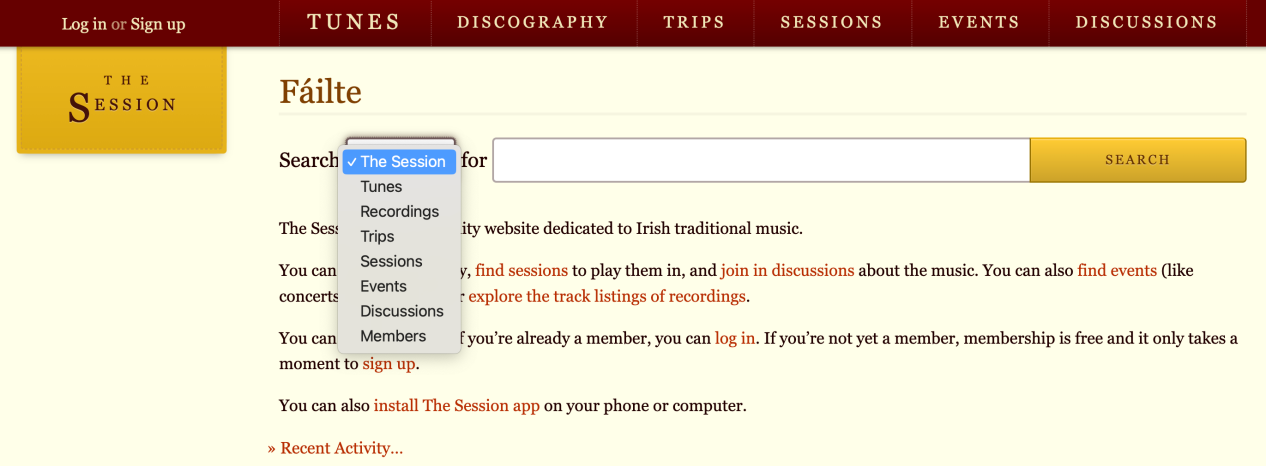
1. **The Session**

The homepage search interface is like:

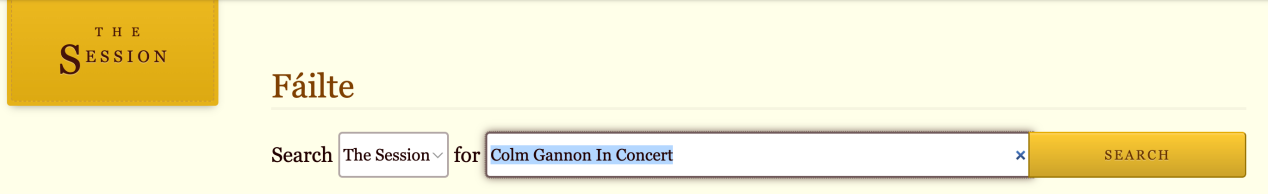


* 1. **Search with the option “The Session”**

1.1.0 for “A confirmar”

It’s wield that you can not find info about the entity session with the option “The Session”...

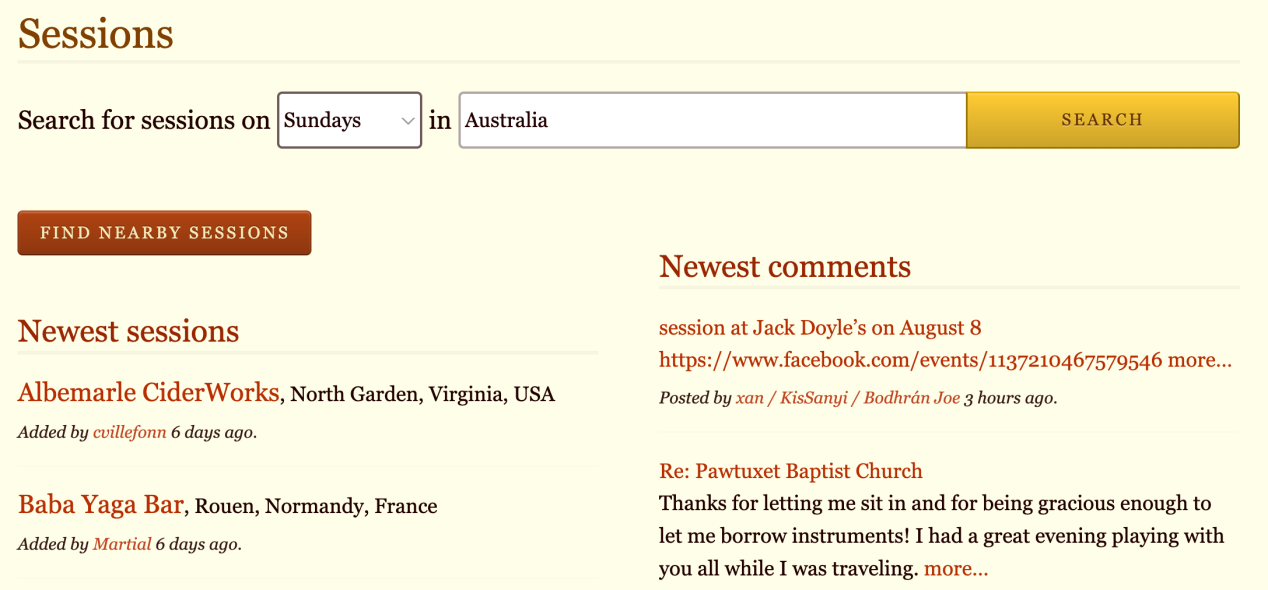
1.1.1 for “Colm Gannon In Concert”



the results are:

We can not find the event whose name is “Colm Gannon In Concert”. But we tried click the Tune “Colm’s” (Colm:科鲁姆)...

* 1. **Search with the option “Sessions”** 
     1. Webpage Query



the result is(https://thesession.org/sessions/search?day=Sunday&q=Australia):



The returned result includes the geographical info, by clicking the blue label as above, it shows:

* + 1. Using SPARQL against RDF

We haven’t a property to confine whether it is Sunday or Monday or ...

```

prefix wdt: <http://www.wikidata.org/prop/direct/>

select \* where{

?session wdt:P17 “Australia”;

wdt:P2561 ?name;

?p ?o

}

```

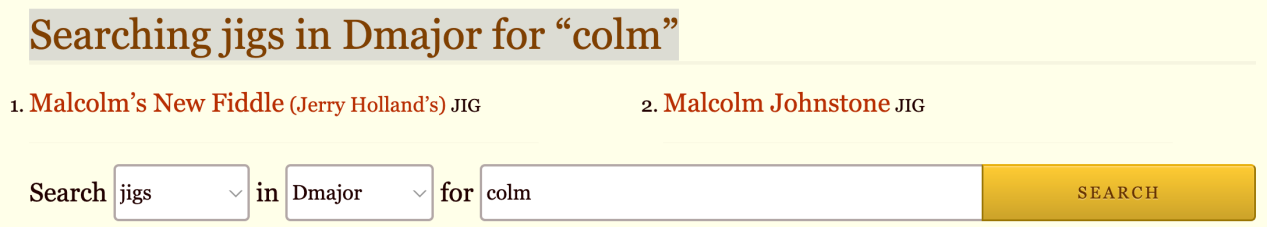
after execution, we get a lot of results. Just take “The last Supper” for example:

|  |  |  |  |
| --- | --- | --- | --- |
| session | name | p | o |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.w3.org/1999/02/22-rdf-syntax-ns#type | http://www.wikidata.org/entity/Q932410 |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P2561 | The Last Supper |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P131 | https://www.wikidata.org/wiki/Q11568 |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P131 | Darwin |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P17 | https://www.wikidata.org/wiki/Q408 |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P17 | Australia |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P276 | https://www.wikidata.org/wiki/Q3235 |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P276 | Northern Territory |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P6375 | 1/35 Cavenagh St |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P625 | Point(-12.46182823 130.84178162) |
| https://thesession.org/sessions/8851 | The Last Supper | http://www.wikidata.org/prop/direct/P585 | 4/14/2024 8:05 |

* 1. **Search with the option “Tunes”**

**-- jigs in Dmajor for “colm”**

* + 1. Webpage Query



* + 1. Using SPARQL against RDF

```

prefix rdf:<>

prefix wd:<http://www.wikidata.org/entity/>

prefix wdt:<http://www.wikidata.org/prop/direct/>

select distinct \*

where {

?Tune rdf:type wd:Q170412;

wdt:P2561 ?nameOfTune;

wdt:P2308 "jig";

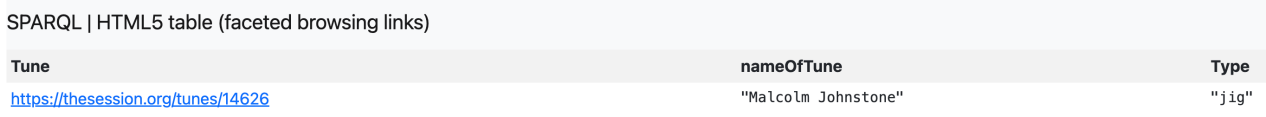
wdt:P2308 ?Type;

wdt:P826 "D major".

filter(contains(str(?nameOfTune), "colm")).

}

```

The result is like:

* 1. **Search other entities or properties which can not be directly serached on the webpage**
     1. Explore the structure of the database

First, we can explore the **chains consisting of classes and properties** by executing:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?classOfDomain ?property ?classOfRange

FROM <the named graph>

WHERE {

{?subject a ?classOfDomain.

?subject ?property ?object1 .

FILTER (?property != rdf:type)

}

union

{?subject a ?classOfDomain.

?subject ?property ?object1 .

?object1 a ?classOfRange.}

}

```

(Please refer to <https://github.com/DDMAL/linkedmusic-datalake/issues/88>)

A long chain is like:

(value of type/genre)<-[type/genre, wdt:P136]-(tune\_set, wd:Q36161)-[corresponds to, <https://thesession.org/tunes>]->(melody/tune, wd:Q170412)<-[corresponds to, <https://thesession.org/tunes>]-(discography, Q273057)-[author, wdt:P50]->(value of author)

* + 1. Find those entities or properties which are not distinct on the webpage

Such as entity “sets” and...

For example, after execution, we can see an entity classified as “set”(wd:Q36161, though this category is questionable). and its relationship with tunes:

(wd:Q36161, set)-[https://thesession.org/tunes]->(wd:Q170412, melody).

We can also see what attributes does set have, such as name(wdt:P2561), type(wdt:P136), meter(wdt:P3440). Above all if you want to see the related tunes, type, meter of a tune\_set whose name is “12 Weeks And A Day”, you can query using SPARQL:

```

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

prefix wd: <http://www.wikidata.org/entity/>

prefix wdt: <http://www.wikidata.org/prop/direct/>

select \* where{

?entity rdf:type wd:Q36161; # The assertion of its type/class is somtimes very important, especially when the properties of this entity are commonly used such as wdt:P2561.

wdt:P2561 ”12 Weeks And A Day”;

<https://thesession.org/tunes> ?tune;

wdt:P136 ?type;

wdt:P2308 ?type\_wiki;

wdt:P3440 ?meter

}

```

* 1. **Search the chains of entity1->property1->entity2->propery2..., which can not be directly serached on the webpage**

Regarding the forementioned chain, we may query with a “chain of quesiton”:

*Please find some tune\_set with type/genre “polka”. The tune\_set should have tunes which Monks Of The Screw Trio’s discography(原本表示唱片，须重新考虑reconciliation) also has.*

The corresponding SPARQL can be:

```

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

prefix wd: <http://www.wikidata.org/entity/>

prefix wdt: <http://www.wikidata.org/prop/direct/>

select \* where{

?tune\_set rdf:type wd:Q36161;# We possibly don’t need this triples...

wdt:P136 “polka”;

<https://thesession.org/tunes> ?tune.

?discography rdf:type wd:Q273057;

wdt:P50 “Monks Of The Screw Trio”;

<https://thesession.org/tunes> ?tune.

}

```

The query result is like:

|  |  |  |
| --- | --- | --- |
| tune\_set | tune | discography |
| https://thesession.org/members/101858/sets/52599 | https://thesession.org/tunes/1650 | https://thesession.org/recordings/1001 |
| https://thesession.org/members/101941/sets/22891 | https://thesession.org/tunes/17936 | https://thesession.org/recordings/1001 |
| **https://thesession.org/members/104992/sets/12112** | **https://thesession.org/tunes/259** | **https://thesession.org/recordings/1001** |
| https://thesession.org/members/105138/sets/3268 | https://thesession.org/tunes/259 | https://thesession.org/recordings/1001 |

...

1. **MusicBrainz**

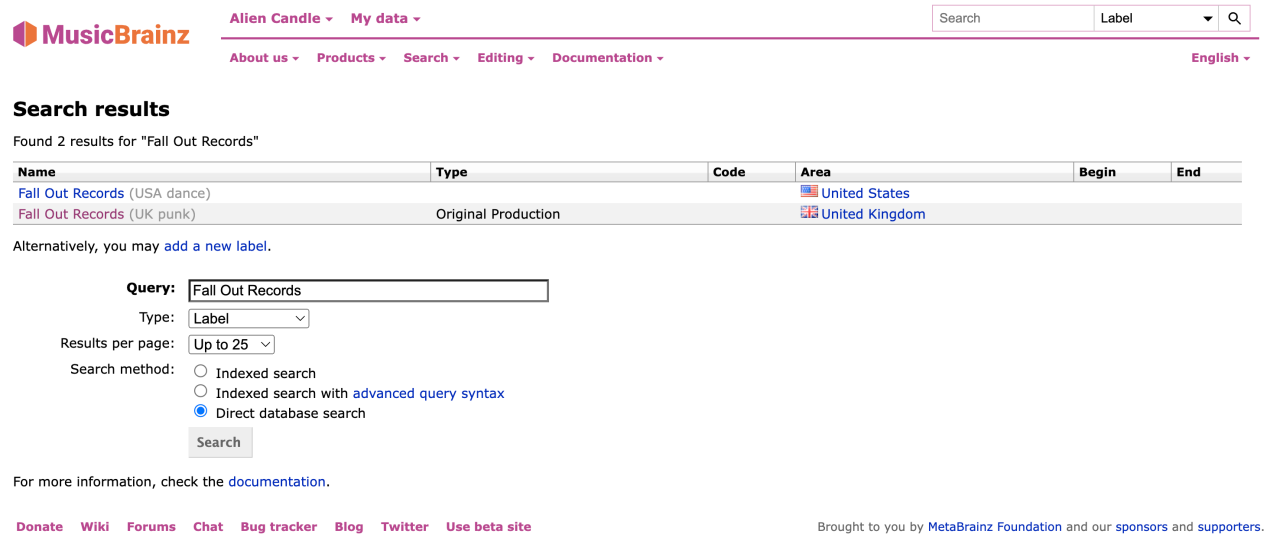
Navigation: <https://musicbrainz.org/> > <https://musicbrainz.org/search>

* 1. **Search a specific entity**

Eg: Search a Label whose name is “Fall Out Records”

* + 1. Webpage Query

<https://musicbrainz.org/search?query=Fall+Out+Records&type=label&limit=25&method=direct>



* + 1. Using SPARQL against RDF

```

prefix rdf:<>

select \* where {

?entity rdf:type <>.

#feed back the properties shown in the table as above

}

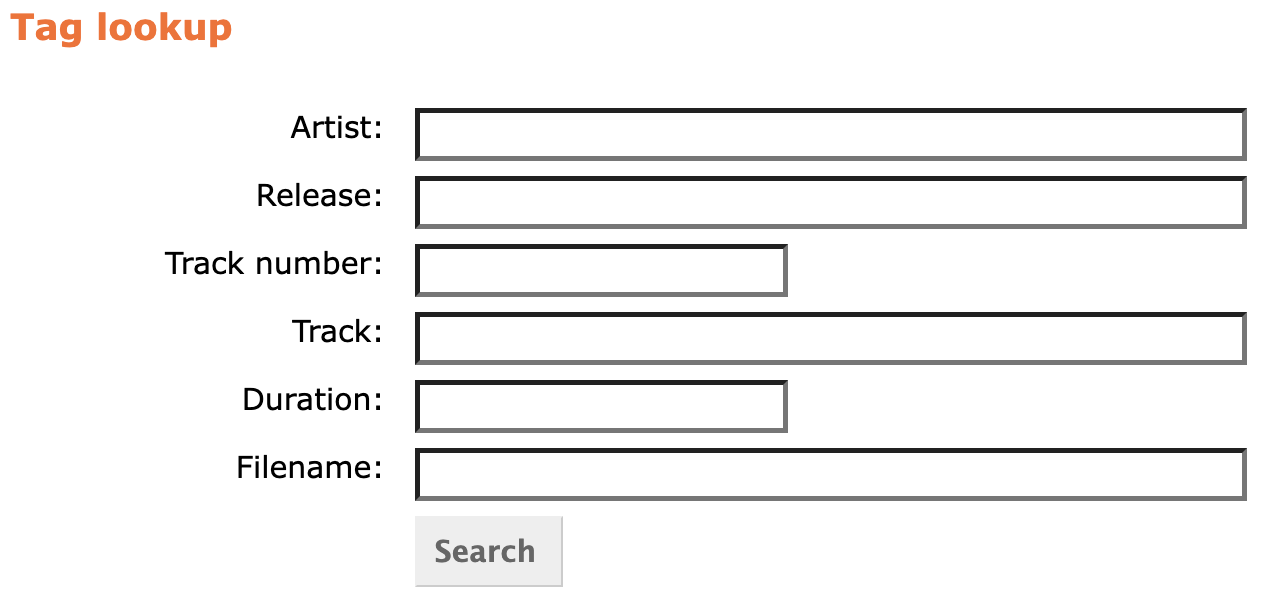
```

* 1. **Please find the Record Label whose genre is punk and whose area is United States.**
  2. **Search the schema of MusicBrainz**

2.1.1 Webpage Query

2.1.2 ...

* 1. **Tag lookup**



* 1. **Statistic Analysis**
  2. **Query with WikiData**
     1. Supplement the data