21.07.08 gremlin 튜토리얼 -2

노트북: 6.Graph API / DB
만든 날짜: 2021-07-12 오전 8:19 **수정한 날짜:** 2021-07-12 오전 8:20
작성자: 황인범

그래프 vertex / edge 추가

#addV - Vertex 추가

```
# 새로운 그래프를 생성 open 일경우에는 빈 것을 올리는 것 같다.)
graph = TinkerGraph.open()
# 표준 OLTP 순회 엔진을 사용하여 그래프에서 그래프 순회 소스를 만듭니다.
g = traversal().withEmbedded(graph)
#addV로 vertex를 추가하고 해당 vertex의 속성 지정해준다. add V 는 vertext를 나타
낸다.
#key : value 형태로서 id 라는 key와 1 이라는 value 값을 가진다.
#총 3개의 property를 가지며 id :1 / name : marko / age : 29 다.
v1 = g.addV("person").property(id, 1).property("name",
"marko").property("age", 29).next()
#1 개의 software 라는 vertex 와 총 3개의property를 가지며 id :3 / name : lop /
lang : java 다.
v2 = g.addV("software").property(id, 3).property("name",
"lop").property("lang", "java").next()
```

addV -> add + vertex 로서 vertex를 생성해주는 명령어

property는 해당 vertex의 속성을 나타낸다.

#addE - edge 추가

```
# add e 는 edge를 나타낸다.
g.addE("created").from(v1).to(v2).property(id, 9).property("weight", 0.4)
```

addE -> add + edge 로서 edge를 생성해주는 명령어

from 과 to 로 되어 있으며 property는 해당 다리의 속성을 나타낸다. weight 같은 경우에는 가중치라고 생각하면 된다.

여기서는 v1->v2로 연결되는 다리를 만들고 이 다리의 id값은 9 이고 가증치는 0.4 이다.

그래프 순회-단순 유지

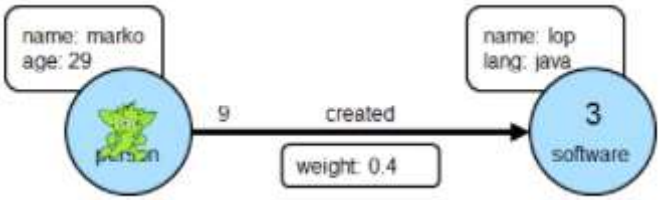
has - 찾기

1.기본적인 사용법

```
# 그래프에서 name 이 "marko" 인 것 찾기찾기
g.V().has('name','marko')
==>v[1]
```

has -> 해당 vertex를 찾을때 사용하는 것

여기서는 해당 vertex의 속성값으로 검색을한다.



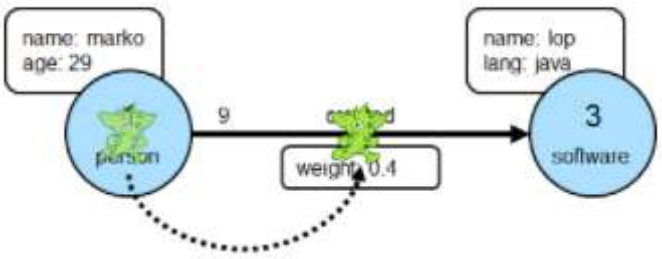
2.효과적인 has 사용법

```
# "name"속성 키가 "person"정점을 참조하도록 필터의 일부로 정점 레이블을 포함하여
개선
gremlin> g.V().has('person','name','marko')
==>v[1]
```

위와같이 person(vertex)을 지정해주면 해당 값을 찾을때 쫌 더 효과적인것 같다.

#outE - 이동

1.edge에 따라 순회하도록 요청



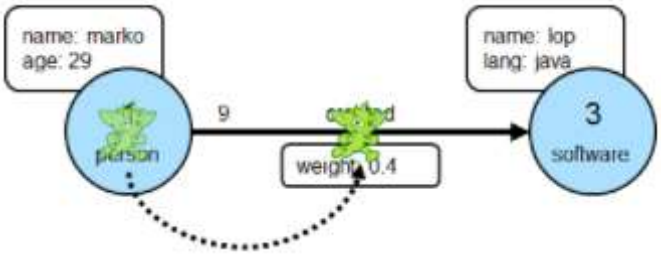

```
#Gremlin이 "marko"를 찾았으므로 그는 "생성 된"가장자리를 따라 "소프트웨어"정점으
로 "걷도록"요청
gremlin> g.V().has('person','name','marko').outE('created')
==>e[9][1-created->3]
```

해당 노드를 지정하고 outE 를통해 길을 따라 걷도록 한다.

해당 노드는 person 중에 property가 name - marko 인걸 발견하고 이것을 outE를 통해 created라는 길을 걷게 한다.

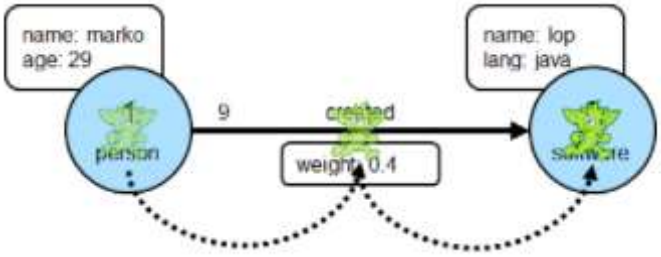
그 결과로 ==>e[9][1-created->3] 이렇게 나온다. edge9 번을 가르키는데 1-3으로 가는 것을 가리킨다.

#outE().inV() - 다른 쪽 끝에있는 정점에 도달



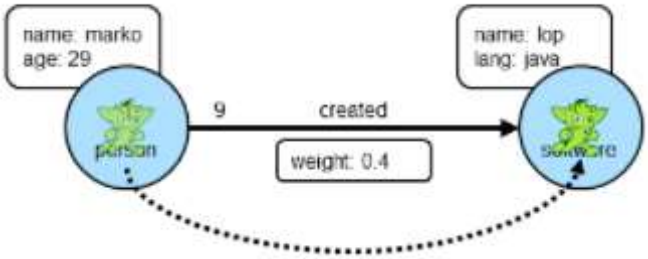

```
#가장자리의 다른 쪽 끝에있는 정점에 도달하려면 Gremlin에게를 사용하여 가장자리에서 들어오는 정점으로 이동하도록 지시
g.V().has('person','name','marko').outE('created').inV()
==>v[3]
```

#out() - 다른쪽 끝에 있는 정점에 도달




```
g.V().has('person','name','marko').out('created')
==>v[3]
```

#out().values() - 정점의 속성 엑세스




```
# Gremlin이 "Marko가 만든 소프트웨어"에 도달 했으므로 "소프트웨어"정점의 속성에 액세스
gremlin> g.V().has('person','name','marko').out('created').values('name')
==>lop
```