

AgensBrowser Guide

SKAI worldwidel Inc.

2021 년 10 월 08 일



Copyright Notice

Copyright © 2017-2025, SKAI worldwide Inc. All Rights Reserved.

Restricted Rights Legend

PostgreSQL is Copyright © 1996-2025 by the PostgreSQL Global Development Group.

Postgres95 is Copyright © 1994-5 by the Regents of the University of California.

AgensGraph is Copyright © 2017-2025 by SKAI worldwide Inc.

소스코드와 바이너리 형태의 재배포와 사용은 수정 여부와 관계없이 다음 조건을 충족할 때 가능하다.

소스코드를 재배포하기 위해서는 반드시 위의 저작권 표시, 지금 보이는 조건들과 다음과 같은 면책조항을 유지하여야만 한다. 바이너리 형태의 재배포는 배포판과 함께 제공되는 문서 또는 다른 형태로 위의 저작권 표시, 지금 보이는 조건들과 다음과 같은 면책조항을 명시해야 한다. 사전 서면 승인 없이는 저자의 이름이나 기여자들의 이름을 이 소프트웨어로부터 파생된 제품을 보증하거나 홍보할 목적으로 사용할 수 없다. 본 SW 는 저작권자와 기여자들에 의해 “있는 그대로” 제공될 뿐이며, 상품가치나 특정한 목적에 부합하는 묵시적 보증을 포함하여(단, 이에 제한되지 않음), 어떠한 형태의 보증도 하지 않는다.

어떠한 경우에도 재단과 기여자들은 제품이나 서비스의 대체 조달, 또는 데이터, 이윤, 사용상의 손해, 업무의 중단 등을 포함하여(단, 이에 제한되지 않음), 본 소프트웨어를 사용함으로써 발생한 직접적이거나, 간접적 또는 우연적이거나, 특수하거나, 전형적이거나, 결과적인 피해에 대해, 계약에 의한 것이든, 엄격한 책임, 불법행위(또는 과실 및 기타 행위를 포함)에 의한 것이든, 이와 여타 책임 소재에 상관없이, 또한 그러한 손해의 가능성이 예견되어 있었다 하더라도 전혀 책임을 지지 않는다.

Trademarks

AgensGraph®는 SKAI worldwide Inc.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., AgensGraph Commercial License, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

기술문서 정보

제목 : AgensBrowser Guide

발행일 : 2021 년 11 08 일

소프트웨어 버전 : AgensBrowser 2.1

기술문서 버전 : v1.0

Introduction

AgensBrowser is

AgensBrowser 는 웹 브라우저를 통해 AgensGraph 를 관리할 수 있는 사용자 인터페이스를 제공하여 AgensGraph DB 의 각 구성 요소들의 상태를 모니터링하고 제어할 수 있다.

주요특징은 다음과 같다.

- RESTful-API 지원
- Meta Information 제공 및 자동 갱신
- Label 생성 및 삭제
- Cypher 및 Ansi-SQL 입력, 실행, 저장
- Project Load/Save/Delete 지원

- Graph 탐색 도구와 스타일링, 요약 기능 제공
- Graphson 의 import, export 와 image export 제공
- Query History 조회
- LiveShare For Report 기능

System Requirements

AgensBrowser 를 실행하기 위해서 다음과 같은 환경을 갖추어야 한다.

- Server
 - OS : Windows 8 or higher, OS X 10 or higher, All kinds of Linux/Unix OSes
 - Memory : 8GB or more
 - Java version : 1.8
- Client
 - Browsers : Javascript ES6 Browsers (Chrome, Firefox, Edge, Safari etc.)
 - *Internet Explorer 미지원*

Installation

Environment Setting

- Java 1.8 설치
- AgensGraph 설치 (AgensGraph v1.3 이상)
 - AgensBrowser 관리용 DB 설치
 - AgensBrowser Admin 계정 생성
- binary 준비

```
agens-browser-web-2.1
```

Installation and Startup

1. 압축파일 해제 및 파일 확인

- agents-browser-web-2.1.jar
- agents-config.yml

2. Yml 파일 수정

agents-config.yml 파일을 열어 실제 환경에 맞게 접속 정보를 수정한다. Yml 파일에 지정된 graph_path 로 접속되며, 로그인 계정은 graph_path 의 owner 여야 한다. 서비스 시작시 지정된 graph_path 는 변경할 수 없다.

- server
 - port : AgensBrowser 웹서버 포트
- logging
 - config : AgensBrowser 의 로그 설정으로 현재 폴더 아래의 logs 디렉토리에 날짜별로 생성
- agents
 - api :
 - query_timeout : query 타임아웃 설정(Default: 600000, 10Min)
 - jwt :
 - expiration : client 접속 만료 시간 설정(Default: 43200, 12H)
 - inner :
 - url : 쿼리 log 와 프로젝트 저장 용도로 사용되며 현재 폴더의 agents-db.db 로 파일 생성
(삭제되면 초기 데이터와 함께 다시 생성됨)
 - outer :
 - url

- IP : 데이터베이스 서버의 IP
- Port : 데이터베이스 서버의 Port
- Database : 접속할 데이터베이스명
- graph_path : AgensGraph 에 생성한 graph name
- username : 접속할 데이터베이스의 user name
- password : 접속할 데이터베이스의 user password
- max_rows : 반환할 최대 row 수 (1000 이하 권장)
- client :
 - guest-key : 저장된 project 를 report 용으로 사용하기 위한 공개키

ex) http://192.168.0.56:8084/#/report/agens/1130

 - animation-enabled : client 의 graph 레이아웃 적용시 애니메이션 적용 여부
 - title-shown : client 의 graph 출력시 node 의 title 출력 여부

(true 설정시 렌더링 성능이 저하되므로 default 로 false 권고)

```
## config context path to "/" by setting an empty string
server:
  port: 8085    -- AgensBrowser 웹서버 포트

logging:
  config: classpath:logback-agens.xml

spring:
  pid:
    file: agensbrowser.pid
  main:
```

```

    banner-mode: "off"
resources:
    cache:
        period: 3600
    static-locations: classpath:/META-INF/resources/,classpath:/resources/, \
                        classpath:/static/,classpath:/public/

servlet:
    multipart:
        ## MULTIPART (MultipartProperties)
        enabled: true # Enable multipart uploads
        file-size-threshold: 2KB # Threshold after which files are written to disk.
        max-file-size: 200MB # Max file size.
        max-request-size: 215MB # Max Request Size

agens:
    api:
        base-path: /api
        query-timeout: 600000 # 1000 ms = 1 sec -- 타임아웃 10 분
    jwt:
        header: Authorization # not used
        secret: agensBrowserKey
        expiration: 43200 # unit: sec (12 Hour = 43200 sec)
    inner:
        datasource:
            driverClassName: org.h2.Driver
            url: jdbc:h2:file:./agens-h2;DB_CLOSE_DELAY=-1;MODE=MySQL
            username: sa
            password:
    outer:
        datasource:
            driverClassName: org.postgresql.Driver
            url: jdbc:postgresql://127.0.0.1:5432/agens?ApplicationName=Agens
Browser
    graph_path: SKAI worldwide
    username: agens
    password: agens
    max-rows: 1000 -- 최대 반환 rows
    file:
        download-dir: ./downloads # for image file, etc..
        upload-dir: ./uploads # for Graphson, Graphml, etc.. (not yet support EXCEL)
    client:
        mode: prod # mode : dev or prod
        guest-key: agens # url middle-value for report output

```

```

-- 저장된 project 를 report 용으로 사용하
기 위한 공개키
    animation-enabled: true    # to use animation at applying layout
                                -- client 의 graph 레이아웃 적용시 애니메
이션 적용 여부
    title-shown: false        # to show node title by default
                                -- client 의 graph 출력시 node 의 title
출력 여부
                                -- (true 설정시 렌더링 성능 저하되므로 de
fault 로 false 권고)
    product:
        name: AgensBrowser-web
        version: 2.1
        hello-msg: AgensBrowser web v2.1    -- hello message

```

3. 실행파일 생성

text 파일에 아래의 내용을 기입 후 확장자 *.sh(unix)* 또는 *.bat(window)*로 저장한다.
(ex. *agens-browser.sh* / *agens-browser.bat*)

```
java -jar agens-browser-web-2.1.jar --spring.config.name=agens-config
```

4. 서비스 실행

```
{ }      $ sh agens-browser.sh  -- linux      $ agens-browser.bat      --
windows
```

5. 서비스 실행 확인

```

java -jar agensbrowser-web-2.1.jar --spring.config.name=agens-config
<config> agens.datasource.url = jdbc:postgresql://127.0.0.1:5432#/agens?ApplicationName=AgensBrowser
<config> agens.datasource.schema = bitnine

=====
AgensBrowser web v2.1
=====

check version of AgensGraph ... v1.3 or over

first loading META-info of AgensGraph starts...
reload[000]: bitnine,labels=3(2/1),relations=1,isDirty=false ==> OK!
healthInfo ==> {"product_version":"2.1","user_name":"agens","established_connections":6,"idle_connections":6,"descriptio
n":"","busy_connections":0,"jdbc_url":"jdbc:postgresql://127.0.0.1:5432#/agens?ApplicationName=AgensBrowser","product_
name":"AgensBrowser-web","test_time":"2018-12-20 10:04:07","cp_type":"hikari","schema_image":"","active_sessions":0,"gro
up":"health","is_closed":false}

```

Login

Token 기반 인증방식으로 config 파일에 명시된 AgensGraph 계정(ID/PW)으로 자동 로그인 된다.

Database 접속계정은 admin 계정이어야 한다.

AgensBrowser 에 연결하기 위해 웹 브라우저의 주소창에 아래의 URL 을 입력한다.

```
http://DB_SERVER_IP_ADDRESS:WEB_SERVER_PORT/index.html
```

로컬 데이터베이스를 사용하고 웹 서버 포트 8085 를 사용하는 경우의 예이다.

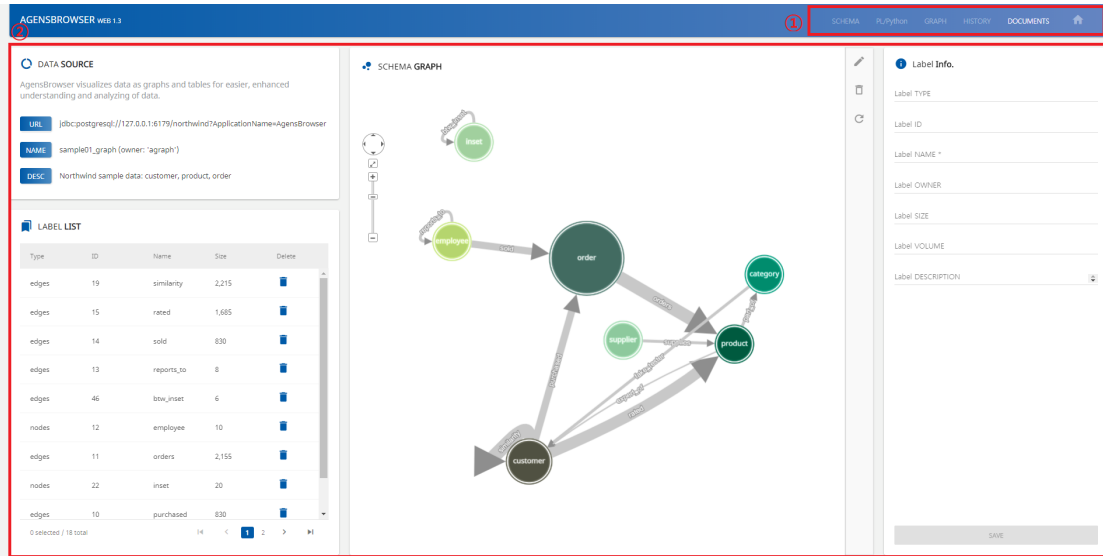
```
http://localhost:8085/index.html
```

Logout

접속한 브라우저 창을 종료하면 로그아웃되며, config 파일의 expiration 설정으로 client 접속 만료시간 설정이 가능하다.

Screen Layout

로그인에 성공하면 아래의 그림과 같은 메인화면으로 이동한다. 메인화면은 Database 접속 정보 및 graph 메타 정보를 보여주며, AgensBrowser 에서 제공하는 기능을 선택하여 사용할 수 있는 메뉴들로 구성되어 있다.



메인 화면은 다음 2 가지 영역으로 나뉘어진다.

- Menu Area (①)
AgensBrowser 에서 사용되는 기능을 세부적으로 구분하여 주요 메뉴를 제공한다.
메뉴 영역을 통해서 AgensBrowser 에서 제공하는 기능을 선택하여 세부 메뉴 및 화면으로 이동한다.

다음은 각 메뉴 아이콘에 대한 설명이다.

아이콘	설명
	접속정보와 메타 그래프 정보 확인이 가능하다.
	PL/Python function 생성 및 목록 확인이 가능하다.
	Query Editor, Query Ouptut, Graph Visualiztion, Project Save 등을 수행할 수 있다.
	Query Log 확인이 가능하다.

AgensBrowser 메뉴얼 확인이 가능하다.

- Main Area (②)
실제 작업영역으로 메뉴를 통해 수행한 업무에 대한 제어, 모니터링을 제공한다.

Functions Detail

Schema Menu

사용자의 접속 정보 및 AgensGraph 의 Graph Meta Data 를 보여준다.

The screenshot displays the AgensGraph Schema Menu interface, which is divided into four main panels:

- DATA SOURCE:** Contains connection details for the data source.
 - URL: jdbc:postgresql://127.0.0.1:5432/northwind?ApplicationName=AgensBrowser
 - NAME: sample01_graph (owner: 'agens')
 - DESC: Northwind sample data: customer, product, order
- LABEL LIST:** A table listing the labels in the graph.

Type	ID	Name	Size	Delete
edges	19	similarity	2,215	
edges	13	reports_to	8	
edges	100	orders	2,155	
edges	99	supplies	77	
edges	21	expert_of	1	
nodes	97	product	77	
nodes	9	order	830	
nodes	5	supplier	29	
edges	14	sold	830	
- SCHEMA GRAPH:** A visual representation of the graph structure, showing nodes (customer, order, product, category, supplier, employee) and their relationships (edges) with labels like 'reports_to', 'orders', 'supplies', 'expert_of', 'sold', and 'similarity'.
- Label Info:** A panel for editing the properties of a selected label.
 - Label TYPE
 - Label ID
 - Label NAME *
 - Label OWNER
 - Label SIZE
 - Label VOLUME
 - Label DESCRIPTION
 - SAVE button

Schema 는 다음의 항목으로 구성된다.

- Data Source
- Label List
- Schema Graph & Label Info

Data Source

Data Source 항목은 connection 정보를 보여준다.



DATA SOURCE

AgensBrowser visualizes data as graphs and tables for easier, enhanced understanding and analyzing of data.

URL

jdbc:postgresql://127.0.0.1:6179/northwind?ApplicationName=AgensBrowser

NAME

sample01_graph (owner: 'agraph')

DESC

Northwind sample data: customer, product, order

Label List

Node 와 edge label 들의 내역을 확인할 수 있으며, Delete(🗑️)을 선택하여 node 및 edge 레이블을 삭제할 수 있다.

 LABEL LIST

Type	ID	Name	Size	Delete
edges	19	similarity	2,215	
edges	15	rated	1,685	
edges	14	sold	830	
edges	13	reports_to	8	
edges	46	btw_inset	6	
nodes	12	employee	10	
edges	11	orders	2,155	
nodes	22	inset	20	
edges	10	purchased	830	

0 selected / 18 total

 < 1 > 

- Label 삭제

Delete 의 아이콘을 누르면 아래의 그림과 같이 Label 을 삭제하기 전 삭제될 label 에 대한 정보를 보여주고 삭제를 진행할 것인지 여부를 묻는다. Label 을 삭제하면 관련 데이터 또한 삭제되므로 사용 시 주의가 필요하다. 아래의

OK/CANCEL 버튼으로 삭제를 수행/취소할 수 있다.

 **CONFIRM INFO**

Are you sure to delete this LABEL?
If you remove label, All data will also be removed!

TYPE
nodes

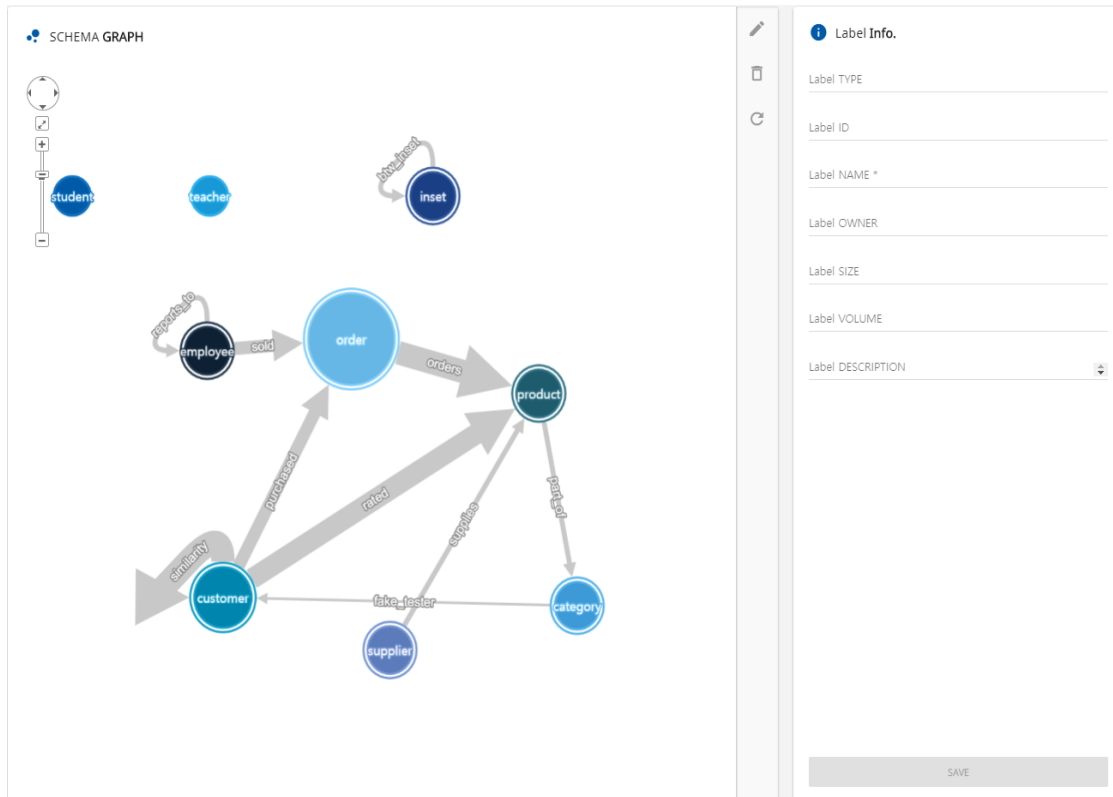
ID
3

NAME
product

SIZE
77

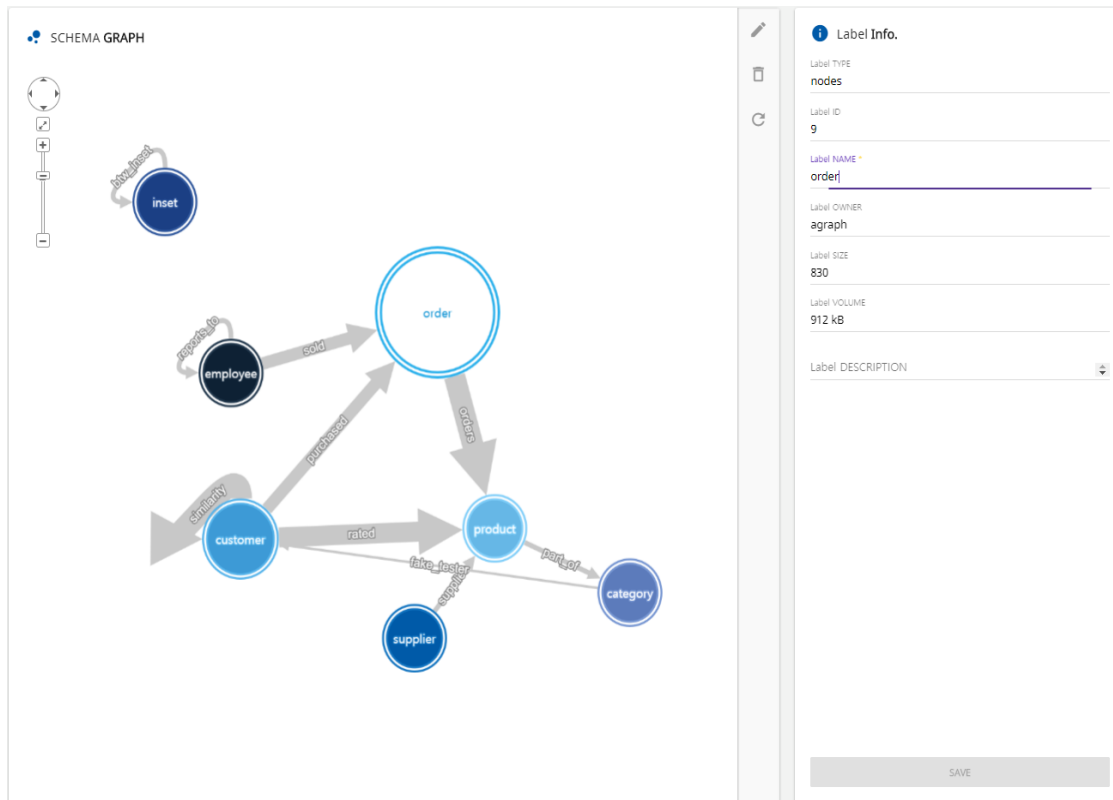
Schema Graph & Label Info

Meta 데이터의 node label 과 edge label 의 관계를 그래프로 보여준다.



각각의 node 및 edge 를 선택하여 Label Info 에서 상세 내역을 확인할 수 있다.

다음은 order 이라는 node label 을 선택했을 때의 결과이다.



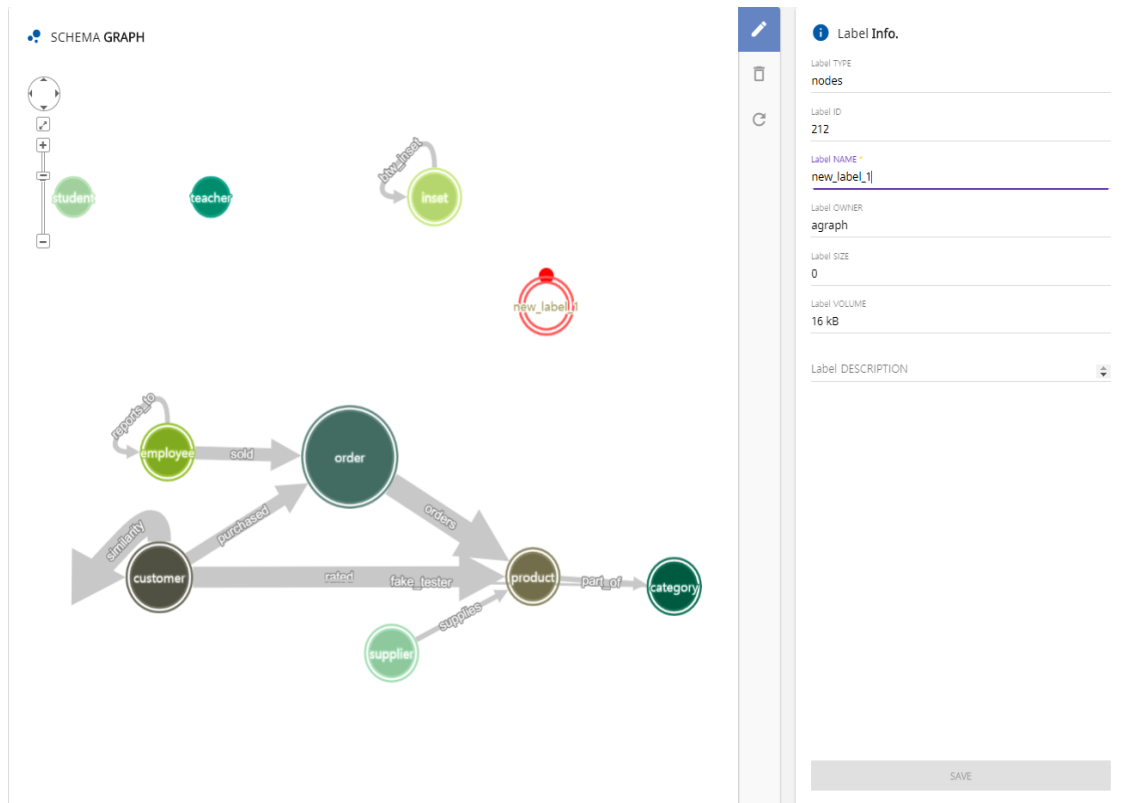
- Label 편집

Schema Graph 영역의 Edit Schema() 버튼을 누르고 node 또는 edge 를 선택하면 Label Info 가 활성화 되며 Label name, description 수정이 가능하다.

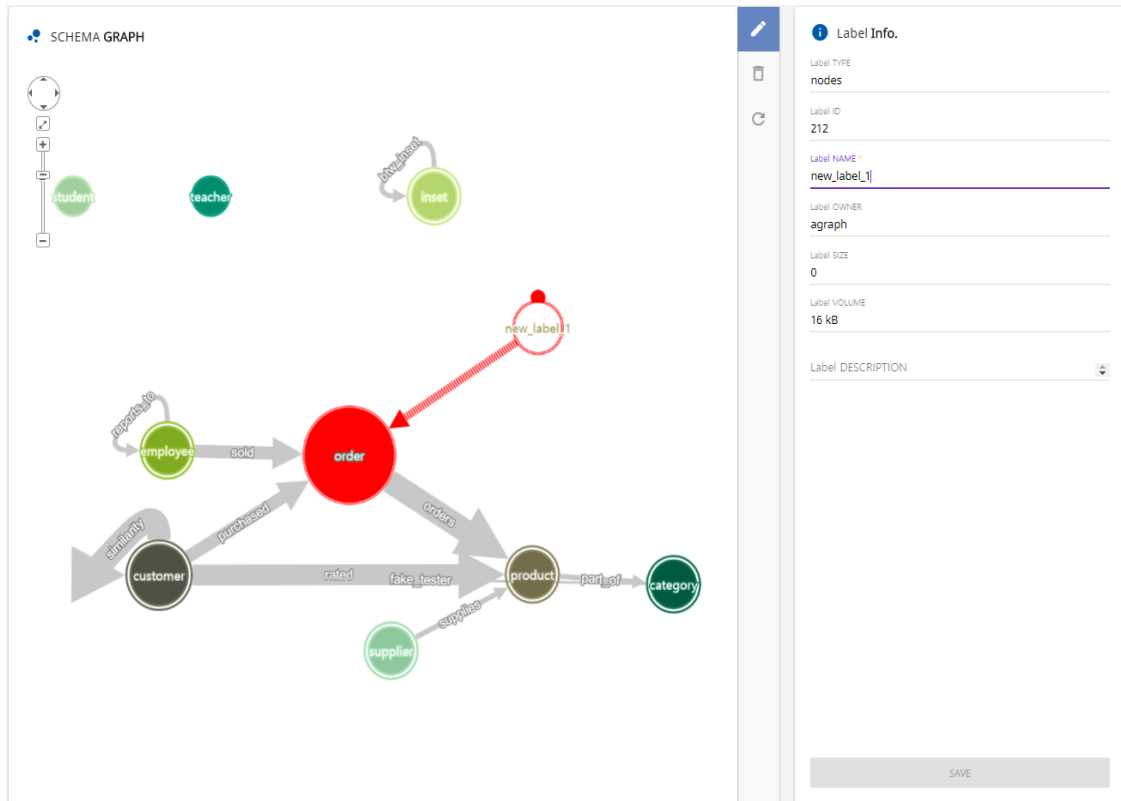
- Label 생성

Edit mode 상태에서 canvas 를 더블 클릭하면 node 를 생성 할 수 있으며, node 의 빨간점을 드래그 하면 아래와 같이 다른 node 간의 edge 생성이 가능하다.

- Node 생성



- Edge 생성



- Label 삭제

Schema Graph 영역의 Remove() 버튼을 누르고 node 또는 edge 를 선택하면 삭제될 label 에 대한 정보를 보여주고 삭제를 진행할 것인지 여부를 묻는다.

 CONFIRM INFO

Are you sure to delete this LABEL?
If you remove label, All data will also be removed!

TYPE
nodes

ID
3

NAME
product

SIZE
77

OK

CANCEL

PL/Python Menu

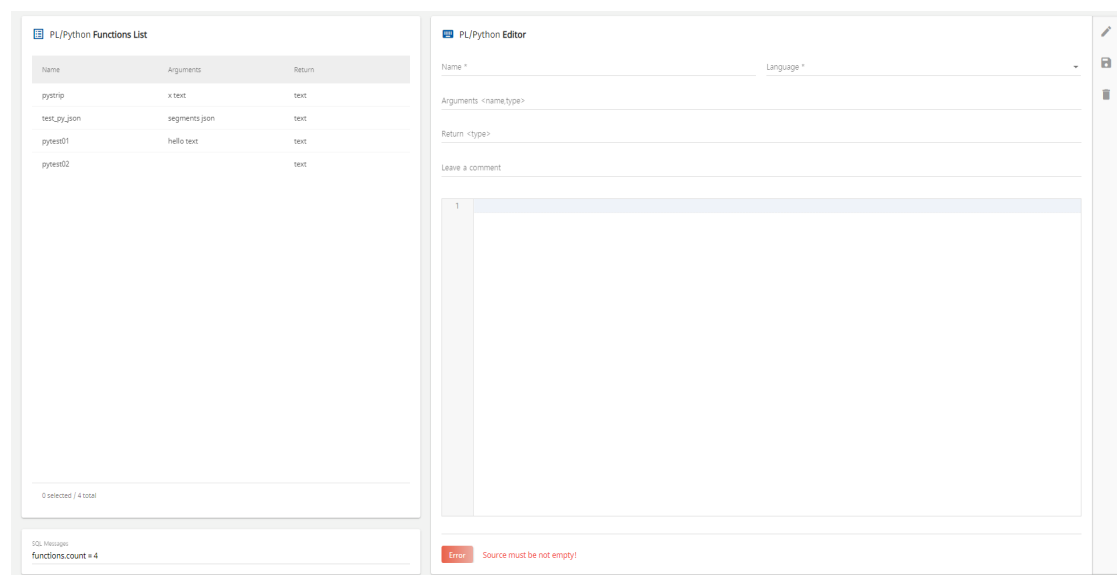
AgensGraph 에서는 PL/Python procedural language 를 사용하면 응용프로그램 내에서 python 으로 작성된 함수를 작성 및 실행할 수 있다.

PL/Python 을 사용하기 위해서는 extension 을 생성해야한다.

```
agens=# CREATE EXTENSION plpython2u;
```

PL/Python 에서 사용되는 Python 문법은 기존 Python language 를 따르며 자세한 문법은 [Developer Manual](#) 을 참조한다.

AgensBrowser 의 PL/Python 영역에서는 AgensGraph 데이터 베이스에 생성된 PL/Python Function list 조회 및 Function 생성이 가능하다.



PL/Python Functions List

AgensGraph 에 생성된 PL/Python Function 목록이다. 생성된 Functions 을 누르면 PL/Python Editor 에서 Functions 의 상세 내용을 확인할 수 있다. Function Editor 는 다음에서 설명하도록 한다.

PL/Python Functions List

Name	Arguments	Return
pystrip	x text	text
test_py_json	segments json	text
pytest01	hello text	text
pytest02		text

PL/Python Editor

PL/Python Editor 를 이용하여 Function 작성이 가능하다. new() 버튼을 누르면 새로운 pl/python 을 작성할 수 있다. 또한 생성된 function 을 선택한 후 delete()버튼을 누르면

함수를 삭제 할 수 있다.

PL/Python Editor

Name *

pymax

Language *

plpython2u

Arguments <name,type>

a int, b int

Return <type>

integer

Leave a comment

pl/python test

1

if a > b:

2

return a

3

return b

- 1) Name
Function 명을 작성 한다.
- 2) Language
Internal, C, SQL, PL/pgSQL, PL/Python 을 선택할 수 있다. 단, Functions List 에는 PL/Python 에 대한 내역만 조회된다.
- 3) Arguments <name, type>
함수에서 사용될 argument name, type 을 지정한다.
- 4) Return
return 될 type 을 지정한다.
- 5) Leave a comment
Function 에 대한 간략한 설명을 작성할 수 있다.

- 6) 함수의 본문을 작성한 후 save(💾)버튼을 누르면 작성한 함수가 생성되고, list 에 업데이트 된다.

PL/Python Functions List

Name	Arguments	Return
pystrip	x text	text
test_py_json	segments json	text
pytest02		text
pytest01	hello text	text
pymax	a integer, b integer	integer

1 selected / 5 total

SQL Messages

SUCCESS: UPDATE function 'pymax' affected

PL/Python 작성 및 실행 결과의 예이다. 실행은 Graph 탭에서 수행할 수 있다.

PL/Python Editor

Name *

pymax

Language *

plpython2u

Arguments: <name:type>

a integer, b integer

Return <type>

integer

Leave a comment

test

1 if a > b:

2 return a

3 return b

<> Query Editor (Cypher Or Ansi-SQL)

1 select pymax (1, 2);

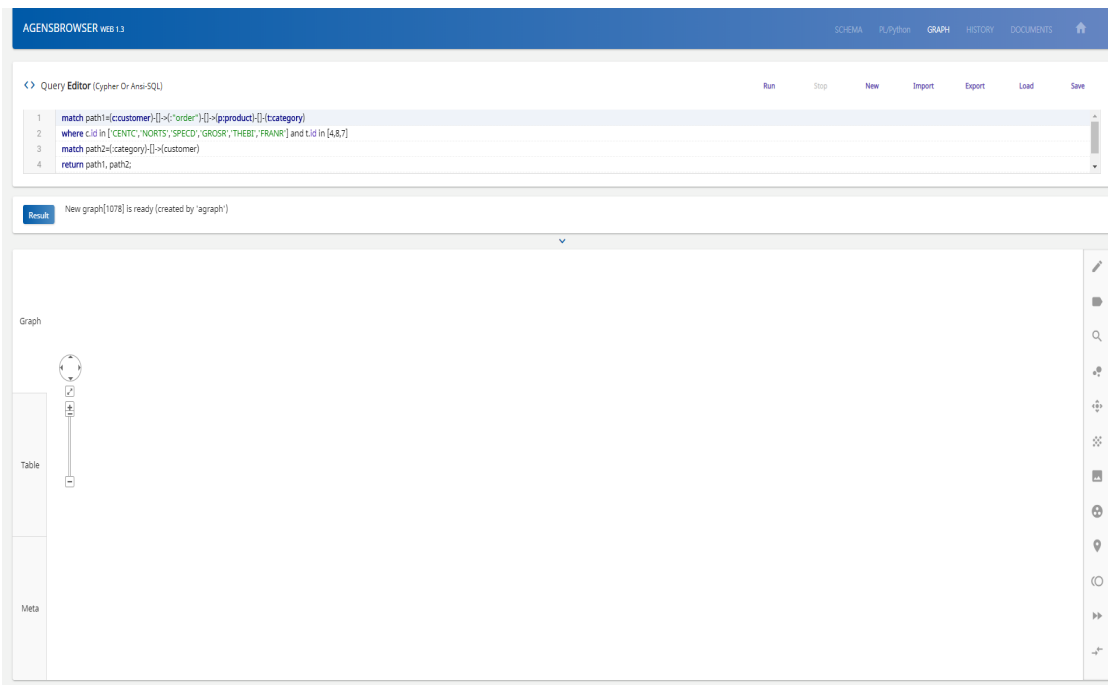
Result

return 1 rows (cols=1), no graph

Graph Menu

AgensGraph 데이터베이스에 대해 SQL 및 Cypher 쿼리할 수 있는 기능을 제공한다. 쿼리의 결과는 table 뿐만 아니라 graph 방식으로 데이터를 시각화 할 수 있으며 조회한 그래프에 대한 layout 변경 및 label style 적용, 이미지 export 가 가능하다.

Notice: Query to Graph DB 수행중 페이지를 전환하면 작업중인 이전 내용은 저장되지 않는다.



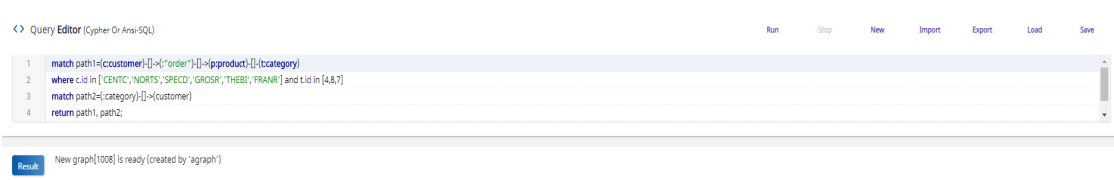
Query to Graph DB 는 다음의 항목으로 구성된다.

- Query Editor
- Result Message
- Graph Tab
- Table Tab
- Meta Tab

Query Editor

AgensGraph 데이터베이스에 대해 SQL 및 Cypher 쿼리를 수행 할 수 있는 기능을 제공한다. 주석은 -- (Ctrl + /)으로 작성 가능하다. 질의 결과는 세가지 타입(Graph, Meta, Table)으로 제공되며 각 Tab 에서 확인할 수 있다.

Notice: Agensbrowser 접속시 yml 에 지정된 graph_path 로 로그인되며, graph_path 변경(set graph_path) 명령어는 동작하지 않는다.



다음은 상단의 아이콘에 대한 설명이다.

- Run (**Run**)

작성한 쿼리를 실행한다.

해당 버튼을 누르면 쿼리가 수행되며 result 에 수행 결과 또는 에러 메시지를 보여준다. 마우스로 선택 영역 문장만 실행 가능하다.

- Stop (**Stop**)

실행중인 쿼리를 중지한다.

- New (**New**)

New 버튼을 이용하여 새로운 graph file 을 생성한다.

해당 버튼을 누르면 editor 및 result 영역이 초기화 된다.

- Import (**Import**)

Graphson type 으로 export 한 데이터로 Import 를 수행한다. Import 가 성공하면 아래와 같은 메시지가 출력된다.



- Export (**Export**)

수행한 Query 에 대한 결과를 graphson type 으로 Export 한다. 다운로드 폴더에 저장되며, graph_xxxx.graphson 으로 저장된다.

Tip: Graphson type 은 개별 그래프 요소 (node, edge)에 대한 JSON 기반의 형식이다. 자세한 사항은 [GraphSON-Format](#) 를 참고한다.

- Load ([Load](#))

Save 된 Project 를 load 할 수 있다. Load 버튼을 누르면 아래와 같이 Save 된 Project 의 목록을 확인할 수 있다.

The screenshot shows the 'User Projects' interface. At the top, there's a 'User Projects' header with a folder icon and a refresh button. Below it, a subtitle reads 'Select project and edit query'. A search bar with a magnifying glass icon and the placeholder text 'Type to filter the title column' is positioned above a table. The table has five columns: 'ID', 'Title', 'Create Date', 'Update Date', and 'Remove'. It lists seven projects with IDs ranging from 1099 down to 1033. Each row has a trash can icon in the 'Remove' column. Below the table, it says '0 selected / 7 total'. To the right of the table is a 'Preview' panel showing a graph visualization with nodes and edges. A 'Close' button is at the bottom right of the preview panel.

ID	Title	Create Date	Update Date	Remove
1099	image02	12/06 14:19	12/06 14:19	
1098	create node test	12/05 15:45	12/05 15:45	
1097	image upload test02	12/04 23:21	12/04 23:21	
1065	test	12/03 10:59	12/03 10:59	
1035	grouping01	11/20 01:27	11/20 01:27	
1034	base01	11/20 01:11	11/20 01:11	
1033	test19	11/19 22:11	11/19 22:11	

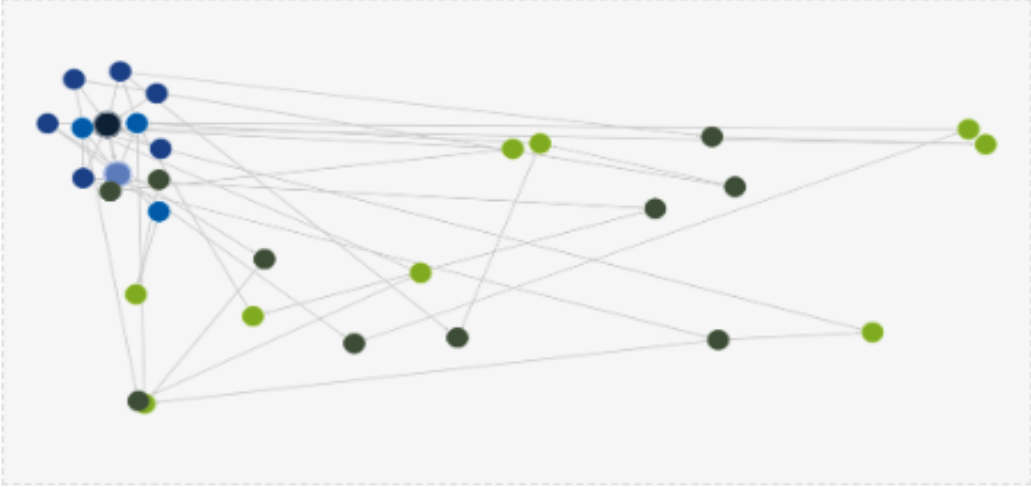
목록의 project 를 선택하면 저장된 project 에 대한 수행 Query 및 Graph 결과를 Graph 영역에서 확인할 수 있다. project 를 load 하기 전 preview 를 확인 할 수 있으며, Remove()를 선택하여 저장된 Project 를 삭제 할 수도 있다. 또한 특정 project 의 title 을 검색할 수 있다.

- Save ([Save](#))

수행한 쿼리 결과를 저장할 수 있다. Save 버튼을 누르면 아래와 같이 project 를 저장할 수 있다. Title 은 필수 기입 사항이며, 해당 project 에 대한 설명을 Description 에 선택 사항으로 작성 할 수 있다.

 **Project Save**

Type title and description about your project



ID

Title *

Description

Create Date

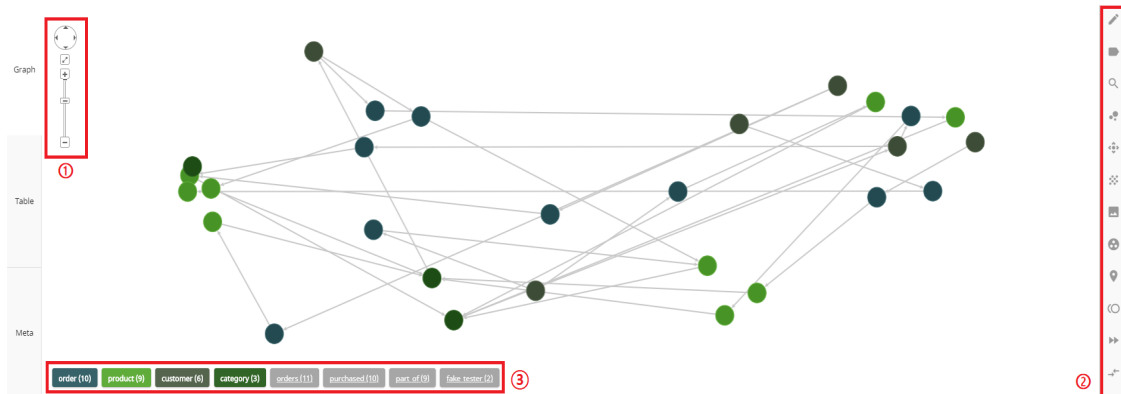
Update Date

Submit

Cancel

Graph Tab

수행한 쿼리에 대한 결과를 그래프 형태로 시각화하여 보여준다.



그래프 편집기에서 지원되는 작업은 다음과 같다.

- Graph 편집/수정
- Graph 스타일링
- Graph 변형
 - group, filter
- Graph 탐색
 - 중심점 탐색
 - title 탐색
 - 최단 경로 탐색
 - 연결 컴포넌트
 - 타임 라인

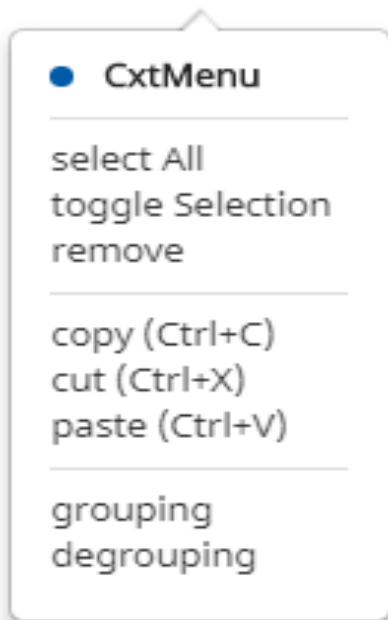
- 원형 고리
- 오버레이 (완전/부분 매칭)

그래프창의 왼쪽 상단에 위치해 있는 컨트롤러 (①) 를 이용해 화면을 pan(위치이동) 시키거나 zoom(확대축소) 시킬수 있다. 버튼으로 화면맞춤이 가능하다.

먼저 Graph 영역에서 graph 편집/수정 및 Label Style 변경에 대한 설명이다.

- Graph 편집/수정

Graph 영역에서 오른쪽 마우스를 클릭하면 아래와 같은 팝업 메뉴가 활성화 된다.
전체 선택, toggle 선택, 삭제, 복사/잘라내기/붙여넣기 및 그룹화가 가능하다.



<단축키>

- * 전체선택 : Ctrl + A
- * 편집모드 : Ctrl + E
- * Copy : Ctrl + C
- * Cut : Ctrl + X
- * Paste : Ctrl + V
- * Undo : Ctrl + Z
- * Redo : Ctrl + Y

- Label Styles

아래의 Label 명을 선택한다.




NODES Editor

- ID

8.26

- Label

customer

- Properties

key	value	type	del
__new__	__new__	STRING	
address	54, rue Royale	STRING	
city	Nantes	STRING	
companyname	France restauration	STRING	
contactname	Carine Schmitt	STRING	

15 total

1

2

3

Add Property
Update



2) Edit Node/Edge

edit mode 버튼을 누른 후 canvas 를 더블 클릭하면 node 및 edge 를 생성 할 수 있다.
label 을 지정하고, property 를 추가할 수 있다.


NODES Editor

- ID

_id_amLct

- Label

label name

- Properties

key	value	type	del
No data to display			

0 total

Add Property
Update



EDGES Editor

• ID

8a16ea7d-c9e8-444c-9b95-

• Source-Node ID

_id_amLct

• Target-Node ID

4,5

• Label

label name

• Properties

key	value	type	del
-----	-------	------	-----

• No data to display

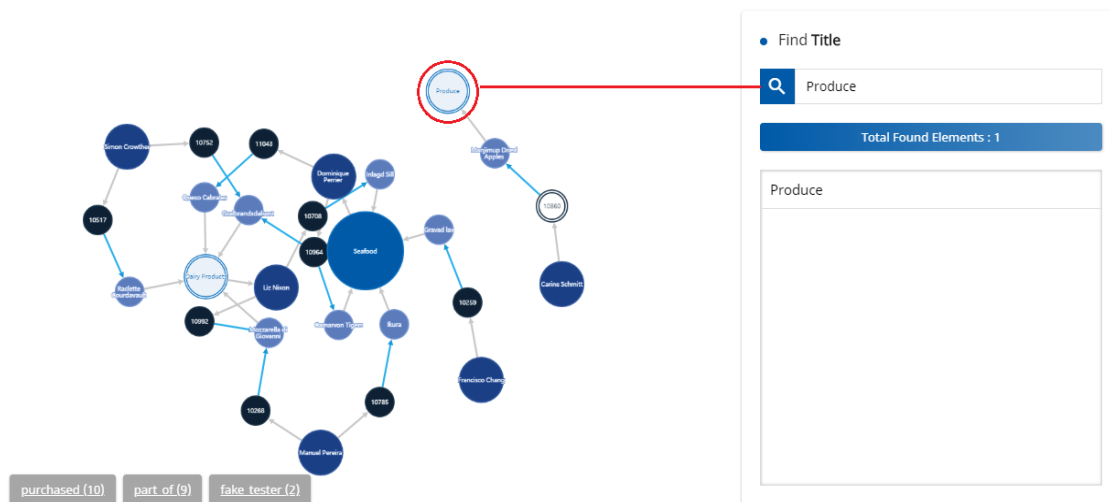
0 total

Add Property

Update

• Show/Hide Title ()

Title 을 보여주거나 숨길 수 있으며, Title 로 원하는 node label 을 검색 할 수 있다.



- Filtering & Grouping (🔍)

Label 을 property 를 기준으로 filtering 하거나 grouping 할 수 있다.

Search In Result

• Label Info.

Label nodes [count=10]

Name order

Properties (17)

- \$ALL (for groupby)
- shippostalcode (string)
- shipcountry (string)
- orderid (number) ☒
- freight (number)

+ Group + Filter Refresh

• Filter By List

order
orderid (number)

eq 4

• GroupBy List

order
orderid (number)

Reload

- filterBy : 사용자 쿼리 그래프를 대상으로 사용자 입력 조건으로 filter 처리한 그래프로 변형

다음은 Order 라는 node 의 properties 중 shipcountry 에서 'UK'를 포함하는 node 를 찾는 예이다.

Search In Result

• Label Info.

Label nodes [count=10]

Name order

Properties (17)

- \$ALL (for groupby)
- shippostalcode (string)
- shipcountry (string) ☒
- orderid (number)
- freight (number)
- shipaddress (string)

+ Group + Filter Refresh

• Filter By List

order
shipcountry (string)

contains UK

• GroupBy List

Reload

- **Layout Type**

Random (Default)

Grid

Breadth-First

Concentric

Cola

Cose

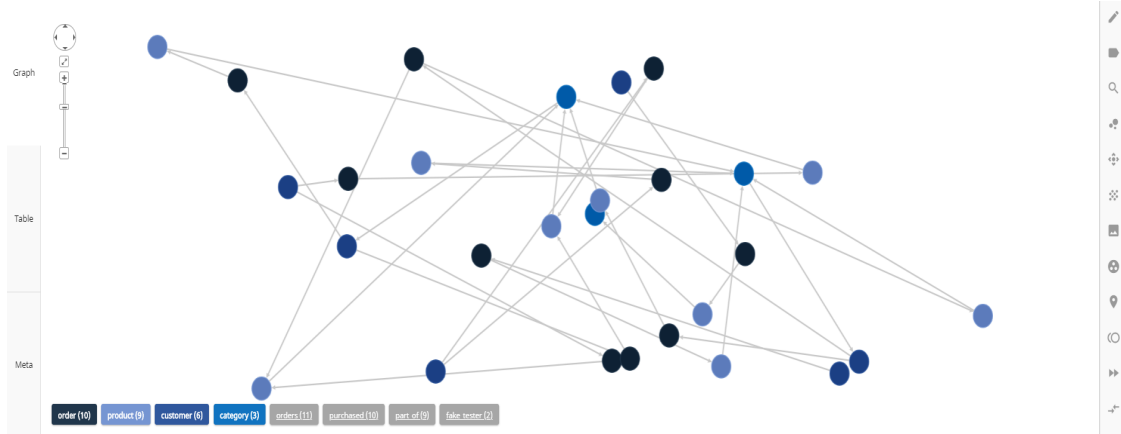
Cose-Bilkent (Slow)

Dagre (Hierarchy)

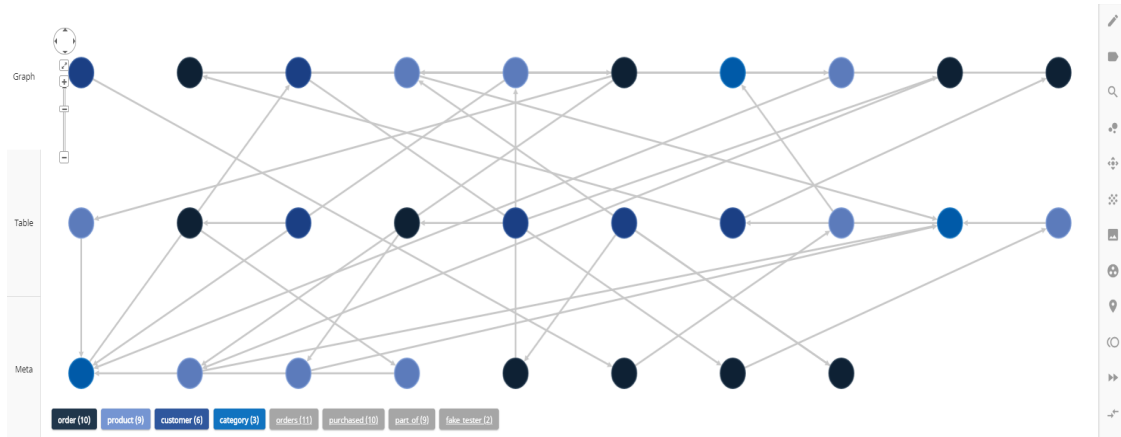
Klay (Hierarchy)

Euler (Very Fast)

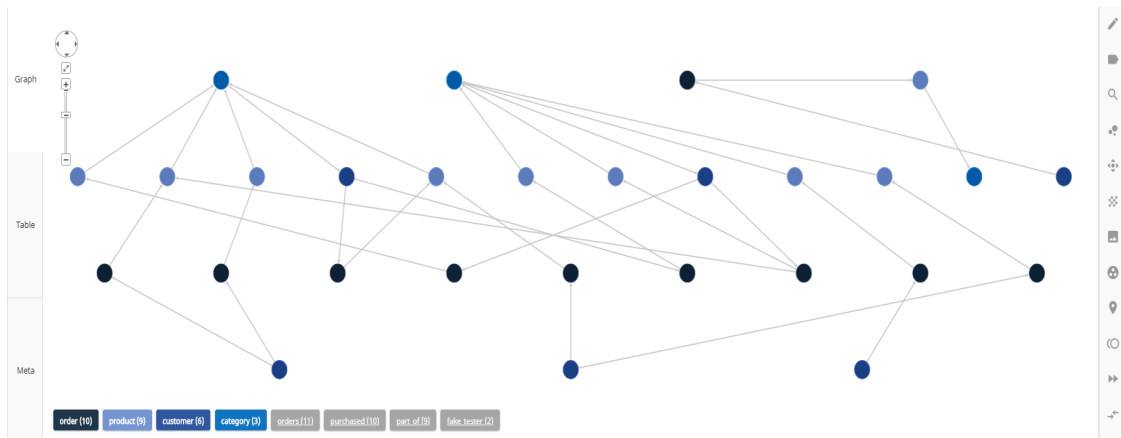
1) Random (Default) : 의외의 형태를 발견할 수 있도록 임의로 노드를 배치



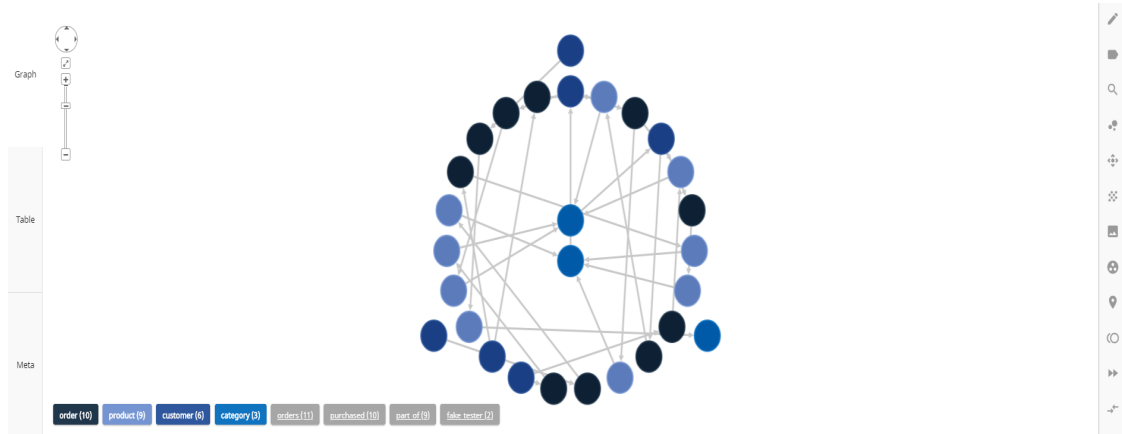
2) Grid : 격자 구조의 그리드 영역에 노드를 배치



3) Breadth-First : 간선이 많은 노드를 상단에 배치하고 아래쪽에 딸린 노드를 배치하는 너비우선 방식



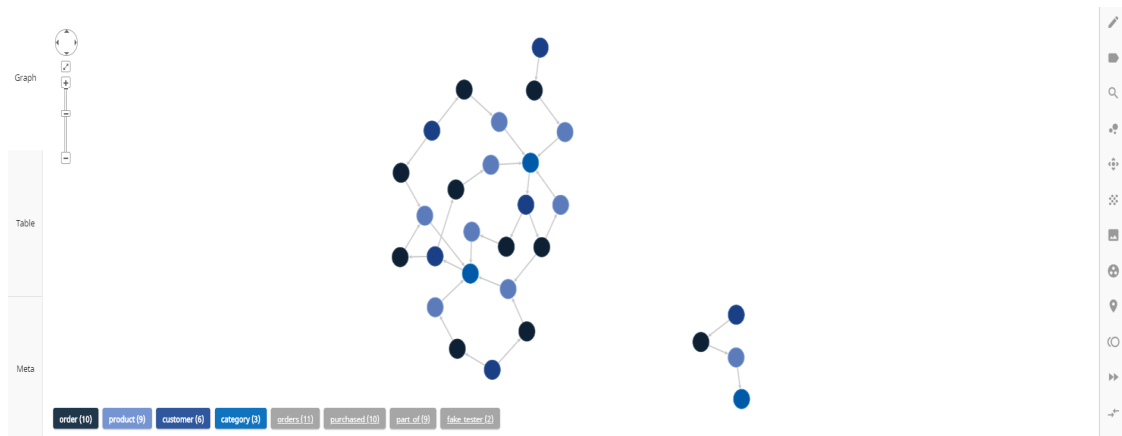
4) Concentric : 간선이 많은 노드를 중심에 두고 연결된 노드를 바깥쪽에 배치



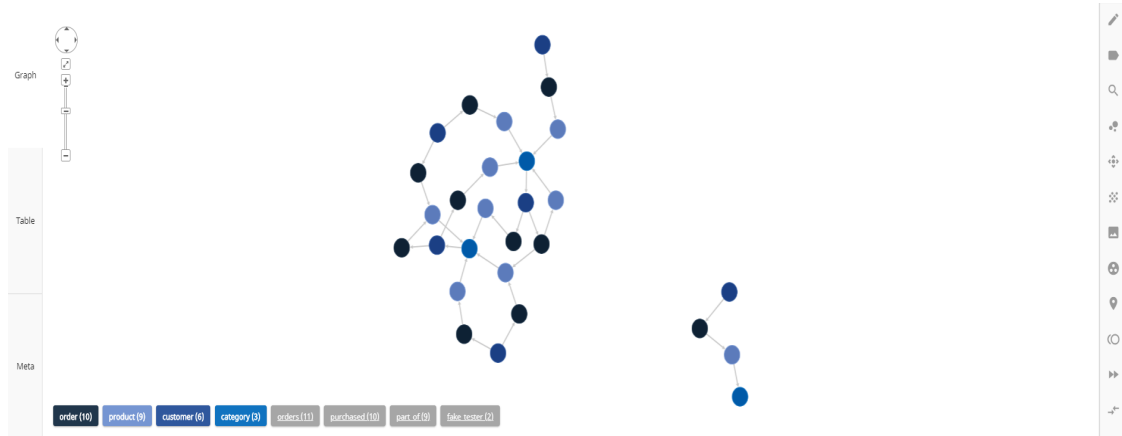
5) Cola : 연결된 노드가 많은 덩어리가 위쪽에 위치하도록 배치



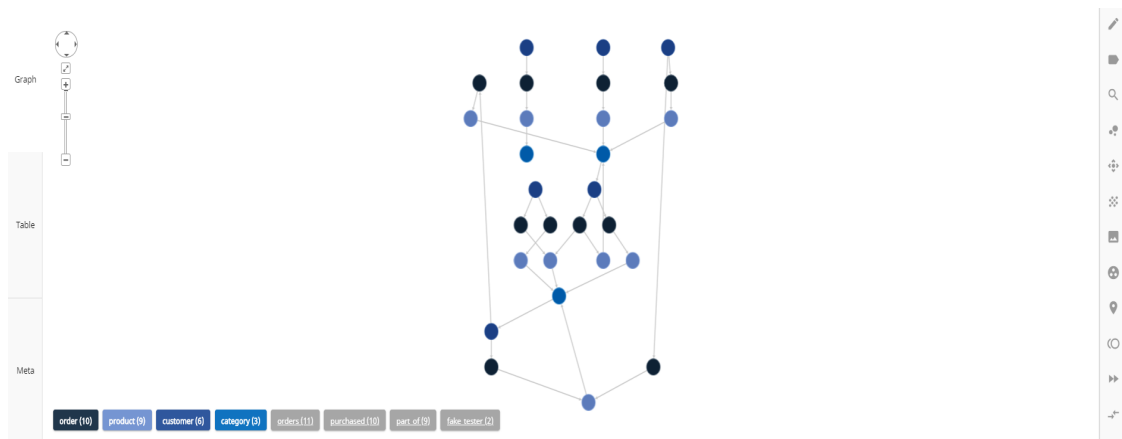
6) Cose : 연결된 노드 군집간의 거리를 두어 군집 분포를 알아보기 쉽게 배치



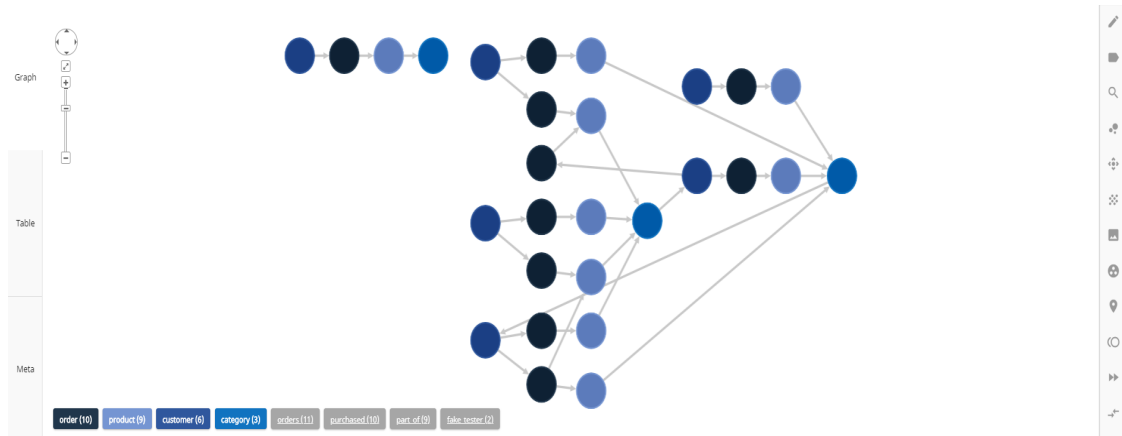
7) Cose-Bilkent (Slow) : Cose 배치를 수정한 방식



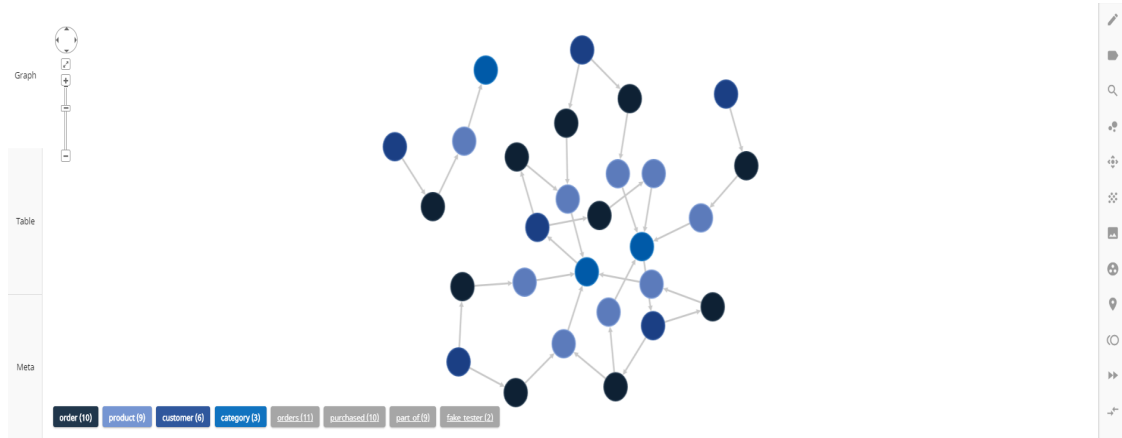
8) Dagre (Hierarchy) : 연결된 노드 군집을 수평으로 배치하여 종류를 구분하기 쉽도록 도움



9) Klay (Hierarchy) : 좌에서 우로 수평방향으로 계층적 레이아웃 형성



- 10) Euler (Very Fast) : quad-tree 자료구조를 이용한 force-directed 최적화로 큰 그래프도 빠른 시간내에 배치



- Centrality (🔍)
Betweenness, Closeness, Degree, PageRank 및 그래프 데이터의 속성값 기준으로 node 의 크기를 변경시키는 byValue 기능을 제공한다. 스타일 적용을 초기화 시키려면 Reset Styles 을 선택한다.

● Centrality Methods

Reset Styles

Betweenness

Closeness

Degree

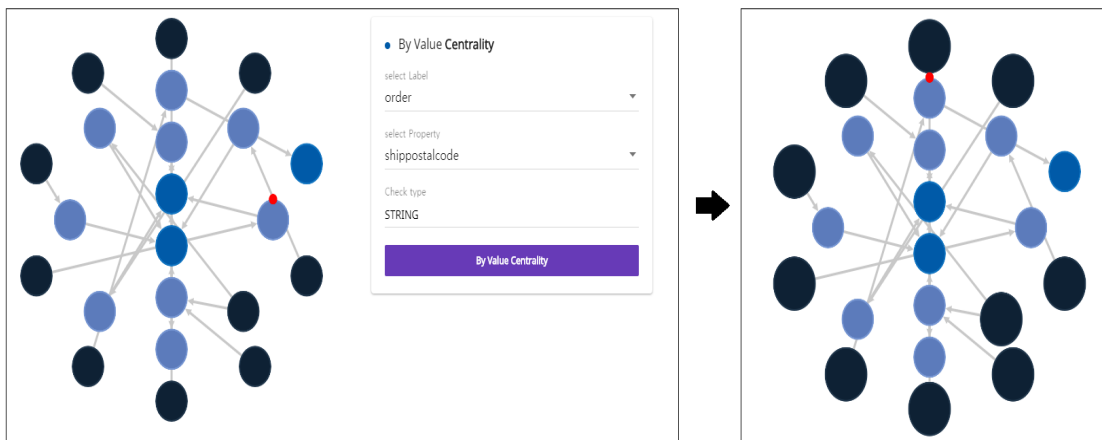
PageRank

By Value

Betweenness, Closeness, Degree, PageRank 에 대한 간단한 설명이다.

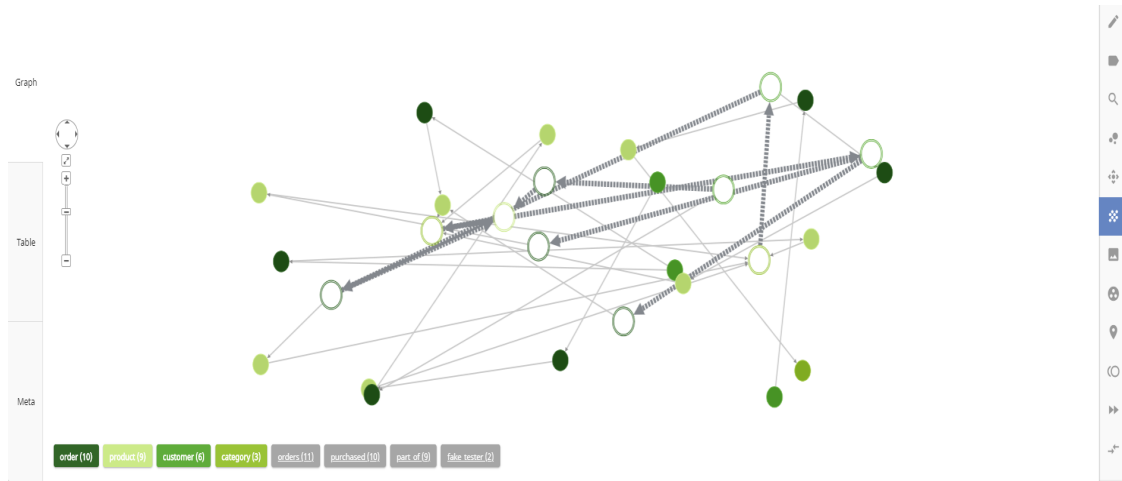
- Betweenness : 네트워크의 모든 노드 쌍 간의 shortest path 가 해당 노드를 지나는지를 고려한 centrality 이다.
- Closeness : 네트워크의 모든 node 로부터 얼마나 가깝게 위치해있는지를 고려한 centrality 이다.
- Degree : 각 node 별로 직접 연결된 edge 의 weight 만을 고려한 centrality 이다.
- PageRank : 링크 분석 알고리즘으로 참조하는 node 와 피참조 node 간의 following 관계를 가중치로 할당하여 네트워크상의 상대적인 중요성을 측정한다.
- ByValue : 그래프 데이터의 속성값 기준으로 node 의 크기 고려한 centrality 이다.

다음은 label 의 속성값을 기준으로 node 의 크기를 변경 시킨 결과의 예이다.



- Highlight Neighbors (🔍)
Neighbor node 를 구분할 수 있는 highlight 기능을 활성/비활성화할 수 있다. 이 기능으로 인접 node(하위 3 단계까지) 중 edge 의 label 이 중복되지 않는 node 를

neighbors 로 선택할 수 있다.



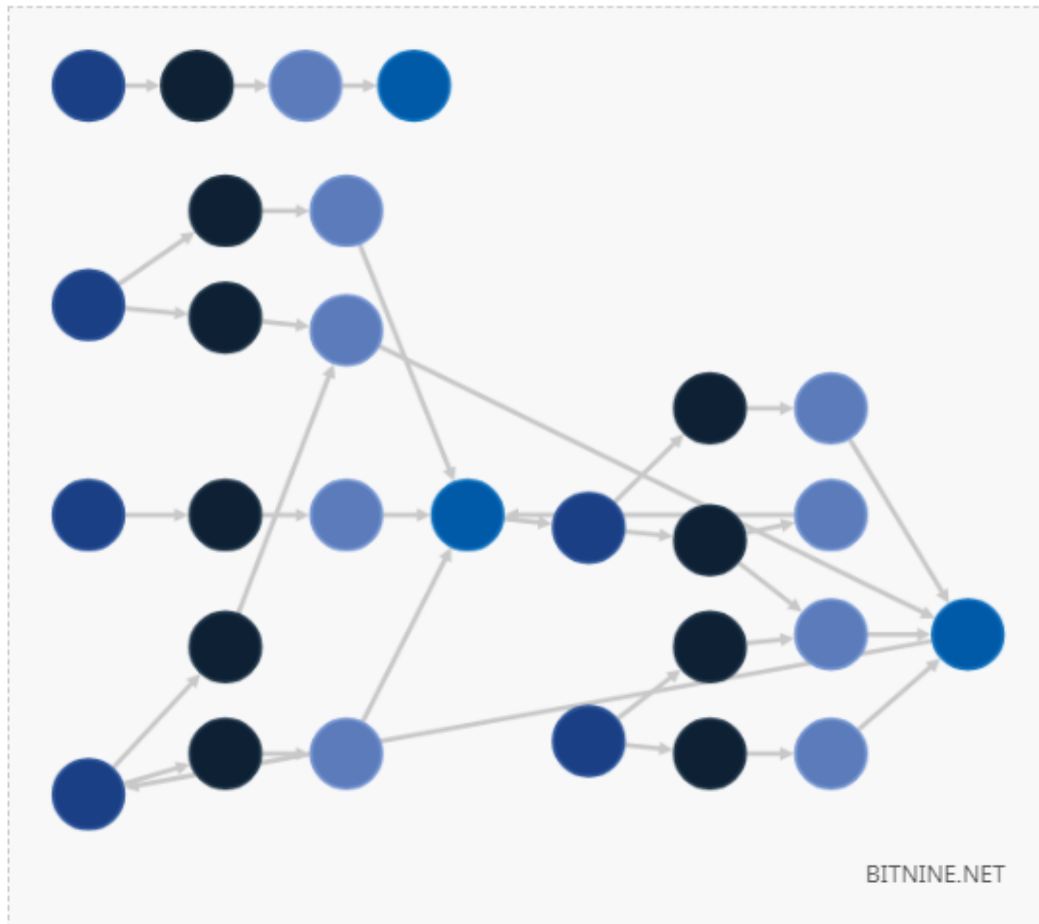
- Image Export ()
수행한 결과를 이미지 (PNG) 로 저장할 수 있다. water-mark 설정이 가능하며,

저장된 이미지의 바탕색이 투명하여 프레젠테이션 자료로 활용하기에 유용하다.



Image Export

Save PNG image file with watermark



Water-mark

BITNINE.NET

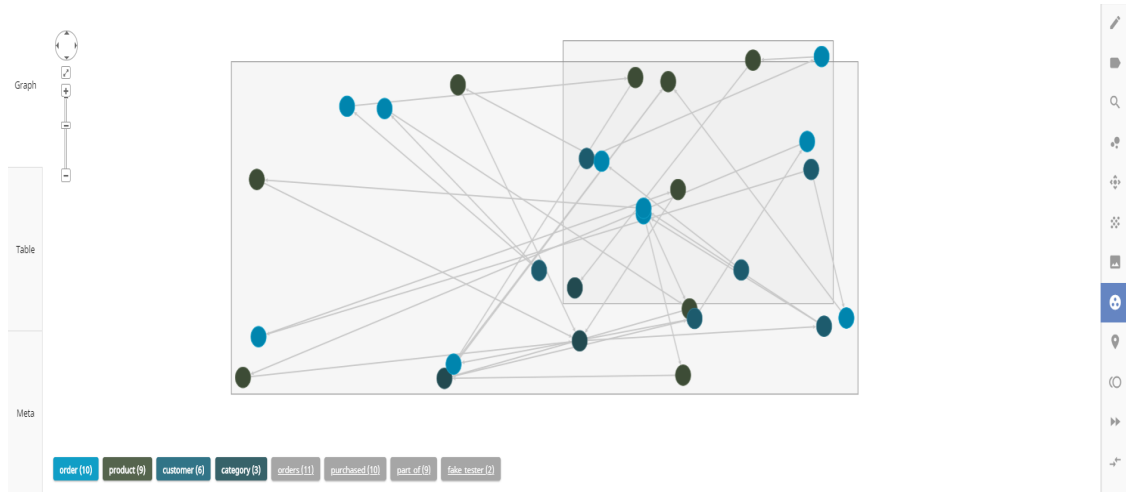


File name *

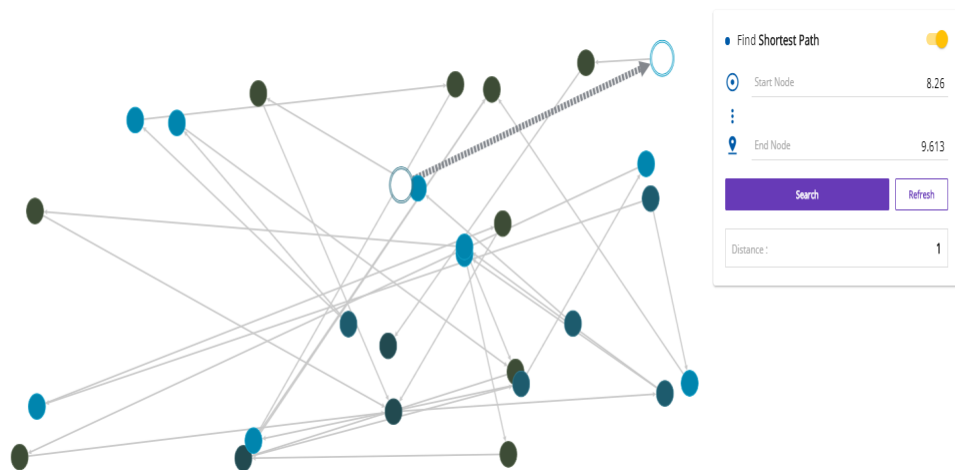
Submit

Cancel

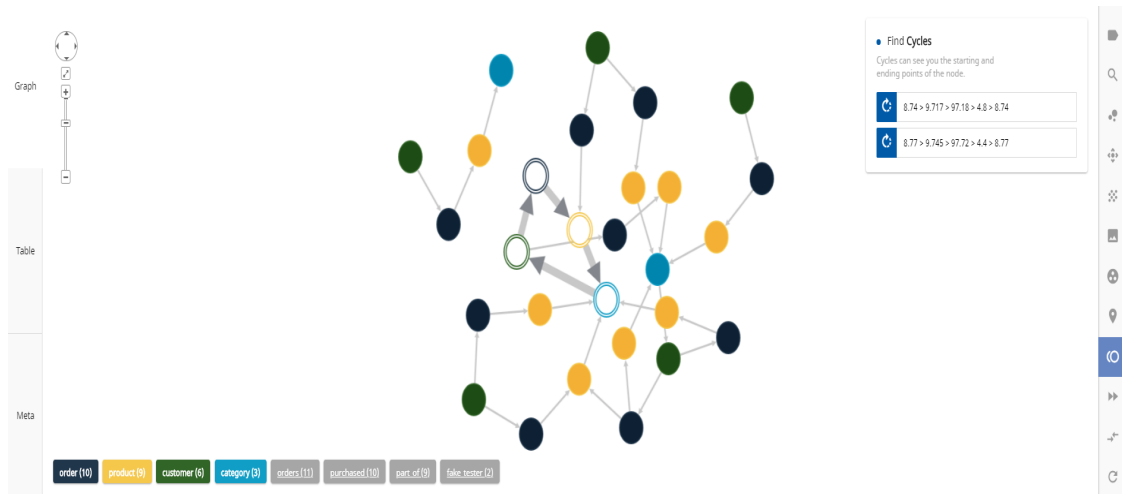
- Find Connected Group (🔗)
연결된 컴포넌트를 확인할 수 있다.



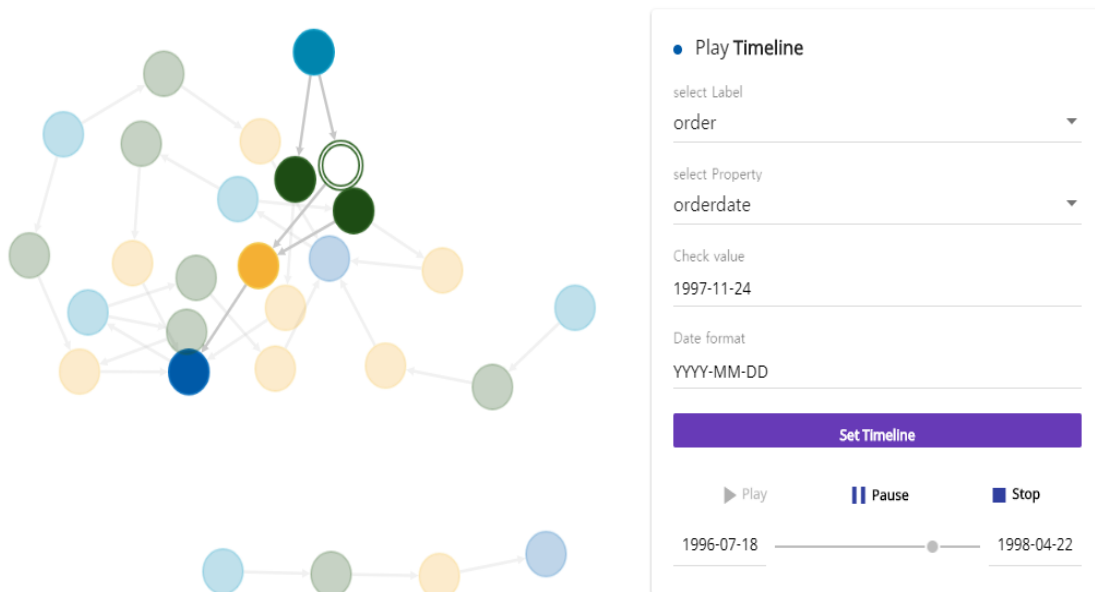
- Find Shortest Path (📍)
최단경로를 확인 할 수 있다. Start Node 및 End Node 를 각각 선택 한 후 Search 를 수행하면 Start Node 와 End Node 간의 shortest path 결과를 보여준다. Directed Option(👉)을 선택하면 양방향 결과를 확인 할 수 있다.



- Find Cycles (🔄)
원형고리 탐색(Cycle detection)이 가능하다.

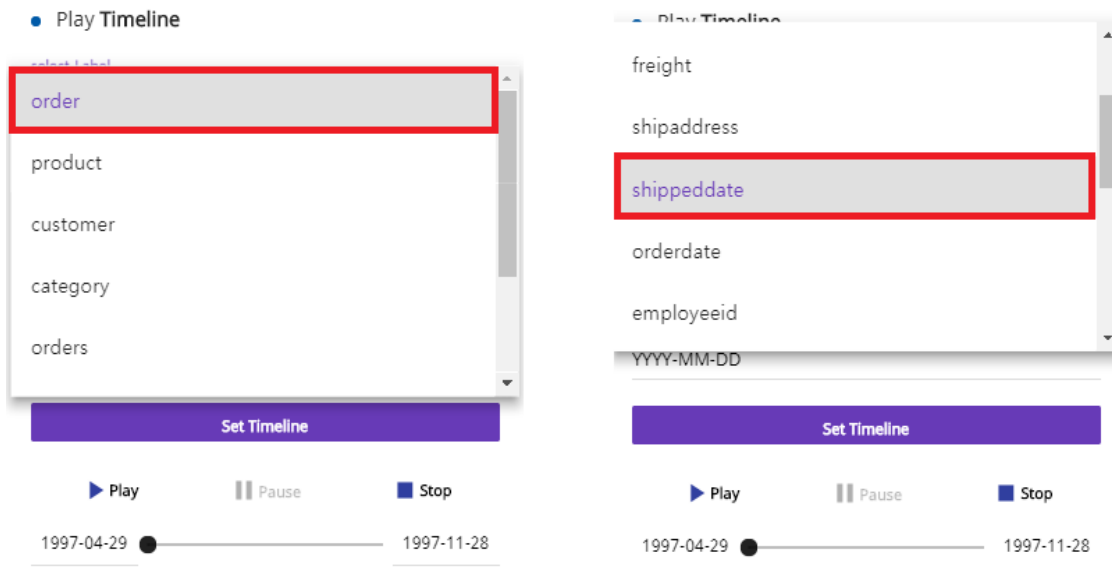


- Play Timeline (▶▶)
Date 를 기준으로 흐름을 보여주는 기능이다. 선택한 Property 의 Date 또는 Time 값 형식을 기입한다. (default 로 YYYY-MM-DD 설정)



Play Timeline 을 사용하는 예이다. select Label 항목으로 label 을 선택한 후, select Property 에서 type 이 date type 인 property 를 선택한다. Set Timeline 버튼을 누르면 데이터가 삽입된다. Play 버튼을 누르면 timeline 이 동작하며, Date 컬럼을 기준으로 최소

날짜부터 최대 날짜까지 3 초 간격(interval 조정불가)으로 동작한다.



- Overlay Graph (↔) 저장된 프로젝트를 불러와 현재 그래프 결과와 완전매칭(동일 ID 의 노드 매칭) 또는 부분매칭(동일 ID 중심으로 연결된 이웃 노드에 대해 동일 Label 과 동일 Direction 매칭) 으로 그래프를 병합할 수 있다. 화면의 오른쪽에서 프로젝트의 Matching Test 를 확인할 수 있으며, project 목록 중 원하는 project 를 선택하여 결과를 확인할 수 있다.

User Projects (Select Project And Overlay Test)

ID	Title	Create Date	Update Date
1098	create node test	12/05 15:45	12/05 15:45
1097	image upload test02	12/04 23:21	12/04 23:21
1065	test	12/03 10:59	12/03 10:59
1035	grouping01	11/20 01:27	11/20 01:27
1034	base01	11/20 01:11	11/20 01:11
1033	test19	11/19 22:11	11/19 22:11

6 total

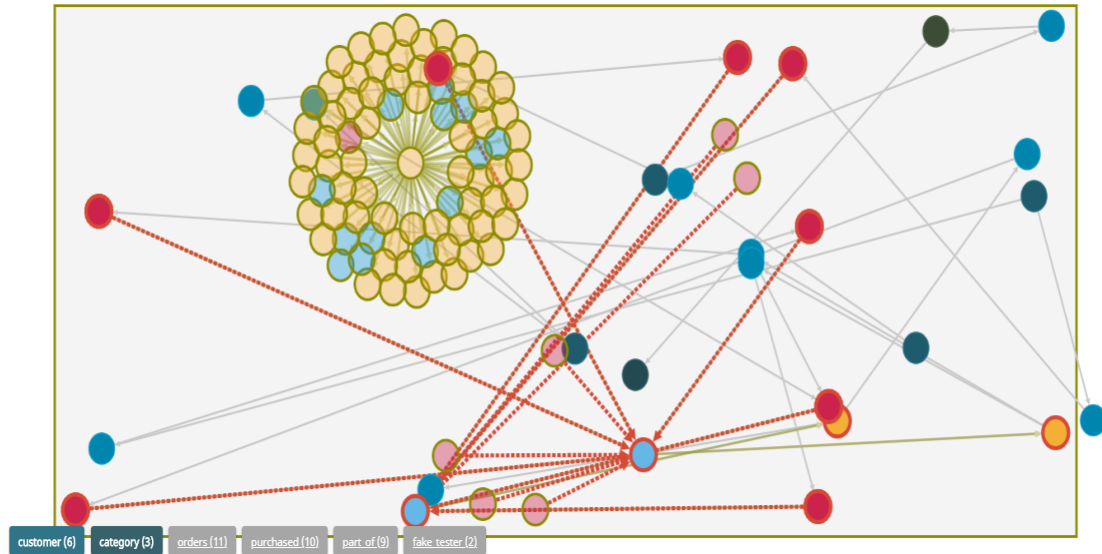
Matching Test

test19[1033]: match=9, union=30 (source=28, target=11)

OK

Close

그래프를 병합한 결과이다. 완전매칭의 경우 빨간 실선으로 표시되며, 부분매칭의 경우 점선으로 표시된다.



- Reload Graph ()
결과를 확인하고 원복할 수 있다.

Table Tab

수행한 쿼리에 대한 테이블 결과를 보여준다.

	Index	path1	path2
Graph	0	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
	1	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
	2	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
	3	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
Table	4	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
	5	{ "nodes": [{ "data": { "size": 1, "id": "4.4", "label": "category", "props": { "name": "Dairy Products", "description": "Cheeses", "categoryname": "Dai..."	
	6	{ "nodes": [{ "data": { "size": 1, "id": "8.26", "label": "customer", "props": { "country": "France", "address": "54, rue Royale", "city": "Nantes", "conta..."	
	7	{ "nodes": [{ "data": { "size": 1, "id": "9.710", "label": "product", "props": { "supplierid": 4, "productid": 10, "reorderlevel": 0, "quantityperunit": "12..."	
Meta	8	{ "nodes": [{ "data": { "size": 1, "id": "9.717", "label": "order", "props": { "shippostalcode": "75016", "shipcountry": "France", "orderid": 10964, "freig...	
	9	{ "nodes": [{ "data": { "size": 1, "id": "4.8", "label": "category", "props": { "name": "Seafood", "description": "Seaweed and fish", "categoryname": "S..."	
	22 total		14 < 1 2 3 > H

Table 내 Cell 을 클릭하면 다이얼로그 박스로 전체 값을 확인할 수 있다. 또한 다이얼로그 박스의 copy 버튼을 누르면 전체 값이 clipboard 에 복사가 된다.

 **Cell Value**

You can see the data in the table in a different style,
and you can right-click to copy the data

COPY

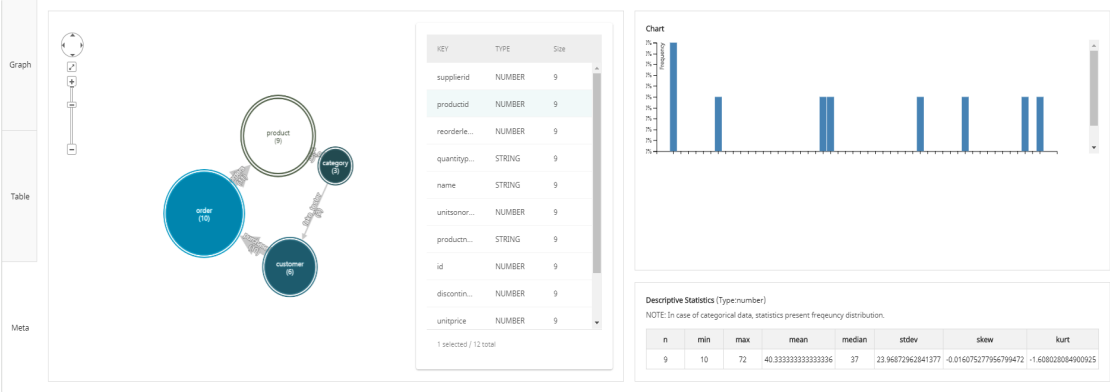
```
{
  "nodes": [
    {
      "size": number 1,
      "id": string "4.8",
      "label": string "category",
      "props": {
        "name": string "Seafood",
        "description": string "Seaweed and fish",
        "categoryname": string "Seafood",
        "id": number 8,
        "picture": string "",
        "categoryid": number 8
      }
    },
    {
      "size": number 1,
      "id": string "8.74",
      "label": string "customer",

```

Meta Tab

결과 그래프에 대한 메타 그래프 제공하고 각 라벨별 속성 정보를 확인할 수 있다. 속성 클릭시 결과 데이터 상의 빈도 막대차트와 기술통계 제공하며, 차트 및 통계 결과는 숫자와

문자 타입만 보여준다.



수행한 Query 의 내역과 결과를 조회할 수 있다.

다음과 같이 User IP 로 쿼리를 찾을 수 있다.

쿼리 부분의 상세를 확인하려면 Expand/Collapse Row() 버튼을 누른다.

쿼리 결과 상세를 확인하려면 Expand/Collapse Row() 버튼을 누른다.

Copyright © SKAI worldwide Co., Ltd. 2025. All rights reserved.

Documents Menu

AgensBrowser 사용을 위한 상세 내용은 Homepage 의 Documents 를 참고하도록 한다.

LiveShare For Report

Report 는 그래프 렌더링과 간단한 그래프 편집기능, 라벨 리스트 출력과 속성 데이터의 출력까지 사용 가능하다.

Report 기능

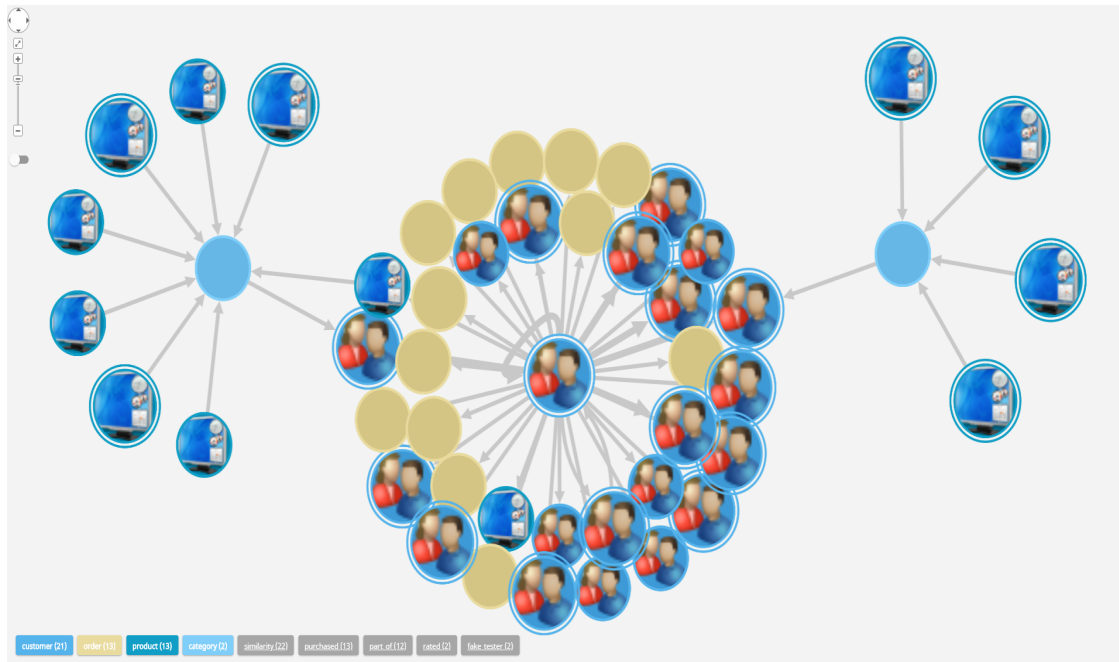
사용자가 저장한 project 를 report 용으로 확인할 수 있다. guest-key 는 config.yml 에서 작성한 내용을 참고하며, project-id 는 Graph tab 의 Load 메뉴의 ID 로 확인할 수 있다.

```
http://DB_SERVER_IP_ADDRESS:WEB_SERVER_PORT/#/report/GUEST-KEY/PROJECT-ID
```

project report 를 확인하는 예이다.

```
http://localhost:8085/#/report/agens/1130
```

project report 결과이다. Graph 영역에서 오른쪽 마우스를 클릭하면 팝업 메뉴가 활성화 된다. 메뉴 상세 내용은 [Graph 편집/수정](#)을 참조한다.



Report 의 HTML Embed 사용

전용 시각화 도구를 사용하지 않고 사용자가 저장한 그래프 시각화 정보를 embed 태그를 이용하여 또다른 웹페이지 내부에 그래프를 삽입할 수 있는 기능을 제공한다.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>AgensWeb Embed</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="assets/icons/favicon-bn.ico">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<style>
  .border-styles {
    margin: 2px 0;
    padding: 1px 3px;
    border-width: 2px;
    border-color: '#000';
    border-style: solid;
  }
</style>

</head>
<body>
<h2> Agens Browser 2.1 </h2>
<h4> is coming soon. ^^</h4>
<div>
  <embed type="text/html" class="border-styles"
    src="http://IP:PORT/#/report/agens/1130"  --embed 태그를 기입한다.
    width="800"
    height="600" />
</div>

</body>
</html>
```

HTML Embed 를 사용한 report 결과이다.

Agens Browser 2.1

is coming soon. ^^

