

SELECT

- **Uitleg:** Gebruikt om specifieke variabelen te selecteren uit de RDF-datasets.
- **Voorbeeld query:**

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o.
}
```

- **Resultaten:**

?s	?p	?o
:John	:hasAge	"28"
:Jane	:hasName	"Jane"
:Bob	:hasAge	"35"

CONSTRUCT

- **Uitleg:** Gebruikt om nieuwe RDF triples te construeren gebaseerd op de resultaten van de query.
- **Voorbeeld query:**

```
CONSTRUCT {
  ?book ex:hasInformation ?info .
  ?info dc:title ?title ;
        dc:creator ?author .
}
WHERE {
  ?book a ex:Book ;
        dc:title ?title ;
        dc:creator ?author .
}
```

- **Triples:**

```
ex:book1 ex:hasInformation ex:book1Info .
ex:book1Info dc:title "SPARQL by Example" ;
             dc:creator "John Doe" .

ex:book2 ex:hasInformation ex:book2Info .
ex:book2Info dc:title "Semantic Web Programming" ;
             dc:creator "Jane Smith" .

.
```

ASK

- **Uitleg:** Geeft een booleaanse waarde terug (true/false) om te bepalen of de query condities voldoen.
- **Voorbeeld query:**

```
ASK {
  ex:book1 ex:hasInformation ex:book1Info .
}
```

- **Resultaat:** true

INSERT

- **Uitleg:** Wordt gebruikt om nieuwe data toe te voegen aan de RDF store.
- **Voorbeeld query:**

```
INSERT DATA {
  :Alice :hasName "Alice" .
}
```

- **Triples:**

```
:Alice :hasName "Alice" .
```

WHERE

- **Uitleg:** Specificeert de patronen en condities waaraan de data moet voldoen voor de query.
- **Voorbeeld query:**

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o.
}
```

- **Resultaten:**

?s	?p	?o
:John	:hasAge	"28"
:Jane	:hasName	"Jane"
:Bob	:hasAge	"35"
:Alice	:hasName	"Alice"

LIMIT

- **Uitleg:** Beperkt het aantal resultaten dat teruggegeven wordt door de query.
- **Voorbeeld query:**

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o.
}
LIMIT 1
```

- **Resultaten:**

?s	?p	?o
:John	:hasAge	"28"

ORDER BY

- **Uitleg:** Orden de resultaten op basis van specifieke variabelen.
- **Voorbeeld query:**

```
SELECT ?s ?age
WHERE {
  ?s :hasAge ?age.
}
ORDER BY DESC(?age)
LIMIT 2
```

- **Resultaten:**

?s	?age
:Bob	"35"
:John	"28"

FILTER

- **Uitleg:** Filtert resultaten gebaseerd op condities.
- **Voorbeeld query:**

```

SELECT ?s ?age
WHERE {
  ?s :hasAge ?age.
  FILTER(?age > 30)
}

```

- **Resultaten:**

?s	?age
:Bob	"35"

FROM

- **Uitleg:** Specificeert de dataset waarvan de data geselecteerd moet worden.
- **Voorbeeld query:**

```

SELECT ?s ?p ?o
FROM <http://example.org/graph>
WHERE {
  ?s ?p ?o.
}

```

CONCAT

- **Uitleg:** Concateneert strings samen.
- **Voorbeeld query:**

```

SELECT (CONCAT("Hello, ", ?name) AS ?greeting)
WHERE {
  ?s :hasName ?name.
}

```

- **Resultaten:**

?greeting
"Hello, Alice"

BIND

- **Uitleg:** Bindt een expressie aan een variabele binnen de query.
- **Voorbeeld query:**

```

SELECT ?s ?fullName
WHERE {
  ?s :firstName ?firstName.
  ?s :lastName ?lastName.
  BIND(CONCAT(?firstName, " ", ?lastName) AS ?fullName)
}

```

- **Resultaten:**

?s	?fullName
:John	"John Doe"

CONTAINS

- **Uitleg:** Controleert of een string een substring bevat.
- **Voorbeeld query:**

```

SELECT ?s
WHERE {
  ?s :hasName ?name.
  FILTER(CONTAINS(?name, "Jane"))
}

```

• Resultaten:

?s
:Jane