Graql Cheatsheet

This cheatsheet is a quick reference guide for those already familiar with Graql. For further information and additional documentation, please see the Graql section on our developer portal at **tinyurl.com/graqldocs**.

Variables

Variables start with `$` followed by alphanumeric characters.

Queries

match

A `match` query will search the graph for any subgraphs that match the given pattern.

`match [pattern; ...] [modifiers]`

```
match $x isa person;
```

```
# Match several patterns together
match $x isa person, has firstname "John";
```

Modifiers

```
# Select particular variables from the query
match $x isa person, has firstname $f, has surname $s;
select $f, $s;

# Skip some results & limit the number returned
match $x isa person, has identifier $id;
limit 10; offset 5;
```

```
# De-duplicate the results of a query
match $x isa person, has firstname $y; select $y;
distinct;
```

```
# Order by variable [ asc | desc ]
match $x isa person, has surname $n;
order by $n desc;
```

ask

An `ask` query will return whether the given match query has any results.

`[match] ask`

```
match $x isa person, has name 'James Cameron';
ask;
```

insert

An insert query will insert the specified variable patterns into the graph.

`insert [ pattern ; ... ]`

```
insert has identifier "Titus Groan" isa person;
```

If a match query is provided, the query will insert the given variable patterns for every result of the match query.

```
match $p has identifier "Minnie Downs";
insert $p has middlename "Mathilda";

match $b has name "Tim Burton";
$m isa movie; (director: $b, $m);
match $d has name "Johnny Depp";
# Insert a relation
insert (actor: $d, production-with-cast: $m) isa
has-cast;
```

delete

A delete query will delete the specified variable patterns for every result of the match query.

match delete [pattern ; ...]

```
match $x isa person; delete $x;
```

Pattern Matching

Match a variable.

identifier [property, ...]

```
match $x isa person, value "Guillermo del Toro";
```

Match either the left or right pattern.

pattern *or* pattern

```
match $x isa movie or $x isa person;
```

Match either pattern to the left of `**or**` or all the patterns to the right.

{ [pattern ; ...] }

```
match $x isa person, has identifier $y;
{$y value contains "Elizabeth";} or
{$y value contains "Mary";};
```

Type Properties

Specify the type of a concept.

isa type

```
match $x isa person;
```

Match concepts and their types.

```
match $x isa $y;
```

Match the concept with a particular ID.

id {string}

```
match $x id '12345';
```

Match concepts with a value that contains the given string.

value [=|!=|<|<=|>=|>|contains] {value}

```
match $m value contains "William Titus, Jr";
```

Match concepts with a resource matching a predicate.

has resource [=|!=|<|<=|>=|>|contains] {value}

```
match $p isa person, has age > 80;
```

Join Our Community!

GRAKN.AI

Don't forget to sign up for our regular newsletter and register on Slack for up-to-the-minute announcements about GRAKN.AI.