

## Documentation: Wikidata SPQRL and Neo4j Cypher queries

WIKIDATA SPQRL-Processor class	Neo4j Cypher
<pre> SELECT ?processor ?processorLabel ?ofsubClass ?ofsubClassLabel ?ofsubClass1 ?ofsubClassLabe2 ?subclass ?subclassLabel ?subclass1 ?subclass1Label ?hasUse ?hasUseLabel ?hasproperty ?haspropertyLabel ?haspart ?haspartLabel ?partUse ?partUseLabel WHERE {    ?processor rdfs:label ?processorLabel .   FILTER(LANG(?processorLabel) = "en")   {     SELECT ?ofsubClass ?ofsubClassLabel WHERE {       ?ofsubClass wdt:P279 wd:Q1466064 .       ?ofsubClass rdfs:label ?ofsubClassLabel .       FILTER(LANG(?ofsubClassLabel) = "en")     }     LIMIT 50   }   OPTIONAL{ ?processor wdt:P279 ?subclass .   ?subclass rdfs:label ?subclassLabel .   FILTER(LANG(?subclassLabel) = "en") }    OPTIONAL{ ?subclass wdt:P279 ?subclass1 .   ?subclass1 rdfs:label ?subclass1Label .   FILTER(LANG(?subclass1Label) = "en") }    OPTIONAL{     ?processor wdt:P366 ?hasUse .     ?hasUse rdfs:label ?hasUseLabel .     FILTER(LANG(?hasUseLabel) = "en")   }    OPTIONAL{     ?processor wdt:P1963 ?hasproperty .     ?hasproperty rdfs:label ?haspropertyLabel .     FILTER(LANG(?haspropertyLabel) = "en")   }   OPTIONAL{     ?processor wdt:P527 ?haspart .     ?haspart rdfs:label ?haspartLabel .     FILTER(LANG(?haspartLabel) = "en")   }    OPTIONAL {     ?haspart wdt:P366 ?partUse .     ?partUse rdfs:label ?partUseLabel .     FILTER(LANG(?partUseLabel) = "en")   }   }   FILTER(?processor = wd:Q1466064) } </pre>	<pre> CALL apoc.load.json("file:/Users/rutujajagtap/Downloads/ PROCESSORNew1.json") YIELD value  MERGE (p:processor {name: value.processorLabel}) MERGE (su:processor {name: value.ofsubClassLabel}) MERGE (su)-[:is_subclass_of]-&gt;(p)  MERGE (s:processor {name: value.subclassLabel}) MERGE (p)-[:subclass_of]-&gt;(s)  MERGE(s1:processor{name:value.subclass1Label}) MERGE(s)-[:subclass_of]-&gt;(s1)  MERGE (u:hasUse {name: value.hasUseLabel}) MERGE (p)-[:has_use]-&gt;(u) MERGE (pr:hasproperty {name: value.haspropertyLabel}) MERGE (p)-[:has_property]-&gt;(pr) MERGE (part:haspart {name: value.haspartLabel}) MERGE (p)-[:has_part]-&gt;(part)  FOREACH(ignoreMe IN CASE WHEN value.partUseLabel IS NOT NULL THEN [1] ELSE {} END   MERGE (partUse:partUse {name: value.partUseLabel}) MERGE (part)-[:has_use]-&gt;(partUse) )  MATCH (n) RETURN n  MATCH p=()-[r:has_property]-&gt;() RETURN p  MATCH p=()-[r:has_use]-&gt;() RETURN p LIMIT 25  MATCH p=()-[r:has_part]-&gt;() RETURN p LIMIT 25  MATCH p=()-[r:is_subclass_of]-&gt;() RETURN p  MATCH p=()-[r:subclass_of]-&gt;() RETURN p </pre>

WIKIDATA SPRQL-Motor car class	Neo4j Cypher
<pre> SELECT ?motorCar ?motorCarLabel ?ofsubClass ?ofsubClassLabel ?subclass ?subclassLabel ?subclass1 ?subclass1Label ?subclass2 ?subclass2Label ?hasUse ?hasUseLabel ?effect ?effectLabel ?part1 ?part1Label ?part2 ?part2Label  WHERE { ?motorCar rdfs:label ?motorCarLabel . FILTER(LANG(?motorCarLabel) = "en")  {   SELECT ?ofsubClass ?ofsubClassLabel WHERE {     ?ofsubClass wdt:P279 wd:Q1420 .     ?ofsubClass rdfs:label ?ofsubClassLabel .     FILTER(LANG(?ofsubClassLabel) = "en")   }   LIMIT 30 } ?motorCar wdt:P366 ?hasUse . ?hasUse rdfs:label ?hasUseLabel . FILTER(LANG(?hasUseLabel) = "en")  ?motorCar wdt:P1542 ?effect . ?effect rdfs:label ?effectLabel . FILTER(LANG(?effectLabel) = "en")  OPTIONAL{ ?motorCar wdt:P279 ?subclass . ?subclass rdfs:label ?subclassLabel . FILTER(LANG(?subclassLabel) = "en")  OPTIONAL{ ?subclass wdt:P279 ?subclass1 . ?subclass1 rdfs:label ?subclass1Label . FILTER(LANG(?subclass1Label) = "en") }  OPTIONAL{ ?subclass1 wdt:P279 ?subclass2 . ?subclass2 rdfs:label ?subclass2Label . FILTER(LANG(?subclass2Label) = "en") } } OPTIONAL{ ?motorCar wdt:P527 ?part1 . ?part1 rdfs:label ?part1Label . FILTER(LANG(?part1Label) = "en") } OPTIONAL { ?part1 wdt:P527 ?part2 . ?part2 rdfs:label ?part2Label . FILTER(LANG(?part2Label) = "en") }  FILTER(?motorCar = wd:Q1420) }</pre>	<pre> CALL apoc.load.json("file:/Users/rutujajagtap/ Downloads/MotorCarFinal.json") yield value  MERGE (m:auto{name:value.motorCarLabel}) MERGE (su:auto{name:value.ofsubClassLabel}) MERGE (m)-[:superClass_of]-&gt;(su) MERGE (s:auto{name:value.subclassLabel}) MERGE (m)-[:subClass_of]-&gt;(s) MERGE (s1:auto{name:value.subclass1Label}) MERGE (s)-[:subClass_of]-&gt;(s1)  MERGE (s2:auto{name:value.subclass2Label}) MERGE (s1)-[:subClass_of]-&gt;(s2)  MERGE (u:use {name: value.hasUseLabel}) MERGE (m)-[:has_use]-&gt;(u) MERGE (e:effect{name:value.effectLabel}) MERGE (m)-[:has_effect]-&gt;(e) MERGE (p1:part {name: value.part1Label}) MERGE (m)-[:has_part]-&gt;(p1) FOREACH( ignoreMe IN CASE WHEN value.part2Label IS NOT NULL THEN [1] ELSE {} END   MERGE (p2:part {name:value.part2Label}) MERGE (p1)-[:has_part]-&gt;(p2) )  MATCH p=()-[r:subClass_of]-&gt;() RETURN p  MATCH p=()-[r:superClass_of]-&gt;() RETURN p  MATCH p=()-[r:has_use]-&gt;() RETURN p LIMIT 25  MATCH p=()-[r:has_part]-&gt;() RETURN p  MATCH p=()-[r:has_effect]-&gt;() RETURN p LIMIT 25</pre>

WIKIDATA SPARQL -Electronic Component	Neo4j Cypher
SELECT ?eleComp ?eleCompLabel ?ofsubClass ?ofsubClassLabel ?subclass ?subclassLabel ?hasquality ?hasqualityLabel  WHERE { ?eleComp rdfs:label ?eleCompLabel . FILTER(LANG(?eleCompLabel) = "en") { SELECT ?ofsubClass ?ofsubClassLabel WHERE { ?ofsubClass wdt:P279 wd:Q11653 . ?ofsubClass rdfs:label ?ofsubClassLabel . FILTER(LANG(?ofsubClassLabel) = "en") } LIMIT 30 }  OPTIONAL{ ?eleComp wdt:P279 ?subclass . ?subclass rdfs:label ?subclassLabel . FILTER(LANG(?subclassLabel) = "en") }  OPTIONAL{ ?eleComp wdt:P1552 ?hasquality . ?hasquality rdfs:label ?hasqualityLabel . FILTER(LANG(?hasqualityLabel) = "en") }  FILTER(?eleComp = wd:Q11653) }	CALL apoc.load.json("file:/Users/rutujajagtap/ Downloads/ElComp.json") yield value  MERGE(e:eleComp{name:value.eleCompLabel})  MERGE(su:eleComp{name:value.ofsubClassLabel}) MERGE (e)-[:has_superClass]->(su) MERGE(s:eleComp{name:value.subclassLabel}) MERGE (e)-[:subClass_of]->(s)  MERGE(q:qua{name:value.hasqualityLabel}) MERGE (e)-[:has_quality]->(q)  MATCH p=()->() RETURN p  MATCH p=()-[r:has_quality]->() RETURN p LIMIT 25  MATCH p=()-[r:subClass_of]->() RETURN p LIMIT 25  MATCH p=()-[r:superClass_of]->() RETURN p  