Pathway Studio
Graph Object Query Language (GOQL)

Pathway Studio Object Graph Query Language (GOQL) was designed to resemble well known SQL language used by most modern RDBMS. At this moment it only supports SELECT statement allowing to query objects stored in the database. There are no capabilities to modify or add objects.
GOQL queries can be executed using "**Advanced search**" dialog in Pathway Studio by pasting it "*Search box*" or in SOAP API queries from 3d party applications.
Use systemic names of the properties and object names without white spaces to avoid escaping double quotes in your script. E.g. use **RelationNumberOfReferences** instead of "# of Total References"

SELECT Syntax

```
<query> ::= SELECT <objectClass> WHERE <search_condition>

<objectClass> ::= Entity | Network | Group | OntologicalNode | Relation | Experiment |
Annotation | Chip

<search_condition> ::= { [ NOT ] <predicate> | ( <search_condition> ) }
[ { AND | OR } [ NOT ] { <predicate> | ( <search_condition> ) } ]

<predicate> ::= { { <property> | ( <object_properties> ) } { = | < | > | >= | <= | != | is | LIKE }
<expression>
   | { <property> | ( <object_properties> ) } IN ( <expressions> ) |
   NeighborOf [<direction_clause> ] ( <query> ) |
   HasHeighbors ( <list_of_queries> ) |
   HasHeighbors <directional_queries> |
   MemberOf ( <query> ) |
   ParentOf ( <query> ) |
   Connected [ by [<direction_clause> ] ( <query> ) ] [ to [ <direction_clause> ] ( <query> ) ] |
   InOntology ( <query> ) [ inRange <expression> ] <ont_direction_clause> ( <query> )
   }

<object_properties> ::= <property> [ , <object_properties> ]

<expressions> ::= <expression> [ , <expressions> ]

<direction_clause> ::= { upstream | downstream | nondirected }
```

<ont_direction_clause> ::= { over | under }

<list_of_queries> ::= (*<query>*) [{ , | AND } *<list_of_queries>*]

<directional_queries> ::= { *<direction_clause>* | anydir } (*<query>*) [*<directional_queries>*]

Arguments

<property>

Name of the property of database object. When query is entered via user interface, DisplayName of the property is usually expected. Command Line interface usually expects RNEF names. Double quotation marks can be used to quote property names with whitespaces.

Special properties:
objectType
objectClass
id

<expression>

String or numeric expression. Single quotation marks can be used to quote string expressions containing whitespaces.

LIKE

Equivalent to LIKE pattern matching operator in SQL. String expression is passed directly to underlying RDBMS, so the syntax of the expression and set of wildcard characters depends on used RDBMS. Please refer to corresponding SQL reference.

IN

Corresponds to IN in SQL, allows to use multiple expressions in comparison conditions.
Implemented as an alias to =

NeighborOf

When used in search condition applied to *entities*, matches entities participating in relations matching specified sub-query. objectClass specified for sub-query must be Relation in this case.
When used in search condition applied to *relations*, matches relations having entities matching specified sub-query. objectClass specified for sub-query must be Entity in this case.
Optionally allows specifying direction of the link to neighbor.

Example (all outgoing Expressions from Protein named 'ALB'):

```
SELECT Relation WHERE NeighborOf downstream ( SELECT Entity WHERE objectType=Protein
AND Name=ALB) AND objectType=Expression
```

HasNeighbors

Allows to specify list of sub-queries matching neighbors. Each sub-query must match at least one distinct neighbor.

Example (select all Bindings connect Protein named 'ALB' with at least one other protein):

```
SELECT Relation WHERE
        HasNeighbors ((SELECT Entity WHERE objectType=Protein AND Name=ALB), (SELECT
Entity WHERE objectType=Protein AND Name!=ALB)) AND objectType=Binding
```

This query is equivalent to:

```
SELECT NeighborOf (SELECT Entity WHERE objectType=Protein AND Name=ALB) AND
       NeighborOf (SELECT Entity WHERE objectType=Protein AND Name!=ALB) AND
objectType=Binding
```

Connected

Connected condition represents a combination of 2 neighbor conditions. When used in entity search conditions, allows to specify sub-queries for both connected entities and relations connecting those entities. When used in relation search condition, it is equivalent to NeighborOf operator with sub-query matching entities.

Optionally allows specifying direction of the links to neighbor or connected entity.

Example (select all Entities with Binding to Protein with name 'ALB'):

```
SELECT Entity Connected by (SELECT Relation WHERE objectType=Binding) to (SELECT
Entity WHERE Name=ALB AND objectType=Protein)
```

MemberOf

Matches entities or relations which are members of pathways matching specified sub-query. objectClass specified for sub-query must be Group, Network or OntologicalNode. Matches only Group/Network members, not ontological relationships or folders.

Example (select all members of given group):

```
SELECT Entity WHERE MemberOf (SELECT Group WHERE URN='urn:agi-group:uuid-ef604080-
e45d-43ea-9d95-7f4d02af5f80')
```

Example (select all relations in given Pathway):

```
MemberOf (SELECT Network WHERE URN='urn:agi-pathway:uuid-179d385f-5fa2-4be9-a64b-
8b183df5618e')
```

ParentOf

Can only be used for *Network*, *Group* or *OntologicalNode* search conditions. Allows matching pathways having as members entities or relations matching sub-query.

Example (select Groups and Networks with particular Entity):

```
SELECT OntologicalNode WHERE ParentOf (SELECT Entity WHERE Name='ACE')
```

InOntology

Matches objects in ontologies with given relative position to seeds and connected with given annotations Annotations. Matched objectClass could be Entity, Group, Network or OntologicalNode that is generalization of these three.

The first subquery must select Annotations forming the ontology and the second one select seeds to start from (*OntologicalNode, Entity, Group, Network*).

inRange is optional and limits the recursion. By default the recursion is infinite. *inRange 1* gives analogue to *Connect* clause, but for Annotations instead of Relations.

Example (select all Entities (not Sematic Concepts!) directly or indirectly under drugs concept regarding Pathway Studio Ontology):

```
SELECT Entity WHERE InOntology (SELECT Annotation WHERE Ontology='Pathway Studio
Ontology' AND Relationship='is-a') under (SELECT OntologicalNode WHERE Name='drugs')
```

Example (select semantic-concepts which are either direct or indirect 2nd level super-classes for drugs in Pathway Studio Ontology)

```
SELECT OntologicalNode WHERE InOntology (SELECT Annotation WHERE Ontology='Pathway
Studio Ontology' AND Relationship='is-a') inRange 2 over (SELECT OntologicalNode WHERE
Name='monoclonal antibody drug') AND objectType=SemanticConcept
```

AnnotatedBy

Can only be used in entity search conditions. Allows querying ontological relationships. Matches entities which are child concepts of entities matching specified sub-query.

ParentOf

Can only be used in entity search conditions. Allows querying ontological relationships. Matches entities which are parent concepts of entities matching specified sub-query.

Examples

Find proteins with name or alias 'p53'

```
SELECT Entity WHERE objectType = Protein AND (Name,Alias) = p53
```

Find Transcription factors regulated by small molecule 'Selenium'

```
SELECT Entity
```

```
WHERE Connected by (SELECT Relation WHERE objectType IN (Regulation, DirectRegulation))
to (SELECT Entity WHERE objectType='Small Molecule' AND (Name,Alias)='Selenium')
AND AnnotatedBy recursive (SELECT Group WHERE Name='Transcription factors' AND
Source='Pathway Studio Ontology')
```

Find all Protein – *Expression* --> Functional Class relations

```
SELECT Relation
WHERE NeighborOf downstream ( SELECT Entity WHERE objectType=Protein )
AND NeighborOf upstream ( SELECT Entity WHERE objectType='Functional Class' )
AND objectType=Expression
```

Find all relations of type *Binding* between Protein MDM2 and Functional Classes

```
SELECT Relation
WHERE HasNeighbors( (SELECT Entity WHERE objectType=Protein AND Name=MDM2),
(SELECT Entity WHERE objectType='Functional Class'))
AND objectType=Binding
```

Find kinase targets of known inhibitors (small molecules with Pharmapendum ID, present on pathways)

```
SELECT Relation WHERE
NeighborOf downstream ( SELECT Entity WHERE objectType='Small Molecule' AND !(
"PharmaPendium ID" = null)
AND MemberOf ( SELECT Network WHERE objectType=Pathway ))
AND NeighborOf upstream ( SELECT Entity WHERE objectType=Protein
AND AnnotatedBy recursive ( SELECT Group WHERE Name='Protein kinases' ) )
AND objectType=DirectRegulation
AND Effect = 'negative'
```

Select relations for proteins involved in insulin resistance

```
select Relation where NeighborOf downstream ( SELECT Entity WHERE
objectType=('Complex','Functional Class','Protein') )
AND NeighborOf upstream ( SELECT Entity WHERE (Name,Alias)='insulin resistance' )
```

Select proteins involved in insulin resistance in skeletal muscles

```
select Relation where NeighborOf downstream ( SELECT Entity WHERE
objectType=('Complex','Functional Class','Protein') )
AND NeighborOf upstream ( SELECT Entity WHERE (Name,Alias)='insulin resistance' )
AND ("CellType" LIKE 'myo%' OR "Organ" LIKE 'skeletal muscle%' OR "Tissue" LIKE '%muscle%')
```

2-step expand upstream from diseases towards proteins

```
select Entity WHERE objectType=Protein AND HasNeighbors (
select Relation NeighborOf upstream (
select Entity WHERE objectType=Protein AND HasNeighbors (
Select Relation WHERE NeighborOf upstream (SELECT Entity WHERE
(Name,Alias)='phenylketonuria')))))
```

Usptream expand from proteins neighboring disease in all directions

```
select Entity WHERE objectType=Protein AND HasNeighbors (
select Relation NeighborOf upstream (
select Entity WHERE objectType=Protein AND HasNeighbors (
Select Relation WHERE  HasNeighbors (SELECT Entity WHERE (Name,Alias)='phenylketonuria')))))
```

Select *FunctionalClass* physically interacting with MDM2

```
Select Relation WHERE HasNeighbors((SELECT Entity WHERE objectType=Protein AND
Name=MDM2),
(SELECT Entity WHERE objectType='Functional Class'))
AND objectType in (Binding,  DirectRegulation)
```

Select biomarkers for disease phenylketonuria

```
select Entity WHERE objectType=Protein AND HasNeighbors (
Select Relation WHERE NeighborOf downstream (SELECT Entity WHERE
(Name,Alias)='phenylketonuria') AND objectType in (Biomarker, QuantitativeChange))
```

Select all proteins linked to all types of diabetes

```
select Entity WHERE objectType in ('Protein', 'Complex', 'Functional Class')
AND HasNeighbors (Select Relation WHERE NeighborOf
(SELECT Entity WHERE objectType = Disease AND  ("Alias","Name") LIKE '%diabetes%' )
)
```

Find all protein-protein relations

```
Select Relation WHERE HasNeighbors((SELECT Entity WHERE objectType=Protein),
(SELECT Entity WHERE objectType=Protein))
```

Select Entities linked to both Experimental asthma and to Asthma only by one reference

```
select Entity WHERE HasNeighbors (
Select Relation WHERE NeighborOf (SELECT Entity WHERE (Name,Alias)='Experimental asthma'))
AND HasNeighbors (
Select Relation WHERE NeighborOf (SELECT Entity WHERE (Name,Alias)='Asthma') AND "# of
References" <2)
```

Select Entities and Relations linked to Experimental asthma and to Asthma by one reference

```
select Relation where NeighborOf (
select Entity WHERE HasNeighbors (
Select Relation WHERE NeighborOf (SELECT Entity WHERE (Name,Alias)='Experimental asthma'))
AND HasNeighbors (
Select Relation WHERE NeighborOf (SELECT Entity WHERE (Name,Alias)='Asthma') AND "# of
References" <2) )
and NeighborOf (SELECT Entity WHERE (Name,Alias) in ('Asthma', 'Experimental asthma'))
```

Select MDM2 Small molecule regulators with certain XLogP-AA"

```
SELECT Relation WHERE HasNeighbors(
(SELECT Entity WHERE objectType=Protein AND Name=MDM2),
(SELECT Entity WHERE objectType = SmallMol AND !("CAS ID" is null) AND ("XLogP-AA" = '-1.7'))
)
```

Select MTOR direct phosphorylation targets

```
select Relation where NeighborOf downstream (
SELECT Entity WHERE objectType=('Complex','Functional Class','Protein') AND Name =
('TORC1','TORC2','MTOR'))
AND NeighborOf upstream ( SELECT Entity WHERE objectType=('Complex','Functional
Class','Protein') )
AND objectType='ProtModification'
```

Select MTOR indirect phosphorylation targets

```
select Relation where NeighborOf downstream (
select Entity WHERE HasNeighbors (
select Relation where NeighborOf downstream (
SELECT Entity WHERE objectType=('Complex','Functional Class','Protein') AND Name =
('TORC1','TORC2','MTOR'))
AND NeighborOf upstream ( SELECT Entity WHERE objectType=('Complex','Functional
Class','Protein') )
AND objectType='ProtModification' and Effect = ('positive, null, 'unknown')))
AND objectType='ProtModification' and Effect = ('positive, null, 'unknown')
```

Find side-effects of Carvedilol

```
Select Relation where NeighborOf downstream (select Entity where Name = 'Carvedilol')
AND NeighborOf (select Entity where objectType = 'Disease')
AND (Effect = 'positive' OR Effect = null)
AND objectType = 'Regulation'
```

Find inhibitors of protein kinases in pathways

```
SELECT Relation WHERE
NeighborOf downstream ( SELECT Entity WHERE objectType='Small Molecule' AND !("CAS ID" is null)
AND MemberOf ( SELECT Network WHERE objectType=Pathway ))
AND NeighborOf upstream ( SELECT Entity WHERE objectType=Protein
AND AnnotatedBy recursive ( SELECT Group WHERE Name='Protein kinases' ) )
AND objectType=DirectRegulation
AND Effect = 'negative'
```

Find CNS disorders with at least one relations

```
SELECT Entity WHERE objectType=Disease
AND AnnotatedBy recursive ( SELECT Entity WHERE Name='central nervous system disease' )
AND "Total Connectivity" > 0
```

Find disease that have a link to Protein form a group *'List of CNS disorders'*

```
SELECT Entity WHERE objectType=Disease
AND NeighborOf  (
SELECT Relation WHERE NeighborOf  (select Entity WHERE objectType=Protein)
        AND NeighborOf  (SELECT Entity WHERE AnnotatedBy recursive ( SELECT Group WHERE Name='List of CNS disorders' ))
)
```

How many genes - disease modifiers are in Pathway Studio?

```
Select Entity where objectType = Protein and
NeighborOf  (SELECT Relation WHERE objectType= Regulation
AND NeighborOf  (SELECT Entity where  objectType = Disease)
AND NeighborOf  (SELECT Entity where  objectType = Protein)
)
```

How many disease genetic modifiers are in Pathway Studio?

```
Select Entity where objectType = Protein and
NeighborOf  (SELECT Relation WHERE objectType= GeneticChange
AND NeighborOf  (SELECT Entity where  objectType = Disease)
AND NeighborOf  (SELECT Entity where  objectType = Protein)
)
```

How many disease with known genetic modifiers are in Pathway Studio?

```
Select Entity where objectType = Disease and
NeighborOf  (SELECT Relation WHERE objectType= GeneticChange
AND NeighborOf  (SELECT Entity where  objectType = Disease)
AND NeighborOf  (SELECT Entity where  objectType = Protein)
AND "# of References" > 0
)
```

Find links between anti-cancer drugs and mutations.

Also try: Sentence LIKE '%mutant%' OR Sentence LIKE '%genetic variation%' OR Sentence LIKE '%polymorphism%' OR Sentence LIKE '%allele%'

```
select Relation where NeighborOf downstream ( SELECT Entity WHERE objectType=('Small
Molecule') )
AND NeighborOf upstream ( SELECT Entity where  objectType = Disease AND AnnotatedBy recursive
( SELECT Group WHERE Name='Cancers') )
AND (Sentence LIKE '%mutation%')
AND Effect = 'negative'
```

Find somatic mutations in cancer

```
SELECT Relation WHERE objectType= GeneticChange
AND NeighborOf  (SELECT Entity where  objectType = Disease AND AnnotatedBy recursive ( SELECT
Group WHERE Name='Cancers') )
AND "Sentence" LIKE '%somatic%'
```

Find inherited germline mutations in cancer

```
SELECT Relation WHERE objectType= GeneticChange
AND NeighborOf  (SELECT Entity where  objectType = Disease AND AnnotatedBy recursive ( SELECT
Group WHERE Name='Cancers') )
AND ("Sentence" LIKE '%germ%' OR "Sentence" LIKE '%heredet%' OR "Sentence" LIKE '%herit%')
```

Find proteins involved in mitochondrial function of beta-cells

```
select Relation WHERE HasNeighbors (
(select Entity WHERE  Name = ('type 1 diabetes','beta-cell function', 'insulin-secreting cell'),
(select Entity WHERE objectType=('Protein', 'Complex','Functional Class'))
AND "Sentence" LIKE '%mitochondria%')
```

Find secreted proteins

```
SELECT Entity where objectType=('Protein', 'Functional Class','Complex') AND Connected by
(SELECT Relation WHERE objectType='MolTransport' AND Sentence LIKE '%secret%' ) to (SELECT
Entity WHERE  objectType='Protein')
```

Find proteins suppressing immune response and expressed on melanoma.

Requires creating two groups first: '*Tumor wants to Suppress*' groups contains all CellProcess and Cell that can be down-regulated to suppress immune response, '*Melanoma secreted and cell surface molecules*' containing entities secreted by Melanoma or expressed on its surface

```
SELECT Entity Connected by (SELECT Relation WHERE objectType=Regulation AND Effect=negative)
to (SELECT Entity WHERE MemberOf ( SELECT Group WHERE Name='Tumor wants to Suppress') )
AND MemberOf ( SELECT Network WHERE Name='Melanoma secreted and cell surface molecules')
```

To create '*Melanoma secreted and cell surface molecules*' containing entities secreted by Melanoma or expressed on its surface combine results of the following queries in one group

```
SELECT Entity WHERE objectType='Small molecule' AND Class='Endogenous compound' AND
objectType=CellExpression AND Mechanism=surface
```

```
SELECT Entity WHERE objectType='Protein’ AND objectType=CellExpression AND
Mechanism=surface
```

```
SELECT Entity WHERE objectType='Protein’ AND objectType=MolTransport
```

```
SELECT Entity WHERE objectType='Small molecule' AND Class='Endogenous compound' AND
objectType=MolTransport
```