

Mock SPARQL Update Queries for Common Changes

| | | | | | | |
|-------------------------|------------------------------|-----------------------------|----------------------------|-----------------------------------|------------------------------|-------------------------------|
| Charter | Project Page | Terminology | References | Working Documents | Deliverables | Meeting Notes |
|-------------------------|------------------------------|-----------------------------|----------------------------|-----------------------------------|------------------------------|-------------------------------|

- [Steps to replace an authoritative label via SPARQL](#)
- [Add a new variant:](#)
- [Delete an LCSH resource](#)
- [Deprecate a resource](#)
- [Resource deprecated, but we have a replacement URI](#)

The following SPARQL Update queries are probably mostly correct, but may not be perfect. They focus on LC's LCSH vocabulary, whose resources are sufficiently complex as to make the query patterns more complex than others.

Steps to replace an authoritative label via SPARQL

Find all the subjects with the matching triple pattern,
and replace the matched triples with the INSERT triple pattern.

```
DELETE { ?s madsrdf:authoritativeLabel "Dog catcher" }
INSERT { ?s madsrdf:authoritativeLabel "Dog snatcher" }
WHERE {
  ?s madsrdf:authoritativeLabel "Dog catcher"
}
```

Question:

- Is there an issue using ?s with it matching many entities? Should it be a single URI and update the label for that specific entity?

Add a new variant:

```
INSERT {
  <lcsh:sh123456> madsrdf:hasVariant _:v .
  _:v a madsrdf:Variant .
  _:v madsrdf:variantLabel "Dog catcher" .
}
WHERE {
  <lcsh:sh123456> madsrdf:authoritativeLabel "Dog snatcher"
}
```

Question:

- This one identifies the subject.

Delete an LCSH resource

1) Delete related component list. Thanks to: <https://afs.github.io/rdf-lists-sparql>

```
DELETE {
  ?z rdf:first ?head ; rdf:rest ?tail .
}
WHERE {
  <lcsh:sh123456> madsrdf:componentList ?list .
  ?list rdf:rest* ?z .
  ?z rdf:first ?head ;
    rdf:rest ?tail .
};
```

2) Delete related elements list. Thanks to: <https://afs.github.io/rdf-lists-sparql>

```
DELETE {
    ?z rdf:first ?head ; rdf:rest ?tail .
}
WHERE {
    <lcsh:sh123456> madsrdf:elementList ?list
    ?list rdf:rest* ?z .
    ?z rdf:first ?head ;
        rdf:rest ?tail .
};
```

3) Delete related variant elements list. Thanks to: <https://afs.github.io/rdf-lists-sparql>

```
DELETE {
    ?z rdf:first ?head ; rdf:rest ?tail .
}
WHERE {
    <lcsh:sh123456> madsrdf:hasVariant ?v .
    ?v madsrdf:elementList ?list
    ?list rdf:rest* ?z .
    ?z rdf:first ?head ;
        rdf:rest ?tail .
};
```

4) Delete related resources identified with blank nodes.

```
DELETE {
    ?o ?p1 ?o1 .
    ?o1 ?p2 ?o2 .
}
WHERE {
    <lcsh:sh123456> ?p ?o .
    FILTER( isBlank(?o) ) .
    ?o ?p1 ?o1 .
    OPTIONAL {
        FILTER( isBlank(?o1) ) .
        ?o1 ?p2 ?o2 .
    }
}
```

5) Delete triples that link to deleted resource. NB: If ?s is a blank node, it may be because this Thing is a member of a list. That's just a plain problem.

```
DELETE {
    ?s ?p <lcsh:sh123456> .
}
WHERE {
    ?s ?p <lcsh:sh123456> .
    FILTER( isIRI(?s) ) .
}
```

6) Delete triples where the subject is the resource's URI.

```
DELETE {
    <lcsh:sh123456> ?p ?o .
}
WHERE {
    <lcsh:sh123456> ?p ?o .
}
```

Deprecate a resource

By rights, there are so many changes that it would probably be better, in reality, to delete the resource and replace it.

1) Removes the Authority type and replaces it with Deprecated.

```
DELETE {
  <lcsh:sh123456> a madsrdf:Authority .
}
INSERT {
  <lcsh:sh123456> a madsrdf:DeprecatedAuthority .
}
WHERE {
  <lcsh:sh123456> a madsrdf:Authority .
}
```

2) Removes the authLabel and replaces with depLabel.

```
DELETE {
  <lcsh:sh123456> madsrdf:authoritativeLabel ?label .
}
INSERT {
  <lcsh:sh123456> madsrdf:deprecatedLabel ?label .
}
WHERE {
  <lcsh:sh123456> madsrdf:authoritativeLabel ?label .
}
```

Resource deprecated, but we have a replacement URI

1) Find where the old URI is used, replace it with the new one. Not sure this can be done in one step; might need two steps. The first to add the new triple, the second to delete the old one.

```
INSERT {
  ?s ?p <lcsh:new>
}
DELETE {
  ?s ?p <lcsh:old>
}
WHERE {
  ?s ?p <lcsh:old>
}
```

If "Deprecate a resource" above would better involve deleting the existing resource first, it would be best to replace the old URI with the new one in relationships first.