

Lösung

Aufgabe 3.1: Betrachten Sie das folgende RDF-Dokument mit Informationen über einige Himmelskörper:

@prefix ex: <http://example.org/> .

ex:Sonne	ex:radius	"1.392e6"^^xsd:double ;
	ex:satellit	ex:Merkur, ex:Venus, ex:Erde, ex:Mars .
ex:Merkur	ex:radius	"2439.7"^^xsd:double .
ex:Venus	ex:radius	"6051.9"^^xsd:double .
ex:Erde	ex:radius	"6372.8"^^xsd:double ;
	ex:satellit	ex:Mond .
ex:Mars	ex:radius	"3402.5"^^xsd:double ;
	ex:satellit	ex:Phobos, ex:Deimos .
ex:Mond	ex:name	"Mond@de", "Moon@en" ;
	ex:radius	"1737.1"^^xsd:double .
ex:Phobos	ex:name	"Phobos" .
ex:Deimos	ex:name	"Deimos" .

SPARQL-Anfragen

1. Objekte, die um die Sonne oder um einen Satelliten der Sonne kreisen.

@PREFIX ex: <http://example.org/>.

```
SELECT ?objekt
WHERE {
  { ex:Sonne      ex:satellit      ?objekt.} UNION
  { ex:Sonne      ex:satellit      ?objekt_tmp.
    ?objekt_tmp   ex:satellit      ?objekt.}
}
```

2. Objekte mit einem Volumen von über $2 \cdot 10^{10}$ (km³) (Himmelskörper mit Radius können als kugelförmig angenommen werden, d.h. $V = \frac{4}{3} \times \text{Pi} \times r^3$) und, falls vorhanden, dem Objekt, dessen Satellit sie sind.

```
SELECT ?objekt ?mittelpunkt
WHERE {
  { ?objekt      ex:radius      ?rad.}
  OPTION { ?mittelpunkt   ex:satellit      ?objekt.}
  FILTER (4/3 * 3.1416 * ?rad * ?rad * ?rad > 20000000000)
}
```

3. Objekte mit einem Satelliten, für den ein englischsprachiger Name gegeben ist, die außerdem Satellit eines Objektes von über 3000 (km) Durchmesser sind.

```
SELECT ?objekt
WHERE {
  { ?objekt          ex:satellit          ?satellit.
    ?satellit        ex:name              ?name.
    ?mittelpunkt      ex:satellit          ?objekt.
    ?mittelpunkt      ex:radius            ?rad.
  }
  FILTER (langMATCHES (LANG(?name), "en") )
  FILTER (2*rad > 3000)
}
```

4. Objekte mit zwei oder mehr Satelliten (nehmen Sie an, dass unterschiedliche URIs hier unterschiedliche Objekte bezeichnen).

```
SELECT ?objekt
WHERE {
  { ?objekt          ex:satellit          ?satellit1.
    ?objekt          ex:satellit          ?satellit2.
  }
  FILTER (!sameTERM (?satellit1, ?satellit2) )
}
```

Aufgabe 3.2 Übersetzen Sie die Anfragen aus Aufgabe 3.1 in Ausdrücke der SPARQL-Algebra. Dabei können Join-Operationen mit dem leeren Graphen Z sofort vereinfacht werden.



```
1.  SELECT  ?objekt
    WHERE  {
      { ex:Sonne      ex:satellit      ?objekt. } UNION
      { ex:Sonne      ex:satellit      ?objekt_tmp.
        ?objekt_tmp  ex:satellit      ?objekt. }
    }
```



```
{
  BGP ( <http://example.org/sonne>  <http://example.org/satellit>  ?objekt.),
  UNION
  { BGP ( <http://example.org/sonne >  <http://example.org/satellit>  ?objekt_tmp. ),
    BGP (?objekt_tmp  <http://example.org/satellit>  ?objekt. )
  }
}
```

```
{  
  BGP ( <http://example.org/sonne>    <http://example.org/satellit>  ?objekt.} ),  
  UNION{  
    BGP ( <http://example.org/sonne >    <http://example.org/satellit>  ?objekt_tmp. ),  
    BGP (?objekt_tmp <http://example.org/satellit>  ?objekt. ) }  
}
```



```
{  
  BGP ( <http://example.org/sonne>    <http://example.org/satellit>  ?objekt.} ),  
  UNION{  
    Join (  
      BGP ( <http://example.org/sonne >    <http://example.org/satellit>  ?objekt_tmp. ),  
      BGP (?objekt_tmp <http://example.org/satellit>  ?objekt. )  
    )  
}
```

```
{  
  BGP ( <http://example.org/sonne> <http://example.org/satellit> ?objekt.} ),  
  UNION{  
    Join (  
      BGP ( <http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp. ),  
      BGP (?objekt_tmp <http://example.org/satellit> ?objekt. )  
    )  
  }  
}
```



Union

```
Union (  
  BGP ( <http://example.org/sonne> <http://example.org/satellit> ?objekt.} ),  
  Join (  
    BGP ( <http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp. ),  
    BGP (?objekt_tmp <http://example.org/satellit> ?objekt. )  
  )  
)
```

2. **SELECT** ?objekt ?mittelpunkt
WHERE {
 { ?objekt ex:radius ?rad.}
OPTION { ?mittelpunkt ex:satellit ?objekt.}
FILTER (4/3 * 3.1416 * ?rad * ?rad * ?rad > 20000000000)
 }

BGP introduced & Filter removed

{
 {*BGP* (?objekt <http://example.org/radius> ?rad.)}
OPTION {*BGP* (?mittelpunkt <http://example.org/satellit> ?objekt.)}
 }

LeftJoin

LeftJoin (*BGP* (?objekt <http://example.org/radius> ?rad.),
BGP (?mittelpunkt <http://example.org/satellit> ?objekt.),
 true
)

```
LeftJoin (BGP (?objekt <http://example.org/radius> ?rad. ),  
           BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),  
           true  
           )
```



Filter introduced

```
Filter(  $(4/3 * 3.1416 * ?rad * rad * rad > 20000000000)$  ,  
LeftJoin (BGP (?objekt <http://example.org/radius> ?rad. ),  
           BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),  
           true  
           )  
)
```



```

3.  SELECT  ?objekt
      WHERE {
        { ?objekt          ex:satellit          ?satellit.
          ?satellit       ex:name              ?name.
          ?mittelpunkt     ex:satellit          ?objekt.
          ?mittelpunkt     ex:radius            ?rad.
        FILTER (langMATHCES (LANG(?name), "en") )
        FILTER (2*rad > 3000)
      }

```



BGP introduced & Filter removed

```

{
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )
  BGP (?satellit <http://example.org/name> ?name. )
  BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt. )
  BGP ( ?mittelpunkt <http://example.org/radius> ?rad. )
}

```

```
{  
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  BGP (?satellit <http://example.org/name> ?name. )  
  BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.)  
  BGP ( ?mittelpunkt <http://example.org/radius> ?rad.)  
}
```



```
{  
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  BGP (?satellit <http://example.org/name> ?name. )  
  Join(BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.) ,  
  BGP ( ?mittelpunkt <http://example.org/radius> ?rad.))  
}
```

```
{  
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  BGP (?satellit <http://example.org/name> ?name. )  
  Join(BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.),  
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.))  
}
```



```
{  
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  Join(BGP (?satellit <http://example.org/name> ?name. ),  
  Join(BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.),  
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.)))  
}
```

```
{  
  BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  Join(BGP (?satellit <http://example.org/name> ?name. ),  
  Join(BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.),  
  BGP ( ?mittelpunkt <http://example.org/radius> ?rad.)))  
}
```



```
{  
  Join(BGP ( ?objekt <http://example.org/satellit> ?satellit. )  
  Join(BGP (?satellit <http://example.org/name> ?name. ),  
  Join(BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.),  
  BGP ( ?mittelpunkt <http://example.org/radius> ?rad.)))  
}
```

{

Join(BGP (?objekt <http://example.org/satellit> ?satellit.)

Join(BGP (?satellit <http://example.org/name> ?name.),

Join(BGP (?mittelpunkt <http://example.org/satellit> ?objekt.),

BGP (?mittelpunkt <http://example.org/radius> ?rad.)))))

}



Filter Introduced

Filter(((langMATCHES(LANG(?name), "en")) && (2 * ?rad > 3000)),

Join(BGP (?objekt <http://example.org/satellit> ?satellit.)

Join(BGP (?satellit <http://example.org/name> ?name.),

Join(BGP (?mittelpunkt <http://example.org/satellit> ?objekt.),

BGP (?mittelpunkt <http://example.org/radius> ?rad.)))))

)

```

4.  SELECT  ?objekt
      WHERE {
        { ?objekt      ex:satellit      ?satellit1.
          ?objekt      ex:satellit      ?satellit2.
        }
        FILTER (!sameTERM (?satellit1, ?satellit2) )
      }

```



BGP Introduced and Filter removed

```

{
  BGP ( ?objekt <http://example.org/satellit> ?satellit1. )
  BGP (?objekt <http://example.org/satellit> ?satellit2. )
}

```

```
{  
    BGP ( ?objekt <http://example.org/satellit> ?satellit1. )  
    BGP (?objekt <http://example.org/satellit> ?satellit2. )  
}
```



```
{  
Join ( BGP ( ?objekt <http://example.org/satellit> ?satellit1. ),  
        BGP (?objekt <http://example.org/satellit> ?satellit2. )  
      )  
}
```

```
{  
Join (BGP (?objekt <http://example.org/satellit> ?satellit1. ),  
      BGP (?objekt <http://example.org/satellit> ?satellit2. )  
)  
}
```



Filter Introduced

```
Filter( (!sameTERM(?satellit1, ?satellit2)),  
      Join (BGP (?objekt <http://example.org/satellit> ?satellit1. ),  
            BGP (?objekt <http://example.org/satellit> ?satellit2. )  
            )  
      )
```


Aufgabe 3.3 Berechnen Sie die Ergebnisse der Ausdrücke aus Aufgabe 3.2 bezüglich der Wissensbasis aus Aufgabe 3.1. Berechnen Sie Teilergebnisse schrittweise wie auf den Folien 24–32 der Vorlesung “Semantic von SPARQL” dargestellt. Stimmen Ihre Ergebnisse mit den erwarteten Resultaten überein?

Union (

BGP (<http://example.org/sonne> <http://example.org/satellit> ?objekt.}),

Join (

BGP (<http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp.),

BGP (?objekt_tmp <http://example.org/satellit> ?objekt.)

)

)



Berechnung

objekt_tmp
ex:Merkur
ex:Venus
ex:Erde
ex:Mars

Union (

BGP (<http://example.org/sonne> <http://example.org/satellit> ?objekt.}),

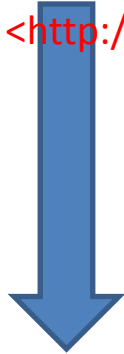
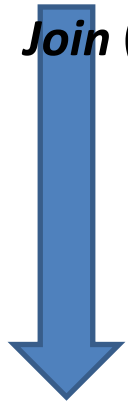
Join (

BGP (<http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp.),

BGP (?objekt_tmp <http://example.org/satellit> ?objekt.)

)

)



Berechnung

objekt
ex:Merkut
ex:Venus
ex:Erde
ex:Mars

objekt_tmp
ex:Merkur
ex:Venus
ex:Erde
ex:Mars

objekt_tmp	objekt
ex:Sonne	ex:Merkur
ex:Sonne	ex:Venus
ex:Sonne	ex:Erde
ex:Sonne	ex:Mars
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

Union (

BGP (<http://example.org/sonne> <http://example.org/satellit> ?objekt.}),

Join (

BGP (<http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp.),

BGP (?objekt_tmp <http://example.org/satellit> ?objekt.)

)

)

Objekt_tmp	objekt
ex:Sonne	ex:Merkur
ex:Sonne	ex:Venus
ex:Sonne	ex:Erde
ex:Sonne	ex:Mars
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

objekt_tmp
ex:Merkur
ex:Venus
ex:Erde
ex:Mars

Join

Objekt_tmp	objekt
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

Union (

BGP (<http://example.org/sonne> <http://example.org/satellit> ?objekt.}),

Join (

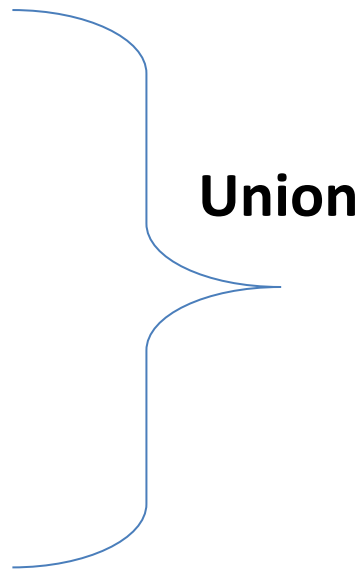
BGP (<http://example.org/sonne > <http://example.org/satellit> ?objekt_tmp.),

BGP (?objekt_tmp <http://example.org/satellit> ?objekt.)

)

)

objekt
ex:Merkur
ex:Venus
ex:Erde
ex:Mars



Objekt_tmp	objekt
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

objekt
ex:Merkur
ex:Venus
ex:Erde
ex:Mars
ex:Mond
ex:Phobos
ex:Deimos

```

Filter( (4/3 * 3.1416 * ?rad * rad * rad > 200000000000) ,
LeftJoin ( BGP ( ?objekt <http://example.org/radius> ?rad. ),
           BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),
           true
         )
)

```

objekt	rad
ex:Sonne	"1.392e6"^^xsd::double
ex:Merkur	"2439.7"^^xsd:double
ex:Venus	"6051.8"^^xsd:double
ex:Erde	"6372.8"^^xsd:double
ex:Mars	"3402.5"^^xsd:double
ex:Mond	"1737.1"^^xsd:double

```

Filter( (4/3 * 3.1416 * ?rad * rad *rad > 200000000000) ,
LeftJoin (BGP ( ?objekt <http://example.org/radius> ?rad. ),
             BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),
             true
           )
)

```

mittelpunkt	objekt
ex:Sonne	ex:Merkur
ex:Sonne	ex:Venus
ex:Sonne	ex:Erde
ex:Sonne	ex:Mars
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

```

Filter( (4/3 * 3.1416 * ?rad * rad * rad > 200000000000) ,
LeftJoin (BGP ( ?objekt <http://example.org/radius> ?rad. ),
          BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),
          true
        )
)

```

objekt	rad	mittelpunkt
ex:Sonne	"1.392e6"^^xsd:double	
ex:Merkur	"2439.7"^^xsd:double	ex:Sonne
ex:Venus	"6051.8"^^xsd:double	ex:Sonne
ex:Erde	"6372.8"^^xsd:double	ex:Sonne
ex:Mars	"3402.5"^^xsd:double	ex:Sonne
ex:Mond	"1737.1"^^xsd:double	ex:Erde

```

Filter( (4/3 * 3.1416 * ?rad * rad * rad > 200000000000) ,
LeftJoin (BGP ( ?objekt <http://example.org/radius> ?rad. ),
          BGP (?mittelpunkt <http://example.org/satellit> ?objekt. ),
          true
        )
)

```

objekt	rad	mittelpunkt
ex:Sonne	"1.392e6"^^xsd:double	
ex:Merkur	"2439.7"^^xsd:double	ex:Sonne
ex:Venus	"6051.8"^^xsd:double	ex:Sonne
ex:Erde	"6372.8"^^xsd:double	ex:Sonne
ex:Mars	"3402.5"^^xsd:double	ex:Sonne
ex:Mond	"1737.1"^^xsd:double	ex:Erde


```

Filter( ((langMATCHES( LANG(?name), „en“)) && (2 * ?rad > 3000)),
  Join(Join(Join (BGP ( ?objekt <http://example.org/satellit> ?satellit. ),
    BGP (?satellit <http://example.org/name> ?name. )
  ),
    BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.
  ),
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.
  ),
)

```

satellit	objekt
ex:Sonne	ex:Merkur
ex:Sonne	ex:Venus
ex:Sonne	ex:Erde
ex:Sonne	ex:Mars
ex:Erde	ex:Mond
ex:Mars	ex:Phobos
ex:Mars	ex:Deimos

satellit	name
ex:Mond	"Mond@de"
ex:Mond	"Moon@en"
ex:Phobos	"Phobos"
ex:Deimos	"Deimos"

```

Filter( ((langMATCHES( LANG(?name), „en“)) && (2 * ?rad > 3000)),
  Join(Join(Join (BGP ( ?objekt <http://example.org/satellit> ?satellit. ),
    BGP (?satellit <http://example.org/name> ?name. )
  ),
    BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.
  ),
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.
  ),
)

```

objekt	satellit	name
ex:Erde	ex:Mond	"Mond@de"
ex:Erde	ex:Mond	"Moon@en"
ex:Mars	ex:Phobos	"Phobos"
ex:Mars	ex:Deimos	"Deimos"

```

Filter( ((langMATCHES( LANG(?name), „en“)) && (2 * ?rad > 3000)),
  Join(Join(Join (BGP ( ?objekt <http://example.org/satellit> ?satellit. ),
    BGP (?satellit <http://example.org/name> ?name. )
  ),
    BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.
  ),
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.
  ),
)

```

mittelpunkt	objekt	satellit	name
ex:Sonne	ex:Erde	ex:Mond	"Mond@de"
ex:Sonne	ex:Erde	ex:Mond	"Moon@en"
ex:Sonne	ex:Mars	ex:Phobos	"Phobos"
ex:Sonne	ex:Mars	ex:Deimos	"Deimos"

```

Filter( ((langMATCHES( LANG(?name), „en“)) && (2 * ?rad > 3000)),
    Join(Join(Join (BGP ( ?objekt <http://example.org/satellit> ?satellit. ),
        BGP (?satellit <http://example.org/name> ?name. )
    ),
        BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.
    ),
        BGP ( ?mittelpunkt <http://example.org/radius> ?rad.
    ),
)

```

mittelpunkt		objekt	satellit	name
ex:Sonne	"1.392e6"^^xsd:double	ex:Erde	ex:Mond	"Mond@de"
ex:Sonne	"1.392e6"^^xsd:double	ex:Erde	ex:Mond	"Moon@en"
ex:Sonne	"1.392e6"^^xsd:double	ex:Mars	ex:Phobos	"Phobos"
ex:Sonne	"1.392e6"^^xsd:double	ex:Mars	ex:Deimos	"Deimos"

```

Filter( ((langMATCHES( LANG(?name), „en“)) && (2 * ?rad > 3000)),
  Join(Join(Join (BGP ( ?objekt <http://example.org/satellit> ?satellit. ),
    BGP (?satellit <http://example.org/name> ?name. )
  ),
    BGP ( ?mittelpunkt <http://example.org/satellit> ?objekt.
  ),
    BGP ( ?mittelpunkt <http://example.org/radius> ?rad.
  ),
)

```

mittelpunkt		objekt	satellit	name
ex:Sonne	"1.392e6"^^xsd:double	ex:Erde	ex:Mond	"Mond@de"
ex:Sonne	"1.392e6"^^xsd:double	ex:Erde	ex:Mond	"Moon@en"
ex:Sonne	"1.392e6"^^xsd:double	ex:Mars	ex:Phobos	"Phobos"
ex:Sonne	"1.392e6"^^xsd:double	ex:Mars	ex:Deimos	"Deimos"

Filter((!sameTERM(?satellit1, ?satellit2)),

Join (**BGP** (?objekt <http://example.org/satellit> ?satellit1.),
BGP (?objekt <http://example.org/satellit> ?satellit2.)
)

)

objekt	satellit1	satellit2
ex:Sonne	ex:Merkur	ex:Merkur
ex:Sonne	ex:Merkur	ex:Venus
ex:Sonne	ex:Merkur	ex:Erde
ex:Sonne	ex:Merkur	ex:Mars
ex:Sonne	ex:Venus	ex:Merkur
ex:Sonne	ex:Venus	ex:Erde
ex:Sonne	ex:Venus	ex:Mars
ex:Sonne	ex:Erde	ex:Merkur
ex:Sonne	ex:Erde	ex:Venus

objekt	satellit1	satellit2
ex:Sonne	ex:Erde	ex:Marse
ex:Sonne	ex:Mars	ex:Merkur
ex:Sonne	ex:Mars	ex:Venus
ex:Sonne	ex:Mars	ex:Erde
ex:Mars	ex:Phobos	ex:Deimos
ex:Mars	ex:Deimos	ex:Phobo

Filter((!sameTERM(?satellit1, ?satellit2)),

Join (**BGP** (?objekt <http://example.org/satellit> ?satellit1.),

BGP (?objekt <http://example.org/satellit> ?satellit2.)

)

)

objekt	satellit1	satellit2
ex:Sonne	ex:Merkur	ex:Venus
ex:Sonne	ex:Merkur	ex:Erde
ex:Sonne	ex:Merkur	ex:Mars
ex:Sonne	ex:Merkur	ex:Merkur
ex:Sonne	ex:Venus	ex:Venus
ex:Sonne	ex:Venus	ex:Erde
ex:Sonne	ex:Venus	ex:Mars
ex:Sonne	ex:Venus	ex:Merkur
ex:Sonne	ex:Erde	ex:Venus

ex:Sonne	ex:Erde	ex:Erde
ex:Sonne	ex:Erde	ex:Marse
ex:Sonne	ex:Erde	ex:Erde
ex:Sonne	ex:Mars	ex:Merkur
ex:Sonne	ex:Mars	ex:Venus
ex:Sonne	ex:Mars	ex:Erde
ex:Sonne	ex:Mars	ex:Mars
ex:Erde	ex:Moon	ex:Moon
ex:Mars	ex:Phobos	ex:Phobos
ex:Mars	ex:Phobos	ex:Deimos
ex:Mars	ex:Deimos	ex:Phobo
ex:Mars	ex:Deimos	ex:Deimos

Aufgabe 3.4 Durch Kombination von Filtern mit optionalen Graph-Mustern kann man in SPARQL auch nach Elementen suchen, für die eine bestimmte Information nicht angegeben ist. Formulieren Sie eine Anfrage nach allen Himmelskörpern, die keinen Satelliten haben.

Gehen Sie dabei davon aus, dass die o.a.Wissensbasis durch Tripel ergänzt wurde, die sämtlichen Himmelskörpern via `rdf:type` den Typ Himmelskörper zuweisen.

@PREFIX ex: <http://example.org/>.

SELECT ?objekt

WHERE {

 { ?objekt rdf:type ex:Himmelskörpern.}

OPTION { ?objekt ex:satellit ?satellit. }

FILTER (!BOUND (?satellit))

}

Note:

xsd:boolean bound (variable var)

Returns true if var is bound to a value.