Guide to SPARQL Queries for Measuring Reuse in DITA RDF Knowledge Graphs

Michael Iantosca
Senior Director of Knowledge Platforms and Engineering
Avalara Inc.
Michael.Iantosca@avalara.com

Introduction

This regenerated guide provides a **complete, structured set of SPARQL queries** to measure content reuse, advanced governance analytics, and optimization opportunities in your DITA RDF knowledge graph in GraphDB.

Core Queries

1. **Number of topics reused**
    - **What it is:** Counts how many unique topics are reused anywhere.
    - **Why valuable:** Measures breadth of reuse; how many topics are contributing to efficiency and consistency.
2. **Number of instances of reuse**
    - **What it is:** Counts total reuse links regardless of topic (e.g. topic A reused in 5 places counts as 5).
    - **Why valuable:** Shows the overall reuse load, useful for ROI calculations on reuse strategies.
3. **Reuse ratio**
    - **What it is:** The proportion of topics reused at least once versus total topics.
    - **Why valuable:** High-level indicator of reuse effectiveness across the library.
4. **Conditional usage analysis**
    - **What it is:** Lists topics with conditional attributes and their values.

- **Why valuable:** Helps manage conditional complexity and optimize variant management strategies.

5. **Reuse frequency per topic**
    - **What it is:** Counts how often each topic is reused.
    - **Why valuable:** Identifies top reusable assets and validates modularization efforts.
6. **Topics reused across products**
    - **What it is:** Lists topics reused across multiple products.
    - **Why valuable:** Reveals shared content opportunities for maintenance efficiency and branding consistency.
7. **Sub-topic (fragment) reuse**
    - **What it is:** Counts reuse of fragments (e.g. re-usable paragraphs or tables).
    - **Why valuable:** Shows fine-grained reuse patterns that drive micro-efficiencies in authoring.
8. **Topics reused by number of products**
    - **What it is:** Counts how many different products reuse each topic.
    - **Why valuable:** Highlights mission-critical content used widely, prioritizing maintenance or governance checks.
9. **Reuse by type**
    - **What it is:** Breakdown of reuse by predicate (conref, keyref, xref).
    - **Why valuable:** Reveals which reuse mechanisms are being used most, guiding training and architecture optimization.
10. **Topics never reused**
    - **What it is:** Lists topics that are not reused anywhere.
    - **Why valuable:** Identifies potential candidates for archiving, reauthoring for reuse, or deletion to reduce content bloat.

Advanced Queries

1. **Average reuse per topic**
    - **What it is:** Calculates average number of reuse instances per reused topic.

- **Why valuable:** Gauges the typical reuse density, informing reuse model success or opportunity areas.

2. **Reuse by taxonomy class**
    - **What it is:** Counts reuse instances grouped by taxonomy class (e.g. concept, task, reference).
    - **Why valuable:** Shows which topic types are most reusable, aiding content strategy refinement.
3. **Topics reused across maximum deliverables**
    - **What it is:** Lists topics used in the highest number of deliverables.
    - **Why valuable:** Highlights high-value topics critical for output consistency and maintenance priority.
4. **Topics reused but not updated recently**
    - **What it is:** Lists reused topics with their last modified date.
    - **Why valuable:** Flags potential risks from outdated reused content affecting multiple deliverables.
5. **Reuse hotspots by author**
    - **What it is:** Counts reuse instances grouped by author.
    - **Why valuable:** Identifies authors creating highly reusable content, informing training and recognition.
6. **Reuse chains (nested reuse)**
    - **What it is:** Finds topics that themselves reuse other topics (nested reuse).
    - **Why valuable:** Reveals complex dependencies that can increase risk if not managed carefully.
7. **Reuse orphan detection**
    - **What it is:** Lists reusable topics never actually reused.
    - **Why valuable:** Identifies wasted preparation effort or content that should be promoted for reuse.
8. **Conditional reuse analysis**
    - **What it is:** Counts reuse instances per condition attribute/value pair.
    - **Why valuable:** Analyzes how conditionalization intersects with reuse to manage variant complexity efficiently.

9. **Reuse potential by product coverage**
    - **What it is:** Lists topics used in only one product.
    - **Why valuable:** Identifies potential reusable content that has not yet been leveraged cross-product.
10. **Reuse trends over time**
    - **What it is:** Counts reuse instances per month.
    - **Why valuable:** Shows trends in reuse behavior to evaluate impact of governance initiatives or tool adoption.
11. **High-risk single points of reuse**
    - **What it is:** Lists topics reused across many products (critical single points of failure).
    - **Why valuable:** Helps prioritize testing, review, and update workflows to mitigate risk.
12. **Duplicate content detection**
    - **What it is:** Finds topics with identical content hashes.
    - **Why valuable:** Detects duplicate content candidates for merge and reuse optimization to reduce maintenance load.

Overall Value

These queries enable:

- ✔ **Reuse ROI measurement**
- ✔ **Governance policy enforcement**
- ✔ **Optimization of authoring and maintenance efficiency**
- ✔ **Risk management for widely reused or outdated content**
- ✔ **Strategic insight into reuse trends and content architecture**

Assumptions

- Ontology uses the `dita:` prefix (adjust as needed).
- Reuse predicates include `dita:conref`, `dita:keyref`, `dita:xref`.
- Product taxonomy tagging via `dita:hasProduct`.

Core Queries

1. Number of topics reused

```
SELECT (COUNT(DISTINCT ?reusedTopic) AS ?numTopicsReused)
WHERE {
  ?reuseRelation ?predicate ?reusedTopic .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
```

2. Number of instances of reuse

```
SELECT (COUNT(?reusedTopic) AS ?numReuseInstances)
WHERE {
  ?reuseRelation ?predicate ?reusedTopic .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
```

3. Reuse ratio

Retrieve total topics:

```
SELECT (COUNT(?topic) AS ?totalTopics)
WHERE {
  ?topic a dita:Topic .
}
```

Calculate externally:

```
Reuse Ratio = Number of topics reused / Total number of topics
```

4. Conditional usage analysis

```
SELECT ?topic ?conditionAttr ?conditionValue
WHERE {
  ?topic dita:hasCondition ?condition .
  ?condition dita:conditionAttribute ?conditionAttr ;
             dita:conditionValue ?conditionValue .
}
```

5. Reuse frequency per topic

```
SELECT ?reusedTopic (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?reusedTopic .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?reusedTopic
ORDER BY DESC(?reuseCount)
```

6. Topics reused across products

```
SELECT ?reusedTopic (GROUP_CONCAT(DISTINCT ?product; SEPARATOR=", ") AS ?prod
ucts)
WHERE {
  ?reuseInstance ?predicate ?reusedTopic .
  ?reuseInstance dita:hasProduct ?product .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?reusedTopic
```

7. Sub-topic (fragment) reuse

```
SELECT ?fragment (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?fragment .
  ?fragment a dita:Fragment .
  FILTER (?predicate IN (dita:conref, dita:keyref)) .
}
GROUP BY ?fragment
ORDER BY DESC(?reuseCount)
```

8. Topics reused by number of products

```
SELECT ?reusedTopic (COUNT(DISTINCT ?product) AS ?numProducts)
WHERE {
  ?reuseInstance ?predicate ?reusedTopic .
  ?reuseInstance dita:hasProduct ?product .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?reusedTopic
ORDER BY DESC(?numProducts)
```

9. Reuse by type

```
SELECT ?predicate (COUNT(*) AS ?count)
WHERE {
  ?reuseInstance ?predicate ?reusedTopic .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?predicate
```

10. Topics never reused

```
SELECT ?topic
WHERE {
  ?topic a dita:Topic .
  FILTER NOT EXISTS {
    ?reuseInstance ?predicate ?topic .
    FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
  }
}
```

Advanced Queries

1. Average reuse per topic

```
SELECT (AVG(?reuseCount) AS ?avgReusePerTopic)
WHERE {
  SELECT ?reusedTopic (COUNT(?reuseInstance) AS ?reuseCount)
  WHERE {
    ?reuseInstance ?predicate ?reusedTopic .
    FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
  }
  GROUP BY ?reusedTopic
}
```

2. Reuse by taxonomy class

```
SELECT ?taxonomyClass (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?reusedTopic .
  ?reusedTopic dita:hasTaxonomyClass ?taxonomyClass .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?taxonomyClass
ORDER BY DESC(?reuseCount)
```

3. Topics reused across maximum deliverables

```
SELECT ?topic (COUNT(DISTINCT ?deliverable) AS ?numDeliverables)
WHERE {
  ?deliverable dita:includesTopic ?topic .
}
GROUP BY ?topic
ORDER BY DESC(?numDeliverables)
```

4. Topics reused but not updated recently

```
SELECT ?topic ?lastModifiedDate (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?topic .
  ?topic dita:lastModifiedDate ?lastModifiedDate .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?topic ?lastModifiedDate
ORDER BY ?lastModifiedDate
```

5. Reuse hotspots by author

```
SELECT ?author (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?topic .
```

```
  ?topic dita:author ?author .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?author
ORDER BY DESC(?reuseCount)
```

6. Reuse chains (nested reuse)

```
SELECT ?outerTopic ?innerTopic
WHERE {
  ?reuseInstance1 ?predicate1 ?outerTopic .
  ?outerTopic ?predicate2 ?innerTopic .
  FILTER (?predicate1 IN (dita:conref, dita:keyref, dita:xref)) .
  FILTER (?predicate2 IN (dita:conref, dita:keyref, dita:xref)) .
}
```

7. Reuse orphan detection

```
SELECT ?topic
WHERE {
  ?topic dita:isReusable true .
  FILTER NOT EXISTS {
    ?reuseInstance ?predicate ?topic .
    FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
  }
}
```

8. Conditional reuse analysis

```
SELECT ?conditionAttr ?conditionValue (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance ?predicate ?topic .
  ?topic dita:hasCondition ?condition .
  ?condition dita:conditionAttribute ?conditionAttr ;
             dita:conditionValue ?conditionValue .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?conditionAttr ?conditionValue
ORDER BY DESC(?reuseCount)
```

9. Reuse potential by product coverage

```
SELECT ?topic (COUNT(DISTINCT ?product) AS ?numProducts)
WHERE {
  ?topic dita:hasProduct ?product .
}
GROUP BY ?topic
HAVING (?numProducts = 1)
ORDER BY ?topic
```

10. Reuse trends over time

```
SELECT ?month (COUNT(?reuseInstance) AS ?reuseCount)
WHERE {
  ?reuseInstance dita:reuseDate ?date .
  BIND(SUBSTR(STR(?date), 1, 7) AS ?month)
}
GROUP BY ?month
ORDER BY ?month
```

11. High-risk single points of reuse

```
SELECT ?topic (COUNT(DISTINCT ?product) AS ?numProducts)
WHERE {
  ?reuseInstance ?predicate ?topic .
  ?reuseInstance dita:hasProduct ?product .
  FILTER (?predicate IN (dita:conref, dita:keyref, dita:xref)) .
}
GROUP BY ?topic
ORDER BY DESC(?numProducts)
```

12. Duplicate content detection

```
SELECT ?hash (COUNT(?topic) AS ?numTopics)
WHERE {
  ?topic dita:contentHash ?hash .
}
GROUP BY ?hash
HAVING (?numTopics > 1)
```

Next Steps

- Adapt queries to your namespace and schema.
- Integrate into GraphDB dashboards or reuse reporting pipelines.
- Extend ontology to capture required properties systematically.
