**Course: Advance Bio Informatics**

**Module Title: Bio Query Language**

**Module No: 79**

**Bio Query Language**

* RDQL
* SPARQL

**RDQL: RDF Data Query Language**

**Basic Syntax:**

SELECT vars

FROM documents

WHERE Expressions

AND Filters

USING Namespace declarations

**Clauses**

Identifies the variables to be returned to the application.

If not all the variables are needed by the application, then specifying the required results can reduce the amount of memory needed for the results set as well as providing information to a query optimizer.

**FROM:** specifies the model by URI.

**WHERE:** This specifies graph pattern as a list of Triple patterns.

Constraints RDF triples (subject, predicate, object)

**AND:** Specifies the

Boolean expressions, indicates constraints that RDQL variables must follow.

**USING:**

A way to shorten the length of URIs.

Helps make for an easier to understand syntax.

This is not a namespace mechanism;

Simple abbreviation mechanism for long URIs by defining a string prefix.

**SPARQL**

Query language for getting information from RDF graphs.

Extract information in form of URIs, blank nodes, plain and typed literals.

Extract RDF subgraphs, construct new RDF graphs.

Based on information in the queried graphs matching graph patterns

variables – global scope;

indicated by ‘?‘ or ‘$‘

query terms – based on

Turtle syntax terms delimited by "<>" are *relative URI references*

data description format - Turtle

**RDF Dataset**

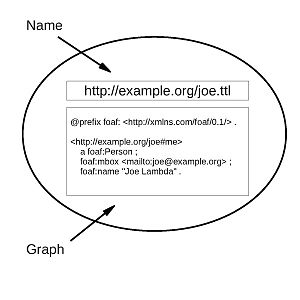
Multiple RDF graphs:

Graph queries that involve information from more than one graph .

**RDF Dataset** in SPARQL **background graph**, which does not have a name, and zero or more named graphs, identified by URI reference.

Named Graphs

Background Graphs



**Query Forms**

**SELECT:** returns all, or a subset of the variables bound in a query pattern match **formats:** XML or RDF/XML.

**CONSTRUCT:** returns an RDF graph constructed by substituting variables in a set of triple templates.

**DESCRIBE:** returns an RDF graph that describes the resources found.

**ASK:** returns whether a query pattern matches or not.

**JENA**

ARQ implements SPARQL.

Parses queries expressed in RDQL or its own internal language.

Not yet part of standard Jena distribution;

Availbale from either Jena‘s CVS repository or as a self-contained download.

**Twinkle**

Simple Java interface that wraps the ARQ SPARQL Processor library (the add -on to Jena).

**Redland**

Set of free software packages that provide support for RDF, including querying with RDQL and SPARQL using the Rasqal RDF Query Library.