



Abstract: This document provides a comprehensive guide on utilizing **100 SPARQL queries** for managing and optimizing content within a **DITA Knowledge Graph**. It emphasizes the transformative potential of knowledge graphs in handling technical content, which is increasingly recognized as a strategic asset in modern digital ecosystems. The guide outlines the complexities of managing DITA content at scale and introduces the concept of a knowledge graph as a solution that enables dynamic querying and content governance.

100 SPARQL Queries You Can Use with a DITA Knowledge Graph

Plus 50 bonus queries for DITA Graph RAG content retrieval and metrics

Michael Iantosca
Michael.lantosca@avalara.com

100 SPARQL Queries You Can Use with a DITA Knowledge Graph

Introduction

In the modern digital ecosystem, technical content is no longer just documentation, it's a strategic asset. Managing large DITA-based libraries efficiently is essential for ensuring content quality, regulatory compliance, user satisfaction, localization success, and operational cost control. However, managing DITA content at scale, manually or with traditional cCMS (Component Content Management Systems), is extraordinarily complex.

A DITA Knowledge Graph, representing topics, maps, relationships, metadata, and reuse structures semantically, empowers content teams to query, analyze, and govern technical content in ways previously impossible.

Why is a DITA knowledge graph transformative?

- Traditional CCMSs rely on folder structures, manual reports, and local checks.
- Knowledge graphs allow dynamic, federated, real-time querying across millions of topics.
- They reveal hidden risks, reuse patterns, quality issues, content decay, and optimization opportunities, at scale.

Without a graph, many of the operations listed below would require manual audits, risky assumptions, or unscalable automation. With a graph, you can X-ray your DITA library with SPARQL, automate governance, and strategically optimize your entire content operation.

This guide provides 100 practical SPARQL queries categorized by use case, each with:

- An introduction to the purpose.
- The SPARQL query.
- The business value unlocked.

Automating construction and maintenance of knowledge graphs for DITA

[Document Object Model Graph RAG](#) provides a complete guide about how to fully automate the building and updating of knowledge graphs from DITA XML content. You can download a copy at <https://thinkingdocumentation.com/downloads>.

Table of Contents

100 SPARQL Queries You Can Use with a DITA Knowledge Graph.....	1
Introduction	1
Why is a DITA knowledge graph transformative?	1
Automating construction and maintenance of knowledge graphs for DITA	1
100 General SPARQL Queries you can use with a DITA Knowledge Graph	8
Query #1: List all DITA topics and their types	8
Query #2: Find topics missing short descriptions.....	8
Query #3: Find all tasks and their steps.....	9
Query #4: Detect orphaned topics not referenced by any other document.....	9
Query #5: Find broken links inside the documentation corpus	10
Query #6: Find reused fragments (conrefs) and their reuse count.....	10
Query #7: Topics missing audience targeting metadata	11
Query #8: Find tasks missing result expectations.....	12
Query #9: Find concepts missing examples.....	12
Query #10: Detect glossary terms missing definitions	13
Query #11: Identify topics missing metadata for review status.....	13
Query #12: Find topics reused across multiple products	14
Query #13: Detect deprecated references in topics	14
Query #14: Identify tasks missing prerequisite sections	15
Query #15: Find concepts without defined relationships to other concepts	15
Query #16: Detect glossary terms used but not defined	16
Query #17: List topics containing multiple audience conditions	16
Query #18: Detect tasks without result validation steps	17
Query #19: Find topics linking to broken external URLs	17
Query #20: Identify overly reused fragments that risk maintenance issues	18
Query #21: Identify topics with missing or outdated keywords	18
Query #22: Detect topics where the short description is identical to the first para	19

Query #23: Find concepts without any example sections	19
Query #24: Identify topics modified after their last review date	20
Query #25: Find topics that have never been reviewed	20
Query #26: Identify topics with conditional steps (for only certain audiences)	21
Query #27: Find key definitions (keydefs) not referenced anywhere	21
Query #28: Detect topics with unusually high or low word counts	22
Query #29: List tasks sharing the same goal metadata	22
Query #30: Find glossary terms without usage examples	23
Query #31: Identify mentions of outdated technologies	23
Query #32: Find topics authored but never reviewed	24
Query #33: Identify overloaded content reused too often	24
Query #34: Detect steps missing result expectations	25
Query #35: Identify competitor names in content	25
Query #36: Find references to topics that no longer exist	26
Query #37: Identify topics with vague titles (e.g., "Introduction" or "Miscellaneous") ..	26
Query #38: Detect topics buried too deep in navigation hierarchies (>5 levels)	27
Query #39: List all external links to third-party websites	27
Query #40: Detect topics combining multiple content topic types	28
Query #41: Find glossary entries that are never referenced	28
Query #42: Detect images reused excessively	29
Query #43: List topics consisting of only one sentence	29
Query #44: Identify topics where short description duplicates first paragraph	30
Query #45: Track topics with the highest edit counts	30
Query #46: Identify topics updated but not re-reviewed	31
Query #47: Identify topics containing conflicting audience conditions	31
Query #48: Find topics with excessively large images attached	32
Query #49: Detect steps lacking verification or confirmation results	32
Query #50: List topics linking outside the authorized project scope	33
Query #51: Detect topics missing copyright information	33

Query #52: Find maps missing defined table of contents (ToC) ordering	34
Query #53: List conditional attributes used in the content	34
Query #54: Identify topics reused without being flagged for reuse	35
Query #55: Find topics using direct formatting instead of semantic tagging	35
Query #56: Identify topics that were translated but not updated after changes	36
Query #57: List topics where short descriptions exceed 150 characters	36
Query #58: Detect nested conref chains deeper than two levels	37
Query #59: Find topics reused in conflicting conditional profiles	37
Query #60: Identify topics linking to future-dated deliverables	38
Query #61: Find steps containing "click here" phrasing	38
Query #62: Identify topics linking externally to broken resources	39
Query #63: Detect topics using conflicting conditional profiles	39
Query #64: Find concept topics missing examples	40
Query #65: Identify tasks without error recovery procedures	40
Query #66: Find glossary entries defining obsolete terms	41
Query #67: Detect unreferenced reusable fragments	41
Query #68: Identify glossary terms used but not defined	42
Query #69: Detect reused topics conflicting across audience profiles	42
Query #70: Find tasks missing estimated duration metadata	43
Query #71: Detect topics with identical titles	43
Query #72: Identify topics authored by missing or unknown authors	44
Query #73: Detect glossary entries missing context examples	44
Query #74: Find topics updated after translation but not retranslated	44
Query #75: Identify content translated into unsupported locales	45
Query #76: Identify topics missing abstract sections	45
Query #77: Find content with inconsistent product names	46
Query #78: Detect over-conditionalized topics	46
Query #79: Identify topics referencing obsolete software versions	47
Query #80: Detect topics without cross-references to related content	47

Query #81: List topics whose file names don't match their titles	48
Query #82: Detect topics linking to disallowed external domains	48
Query #83: Find tasks missing expected output descriptions	49
Query #84: Identify topics without security classifications	49
Query #85: Detect reused topics with conflicting reuse metadata	50
Query #86: Find topics missing translation status metadata	50
Query #87: Detect topics with more than 10 conditional attributes	51
Query #88: Identify reused images missing alt text	51
Query #89: Find steps referencing deprecated UI elements	52
Query #90: Detect topics using inline styles violating branding guidelines	52
Query #91: Identify conditional attributes missing documented values	53
Query #92: Detect tasks lacking prerequisite descriptions	53
Query #93: Identify topics modified after last localization	54
Query #94: Detect glossary entries missing part-of-speech indicators	54
Query #95: Find content reusing obsolete fragments	55
Query #96: Identify topics using deprecated metadata elements	55
Query #97: Detect tasks lacking verification checkpoints	56
Query #98: Identify topics that reference both internal and external URLs	56
Query #99: Find tasks missing role-specific guidance	56
Query #100: Detect content with inconsistent date formatting	57
25 SPARQL Queries for DITA Knowledge Graph RAG Content Retrieval	58
RAG Query #1: Retrieve All DITA Maps	58
RAG Query #2: Retrieve All DITA Topics	58
RAG Query #3: Retrieve Specific Topic Types (Tasks)	59
RAG Query #4: Find Topics Referenced in Maps	59
RAG Query #5: Retrieve Topics Without a Map Reference	59
RAG Query #6: Retrieve All Elements Within a Specific Topic	60
RAG Query #7: Retrieve Topics by Short Description (Shortdesc)	60
RAG Query #8: Retrieve Maps Containing a Specific Topic	61

RAG Query #9: Retrieve Topics by Metadata (Audience, Platform)	61
RAG Query #10: Retrieve All Key Definitions	62
RAG Query #11: Retrieve Topics Filtered by Conditional Attribute (e.g., Product)	62
RAG Query #12: Retrieve Topics Missing a Title	63
RAG Query #13: Detect Broken Topic References.....	63
RAG Query #14: List All Cross-References Between Topics.....	64
RAG Query #15: Retrieve Topics Using a Specific Key	64
RAG Query #16: Retrieve Elements With a Specific Role (e.g., UI Elements)	65
RAG Query #17: Retrieve Reusable Content (Conrefs).....	66
RAG Query #18: Retrieve Glossary Entries.....	66
RAG Query #19: Retrieve Topics With External Links.....	67
RAG Query #20: Retrieve Topics By Author Metadata.....	67
RAG Query #21: Retrieve Recently Modified Topics	68
RAG Query #22: Retrieve Maps for a Specific Output Class (e.g., Training)	68
RAG Query #23: Retrieve Topics Using a Specific Metadata Value	69
RAG Query #24: Retrieve Topics Sorted by Title Alphabetically.....	69
RAG Query #25: Detect and List Duplicate Titles Across Topics.....	70
25 SPARQL Queries for Mining DITA Content Metrics	72
Metrics Query #1: Count Total Number of Topics	72
Metrics Query #2: Count Total Number of Maps.....	72
Metrics Query #3: Average Number of Topics per Map	73
Metrics Query #4: Find Top 10 Most Reused Topics	73
Metrics Query #5: Count Orphaned Topics (Not Referenced in Maps).....	74
Metrics Query #6: Identify Topics with Missing Short Descriptions	75
Metrics Query #7: Identify Topics Missing Titles	75
Metrics Query #8: Total Number of Conref Reuse Instances	76
Metrics Query #9: Distribution of Topics by Type (Concept, Task, Reference)	76
Metrics Query #10: Percentage of Topics Reused in More Than One Map	77
Metrics Query #11: Identify Topics with the Most Conditional Attributes.....	78

Metrics Query #12: Count Topics with Broken References	78
Metrics Query #13: Average Number of Cross-References per Topic	79
Metrics Query #14: Count of Glossary Entries (Glossentry Topics)	80
Metrics Query #15: Average Topic Size (Elements per Topic)	80
Metrics Query #16: Identify Topics Last Modified Over X Years Ago	81
Metrics Query #17: Identify Authors with Most Contributions	81
Metrics Query #18: Identify Maps with the Highest Topic Counts	82
Metrics Query #19: Percentage of Topics With External Links	83
Metrics Query #20: Identify Topics with the Most Key References	83
Metrics Query #21: Calculate Reuse ROI Estimate	84
Metrics Query #22: Topics per Author (Average)	84
Metrics Query #23: Detect Duplicate Titles Across Topics	85
Metrics Query #24: Maps Containing Only One Topic	86
Metrics Query #25: Average Age of Topics (Content Freshness)	86
Summary: Why DITA Knowledge Graphs Are the Future of Content Management	87
Appendix: How to Run These Queries in Your Environment	87

100 General SPARQL Queries you can use with a DITA Knowledge Graph

Query #1: List all DITA topics and their types

Introduction

Understanding the full inventory of topics is foundational for any large content governance program.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title ?type

WHERE {

    ?topic a ?type ;
        dita:title ?title .

    FILTER(?type IN (dita:Concept, dita:Task, dita:Reference, dita:Topic))

}

ORDER BY ?title
```

Business Value

Provides a global content inventory to prioritize updates, audits, reuse optimization, and publishing planning.

Query #2: Find topics missing short descriptions

Introduction

Short descriptions (shortdesc) are critical for SEO, usability, and scannability. Topics missing them undermine content effectiveness.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title

WHERE {

    ?topic a dita:Topic ;
        dita:title ?title .

}
```

```
FILTER NOT EXISTS { ?topic dita:shortdesc ?shortdesc }  
}  
ORDER BY ?title
```

Business Value

Ensures topics meet content quality standards, improves search ranking, user satisfaction, and content snippet generation.

Query #3: Find all tasks and their steps

Introduction

Mapping task steps is critical for modularity, translation readiness, and task simplification.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
SELECT ?task ?taskTitle ?step  
WHERE {  
  ?task a dita:Task ;  
    dita:title ?taskTitle ;  
    dita:hasStep ?step .  
}  
ORDER BY ?task ?step
```

Business Value

Improves modular authoring, localization scoping, and workflow automation for procedural content.

Query #4: Detect orphaned topics not referenced by any other document

Introduction

Orphaned topics waste maintenance resources and clutter the content library.

SPARQL Query

PREFIX dita: <http://example.org/dita#>

SELECT ?orphan

WHERE {

 ?orphan a dita:Topic .

 FILTER NOT EXISTS { ?anyDoc dita:xref ?orphan }

}

Business Value

Helps prioritize cleanup, reduce translation waste, and maintain a lean, useful content set.

Query #5: Find broken links inside the documentation corpus

Introduction

Broken links harm user trust and navigation efficiency.

SPARQL Query

PREFIX dita: <http://example.org/dita#>

SELECT ?source ?brokenLink

WHERE {

 ?source dita:xref ?brokenLink .

 FILTER NOT EXISTS { ?brokenLink a dita:Topic }

}

Business Value

Protects user experience, brand credibility, and output quality by finding link errors before publishing.

Query #6: Find reused fragments (conrefs) and their reuse count

Introduction

Identify content fragments reused in multiple locations to optimize reuse governance.

SPARQL Query

PREFIX dita: <http://example.org/dita#>

SELECT ?fragment (COUNT(?use) AS ?useCount)

WHERE {

 ?use dita:conref ?fragment .

}

GROUP BY ?fragment

HAVING (COUNT(?use) > 1)

ORDER BY DESC(?useCount)

Business Value

Reused content management reduces maintenance costs and translation expenses.

Query #7: Topics missing audience targeting metadata

Introduction

Ensure every topic has audience attributes to enable personalized content delivery.

SPARQL Query

PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title

WHERE {

 ?topic a dita:Topic ;

 dita:title ?title .

 FILTER NOT EXISTS { ?topic dita:audience ?aud }

}

ORDER BY ?title

Business Value Improves content personalization, findability, and conditional publishing readiness.

Query #8: Find tasks missing result expectations

Introduction

Task topics should describe the expected outcome after completion.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:result ?result }

}
```

Business Value Enhances instructional quality, user success rates, and procedure validation.

Query #9: Find concepts missing examples

Introduction

Concept topics without examples risk being too abstract for effective understanding.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?concept

WHERE {

  ?concept a dita:Concept .

  FILTER NOT EXISTS { ?concept dita: example ?example }

}
```

Business Value

Boosts comprehension, adoption rates, and user satisfaction.

Query #10: Detect glossary terms missing definitions

Introduction

Glossary entries without definitions confuse users and hurt terminology management.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry .

  FILTER NOT EXISTS { ?glossEntry dita:glossdef ?definition }

}
```

Business Value

Strengthens terminology quality and cross-project consistency.

Query #11: Identify topics missing metadata for review status

Introduction

Review status metadata ensures all topics are validated for quality assurance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic a dita:Topic .

  FILTER NOT EXISTS { ?topic dita:reviewStatus ?status }

}
```

Business Value

Ensures regulatory compliance and editorial consistency.

Query #12: Find topics reused across multiple products

Introduction

Identify content that needs conditional filtering based on product usage.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic (COUNT(?product) AS ?productCount)

WHERE {

    ?topic dita:audience ?product .

}

GROUP BY ?topic

HAVING (COUNT(?product) > 1)
```

Business Value

Optimizes reuse, improves product-specific documentation, and simplifies maintenance.

Query #13: Detect deprecated references in topics

Introduction

Ensure no topics link to deprecated APIs, products, or outdated content.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?xref

WHERE {

    ?topic dita:xref ?xref .

    ?xref dita:status "deprecated" .

}
```

Business Value

Maintains up-to-date, trustworthy documentation.

Query #14: Identify tasks missing prerequisite sections

Introduction

Prerequisites clarify what users need before starting a task.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:prereq ?pre }

}
```

Business Value

Improves task completion success rates and user satisfaction.

Query #15: Find concepts without defined relationships to other concepts

Introduction

Linked concepts enhance navigability and learning paths.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?concept

WHERE {

  ?concept a dita:Concept .

  FILTER NOT EXISTS { ?concept dita:relatedConcept ?other }

}
```

Business Value

Strengthens content discoverability and user education flow.

Query #16: Detect glossary terms used but not defined

Introduction

Terms should be properly linked to glossary definitions.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?term

WHERE {

  ?topic dita:mentionsTerm ?term .

  FILTER NOT EXISTS { ?term a dita:GlossEntry }

}
```

Business Value

Boosts consistency and user trust in terminology.

Query #17: List topics containing multiple audience conditions

Introduction

Complex audience targeting needs monitoring for correctness.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic (COUNT(?audience) AS ?audCount)

WHERE {

  ?topic dita:audience ?audience .

}

GROUP BY ?topic

HAVING (COUNT(?audience) > 1)
```

Business Value

Improves personalization control and reduces publishing errors.

Query #18: Detect tasks without result validation steps

Introduction

Tasks should include steps to verify successful completion.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:validation ?validation }

}
```

Business Value

Enhances instructional reliability and error prevention.

Query #19: Find topics linking to broken external URLs

Introduction

Broken external links damage trust and usability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?link

WHERE {

  ?topic dita:externalLink ?link .

  ?link dita:status "broken" .

}
```

Business Value

Protects brand reputation and improves user experience.

Query #20: Identify overly reused fragments that risk maintenance issues

Introduction

Excessive reuse can create fragile, brittle systems.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?fragment (COUNT(?usage) AS ?useCount)

WHERE {

    ?usage dita:conref ?fragment .

}

GROUP BY ?fragment

HAVING (COUNT(?usage) > 20)

ORDER BY DESC(?useCount)
```

Business Value

Highlights content maintenance hotspots and allows proactive risk management.

Query #21: Identify topics with missing or outdated keywords

Introduction

Keywords enhance searchability and content categorization. Missing or outdated ones harm findability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

    ?topic a dita:Topic .

    FILTER NOT EXISTS { ?topic dita:keyword ?kw }

}
```

Business Value

Improves SEO, dynamic publishing, and navigation structures.

Query #22: Detect topics where the short description is identical to the first para

Introduction

Short descriptions should summarize, not duplicate, the topic opening.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic dita:shortdesc ?sd ;
        dita:firstParagraph ?fp .
  FILTER(?sd = ?fp)
}
```

Business Value

Enhances scannability, snippet usability, and metadata clarity.

Query #23: Find concepts without any example sections

Introduction

Examples are crucial for understanding complex concepts.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?concept
WHERE {
  ?concept a dita:Concept .
  FILTER NOT EXISTS { ?concept dita:example ?example }
}
```

Business Value

Boosts comprehension and supports knowledge retention.

Query #24: Identify topics modified after their last review date

Introduction

Content modified post-review may be unvalidated and noncompliant.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?lastModified ?lastReviewed

WHERE {

  ?topic dita:lastModified ?lastModified ;
        dita:lastReviewed ?lastReviewed .

  FILTER(?lastModified > ?lastReviewed)

}
```

Business Value

Ensures up-to-date, validated deliverables and reduces compliance risks.

Query #25: Find topics that have never been reviewed

Introduction Unreviewed topics increase publishing risk and introduce errors.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic a dita:Topic .

  FILTER NOT EXISTS { ?topic dita:reviewedBy ?reviewer }

}
```

Business Value

Improves content governance and publishing reliability.

Query #26: Identify topics with conditional steps (for only certain audiences)

Introduction

Conditional steps help deliver customized workflows, but must be properly tracked.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task ?step

WHERE {

  ?task a dita:Task ;

    dita:hasStep ?step .

  ?step dita:audience ?audience .

}
```

Business Value

Improves accuracy for audience-specific deliverables.

Query #27: Find key definitions (keydefs) not referenced anywhere

Introduction

Unused keys clutter the content system and risk inconsistencies.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?keydef

WHERE {

  ?keydef a dita:KeyDefinition .

  FILTER NOT EXISTS { ?anyTopic dita:keyref ?keydef }

}
```

Business Value

Reduces complexity, streamlines maintenance.

Query #28: Detect topics with unusually high or low word counts

Introduction

Outlier topics often need revision for better modularity or depth.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?wordCount

WHERE {

  ?topic a dita:Topic ;

        dita:wordCount ?wordCount .

  FILTER(?wordCount < 50 || ?wordCount > 2000)

}
```

Business Value

Improves modularity and usability.

Query #29: List tasks sharing the same goal metadata

Introduction Duplicate task goals may indicate redundant procedures.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?taskA ?taskB

WHERE {

  ?taskA a dita:Task ; dita:goal ?goal .

  ?taskB a dita:Task ; dita:goal ?goal .

  FILTER(?taskA != ?taskB)

}
```

Business Value

Optimizes task libraries, reduces duplication.

Query #30: Find glossary terms without usage examples

Introduction

Glossary terms should demonstrate usage in context for clarity.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry .

  FILTER NOT EXISTS { ?glossEntry dita:example ?example }

}
```

Business Value

Enhances glossary value and training usability.

Query #31: Identify mentions of outdated technologies

Introduction

References to obsolete technologies degrade content relevance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:body ?bodyText .

  FILTER(CONTAINS(LCASE(STR(?bodyText)), "windows xp"))

}
```

Business Value

Supports modernization initiatives and compliance.

Query #32: Find topics authored but never reviewed

Introduction

Ensures all authored material has gone through the review process.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

    ?topic a dita:Topic ;

        dita:author ?author .

    FILTER NOT EXISTS { ?topic dita:reviewer ?reviewer }

}
```

Business Value

Boosts quality assurance and reduces errors.

Query #33: Identify overloaded content reused too often

Introduction

High-reuse fragments become fragile and need governance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?fragment (COUNT(?usage) AS ?count)

WHERE {

    ?usage dita:conref ?fragment .

}

GROUP BY ?fragment

HAVING (COUNT(?usage) > 20)

ORDER BY DESC(?count)
```

Business Value

Improves reuse resilience.

Query #34: Detect steps missing result expectations

Introduction

Each step should produce a measurable result.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?step

WHERE {

  ?step a dita:Step .

  FILTER NOT EXISTS { ?step dita:result ?result }

}
```

Business Value

Boosts task effectiveness and user confidence.

Query #35: Identify competitor names in content

Introduction

Mentions of competitors might cause brand and legal risks.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:body ?bodyContent .

  FILTER(CONTAINS(LCASE(STR(?bodyContent)), "competitorcorp"))

}
```

Business Value

Prevents branding and legal exposure.

Query #36: Find references to topics that no longer exist

Introduction

Broken topic links damage user navigation and publishing integrity.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?sourceTopic ?missingTarget

WHERE {

  ?sourceTopic dita:xref ?missingTarget .

  FILTER NOT EXISTS { ?missingTarget a dita:Topic }

}
```

Business Value

Reduces broken links that harm usability and output quality.

Query #37: Identify topics with vague titles (e.g., "Introduction" or "Miscellaneous")

Introduction

Precise titles improve SEO, findability, and information scent.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title

WHERE {

  ?topic a dita:Topic ; dita:title ?title .

  FILTER (regex(lower(str(?title)), "introduction|overview|miscellaneous|general|other"))

}
```

Business Value

Strengthens topic clarity and search relevance.

Query #38: Detect topics buried too deep in navigation hierarchies (>5 levels)

Introduction

Deep navigation hurts user orientation and task efficiency.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?root ?descendant (COUNT(?path) AS ?depth)

WHERE {

  ?root dita:hasTopicRef+ ?descendant .

}

GROUP BY ?root ?descendant

HAVING (COUNT(?path) > 5)

ORDER BY DESC(?depth)
```

Business Value

Optimizes structure for flatter, faster navigation.

Query #39: List all external links to third-party websites

Introduction

External links should be monitored for health and appropriateness.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?externalHref

WHERE {

  ?topic a dita:Topic ; dita:href ?externalHref .

  FILTER (STRSTARTS(STR(?externalHref), "http"))

}
```

Business Value

Prevents broken, outdated, or risky external references.

Query #40: Detect topics combining multiple content topic types

Introduction

Maintaining genre purity improves modularity and cognitive load.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title

WHERE {

  ?topic a ?type ; dita:title ?title .

  FILTER(?type IN (dita:Task, dita:Concept, dita:Reference))

  FILTER EXISTS { ?topic a dita:Task }

  FILTER EXISTS { ?topic a dita:Concept }

}
```

Business Value

Improves topic purpose clarity and reusability.

Query #41: Find glossary entries that are never referenced

Introduction

Unused glossary entries bloat deliverables unnecessarily.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry .

  FILTER NOT EXISTS { ?topic dita:xref ?glossEntry }

}
```

Business Value

Keeps glossary current and efficient.

Query #42: Detect images reused excessively

Introduction

Overused images may lose explanatory power.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?image (COUNT(?topic) AS ?usageCount)

WHERE {

  ?topic dita:image ?image .

}

GROUP BY ?image

HAVING (COUNT(?topic) > 20)

ORDER BY DESC(?usageCount)
```

Business Value

Promotes visual variety and contextual clarity.

Query #43: List topics consisting of only one sentence

Introduction

Thin content underdelivers on user expectations.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?sentenceCount

WHERE {

  ?topic a dita:Topic ; dita:sentenceCount ?sentenceCount .

  FILTER(?sentenceCount <= 1)

}
```

```
}
```

Business Value

Improves depth and value of topics.

Query #44: Identify topics where short description duplicates first paragraph

Introduction

Duplicate short descriptions waste content real estate.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic dita:shortdesc ?shortdesc ;
        dita:firstParagraph ?firstPara .
  FILTER (STR(?shortdesc) = STR(?firstPara))
}
```

Business Value Improves metadata quality and snippet generation.

Query #45: Track topics with the highest edit counts

Introduction

High edit frequency signals complex or unstable topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?editCount
WHERE {
  ?topic a dita:Topic ; dita:editCount ?editCount .
}
```

ORDER BY DESC(?editCount)

Business Value

Targets unstable content for redesign or deeper review.

Query #46: Identify topics updated but not re-reviewed

Introduction

Content modified after review risks being outdated or inaccurate.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?lastModified ?lastReviewed

WHERE {

  ?topic dita:dateModified ?lastModified ;
         dita:dateReviewed ?lastReviewed .

  FILTER (?lastModified > ?lastReviewed)

}
```

Business Value

Supports compliance and governance in content operations.

Query #47: Identify topics containing conflicting audience conditions

Introduction

Conflicting audience conditions can cause incorrect filtering and publishing errors.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?audience1 ?audience2

WHERE {

  ?topic dita:audience ?audience1, ?audience2 .

  FILTER(?audience1 != ?audience2)

}
```


Business Value

Improves publishing precision and audience targeting.

Query #48: Find topics with excessively large images attached

Introduction

Large images can slow down page loads and damage mobile usability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?image

WHERE {

  ?topic dita:image ?image .

  ?image dita:fileSize ?size .

  FILTER(?size > 5000000) # Over 5MB

}
```

Business Value

Optimizes media performance for better user experience.

Query #49: Detect steps lacking verification or confirmation results

Introduction

Critical tasks should confirm success or provide error detection at completion.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?step

WHERE {

  ?step a dita:Step .

  FILTER NOT EXISTS { ?step dita:confirmation ?confirmation }

}
```

Business

Value Improves procedure safety and completeness.

Query #50: List topics linking outside the authorized project scope

Introduction

Cross-linking into unauthorized spaces introduces security and publishing risks.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?externalReference

WHERE {

  ?topic dita:xref ?externalReference .

  FILTER NOT EXISTS { ?externalReference a dita:Topic }

}
```

Business Value

Protects content boundaries and security policies.

Query #51: Detect topics missing copyright information

Introduction

Proper copyright metadata is crucial for legal protection.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic a dita:Topic .

  FILTER NOT EXISTS { ?topic dita:copyrightHolder ?holder }

}
```

Business Value

Ensures intellectual property rights are properly protected.

Query #52: Find maps missing defined table of contents (ToC) ordering

Introduction

Proper ToC ordering improves navigation and publishing structure.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map

WHERE {

  ?map a dita:Map .

  FILTER NOT EXISTS { ?map dita:tocOrder ?order }

}
```

Business Value

Enhances document usability and professionalism.

Query #53: List conditional attributes used in the content

Introduction

Identifying active conditions improves profiling control.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT DISTINCT ?attribute

WHERE {

  ?element ?attribute ?value .

  FILTER(STRSTARTS(STR(?attribute), "http://example.org/dita#conditionalAttribute"))

}
```

Business Value

Streamlines conditional publishing strategies.

Query #54: Identify topics reused without being flagged for reuse

Introduction

All reused content should be governed by explicit reuse policies.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:conref ?usedBy .

  FILTER NOT EXISTS { ?topic dita:reusability "true" }

}
```

Business Value

Enhances reuse governance and maintenance reliability.

Query #55: Find topics using direct formatting instead of semantic tagging

Introduction

Hardcoded styles (bold, italic) hinder consistency and adaptive publishing.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?body

WHERE {

  ?topic dita:body ?body .

  FILTER(REGEX(STR(?body), "<b>|<i>|<u>|<font>", "i"))

}
```

Business Value

Enforces semantic structure for multi-channel publishing.

Query #56: Identify topics that were translated but not updated after changes

Introduction

Translations must reflect the current source.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?translationDate ?modifiedDate

WHERE {

  ?topic dita:dateTranslated ?translationDate ;
        dita:dateModified ?modifiedDate .

  FILTER(?modifiedDate > ?translationDate)

}
```

Business Value

Prevents out-of-sync localized content.

Query #57: List topics where short descriptions exceed 150 characters

Introduction

Short descriptions should be concise for scanning and SEO.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?shortdesc

WHERE {

  ?topic dita:shortdesc ?shortdesc .

  FILTER(STRLEN(STR(?shortdesc)) > 150)

}
```

Business Value

Improves snippet usability and findability.

Query #58: Detect nested conref chains deeper than two levels

Introduction

Deep conref nesting increases maintenance fragility.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?outer ?middle ?inner

WHERE {

  ?outer dita:conref ?middle .

  ?middle dita:conref ?inner .

}
```

Business Value

Reduces build errors and maintenance risks.

Query #59: Find topics reused in conflicting conditional profiles

Introduction

Reuse conflicts can corrupt deliverable accuracy.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?cond1 ?cond2

WHERE {

  ?usage1 dita:conref ?topic ; dita:conditionalAttribute ?cond1 .

  ?usage2 dita:conref ?topic ; dita:conditionalAttribute ?cond2 .

  FILTER(?cond1 != ?cond2)

}
```

Business Value

Protects content fidelity across personalized outputs.

Query #60: Identify topics linking to future-dated deliverables

Introduction

Premature linking causes user confusion and broken workflows.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?futureTarget

WHERE {

  ?topic dita:xref ?futureTarget .

  ?futureTarget dita:releaseDate ?releaseDate .

  FILTER(?releaseDate > NOW())

}
```

Business Value

Improves temporal integrity of content networks.

Query #61: Find steps containing "click here" phrasing

Introduction

Poor UX writing practices like "click here" reduce accessibility and clarity.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?step ?text

WHERE {

  ?step a dita:Step ;

    dita:stepText ?text .

  FILTER (CONTAINS(LCASE(STR(?text)), "click here"))

}
```

Business Value

Improves usability, accessibility, and SEO.

Query #62: Identify topics linking externally to broken resources

Introduction

Broken external links degrade user trust and SEO ranking.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?externalLink

WHERE {

  ?topic dita:externalLink ?externalLink .
  ?externalLink dita:linkStatus "broken" .
}
```

Business Value

Protects user experience and brand credibility.

Query #63: Detect topics using conflicting conditional profiles

Introduction

Conflicting conditions confuse output processing and personalization.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:conditionalAttribute ?condA, ?condB .

  FILTER(?condA != ?condB)
}
```

Business Value

Ensures clean profiling and output generation.

Query #64: Find concept topics missing examples

Introduction

Concepts without examples fail to anchor abstract information for users.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?concept

WHERE {

  ?concept a dita:Concept .

  FILTER NOT EXISTS { ?concept dita:example ?example }

}
```

Business Value

Improves concept usability and teaching effectiveness.

Query #65: Identify tasks without error recovery procedures

Introduction

Tasks must guide users on what to do when failures occur.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:recovery ?recovery }

}
```

Business Value

Enhances user confidence and task reliability.

Query #66: Find glossary entries defining obsolete terms

Introduction

Outdated glossary terms dilute the precision of the terminology set.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry ;

              dita:glossterm ?term .

  FILTER(CONTAINS(LCASE(STR(?term)), "legacy"))

}
```

Business Value

Streamlines active terminology and boosts clarity.

Query #67: Detect unreferenced reusable fragments

Introduction

Unused fragments clutter libraries and increase confusion.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?fragment

WHERE {

  ?fragment a dita:ContentFragment .

  FILTER NOT EXISTS { ?topic dita:conref ?fragment }

}
```

Business Value

Improves reuse library health and maintainability.

Query #68: Identify glossary terms used but not defined

Introduction

Undefined terms create inconsistency and reduce user confidence.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?term

WHERE {

  ?topic dita:mentionsTerm ?term .

  FILTER NOT EXISTS { ?term a dita:GlossEntry }

}
```

Business Value

Strengthens terminology management and localization quality.

Query #69: Detect reused topics conflicting across audience profiles

Introduction

Reuse inconsistencies across audiences cause publishing errors.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?use1 dita:conref ?topic ; dita:audience ?aud1 .

  ?use2 dita:conref ?topic ; dita:audience ?aud2 .

  FILTER(?aud1 != ?aud2)

}
```

Business Value

Ensures correctness in audience-specific outputs.

Query #70: Find tasks missing estimated duration metadata

Introduction

Duration helps users plan and allocate resources for task completion.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:estimatedDuration ?duration }

}
```

Business Value

Enhances task transparency and usability.

Query #71: Detect topics with identical titles

Introduction

Duplicate titles confuse users and damage findability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?title (COUNT(?topic) AS ?count)

WHERE {

  ?topic a dita:Topic ; dita:title ?title .

}

GROUP BY ?title

HAVING (COUNT(?topic) > 1)
```

Business Value

Supports disambiguation and improves navigation.

Query #72: Identify topics authored by missing or unknown authors

Introduction

Missing authorship records hinder content ownership.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic a dita:Topic .

  FILTER NOT EXISTS { ?topic dita:author ?author }

}
```

Business Value

Improves accountability and content traceability.

Query #73: Detect glossary entries missing context examples

Introduction Examples help clarify glossary term usage.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry .

  FILTER NOT EXISTS { ?glossEntry dita:example ?example }

}
```

Business Value

Enhances glossary effectiveness and user learning.

Query #74: Find topics updated after translation but not retranslated

Introduction

Mismatch between source and translated versions risks outdated localized content.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:dateModified ?mod ;
        dita:dateTranslated ?trans .

  FILTER(?mod > ?trans)

}
```

Business Value

Protects localized deliverable integrity.

Query #75: Identify content translated into unsupported locales

Introduction

Translations for unsupported markets waste resources and confuse systems.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?locale

WHERE {

  ?topic dita:locale ?locale .

  FILTER(?locale NOT IN ("en", "de", "fr", "es"))

}
```

Business Value

Aligns localization strategy with business priorities.

Query #76: Identify topics missing abstract sections

Introduction

Abstracts provide quick topic summaries and context for users.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

    ?topic a dita:Topic .

    FILTER NOT EXISTS { ?topic dita:abstract ?abstract }

}
```

Business Value

Improves content scannability and comprehension.

Query #77: Find content with inconsistent product names

Introduction

Inconsistent terminology undermines brand consistency.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

    ?topic dita:body ?text .

    FILTER(CONTAINS(LCASE(STR(?text)), "wrongproductname"))

}
```

Business Value

Strengthens branding and prevents confusion.

Query #78: Detect over-conditionalized topics

Introduction

Excessive conditional profiling complicates content maintenance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?topic (COUNT(?cond) AS ?condCount)
WHERE {
  ?topic dita:conditionalAttribute ?cond .
}
GROUP BY ?topic
HAVING (COUNT(?cond) > 5)
```

Business Value

Simplifies delivery pipelines and improves readability.

Query #79: Identify topics referencing obsolete software versions

Introduction

Keeping software references current maintains relevance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
SELECT ?topic
WHERE {
  ?topic dita:body ?bodyText .
  FILTER(CONTAINS(LCASE(STR(?bodyText)), "version 1.0"))
}
```

Business Value Prevents outdated information from reaching users.

Query #80: Detect topics without cross-references to related content

Introduction

Cross-linking improves navigation and discoverability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
SELECT ?topic
```



```
WHERE {  
  ?topic a dita:Topic .  
  FILTER NOT EXISTS { ?topic dita:xref ?xref }  
}
```

Business Value

Strengthens UX and content connectivity.

Query #81: List topics whose file names don't match their titles

Introduction

File-title mismatches hinder traceability and publishing accuracy.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
SELECT ?topic ?filename ?title  
WHERE {  
  ?topic dita:filename ?filename ; dita:title ?title .  
  FILTER(!CONTAINS(LCASE(STR(?filename)), LCASE(STR(?title))))  
}
```

Business Value

Improves CMS hygiene and downstream automation reliability.

Query #82: Detect topics linking to disallowed external domains

Introduction

Security policies may restrict linking to certain external sites.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
SELECT ?topic ?externalLink  
WHERE {
```

```
?topic dita:externalLink ?externalLink .  
  
FILTER(CONTAINS(STR(?externalLink), "untrusteddomain.com"))  
  
}
```

Business Value

Reduces security risks and improves compliance.

Query #83: Find tasks missing expected output descriptions

Introduction

Expected outputs verify if a task was successful.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?task  
  
WHERE {  
  
    ?task a dita:Task .  
  
    FILTER NOT EXISTS { ?task dita:expectedOutput ?output }  
  
}
```

Business Value

Improves user validation and task effectiveness.

Query #84: Identify topics without security classifications

Introduction

Security classifications help protect sensitive information.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic  
  
WHERE {  
  
    ?topic a dita:Topic .  
  
}
```

```
FILTER NOT EXISTS { ?topic dita:securityClassification ?classification }  
}
```

Business Value

Improves information security governance.

Query #85: Detect reused topics with conflicting reuse metadata

Introduction

Conflicting reuse metadata introduces governance problems.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic ?reuser1 ?reuser2  
  
WHERE {  
  
  ?reuser1 dita:conref ?topic .  
  
  ?reuser2 dita:conref ?topic .  
  
  FILTER(?reuser1 != ?reuser2)  
  
}
```

Business Value

Enforces cleaner reuse management practices.

Query #86: Find topics missing translation status metadata

Introduction

Tracking translation status ensures deliverables are localization-ready.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic  
  
WHERE {  
  
  ?topic a dita:Topic .  
  
}
```

```
FILTER NOT EXISTS { ?topic dita:translationStatus ?status }  
}
```

Business Value

Improves localization project planning and tracking.

Query #87: Detect topics with more than 10 conditional attributes

Introduction

Over-complexity from conditionalization creates fragile content.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic (COUNT(?cond) AS ?condCount)  
  
WHERE {  
    ?topic dita:conditionalAttribute ?cond .  
}  
  
GROUP BY ?topic  
  
HAVING (COUNT(?cond) > 10)
```

Business Value

Simplifies content and reduces maintenance costs.

Query #88: Identify reused images missing alt text

Introduction

Alt text improves accessibility and SEO for images.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?image  
  
WHERE {  
    ?image a dita:Image .
```

```
FILTER NOT EXISTS { ?image dita:altText ?alt }  
}
```

Business Value

Ensures accessibility compliance and better UX.

Query #89: Find steps referencing deprecated UI elements

Introduction

Procedures must stay aligned with current UI designs.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?step  
  
WHERE {  
  
  ?step a dita:Step ; dita:stepText ?text .  
  
  FILTER(CONTAINS(LCASE(STR(?text)), "oldbutton"))  
  
}
```

Business Value

Keeps documentation current and avoids user confusion.

Query #90: Detect topics using inline styles violating branding guidelines

Introduction

Inline formatting breaks corporate content standards.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic ?body  
  
WHERE {  
  
  ?topic dita:body ?body .  
  
  FILTER(REGEX(STR(?body), "<font|color|size", "i"))  
  
}
```

```
}
```

Business Value

Strengthens brand compliance and improves multi-channel publishing.

Query #91: Identify conditional attributes missing documented values

Introduction

Undocumented conditions create ambiguity in outputs.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?cond
WHERE {
    ?topic ?cond ?value .
    FILTER(STRSTARTS(STR(?cond), "http://example.org/dita#conditionalAttribute"))
    FILTER(STRLEN(STR(?value)) = 0)
}
```

Business Value

Improves conditional processing clarity and reduces errors.

Query #92: Detect tasks lacking prerequisite descriptions

Introduction

Prerequisites help users understand setup needs.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task
WHERE {
    ?task a dita:Task .
    FILTER NOT EXISTS { ?task dita:prereq ?prereq }
```

```
}
```

Business Value

Improves task success rates and user satisfaction.

Query #93: Identify topics modified after last localization

Introduction

Updated content must trigger localization updates.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?modified ?localized

WHERE {

  ?topic dita:dateModified ?modified ;
        dita:dateLocalized ?localized .

  FILTER(?modified > ?localized)

}
```

Business Value Protects localized content integrity.

Query #94: Detect glossary entries missing part-of-speech indicators

Introduction

Glossaries are clearer with grammatical metadata.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?glossEntry

WHERE {

  ?glossEntry a dita:GlossEntry .

  FILTER NOT EXISTS { ?glossEntry dita:partOfSpeech ?pos }

}
```

Business Value

Strengthens glossary clarity and localization quality.

Query #95: Find content reusing obsolete fragments

Introduction

Old reused content risks brand and information accuracy.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?usage ?fragment

WHERE {

  ?usage dita:conref ?fragment .

  ?fragment dita:status "obsolete" .

}
```

Business Value

Protects brand reputation and ensures content freshness.

Query #96: Identify topics using deprecated metadata elements

Introduction

Stale metadata causes processing errors.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:deprecatedMetadata ?meta .

}
```

Business Value

Ensures smooth delivery and future-proofs systems.

Query #97: Detect tasks lacking verification checkpoints

Introduction

Procedural verification is critical for success.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:validation ?validation }

}
```

Business Value

Boosts user trust and procedural integrity.

Query #98: Identify topics that reference both internal and external URLs

Introduction

Mixed-link topics must be monitored for reliability.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic

WHERE {

  ?topic dita:xref ?internal .

  ?topic dita:externalLink ?external .

}
```

Business Value Helps track content portability and link management.

Query #99: Find tasks missing role-specific guidance

Introduction

Different user roles may require specialized task versions.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task

WHERE {

  ?task a dita:Task .

  FILTER NOT EXISTS { ?task dita:role ?role }

}
```

Business Value

Enhances personalization and task success.

Query #100: Detect content with inconsistent date formatting

Introduction

Inconsistent dates cause confusion and localization problems.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?date

WHERE {

  ?topic dita:dateCreated ?date .

  FILTER(!REGEX(STR(?date), "^\\d{4}-\\d{2}-\\d{2}$"))

}
```

Business Value

Improves content uniformity and global readiness.

25 SPARQL Queries for DITA Knowledge Graph RAG

Content Retrieval

RAG Query #1: Retrieve All DITA Maps

Introduction

Retrieves all DITA maps in the knowledge graph.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map ?title
WHERE {
  ?map a dita:Map .
  OPTIONAL { ?map dita:title ?title . }
}
```

Business Value

Quickly locate all deliverables for structure analysis and reuse.

RAG Query #2: Retrieve All DITA Topics

Introduction

Retrieves all standalone topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?title
WHERE {
  ?topic a dita:Topic .
  OPTIONAL { ?topic dita:title ?title . }
}
```

Business Value

Full inventory for auditing and reuse.

RAG Query #3: Retrieve Specific Topic Types (Tasks)

Introduction

Retrieves only task topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?task ?title
WHERE {
  ?task a dita:Task .
  OPTIONAL { ?task dita:title ?title . }
}
```

Business Value:

Focuses on procedural content for quality control.

RAG Query #4: Find Topics Referenced in Maps

Introduction

Lists topics linked in maps.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map ?topic
WHERE {
  ?map a dita:Map .
  ?map dita:references ?topic .
}
```

Business Value

Measures reuse and topic impact.

RAG Query #5: Retrieve Topics Without a Map Reference

Introduction

Finds orphaned topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic a dita:Topic .
  FILTER NOT EXISTS { ?map dita:references ?topic . }
}
```

Business Value

Identifies unused content for cleanup.

RAG Query #6: Retrieve All Elements Within a Specific Topic

Introduction

Lists all elements inside a topic.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?element
WHERE {
  ?topic a dita:Topic .
  ?topic dita:hasElement ?element .
}
```

Business Value Supports fine-grained content auditing.

RAG Query #7: Retrieve Topics by Short Description (Shortdesc)

Introduction

Retrieves topics with a short description.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?shortdesc
```

```
WHERE {  
  ?topic a dita:Topic .  
  ?topic dita:shortdesc ?shortdesc .  
}
```

Business Value

Enhances search engine and teaser optimization.

RAG Query #8: Retrieve Maps Containing a Specific Topic

Introduction Finds maps where a specific topic appears.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?map  
WHERE {  
  ?map a dita:Map .  
  ?map dita:references <http://example.org/dita/topic1234> .  
}
```

Business Value

Tracks reuse footprint.

RAG Query #9: Retrieve Topics by Metadata (Audience, Platform)

Introduction

Fetches topics filtered by metadata.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?topic  
WHERE {  
  ?topic a dita:Topic .  
  ?topic dita:audience "developer" .  
}
```

Business Value

Enables persona-specific content extraction.

RAG Query #10: Retrieve All Key Definitions

Introduction

Retrieves all keydef entries.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?key ?keyName ?keyTarget
WHERE {
  ?key a dita:Keydef .
  ?key dita:keyName ?keyName .
  ?key dita:keyTarget ?keyTarget .
}
```

Business Value

Manages keys for reuse.

RAG Query #11: Retrieve Topics Filtered by Conditional Attribute (e.g., Product)

Introduction

Retrieve all topics that are conditionally tagged for a specific product line.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic a dita:Topic .
  ?topic dita:product "ProductA" .
}
```

```
}
```

Business Value

Enables conditional publishing by isolating product-specific content for customized outputs or content lifecycle management.

RAG Query #12: Retrieve Topics Missing a Title

Introduction

Identify topics missing required titles — often invalid content for delivery.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic a dita:Topic .
  FILTER NOT EXISTS { ?topic dita:title ?title . }
}
```

Business Value

Content without titles can break publications; finding and fixing these ensures content compliance and usability.

RAG Query #13: Detect Broken Topic References

Introduction

Find links to topics that no longer exist.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map ?brokenReference
```



```
WHERE {  
  ?map a dita:Map .  
  ?map dita:references ?brokenReference .  
  FILTER NOT EXISTS { ?brokenReference a dita:Topic . }  
}
```

Business Value

Prevents build errors and broken links by identifying outdated cross-references before publication.

RAG Query #14: List All Cross-References Between Topics

Introduction

List all internal links (cross-references) between topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?sourceTopic ?targetTopic  
WHERE {  
  ?sourceTopic dita:crossRef ?targetTopic .  
}
```

Business Value

Helps visualize topic relationships, critical for re-architecting information hierarchies and navigation paths.

RAG Query #15: Retrieve Topics Using a Specific Key

Introduction

Find all topics that use a particular key reference.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?topic
```

```
WHERE {
```

```
  ?topic a dita:Topic .
```

```
  ?topic dita:keyref "product-name" .
```

```
}
```

Business Value

Identifies content dependency on key definitions, helping with large-scale key refactoring or troubleshooting.

RAG Query #16: Retrieve Elements With a Specific Role (e.g., UI Elements)

Introduction

Find all elements tagged as UI labels (buttons, menus, etc.).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?element
```

```
WHERE {
```

```
  ?element dita:role "ui-label" .
```

```
}
```

Business Value

Supports UI documentation extraction, audit, and consistency checking across content deliverables.

RAG Query #17: Retrieve Reusable Content (Conrefs)

Introduction

Find all reusable content instances using conref attributes.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?topic ?reusedElement
```

```
WHERE {
```

```
  ?topic a dita:Topic .
```

```
  ?topic dita:conref ?reusedElement .
```

```
}
```

Business Value

Highlights reusable content assets, enabling improved reuse strategies and content efficiency measurement.

RAG Query #18: Retrieve Glossary Entries

Introduction

Pull all glossary entry topics (glossentry) for glossary generation or maintenance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?glossEntry ?term
```

```
WHERE {
```

```
  ?glossEntry a dita:GlossEntry .
```

```
  OPTIONAL { ?glossEntry dita:term ?term . }
```

```
}
```

Business Value

Maintains terminological consistency across your knowledge base and enhances search and information retrieval.

RAG Query #19: Retrieve Topics With External Links

Introduction

List all topics containing outbound external links (e.g., to websites).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic ?externalLink
WHERE {
  ?topic a dita:Topic .
  ?topic dita:externalLink ?externalLink .
}
```

Business Value

Facilitates link validation tasks, preventing broken or outdated external references that could damage credibility.

RAG Query #20: Retrieve Topics By Author Metadata

Introduction

Find topics authored by a specific person or team.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic a dita:Topic .
}
```

```
?topic dita:author "Jane Doe" .  
}
```

Business Value

Supports authorship tracking, contributor reporting, and performance metrics for documentation teams.

RAG Query #21: Retrieve Recently Modified Topics

Introduction

Find topics modified in a certain timeframe (if modification dates are stored).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>  
  
SELECT ?topic ?modifiedDate  
WHERE {  
  ?topic a dita:Topic .  
  ?topic dita:modified ?modifiedDate .  
  FILTER (?modifiedDate > "2025-01-01T00:00:00Z"^^xsd:dateTime)  
}
```

Business Value

Supports dynamic publishing or incremental review of newly created or updated content.

RAG Query #22: Retrieve Maps for a Specific Output Class (e.g., Training)

Introduction

Find DITA maps intended for a specific publishing output class.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?map
WHERE {
  ?map a dita:Map .
  ?map dita:outputclass "training" .
}
```

Business Value

Tailors content retrieval and publishing pipelines by deliverable type, improving audience targeting.

RAG Query #23: Retrieve Topics Using a Specific Metadata Value

Introduction

Retrieve topics with custom metadata fields, like "regulatory" for compliance.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic
WHERE {
  ?topic a dita:Topic .
  ?topic dita:customMetadata "regulatory" .
}
```

Business Value

Supports regulatory documentation audits, compliance reviews, and specialized publication filtering.

RAG Query #24: Retrieve Topics Sorted by Title Alphabetically

Introduction

Pull and sort all topic titles in alphabetical order.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?topic ?title
```

```
WHERE {
```

```
  ?topic a dita:Topic .
```

```
  ?topic dita:title ?title .
```

```
}
```

```
ORDER BY ASC(?title)
```

Business Value

Facilitates alphabetical indexes, search enhancements, and better UX in help systems.

RAG Query #25: Detect and List Duplicate Titles Across Topics

Introduction

Find multiple topics sharing the same title (potential duplicates).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?title (COUNT(?topic) AS ?count)
```

```
WHERE {
```

```
  ?topic a dita:Topic .
```

```
  ?topic dita:title ?title .
```

```
}
```

```
GROUP BY ?title
```

```
HAVING (COUNT(?topic) > 1)
```

Business Value

Helps identify possible duplication of issues leading to search confusion, SEO problems, or maintenance inefficiencies.

25 SPARQL Queries for Mining DITA Content Metrics

Metrics Query #1: Count Total Number of Topics

Introduction

Counts the total number of topics stored in the graph — a baseline content volume metric.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

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

Business Value

Measures total inventory size, an essential KPI for content production scalability and maintenance budgeting.

Metrics Query #2: Count Total Number of Maps

Introduction

Counts the total number of DITA maps available.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?map) AS ?totalMaps)
WHERE {
    ?map a dita:Map .
}
```

Business Value

Indicates the number of deliverables and supports product documentation portfolio tracking.

Metrics Query #3: Average Number of Topics per Map

Introduction

Calculates the average number of topics referenced per map.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (AVG(?count) AS ?averageTopicsPerMap)
WHERE {
  SELECT (COUNT(?topic) AS ?count)
  WHERE {
    ?map a dita:Map .
    ?map dita:references ?topic .
  }
  GROUP BY ?map
}
```

Business Value

Shows document size trends and complexity, helping optimize user comprehension and content modularity.

Metrics Query #4: Find Top 10 Most Reused Topics

Introduction

Finds the most heavily reused topics across all maps.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic (COUNT(?map) AS ?reuseCount)
```

```
WHERE {  
  ?map dita:references ?topic .  
}  
GROUP BY ?topic  
ORDER BY DESC(?reuseCount)  
LIMIT 10
```

Business Value

Measures reuse ROI; highly reused content saves time and cost across deliverables.

Metrics Query #5: Count Orphaned Topics (Not Referenced in Maps)

Introduction

Counts topics not linked in any map.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT (COUNT(?topic) AS ?orphanedTopics)  
WHERE {  
  ?topic a dita:Topic .  
  FILTER NOT EXISTS {  
    ?map dita:references ?topic .  
  }  
}
```

Business Value

Identifies content waste and opportunities for cleanup or repurposing.

Metrics Query #6: Identify Topics with Missing Short Descriptions

Introduction

Counts topics missing a shortdesc.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?topic) AS ?missingShortdesc)
WHERE {
  ?topic a dita:Topic .
  FILTER NOT EXISTS { ?topic dita:shortdesc ?shortdesc . }
}
```

Business Value

Shortdescs boost findability and scanability; missing ones signal UX improvement areas.

Metrics Query #7: Identify Topics Missing Titles

Introduction

Counts topics that lack a title.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?topic) AS ?missingTitles)
WHERE {
  ?topic a dita:Topic .
  FILTER NOT EXISTS { ?topic dita:title ?title . }
}
```

Business Value

Critical for ensuring publishable quality; content without titles risks broken builds and poor UX.

Metrics Query #8: Total Number of Conref Reuse Instances

Introduction

Counts how many conref reuse links exist across the corpus.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?conref) AS ?totalConrefs)
WHERE {
    ?conref a dita:Conref .
}
```

Business Value

Measures modular content health and reuse maximization, critical for efficient content scaling.

Metrics Query #9: Distribution of Topics by Type (Concept, Task, Reference)

Introduction

Breaks down the corpus by topic types.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?type (COUNT(?topic) AS ?count)
WHERE {
    ?topic a ?type .
    FILTER (?type IN (dita:Concept, dita:Task, dita:Reference))
}
```

```
}  
  
GROUP BY ?type
```

Business Value

Shows content model balance, supporting information architecture audits and optimization.

Metrics Query #10: Percentage of Topics Reused in More Than One Map

Introduction

Calculates the percentage of topics reused across multiple maps.

```
SPARQL Query  
  
PREFIX dita: <http://example.org/dita#>  
  
SELECT (COUNT(DISTINCT ?topic) AS ?reusedTopics)  
WHERE {  
  SELECT ?topic  
  WHERE {  
    ?map dita:references ?topic .  
  }  
  GROUP BY ?topic  
  HAVING (COUNT(DISTINCT ?map) > 1)  
}
```

Business Value

Higher reuse percentages mean faster time-to-market, lower cost-per-page, and better content ROI.

Metrics Query #11: Identify Topics with the Most Conditional Attributes

Introduction

Lists topics with the highest number of conditional attributes (e.g., audience, platform, product).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic (COUNT(?condition) AS ?conditionCount)
WHERE {
  ?topic a dita:Topic .
  ?topic ?condition ?value .
  FILTER(?condition IN (dita:audience, dita:platform, dita:product))
}
GROUP BY ?topic
ORDER BY DESC(?conditionCount)
```

Business Value

High conditionalization indicates complexity and potential fragility in content delivery; monitoring it improves robustness.

Metrics Query #12: Count Topics with Broken References

Introduction

Finds and counts topics containing links to non-existent targets.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?sourceTopic) AS ?brokenReferenceTopics)
WHERE {
```

```
?sourceTopic dita:crossRef ?targetTopic .  
FILTER NOT EXISTS { ?targetTopic a dita:Topic . }  
}
```

Business Value

Maintains content integrity and reduces the risk of broken links that frustrate users.

Metrics Query #13: Average Number of Cross-References per Topic

Introduction

Calculates average internal linking per topic.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT (AVG(?count) AS ?averageCrossRefs)  
WHERE {  
  SELECT (COUNT(?crossRef) AS ?count)  
  WHERE {  
    ?topic a dita:Topic .  
    OPTIONAL { ?topic dita:crossRef ?crossRef . }  
  }  
  GROUP BY ?topic  
}
```

Business Value

Indicates content interconnectedness — better navigation and contextual help systems come from healthy linking.

Metrics Query #14: Count of Glossary Entries (Glossentry Topics)

Introduction

Counts the glossary topics available.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (COUNT(?glossEntry) AS ?totalGlossaryEntries)
WHERE {
  ?glossEntry a dita:GlossEntry .
}
```

Business Value

Glossaries improve knowledge transfer, brand consistency, and product terminology quality.

Metrics Query #15: Average Topic Size (Elements per Topic)

Introduction

Measures average topic size by counting child elements.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT (AVG(?elementCount) AS ?averageElementsPerTopic)
WHERE {
  SELECT (COUNT(?element) AS ?elementCount)
  WHERE {
    ?topic a dita:Topic .
    ?topic dita:hasElement ?element .
  }
}
```

```
GROUP BY ?topic
```

```
}
```

Business Value

Supports optimization for modularity — smaller, focused topics are easier to maintain and reuse.

Metrics Query #16: Identify Topics Last Modified Over X Years Ago

Introduction

Finds outdated topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT ?topic
```

```
WHERE {
```

```
  ?topic a dita:Topic .
```

```
  ?topic dita:modified ?date .
```

```
  FILTER (?date < "2020-01-01T00:00:00Z"^^xsd:dateTime)
```

```
}
```

Business Value

Flags content needing review — prevents stale or obsolete information delivery.

Metrics Query #17: Identify Authors with Most Contributions

Introduction

Ranks authors by the number of topics they've created.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT ?author (COUNT(?topic) AS ?topicsAuthored)
WHERE {
  ?topic a dita:Topic .
  ?topic dita:author ?author .
}
GROUP BY ?author
ORDER BY DESC(?topicsAuthored)
```

Business Value

Supports resource planning, capacity metrics, and recognizing high-performing contributors.

Metrics Query #18: Identify Maps with the Highest Topic Counts

Introduction

Finds the largest deliverables (maps with most topics).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map (COUNT(?topic) AS ?topicCount)
WHERE {
  ?map a dita:Map .
  ?map dita:references ?topic .
}
GROUP BY ?map
ORDER BY DESC(?topicCount)
```

Business Value

Large maps may need modularization to improve usability and maintenance.

Metrics Query #19: Percentage of Topics With External Links

Introduction

Measures how much content points to external resources.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

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

Business Value

External links add richness but risk link rot; tracking rates helps manage this risk.

Metrics Query #20: Identify Topics with the Most Key References

Introduction

Finds topics making heavy use of keyref.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?topic (COUNT(?keyref) AS ?keyrefCount)
WHERE {
    ?topic a dita:Topic .
    ?topic dita:keyref ?keyref .
}
GROUP BY ?topic
```

```
ORDER BY DESC(?keyrefCount)
```

Business Value

Heavy keyref usage implies high dependency — must manage carefully to avoid cascading impacts.

Metrics Query #21: Calculate Reuse ROI Estimate

Introduction

Estimates content value saved through reuse (simple formula).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT (COUNT(?reusedTopic) AS ?reusedTopics)
```

```
WHERE {
```

```
  ?map dita:references ?reusedTopic .
```

```
}
```

(Post-process: Estimated Savings = ReusedTopics × Average Cost per Topic)

Business Value

Justifies investment in structured content and reuse architecture.

Metrics Query #22: Topics per Author (Average)

Introduction

Calculates the average topics produced per author.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT (AVG(?topics) AS ?averageTopicsPerAuthor)
```

```
WHERE {  
  SELECT (COUNT(?topic) AS ?topics)  
  WHERE {  
    ?topic a dita:Topic .  
    ?topic dita:author ?author .  
  }  
  GROUP BY ?author  
}
```

Business Value

Supports content team productivity analysis and planning.

Metrics Query #23: Detect Duplicate Titles Across Topics

Introduction

Lists potential duplicate topics based on shared titles.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>  
  
SELECT ?title (COUNT(?topic) AS ?count)  
WHERE {  
  ?topic a dita:Topic .  
  ?topic dita:title ?title .  
}  
GROUP BY ?title  
HAVING (COUNT(?topic) > 1)
```

Business Value

Duplicate titles confuse users and hurt SEO and navigation; finding them improves quality.

Metrics Query #24: Maps Containing Only One Topic

Introduction

Finds trivial maps (possibly candidates for merging).

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>

SELECT ?map
WHERE {
  {
    SELECT ?map (COUNT(?topic) AS ?topicCount)
    WHERE {
      ?map a dita:Map .
      ?map dita:references ?topic .
    }
    GROUP BY ?map
    HAVING (?topicCount = 1)
  }
}
```

Business Value

Improves documentation modularity and efficiency by identifying poorly structured deliverables.

Metrics Query #25: Average Age of Topics (Content Freshness)

Introduction

Calculates average modification age across all topics.

SPARQL Query

```
PREFIX dita: <http://example.org/dita#>
```

```
SELECT (AVG(xsd:integer(YEAR(NOW()) - YEAR(?modified))) AS ?averageAge)
WHERE {
  ?topic a dita:Topic .
  ?topic dita:modified ?modified .
}
```

Business Value

Demonstrates how fresh (or stale) your knowledge base is, enabling content lifecycle management.

Summary: Why DITA Knowledge Graphs Are the Future of Content Management

Technical content is evolving into a strategic business asset, yet managing it at scale remains challenging with traditional CMS and CCMS tools. As content libraries grow, manual audits, static reporting, and folder-based navigation models fail to keep pace with demands for quality, speed, and personalization.

A DITA Knowledge Graph revolutionizes content operations by:

- Modeling relationships, metadata, reuse, and governance rules semantically.
- Enabling dynamic, flexible, real-time queries across millions of topics.
- Revealing hidden risks, broken structures, and optimization opportunities.
- Allowing predictive analytics, trend detection, and strategic planning for content.

By adopting a knowledge graph approach, organizations move beyond document management toward living, intelligent content ecosystems that are scalable, auditable, and adaptive to business needs.

Appendix: How to Run These Queries in Your Environment

1. Prepare your DITA content

- Export DITA topics, maps, and metadata.
- Map DITA structures to RDF triples using a DITA-specific ontology.

2. Load into a Graph Database

- Use tools such as GraphWise GraphDB
- Import RDF-formatted DITA data for semantic storage.

3. Connect via SPARQL Endpoints

- Most graph databases provide built-in SPARQL query consoles.
- Optionally connect external analytics or reporting tools.

4. Run Queries

- Paste and execute SPARQL queries from this guide.
- Visualize results directly or export to CSV/JSON.

5. Automate and Monitor

- Schedule regular SPARQL checks for broken links, outdated content, reuse optimization, etc.
- Integrate query outputs into governance dashboards.

6. Expand Ontology Over Time

- Capture evolving metadata (e.g., taxonomy updates, audience segments, compliance tags) to enhance graph intelligence.

The construction of this document was assisted with generative AI.