

今日からはじめるGraph DB

- neo4jでBook Networkを作ってみた -

目次

1. データベースとはどんなものか？
2. RDBMSとGraph DB
3. Book Networkを作ってみた
4. まとめ

1. データベースとはどんなものか？

データベースとは？

データベースとは、**構造化**した情報またはデータの組織的な**集合**であり、通常はコンピューター・システムに電子的に格納されています。

ORACLE (<https://www.oracle.com/jp/database/what-is-database/>)

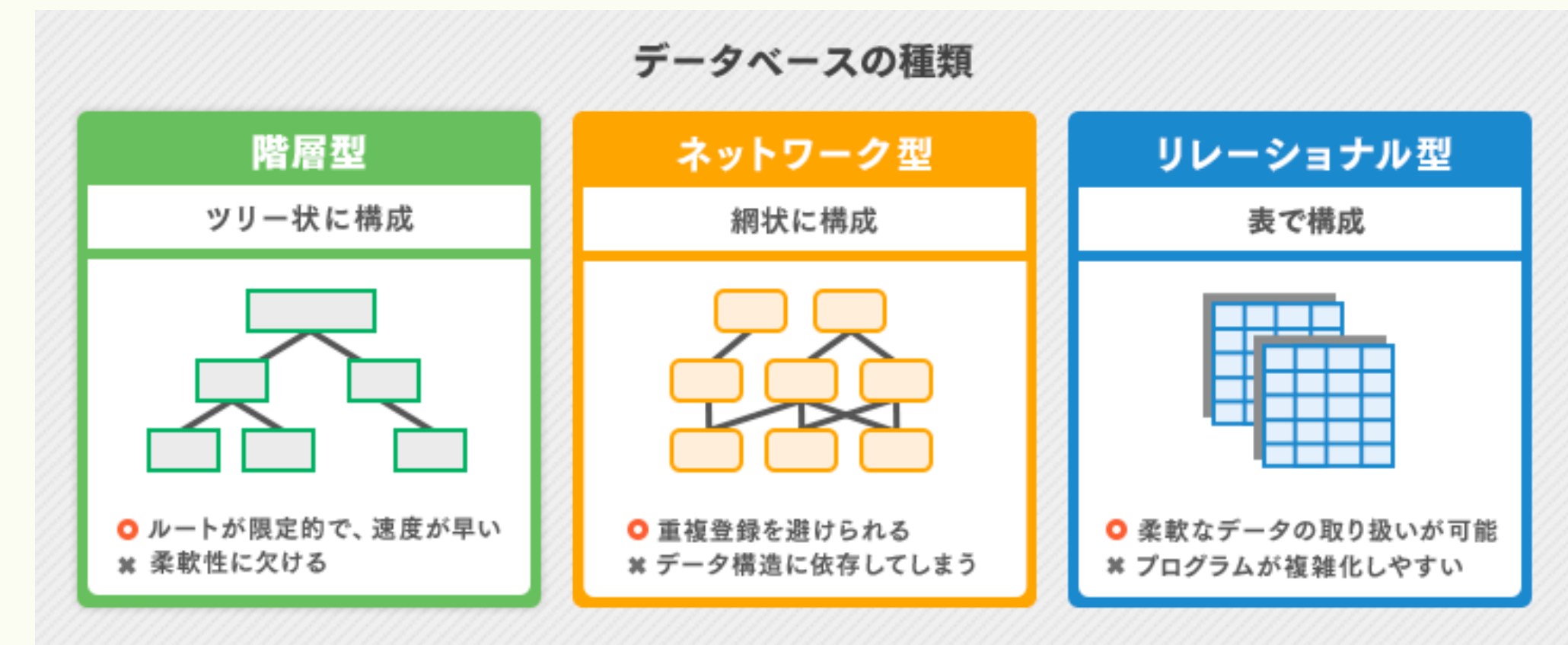
1. データが「構造化」されている

何かしらのルールに則ってデータが格納されている。

例) 階層構造・テーブル構造・グラフ構造

2. データの「集合」である

多数のデータが同じ構造で格納されている。



ITトレンド (<https://it-trend.jp/database/article/89-0066>)

データベースの役割

1. 大量のデータをまとめて管理する

同一構造のデータを一箇所にまとめて管理することで、データを利用しやすくする。

2. データの検索・編集を容易にする

SQL (Structured Query Language) のような専用のクエリ言語を利用することで、データに対して簡単にCRUD操作（登録／検索／更新／削除）を行うことができる。

▶ SQL

```
SELECT      *  
FROM        user_table  
WHERE       age >= 20  
AND         address = 'tokyo';
```

user_table

user_id	user_name	age	email	address
001	AAA	18	aaa@mail.com	tokyo
002	BBB	23	bbb@mail.com	tokyo
003	CCC	26	ccc@mail.com	osaka
004	DDD	32	ddd@mail.com	nagoya
005	EEE	47	eee@mail.com	tokyo

代表的なデータベース製品

1. Oracle Database

Oracle社製のRDBMS（Relational DataBase Management System）。
堅牢性が非常に高く、法人向けデータベースとして高いシェアを誇る。



2. SQLServer

Microsoft社製のRDBMS（Relational DataBase Management System）。
Windows製品との親和性が高く、中規模～大規模法人でよく利用されている。



3. PostgreSQL

OSS（Open Source Software）として提供されている無償のデータベースソフト。
商用のデータベースソフトと比較しても遜色ない機能性を誇る。



2. RDBMSとGraph DB

RDBMSとは？

Relational DataBase Management Systemの略。

リレーショナルデータベースは一件のデータを複数の属性の値の組として表現し、組を列挙することでデータを格納していく方式で、属性を列、組を行とする表（**テーブル**）の形で示されることが多い。

IT用語辞典 (<https://e-words.jp/w/RDBMS.html>)

user_table

user_id	user_name	email
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table

user_id	date	tweet
001	2020/02/12	仕事つかれた。
001	2020/02/12	ねむい。
001	2020/02/10	上司うざい。
002	2020/03/27	今週も一週間おつかれさま～。
002	2020/04/04	明日も頑張ろう！！
003	2020/05/28	クソワロタwww

follow_table

user_id	follow_id
001	002
002	001
002	003
003	002

RDBMSの強み

1. トランザクション制御によるデータの一貫性の保証（ACID特性）
2. 正規化（データの重複排除）による更新コストの削減
3. SQLによる複雑な検索・更新処理のサポート

非正規形の例

user_table

user_id	user_name	email
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table

user_id	user_name	date	tweet
001	AAA	2020/02/12	仕事つかれた。
001	AAA	2020/02/12	ねむい。
001	AAA	2020/02/10	上司うざい。
002	BBB	2020/03/27	今週も一週間おつかれさま～。
002	BBB	2020/04/04	明日も頑張ろう！！
003	CCC	2020/05/28	クソワロタwww

RDBMSのデータ取得方法

キーとなる項目を使ってテーブル間のデータを結合し、条件に合致するデータを取得する。

例) user001がフォローしているユーザーのユーザー名と全ツイートを取得する。

```
SELECT ut.user_name, tt.tweet FROM tweet_table tt
INNER JOIN user_table ut ON tt.user_id = ut.user_id
WHERE tt.user_id IN (SELECT ft.follow_id FROM follow_table ft WHERE ft.user_id = '001');
```

user_table ut

user_id	user_name	email
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table tt

user_id	date	tweet
001	2020/02/12	仕事つかれた。
001	2020/02/12	ねむい。
001	2020/02/10	上司うざい。
002	2020/03/27	今週も一週間おつかれさま～。
002	2020/04/04	明日も頑張ろう！！
003	2020/05/28	クソワロタwww

follow_table ft

user_id	follow_id
001	002
002	001
002	003
003	002

RDBMSのデータ取得方法

キーとなる項目を使ってテーブル間のデータを結合し、条件に合致するデータを取得する。

例) user001がフォローしているユーザーのユーザー名と全ツイートを取得する。

```
SELECT ut.user_name, tt.tweet FROM tweet_table tt
INNER JOIN user_table ut ON tt.user_id = ut.user_id
WHERE tt.user_id IN (SELECT ft.follow_id FROM follow_table ft WHERE ft.user_id = '001');
```

user_table ut

user_id	user_name	email
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table tt

user_id	date	tweet
001	2020/02/12	仕事つかれた。
001	2020/02/12	ねむい。
001	2020/02/10	上司うざい。
002	2020/03/27	今週も一週間おつかれさま～。
002	2020/04/04	明日も頑張ろう！！
003	2020/05/28	クソワロタwww

follow_table ft

user_id	follow_id
001	002
002	001
002	003
003	002

RDBMSのデータ取得方法

キーとなる項目を使ってテーブル間のデータを結合し、条件に合致するデータを取得する。

例) user001がフォローしているユーザーのユーザー名と全ツイートを取得する。

```
SELECT ut.user_name, tt.tweet FROM tweet_table tt
INNER JOIN user_table ut ON tt.user_id = ut.user_id
WHERE tt.user_id IN (SELECT ft.follow_id FROM follow_table ft WHERE ft.user_id = '001');
```

user_table ut

user_id	user_name	email
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table tt

user_id	date	tweet
001	2020/02/12	仕事つかれた。
001	2020/02/12	ねむい。
001	2020/02/10	上司うざい。
002	2020/03/27	今週も一週間おつかれさま～。
002	2020/04/04	明日も頑張ろう！！
003	2020/05/28	クソワロタwww

follow_table ft

user_id	follow_id
001	002
002	001
002	003
003	002

RDBMSのデータ取得方法

キーとなる項目を使ってテーブル間のデータを結合し、条件に合致するデータを取得する。

例) user001がフォローしているユーザーのユーザー名と全ツイートを取得する。

```
SELECT ut.user_name, tt.tweet FROM tweet_table tt
INNER JOIN user_table ut ON tt.user_id = ut.user_id
WHERE tt.user_id IN (SELECT ft.follow_id FROM follow_table ft WHERE ft.user_id = '001');
```

user_table ut

user_id	user_name	email
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
001	AAA	aaa@mail.com
002	BBB	bbb@mail.com
002	BBB	bbb@mail.com
003	CCC	ccc@mail.com

tweet_table tt

user_id	date	tweet
001	2020/02/12	仕事つかれた。
001	2020/02/12	ねむい。
001	2020/02/10	上司うざい。
002	2020/03/27	今週も一週間おつかれさま～。
002	2020/04/04	明日も頑張ろう！！
003	2020/05/28	クソワロタwww

follow_table ft

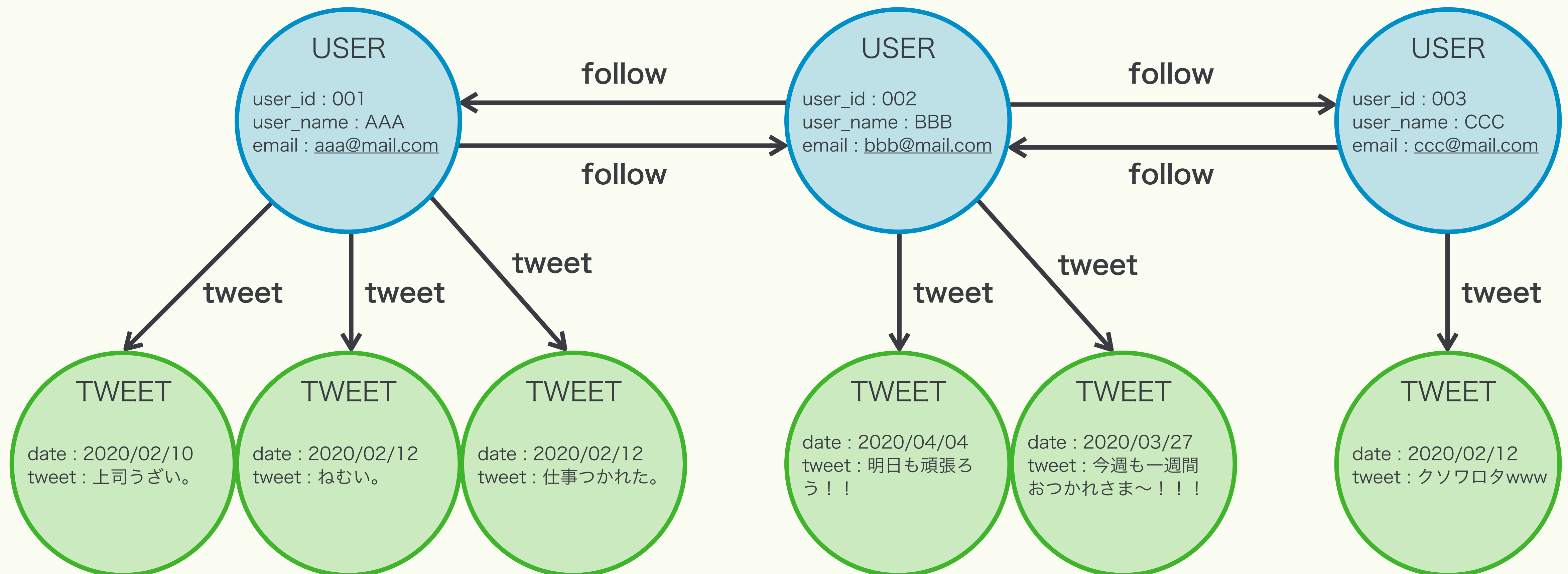
user_id	follow_id
001	002
002	001
002	003
003	002

Graph DBとは？

グラフDBとは一言で言うと、グラフ構造を備えたデータベースのことである。（中略）

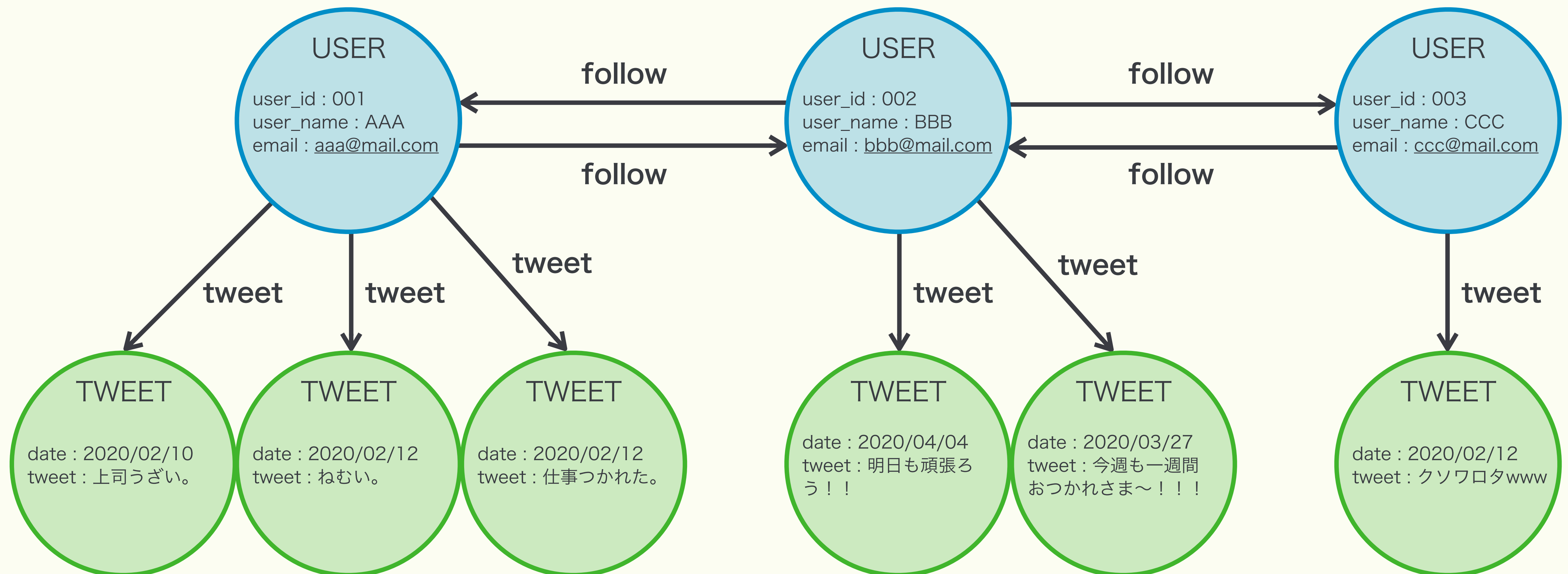
グラフは「**ノード**」「**エッジ**」「**プロパティ**」の3要素によって、ノード間の「関係性」を表現できる。

iMagazine (<https://www.imagazine.co.jp/12805-2/>)



Graph DBの強み

1. データの構造が視覚的にわかりやすい
2. つながりを持つデータの検索が高速

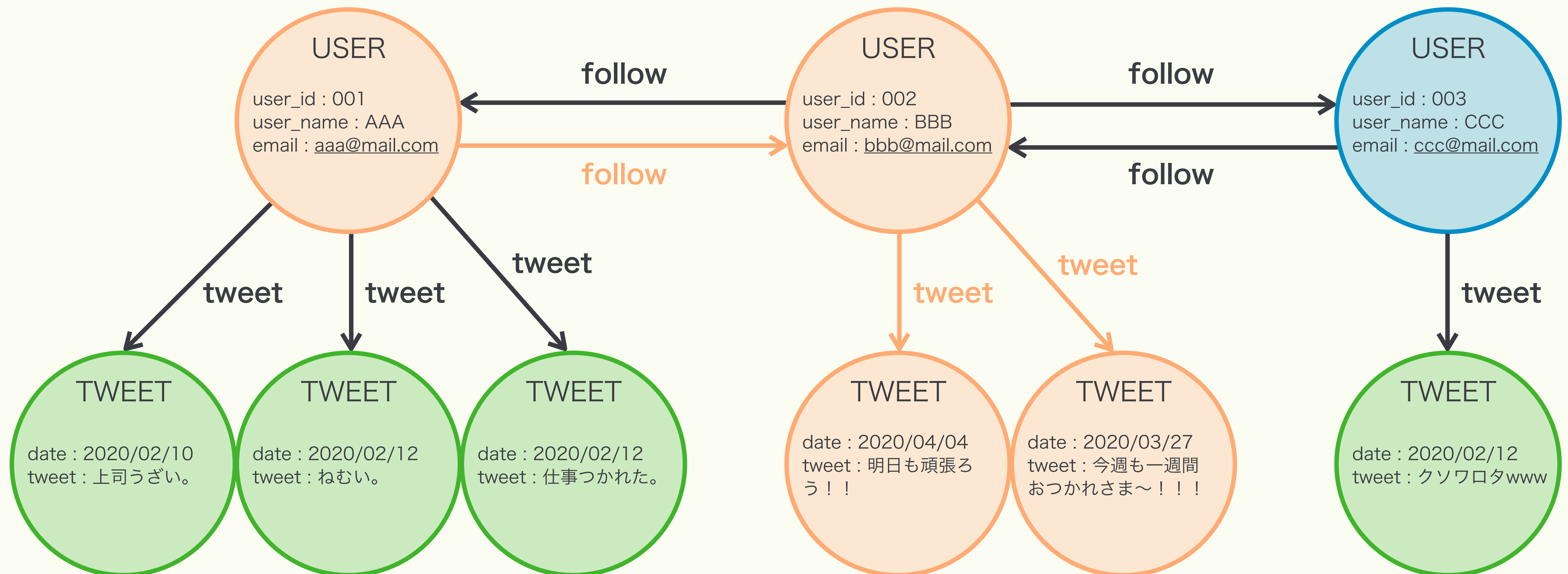


Graph DBのデータ取得方法

データ間のつながりを辿ることで、条件に合致するデータを取得する。

例) user001がフォローしているユーザーのユーザー名と全ツイートを取得する。

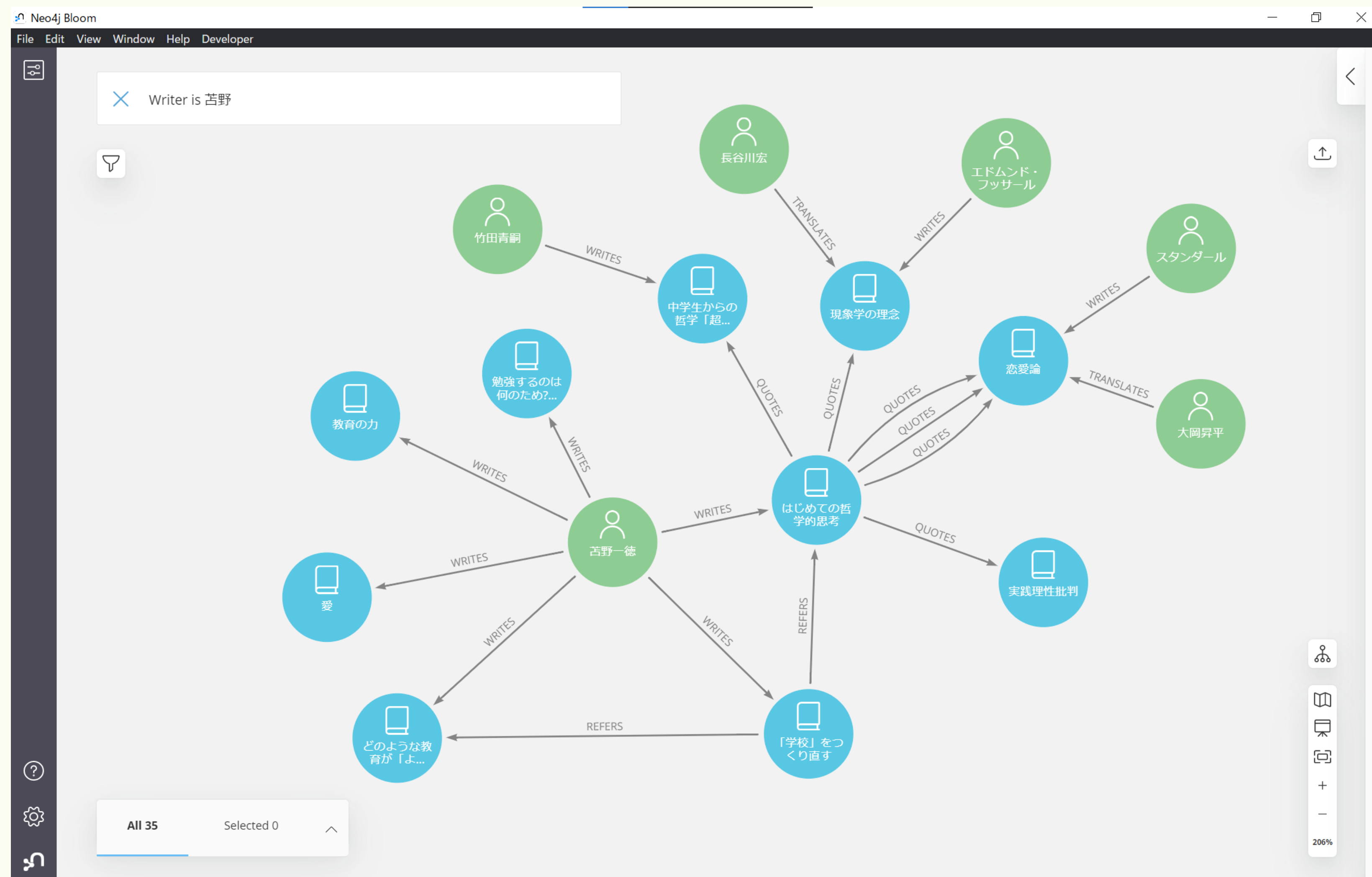
```
MATCH (u1:USER{user_id:001})-[f:follow]->(u2:USER)-[t1:tweet]->(t2:TWEET) RETURN u2, t1, t2
```



3. Book Networkを作ってみた

何を作ったか

ある本の著者・翻訳者および本同士の参照・引用関係を可視化するGraph DBを作ってみた。



使ったデータベース

▶ neo4j

Neo Technology社によって開発されているOSSのGraph DB。
Cypher Query Languageという専用のクエリ言語を使うことで、
グラフ構造のデータを簡単に検索することができる。



▶ Cypher Query Language

ノードとリレーションの関係を直感的に記述してデータを検索することができる。

MATCH (p:Person) - [w:WRITES] -> (b:Book) **RETURN** p,w,b

人

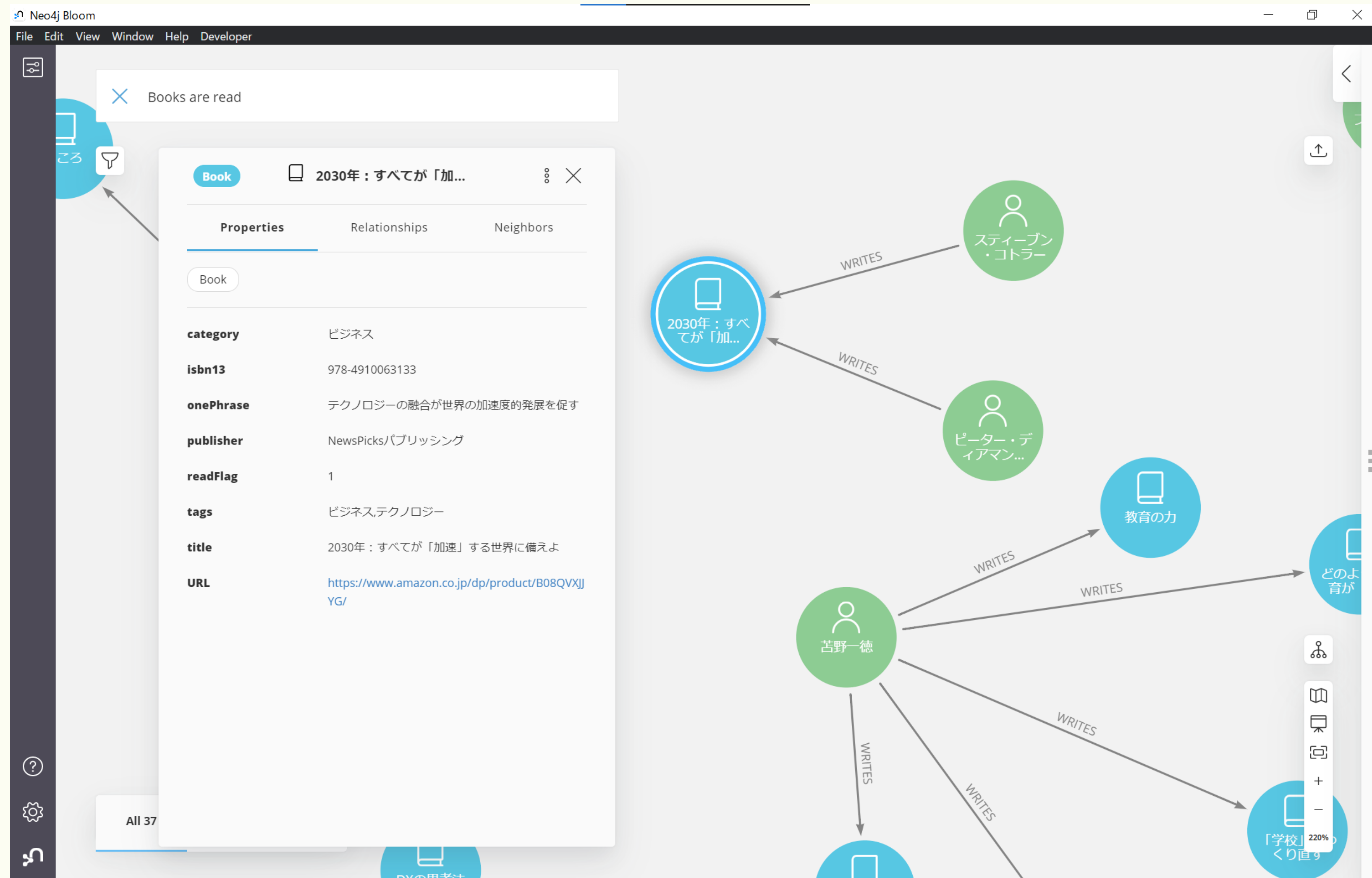
書いた

本

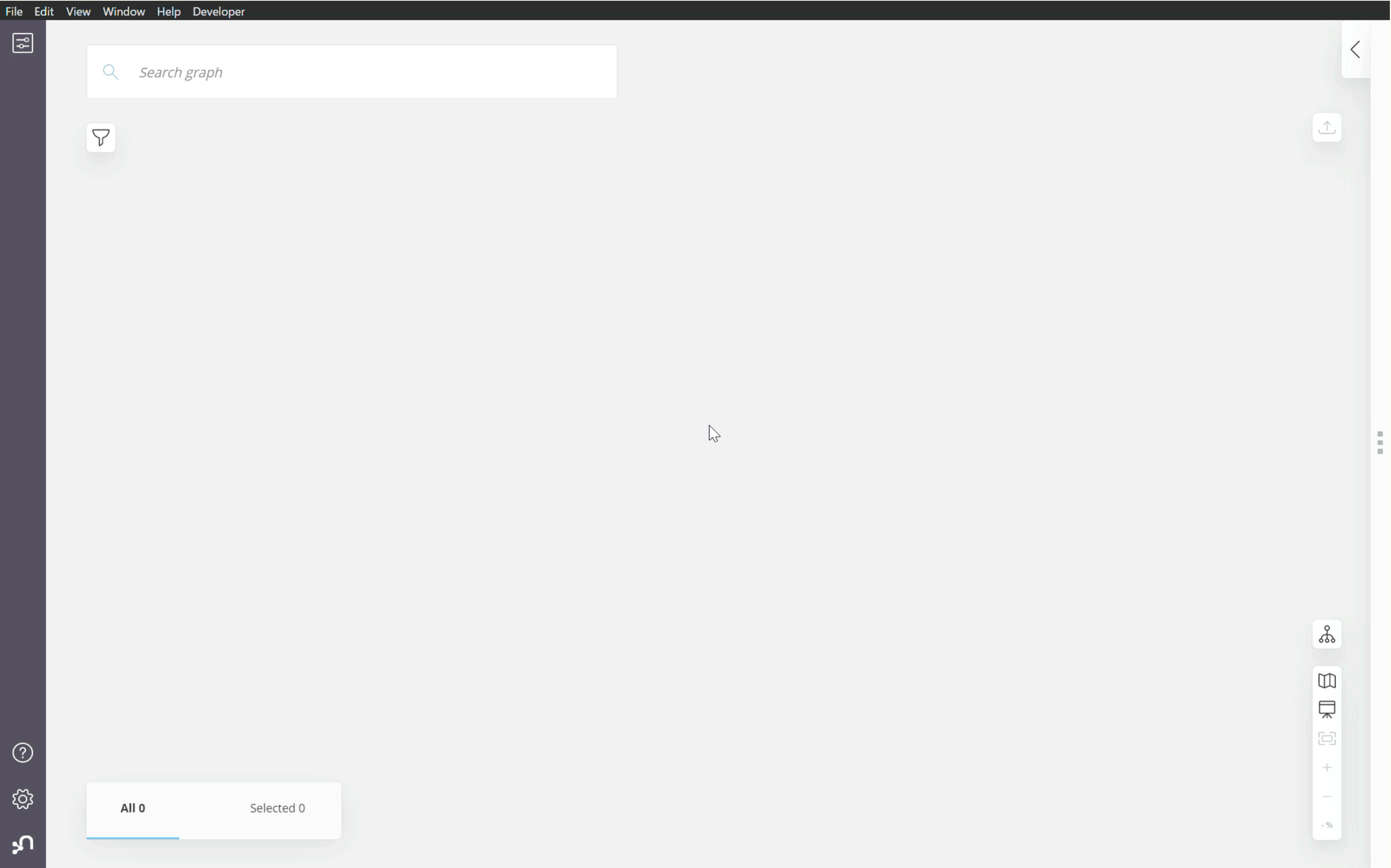
p・w・bの関係を返す

できること

著者検索・タイトル検索・タグ検索・カテゴリ検索と、本および引用内容の詳細表示。



Demo



4. まとめ

まとめ

▶ データベースとは？

構造化した情報またはデータの組織的な集合のこと。

▶ Graph DBとは？

グラフ構造を備えたデータベースのこと。

▶ Book Networkを作ってみた

ある本の著者・翻訳者および本同士の参照・引用関係を可視化するGraph DBを作ってみた。
データをすべて手作業で登録しなければならないところは改善の余地アリ。

Graph DBを導入して遊んでみてね！！！！