create (sally:Person {name:'Sally', age:32})

create (john:Person {name:'John', age:27})

create (gdb:Book {title:'Graph Databases', authors:['Ian Robinson','Jin Webber']})

match(a:Person), (b:Person) WHERE a.name = "Sally" and b.name = "John" create (a) -[:FRIEND\_OF {since:1357718400}]-> (b)

match(a:Person), (b:Book) WHERE a.name = "John" and b.title = "Graph Databases" create (a) -[:HAS\_READ {rating:4, on:1357718400}]-> (b)

match(a:Person), (b:Book) WHERE a.name = "Sally" and b.title = "Graph Databases" create (a) -[:HAS\_READ {rating:4, on :1360396800}]-> (b)

----------------------------------------------------------------------------------------------------1)John과 Sally는 친구가 된게 언제입니까?

MATCH (sally:Person {name:'Sally'}) MATCH (john:Person {name:'john'}) match (sally)-[r:FRIEND\_OF]->(john) return r.since AS friends\_since

2)Graph Databases의 평균평점은 얼마입니까?

MATCH (gdb:Book {title:'Graph Databases'}) MATCH (gdb)<-[r:HAS\_READ]-() return avg(r.rating) AS average\_rating

3)Graph Databases의 저자는?

MATCH (gdb:Book {title:'Graph Databases'}) return gdb.authors as Authors

4)Sally의 나이는?

MATCH (sally:Person {name:'Sally'}) return sally.age as Sally\_age

5)Sally와 John 중에서 누가 나이가 많은지?

MATCH (people:Person) where people.name = 'John' or people.name = 'Sally' return people.name order by people.age desc limit 1

6)Sally와 John 중에서 누가 먼저 책을 읽었지?

MATCH (people:Person) where people.name = 'John' or people.name = 'Sally' match (people) -[r:HAS\_READ]->(gdb:Book {title:'Graph Databases'}) return people.name order by r.on limit 1

1)본인을 추가

Create (heekim:Person {name: ‘Hee Kim’, age:25}) return heekim

2)관계 및 새로운 노드 추가

match (me:Person {name:'Hee Kim'}) create (me)-[like:Like]->(neo:Database {name:"Neo4j"}) return me, like, neo