

Tutorial: come scrivere query SPARQL semplici

In questo tutorial vedremo come costruire una nuova query SPARQL a partire dalla documentazione fornita in questa sezione.

Acronimi della guida in linea e definizioni:

- ✓ **c** (CLASSI): rappresentano gruppi di individui accomunati dalla condivisione di alcune proprietà. Sono le entità principali descritte dall'ontologia e possono essere considerate al pari delle entità di un modello E-R.
- ✓ **ni** (NAMED INDIVIDUALS): sono anche chiamate istanze delle classi. Possiedono delle proprietà che li relazionano con altre istanze.
- ✓ **op** (OBJECT PROPERTIES): sono le relazioni tra le istanze di due classi (predicati).
- ✓ **dp** (DATA PROPERTIES): rappresentano gli attributi (testuali o numerici) che descrivono le classi, cioè i singoli campi di dato.

Il linguaggio SPARQL prevede diverse tipologie di interrogazione e molti gradi di complessità. Condivide molte parole chiave con il linguaggio SQL ma la logica di costruzione è molto diversa perché la struttura dati interrogata non è un database bensì un grafo.

Per questo tutorial verrà presa in esame un'interrogazione di tipo SELECT per un caso di complessità medio-bassa e, per semplificare si farà uso di analogie con il mondo dei database relazionali.

Supponiamo di voler costruire la query per estrarre questo dato:

“Capienze totali degli asili nido in zona ‘g’ divise per anno scolastico.”

Vediamo quindi quali sono i passaggi che ci permettono di arrivare alla scrittura della query e al risultato desiderato:

- 1) Partiamo dalla classe “Caratteristiche (Scuola)” e vediamo quali informazioni ci servono per costruire la query.

Caratteristiche (Scuola)^C [torna a IdC or Classi IdC](#)

IRI: http://dati.comune.milano.it:8890/schemas/ontomi:scuola_caratteristiche

Insieme di dati costituenti le caratteristiche delle scuole

è definito da
<http://dati.comune.milano.it:8890/schemas/ontomi/>

è nel dominio di
[anno scolastico](#) ^{dp}, [capienza](#) ^{dp}, [classe eta](#) ^{dp}, [codice della unita educativa](#) ^{dp}, [codice scuola](#) ^{dp}, [flag paritaria](#) ^{dp}, [grado della scuola](#) ^{dp}, [ha gestione scuola](#) ^{op}, [ha intervallo temporale](#) ^{op}, [ha scuola anagrafica](#) ^{op}, [id caratteristiche scuola](#) ^{dp}, [id struttura erogazione](#) ^{dp}, [nome della scuola](#) ^{dp}, [tipo di struttura](#) ^{dp}

è nel codominio di
[anagrafica scuola per](#) ^{op}, [gestione scuola per](#) ^{op}, [intervallo temporale per](#) ^{op}

Il nome dell'entità così come è memorizzata sul server è quello riportato in fondo all'indirizzo IRI (come evidenziato in rosso sull'immagine).

I campi dato contenuti nell'entità sono quelli con apice “**dp**” indicati nell'elenco “è nel dominio di”.

I campi con apice “**op**” indicati nell’elenco “è nel codominio di” sono i collegamenti della tabella `scuola_caratteristiche` con le altre.

- 2) Scopriamo dunque che i campi che desideriamo in output sono tutti all’interno di questa classe e sono *anno scolastico* e *capienza*. Cliccando sui rispettivi collegamenti si salta alla descrizione di questi per recuperare i nomi reali con il quale sono memorizzati a sistema e che dovremo usare nella query (`anno_scolastico` e `capienza`).

anno scolastico ^{dp}	torna a IdC or Data Property IdC
IRI: http://dati.comune.milano.it:8890/schemas/ontomi/anno_scolastico	
Periodo dell'anno in cui un'universita(oppure un altro istituto competente in materia di istruzione) svolge la propria attivita, in allineamento alle normative vigenti.	
è definito da http://dati.comune.milano.it:8890/schemas/ontomi/	
ha dominio Alunno (Scuola) ^c Caratteristiche (Scuola) ^c	
ha codominio string	

capienza ^{dp}	torna a IdC or Data Property IdC
IRI: http://dati.comune.milano.it:8890/schemas/ontomi/capienza	
Numero che identifica la capienza massima di persone per la struttura	
è definito da http://dati.comune.milano.it:8890/schemas/ontomi/	
ha dominio Caratteristiche (Scuola) ^c	
ha codominio int	

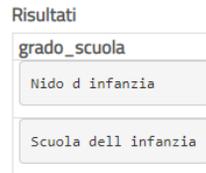
- 3) Sempre all’interno della classe `Caratteristiche (Scuola)` è presente il campo `grado` della `scuola` che presumibilmente contiene la tipologia di scuola (`scuola dell’infanzia` o `nido`). Clicchiamo sul campo per saltare alla sua descrizione e recuperare il nome a sistema (`grado_scuola`).

grado della scuola ^{dp}	torna a IdC or Data Property IdC
IRI: http://dati.comune.milano.it:8890/schemas/ontomi/grado_scuola	
Descrizione del grado della scuola	
è definito da http://dati.comune.milano.it:8890/schemas/ontomi/	
ha dominio Caratteristiche (Scuola) ^c	
ha codominio string	

- 4) Dal momento che vogliamo ottenere la capienza per i soli asili nido nella query occorrerà specificare un filtro su questo campo. Per verificare a sistema quali sono i valori possibili di questo campo è possibile eseguire questa query semplice:

```
prefix ontomi: http://dati.comune.milano.it:8890/schemas/ontomi/
select
  distinct ?grado_scuola
where{
  ?sc a ontomi:scuola_caratteristiche.
  ?sc ontomi:grado_scuola ?grado_scuola.
}
```

- 5) Il sistema restituisce questo risultato:



Il filtro che andrà a comporre la query sarà quindi: `grado_scuola = 'Nido d infanzia'`

- 6) Riepilogando, rispetto all'estrazione desiderata per adesso abbiamo recuperato:
- Campi di output: `anno_scolastico` e `capienza`
 - Tabelle sorgenti: `scuola_caratteristiche`
 - Filtri: `grado_scuola = 'Nido d infanzia'`

L'ultimo filtro che occorre applicare è quello della zona. Sulla classe `scuola_caratteristiche` questa informazione non è presente. Occorre quindi ricercarla su un'altra classe purché questa sia relazionata con la nostra.

- 7) Per verificare la presenza di questa relazione torniamo sulla prima classe (`Caratteristiche (Scuola)`) e vediamo che tra i suoi domini esiste una object property (un predicato) che descrive una relazione chiamata "ha scuola anagrafica".

Caratteristiche (Scuola)^C [torna a IdC or Classi IdC](#)

IRI: http://dati.comune.milano.it:8890/schemas/ontomi/scuola_caratteristiche

Insieme di dati costituenti le caratteristiche delle scuole

è definito da
<http://dati.comune.milano.it:8890/schemas/ontomi/>

è nel dominio di
[anno scolastico](#) ^{dp}, [capienza](#) ^{dp}, [classe eta](#) ^{dp}, [codice della unita educativa](#) ^{dp}, [codice scuola](#) ^{dp}, [flag paritaria](#) ^{dp}, [grado della scuola](#) ^{dp}, [ha gestione scuola](#) ^{op}, [ha intervallo temporale](#) ^{op}, [ha scuola anagrafica](#) ^{op}, [id caratteristiche scuola](#) ^{dp}, [id struttura erogazione](#) ^{dp}, [nome della scuola](#) ^{dp}, [tipo di struttura](#) ^{dp}

è nel codominio di
[anagrafica scuola per](#) ^{op}, [gestione scuola per](#) ^{op}, [intervallo temporale per](#) ^{op}

- 8) Cliccando sul dominio "ha scuola anagrafica" si salta sulla descrizione di questa object property.

ha scuola anagrafica^{op} [torna a IdC or Object Property IdC](#)

IRI: http://dati.comune.milano.it:8890/schemas/ontomi/has_scuola_anagrafica

è definito da
<http://dati.comune.milano.it:8890/schemas/ontomi/>

ha dominio
[Alunno \(Scuola\)](#) ^c
[Caratteristiche \(Scuola\)](#) ^c

ha codominio
[Scuola](#) ^c

Il predicato riporta nell'elenco "ha dominio" la classi di partenza (tra le quali la nostra Caratteristiche (Scuola)) e nell'elenco "ha codominio" quelle di destinazione come "Scuola". Questo predicato è quindi quello corretto che mette in relazione le caratteristiche con la rispettiva anagrafica.

Il nome del predicato così come è memorizzato sul server è quello riportato in fondo all'indirizzo IRI (come evidenziato in rosso sull'immagine)

- 9) Entriamo nel dettaglio della classe Scuola per analizzare quali dati contiene.

Scuola^c [torna a IdC or Classi IdC](#)

IRI: http://dati.comune.milano.it:8890/schemas/ontomi/scuola_anagrafica

Insieme di dati costituenti l'anagrafica delle scuole.

è definito da
<http://dati.comune.milano.it:8890/schemas/ontomi/>

è equivalente a
[school](#) ^c

è nel dominio di
[anagrafica scuola per](#) ^{op}, [cap scuola](#) ^{dp}, [codice scuola](#) ^{dp}, [descrizione scuola](#) ^{dp}, [email scuola](#) ^{dp}, [ha zona](#) ^{op},
[indirizzo scuola](#) ^{dp}, [nome della scuola](#) ^{dp}, [numero civico scuola](#) ^{dp}, [tipo sede scuola](#) ^{dp}, [zona scuola](#) ^{dp}

è nel codominio di
[anagrafica zona per](#) ^{op}, [ha scuola anagrafica](#) ^{op}

Il nome della classe così come è memorizzata sul server è quello riportato in fondo all'indirizzo IRI (come evidenziato in rosso sull'immagine). Tra le informazioni contenute nella classe vi è "zona scuola" che è proprio il campo che stiamo cercando per poter applicare il nostro filtro.

- 10) Verifichiamo per l'attributo zona scuola qual è il suo nome reale a sistema cliccando sull'elemento e prendendo il nome in fondo all'IRI (zona_scuola).

zona scuola^{dp} [torna a IdC](#) or [Data Property IdC](#)

IRI: http://dati.comune.milano.it:8890/schemas/ontomi/zona_scuola

Zona di riferimento dove e' localizzata la scuola.

è definito da
<http://dati.comune.milano.it:8890/schemas/ontomi/>

ha dominio
[Scuola](#)^c

ha codominio
string

- 11) A questo punto abbiamo tutte le informazioni per costruire la query finale:
- Campi di output: anno_scolastico e capienza
 - Classi sorgenti: scuola_caratteristiche, scuola_anagrafica
 - Filtri: grado_scuola = 'Nido d infanzia' e zona_scuola = '9'
 - Predicati: has_scuola_anagrafica

12) Componiamo la query finale spiegandone la struttura:

```

prefix ontomi: <http://dati.comune.milano.it:8890/schemas/ontomi/>
select
  ?anno , sum(?capienza) as ?capienza_totale
where{
  ?sccar a ontomi:scuola_caratteristiche.
  ?sccar ontomi:capienza ?capienza.
  ?sccar ontomi:anno_scolastico ?anno.
  ?sccar ontomi:grado_scuola ?grado.
  ?sccar ontomi:has_scuola_anagrafica ?anag.
  ?anag a ontomi:scuola_anagrafica.
  ?anag ontomi:nome_scuola ?scuola.
  ?anag ontomi:zona_scuola ?zona.
  filter( ?grado = 'Nido d infanzia' and ?zona='9' )
}
group by ?anno

```

(1) La direttiva **PREFIX** è necessaria per indicare l'elenco delle ontologie che si andranno ad interrogare e segue questa sintassi:

PREFIX *alias_ontologia*:<URI ontologia>

(2) Dopo il prefisso **“SELECT”** si elencano tutti gli alias dei campi (data properties) che si desiderano in output separati da virgola.

Importante: ciascun nome campo deve SEMPRE essere preceduto dal carattere **“?”** e deve coincidere con il corrispondente alias assegnato dentro il blocco **WHERE**.

(3) Per ottenere in output il calcolo del totale dei posti si applica l'operatore **SUM** sul campo capienza.

- (4) L'operatore "**AS**" seguito da un nome a scelta (preceduto da **?** e senza spazi) serve a ridefinire il nome di una colonna sul risultato di output.
- (5) Il blocco **WHERE** { ... } deve contenere: le classi sorgenti, le data properties, eventuali filtri.
- (6) L'operatore "**A**" serve a specificare una classe sorgente dal quale si vogliono estrarre i dati e segue questa sintassi:

?[alias_sorgente] A [alias_ontologia]:[nome classe].

dove *alias_sorgente* è un nome a scelta che si userà nella query per riferirsi a quella classe

- (7) Dopo aver specificato con l'operatore "**A**" la classe sorgente deve seguire l'elenco delle data properties coinvolte nella query (campi di output, campi di filtro, campi di join) seguendo questa sintassi:

?[alias_sorgente] [alias_ontologia]:[nome campo] ?[alias_campo].

dove:

- *alias_sorgente* è l'alias scelto al punto (6) per la rispettiva classe sorgente
- *alias_ontologia* è l'alias scelto al punto (1) come prefisso per l'ontologia
- *nome campo* è il nome della data property utilizzata al punto (2)
- *alias_campo* è un nome a scelta che si userà nella query per riferirsi a quella specifica data property

- (8) Per relazionare i dati tra classi occorre specificarle dichiarando i predicati seguendo questa sintassi :

?[alias_classe sorgente] [alias_ontologia]:[nome predicato] ?[alias_classe_destinazione].

- (9) Dentro il blocco facoltativo **FILTER** (...) si inserisce la condizione di filtro utilizzando la logica booleana. Per riferirsi alle data properties bisogna sempre utilizzare gli alias assegnati nel blocco precedente (WHERE).
- (10) La direttiva **GROUP BY** viene utilizzata in congiunzione all'operatore di aggregazione **SUM** affinché il risultato venga raggruppato sulla data property specificata, sempre utilizzando l'alias assegnato nel blocco precedente (WHERE).

Questo è il risultato ottenuto dalla query:

Risultati

anno	capienza_totale
ANNO_SCOL_18_19	1417
ANNO_SCOL_17_18	1416
ANNO_SCOL_16_17	1433

Per approfondimenti sul linguaggio SPARQL fare riferimento a: <https://www.w3.org/TR/rdf-sparql-query/>