

ID	Query
dq1	<p><i>SPARQL:</i> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT DISTINCT ?var1 WHERE { &lt;http://dbpedia.org/resource/%C3%89cole_nationale_sup%C3%A9rieure_d'ing%C3%A9nieurs_de_Poitiers&gt; rdf:type ?var1 . }</p> <p><i>Gremlin:</i> g.V("_URL_", "http://dbpedia.org/resource/%C3%89cole_nationale_sup%C3%A9rieure_d'ing%C3%A9nieurs_de_Poitiers").out('http://www.w3.org/1999/02/22-rdf-syntax-ns#type')._URL_.dedup().count()</p>
dq2	<p><i>SPARQL:</i> PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpedia-owl: &lt;http://dbpedia.org/ontology/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX dbpedia-prop: &lt;http://dbpedia.org/property/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT ?var4 ?var8 ?var10 WHERE { ?var5 dbpedia-owl:thumbnail ?var4 ; rdf:type dbpedia-owl:Person ; rdfs:label "Montreal Carabins"@en ; dbpedia-prop:pageurl ?var8 . OPTIONAL { ?var5 foaf:homepage ?var10 . } }</p> <p><i>Gremlin:</i> t1=new Table(); t2=new Table(); var5=[]; g.V("_URL_", "http://dbpedia.org/ontology/Person").in('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('http://www.w3.org/2000/01/rdf-schema#label', "Montreal Carabins"@en).aggregate(var5).as('var5').out('http://dbpedia.org/ontology/thumbnail').as('var4').back(1).out('http://dbpedia.org/property/pageurl').as('var8').table(t1).iterate(); var5._().as('var5').out('http://xmlns.com/foaf/0.1/homepage').as('var10').table(t2).iterate(); [t1.size(), t2.size()];</p>
dq3	<p><i>SPARQL:</i> PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; SELECT ?var5 ?var6 ?var9 ?var8 ?var4 WHERE { :Treehouse_of_Horror_XX ?var5 ?var6 . ?var6 foaf:name ?var8 . } UNION { ?var9 ?var5 :Treehouse_of_Horror_XX ; foaf:name ?var4 . }</p> <p><i>Gremlin:</i> t1=new Table(); t2=new Table(); g.V("_URL_", "http://dbpedia.org/resource/Treehouse_of_Horror_XX").outE.label.as('var5').back(1).inV.as('var6').has('http://xmlns.com/foaf/0.1/name', T.neq, null).property('http://xmlns.com/foaf/0.1/name').as('var8').table(t1).iterate(); g.V("_URL_", "http://dbpedia.org/resource/Treehouse_of_Horror_XX").inE.label.as('var5').outV.as('var9').has('http://xmlns.com/foaf/0.1/name', T.neq, null).property('http://xmlns.com/foaf/0.1/name').as('var4').table(t2).iterate(); [t1.size(), t2.size()];</p>
dq4	<p><i>SPARQL:</i> PREFIX rdbs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpp: &lt;http://dbpedia.org/property/&gt; PREFIX dbpr: &lt;http://dbpedia.org/resource/&gt; PREFIX dbpo: &lt;http://dbpedia.org/ontology/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT DISTINCT ?var3 ?var4 ?var5 WHERE { ?var3 dbpp:series dbpr:South_Park ; foaf:name ?var4 ; rdfs:comment ?var5 ; rdf:type dbpo:Work . } UNION { ?var3 dbpp:series ?var8 . ?var8 dbpp:redirect dbpr:South_Park . ?var3 foaf:name ?var4 ; rdfs:comment ?var5 ; rdf:type dbpo:Work . }</p> <p><i>Gremlin:</i> t1=new Table(); t2=new Table(); g.V("_URL_", "http://dbpedia.org/resource/South_Park").in('http://dbpedia.org/property/series').as('var3').out('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('_URL_', "http://dbpedia.org/ontology/Work").back(2).has('http://xmlns.com/foaf/0.1/name', T.neq, null).property('http://xmlns.com/foaf/0.1/name').as('var4').back(1).has('http://www.w3.org/2000/01/rdf-schema#comment', T.neq, null).property('http://www.w3.org/2000/01/rdf-schema#comment').as('var5').table(t1).iterate(); g.V("_URL_", "http://dbpedia.org/resource/South_Park").in('http://dbpedia.org/property/redirect').in('http://dbpedia.org/property/series').as('var3').out('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('_URL_', "http://dbpedia.org/ontology/Work").back(2).has('http://xmlns.com/foaf/0.1/name', T.neq, null).property('http://xmlns.com/foaf/0.1/name').as('var4').back(1).has('http://www.w3.org/2000/01/rdf-schema#comment', T.neq, null).property('http://www.w3.org/2000/01/rdf-schema#comment').as('var5').table(t2).iterate(); [t1.size(), t2.size()];</p>
dq5	<p><i>SPARQL:</i> PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX rdbs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpedia: &lt;http://dbpedia.org/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX skos: &lt;http://www.w3.org/2004/02/skos/core#&gt; SELECT DISTINCT ?var0 ?var1 ?var2 ?var3 ?var5 ?var6 ?var7 ?var10 WHERE { ?var0 rdbs:comment ?var1 ; dbpedia2:pageurl &lt;http://www.usds.org/page.cfm?p=276&gt; . OPTIONAL { ?var0 skos:subject ?var6 . } OPTIONAL { ?var0 dbpedia2:industry ?var5 . } OPTIONAL { ?var0 dbpedia2:location ?var2 . } OPTIONAL { ?var0 dbpedia2:locationCountry ?var3 . } OPTIONAL { ?var0 dbpedia2:locationCity ?var9 ; dbpedia2:manufacturer ?var0 . } OPTIONAL { ?var0 dbpedia2:products ?var11 ; dbpedia2:model ?var0 . } OPTIONAL { ?var0 &lt;http://www.georss.org/georss/point&gt; ?var10 . } OPTIONAL { ?var0 rdf:type ?var7 . }</p> <p><i>Gremlin:</i> t1=new Table(); t2=new Table(); t3=new Table(); t5=new Table(); t6=new Table(); t7=new Table(); t10=new Table(); var0=[]; g.V("_URL_", "http://www.usds.org/page.cfm?p=276").in('http://dbpedia.org/property/pageurl').aggregate(var0).as('var0').has('http://www.w3.org/2000/01/rdf-schema#comment', T.neq, null).property('http://www.w3.org/2000/01/rdf-schema#comment').as('var1').table(t1).iterate(); var0._().as('var0').out('http://www.w3.org/2004/02/skos/core#subject').as('var6').table(t6).iterate(); var0._().as('var0').has('http://dbpedia.org/property/industry', T.neq, null).property('http://dbpedia.org/property/industry').as('var5').table(t5).iterate(); var0._().as('var0').out('http://dbpedia.org/property/location').as('var2').table(t2).iterate(); var0._().as('var0').out('http://dbpedia.org/property/locationCountry').as('var3').table(t3).iterate(); var0._().as('var0').has('http://www.georss.org/georss/point', T.neq, null).property('http://www.georss.org/georss/point').as('var10').table(t10).iterate(); var0._().as('var0').out('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').as('var7').table(t7).iterate(); [t1.size(), t2.size(), t3.size(), t5.size(), t6.size(), t7.size(), t10.size()];</p>
dq6	<p><i>SPARQL:</i> PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX dbpedia: &lt;http://dbpedia.org/ontology/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT ?var2 ?var4 WHERE { { ?var2 rdf:type dbpedia:Settlement ; dbpedia2:population ?var4 . FILTER ( xsd:integer(?var4) &gt; 54 ) } UNION { ?var2 rdf:type dbpedia:Settlement ; dbpedia2:populationUrban ?var4 . FILTER ( xsd:integer(?var4) &gt; 54 ) } }</p> <p><i>Gremlin:</i> x=[]; t1=new Table(); t2=new Table(); g.V("_URL_", "http://dbpedia.org/ontology/Settlement").in('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('http://dbpedia.org/property/population', 100).as('var2').property('http://dbpedia.org/property/population').as('var4').table(t1).iterate(); g.V("_URL_", "http://dbpedia.org/ontology/Settlement").in('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('http://dbpedia.org/property/populationUrban', 100).as('var2').property('http://dbpedia.org/property/populationUrban').as('var4').table(t2).iterate(); [t1.size(), t2.size()];</p>
dq7	<p><i>SPARQL:</i> PREFIX rdbs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX dbpedia: &lt;http://dbpedia.org/ontology/&gt; PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; SELECT * WHERE { ?var2 rdf:type dbpedia:Settlement ; rdfs:label "Djanet"@en . ?var6 rdf:type dbpedia:Airport . { ?var6 dbpedia:city ?var2 . } UNION { ?var6 dbpedia:location ?var2 . } { ?var6 dbpedia2:iata ?var5 . } UNION { ?var6 dbpedia:iataLocationIdentifier ?var5 . } OPTIONAL { ?var6 foaf:homepage ?var6_home . } OPTIONAL { ?var6 dbpedia2:nativeName ?var6_name . }</p> <p><i>Gremlin:</i> t1=new Table(); t2=new Table(); t3=new Table(); var6=[]; g.V("_URL_", "http://dbpedia.org/ontology/Settlement").in('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('http://www.w3.org/2000/01/rdf-schema#label', "Djanet"@en).as('var2').in('http://dbpedia.org/ontology/city', 'http://dbpedia.org/ontology/location').dedup().out('http://www.w3.org/1999/02/22-rdf-syntax-ns#type').has('_URL_', "http://dbpedia.org/ontology/Airport").back(2).aggregate(var6).as('var6').copySplit(_).has('http://dbpedia.org/property/iata', T.neq, null).property('http://dbpedia.org/property/iata', _).has('http://dbpedia.org/ontology/iataLocationIdentifier', T.neq, null).property('http://dbpedia.org/ontology/iataLocationIdentifier').exhaustMerge.dedup().as('var5').table(t1).iterate(); var6._().as('var6').out('http://xmlns.com/foaf/</p>

	<pre> 0.1/homepage').as('var6_home').table(t2).iterate(); var6_().as('var6').has('http://dbpedia.org/property/nativename',T.neq, null).property('http://dbpedia.org/property/nativename').as('var6_name').table(t3).iterate(); [t1.size(),t2.size(),t3.size()]; </pre>
dq8	<pre> SPARQL: PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; PREFIX foaf: &lt;http:// xmlns.com/foaf/0.1/&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX dbpedia: &lt;http://dbpedia. org/ontology/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT DISTINCT ?var7 WHERE { ?var3 foaf:homepage ?var7 ; rdf:type dbpedia:SoccerPlayer ; dbpedia2:position ?var16 ; dbpedia2:clubs ?var8 . ?var8 dbpedia:capacity ?var9 . ?var3 dbpedia:birthPlace ?var31 . ?var31 ?var33 ?var34 . OPTIONAL { ?var3 dbpedia:number ?var35 . } FILTER ( ( ( ?var33 = dbpedia2:populationEstimate )    ( ?var33 = dbpedia2:populationCensus ) )    ( ?var33 = dbpedia2:statPop ) ) FILTER ( xsd:integer(?var34) &gt; 7949 ) FILTER ( xsd:integer(?var9) &lt; 1789000 ) FILTER ( ( ( ?var16 = "Goalkeeper"@en )    ( ?var16 = :Goalkeeper_%28association_football%29 ) )    ( ?var16 = :Goalkeeper_%28 football%29 ) ) } Gremlin: g.V("_URL_", "http://dbpedia.org/ontology/SoccerPlayer").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type") .as("var3").or(_().has("http://dbpedia.org/property/position", "Goalkeeper"@en), _().out("http://dbpedia.org/property/ position").has("_URL_", "http://dbpedia.org/resource/Goalkeeper_%28association_football%29"), _().out("http://dbpedia. org/property/position").has("_URL_", "http://dbpedia.org/resource/Goalkeeper_%28football%29")).back(1).out("http:// dbpedia.org/property/clubs").has("http://dbpedia.org/ontology/capacity", T.lt, 1789000).back(2).out("http://dbpedia.org/ ontology/birthPlace").or(_().has("http://dbpedia.org/property/populationEstimate", T.neq, null).filter{it.'http://dbpedia. org/property/populationEstimate'.class==java.lang.Integer &amp;&amp; it.'http://dbpedia.org/property/populationEstimate'&gt; 7949}, _().has("http://dbpedia.org/property/populationCensus", T.neq, null).filter{it.'http://dbpedia.org/property/ populationCensus'.class== java.lang.Integer &amp;&amp; it.'http://dbpedia.org/property/populationCensus'&gt;7949}), _().has( 'http://dbpedia.org/property/statPop', T.neq, null).filter{it.'http://dbpedia.org/property/statPop'.class==java.lang. Integer &amp;&amp; it.'http://dbpedia.org/property/statPop'&gt;7949}).back(2).out("http://xmlns.com/foaf/0.1/homepage") .dedup().count(); </pre>
dq9	<pre> SPARQL: PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; SELECT DISTINCT ?var3 ?var4 ?var2 WHERE { { :United_Farmers_of_Alberta dbpedia2:subsid ?var3 . OPTIONAL { ?var2 :United_Farmers _of_Alberta dbpedia2:parent . } OPTIONAL { :United_Farmers_of_Alberta dbpedia2:divisions ?var4 . } UNION { ?var2 :United_Farmers_of_Alberta dbpedia2:parent . OPTIONAL { :United_Farmers_of_Alberta dbpedia2:subsid ?var3 . } OPTIONAL { :United_Farmers_of_Alberta dbpedia2:divisions ?var4 . } } UNION { :United_Farmers_of_Alberta dbpedia2: divisions ?var4 . OPTIONAL { :United_Farmers_of_Alberta dbpedia2:subsid ?var3 . } OPTIONAL { ?var2 :United_ Farmers_of_Alberta dbpedia2:parent . } } } Gremlin: t1=new Table(); t2=new Table(); t3=new Table(); g.V("_URL_", "http://dbpedia.org/resource/United_Farmers_of_ Alberta").out("http://dbpedia.org/property/subsid").dedup().as("var3").table(t1); g.V("_URL_", "http://dbpedia.org/ property/parent").in("http://dbpedia.org/resource/United_Farmers_of_Alberta").dedup().as("var2").table(t2); g.V("_URL_", "http://dbpedia.org/resource/United_Farmers_of_Alberta").out("http://dbpedia.org/property/divisions").dedup().as("var4") .table(t3); [t1.size(),t2.size(),t3.size()]; </pre>
dq10	<pre> SPARQL: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpedia-owl: &lt;http://dbpedia.org/ontology/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT DISTINCT ?var5 WHERE { ?var2 rdf:type dbpedia-owl:Person ; dbpedia-owl:nationality ?var4 . ?var4 rdfs:label ?var5 . ?var2 rdfs:label "Simone Cecchetti"@en . FILTER ( lang(?var5) = "en" ) } Gremlin: g.V("_URL_", "http://dbpedia.org/ontology/Person").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").has("http: //www.w3.org/2000/01/rdf-schema#label", "Simone Cecchetti"@en).out("http://dbpedia.org/ontology/nationality") .property("http://www.w3.org/2000/01/rdf-schema#label").filter{it.endsWith("@en")}.dedup().count(); </pre>
dq11	<pre> SPARQL: PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; SELECT * WHERE { \$100,000.Challenge rdfs:comment ?var0 . FILTER ( lang(?var0) = "en" ) } UNION { \$100,000.Challenge foaf:depiction ?var1 . } UNION { \$100,000.Challenge foaf:homepage ?var2 . } } Gremlin: t1=new Table(); t2=new Table(); t3=new Table(); g.V("_URL_", "http://dbpedia.org/resource/\$100,000.Challenge") .property("http://www.w3.org/2000/01/rdf-schema#comment").filter{it.endsWith("@en")}.as("var0").table(t1).iterate(); g.V("_URL_", "http://dbpedia.org/resource/\$100,000.Challenge").out("http://xmlns.com/foaf/0.1/depiction").as("var1") .table(t2).iterate(); g.V("_URL_", "http://dbpedia.org/resource/\$100,000.Challenge").out("http://xmlns.com/foaf/0.1/ homepage").as("var2").table(t3).iterate(); [t1.size(),t2.size(),t3.size()]; </pre>
dq12	<pre> SPARQL: PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX skos: &lt;http://www.w3.org/2004/02/skos/core#&gt; SELECT ?var6 ?var8 ?var10 ?var4 WHERE { ?var4 skos:subject :Tropical_fish ; foaf:name ?var6 . OPTIONAL { ?var4 rdfs:comment ?var8 . FILTER ( lang(?var8) = "en" ) } } Gremlin: t1=new Table(); t2=new Table(); var4=[]; g.V("_URL_", "http://dbpedia.org/resource/Tropical_fish").in("http://www.w3. org/2004/02/skos/core#subject").aggregate(var4).as("var4").has("http://xmlns.com/foaf/0.1/name", T.neq, null).property( 'http://xmlns.com/foaf/0.1/name').as("var6").table(t1).iterate(); var4_().as("var4").has("http://www.w3.org/2000/01/ rdf-schema#comment", T.neq, null).property("http://www.w3.org/2000/01/rdf-schema#comment").filter{it.endsWith( "@en")}.as("var8").table(t2).iterate(); [t1.size(),t2.size()]; </pre>
dq13	<pre> SPARQL: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbo: &lt;http://dbpedia.org/ontology/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT DISTINCT ?var2 ?var3 WHERE { ?var2 rdf:type dbo: Wrestler ; rdfs:label ?var3 . FILTER regex(str(?var3), "sosay"@en, "i") } Gremlin: t=new Table(); g.V("_URL_", "http://dbpedia.org/ontology/Wrestler").in("http://www.w3.org/1999/02/22-rdf-syntax-ns #type").as("var2").has("http://www.w3.org/2000/01/rdf-schema#label", T.neq, null).property("http://www.w3.org/2000/ 01/rdf-schema#label").filter{it.equalsIgnoreCase("sosay"@en)}.as("var3").table(t).iterate(); [t.size()]; </pre>
dq14	<pre> SPARQL: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpedia2: &lt;http://dbpedia.org/property/&gt; PREFIX dbpedia: &lt;http://dbpedia.org/ontology/&gt; SELECT ?var3 ?var4 ?var5 WHERE { { dbpedia2:imageName ?var3 ?var4 . FILTER ( ( ( ( str(?var3) = rdfs:label ) &amp;&amp; ( lang(?var4) = "en" ) )    ( str(?var3) = dbpedia:abstract ) &amp;&amp; ( lang(?var4) = "en" ) ) )    ( ( str(?var3) = rdfs:comment ) &amp;&amp; ( lang(?var4) = "en" ) ) )    ( ( str(?var3) != dbpedia: abstract ) &amp;&amp; ( str(?var3) != rdfs:comment ) ) &amp;&amp; ( str(?var3) != rdfs:label ) ) } UNION { ?var5 ?var3 dbpedia2: imageName . FILTER ( ( str(?var3) = dbpedia:owner )    ( str(?var3) = dbpedia2:redirect ) ) } } Gremlin: t1=new Table(); t2=new Table(); g.V("_URL_", "http://dbpedia.org/property/imageName").outE.and(_().has('label', T.neq, "http://dbpedia.org/ontology/abstract"), _().has('label', T.neq, "http://www.w3.org/2000/01/rdf-schema# comment"), _().has('label', T.neq, "http://www.w3.org/2000/01/rdf-schema#label")).label.as("var3").back(1).inV._URL_ .as("var4").table(t1).iterate(); g.V("_URL_", "http://dbpedia.org/property/imageName").inE("http://dbpedia.org/ontology /owner", "http://dbpedia.org/property/redirect").label.as("var3").back(1).outV._URL_.as("var5").table(t2).iterate(); [t1.size(),t2.size()]; </pre>
dq15	<pre> SPARQL: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; SELECT ?var1 WHERE { { ?var1 rdfs:label "1981 albums" "@en . } UNION { ?var1 rdfs:label "1981 albums"@en . } FILTER ( ( ( ( regex(str(?var1), "http://dbpedia.org/resource /")    regex(str(?var1), "http://dbpedia.org/ontology/") )    regex(str(?var1), "http://www.w3.org/2002/07/owl") )    regex(str(?var1), "http://www.w3.org/2001/XMLSchema") )    regex(str(?var1), "http://www.w3.org/2000/01/rdf- schema") )    regex(str(?var1), "http://www.w3.org/1999/02/22-rdf-syntax-ns") ) } } </pre>

	<p><i>Gremlin</i>: <code>g.V().has('http://www.w3.org/2000/01/rdf-schema#label','1981 albums"@en').has('_URL_',T.neq,null)._URL_.or(_).filter{it.startsWith("http://dbpedia.org/resource/")}, _().filter{it.startsWith("http://dbpedia.org/ontology/")}, _().filter{it.startsWith("http://www.w3.org/2002/07/owl")}, _().filter{it.startsWith("http://www.w3.org/2001/XMLSchema")}, _().filter{it.startsWith("http://www.w3.org/2000/01/rdf-schema")}, _().filter{it.startsWith("http://www.w3.org/1999/02/22-rdf-syntax-ns")}.back(2).count();</code></p>
dq16	<p><i>SPARQL</i>: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX dbpedia-owl: &lt;http://dbpedia.org/ontology/&gt; PREFIX foaf: &lt;http://xmlns.com/foaf/0.1/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT ?var4 WHERE { ?var2 rdf:type dbpedia-owl:Person ; rdfs:label "Dan Bricklin"@en ; dbpedia-owl:thumbnail ?var4 . }</p> <p><i>Gremlin</i>: <code>g.V('_URL_', "http://dbpedia.org/ontology/Person").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").has("http://www.w3.org/2000/01/rdf-schema#label", "Dan Bricklin"@en).out("http://dbpedia.org/ontology/thumbnail").count();</code></p>
dq17	<p><i>SPARQL</i>: PREFIX : &lt;http://dbpedia.org/resource/&gt; PREFIX dbpedia-owl: &lt;http://dbpedia.org/ontology/&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT * WHERE { ?var1 rdf:type dbpedia-owl:Organisation . ?var2 dbpedia-owl:foundationPlace :Netherlands . ?var4 dbpedia-owl:developer ?var2 ; rdf:type dbpedia-owl:Work . }</p> <p><i>Gremlin</i>: <code>t1=new Table(); t2=new Table(); g.V("_URL_", "http://dbpedia.org/ontology/Organisation").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").as("var1").table(t1); g.V("_URL_", "http://dbpedia.org/resource/Netherlands").in("http://dbpedia.org/ontology/foundationPlace").as("var2").in("http://dbpedia.org/ontology/developer").as("var4").out("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").has("_URL_", "http://dbpedia.org/ontology/Work").table(t2); [t1.size(),t2.size()];</code></p>
dq18	<p><i>SPARQL</i>: PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX dbpprop: &lt;http://dbpedia.org/property/&gt; SELECT ?var0 ?var1 ?var2 ?var3 WHERE { ?var6 rdf:type &lt;http://dbpedia.org/ontology/Artist&gt; ; dbpprop:name ?var0 ; dbpprop:pages ?var1 ; dbpprop:isbn ?var2 ; dbpprop:author ?var3 . }</p> <p><i>Gremlin</i>: <code>t1=new Table(); t2=new Table(); t3=new Table(); t4=new Table(); var6=[]; g.V("_URL_", "http://dbpedia.org/ontology/Artist").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").aggregate(var6).as("var6").has("http://dbpedia.org/property/name", T.neq,null).property("http://dbpedia.org/property/name").as("var0").table(t1).iterate(); var6._().dedup().as("var6").has("http://dbpedia.org/property/pages", T.neq,null).property("http://dbpedia.org/property/pages").as("var1").table(t2).iterate(); var6._().dedup().as("var6").has("http://dbpedia.org/property/isbn", T.neq,null).property("http://dbpedia.org/property/isbn").as("var2").table(t3).iterate(); var6._().dedup().as("var6").has("http://dbpedia.org/property/author", T.neq,null).property("http://dbpedia.org/property/author").as("var3").table(t4).iterate(); [t1.size(),t2.size(),t3.size(),t4.size()];</code></p>
dq19	<p><i>SPARQL</i>: PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; SELECT DISTINCT ?var7 ?var4 ?var6 ?var5 WHERE { &lt;http://dbpedia.org/resource/Charity_Land&gt; ?var4 ?var5 . OPTIONAL { ?var5 rdfs:label ?var6 . } }</p> <p><i>Gremlin</i>: <code>t1=new Table(); t2=new Table(); var4=[]; var5=[]; g.V("_URL_", "http://dbpedia.org/resource/Charity_Land").outE("label,aggregate(var4).as('var4').back(2).inV.aggregate(var5)._URL_.as('var5').table(t1).iterate(); var5._().URL_.as('var5').back(1).has('http://www.w3.org/2000/01/rdf-schema#label', T.neq,null).property('http://www.w3.org/2000/01/rdf-schema#label').as('var6').table(t2).iterate(); [t1.size(),t2.size()];</code></p>
dq20	<p><i>SPARQL</i>: SELECT ?var2 WHERE { ?var3 &lt;http://xmlns.com/foaf/0.1/homepage&gt; ?var2 ; &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#type&gt; &lt;http://dbpedia.org/ontology/TennisLeague&gt; . }</p> <p><i>Gremlin</i>: <code>g.V("_URL_", "http://dbpedia.org/ontology/TennisLeague").in("http://www.w3.org/1999/02/22-rdf-syntax-ns#type").out("http://xmlns.com/foaf/0.1/homepage").count();</code></p>
lq1	<code>y=new HashSet(); g.V[i+1..i+15959].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq2	<code>y=new HashSet(); g.V[i+1..i+15959].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq3	<code>y=new HashSet(); g.V[i+1..i+15959].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq4	<code>y=new HashSet(); g.V[i+1..i+100].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq5	<code>y=new HashSet(); g.V[i+1..i+1000].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq6	<code>y=new HashSet(); g.V[i+1..i+10000].aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).in("http://dbpedia.org/ontology/isPartOf").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq7	<code>y=new HashSet(); g.V[j+1].aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq8	<code>y=new HashSet(); g.V[j+1].aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq9	<code>y=new HashSet(); g.V[j+1].aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).iterate(); y.size();</code>
lq10	<code>y=new HashSet(); g.V[j+1..j+10].aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).out("http://dbpedia.org/ontology/team").dedup().in("http://dbpedia.org/ontology/team").dedup().except(y).aggregate(y).iterate(); y.size();</code>

	<code>team').dedup().except(y.aggregate(y).out('http://dbpedia.org/ontology/team').dedup().in('http://dbpedia.org/ontology/team').dedup().except(y.aggregate(y).iterate()); y.size());</code>
lq11	<code>y=new HashSet(); g.V[j+1..j+100].aggregate(y).out('http://dbpedia.org/ontology/team').dedup().in('http://dbpedia.org/ontology/team').dedup().except(y.aggregate(y).out('http://dbpedia.org/ontology/team').dedup().in('http://dbpedia.org/ontology/team').dedup().except(y.aggregate(y).iterate());y.size())</code>
lb1	<code>(add node) def temps = [?] as byte[]; v=g.addVertex(null,ntype:?,ntime:?,nversion:?,ndata:new String(temps)); v.id;</code>
lb2	<code>(update node) def temps = [?] as byte[]; v=g.getVertex(?); if(v!=null){v.ntype=?; v.nversion=?; v.ntime=?; v.ndata=new String(temps);}; v==null?-1:v.id;</code>
lb3	<code>(delete node) v=g.getVertex(?); if(v!=null &amp;&amp; v.ntype==?) g.removeVertex(v); else v=null; v==null?-1:v.id;</code>
lb4	<code>(get node) v=g.getVertex(?); (v==null    v.ntype!=2048)?[]:[v.id, v.ntype, v.nversion, v.ntime, v.ndata];</code>
lb5	<code>(add link) def temps = [?] as byte[]; x=[]; v1=g.getVertex(?); v2=g.getVertex(?); e=null; e1=null; if(v1!=null&amp;&amp;v2!=null){pe=v1.outE.filter{it.getInVertex()==v2 &amp;&amp; it.label=='?'}.fill(x); e=(x.size()&gt;0)?x[0]:null;}; if(e!=null){e.etime=?; e.eversion=?; e.evisibility=?; e.edata=new String(temps);}; else if(v1!=null&amp;&amp;v2!=null){e1 = g.addEdge(null, v1, v2, '?', [evisibility:?, etime:?, eversion:?, edata:new String(temps)]);}; e1==null?(e==null?false:true):true;</code>
lb6	<code>(delete link) x=[]; v1=g.getVertex(?); v2=g.getVertex(?); e=null; ret=-1; if(v1!=null&amp;&amp;v2!=null){pe=v1.outE.filter{it.getInVertex()==v2 &amp;&amp; it.label=='?'}.fill(x); e=(x.size()&gt;0)?x[0]:null;}; if(e!=null){if(e.evisibility==0){ret=0;}; else{e.evisibility=0;ret=1;};}; ret;</code>
lb7	<code>(update link) def temps = [?] as byte[]; x=[]; v1=g.getVertex(?); v2=g.getVertex(?); e=null; e1=null; if(v1!=null&amp;&amp;v2!=null){pe=v1.outE.filter{it.getInVertex()==v2 &amp;&amp; it.label=='?'}.fill(x); e=(x.size()&gt;0)?x[0]:null;}; if(e!=null){e.etime=?; e.eversion=?; e.evisibility=?; e.edata=new String(temps);}; else if(v1!=null&amp;&amp;v2!=null){e1 = g.addEdge(null, v1, v2, '?', [evisibility:?, etime:?, eversion:?, edata:new String(temps)]);}; e1==null?(e==null?true:false):false;</code>
lb8	<code>(count link) v1=g.getVertex(?);v1==null?0:v1.outE('???').filter{it.evisibility==1}.count();</code>
lb9	<code>(multiget link) x=[]; y=[]; def id2s = [?] as long[]; v1=g.getVertex(?); if(v1!=null) v1.outE('???').filter{it.getInVertex().id in id2s}.fill(x); for(i=0; i&lt;x.size(); i++) y.add([x[i].getInVertex().id, x[i].evisibility, x[i].eversion, x[i].etime, x[i].edata]); y;</code>
lb10	<code>(get link list) x=[]; y=[]; v1=g.getVertex(?); if(v1!=null) v1.outE('???').filter{it.evisibility==1}.fill(x); for(i=0; i&lt;x.size(); i++) y.add([x[i].getInVertex().id, x[i].evisibility, x[i].eversion, x[i].etime, x[i].edata]); y;</code>

Table 10: Benchmark queries used in the experiments. dq1-dq20: DBPedia Benchmark queries generated from SPARQL queries; lq1-lq11: DBPedia Benchmark long-path queries; lb1-lb10: LinkBench queries.