description
3. Number of statements and date of last revision

Use property and item filters to refine a search.



The screenshot shows the Wikidata search results page for the query 'Maiko Yamamoto'. The page includes a sidebar with navigation links, a search bar, and a list of results. The results are displayed in a table-like format with three main columns: item label, description, and statement count/date. The first result is 'Maiko Yamamoto (Q98372549)', a Japanese Canadian theatre director and playwright, with 9 statements and a last revision date of 16:57, 21 August 2020. The second result is 'Maiko Yamamoto (Q11468323)', a Japanese announcer, with 16 statements, 1 sitelink, and a last revision date of 22:46, 12 August 2020. The third result is 'Comparison of gingiva-derived and bone marrow mesenchymal stem cells for osteogenesis (Q90965681)', a scientific article published on 10 September 2019, with 15 statements and 0 sitelinks, and a last revision date of 02:18, 29 July 2020. The fourth result is 'Osteogenic Potential of Mouse Adipose-Derived Stem Cells Sorted for CD90 and CD105 In Vitro (Q34274738)', a scientific article, with 51 statements, 0 sitelinks, and a last revision date of 11:03, 5 June 2020.

Special page

English Not logged in Talk Contributions Create account Log in

Search Wikidata

Search results

To search for Wikidata items by their title on a given site, use [Special:ItemByTitle](#).

Q Maiko Yamamoto Search

Advanced search: Sort by relevance X

Search in: (Main) X (Property) X

Results 1 – 20 of 104

Maiko Yamamoto (Q98372549)	Japanese Canadian theatre director and playwright	9 statements, 0 sitelinks - 16:57, 21 August 2020
Maiko Yamamoto (Q11468323)	Japanese announcer	16 statements, 1 sitelink - 22:46, 12 August 2020
Comparison of gingiva-derived and bone marrow mesenchymal stem cells for osteogenesis (Q90965681)	scientific article published on 10 September 2019	15 statements, 0 sitelinks - 02:18, 29 July 2020
Osteogenic Potential of Mouse Adipose-Derived Stem Cells Sorted for CD90 and CD105 In Vitro (Q34274738)	scientific article	51 statements, 0 sitelinks - 11:03, 5 June 2020

Workshop #3 Part 1

History of Page Revisions

Every page in a Wiki has a **history**. In Wikidata, every item has a history page.

A page's history lets you see **who** made a revision, **when** it was done, and the **scope** of the revision in bytes.

It is also possible to compare different versions.



Item Discussion Read View history Search Wikidata

Revision history of "Maiko Yamamoto" (Q11469323)

View logs for this item (view abuse log)

Filter revisions

Diff selection: Mark the radio boxes of the revisions to compare and hit enter or the button at the bottom.
Legend: (cur) = difference with latest revision, (prev) = difference with preceding revision, m = minor edit.

Compare selected revisions

- ☒ (cur | prev) 22:46, 12 August 2020 Local (talk | contribs) .. (13,978 bytes) (+1,473) .. (Added reference to claim: height (P2048): 162 centimetre, #quickstatements; #temporary_batch_1597252818044) (undo) (Tag: quickstatements [2.0])
- ☒ (cur | prev) 22:46, 12 August 2020 Local (talk | contribs) .. (12,503 bytes) (+450) .. (Added qualifier: point in time (P585): 2009, #quickstatements; #temporary_batch_1597252818044) (undo) (Tag: quickstatements [2.0]) (restore)
- ☐ (cur | prev) 22:46, 12 August 2020 Local (talk | contribs) .. (12,053 bytes) (+423) .. (Created claim: height (P2048): 162 centimetre, #quickstatements; #temporary_batch_1597252818044) (undo) (Tag: quickstatements [2.0]) (restore)
- ☐ (cur | prev) 08:40, 11 July 2020 MsynBot (talk | contribs) .. (11,630 bytes) (0) .. (Updated item: update reference: move based on heuristic (P887) reference qualifier value to deduced from given name (Q6052498) #msynbotfix2) (undo) (Tag: PAWS [1.2]) (restore)
- ☐ (cur | prev) 10:36, 11 September 2019 LogainmBot (talk | contribs) .. (11,630 bytes) (+79) .. (Updated item: Irish description updated) (undo) (restore)
- ☐ (cur | prev) 21:44, 18 May 2019 Sakretsu (talk | contribs) .. (11,551 bytes) (+408) .. (Created claim: name in native language (P1559): 山本舞衣子, #quickstatements; batch #12993 by User:Sakretsu) (undo) (Tag: QuickStatements [1.3]) (restore)
- ☐ (cur | prev) 00:55, 9 May 2019 MsynBot (talk | contribs) .. (11,143 bytes) (0) .. (Updated item: migrate 1 reference qualifier from imported from Wikimedia project (P143) to stated in (P248) (value: Yahoo! Japan Talent Database (Q27048856); #msynbotfix2) (undo) (restore)
- ☐ (cur | prev) 10:38, 26 February 2019 XabatuBot (talk | contribs) .. (11,143 bytes) (+79) .. (Added [srf] description: Ilucutura xaponena) (undo) (restore)
- ☐ (cur | prev) 10:38, 26 February 2019 XabatuBot (talk | contribs) .. (11,064 bytes) (+75) .. (Added [srf] label: Maiko Yamamoto) (undo) (restore)
- ☐ (cur | prev) 17:19, 5 February 2019 Emijipbot (talk | contribs) .. (10,989 bytes) (+73) .. (Updated item: BOT - Adding labels (1 languages): es) (undo) (restore)
- ☐ (cur | prev) 11:30, 26 October 2018 Ghuron (talk | contribs) .. (10,916 bytes) (+450) .. (Added reference to claim: sex or gender (P21): female (Q681072), #quickstatements; #temporary_batch_1540533295978) (undo) (Tag: QuickStatements [1.1]) (restore)

Workshop #3



End of Part 1

Search for the Stars!

Workshop #3



Part 2

SparQL Query Syntax



Workshop #3 Part 2

SparQL Language

Queries are text-based and follow SparQL syntax.

URL of Wikidata query editor: <https://query.wikidata.org/>

SparQL language is highly structured and admittedly complex and a bit off-putting. But it is VERY powerful. You can create a SparQL query without having to memorize anything.

To learn how to use SparQL, I recommend this tutorial:

[https://www.wikidata.org/wiki/Wikidata:SPARQL tutorial](https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial)

Workshop #3 Part 2

A SparQL Query from an example

I recommend that you always start with an example (on button bar).

That way, you don't need to write your request from scratch!

Service	Examples	Help	
QL query or choose a query example)			

The screenshot shows the Wikidata Query Service interface. At the top, a word cloud displays various Wikidata properties such as 'author (P50)', 'point in time (P585)', 'WikiPathways ID (P2410)', 'female (Q6581072)', 'located in the administrative territorial entity (P131)', 'end time (P582)', 'coordinate location (P625)', 'image (P18)', 'population (P1082)', 'date of death (P570)', 'date of birth (P569)', 'inception (P571)', 'scholarly article (Q13442814)', 'subclass of (P279)', 'human (Q5)', 'country (P17)', 'occupation (P106)', 'sex or gender (P21)', 'position held (P39)', 'award received (P166)', 'child (P40)', 'part of (P361)', 'country of citizenship (P27)', 'item for this sense (P5137)', and 'creator (P170)'. The central text reads 'instance of (P31)'. Below the word cloud is a search bar with the placeholder 'Type to filter' and a button with '329'. Underneath is a section titled 'Simple Queries' containing a list of query examples: 'Cats', 'Goats', and 'Horses (showing some info about them)'. The 'Cats' query is highlighted with a red rectangle. Each query example has a camera icon and an eye icon to its right.

Workshop #3 Part 2

A SparQL Query

Comment

Variable

Service is for
label names
Here, the choice
of interface
language, in
French and if
there are no
results, in English
as the default
choice

The screenshot shows the Wikidata Query Service interface. At the top, there's a header with the Wikidata logo, the text "Wikidata Query Service", and buttons for "Examples", "Help", and "More tools". On the right, there's a language selector set to "English". On the left, there's a sidebar with icons for information, query, pin, diamond, and folder. The main area displays a SPARQL query with line numbers 1 through 9. Annotations with arrows point to specific parts of the query: "Comment" points to the query text, "Variable" points to the variable "?item", and the descriptive text points to the "SERVICE" clause and the language parameter.

```
1 #Cats
2 SELECT ?item ?itemLabel
3 WHERE
4 {
5   ?item wdt:P31 wd:Q146 .
6 }
7 SERVICE wikibase:label
8 { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" . }
9 }
```


Workshop #3 Part 2

Editor of the Wikidata Query Service

You don't need to remember the identifiers of your elements. You can use the auto-completion feature.

To do so, type the name of the element or property after the namespace identifier (« wd: » for the elements or « wdt » for the properties), then press on the keyboard keys [Control-Space].

```
{  
  ?vClasse    wdt:P279*    wd:artiste de scène .  
  OPTIONAL {  
    ?vClasse  rdfs:label  ?vEtiqu  
    FILTER (LANG(?vEtiqu) = "en")  
  }  
}
```

Workshop #3 Part 2

SparQL Language: Basic Syntax

```
SELECT ?vClasse ?vDesc
WHERE
{
  ?vClasse      wdt:P279*      wd:Q28640      .

  SERVICE wikibase:label
  { bd:serviceParam wikibase:language
    "[AUTO_LANGUAGE],fr,en" . }
}
```

SELECT: list of variables you want returned

WHERE: search criteria used to restrict

SERVICE: supplemental tools

GROUP BY: group the results according to one or several variables

ORDER BY: sort order of the results

Workshop #3 Part 2

SparQL Language: Comments

Comments are preceded by a #
(hash mark)

Some comments can change the
display of results

Example: `#defaultView:Map` to
display the results as geographic
coordinates on a map

```
1 #Chats
2 SELECT ?item ?itemLabel
3 WHERE
4 {
5   ?item wdt:P31 wd:Q146 .
6 }
7 SERVICE wikibase:label
8 { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],fr,en" . }
9 }
```

The screenshot shows the Wikidata Query Service interface. The query is as follows:

```
1 #Chats
2 SELECT ?item ?itemLabel
3 WHERE
4 {
5   ?item wdt:P31 wd:Q146 .
6 }
7 SERVICE wikibase:label
8 { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],fr,en" . }
9 }
```

Annotations in the image:

- A blue box highlights `?item` in line 5.
- An orange box highlights `wdt:P31` in line 5.
- A pink oval highlights `wd:Q146` in line 5.

Workshop #3 Part 2

SparQL Language: Variables

Variables are key to extracting data.

- Variables start with ? (question mark): example "?item"
- The name of a variable can be a series of upper and lowercase letters.
- There are no spaces in variable names.



```
1 #Chats
2 SELECT ?item ?itemLabel
3 WHERE
4 {
5   ?item wdt:P31 wd:Q146 .
6 }
7 SERVICE wikibase:label
8   { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],fr,en" . }
9 }
```

Workshop #3 Part 2

SparQL Language: Searching with Triples

In this example, we are searching for all items whose value, for the property "instance of (P31)" is "house cat (Q146)".

Since the variable is in the position of the main element (the subject), this request will extract all the elements in triplets where the property and the value are present.

```
1 #Chats
2 SELECT ?item ?itemLabel
3 WHERE
4 {
5   ?item wdt:P31 wd:Q146 .
6
7   SERVICE wikibase:label
8     { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],fr,en". }
9 }
```

Workshop #3

A short horizontal bar with a teal segment on the left and a magenta segment on the right.

End of Part 2

Workshop #3



Part 3

Viewing Query Results in WDQS

Workshop #3 Part 3

Subclasses of "performing artist"

```
SELECT ?vClass ?vLabel
WHERE
{
  ?vClass wdt:P279* wd:Q713200 .
  OPTIONAL {
    ?vClass rdfs:label ?vLabel
    FILTER (LANG(?vLabel) = "en")
  }
}
```

?vClass = variable containing all the classes that qualify

?vLabel = variable containing the name of the class of the variable **?vClass**

P279 = subclass of

Q713200 = performing artist

Note the period.

<https://w.wiki/bmh>

Workshop #3 Part 3

Subclasses of "performing artist"

Here is the result of the query.

The columns are the variables from the **SELECT** section.

In the results header, you can see
« 185 results in 319 ms ».

You can download a file containing
the results to your computer.

185 résultats en 341ms		</> Code		Télécharger		Lien	
vClasse		vEtiqu					
Q7358		clown					
Q82723		geisha					
Q138858		entertainer					
Q158852		conductor					
Q177718		sideman					
Q214970		virtuoso					
Q215548		jester					
Q215793		kapellmeister					
Q226008		Japanese idol					
Q6901699		monologist					
Q6957294		Nacnī					

Workshop #3 Part 3

WDQS : Bubble Chart Query

- We are searching for all occupations that are **subclasses of "performing artist (Q713200)"**.
- Next, we search for all artists whose occupation is **one of the occupations found** previously.
- Finally, we limit the search to artists who are **Canadian citizens**.

```
SELECT ?vOccup ?vOccupLabel (COUNT(?vOccup) AS ?vDecompte)
WHERE
{
  ?vOccup      wdt:P279      wd:Q713200      .

  ?vArtiste    wdt:P106*     ?vOccup        .

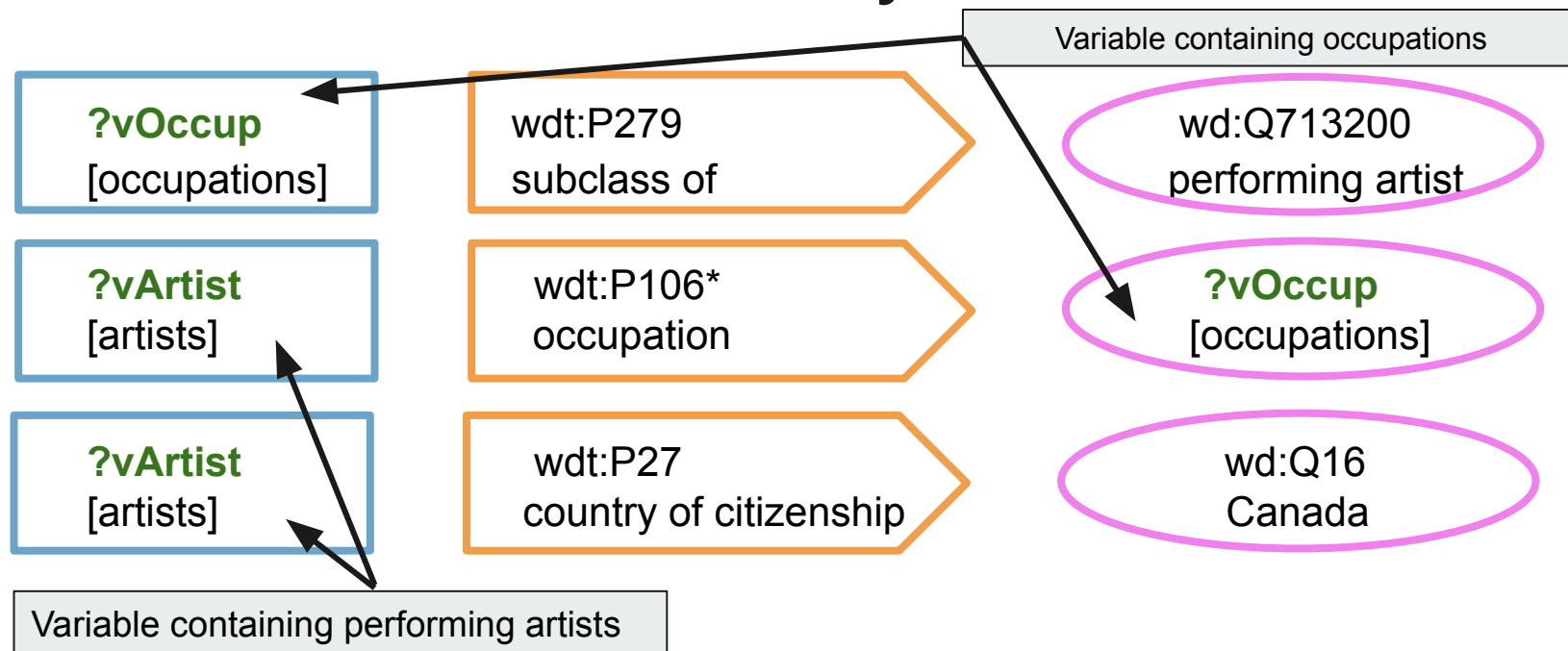
  ?vArtiste    wdt:P27       wd:Q16          .

  SERVICE wikibase:label
  { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],fr,en". }
}
GROUP BY ?vOccup ?vOccupLabel
```

<https://w.wiki/bmf>

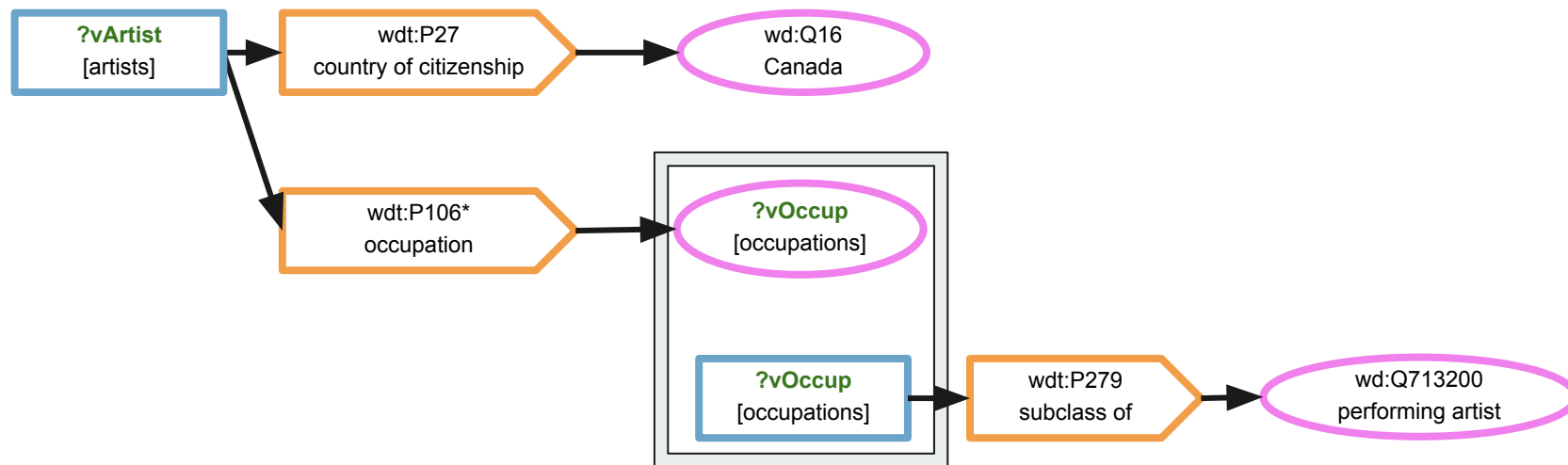
Workshop #3 Part 3

WDQS : BubbleChart Query (cont.)



Workshop #3 Part 3

WDQS : BubbleChart Query (cont.)

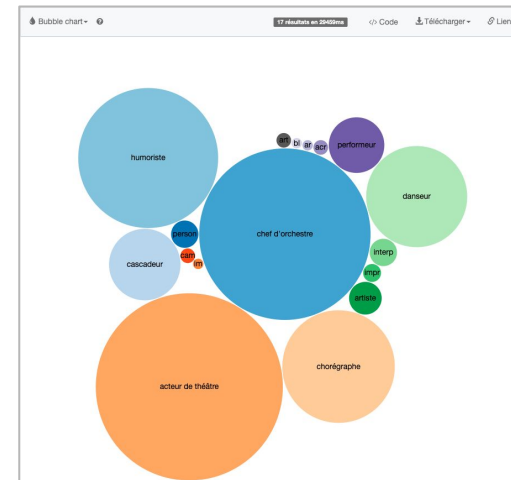
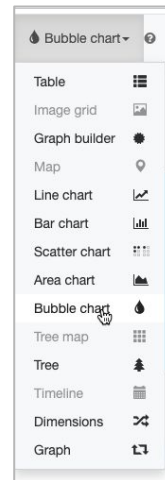


Workshop #3 Part 3

WDQS: Viewing Results as a BubbleChart

To display results as a **Bubble chart**, you need at least two columns of data.

A column with text or entities and a column with numeric values. Each entity is represented as a bubble, and the number sets the size of the bubble.



Workshop #3 Part 3

WDQS: Map View Query

1. Search for items that are theatres (the physical space, not the art form).
2. We extract each theatre's coordinates (with the variable **?vCoord**).
3. We limit the results to Canadian theatres.
4. Note the comment at the end of the query telling it to display the results on a map

```
#Théâtres
SELECT ?vTheatre ?vTheatreLabel ?vCoord
WHERE
{
  ?vTheatre    wdt:P31*    wd:Q24354    .
  ?vTheatre    wdt:P625    ?vCoord      .
  ?vTheatre    wdt:P17     wd:Q16       .

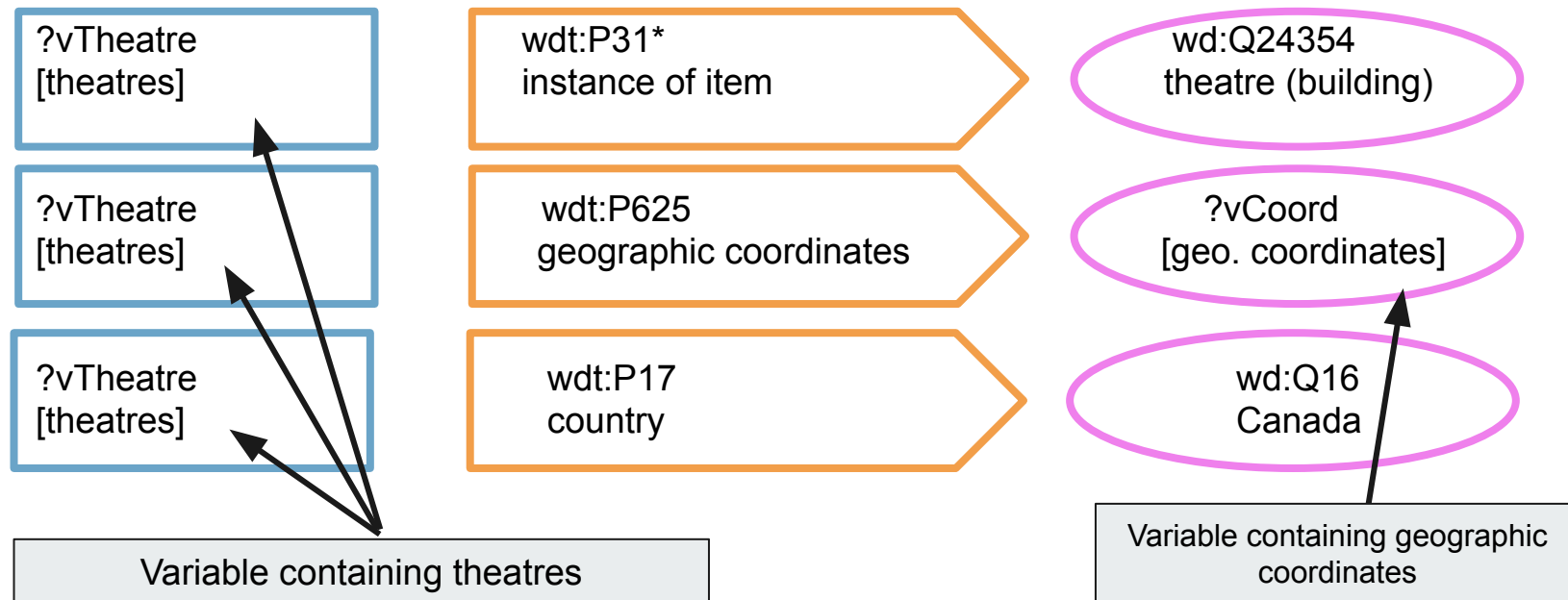
  SERVICE wikibase:label
  { bd:serviceParam wikibase:language ["no_bmgUAGE"],fr,en". }
}

#defaultView:Map
```

<https://w.wiki/bmd>

Workshop #3 Part 3

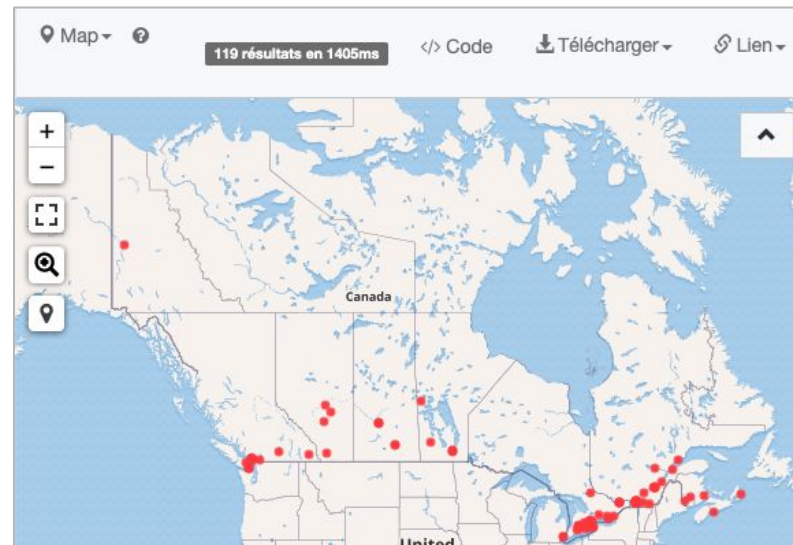
WDQS: Map View Query (cont.)



Workshop #3 Part 3

WDQS: Viewing Results on a Map

- You can scroll the map, but in this example, there are no point outside of Canada.
- You can add a map created with Wikidata's map engine in any Web page, including your own websites.



Workshop #3



End of Part 3

Workshop #3



Recap

What you need to know



Workshop #3

Part 1 : Things to Remember

- Use an internal Wikidata search for searches with the name (label) of an item.
- It is important to avoid recreating items that already exist — duplicates.
Always perform a search before you create a new item.
- A Wiki page's history allows us to know the evolution of a page, and thus to know who made what change and when the revision was made.
- For more complex searches based on an item's statements, use Wikidata Query Service (WDQS).

Workshop #3

Part 2: Things to Remember

- Wikidata Query Services uses SparQL syntax.
- Avoid having to memorize syntax. Use an example as your starting point.
- Basic elements of SparQL syntax: structure, comments, variables.
- A SparQL query consists of triples.



Workshop #3

Part 3: Things to Remember

- You can make very complexe searches with Wikidata Query Service.
- Query results can be displayed as: tables, maps or charts of various types such as bubble charts, line charts, area charts, tree maps, etc.
- With WDQS, you can control:
 - The language of the labels in the results of the query
 - The maximum number of results

Workshop #3

Acknowledgments

Main Project Partners



Financial partners



Conseil des arts du Canada Canada Council for the Arts

Canada 

Québec 



Workshop #3



End of Workshop

Keep on searching!

Workshop #3



Nom de la diapositive