

SPARQL Queries for Data Management



This page contains information from older versions of VIVO.

People

There are cases when a person or an individual VIVO entity cannot be deleted completely from VIVO

Possible reasons:

The individual may have statements that do not belong to the assertions graph.

To find out, do the following:

Select 'edit this individual' (admin action) near the top of the profile page, and then select 'raw statements with this resource as subject'.

And here is an example of statements that are found in the asserted-tbox and scratchpad graphs.

<<http://vitro.mannlib.cornell.edu/default/asserted-tbox>>

<<http://vitro.mannlib.cornell.edu/default/vitro-kb-inf-scratchpad>>

Solution 1 to remove scratchpad:

1. Go to Site Admin -> Ingest Tools -> Manage Jena Models
2. Look for <http://vitro.mannlib.cornell.edu/default/vitro-kb-inf-scratchpad> model and then click the "Remove" button.

Solution 2 to remove from asserted-tbox:

1. First, BACKUP the VIVO database !!
2. As an example, modTime is the data that needs to be remove. Run the following SPARQL query to look for modTime.

```
construct
{
  ?s <http://vitro.mannlib.cornell.edu/ns/vitro/0.7#modTime> ?o .
}
where
{
  graph <http://vitro.mannlib.cornell.edu/default/asserted-tbox>
  {
    ?s <http://vitro.mannlib.cornell.edu/ns/vitro/0.7#modTime> ?o .
  }
}
```

3. Save the returned results as an n-triple file, e.g. modTime_retract.nt.
4. Create a model called tempModTime and then load this modTime_retract.nt file to the model.
5. Subtract this tempModTime from asserted-tbox by going to Site Admin -> Ingest Tools -> Subtract One Model from Another.

Specify the asserted-tbox as the model to be subtracted from, the tempModTime model as the model to subtract, and the asserted-tbox as the model in which difference should be saved. This will remove the modTime statements from the asserted-tbox model.

Summary:

The above solutions should result in the complete removal of the individual from VIVO.

This query returns the uri and ufid and label of all foaf:Persons

```

SELECT ?p ?u ?l
WHERE
{
  ?p ufVivo:ufid ?u .
  ?p rdf:type foaf:Person .
  ?p rdfs:label ?l .
}

```

This query return a count of uri's with moniker a moniker of "OPS" in the class core:FacultyMember

```

SELECT COUNT(?label)
WHERE
{
  ?person rdf:type core:FacultyMember .
  ?person vitro:moniker "OPS" .
  ?person rdfs:label ?label .
}

```

This query return a count of ufid's for entities of supertype Person

```

SELECT COUNT(?ufid)
WHERE
{
  ?person rdf:type foaf:Person .
  ?person ufVivo:ufid ?ufid
}

```

This query returns a count of the label specified

```

SELECT COUNT(?l)
WHERE
{
  ?s rdf:type foaf:Person .
  ?s vitro:moniker "insert moniker here" .
  ?s rdfs:label ?l .
}

```

This query shows positions that are attached to people and are not attached to orgs:

```

SELECT ?posn ?person ?org
WHERE
{
  ?posn rdf:type core:Position .
  ?posn core:positionForPerson ?person
  OPTIONAL { ?posn core:positionInOrganization ?org }
  FILTER (!BOUND(?org))
}

```

This query shows positions that are attached to orgs and are not attached to people:

```

SELECT ?posn ?org ?person
WHERE
{
  GRAPH <http://vitro.mannlib.cornell.edu/default/vitro-kb-2> {
    ?posn rdf:type core:Position .
    OPTIONAL { ?posn core:positionForPerson ?person }
    ?posn core:positionInOrganization ?org
    FILTER (!BOUND(?person))
  }
}

```

Count all people by type

```

SELECT ?type COUNT(?type)
WHERE
{
  ?person rdf:type foaf:Person .
  ?person rdf:type ?type
}
GROUP BY ?type

```

Count core:facultymember

```

SELECT ?person
WHERE
{
  ?person rdf:type core:FacultyMember
}

```

DSR – find Persons ingested by DSR Harvester

```

SELECT COUNT (?person)
WHERE
{
  ?person rdf:type foaf:Person .
  ?person ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?person rdfs:label ?label }
}

```

General - count core:Person

```

SELECT (COUNT(?label) as ?count)
WHERE
{
  ?org rdf:type foaf:Person .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General - count distinct people

```

SELECT (COUNT(DISTINCT ?label) as ?distinct)
WHERE
{
  ?org rdf:type foaf:Person .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General - distinct people

```

SELECT DISTINCT ?label
WHERE
{
  ?org rdf:type foaf:Person .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General – Find core:FacultyMember (no limit)

```

SELECT ?person
WHERE
{
  ?person rdf:type core:FacultyMember
}

```

General – Find hand-entered people

```
SELECT ?person
WHERE
{
  ?person core:educationalTraining ?training
}
```

*General – Find Persons whose labels have odd characters *

```
SELECT DISTINCT ?o ?n
where {
  ?o a foaf:Person .
  ?o rdfs:label ?n .
  filter(regex(?n,"[~!#$%^*=@\\d&]"))
}
```

PeopleSoft- count all people in core:Faculty class from PS run

```
SELECT (COUNT(?faculty) as ?count)
WHERE
{
  ?faculty rdf:type core:Faculty .
  ?faculty ufVivo:harvestedBy "PeopleSoft-Harvester" .
  OPTIONAL { ?faculty rdfs:label ?label }
}
```

PeopleSoft- Identify all people in core:Faculty class from PS run

```
SELECT ?faculty ?label
WHERE
{
  ?faculty rdf:type core:Faculty .
  ?faculty ufVivo:harvestedBy "PeopleSoft-Harvester" .
  OPTIONAL { ?faculty rdfs:label ?label }
}
```

Organizations

This query returns the uri, label and department_ID of all organizations:

```
SELECT ?uri ?label ?deptID
WHERE
{
  ?uri rdf:type foaf:Organization .
  ?uri rdfs:label ?label .
  ?uri ufVivo:deptID ?deptID
}
```

This query returns the uri and department_ID of core:Department entities

```
SELECT ?uf_org ?deptID
WHERE
{
  ?uf_org rdf:type core:Department .
  ?uf_org ufVivo:deptID ?deptID
}
LIMIT 20
```

This query constructs the uri and class UF AcadDepartment for entities with a department ID that are in core:Department

```

CONSTRUCT
{
  ?uf_org rdf:type ufVivo:UFAcademicDepartment
}
WHERE
{
  ?uf_org rdf:type core:Department .
  ?uf_org ufVivo:deptID ?deptID
}

```

This query return the uri label and moniker of all foaf:Organizations. Can be used to spell check label and monikers

```

SELECT ?uri ?label ?moniker
WHERE
{
  ?uri rdf:type foaf:Organization .
  ?uri rdfs:label ?label .
  ?uri vitro:moniker ?moniker
}

```

This query returns a list of academic departments that do not have a department ID (non UF):

```

SELECT ?label ?not_uf_acad_dept
WHERE
{
  ?not_uf_org rdf:type core:AcademicDepartment .
  ?not_uf_org rdfs:label ?label .
  OPTIONAL { ?not_uf_org ufVivo:deptID ?deptID } .
  FILTER (!BOUND)
}

```

This query construct the uri and class ExtAcadDepart for entities without a department ID that are in core:AcadDepartment

```

CONSTRUCT
{
  ?not_uf_org rdf:type ufVivo:ExternalAcademicDepartment
}
WHERE
{
  ?not_uf_org rdf:type core:AcademicDepartment .
  OPTIONAL { ?not_uf_org ufVivo:deptID ?deptID } .
  FILTER (!BOUND)
}

```

DSR – Find all harvested organizations

```

SELECT ?org ?label ?sponsorAwardId
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?org rdfs:label ?label }
  OPTIONAL { ?org core:sponsorAwardId ?sponsorAwardId }
}
ORDER BY ?label ?sponsorAwardId

```

DSR – Find all harvested organizations without a label

```

SELECT ?org ?label
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?org rdfs:label ?label }
  FILTER(!BOUND(?label))
}

```

DSR – Find all harvested orgs with a sponsorAwardId

```

SELECT ?sponsorAwardId ?org
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  ?org core:sponsorAwardId ?sponsorAwardId
}
ORDER BY ?sponsorAwardId ?org

```

DSR – Find duplicate harvested organizations

```

SELECT ?label ?sponsorAwardId count(?org)
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org rdfs:label ?label .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  ?org core:sponsorAwardId ?sponsorAwardId
}
GROUP BY ?label ?sponsorAwardId
HAVING (COUNT(?org) > 1)
ORDER BY ?label ?sponsorAwardId

```

DSR – Find orgs with a sponsorAwardId and (deptId or core:awardsGrant)

```

SELECT DISTINCT ?org ?sponsorAwardId
WHERE
{
  {
    ?org ufVivo:deptID ?deptID .
    ?org core:sponsorAwardId ?sponsorAwardId .
  }
  UNION
  {
    ?org core:administers ?administers .
    ?org core:sponsorAwardId ?sponsorAwardId .
  }
  UNION
  {
    ?org ufVivo:deptID ?deptID .
    ?org core:awardsGrants ?awardedGrant
  }
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  ?org rdf:type foaf:Organization
}

```

DSR – Find orgs with both a core:administers and sponsorAwardId

```

SELECT DISTINCT ?org ?sponsorAwardId
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org core:sponsorAwardId ?sponsorAwardId .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  ?org core:administers ?administers
}

```

DSR – Find orgs with multiple sponsorAwardId

```

SELECT ?org COUNT(?sponsorAwardId)
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org ufVivo:harvestedBy "DSR-Harvester" .
  ?org core:sponsorAwardId ?sponsorAwardId
}
GROUP BY ?org
HAVING (COUNT(?sponsorAwardId) > 1)

```

DSR – Find all harvested entities without a type

```

SELECT ?entity ?type
WHERE
{
  ?entity ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?entity rdf:type ?type }
  FILTER(!BOUND(?type))
}

```

General - distinctly labeled organizations

```

SELECT DISTINCT ?label
WHERE
{
  ?org rdf:type foaf:Organization .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General - Find all organizations with a sponsor award ID

```

SELECT ?org ?label ?sponsorAwardId
WHERE
{
  ?org rdf:type foaf:Organization .
  ?org core:sponsorAwardId ?sponsorAwardId .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General - List of all organizations

```

SELECT ?org ?label
WHERE
{
  ?org rdf:type foaf:Organization .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General – Find all distinct excludeEntity

```

SELECT DISTINCT ?label
WHERE
{
  ?org rdf:type excludeEntity .
  OPTIONAL { ?org rdfs:label ?label }
}

```

General – Find all entities with >1 UFIDs

```

SELECT ?uri
WHERE
{
  ?uri ufVivo:ufid ?ufid .
}
GROUP BY ?uri
HAVING (COUNT(?ufid) > 1)

```

General – Find blank nodes

```

SELECT ?type
WHERE
{
  ?individual rdf:type ?type .
  OPTIONAL {?type ?predicate ?something}
  FILTER (isBlank(?type))
  FILTER (!BOUND(?something))
}

```

PeopleSoft – Find departments missing foaf:Organization type

```

SELECT ?org COUNT(?type)
WHERE
{
  ?org ufVivo:harvestedBy "PeopleSoft-Harvester" .
  ?org rdf:type ?type
}
GROUP BY ?org
HAVING (COUNT(?type) < 1)

```

PeopleSoft – find organizations with duplicate deptIDs

```

SELECT DISTINCT ?uri ?label str(?deptID)
WHERE
{
  ?uri rdf:type foaf:Organization .
  OPTIONAL {?uri rdfs:label ?label }.
  ?uri ufVivo:deptID ?deptID
}

```

Grants

This query return the label, and uri of all grants harvested from the DSR:

```

SELECT ?uri ?label
WHERE
{
  ?uri rdf:type core:Grant .
  ?uri rdfs:label ?label .
  ?uri ufVivo:harvestedBy "DSR-Harvester" .
}

```

DSR – Count all harvested grants

```
SELECT (COUNT(?grant) as ?count)
WHERE
{
  ?grant rdf:type core:Grant .
  ?grant ufVivo:harvestedBy "DSR-Harvester"
}
```

DSR – Count number of grants with no related role

```
SELECT (COUNT(?grant) as ?count)
WHERE
{
  ?grant rdf:type core:Grant .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?grant core:relatedRole ?relatedRole }
  FILTER(!BOUND(?relatedRole))
}
```

DSR – Find all harvested grants

```
SELECT ?grant ?label
WHERE
{
  ?grant rdf:type core:Grant .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?grant rdfs:label ?label }
}
```

DSR – Find grants that have no administeredBy

```
SELECT ?grant
WHERE
{
  ?grant rdf:type core:Grant .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?grant core:administeredBy ?administeredBy }
  FILTER(!BOUND(?administeredBy))
}
```

DSR – Find grants with more than 1 person attached to them

```
SELECT ?grant COUNT(?role)
WHERE
{
  ?grant core:relatedRole ?role .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
}
GROUP BY ?grant
HAVING (COUNT(?role) > 1)
```

DSR – Find grants with more than one PI

```

SELECT ?grant COUNT(?role)
WHERE
{
  ?grant core:relatedRole ?role .
  ?role rdf:type core:PrincipalInvestigatorRole .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
}
GROUP BY ?grant
HAVING (COUNT(?role) > 1)

```

DSR – Find grants with no related role

```

SELECT ?grant
WHERE
{
  ?grant rdf:type core:Grant .
  ?grant ufVivo:harvestedBy "DSR-Harvester" .
  OPTIONAL { ?grant core:relatedRole ?relatedRole }
  FILTER(!BOUND(?relatedRole))
}

```

Publications

PubMed - Identify all publications with JournalXXXX-XXXX issues

```

SELECT ?infores ?label ?hasPublicationVenue
WHERE
{
  ?infores rdf:type core:InformationResource .
  ?infores ufVivo:harvestedBy "PubMed-Harvester" .
  OPTIONAL { ?infores rdfs:label ?label }
  OPTIONAL { ?infores core:hasPublicationVenue ?hasPublicationVenue }
}
ORDER BY ?label ?hasPublicationVenue

```

PubMed – Find all pmids

```

SELECT ?pub ?pmid
WHERE
{
  ?pub bibo:pmid ?pmid .
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
}

```

PubMed – Find all publications whose journals has a publication venue

```

SELECT ?pub
WHERE
{
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
  ?journal core:hasPublicationVenue ?hasPublicationVenue .
}

```

PubMed – Find pubs that have not been assigned a type

```

SELECT ?pub COUNT(?type)
WHERE
{
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
  ?pub rdf:type ?type .
  ?pub bibo:pmid ?pmid
}
GROUP BY ?pub
HAVING (COUNT(?type) < 3)

```

PubMed – Show counts of the different types

```

SELECT ?type COUNT(?type)
WHERE
{
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
  ?pub bibo:pmid ?pmid .
  ?pub rdf:type ?type
}
GROUP BY ?type
ORDER BY ?type

```

PubMed--Count all PubMed Harvested entities

```

SELECT (COUNT(?pub) as ?count)
WHERE
{
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
}

```

PubMed--Identify PubMed Harvested entities

```

SELECT ?pub COUNT(?type)
WHERE
{
  ?pub ufVivo:harvestedBy "PubMed-Harvester" .
  ?pub rdf:type ?type .
}
GROUP BY ?pub

```

PubMed--Identify publications of an individual; and the linked authorships and venue associated with the publications

```

CONSTRUCT
{
  ?publication ?p ?o .
  ?publication vivo:informationResourceInAuthorship ?allAuthShips .
  ?venue vivo:publicationVenueFor ?publication .
  ?allAuthShips ?pp ?oo .
  ?allAuthShips vivo:linkedAuthor ?allAuthors .
  ?allAuthors vivo:authorInAuthorship ?allAuthShips .
}
WHERE
{
  <myAuthorURI> vivo:authorInAuthorship ?authship .
  ?authship vivo:linkedInformationResource ?publication .
  ?publication ?p ?o .
  ?publication vivo:hasPublicationVenue ?venue .
  ?publication vivo:informationResourceInAuthorship ?allAuthShips .
  ?allAuthShips vivo:linkedAuthor ?allAuthors .
  ?allAuthShips ?pp ?oo .
}

```

PubMed--Identify a certain type of publications of an individual; and the linked authorships and venue associated with the publications

```

CONSTRUCT
{
  ?publication ?p ?o .
  ?publication vivo:informationResourceInAuthorship ?allAuthShips .
  ?venue vivo:publicationVenueFor ?publication .
  ?allAuthShips ?pp ?oo .
  ?allAuthShips vivo:linkedAuthor ?allAuthors .
  ?allAuthors vivo:authorInAuthorship ?allAuthShips .
}
WHERE
{
  <myAuthorURI> vivo:authorInAuthorship ?authship .
  ?authship vivo:linkedInformationResource ?publication .
  ?publication ?p ?o .
  ?publication <http://vitro.mannlib.cornell.edu/ns/vitro/0.7#mostSpecificType> <http://vivoweb.org/ontology/core#EditorialArticle> .
  ?publication vivo:hasPublicationVenue ?venue .
  ?publication vivo:informationResourceInAuthorship ?allAuthShips .
  ?allAuthShips vivo:linkedAuthor ?allAuthors .
  ?allAuthShips ?pp ?oo .
}

```

Other

This query constructs RDF for the removal of text in the overview below a certain number of characters included to show Regex function for string length

```

CONSTRUCT
{
  ?person <http://vivoweb.org/ontology/core#overview> ?overview
}
WHERE
{
  ?person rdf:type foaf:Person .
  ?person core:overview ?overview
  FILTER(!REGEX(?overview, ".....+"))
}

```

General - Filter results by regex limit of string length

```

SELECT ?person ?overview
WHERE
{
  ?person rdf:type foaf:Person .
  ?person core:overview ?overview
  FILTER(!REGEX(?overview, ".....+"))
}

```

General – Find the number of triples in vivo

```

SELECT COUNT(?s)
WHERE
{
  ?s ?p ?o
}

```

General - Find entities with >1 rdfs:label

```
SELECT ?uri ?label COUNT(?label)
WHERE
{
  ?uri rdfs:label ?label .
}
GROUP BY ?uri
HAVING (COUNT(?label) > 1)
```