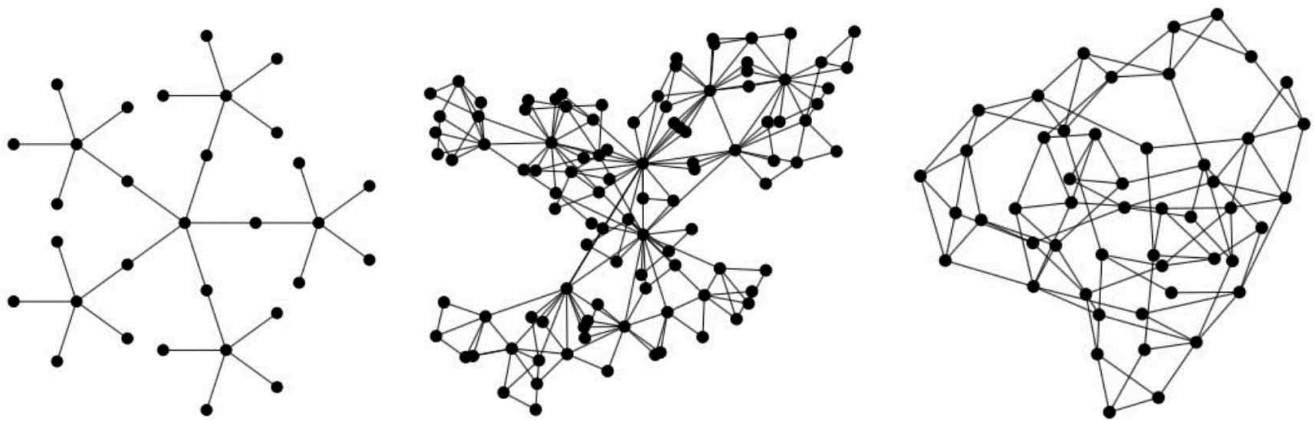


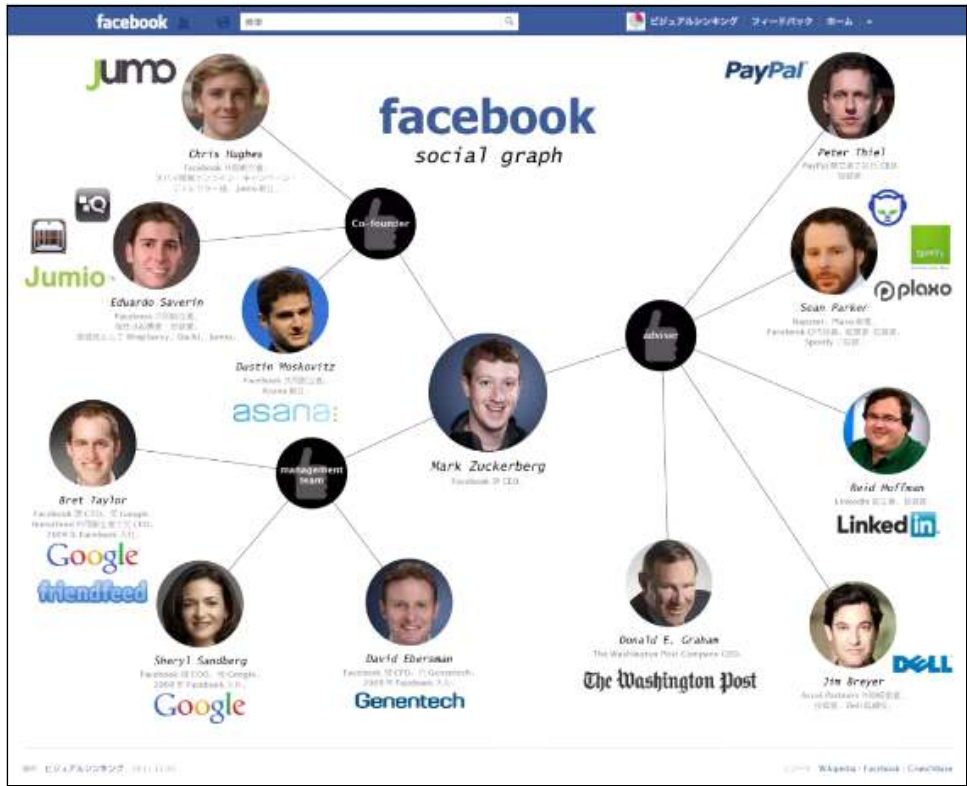
■ 그래프 데이터베이스(Graph Database) 역사

- 그래프 데이터베이스는 그래프 이론에 토대를 둔 일종의 NoSQL 데이터베이스
- 그래프(Graph): 주어진 몇 개의 점(꼭짓점)과 그 점을 끝점으로 하는 몇 개의 선(변)으로 이루어진 도형
- 수학자 오일러의 그래프 이론
- 그래프 이론: n 개의 점들로 이루어진 집합과 점들 간의 연결(관계)를 다루는 학문으로 특정 집단 내 대상들 간의 관계를 그래프로 나타내어 연구함

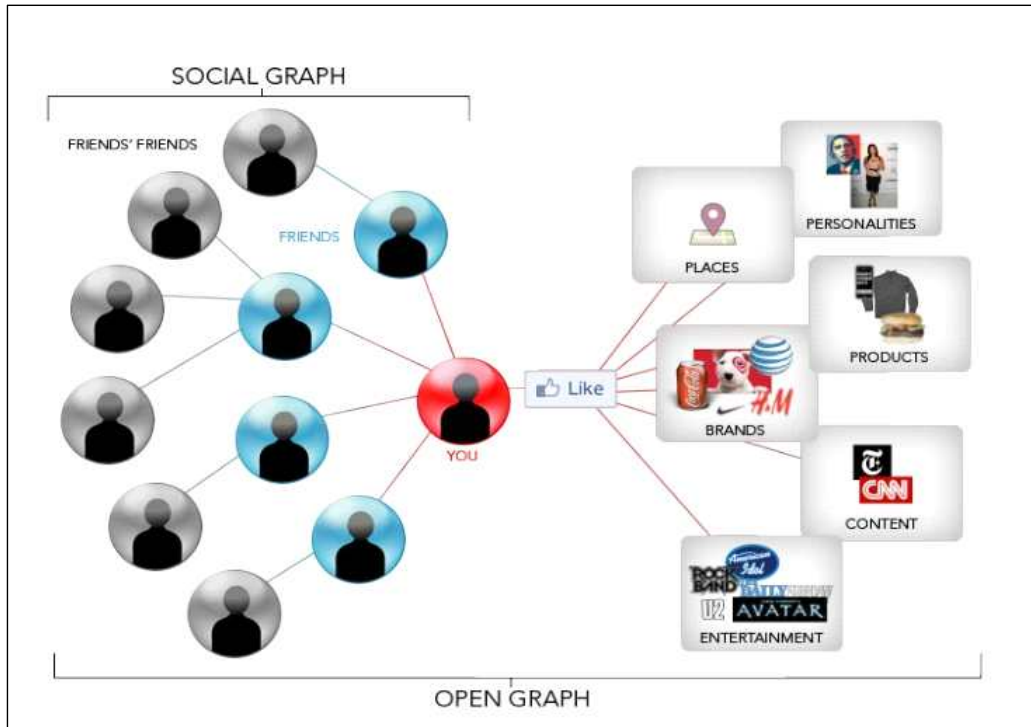


- 그래프 데이터베이스는 기업들이 도입하는 첨단 기술 상당수처럼 우수 기술 기업의 대규모 데이터센터 부문에서 처음으로 관심을 끌기 시작
- 인터넷과 서로 연결된 데이터의 특징 때문에 페이스북과 구글 같은 회사들은 수십 년 동안 일종의 그래프 데이터베이스를 기반으로 사업을 해옴
- 마크 저커버그는 2010년 F8에서 **오픈 그래프 (New Open Graph)** 제시

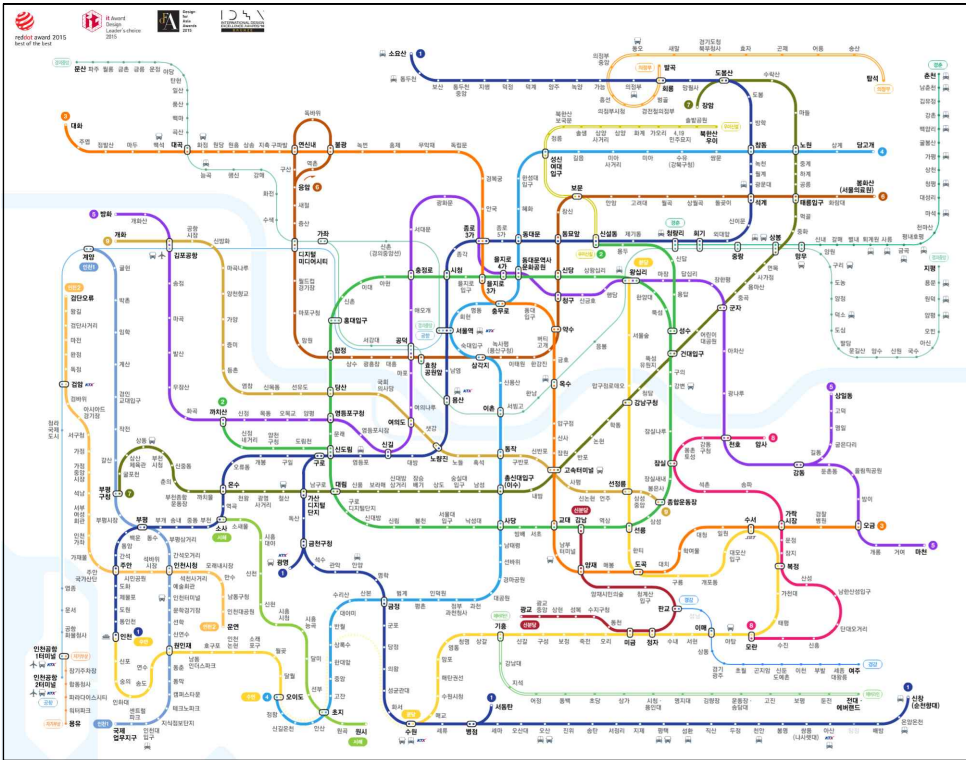
- 페이스북이 사용자의 ‘소셜 그래프(Social Graph)’라고 강조



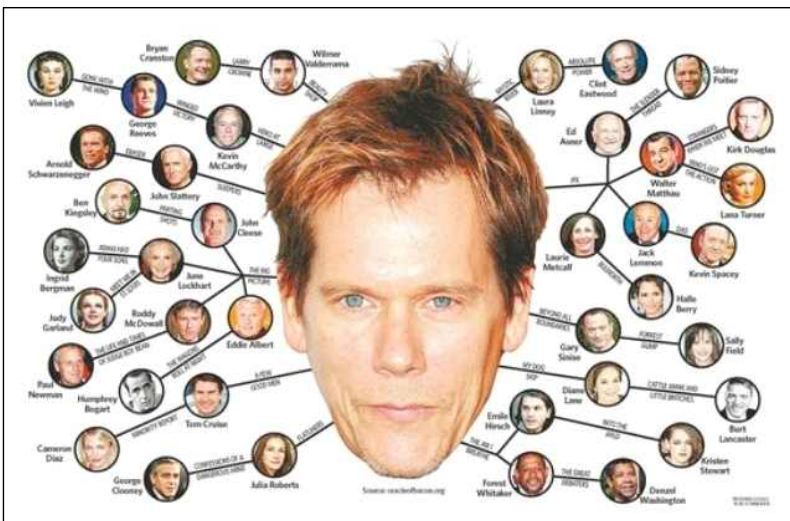
- 오픈 그래프 프로토콜(Open Graph Protocol)을 토대로, 즉 수학 이론을 특정 컴퓨터 과학 애플리케이션으로 바꿔 소셜 네트워크를 구축했기 때문
- 오픈 그래프는 페이스북 커넥트를 통해 연결된 페이스북 앱이 나의 페이스북 개인 정보들을 토대로 서비스를 제공해주는 기능



- 실생활에서 응용되는 대표적인 그래프: 지하철 노선도
- 서울대입구역에서 충무로역까지 가장 빠르게 가는 경로(관계)는?
- 지하철을 타고 서울대입구역에서 충무로역을 갈 때는 두 역사이의 거리, 역의 실제 위치, 방위 등을 생각할 필요가 없이 어느 역(점)들이 어떻게 이어져 있는지(관계)만 고려하면 찾아 갈 수 있음

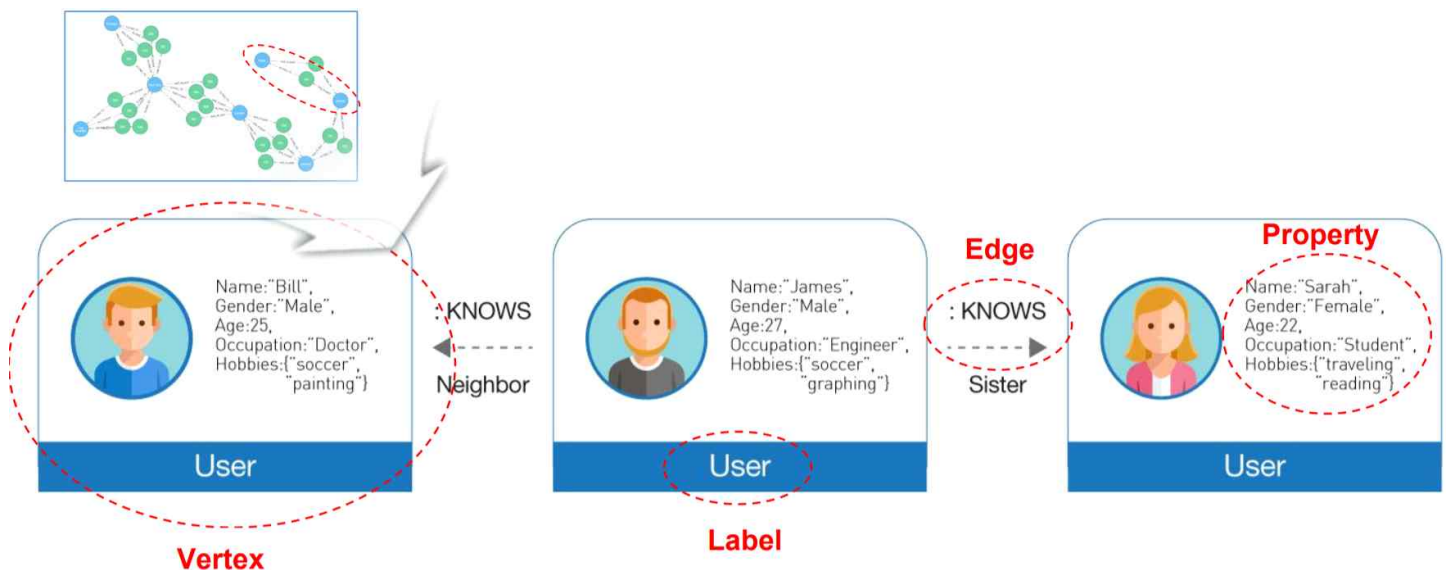


- 케빈 베이컨의 6다리 법칙: 지구에 있는 모든 사람들은 최대 6단계 이내에서 서로 아는 사람으로 연결 될 수 있음



■ 그래프 데이터(Graph Data)

- 모든 종류의 데이터 간의 관계를 점(Vertex/Node)과 선(Edge of Relationship)으로 연결한 데이터 모델
- 그래프(Graph)는 점(Vertex)과 선(Edge)의 연결로 데이터를 저장하여, 특정 대상 간 상관관계를 파악하기 용이 함
- 노드(Node): 그래프의 점(Vertex)으로, 객체를 나타냄
- 관계(Relation): 그래프의 선(Edge)으로, 해당 객체 간의 다양한 관계를 보여줌
- 속성(Property): 노드와 관계의 속성 값



Vertex

+

Edge

=

Graph



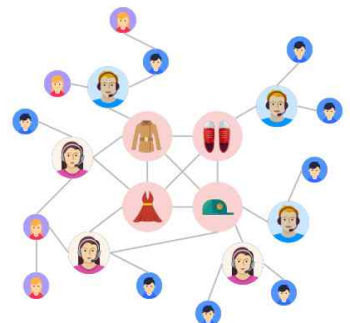
대상 객체를 표현함

ex) 홍길동/21세/남/...
ex) 모자/캠/남성용/...



관계를 표현함

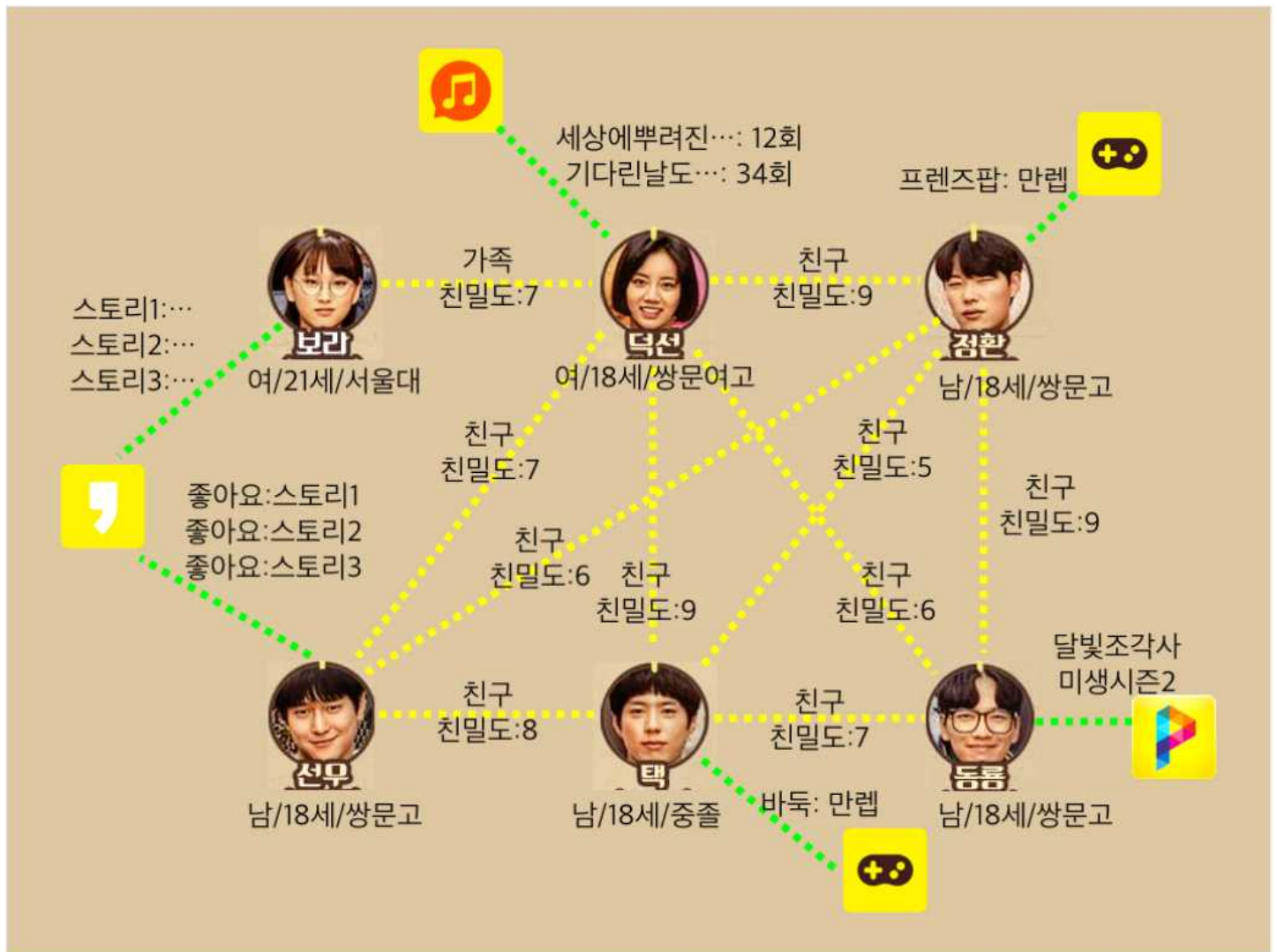
ex) 구입하다, 대행하다, 선물하다



대상 간 인과관계를 표현함

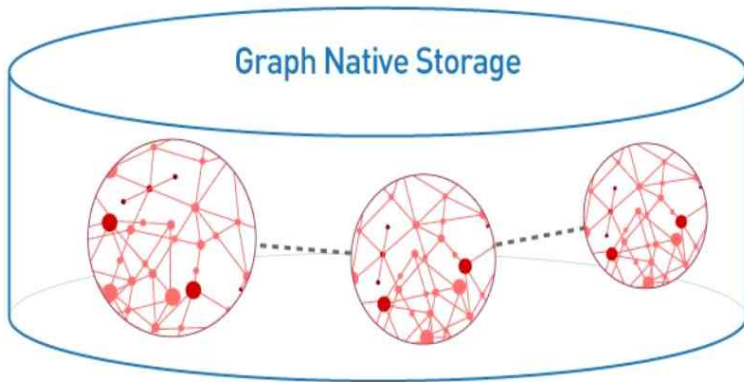
ex) 000쇼핑몰 8월 구매 흐름

- 소셜 그래프의 구성



■ 그래프 데이터베이스 정의 및 특징(Graph Database)

1. 그래프 데이터모델을 기반으로 데이터를 저장, 관리, 분석할 수 있는 기술을 제공하는 데이터베이스



-  **Graph Query Language**
데이터 처리 및 분석을 위한 직관적인 언어
-  **Administration Tools**
시스템 관리, 백업, 모니터링 등과 같은 관리 도구
-  **Data Management**
데이터 캐시, 클러스터 구성, 트랜잭션 관리

2. 종류로 Neo4j, Titan, Giraph 등이 있음

- Neo4j는 자바기반으로 임베딩 방식이나 REST 방식으로 사용 가능
- <https://neo4j.com/sandbox-v2/?ref=product> 에서 회원 가입 후 간단한 테스트가 가능
- 회원 가입 후 Neo4j Browser를 클릭

Your Current Sandboxes

Recommendations


Get Started


Details

Data Model

Code

Advanced ▾





Get Started with your Neo4j Sandbox

- 1 Visit the **Neo4j Browser**. You'll automatically be authenticated. A tutorial will guide you through the datamodel and example data, while teaching you how property graphs work in real-world use cases.
- 2 Start building your application backed by Neo4j. Write your own code, in PHP, Java, JavaScript, Python, or one of any number of other languages, using [templates provided](#).
- 3 Download [Neo4j](#) to your own computer, or start a long-living Neo4j instance in the cloud on [AWS](#) or other [hosting platforms](#).

Launch a New Sandbox

Each sandbox includes data, interactive guides with example queries, and sample code.

- 제공되는 Movie Graph create 소스를 입력 후 실행

```
1 CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
2 CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
3 CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
4 CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
5 CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
6 CREATE (LillyW:Person {name:'Lilly Wachowski', born:1967})
7 CRFATF (LanaW:Person {name:'Lana Wachowski', born:1965})
```

\$:play movie-graph

The Movie Graph

Create

To the right is a giant code block containing a single Cypher query statement composed of multiple CREATE clauses. This will create the movie graph.

1. Click on the code block
2. Notice it gets copied to the editor above ↑
3. Click the editor's play button to execute
4. Wait for the query to finish

WARNING: This adds data to the current database, each time it is run!

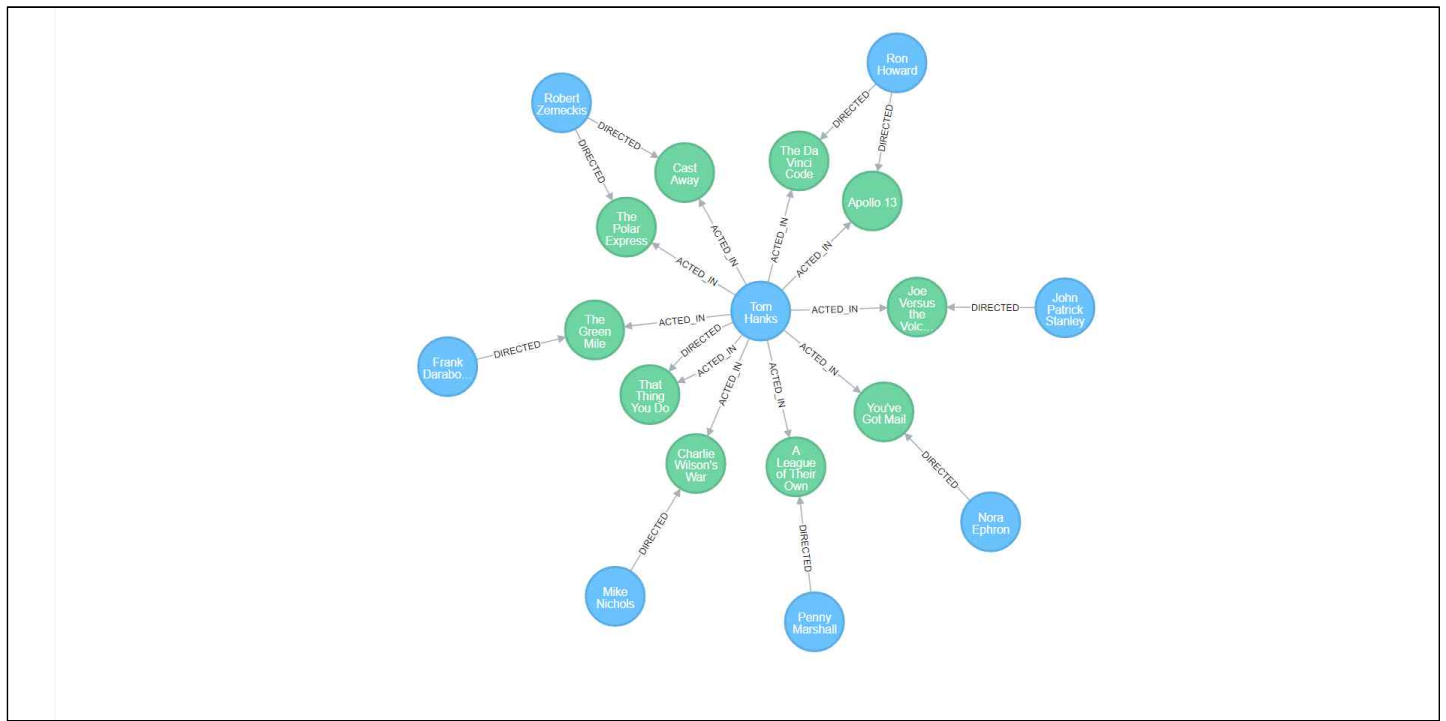
help

cypher

CREATE

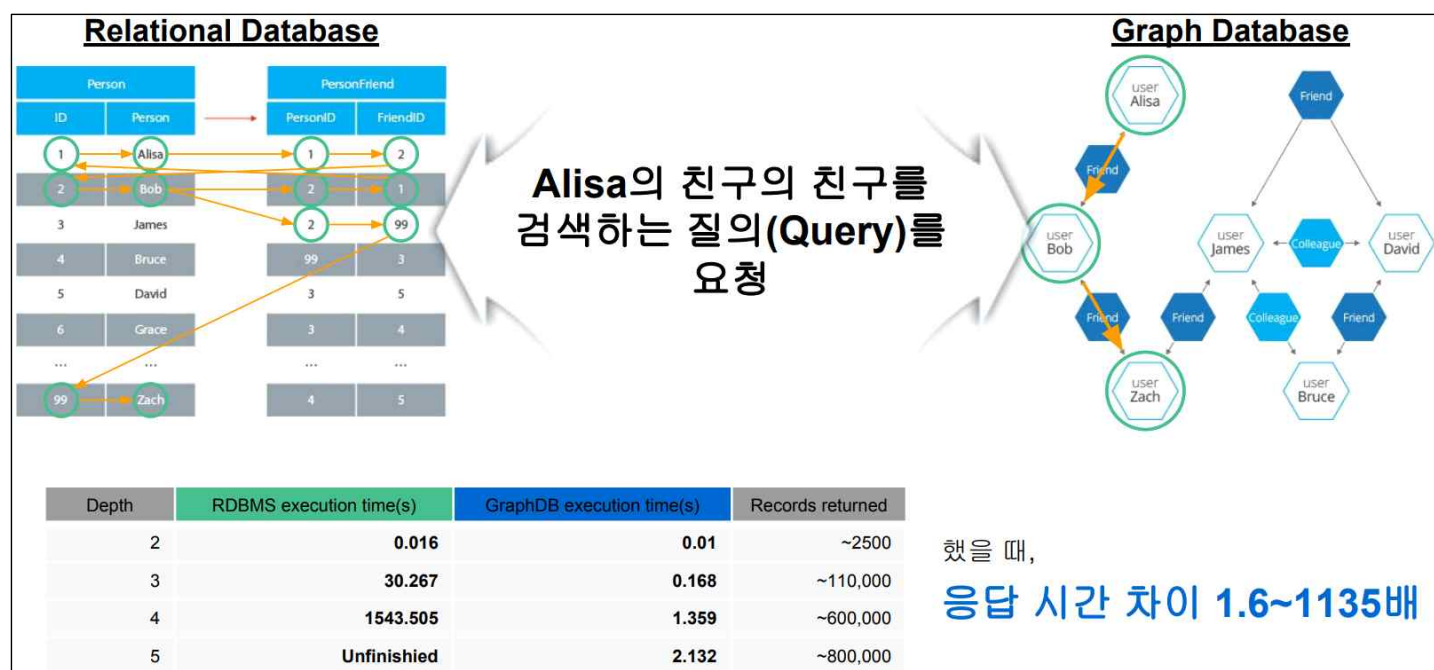
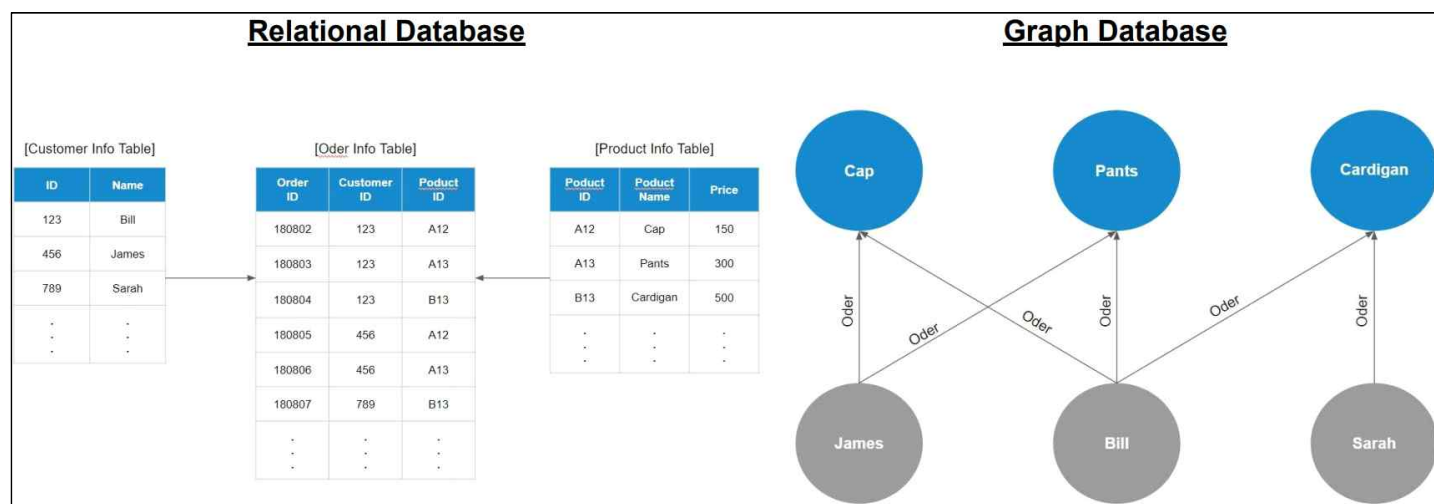
```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
CREATE (LillyW:Person {name:'Lilly Wachowski', born:1967})
CREATE (LanaW:Person {name:'Lana Wachowski', born:1965})
CREATE (JoelS:Person {name:'Joel Silver', born:1952})
CREATE
  (Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix),
  (Carrie)-[:ACTED_IN {roles:['Trinity']}]->(TheMatrix),
```

- 만들어진 Movie Graph를 확인 가능



3. RDB와 GDB 비교

① 데이터 조회시 Join을 위한 테이블 스캔 연산이 없어서 응답속도가 빠름



② 데이터 구조 변경이 자유로운 Schema-less한 데이터 구조로 급변하는 비즈니스 환경에 유연하게 적용 가능

Relational Database

	Col1	Col2	Col3
Row1	111	AAA	!
Row2	222	BBB	@
Row3	333	CCC	#
Row4	444	DDD	\$
Row5	555	EEE	%

↓

	Col1	Col2	Col3	NewCol
Row1	111	AAA	!	Null
Row2	222	BBB	@	Null
Row3	333	CCC	#	Null
Row4	444	DDD	\$	Null
Row5	555	EEE	%	Null
New	666	FFF	^	Value

테이블 형태에 담겨져있는 데이터들이 있을 때, 이에 대한 속성(컬럼)을 추가하는 것은 어려운 업무임

Graph Database

다른 속성을 가진 데이터가 추가 되더라도 별개의 점이기때 상관없으며, 필요 시 데이터들에 대한 관계(선) 설정이 자유로움

③ 시각적인 그래프를 통해 데이터를 확인하므로 직관적으로 문제에 대한 파악 및 insight 도출 가능

Relational Database

고객정보 테이블

ID	Name	Gender	Age
1	Bill	Male	25
2	James	Male	27
3	Sarah	Female	22

주문정보 테이블

Order_ID	Product_ID	User_ID	Quantity	Price
17082101	10235	1	1	164,000
17082102	35432	2	1	19,800
17082103	56821	3	2	22,400

상품정보 테이블

Product_ID	Product_Name	Unit_Price
10235	Electric_Razor	164,000
35432	Lotion_Set	19,800
56821	T_shirt	22,400

상품유형 테이블

Product_ID	Product_Type
10235	Home_Appliances
35432	Beauty
56821	Clothing

Graph Database

Bill

property

- ID: 1
- Name: Bill
- Gender: Male
- Age: 25

구매하다

property

- OrderID: 17082101
- ProductID: 10235
- UserID: 1
- Quantity: 1
- Price: 164,000

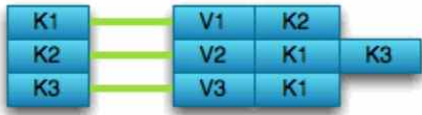
E_Razor

property

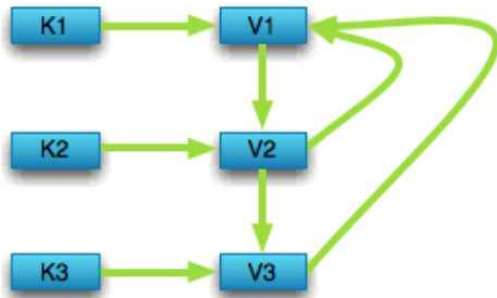
- ProductID: 10235
- ProductName: ElectricRazor
- UnitPrice: 164,000
- ProductType: Home_Appliances

4. Key-Value와 GDB 비교

- Key-Value 모델은 방대한 양의 단순하거나 복잡한 값을 조회하는데 고성능을 발휘함



- 값들이 서로 연결되어 있을 때 그래프를 볼 수 있음
- 연결된 모든 값 사이를 빠르게 이동하고 관계에서 통찰력을 찾을 수 있음



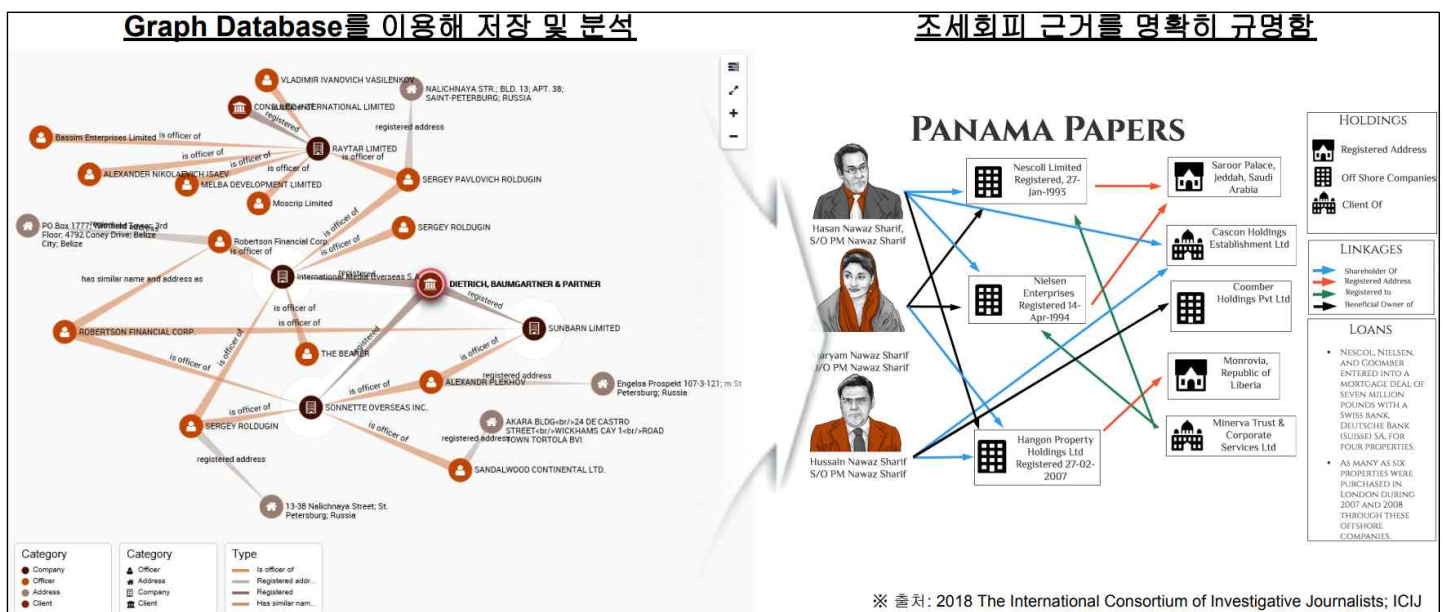
5. 상황 별 필요한 Database 타입

- 복잡한 데이터 연산이 필요한 상황(예: 평균소득계산)은 RDB가 적절
- 유저들의 이동 및 관계를 파악하는 상황(예: 유저가 A지점에서 B지점으로 어떻게 이동했는지 설명)은 GDB가 적절
- 빠른 데이터 연산이 필요한 상황(예: 장바구니 작성)에는 Key-value DB가 적절

■ Graph Database 활용 사례

1. The PANAMA Papers

- 파나마의 최대 로펌 모색 폰세카가 보유하고 있던, 20만개 이상의 조세피난 역외 회사들에 관한, 약 1,150만 건의 기밀문서자료
- 각 국 전·현직 지도자, 정치인, 유명인사들의 조세 회피 관련 정보들이 포함되어 전세계적으로 파장을 일으킨 사건
- ICIJ(The International Consortium of Investigative Journalists)는 해당 사건을 분석하는데, 기존 관계형DB 기반 분석기술로는 수십년이 소요될 것으로 판단 함
- 관계형 DB 기반 분석 기술의 한계:
 - ① 사전 규격화되지 않은 비정형 데이터 처리가 불가능 (이미지, 문서, 영상, 녹취, 메일 등)
 - ② 다양한 상관관계 간 분석이 어려움
- 데이터 유형에 상관없이, 광범위하고 복잡한 관계를 분석할 수 있는 그래프DB를 통해 해결함



2. 글로벌 선도업체들은 기존 관계형DB로 표현할 수 없는 더욱 깊은 관계를 분석하기 위해 그래프DB를 활용하였음

