



Home > Courses > PROG. IK REGULAR > REG - Gasal 2022/2023 > [Reg] Semantic Web (Jejaring Semantik) Gasal 2022/... > Querying Data Graph > Worksheet/PR 3

**Started on** Tuesday, 27 September 2022, 9:50 PM

**State** Finished

**Completed on** Sunday, 2 October 2022, 12:07 AM

**Time taken** 4 days 2 hours

**Grade** Not yet graded

### Information

Anda diberikan basis data mengenai musik yang [https://drive.google.com/file/d/1kcUu5uwVUz5jVB\\_pZb\\_9wMNjemg93jQK/view?usp=sharing](https://drive.google.com/file/d/1kcUu5uwVUz5jVB_pZb_9wMNjemg93jQK/view?usp=sharing)

Tugas Anda adalah membuat beberapa kueri SPARQL untuk menjawab kebutuhan-kebutuhan informasi dari data sesuai soal yang diberikan.

Petunjuk pengerjaan:

- Untuk menyelesaikan soal ini, Anda dapat menggunakan triple store yang Anda sukai, atau jika Anda belum menginstal, silakan unduh BlazeGraph dengan mengikuti petunjuk pada tautan yang diberikan di laman kuliah ini di SCeLE. Petunjuk-petunjuk berikutnya berlaku jika Anda menggunakan BlazeGraph.
- Jalankan blazegraph.jar dengan Java Virtual Machine di command line/terminal sesuai petunjuk di dokumen QuickStart. Perhatikan alamat dan nomor port yang muncul di command line karena itu yang akan dipakai untuk mengakses blazegraph dari browser Anda.
- Upload/update data music.ttl yang Anda sudah unduh sebelumnya ke dalam BlazeGraph dengan mengakses tab Update di aplikasi BlazeGraph.
- Kueri SPARQL dapat dicoba-coba dengan mengakses tab Query di aplikasi BlazeGraph.

**Question 1**

Complete

Marked out of 5.00

Berikan nama semua kelas dan property di dalam data. Untuk setiap kelas, tampilkan berapa banyak instans unik dari kelas tersebut, lalu untuk setiap property, tampilkan berapa banyak pasangan/instans unik yang terhubung dengan property tersebut.

Catatan:

Suatu IRI P merupakan sebuah property, jika ia muncul sebagai predikat di setidaknya sebuah tripel di dalam graf.

Instans dari P adalah setiap pasangan unik x dan y sehingga tripel (x, P, y).

Suatu IRI C di dalam graf merupakan sebuah kelas jika C muncul dalam salah satu tripel berikut:

- (X, rdf:type, C).
- (A, rdfs:subClassOf, C)
- (C, rdfs:subClassOf, D)
- (P, rdfs:domain, C)
- (P, rdfs:range, C)

X merupakan instans dari kelas C apabila:

- (X, rdf:type, C) ada di graf, atau
- ada D sehingga (X, rdf:type, D) ada di graf dan D merupakan subclass dari C.

D merupakan subclass dari C apabila:

- C = D atau
- ada E sehingga (D, rdfs:subClassOf, E) ada di graf dan E merupakan subclass dari C.

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```
SELECT * WHERE {
  {
    SELECT (?class as ?barang) ("class" as ?jenis) (COUNT(?instance) AS ?hitungBarang)
  WHERE {
    {
      SELECT DISTINCT ?class WHERE {
        {
          [ a ?class ]
        } UNION {
          [ rdfs:subClassOf ?class ]
        } UNION {
          ?class rdfs:subClassOf []
        } UNION {
          [ rdfs:domain ?class ]
        } UNION {
          [ rdfs:range ?class ]
        }
      }
    }
    # Simbol asterisk untuk mengecek 0 atau lebih rantai
    ?subclass rdfs:subClassOf* ?class .
    ?instance a ?subclass .
  }
  GROUP by ?class
} UNION {
  SELECT (?prop as ?barang) ("property" as ?jenis) (COUNT(*) AS ?hitungBarang)
WHERE {
  # Menggunakan blank node untuk menghitung banyaknya pasangan yang cocok
  [] ?prop []
}
  GROUP BY ?prop
}
}
```

## Question 2

Complete

Marked out of 5.00

Cari nama semua artis beserta nama album-album yang pernah mereka rilis. Untuk setiap artis, berikan informasi apakah artis tersebut artis solo atau sebuah band. Lalu, untuk setiap album, kembalikan tanggal rilisnya jika ada. Urutkan hasilnya berdasarkan tanggal rilis albumnya.

Berdasarkan query berikut:

```
PREFIX : <http://domain.org/ns/>

SELECT (COUNT(DISTINCT *) as ?tanggalDouble) WHERE {
    ?album :date ?tanggal .
        ?album :date ?tanggal2 .
    FILTER (?tanggal < ?tanggal2)
}
ORDER BY ?tanggal
```

Terdapat beberapa album yang tanggal nya lebih dari satu, diasumsikan tanggal rilis merupakan tanggal yang paling kecil untuk setiap property `:date` pada sebuah album. Untungnya, setiap album memiliki setidaknya satu properti `:date`.

```
PREFIX : <http://domain.org/ns/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT * WHERE {
    ?album rdf:type :Album .
    OPTIONAL {
        ?album :name ?namaAlbum .
    }
    FILTER (!BOUND(?namaAlbum))
}
ORDER BY ?album
```

Berdasarkan query di atas, diketahui ada satu album bernama X dari Ed Sheeran yang tidak memiliki properti nama album. Sehingga ditulis query sebagai berikut.

```
PREFIX : <http://domain.org/ns/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT DISTINCT ?artis ?namaArtis ?jenisArtis ?album ?namaAlbum ?tanggal WHERE {
  ?jenisArtis rdfs:subClassOf :Artist .
  ?artis a ?jenisArtis .
  ?artis :name ?namaArtis
  OPTIONAL {
    ?album :artist ?artis .
    ?album a :Album .
    ?album :date ?tanggal .
    OPTIONAL { ?album :name ?namaAlbum }
  }
  FILTER (
    NOT EXISTS
    {
      ?album :date ?tanggalLain
      FILTER (?tanggal > ?tanggalLain)
    } || !BOUND(?album)
  )
}
ORDER BY ?tanggal
```

### Question 3

Complete

Marked out of 5.00

Cari semua nama band yang memiliki anggota yang juga merupakan artis solo. Kembalikan nama band tersebut beserta nama anggotanya yang menjadi artis solo.. Urutkan hasilnya berdasarkan nama band lalu berdasarkan nama artisnya.

```
PREFIX : <http://domain.org/ns/>
```

```
SELECT * WHERE {
  # Mencari band dan solo artis membernya
  ?band a :Band .
  ?band :member ?soloArtis .
  ?band :name ?namaBand .
  ?soloArtis a :SoloArtist .
  ?soloArtis :name ?namaSoloArtis .
}
ORDER BY ?namaBand ?namaSoloArtis
# Urutkan berdasarkan nama band, kemudian nama solo artis
```

**Question 4**

Complete

Marked out of 5.00

Cari semua judul album yang dirilis di dekade 1970an (tahun 1970 hingga 1979) yang dirilis oleh seorang artis solo dan diproduksi oleh artis itu sendiri (mungkin bersama produser lain). Selain judul album, tampilkan juga nama artis serta tanggal rilisnya.

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

SELECT ?namaArtis ?namaAlbum ?tanggal WHERE {
  ?artis :name ?namaArtis .
  ?artis a :SoloArtist .
  ?album :producer ?artis .
  ?album :artist ?artis .
  ?album a :Album .
  ?album :date ?tanggal .
  OPTIONAL { ?album :name ?namaAlbum } .
  FILTER (?tanggal >= "1970-01-01"^^xsd:date && ?tanggal < "1980-01-01"^^xsd:date)
  FILTER NOT EXISTS {
    ?album :date ?tanggalLain
    FILTER (?tanggal > ?tanggalLain)
  }
}
```

**Question 5**

Complete

Marked out of 5.00

Cari semua judul album yang dirilis di dekade 1980an (tahun 1980 hingga 1989) yang dirilis oleh sebuah band dan diproduksi oleh band itu sendiri atau oleh salah satu anggotanya (mungkin bersama produser lain). Selain judul album, tampilkan juga nama band, tanggal rilis, beserta jumlah track yang ada di dalam album tersebut.

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

# Aggregation dengan group album artis tanggal
SELECT DISTINCT ?album ?artis ?tanggal (COUNT(DISTINCT ?lagu) as ?laguCount) WHERE {
    ?artis :name ?namaArtis .
    ?artis a :Band .
    ?album :artist ?artis .
    ?album a :Album .
    ?album :producer ?produser .
    ?album :date ?tanggal .
    # Mengecek semua lagu pada album ini
    ?album :track ?lagu .
    # Left join dengan nama album
    OPTIONAL { ?album :name ?namaAlbum } .
    FILTER (?tanggal >= "1980-01-01"^^xsd:date && ?tanggal < "1990-01-01"^^xsd:date)
    FILTER NOT EXISTS {
        ?album :date ?tanggalLain # Mengecek tanggal terkecil
        FILTER (?tanggal > ?tanggalLain)
    }
    # Mengecek produsernya merupakan artis (band) atau membernya langsung
    FILTER (?produser = ?artis ||
        EXISTS { ?artis :member ?produser . }
    )
}
GROUP BY ?album ?artis ?tanggal
```

**Question 6**

Complete

Marked out of 5.00

Cari 10 nama penulis lagu dengan jumlah lagu terbanyak (yang ditulisnya). Untuk setiap penulis lagu tersebut, tampilkan juga total berapa album yang berisi lagu-lagu yang ditulisnya. Urutkan hasilnya berdasarkan jumlah lagu.

```
PREFIX : <http://domain.org/ns/>
```

```
SELECT ?penulis ?banyakTulisan ?countAlbum {
  {
    # Mencari penulis dengan jumlah lagu terbanyak
    SELECT ?penulis (COUNT(?lagu) as ?banyakTulisan) WHERE {
      ?lagu a :Song .
      ?lagu :writer ?penulis
    }
    GROUP BY ?penulis
    ORDER BY DESC(?banyakTulisan) # Limit ke 10 teratas saja setelah diurutkan
    LIMIT 10
  }
  {
    # Join dengan kontribusinya terhadap album
    SELECT (?penulis2 as ?penulis) (COUNT(?album) as ?countAlbum) WHERE {
      {
        SELECT DISTINCT ?penulis2 ?album WHERE {
          ?lagulain :writer ?penulis2 .
          ?album :track ?lagulain
        }
        ORDER BY ?penulis2
      }
    }
    GROUP BY ?penulis2
  }
}
ORDER BY DESC(?banyakTulisan)
```



**Question 7**

Complete

Marked out of 5.00

Cari nama semua band yang beranggotakan minimal 5 orang. Tampilkan nama band beserta jumlah anggotanya.

```
PREFIX : <http://domain.org/ns/>
```

```
SELECT * WHERE {  
  {  
    # Mencari jumlah anggota untuk semuanya  
    SELECT ?band ?namaBand (COUNT(?anggota) AS ?anggotaCount) WHERE {  
      ?band a :Band .  
      ?band :member ?anggota .  
      ?band :name ?namaBand .  
    }  
    GROUP BY ?band ?namaBand  
  }  
  # Filter untuk anggotanya >= 5 saja  
  FILTER (?anggotaCount >= 5)  
}
```

**Question 8**

Complete

Marked out of 5.00

Cari semua track/lagu terpanjang maupun terpendek dari setiap album. Tampilkan judul album, judul track, nama artis (band/artis solo), panjang track, beserta nama pencipta lagunya (ditulis dalam satu baris menggunakan GROUP CONCAT dengan separator koma jika suatu lagu memiliki beberapa pencipta). Urutkan berdasarkan panjang track.

PREFIX : <http://domain.org/ns/>

```
SELECT ?judulAlbum ?judulLagu ?namaArtis ?panjangLagu ?paraPenulis WHERE {
  ?album :artist ?artis .
  ?album :name ?judulAlbum .
  ?artis :name ?namaArtis .
  {
    # Mencari semua data lagu dengan panjang dan concat dari penulisnya
    SELECT ?album ?lagu ?judulLagu ?panjangLagu (GROUP_CONCAT(
                                                                    ?namaPenulis;SEPARATOR=", ") AS ?
paraPenulis) WHERE {
      ?album :track ?lagu .
      ?lagu :writer ?penulis .
      ?penulis :name ?namaPenulis .
      ?lagu :length ?panjangLagu .
      ?lagu :name ?judulLagu
    }
    GROUP BY ?album ?lagu ?judulLagu ?panjangLagu
  }
  # Melakukan filter untuk mengecek apakah tidak ada lagu yang lebih panjang dari p
ada dia
  FILTER ((NOT EXISTS
    {
      ?album :track ?laguLain .
      ?laguLain :length ?panjangLaguLain .
      ?laguLain :name ?judulLaguLain .
      FILTER (
        ?panjangLaguLain > ?panjangLagu ||
        (?panjangLaguLain = ?panjangLagu && ?judulLaguLain > ?judulLagu )
      )
    }
    ) ||
    # Melakukan filter untuk mengecek apakah tidak ada lagu yang lebih pend
ek dari pada dia
    (NOT EXISTS
      {
        ?album :track ?laguLain .
        ?laguLain :length ?panjangLaguLain .
        ?laguLain :name ?judulLaguLain .
        FILTER (
          ?panjangLaguLain < ?panjangLagu ||
          (?panjangLaguLain = ?panjangLagu && ?judulLaguLain < ?judulLagu )
        )
      }
      )
    )
}
ORDER BY ?panjangLagu # Urut berdasarkan panjang lagu
```