Apache Jena

- Home
- Download
- Learn
    - Tutorials
    - Overview
    - RDF core API tutorial
    - SPARQL tutorial
    - Manipulating SPARQL using ARQ
    - Using Jena with Eclipse
    - How-To's
    - 
    - References
    - Overview
    - Javadoc
    - RDF API
    - RDF I/O
    - ARQ (SPARQL)
    - RDF Connection - SPARQL API
    - Elephas - tools for RDF on Hadoop
    - Text Search
    - TDB
    - SDB
    - SPARQL over JDBC
    - Fuseki
    - Permissions
    - Assembler
    - Ontology API
    - Inference API
    - Command-line tools
    - Extras
- Javadoc
    - Jena Core
    - ARQ
    - TDB
    - Fuseki
    - Elephas
    - Text Search
    - Spatial Search
    - Permissions
    - JDBC
    - All Javadoc
- Ask
- Get involved
    - Contribute
    - Report a bug
    - 
    - Project
    - About Jena
    - Roadmap
    - Architecture
    - Project team

  - Related projects
  - 
  - ASF
  - Apache Software Foundation
  - License
  - Thanks
  - Become a Sponsor
  - Security
- Improve this Page

1. TUTORIALS
2. SPARQL RESULTS

# Producing Result Sets

SPARQL has four result forms:

- SELECT – Return a table of results.
- CONSTRUCT – Return an RDF graph, based on a template in the query.
- DESCRIBE – Return an RDF graph, based on what the query processor is configured to return.
- ASK – Ask a boolean query.

The SELECT form directly returns a table of solutions as a result set, while DESCRIBE and CONSTRUCT use the outcome of matching to build RDF graphs.

## Solution Modifiers

Pattern matching produces a set of solutions. This set can be modified in various ways:

- Projection - keep only selected variables
- OFFSET/LIMIT - chop the number solutions (best used with ORDER BY)
- ORDER BY - sorted results
- DISTINCT - yield only one row for one combination of variables and values.

The solution modifiers OFFSET/LIMIT and ORDER BY always apply to all result forms.

### OFFSET and LIMIT

A set of solutions can be abbreviated by specifying the offset (the start index) and the limit (the number of solutions) to be returned. Using LIMIT alone can be useful to ensure not too many solutions are returned, to restrict the effect of some unexpected situation.  LIMIT and OFFSET can be used in conjunction with sorting to take a defined slice through the solutions found.

### ORDER BY

SPARQL solutions are sorted by expression, including custom functions.

```
ORDER BY ?x ?y
```

```
ORDER BY DESC(?x)
```

```
ORDER BY x:func(?x)  # Custom sorting condition
```

### DISTINCT

The SELECT result form can take the DISTINCT modifier which ensures that no two solutions returned are the same - this takes place after projection to the requested variables.

## SELECT

The `SELECT` result form is a projection, with DISTINCT applied, of the solution set. `SELECT` identifies which named variables are in the result set. This may be "`*`" meaning "all named variables" (blank nodes in the query act like variables for matching but are never returned).

## CONSTRUCT

CONSTRUCT builds an RDF based on a graph template. The graph template can have variables which are bound by a WHERE clause. The effect is to calculate the graph fragment, given the template, for each solution from the WHERE clause, after taking into account any solution modifiers. The graph fragments, one per solution, are merged into a single RDF graph which is the result.

Any blank nodes explicitly mentioned in the graph template are created afresh for each time the template is used for a solution.

## DESCRIBE

The CONSTRUCT form, takes an application template for the graph results. The DESCRIBE form also creates a graph but the form of that graph is provided the query processor, not the application. For each URI found, or explicitly mentioned in the DESCRIBE clause, the query processor should provide a useful fragment of RDF, such as all the known details of a book. ARQ allows domain-specific description handlers to be written.

## ASK

The ASK result form returns a boolean, true of the pattern matched otherwise false.

Return to index

Copyright © 2011–2019 The Apache Software Foundation, Licensed under the Apache License, Version 2.0.

Apache Jena, Jena, the Apache Jena project logo, Apache and the Apache feather logos are trademarks of The Apache Software Foundation.