# ORION user guide

tresata

Tresata

Every effort has been made to ensure that the information in this manual is accurate. Tresata is not responsible for printing or clerical errors.

Tresata
222 S Church Street

## ORION Query Usage Summary

This is a list of the most common queries and their usage. Note that ‘==’ can be replaced by ‘>,’ ‘<,’ ‘=,’ ‘>=,’ ‘<=,’ or ‘!=’ depending on the field type.

### Select Vertices

**Query: select vertices where “NODE_FIELD==’value’”**
*Usage: will return nodes satisfying specified condition*
Example: Find all nodes named Hercules.

**Query: select vertices where “NODE_FIELD==’value | | NODE_FIELD==’value’”**
*Usage: will return nodes satisfying either one of the specified conditions*
Example: Find all nodes named Hercules or Saturn.

**Query: select vertices where “NODE_FIELD==’value’ && NODE_FIELD==’value’”**
*Usage: will return vertices that satisfy both of the specified conditions*
Example: Find all nodes named Hercules and are demigods.

### In/Out

**Query: select vertices in from (vertices where “NODE_FIELD==’value’”)**
*Usage: will return nodes that have incoming edges from nodes satisfying specified condition*
Example: Find all nodes receiving money from Hercules.

**Query: select vertices out to (vertices where “NODE_FIELD==’value’”)**
*Usage: will return nodes that have outgoing edges to nodes satisfying specified condition*
Example: Find all nodes sending money to Hercules.

**Query: select vertices out from (vertices where “NODE_FIELD==’value’”)**
*Usage: will return nodes with outgoing edges from nodes satisfying specified condition*
Example: Find all nodes Hercules is sending money to.

**Query: select vertices in from (vertices where “NODE_FIELD== ‘value’”) out to (vertices where “NODE_FIELD==’value’”**
*Usage: will return vertices with incoming edges from nodes satisfying first specified condition and outgoing edges to nodes satisfying second specified condition*
Example: Find nodes receiving money from Hercules and sending money to Saturn.

**Query: select vertices in from (vertices where “NODE_FIELD==’value’”) in from (vertices where “NODE_FIELD==’value’”)**
*Usage: will return vertices with incoming edges from nodes satisfying the first specified condition and incoming edges from nodes satisfying the second specified condition*
Example: Find nodes receiving money from Hercules and Saturn.

**Query: select vertices out to (vertices where “NODE_FIELD== ‘value”’) out to (vertices where “NODE_FIELD==’value”’)**
*Usage: will return vertices with outgoing edges to nodes satisfying first specified condition and outgoing edges to nodes satisfying second specified condition*
Example: Find nodes sending money to Hercules and Saturn.

## Select Neighborhood

**Query: select neighborhood DEGREE of vertices where “NODE_FIELD==’value”’**
*Usage: return given degree connections from node satisfying specified condition*
Example: Find all nodes Hercules is directly sending money to.

**Query: select neighborhood DEGREE filter “EDGE_FIELD==’value”’ of vertices where “NODE_FIELD==’value”’**
*Usage: will return neighborhood of vertices satisfying specified node field condition but limit returned edges to those that satisfy the edge field condition*
Example: Find all nodes Hercules is directly sending more than $5 to.

## Label

**Query: select vertices out label LABEL VALUE to vertices where “NODE_FIELD==’value”’**
*Usage: will return nodes with outgoing edges of given label name directed to nodes satisfying the specified condition*
Example: Find Hercules’ father. (label name = father)

**Query: select vertices in label LABEL VALUE from vertices where “NODE_FIELD==’value”’**
*Usage: will return nodes with incoming edges of given label name directed from nods satisfying the specified condition*
Example: Find Hercules’ children. (label name = father)

**Query: select vertices both label LABEL VALUE from vertices where “NODE_FIELD==’value”’**
*Usage: will return all nodes with either incoming or outgoing edges of given label name from node satisfying the specified condition*
Example: Find Hercules’ father and children. (label name = father)

**Query: select vertices in label LABEL VALUE from (vertices where "NODE_FIELD== 'value'") in label LABEL VALUE from (vertices where “NODE_FIELD=='value'")**
*Usage: will return vertices that have incoming edge of first label value from node satisfying first specified condition and another incoming edge of second label value from node satisfying second specified condition*
Example: Find Hercules from his mom’s name and his dad’s name. (label name 1= father, label name 2 = mother)

## As

**Query: select vertices as x where “NODE_FIELD == ‘value’” out/in to/from vertices as y**
*Usage: will return vertices x, all outgoing/incoming edges, and vertices y*
Example: Find Hercules and all relatives.

**Query: select vertices as x where “NODE_FIELD == ‘value’” out/in to/from vertices as y where “NODE_FIELD == ‘value’”**
*Usage: will return vertices x, specified type of connection to vertices y, and vertices*
Example: Find Hercules father, grandfather, and children. (label = father)

**Query: select neighborhood as x of vertices where “NODE_FIELD == ‘value’” out/in to/from vertices as y where “NODE_FIELD == ‘value’”**
*Usage: will return neighborhood x, specified connections to vertices y, and vertices y*
Example: Find Hercules’ father and god uncles.

## In/Out Degree

**Query: select vertices where “_in_degree >value”**
*Usage: will return vertices that have more than specified value of incoming edges*
Example: Find the god with more than 2 incoming edges.

**Query: select vertices where “_out_degree > value”**
*Usage: will return vertices that have more than specified value of outgoing edges*
Example: Find the god with more than 2 outgoing edges.

**Query: select vertices where “_in_degree +_out_degree >value”**
*Usage: will return vertices that have more than the specified sum of incoming and outgoing edges*
Example: Find the god with more than 2 connections.

**Query: select vertices where “_in_degree -_out_degree > value”**
*Usage: will return vertices that have more incoming and outgoing edges than the specified difference*
Example: Find the god with 2 more incoming edges than outgoing edges.

## Sub-queries

**Query: select neighborhood DEGREE of (select vertices where “NODE_FIELD==‘value’” out from (vertices where “NODE_FIELD==‘value’”))**
*Usage: will return neighborhood of vertices satisfying the sub-query*
Example: Find neighborhood of nodes that Hercules sends money to.

**Query: select union (select vertices where “NODE_FIELD==‘value’”, select neighborhood DEGREE of vertices where “NODE_FIELD== ‘value’”)**
*Usage: will return all nodes that satisfy either conditions specified by sub-queries*
Example: Find Hercules and Saturn’s direct connections.

**Query: select intersection (select vertices where “NODE_FIELD==‘value’”, select neighborhood of vertices where “NODE_FIELD==‘value’”)**

*Usage: will return nodes that satisfy both conditions specified by the sub-queries*

Example: Find the gods that are directly connected to Hercules.

## Select Paths

**Query: select paths (select vertices where “NODE_FIELD == ‘value’”) exit “NODE_FIELD == ‘value’”**

Usage: will return all shortest paths between the two specified node types

Example: Find the most direct connection between Hercules and Saturn.

**Query: select all paths (select vertices where “NODE_Field == ‘value’”) exit “NODE_FIELD == ‘value’”**

*Usage: will return all paths between the two specified node types*

Example: Find me all possible connections between Hercules and Saturn

**Query: select all paths (select vertices where “NODE_FIELD == ‘value’”) exit “NODE_FIELD == ‘value’” max 3**

*Usage: will return all 3-hop paths between the two specified node types*

Example: Find all 3-hop connections between Hercules and Saturn

**For support visit the ORION Slack help desk channel.**