

Tips gebruik linked data-omgeving data.bibliotheken.nl

versie 1.1 – 17 februari 2020

Op data.bibliotheken.nl worden beschrijvingen een belangrijk deel van de collectie van de KB volgens linked data-principes beschikbaar gesteld. Er kan door de data gebladerd worden, data kan in verschillende vormen gedownload worden en via de SPARQL-interface kan de data helpen bij het beantwoorden van allerlei onderzoeksvragen. Om goed met de data te kunnen werken is enig inzicht nodig in hoe de data georganiseerd is. Die organisatie van de data is het onderwerp van het eerste deel van dit document. In het tweede deel van dit document worden ter illustratie verschillende voorbeelden gegeven van hoe SPARQL ingezet kan worden om met de data vragen te beantwoorden.

Dit document verwacht van de gebruiker elementaire kennis van linked data en SPARQL.

Suggesties voor verbetering zijn welkom via dataservices@kb.nl.

I. De organisatie van de linked data in data.bibliotheken.nl

1. Expliciet onderscheid tussen dingen en beschrijvingen van die dingen

Binnen data.bibliotheken.nl wordt onderscheid gemaakt tussen *dingen*¹ die in het informatiesysteem beschreven worden enerzijds (meestal dingen ‘uit de echte wereld’, zoals bijvoorbeeld boeken, personen, onderwerpen) en *beschrijvingen* van die dingen anderzijds². In het algemeen heeft het informatiesysteem data.bibliotheken.nl de *dingen* zelf niet beschikbaar. Zo zal data.bibliotheken.nl onmogelijk de 44^e druk van ‘De Aanslag’ van Harry Mulisch kunnen leveren, en natuurlijk Harry Mulisch zelf al helemaal niet. Het systeem kan *wel* een beschrijving van die dingen leveren.

Vanuit een praktisch perspectief is het onderscheid tussen het ding en de beschrijving van het ding belangrijk. Helder moet zijn of een eigenschap in een linked data *triple* betrekking heeft op het *ding* of op de *beschrijving* van het ding. Het maakt bijvoorbeeld veel verschil of een boek een *public domain* –licentie heeft of dat de beschrijving (in essentie de *metadata*) van dat boek een *public domain* –licentie heeft.

Hoewel de beschreven dingen doorgaans zelf geen onderdeel uitmaken van het informatiesysteem hebben ze wel een URI die het ding in kwestie representeert. De beschrijving van het ding heeft een eigen URI. De URI’s van het ding en de bijbehorende beschrijving lijken in data.bibliotheken.nl veel op elkaar. Zo heeft het boek met URI <<http://data.bibliotheken.nl/id/nbt/p357539168>> als beschrijving de URI <<http://data.bibliotheken.nl/doc/nbt/p357539168>>.

Bij het opvragen van het ding zal data.bibliotheken.nl altijd via een automatisch *redirect* doorverwijzen naar de bijbehorende beschrijving van het ding.

Ook semantisch zijn dingen en bijbehorende beschrijvingen aan elkaar gekoppeld. De wijze waarop wordt geïllustreerd met deze *triple*³:

```
<http://data.bibliotheken.nl/id/nbt/p357539168> foaf:isPrimaryTopicOf <http://data.bibliotheken.nl/doc/nbt/p357539168> .
```

Doorgaans biedt het systeem ook de bijbehorende inverse relatie :

```
<http://data.bibliotheken.nl/doc/nbt/p357539168> foaf:primaryTopic <http://data.bibliotheken.nl/id/nbt/p357539168> .
```

waarbij foaf:isPrimaryTopicOf de owl:inverseOf van foaf:primaryTopic is.

De eigenschappen van beschrijvingen van dingen:

In lijn met het voorgaande zijn de eigenschappen (predicaat en object) van de *beschrijving* van een ding eenvoudig te selecteren met:

```
?URI_ding foaf:isPrimaryTopicOf ?beschrijving .  
?beschrijving ?predicaat ?object .
```

Door technische beperkingen in de applicatie van data.bibliotheken.nl geeft variabele ?beschrijving met de huidige opzet van de data niet de verwachte /doc-URI (maar een zogenaamde *blank_node* identifier). De ‘officiële’ /doc-URI van de beschrijving kan opgevraagd worden met

```
?URI_ding foaf:isPrimaryTopicOf/owl:sameAs ?URI_beschrijving .
```

-
- 1 Voor de term ‘ding’ kan ook een andere abstracte term gelezen worden, zoals bijvoorbeeld entiteit, object of resource.
 - 2 Inderdaad, een beschrijving is ook een ding (entiteit, object, etc.) maar de essentie van dit type ding is dat het een ander ding *beschrijft*, vandaar dat hier gesproken wordt over *dingen* versus *beschrijvingen*.
 - 3 In dit document worden gangbare *namespace*-prefixes gebruikt, gebruik bijvoorbeeld <http://prefix.cc/> om bijbehorende namespace URI te achterhalen.

Zie ook *deel II - voorbeeld 2*, voor het werken met eigenschappen van beschrijvingen.

2. Datasets bevatten de beschrijvingen, niet de dingen

De verschillende collecties in de linked data-omgeving worden gepresenteerd op de homepage <http://data.bibliotheken.nl/>. Deze collecties worden semantisch aangeduid als void:Dataset (en doorgaans ook als schema:Dataset). Door via de homepage op de URI van een dataset te klikken volgt er een beschrijving van de inhoud van de dataset.

Bij de organisatie van de data wordt het semantisch zuivere principe gevolgd dat in datasets *beschrijvingen van dingen* zitten, en dus niet de dingen zelf. Dus de 44e druk van “De Aanslag” van Harry Mulisch zit *niet* in de dataset “Nederlandse Bibliografie Totaal” (NBT), maar de *beschrijving* van die publicatie *wel*. Beschrijvingen worden via het void:inDataset *predicaat* aan een dataset gekoppeld. Via de in het voorgaande beschreven relatie tussen dingen en beschrijvingen is het met één extra stap mogelijk de publicaties die bij een dataset horen op te vragen.

Zo selecteert:

```
?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
```

alle publicaties die behoren tot de dataset NBT. Zie ook *deel II - voorbeeld 1*, voor het werken met datasets.

3. Relaties, intern en extern

De datasets en de daaraan gekoppelde publicaties overlappen deels met elkaar. Zo zou een publicatie die beschreven is in de dataset NBT ook beschreven kunnen zijn in de dataset “Short-Title Catalogue Netherlands” (STCN) óf in de set met titels van de Digitale Bibliotheek van de Nederlandse Letteren (“Titels DBNL”). Hoewel het streven is om één publicatie in het systeem maar één URI te geven, zal zo’n publicatie nu in iedere dataset een eigen URI hebben. Deze URI’s zijn in de huidige versie van de data nog niet met elkaar verbonden.

De linked data-omgeving biedt data afkomstig uit verschillende thesauri. Voor onderwerps- of genre-ontsluiting zijn dit de Brinkman, de Gemeenschappelijke Trefwoordenthesaurus en de wat oudere Systematische KBCode (volg de links op de homepage van <http://data.bibliotheken.nl> voor enige uitleg). Bedenk dat deze thesauri in verschillende periodes (die soms deels overlappen) gebruikt zijn bij het catalogiseren en dat ze volgens verschillende principes georganiseerd zijn. Er zijn in de huidige versie van de data geen verbindingen gemaakt tussen gerelateerde termen uit de verschillende thesauri.

In aanvulling op genoemde thesauri bevat data.bibliotheken.nl organisaties (als schema:Organization) afkomstig uit de corporatiethesaurus van de KB en personen (als schema:Person) uit de Nederlandse Thesaurus Auteursnamen (NTA) én uit de auteursthesaurus van DBNL. Let op dat in de huidige opzet ieder pseudoniem als een aparte persoon is opgenomen (nu nog zonder onderlinge link). Wél gekoppeld (via owl:sameAs) zijn de personen uit de auteursthesaurus van DBNL met de equivalenten uit de NTA⁴, bijvoorbeeld:

```
<http://data.bibliotheken.nl/id/thes/p068332556> owl:sameAs <http://data.bibliotheken.nl/id/dbnla/schl039> .
```

Aangezien de personen uit de auteursthesaurus van DBNL in sommige aspecten rijker beschreven zijn dan de personen uit de NTA kan nuttig van deze link gebruik gemaakt worden. Zo biedt DBNL anders dan de NTA wel het gender (via schema:gender) van de persoon en wordt het zo toch mogelijk om in de NBT-set publicaties te selecteren die geschreven zijn door bijvoorbeeld een man. Zie ook *deel II* voor een voorbeeld.

Bij de op de KB-thesauri gebaseerde entiteiten wordt soms ook in externe links voorzien. Specifiek gaat het in de huidige opzet van de data om:

- owl:sameAs-links naar VIAF voor NTA-personen
- owl:sameAs-links naar wikidata voor NTA-personen
- owl:sameAs-links naar dbpedia voor NTA-personen
- SKOS-relaties van Brinkman-concepten naar de Gemeenschappelijke Thesaurus Audiovisuele Archieven van het Instituut voor Beeld en Geluid

4 De NTA-URI kan als de *canonical* ofwel voorkeurs-URI beschouwd worden.

4. Vocabulaires, ontologie en duurzaamheid

Bij het beschrijven van de entiteiten in data.bibliotheken.nl is er naar gestreefd om zo veel als praktisch zinvol gebruik te maken van het schema.org-vocabulaire⁵. In sommige gevallen waar een schema.org-oplossing mogelijk was is desondanks voor een ander vocabulaire gekozen vanwege een betere aansluiting bij heersende *best practices*. Een voorbeeld hiervan is de foaf:isPrimaryTopicOf-link tussen ding en beschrijving.

Bij het gebruik van schema.org worden ook de schema.org-ontologie en de schema.org-beperkingen overgenomen. Zo stelt bijvoorbeeld de definitie van schema:publication (<<http://schema.org/publication>>) dat waarden (dwz het *object* van de *subject*, *predicaat*, *object-triple*) van het type schema:PublicationEvent zijn en het beschreven type (het *subject*) een schema:CreativeWork. Deze restricties gelden dus ook binnen data.bibliotheken.nl. Bedenk wel dat, aangezien de achterliggende *triplestore* niet automatisch triples genereert op basis van deze definities⁶, een query op entiteiten van het type schema:PublicationEvent zal alleen de resultaten geven die *expliciet* als zodanig *triple* in het systeem geladen zijn.

De KB werkt er aan haar bibliografische metadata op te zetten volgens de ontologie van IFLA LRM⁷ en het daarop gebaseerde vocabulaire van RDA⁸. In de huidige opzet zijn er in de linked data-omgeving nog geen beschrijvingen in het RDA-vocabulaire beschikbaar. Wél kunnen de belangrijkste entiteitstypen *impliciet* ook als een equivalent van een LRM-entiteit gezien worden. Dikwijls is die relatie vanzelfsprekend. Zo kan een schema:Person ook als een LRM:Person gezien worden. Bibliografische entiteiten, in data.bibliotheken.nl getypeerd als bijvoorbeeld een schema:Book of een schema:Periodical, kunnen hier als equivalent of verbijzondering van een LRM:Manifestation gezien worden. Beschrijvingen van het type LRM:Item (zoals ze nu bijvoorbeeld voorkomen in de beschrijvingen van de STCN) worden in data.bibliotheken.nl aanvullend getypeerd met het type schema:IndividualProduct. In toekomstige versies van de data zullen ook (equivalenten van) LRM:Work- en LRM:Expression-entiteiten opgenomen worden.

De KB streeft naar persistente URI's maar de beschrijvingen zelf kunnen in nieuwere releases van de data veranderen. Hou er rekening mee dat de dingen in de toekomst meerdere beschrijvingen kunnen gaan krijgen, waarbij iedere beschrijving een eigen URI krijgt. Denk bijvoorbeeld aan een beschrijving gebaseerd op RDA, in aanvulling op de op schema.org-gebaseerde beschrijving.

De beschrijvingen in de verschillende datasets zijn in hoofdlijnen op dezelfde manier opgebouwd. Het streven naar uniformiteit is niet altijd vervuld doordat de bron van iedere dataset eigen kenmerken en eigenaardigheden heeft of doordat voortschrijdend inzicht nog niet overall met terugwerkende kracht doorgevoerd is. In de huidige versie van de data zijn met name de beschrijvingen van de dataset STCN anders opgebouwd dan wat in de overige sets te vinden is. De grotere complexiteit van de beschrijvingen van de STCN-entiteiten is het gevolg van de wens om recht te kunnen doen aan de rijkheid van brondata van de STCN. In deel twee van dit document worden daarom enige STCN-gerelateerde voorbeeldqueries apart behandeld.

Belangrijke wijziging in de data zullen in de toekomst in dit document bijgehouden worden. Doorgaans zal de actualiteit van een dataset in de dataset-beschrijving te vinden zijn via de eigenschap schema:dateModified.

5 <http://schema.org/>

6 Met andere woorden, de *triplestore* doet niet aan *inferencing*.

7 <https://www.ifla.org/publications/node/11412>

8 <https://www.rdatoolkit.org/>

II. Voorbeeldvragen en SPARQL-oplossingen

Dit tweede deel van de handleiding illustreert hoe via SPARQL de linked data van data.bibliotheken.nl bevestigd kan worden. Uit praktische overwegingen worden de queries waar zinvol standaard van de beperking LIMIT 100 voorzien. Pas op bij het weglaten van deze beperking dat vooral queries die veel resultaten opleveren een zeer zware belasting voor server, client en dataverbinding kunnen opleveren.

De interface voor het uitvoeren van SPARQL op data.bibliotheken.nl is te vinden onder <http://data.bibliotheken.nl/sparql>

Voorafgaande aan specifieke queries voor de data in data.bibliotheken.nl volgen een paar generieke voorbeelden voor het verkennen van linked data door middel van SPARQL.

0. Generieke queries voor het verkennen van linked data met SPARQL

Via de homepage van data.bibliotheken.nl kan de data al bladerend verkend worden. Dit kan een aardige eerste indruk geven van wat er beschikbaar is maar voor beter inzicht in wat de data biedt is het zinvol om via SPARQL wat kenmerken van de data in beeld te krijgen. Hier volgende enige handige startpunten. Gezien de generieke aard van de queries wordt er hooguit beperkte uitleg gegeven ten aanzien van de syntax of de werking van de query. Zie hiervoor een goede basishandleiding voor linked data en SPARQL.

Vraag: welke typen entiteiten zijn er aanwezig?

```
select distinct ?type where {  
  [] a ?type .  
}
```

[[bekijk: query, resultaat](#)]

In dit voorbeeld worden alleen de expliciet gedefinieerde entiteiten teruggegeven, niet de entiteitstypen die indirect afgeleid zouden kunnen worden (via *inferencing*).

Vraag: wat zijn de meest voorkomende typen entiteiten, gesorteerd naar voorkomen, die voorzien zijn van URI?

Sommige entiteiten zijn enkel een blank_node en hebben dus geen URI. Die filteren we hier weg. We sorteren naar frequentie van voorkomen.

```
select ?type count (?type) as ?aantal where {  
  ?s a ?type .  
  FILTER(isURI(?s))  
} group by ?type order by desc (?aantal)
```

[[bekijk: query, resultaat](#)]

Vraag: welke predicaten worden voor een specifiek type entiteit toegepast?

In dit voorbeeld worden als type entiteit schema:Book gebruikt:

```
select distinct ?predicaat where {  
  ?subject a schema:Book .  
  ?subject ?predicaat [] .  
}
```

[[bekijk: query, resultaat](#)]

Vraag: Welke mogelijke waarden komen voor bij een specifiek predicaat?

In dit voorbeeld wordt als predicaat schema:genre gebruikt:

```
select distinct ?object where {  
  [] schema:genre ?object .  
} limit 100
```

[[bekijk: query, resultaat](#)]

Vraag: Welke entiteiten hebben een eigenschap die lijkt op een nader in te vullen patroon?

We gaan hier ter illustratie met een FILTER en een reguliere expressie op zoek naar entiteiten die 'Wolkers' in schema:name bevatten. De 'i' bij de regex-aanroep betekent *case insensitive*:

```
select ?subject ?object where {
  ?subject schema:name ?object .
  FILTER(regex(?object, 'Wolkers', 'i'))
} limit 100
```

[*bekijk:* [query](#), [resultaat](#)]

1. Alle publicaties uit de dataset NBT:

Recept

Zoals beschreven in deel I zitten er in een dataset geen publicaties maar wel de beschrijvingen van publicaties. Omdat beschrijvingen en dingen aan elkaar gekoppeld zijn kunnen we in twee stappen alle URI's van publicaties uit een dataset selecteren.

De URI van de dataset NBT vinden we via de homepage van data.bibliotheken.nl.

Oplossing

```
select ?publicatie where {
  ?publicatie foaf:isPrimaryTopicOf ?beschrijving .
  ?beschrijving void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
} limit 100
```

[*bekijk:* [query](#), [resultaat](#)]

Discussie

Met behulp van zogenaamde *property paths* is voorgaande oplossing ook wel korter te noteren als:

```
select ?publicatie where {
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
} limit 100
```

[*bekijk:* [query](#), [resultaat](#)]

2. Alle eigenschappen van de beschrijving van een publicatie:

Recept

De eigenschappen van de beschrijving van een ding zijn niet direct via de URI van de beschrijving op te vragen. De eigenschappen zijn te selecteren via de entiteit die een owl:sameAs-relatie heeft tot de URI van de beschrijving. In dit voorbeeld vragen we om de eigenschappen van de beschrijving <<http://data.bibliotheken.nl/doc/nbt/p357539168>> .

Oplossing

```
select ?predicaat ?object where {
  ?s ?predicaat ?object .
  ?s owl:sameAs <http://data.bibliotheken.nl/doc/nbt/p357539168> .
}
```

[*bekijk:* [query](#), [resultaat](#)]

Discussie

Omdat de triplestore niet ingesteld is om het zogenaamde *inferencing* toe te passen werkt de volgende oplossing, die semantisch wel correct is, nu **niet** binnen data.bibliotheken.nl:

```
select ?predicaat ?object where {
  <http://data.bibliotheken.nl/doc/nbt/p357539168> ?predicaat ?object .
}
```

[*bekijk:* [query](#), [resultaat](#)]

3. Titels betreffende poëzie uit de dataset “Titels DBNL” :

Recept

De aanpak voor publicaties uit de dataset “Titels DBNL” is in voorgaande te vinden. We willen ons beperken tot het genre “poëzie”. Dit wordt in DBNL als tekstwaarde vastgelegd bij schema:genre. Tot slot willen we als resultaat niet de URI van de publicatie maar de titel, hiervoor wordt schema:name gebruikt.

Oplossing

```
select ?titel where {
  ?publicatie schema:name ?titel .
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/dbnlt> .
  ?publicatie schema:genre "poëzie" .
} limit 100
```

[[bekijk: query](#), [resultaat](#)]

Discussie

Hoe weten we dat in deze dataset schema:genre op deze manier gebruikt wordt? We kunnen via data.bibliotheken.nl naar de beschrijving van de dataset (<http://data.bibliotheken.nl/doc/dataset/dbnlt>) en daar dan rondbladeren, om te beginnen via de void:exampleResource. Een betere oplossing is om aan triplestore zelf te vragen wat alle voorkomende predicaten zijn in deze dataset:

```
select distinct(?predicaat) where {
  ?publicatie schema:name ?titel .
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/dbnlt> .
  ?publicatie ?predicaat [] .
} limit 100
```

[[bekijk: query](#), [resultaat](#)]

Als we vervolgens een beeld willen krijgen van het gebruik van schema:genre kunnen we dat met deze query doen:

```
select distinct(?genre) where {
  ?publicatie schema:name ?titel .
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/dbnlt> .
  ?publicatie schema:genre ?genre .
} limit 100
```

[[bekijk: query](#), [resultaat](#)]

Let op dat in andere datasets schema:genre ook een URI in plaats van een letterlijke waarde (datatype ‘string’) gevonden kan worden.

4. Publicaties van een zekere auteur:

Recept

Voor het aangeven van de auteur wordt schema:author gebruikt. Het opvragen van publicaties door bijvoorbeeld Harry Mulisch (<<http://data.bibliotheken.nl/id/thes/p06854796X>>) kan zo heel eenvoudig

Oplossing

```
select ?publicatie where {
  ?publicatie schema:author <http://data.bibliotheken.nl/id/thes/p06854796X> .
} limit 100
```

[[bekijk: query](#), [resultaat](#)]

Discussie

Hoe weten we dat <<http://data.bibliotheken.nl/id/thes/p06854796X>> de URI is van Harry Mulisch? De linked data-omgeving heeft geen regulier zoekveld maar ook met SPARQL kan er tekstueel gezocht worden. In dit geval zoeken we naar een persoon (schema:Person) waarbij in de schema:name “Mulisch” voorkomt. Dit doen we met een FILTER en een reguliere expressie in SPARQL. Van de gevonden personen zien we graag de naam en de URI:

```

select ?s ?naam where {
  ?s a schema:Person .
  ?s schema:name ?naam .
  FILTER(regex(?naam, 'Mulisch', 'i'))
} limit 100

```

[[bekijk: query, resultaat](#)]

Bovenstaande query laat zien dat er twee URI's zijn voor Harry Mulisch. Harry Mulisch is namelijk zowel in de dataset <http://data.bibliotheken.nl/id/dataset/persons> beschreven als in <http://data.bibliotheken.nl/id/dataset/dbnla>. Het opvragen van <http://data.bibliotheken.nl/id/dbnla/muli002> laat zien er tussen beiden via owl:sameAs een relatie is gelegd.

5. Publicaties in de NBT-dataset uit een specifiek genre geschreven door een vrouw:

Recept

Binnen de NBT-dataset wordt gebruik gemaakt van persoonbeschrijvingen afkomstig uit de NTA (dataset <http://data.bibliotheken.nl/id/dataset/persons>). Deze beschrijvingen bevatten geen gender-informatie maar deze kan wel gevonden worden via de owl:sameAs koppeling met de personen uit de DBNL-set (<http://data.bibliotheken.nl/id/dataset/dbnla>). De overige deelvragen zijn in het voorgaande behandeld.

Binnen de NBT-set wordt genre-informatie niet als een string – zoals bij de Titels van DBNL – maar als skos:Concept (via een URI) opgenomen. Hier is als genre <http://data.bibliotheken.nl/id/thes/p075629402> gekozen.

Oplossing

```

select * where {
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
  ?publicatie schema:author/owl:sameAs/schema:gender <http://schema.org/Female> .
  ?publicatie schema:genre <http://data.bibliotheken.nl/id/thes/p075629402> .
}

```

[[bekijk: query, resultaat](#)]

Discussie

Een getrapte aanpak als in het voorgaande heeft als nadeel dat het resultaat alleen voldoende ‘recall’ oplevert als er voldoende koppelingen tussen beide sets met schema:Person –entiteiten zijn gelegd (en uiteraard, als er ook voldoende gender-informatie vast gelegd is). Er is nu niet bekend wat de kwaliteit en dekingsgraad van deze koppelingen is.

Het voorbeeld legt de probleem aan de dag met de opzet van genre- (of vorm-) informatie in de bronthesaurus, de Brinkman. De Brinkman maakt gebruik van samengestelde trefwoorden. Hierbij is er bijvoorbeeld niet een enkelvoudig vormtrefwoord “Reisgidsen” maar worden deze altijd samengesteld gebruikt met een andere term, dus bijvoorbeeld “Duitsland; reisgidsen” en “België; reisgidsen” of zoals in dit voorbeeld “romans en novellen; oorspr. Nederlands”. In de toekomst zullen deze termen van elkaar gescheiden worden zodat er eenvoudig met één enkelvoudig vormconcept naar alle reisgidsen gezocht kan worden. In de huidige versie van de data is het mogelijk om alle reisgidsen geschreven door een vrouw op deze manier op te vragen:

```

select * where {
  ?publicatie foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
  ?publicatie schema:author/owl:sameAs/schema:gender <http://schema.org/Female> .
  ?publicatie schema:genre/skos:prefLabel ?genrelabel .
  FILTER(regex(?genrelabel, 'Reisgidsen', 'i'))
}

```

[[bekijk: query, resultaat](#)]

6. Aantal publicaties over een natuurkundig onderwerp volgens de Brinkman-thesaurus:

Recept

Het is eenvoudig om via ?s schema:about <http://data.bibliotheken.nl/doc/thes/p075616289> publicaties over <http://data.bibliotheken.nl/doc/thes/p075616289> (“Natuurkunde”) te vinden. Als we ook publicaties over specifiekere, onderliggende natuurkundige onderwerpen willen weten dan kunnen we die via zogenaamde *property paths* in SPARQL eenvoudig in één keer opvragen:

Oplossing

```
select count(?s) as ?aantal where {  
  ?s schema:name ?titel .  
  ?s schema:about/skos:broader* <http://data.bibliotheken.nl/id/thes/p075616289> .  
}
```

[*bekijk: [query](#), [resultaat](#)*]

Discussie

Met bovenstaande aanpak worden titels geselecteerd waarvoor geldt:

```
?s schema:about <http://data.bibliotheken.nl/doc/thes/p075616289> .
```

maar ook

```
?s schema:about/skos:broader <http://data.bibliotheken.nl/doc/thes/p075616289> .
```

en

```
?s schema:about/skos:broader/skos:broader <http://data.bibliotheken.nl/doc/thes/p075616289> .
```

etcetera.

7. Publicaties uit een specifiek jaar:

Recept

De publicatiedatum wordt in schema.org als schema:startDate vastgelegd waarbij de waarde normaliter een jaar is. In dit voorbeeld kiezen we jaar "1791".

Oplossing

```
select ?s where {  
  ?s schema:publication/schema:startDate "1791".  
}
```

[*bekijk: [query](#), [resultaat](#)*]

Discussie

Een publicatiejaar is niet altijd precies bekend. Als schema:startDate kan ook een waarde als "179X" voorkomen, dus een datum in het laatste decennium van de 18^e eeuw.

Om de onzekerheid over het jaar van publicatie aan te geven zijn binnen de STCN-set in het schema:publicationEvent ook de properties sem:hasLatestEndTimeStamp en sem:hasEarliestBeginTimeStamp beschikbaar.

8. Top 10 auteurs met de meeste publicaties in dataset NBT:

Recept

De oplossing voor deze vraag is redelijk recht-toe-recht-aan en gebruikt de group by en count-functionaliteit van SPARQL om per auteur te kunnen tellen.

Oplossing

```
select ?naam count(?naam) as ?aantal where {  
  ?s foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .  
  ?s schema:author ?author .  
  ?author a schema:Person .  
  ?author schema:name ?naam .  
} group by ?author ?naam order by desc (?aantal) limit 10
```

[*bekijk: [query](#), [resultaat](#)*]

Discussie

Er wordt hier expliciet om het type schema:Person gevraagd, we willen andersoortige auteurs (specifiek: schema:Organization) uitsluiten. Het is verleidelijk om de group by direct op ?naam te doen en ?author daar weg te laten maar dat levert ongewenste resultaten doordat auteurs met dezelfde naam gecombineerd worden. In praktijk blijkt er helaas regelmatig een lege naam ("") opgenomen te zijn. Bedenk ook dat datgene wat hier per auteur geteld wordt in LRM terminologie *Manifestations* zijn; het zegt dus weinig over het aantal Werken (Works), vertalingen, e.d. Een ruwe indicatie van het aantal werken van bijvoorbeeld Harry Mulisch kan verkregen worden met onderstaande clustering van Nederlandstalige titels:


```

select ?titel count(?titel) as ?aantal where {
  ?s foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/nbt> .
  ?s a schema:Book .
  ?s schema:author <http://data.bibliotheken.nl/id/thes/p06854796X> .
  ?s schema:name ?titel .
  ?s schema:inLanguage "ned" .
} group by ?titel order by desc(?aantal)

```

[[bekijk: query, resultaat](#)]

9. Top 10 auteurs met de meeste publicaties in dataset STCN:

Recept

In de kern is de oplossing gelijk aan de voorgaande. De modellering van auteurs in de STCN gebeurt echter op een andere wijze. Anders dan in de overige datasets zijn de publicaties via een tussenentiteit aan de auteurs gekoppeld. Deze tussenentiteit (een schema:Role) wordt gebruikt om de naam vast te leggen die de auteur gebruikt *in de context van de betreffende publicatie*. Met andere woorden, dankzij de schema:Role tussenentiteit kan er voor één auteur (schema:Person) vastgelegd worden onder welke naam, bijvoorbeeld onder welk pseudoniem, een publicatie is uitgegeven.

Oplossing

```

select ?naam count(?naam) as ?aantal where {
  ?s foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/stcn> .
  ?s schema:author/schema:author ?author .
  ?author a schema:Person .
  ?author schema:name ?naam .
} group by ?author ?naam order by desc (?aantal) limit 10

```

[[bekijk: query, resultaat](#)]

Discussie

Binnen de schema:Role (via ?s schema:author ?role) is meer dan alleen de hier gebruikte schema:name te vinden. Via een pnv:hasName⁹ relatie is er meer detailinformatie over de gebruikte auteursnaam te vinden. Zo geeft daar predicaat pnv:nameSpecification aan of de gebruikte naam een pseudoniem (waarde “*pseudonym*”) of een mystificatie (waarde “*mystification*”) betreft. We zouden de originele vraag zo bijvoorbeeld nader kunnen beperken tot de top 5 van auteurs die een pseudoniem hebben gebruikt:

```

select ?naam count(?naam) as ?aantal where {
  ?s foaf:isPrimaryTopicOf/void:inDataset <http://data.bibliotheken.nl/id/dataset/stcn> .
  ?s schema:author ?role .
  ?role pnv:hasName/pnv:nameSpecification "pseudonym" .
  ?role schema:author ?author .
  ?author a schema:Person .
  ?author schema:name ?naam .
} group by ?author ?naam order by desc (?aantal) limit 5

```

[[bekijk: query, resultaat](#)]

9 Bijbehorende *namespace* URI is <<https://w3id.org/pnv#>> .