# retrieve 100 triples

SELECT ?s ?p ?o

WHERE {?s ?p ?o}

LIMIT 100

or instead can do

SELECT \*

WHERE {?s ?p ?o}

LIMIT 100

# Retrieve 100 distinct predicate

SELECT distinct ?p

WHERE {?s ?p ?o}

LIMIT 100

# Retrieve 100 people

SELECT distinct ?x

WHERE {?x a dbo:Person}

LIMIT 100

# Retrieve predicates that have a resource of type person as subject

SELECT distinct ?s

WHERE {?s ?p ?o.

?s rdf:type dbo:Person}

LIMIT 100

rivedere

# Retrieve the birth names and years of 100 people

SELECT \*

WHERE {?x rdf:type dbo:Person;

dbo:birthYear ?y;

dbo:birthName ?name}

LIMIT 100

# Retrive the birth names of 100 people born in 1984 ordered by name

SELECT \*

WHERE {?x rdf:type dbo:Person;

dbo:birthYear "1984"^^ <http://www.w3.org/2001/XMLSchema#gYear> ;

dbo:birthName ?name}

ORDER BY ?name

LIMIT 100

# Length of nile river

? recuperare

# 9. Retrieve trumpet players that were also bandleaders.

SELECT DISTINCT ?x

WHERE { ?x dbo:occupation dbr:Bandleader;

dbo:instrument dbo:Trumpet

}

LIMIT 100

rivedere

# Retrieve movies that were written, directed and produced by the same person

SELECT DISTINCT ?x ?film

WHERE { ?film rdf:type dbo:Film.

?film dbo:director ?x;

dbo:writer ?x;

dbo:producer ?x

}

LIMIT 100

# 3. Retrieve names of writers and, if available, their birth year.

SELECT DISTINCT ?name ?year

WHERE {

?x a dbo:Writer;

rdfs:label ?name

OPTIONAL {?x dbo:birthYear ?year}}

LIMIT 100

# 3. The same as the previous query but, if available, retrieve their birth place

SELECT DISTINCT ?name ?year ?place

WHERE {

?x a dbo:Writer;

rdfs:label ?name

OPTIONAL {?x dbo:birthYear ?year; dbo:birthPlace ?place}}

LIMIT 100

SELECT DISTINCT ?name ?year ?place

WHERE {

?x a dbo:Writer;

rdfs:label ?name

OPTIONAL {?x dbo:birthYear ?year }

OPTIONAL {?x dbo:birthPlace ?place}

}

LIMIT 100

the second is more correct because the first one ask for place and year at the same time, instead there can be someone that we know the birthplace but not the birthyear

# 3. Retrieve movies that were written, directed and produced by the same person. If available, retrieve their box-office gross

SELECT DISTINCT \*

WHERE { ?fil a dbo:Film; dbo:director ?x; dbo:writer ?x; dbo:producer ?x

OPTIONAL {?film dbo:gross ?gross }

}

LIMIT 100

# Retrieve people that were born in Milan or in Rome

SELECT DISTINCT \*

WHERE { ?x a dbo:Person {?x dbo:birthPlace dbr:Milan} union {?y dbo:birthPlace dbr:Rome}}

LIMIT 100

rivedere

# 3. Same as the previous query, but also retrieve the occupation for people born in Milan.

SELECT DISTINCT \*

WHERE { ?x a dbo:Person {?x dbo:birthPlace dbr:Milan; dbo:occupation ?z} union {?x dbo:birthPlace dbr:Rome}}

LIMIT 1000

# 4. Same as the previous query, but retrieve the occupation for people born in Milan only if available.

SELECT DISTINCT \*

WHERE { ?x a dbo:Person {?x dbo:birthPlace dbr:Milan OPTIONAL { ?x dbo:occupation ?z}} union {?x dbo:birthPlace dbr:Rome}}

LIMIT 1000

ex.1

# 1. Was Natalie Portman born in (some city of) the US?

ASK

WHERE { dbr:Natalie\_Portman dbo:birthPlace ?x.

?x dbo:country dbr:United\_States}

Or instead use

ASK

WHERE { dbr:Natalie\_Portman dbo:birthPlace ?x/dbo:country dbr:United\_States}

# 2. Is Rome the capital of Italy?

rivedere

# Describe 20 people born in Italy

describe ?x

where {?x dbo:birthPlace/dbo:country dbr:Italy}

limit 20

# 5. Construct the graph of the ancestors of dbo:Elizabeth\_II

CONSTRUCT {?x dbo:parent ?y}

WHERE {

dbr:Elizabeth\_II dbo:parent+ ?x.

?x dbo:parent ?y

}

# 6. Construct an RDF graph about characters from “A Song of Ice and Fire” (or some other series you like) and their relatives (and/or some other properties).

CONSTRUCT {?x dbo:relative ?y}

WHERE {

?x dbo:series dbr:Breaking\_Bad.

?x dbo:relative ?y.

}