SPARQL Queries for Interface

Search Objective	Pseudo SPARQL Query	SPARQL Query Example

To find the broader term of a category	SELECT variables *a* and *aLabel* and *b* and *bLabel*
	WHERE
	BIND ([category] to variable *a*)
	a hasLabel *aLabel*
	a isSubClassOf *b*
	b hasLabel *bLabel*
		`SELECT ?a ?aLabel ?b ?bLabel`
		
		`WHERE`
		`{`
		`BIND (wd:Q20983127 as ?a)`
		`?a rdfs:label ?aLabel.`
		`?a wdt:P279 ?b.`
		`?b rdfs:label ?bLabel.`
		
		`}`

To find the narrower term of a category	SELECT variable *a* and *aLabel* and *n* and *nLabel*
	WHERE
	BIND ([given item] to variable *a*)
	a hasLabel *aLabel*
	a inverse of isSubClassOf *n*
	n hasLabel *nLabel*
		`SELECT ?a ?aLabel ?n ?nLabel`
		
		`WHERE`
		`{`
		`BIND (wd:Q2453629 as ?a)`
		`?a rdfs:label ?aLabel.`
		`?a ^wdt:P279 ?n.`
		`?n rdfs:label ?nLabel.`
		
		`}`

To find the related Nomen term of a category	SELECT variable *category* and *match*
	WHERE
	BIND ([given item] to variable *category*)
	
	category hasExactMatch to *match*
	
	or
	
	category hasbroadMatch to *match*
	
	or
	
	category hascloseMatch to *match*
		`SELECT ?category ?match`
		
		`WHERE`
		`{`
		`BIND (wd:Q20983127 as ?category)`
		
		`{?category skos:exactMatch ?match . }`
		`UNION`
		`{?category skos:broadMatch ?match . }`
		`UNION`
		`{?category skos:closeMatch ?match .}`
		
		`FILTER regex(str(?match), "nomen", "i")`
		`}`

Search Objective	**Pseudo SPARQL Query**	**SPARQL Query Example**
	FILTER the values of *match* for the string “nomen”	

To find the related Getty AAT term of a category	SELECT variable *category* and *match*
	WHERE
	BIND ([given item] to variable *category*)
	category hasExactMatch to *match*
	or
	category hasbroadMatch to *match*
	or
	category hascloseMatch to *match*
	FILTER the values of *match* for the string “getty”
		SELECT ?category ?match
		WHERE
		{
		BIND (wd:Q20983127 as ?category)
		{?category skos:exactMatch ?match . }
		UNION
		{?category skos:broadMatch ?match . }
		UNION
		{?category skos:closeMatch ?match .}
		FILTER regex(str(?match), "getty", "i")
		}

To find the related Nomen term of a given item	SELECT variable *a* and *category* and *match*
	WHERE
	BIND ([given item] to variable *a*)
	a isInstanceOf *category*
	category hasExactMatch to *match*
	or
	category hasbroadMatch to *match*
	or
	category hascloseMatch to *match*
	FILTER the values of *match* for the string “nomen”
		SELECT ?a ?category ?match
		WHERE
		{
		BIND (wd:Q102971296 as ?a)
		{?a wdt:P31 ?category .
		?category skos:exactMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:broadMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:closeMatch ?match .}
		FILTER regex(str(?match), "nomen", "i")
		}

Search Objective	Pseudo SPARQL Query	SPARQL Query Example

To find the related Getty term of a given item	SELECT variable *a* and *category* and *match*
	WHERE
	BIND ([category] to variable *a*)
	a isInstanceOf *category*
	category hasExactMatch to *match*
	
	or
	
	category hasbroadMatch to *match*
	
	or
	
	category hascloseMatch to *match*
	
	FILTER the values of *match* for the string "getty"
		SELECT ?a ?category ?match
		
		WHERE
		{
		BIND (wd:Q102971296 as ?a)
		
		{?a wdt:P31 ?category .
		?category skos:exactMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:broadMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:closeMatch ?match .}
		
		FILTER regex(str(?match), "getty", "i")
		}

To see all art objects in a certain category	SELECT variable *a* and *aLabel*
	WHERE
	a isInstanceOf [category iri]
	a hasLabel *aLabel*
		SELECT ?a ?aLabel
		
		WHERE
		{
		?a wdt:P31 wd:Q12826066 ;
		rdfs:label ?aLabel .
		}

To see all art objects categorized in a subclass of a higher level.	SELECT variables *a* and *aLabel* and *c* and *cLabel* and *n* and *nLabel*
	WHERE
	BIND ([category iri] to variable *c)*
	c isInverse of isSubClassOf *n*
	c hasLabel *cLabel*
	a isInstanceOf *n*
	a hasLabel *aLabel*
	n hasLabel *nLabel*
		SELECT ?a ?aLabel ?c ?cLabel ?n ?nLabel
		WHERE
		{
		BIND (wd:Q2453629 as ?c)
		?c ^wdt:P279* ?n.
		?c rdfs:label ?cLabel.
		?a wdt:P31 ?n.
		?a rdfs:label ?aLabel.
		?n rdfs:label ?nLabel.
		}

To see an art object's category	SELECT variable *a* and *category* and *categoryLabel*
	WHERE
		SELECT ?a ?category ?categoryLabel
		
		WHERE

Search Objective	Pseudo SPARQL Query	SPARQL Query Example

	BIND ([object iri] to variable *a*)
	a isInstnceOf *category*
	category hasLabel *categoryLabel*
		{
		BIND (wd:Q103309293 as ?a)
		
		?a wdt:P31 ?category .
		?category rdfs:label ?categoryLabel .
		}

To see an art object’s exact match category in other vocabularies	SELECT variable *a* and *category* and *exactMatch*
	WHERE
	BIND ([object iri] to variable *a*)
	a isInstanceOf *category*
	category hasExactMatch *exactMatch*
		SELECT ?a ?category ?exactMatch
		
		WHERE
		{
		BIND (wd:Q103309293 as ?a)
		
		?a wdt:P31 ?category .
		?category skos:exactMatch ?exactMatch .
		
		}

To see an art object’s category and various matching relationships to the other vocabularies.	SELECT variable *a* and *category* and *match*
	WHERE
	BIND ([object iri] to variable *a*
	
	a isInstanceOf *category*
	category hasExactMatch to *match*
	
	or
	
	category hasbroadMatch to *match*
	
	or
	
	category hascloseMatch to *match*
		SELECT ?a ?category ?match
		
		WHERE
		{
		BIND (wd:Q102971296 as ?a)
		
		{?a wdt:P31 ?category .
		?category skos:exactMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:broadMatch ?match . }
		UNION
		{?a wdt:P31 ?category .
		?category skos:closeMatch ?match .}
		}

To find art objects with string value in Label	SELECT variables *a* and *aLabel* and *c*
	FROM work-class-link graph
	WHERE
	a hasLabel *aLabel*
	a isInstanceOf *c*
		SELECT ?a ?aLabel ?c
		
		FROM <http://work-class-links>
		
		WHERE
		{
		?a rdfs:label ?aLabel .

Search Objective	Pseudo SPARQL Query	SPARQL Query Example
	FILTER *aLabel* values for [insert search value]	?a wdt:P31 ?c . FILTER regex(str(?aLabel)," cup ","i") } LIMIT 100
To see a category, its broader term, and its narrower term(s) in Getty	SELECT variables *term* and *termLabel* and *parentLabel* and *childLabel* WHERE BIND ([aat term] as *term*) *term* hasbroaderPreferred *parent* *term* hasInversebroaderPreferred *child* *term* haspreferredLabel *t* *t* hasliteralvalue *termLabel* *parent* haspreferredLabel *l* *l* hasliteralValue *parentLabel* *child* haspreferredLabel *c* *c* hasliteralValue *childLabel*	SELECT ?term ?termLabel ?parentLabel ?childLabel WHERE { bind (aat:300435539 as ?term) ?term gvp:broaderPreferred ?parent ; ^gvp:broaderPreferred ?child. ?term gvp:prefLabelGVP ?t. ?t skosxl:literalForm ?termLabel. ?parent gvp:prefLabelGVP ?l. ?l skosxl:literalForm ?parentLabel. ?child gvp:prefLabelGVP ?c. ?c skosxl:literalForm ?childLabel. }

To see a category, its broader term, and its narrower term(s) in Nomenclature	SELECT variables *term* and *termLabel* and *parentLabel* and *childLabel*
	WHERE
	BIND ([nom term] as *term*)
	
	term hasbroader *parent*
	term hasnarrower *child*
	term haspreferredLabel *termLabel*
	parent haspreferredLabel *parentLabel*
	child haspreferredLabel *childLabel*
		PREFIX nom: <https://nomenclature.info/nom/>
		
		SELECT ?term ?termLabel ?parentLabel ?childLabel
		
		WHERE { bind (nom:11153 as ?term)
		
		?term skos:broader ?parent ;
		skos:narrower ?child.
		
		?term skos:prefLabel ?termLabel.
		?parent skos:prefLabel ?parentLabel.
		?child skos:prefLabel ?childLabel.
		
		filter(lang(?termLabel)="en")
		filter(lang(?parentLabel)="en")
		filter(lang(?childLabel)="en")
		}

Search Objective	Pseudo SPARQL Query	SPARQL Query Example
To see a category's hierarchy placement (**NOTE**: I have an idea on doing this for the AAT using the parent string. Not sure how best to do this with the other vocabs).	(Not added as this query still needs to be	SELECT ?hierarchy WHERE { BIND (wd:Q2453629 as ?term) {?term wdt:P279 ?parent.} UNION {?term ^wdt:P279 ?child.} ?parent rdfs:label ?parentLabel. ?child rdfs:label ?childLabel. ?term rdfs:label ?termLabel. BIND (str(concat("parent: ", ?parentLabel, "; ","term: ", ?termLabel,"; ", "child: ",?childLabel, ".")) as ?hierarchy) } LIMIT 100

Other thoughts and musings:

Is there a way to create a query that results in suggested terms? Maybe there is a way to create a synonym ring or associative terms that could be used. I don't know how the search interface can accommodate that though.

For example, maybe we could use the dataset from WordNet https://wordnet.princeton.edu/download to match words a user enters that are semantically similar to the vocabulary terms.

Collection of SPARQL Queries (Not already captured)

Description	Query

Sculptures with images in Vanderbilt Wikidata Collection
[Wikidata Endpoint]
	#defaultView:ImageGrid
	select ?art ?artLabel ?pic
	
	where {
	 ?art wdt:P195 wd:Q18563658 .
	  ?art wdt:P31 wd:Q860861 .
	  ?art wdt:P18 ?pic .
	  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".}
	
	}

Sculptures with and without images in Vanderbilt Wikidata Collection
[Wikidata Endpoint]
	#defaultView:ImageGrid
	select ?art ?artLabel ?pic
	
	where {
	 ?art wdt:P195 wd:Q18563658 .
	  ?art wdt:P31 wd:Q860861 .
	  optional {?art wdt:P18 ?pic .}
	  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".}
	
	}

Artists with/without pictures in the Vanderbilt Wikidata Collection
[Wikidata Endpoint]
	#defaultView:ImageGrid
	select ?artist ?artistLabel ?pic
	
	where {
	 ?artist wdt:P6379 wd:Q18563658
	  optional {?artist wdt:P18 ?pic .}
	  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".}
	
	}

Artists in Vanderbilt Wikidata collection with map location of birthplace [Wikidata Endpoint]	#defaultView:Map select ?artist ?artistLabel ?placeLabel ?geo

Description	Query

	where {
	 ?artist wdt:P6379 wd:Q18563658.
	 ?artist wdt:P19 ?place.
	  ?place wdt:P625 ?geo
	
	  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".}
	
	}

Artwork URI, Label, Class URI, and Class Label
[Vanderbilt Endpoint]
	SELECT DISTINCT ?work ?workLabel ?class ?classLabel
	WHERE {
	  ?work wdt:P31 ?class.
	  ?work rdfs:label ?workLabel.
	  ?class rdfs:label ?classLabel.
	}
	limit 10

Construction query to build triple statements for wikidata classes to their parent class
[Vanderbilt Endpoint]
	construct {
	  ?class wdt:P279 ?superclass.
	  ?base_class rdfs:label ?base_label.
	  ?superclass rdfs:label ?super_label.
	  }
	where {
	  # Q102971873 is "Soba-choko (noodle sauce cup) with a design in blue underglaze of a stylized rock and grasses"
	  bind (wd:Q102971873 as ?artwork) # Comment out this line to do all artworks
	  ?artwork wdt:P195 wd:Q18563658. # must be in the Vanderbilt Art Gallery
	  ?artwork wdt:P31 ?base_class. # artwork is an instance of the base class
	  ?base_class wdt:P279* ?class. # the subject class is 0 to many subclass_of links from the base class
	  ?class wdt:P279 ?superclass. # the class must have a superclass
	
	  ?base_class rdfs:label ?base_label.
	  filter(lang(?base_label)="en")
	  ?superclass rdfs:label ?super_label.
	  filter(lang(?super_label)="en")
	  }

List of named graphs	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

Description	Query

[Vanderbilt Endpoint}	PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
	PREFIX skosxl: <http://www.w3.org/2008/05/skos-xl#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
	PREFIX dc: <http://purl.org/dc/elements/1.1/>
	PREFIX dcterms: <http://purl.org/dc/terms/>
	PREFIX dcat: <http://www.w3.org/ns/dcat#>
	PREFIX wd: <http://www.wikidata.org/entity/>
	PREFIX wdt: <http://www.wikidata.org/prop/direct/>
	PREFIX aat: <http://vocab.getty.edu/aat/>
	PREFIX gvp: <http://vocab.getty.edu/ontology#>
	PREFIX sd: <http://www.w3.org/ns/sparql-service-description#>
	
	SELECT DISTINCT ?NamedGraph ?modified ?issued ?publisher ?status
	FROM <https://sparql.vanderbilt.edu/graphs>
	WHERE {
	?Service sd:endpoint <https://sparql.vanderbilt.edu/sparql>.
	?Service sd:availableGraphs ?GraphCollection.
	?GraphCollection sd:namedGraph ?NamedGraph.
	?NamedGraph dcterms:modified ?modified.
	optional {?NamedGraph dcterms:issued ?issued.}
	optional {?NamedGraph dc:publisher ?publisher.}
	optional {?NamedGraph
	<http://rs.tdwg.org/dwc/terms/attributes/status> ?status.}
	}
	order by desc(?issued)

Construction query to link Wikidata item to their classes
[Vanderbilt Endpoint]
	construct {
	  ?artwork wdt:P31 ?class.
	  ?artwork rdfs:label ?label.
	  }
	where {
	  ?artwork wdt:P195 wd:Q18563658.
	  ?artwork wdt:P31 ?class.
	  ?artwork rdfs:label ?label.
	  filter(lang(?label)="en")
	  }

Query of AAT to see various broader classes	select distinct ?broader_subject ?subject_label ?broader_object
	?object_label
	from <http://AATOut_2Terms>
	from <http://AATOut_HierarchicalRels>
	where {
	
	# http://vocab.getty.edu/aat/300043022 is the IRI for "teapots"

Description	Query
	bind(<http://vocab.getty.edu/aat/300193015> as ?broader_subject) ?broader_subject gvp:broaderGeneric ?broader_object. #?broader_subject gvp:broaderPreferred ?broader_object. #?broader_subject gvp:broaderNonPreferred ?broader_object. ?broader_subject skosxl:prefLabel ?labelSubject. ?labelSubject skosxl:literalForm ?subject_label. filter(lang(?subject_label)="en") ?broader_object skosxl:prefLabel ?labelObject. ?labelObject skosxl:literalForm ?object_label. filter(lang(?object_label)="en") }

Query to find equivalent concepts to Getty AAT classes	prefix skos: <http://www.w3.org/2004/02/skos/core#>
	prefix skosxl: <http://www.w3.org/2008/05/skos-xl#>
	
	select distinct ?iri ?prefLabel ?otherConcept
	
	where {
	
	# Use bind to avoid having to enter the ID twice
	bind(<http://vocab.getty.edu/aat/300047090> as ?iri)
	
	?iri skosxl:prefLabel ?labelObject.
	?labelObject skosxl:literalForm ?prefLabel.
	filter(lang(?prefLabel)="en")
	
	# Get the equivalent Wikidata concept
	optional {?iri skos:exactMatch ?otherConcept.}
	}

Query to find equivalent concepts to Nomenclature classes	select distinct ?iri ?prefLabel ?otherConcept ?other_label
	
	where {
	
	# Use bind to avoid having to enter the ID twice
	bind(<https://nomenclature.info/nom/13997> as ?iri)
	
	?iri skos:prefLabel ?prefLabel.
	filter(lang(?prefLabel)="en")
	
	# Get the equivalent AAT or Wikidata concept

Description	Query

	optional {?iri skos:exactMatch ?otherConcept.}
	
	# Get the equivalent concept label
	optional {
	 ?otherConcept skosxl:prefLabel ?labelOther.
	 ?labelOther skosxl:literalForm ?other_label.
	filter(lang(?other_label)="en")
	  }
	}

Query to find subject and object IRIs for Getty AAT hierarchy classes	PREFIX gvp: <http://vocab.getty.edu/ontology#>
	PREFIX skosxl: <http://www.w3.org/2008/05/skos-xl#>
	
	select distinct ?broader_subject ?subject_label ?broader_object ?object_label
	from <http://AATOut_2Terms>
	from <http://AATOut_HierarchicalRels>
	where {
	
	# http://vocab.getty.edu/aat/300043022 is the IRI for "teapots"
	
	<http://vocab.getty.edu/aat/300043022> gvp:broaderGeneric* ?broader_subject.
	?broader_subject gvp:broaderGeneric ?broader_object.
	
	 ?broader_subject skosxl:prefLabel ?labelSubject.
	 ?labelSubject skosxl:literalForm ?subject_label.
	filter(lang(?subject_label)="en")
	
	 ?broader_object skosxl:prefLabel ?labelObject.
	 ?labelObject skosxl:literalForm ?object_label.
	filter(lang(?object_label)="en")
	}

Query to find subject and object IRIs for Nomenclature hierarchy classes	PREFIX skos: <http://www.w3.org/2004/02/skos/core#> select distinct ?broader_subject ?subject_label ?broader_object ?object_label from <http://nomenclature_2022-02-02> where { <https://nomenclature.info/nom/12978> skos:broader* ?broader_subject. ?broader_subject skos:broader ?broader_object.

Description	Query

	?broader_subject skos:prefLabel ?subject_label.
	filter(lang(?subject_label)="en")
	
	?broader_object skos:prefLabel ?object_label.
	filter(lang(?object_label)="en")
	  }

Query that selects nomenclature classes and lists exact matches in Wikidata classes and AAT classes	#selects items in nomenclature and provides preferred label and exact match to other vocabularies. select * where { ?item skos:inScheme <https://nomenclature.info/nom/>; skos:prefLabel ?label; skos:exactMatch ?match. {?match rdfs:label ?matchLabel.} UNION {?match gvp:prefLabelGVP/skosxl:literalForm ?matchLabel .} filter langMatches(lang(?label), "en") } LIMIT 100