

**Part 1: Relational Databases – Use ORACLE**

A. Access Oracle Live SQL to complete Part 1.

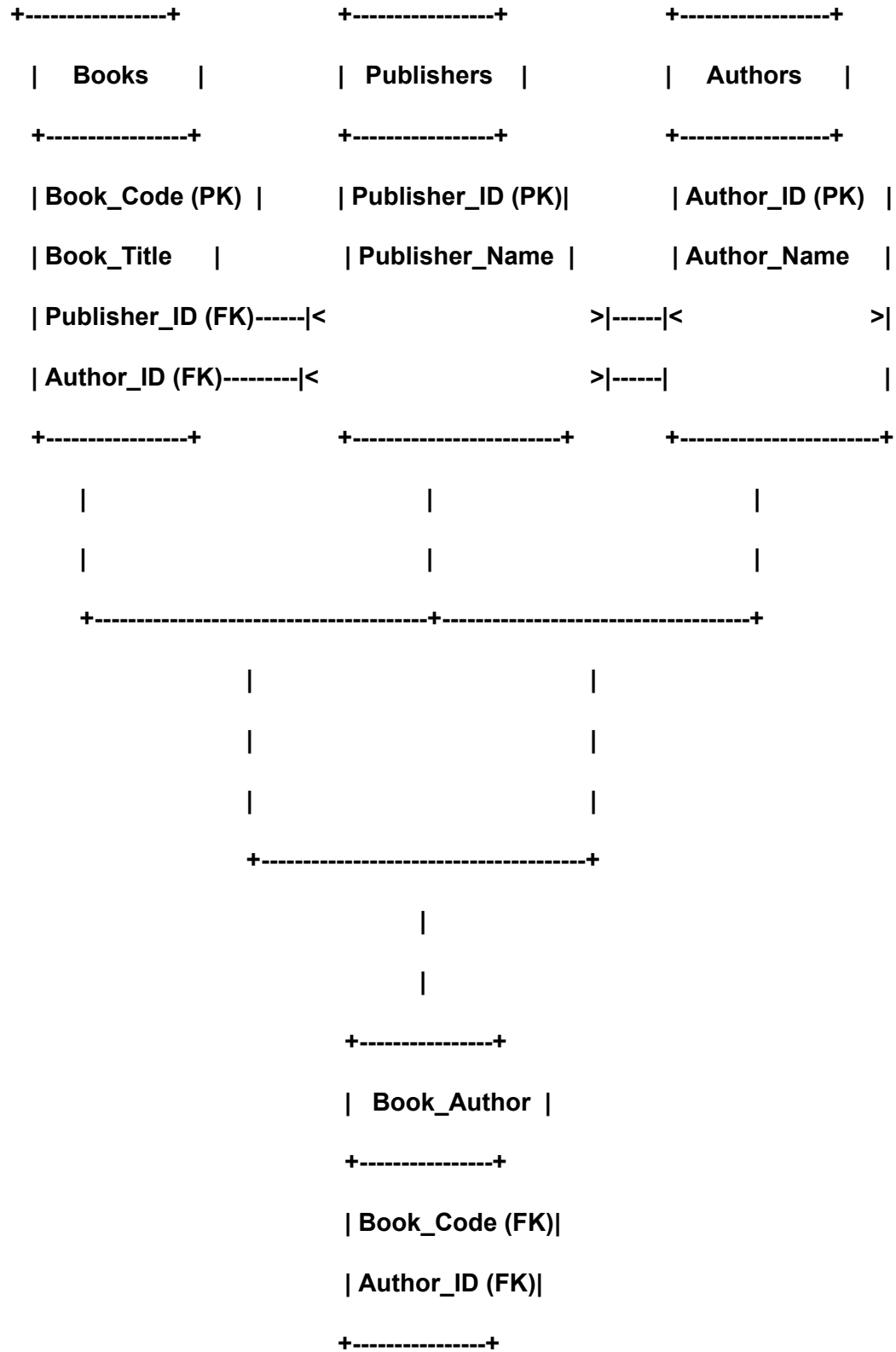
<https://livesql.oracle.com/apex/f?p=590:1000>

B. Normalize the data in Table 1 attached – show the EER diagram for your final results

**Table 1**

Book_Code	Book_Title	Publisher	Author
22	Stranger	Vintage	Camus
13	Dreamcatcher	Scribner	King
18	Beloved	Plume	Morrison
37	Nine	LB Books	Salinger
57	Catch 22	Scribner	Heller
61	Jazz	Plume	Morrison
69	Franny	LB Books	Salinger
75	Fall	Vintage	Camus
96	Grapes	Penguin	Morrison
98	Catcher	LB Books	Salinger

## EER Diagram



C. For each of the following steps, include screenshots of your code and results

1. Use SQL code to create the tables you identified in the EER diagram
2. Use SQL code to show that each of the tables has been created

#### SQL Worksheet

```
1 v CREATE TABLE Publishers (  
2     Publisher_ID INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
3     Publisher_Name VARCHAR2(30)  
4 );  
5
```

Table created.

```
6 v CREATE TABLE Authors (  
7     Author_ID INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
8     Author_Name VARCHAR2(30)  
9 );  
10
```

Table created.

```
11 v CREATE TABLE Books (  
12     Book_Code INT PRIMARY KEY,  
13     Book_Title VARCHAR2(30),  
14     Publisher_ID INT,  
15     Author_ID INT,  
16     FOREIGN KEY (Publisher_ID) REFERENCES Publishers(Publisher_ID),  
17     FOREIGN KEY (Author_ID) REFERENCES Authors(Author_ID)  
18 );  
19
```

Table created.

```

20 v CREATE TABLE Book_Author (
21     Book_Code INT,
22     Author_ID INT,
23     PRIMARY KEY (Book_Code, Author_ID),
24     FOREIGN KEY (Book_Code) REFERENCES Books(Book_Code),
25     FOREIGN KEY (Author_ID) REFERENCES Authors(Author_ID)
26 );
27

```

Table created.

3. Use SQL code to enter the information displayed in Table 1 into the tables you created

```

36 INSERT INTO Publishers (Publisher_Name) VALUES ('Vintage');
37 INSERT INTO Publishers (Publisher_Name) VALUES ('Scribner');
38 INSERT INTO Publishers (Publisher_Name) VALUES ('Plume');
39 INSERT INTO Publishers (Publisher_Name) VALUES ('LB Books');
40 INSERT INTO Publishers (Publisher_Name) VALUES ('Penguin');
41

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```
43 INSERT INTO Authors (Author_Name) VALUES ('Camus');
44 INSERT INTO Authors (Author_Name) VALUES ('King');
45 INSERT INTO Authors (Author_Name) VALUES ('Morrison');
46 INSERT INTO Authors (Author_Name) VALUES ('Salinger');
47 INSERT INTO Authors (Author_Name) VALUES ('Heller');
48
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```
49 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (22, 'Stranger', 1, 1);
50 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (13, 'Dreamcatcher', 2, 2);
51 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (18, 'Beloved', 3, 3);
52 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (37, 'Nine', 4, 4);
53 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (57, 'Catch 22', 2, 5);
54 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (61, 'Jazz', 3, 3);
55 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (69, 'Franny', 4, 4);
56 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (75, 'Fall', 1, 1);
57 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (96, 'Grapes', 5, 3);
58 INSERT INTO Books (Book_Code, Book_Title, Publisher_ID, Author_ID) VALUES (98, 'Catcher', 4, 4);
59
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```
60 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (22, 1);
61 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (13, 2);
62 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (18, 3);
63 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (37, 4);
64 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (57, 5);
65 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (61, 3);
66 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (69, 4);
67 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (75, 1);
68 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (96, 3);
69 INSERT INTO Book_Author (Book_Code, Author_ID) VALUES (98, 4);
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

4. Use SQL code to show that the information has been entered into the appropriate tables

62 `SELECT * FROM Publishers;`

PUBLISHER_ID	PUBLISHER_NAME
1	Vintage
2	Scribner
3	Plume
4	LB Books
5	Penguin

Download CSV

5 rows selected.

63 `SELECT * FROM Authors;`

AUTHOR_ID	AUTHOR_NAME
1	Camus
2	King
3	Morrison
4	Salinger
5	Heller

Download CSV

5 rows selected.

64 `SELECT * FROM Books;`

BOOK_CODE	BOOK_TITLE	PUBLISHER_ID	AUTHOR_ID
22	Stranger	1	1
13	Dreamcatcher	2	2
18	Beloved	3	3
37	Nine	4	4
57	Catch 22	2	5
61	Jazz	3	3
69	Franny	4	4
75	Fall	1	1
96	Grapes	5	3
98	Catcher	4	4

Download CSV

10 rows selected.



65 | `SELECT * FROM Book_Author;`

BOOK_CODE	AUTHOR_ID
13	2
18	3
22	1
37	4
57	5
61	3
69	4
75	1
96	3
98	4

Download CSV

10 rows selected.

5. Run the following queries:
  - a. For each author name, show the book title. Your output should include author name and book title. Order the output by author name in descending order

```

68 v SELECT Authors.Author_Name, Books.Book_Title
69 FROM Books
70 JOIN Book_Author ON Books.Book_Code = Book_Author.Book_Code
71 JOIN Authors ON Book_Author.Author_ID = Authors.Author_ID
72 ORDER BY Authors.Author_Name DESC;

```

AUTHOR_NAME	BOOK_TITLE
Salinger	Nine
Salinger	Catcher
Salinger	Franny
Morrison	Grapes
Morrison	Beloved
Morrison	Jazz
King	Dreamcatcher
Heller	Catch 22
Camus	Stranger
Camus	Fall

Download CSV

10 rows selected.

b. For each publisher, show the books they have published. Your output should include publisher name and book title. Order the output by publisher name in ascending order.

```
75 v SELECT Publishers.Publisher_Name, Books.Book_Title
76 FROM Books
77 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
78 ORDER BY Publishers.Publisher_Name ASC;
79
```

PUBLISHER_NAME	BOOK_TITLE
LB Books	Franny
LB Books	Nine
LB Books	Catcher
Penguin	Grapes
Plume	Beloved
Plume	Jazz
Scribner	Dreamcatcher
Scribner	Catch 22
Vintage	Stranger
Vintage	Fall

Download CSV

10 rows selected.

- c. For each author, show the publisher who has published his work. Your output should include the author's name and the publisher's name. Order the output by author's name in ascending order.

```
80 v SELECT Authors.Author_Name, Publishers.Publisher_Name
81 FROM Authors
82 JOIN Book_Author ON Authors.Author_ID = Book_Author.Author_ID
83 JOIN Books ON Book_Author.Book_Code = Books.Book_Code
84 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
85 ORDER BY Authors.Author_Name ASC;
```

AUTHOR_NAME	PUBLISHER_NAME
Camus	Vintage
Camus	Vintage
Heller	Scribner
King	Scribner
Morrison	Plume
Morrison	Penguin
Morrison	Plume
Salinger	LB Books
Salinger	LB Books
Salinger	LB Books

Download CSV

10 rows selected.

- d. List the title of all the books which Salinger has written.

```
87 v SELECT Books.Book_Title
88 FROM Books
89 JOIN Book_Author ON Books.Book_Code = Book_Author.Book_Code
90 JOIN Authors ON Book_Author.Author_ID = Authors.Author_ID
91 WHERE Authors.Author_Name = 'Salinger';
```

BOOK_TITLE
Nine
Franny
Catcher

Download CSV

3 rows selected.

- e. List the title of all the books published by publisher Vintage

```
93 v SELECT Books.Book_Title
94 FROM Books
95 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
96 WHERE Publishers.Publisher_Name = 'Vintage';
```

BOOK_TITLE
Stranger
Fall

Download CSV

2 rows selected.

f. List the title of all the books and the publisher's name for all books published by Vintage, LB Books or Plume (use IN)

```
98 v SELECT Books.Book_Title, Publishers.Publisher_Name
99 FROM Books
100 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
101 WHERE Publishers.Publisher_Name IN ('Vintage', 'LB Books', 'Plume');
```

BOOK_TITLE	PUBLISHER_NAME
Stranger	Vintage
Beloved	Plume
Nine	LB Books
Jazz	Plume
Franny	LB Books
Fall	Vintage
Catcher	LB Books

Download CSV

7 rows selected.

- g. List the title of the books and the publisher name for all books published by Scribner or Plume (use OR) \*\*\*NOTE: you should only have four lines of output

```
103 v SELECT Books.Book_Title, Publishers.Publisher_Name
104 FROM Books
105 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
106 WHERE Publishers.Publisher_Name = 'Scribner' OR Publishers.Publisher_Name = 'Plume';
```

BOOK_TITLE	PUBLISHER_NAME
Dreamcatcher	Scribner
Beloved	Plume
Catch 22	Scribner
Jazz	Plume

Download CSV

4 rows selected.

- h. List the title of the book and the publisher name for all books published by Penguin and written by Morrison (use AND)

```
108 v SELECT Books.Book_Title, Publishers.Publisher_Name
109 FROM Books
110 JOIN Publishers ON Books.Publisher_ID = Publishers.Publisher_ID
111 JOIN Book_Author ON Books.Book_Code = Book_Author.Book_Code
112 JOIN Authors ON Book_Author.Author_ID = Authors.Author_ID
113 WHERE Publishers.Publisher_Name = 'Penguin' AND Authors.Author_Name = 'Morrison';
```

BOOK_TITLE	PUBLISHER_NAME
Grapes	Penguin

Download CSV

## Part 2: Document Stores – Use MongoDB

### A. INSTRUCTIONS TO ACCESS MONGODB:

1. To Download MongoDB Shell for Windows, Mac or Linux follow the instructions found on the link below
  - a. Windows: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-onwindows/>
  - b. MAC OS: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-os-x/>
  - c. LINUX: <https://www.mongodb.com/docs/manual/administration/install-on-linux/>
2. Link for using a MongoDB terminal online without any downloads
  - a. No SAVE option: <https://www.jdoodle.com/online-mongodb-terminal>
  - b. SAVE option: <https://www.mycompiler.io/online-mongodb-editor>
  - c. DOWNLOAD CODE option: <https://onecompiler.com/mongodb/3xnw4yay2>
3. MongoDB documentation:
  - a. Link for MongoDB manual starting: <https://www.mongodb.com/docs/manual/introduction/>
  - b. Link for MongoDB QUERY Section: <https://docs.mongodb.com/manual/tutorial/querydocuments/>
  - c. Download MongoDB Shell on your computer: <https://www.mongodb.com/try/download/shell>

### B. INSTRUCTIONS FOR MONGODB PART 2 OF PROJECT

- Create the data in Table 1 below in MongoDB
- For each of the following steps, include screenshots of your code and results



- Use MongoDB to enter the information displayed in table 1 into the collection (s) you have created

```
>_MONGOOSH
> use books
< switched to db books
> db.books.insertMany([[{book_code: 22, book_title: "Stranger", publisher: "Vintage", author: "Camus"},
{book_code: 13, book_title: "Dreamcatcher", publisher: "Scribner", author: "King"}, {book_code: 18, book_title: "Beloved", publisher: "Plume", author: "Morrison"},
{book_code: 37, book_title: "Nine", publisher: "LB Books", author: "Salinger"}, {book_code: 57, book_title: "Catch 22", publisher: "Scribner", author: "Heller"},
{book_code: 61, book_title: "Jazz", publisher: "Plume", author: "Morrison"}, {book_code: 69, book_title: "Franny", publisher: "LB Books", author: "Salinger"},
{book_code: 75, book_title: "Fall", publisher: "Vintage", author: "Camus"}, {book_code: 96, book_title: "Grapes", publisher: "Penguin", author: "Morrison"},
{book_code: 98, book_title: "Catcher", publisher: "LB Books", author: "Salinger"}]])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('663ec31417e333091927fc38'),
    '1': ObjectId('663ec31417e333091927fc39'),
    '2': ObjectId('663ec31417e333091927fc3a'),
    '3': ObjectId('663ec31417e333091927fc3b'),
    '4': ObjectId('663ec31417e333091927fc3c'),
    '5': ObjectId('663ec31417e333091927fc3d'),
    '6': ObjectId('663ec31417e333091927fc3e'),
    '7': ObjectId('663ec31417e333091927fc3f'),
    '8': ObjectId('663ec31417e333091927fc40'),
    '9': ObjectId('663ec31417e333091927fc41')
  }
}
books>
```

- Use MongoDB to show that the information has been entered into the collection(s)

```
> db.books.find();
< {
  _id: ObjectId('663ec31417e333091927fc38'),
  book_code: 22,
  book_title: 'Stranger',
  publisher: 'Vintage',
  author: 'Camus'
}
{
  _id: ObjectId('663ec31417e333091927fc39'),
  book_code: 13,
  book_title: 'Dreamcatcher',
  publisher: 'Scribner',
  author: 'King'
}
{
  _id: ObjectId('663ec31417e333091927fc3a'),
  book_code: 18,
  book_title: 'BeLoved',
  publisher: 'Plume',
  author: 'Morrison'
}
{
  _id: ObjectId('663ec31417e333091927fc3b'),
  book_code: 37,
  book_title: 'Nine',
  publisher: 'LB Books',
  author: 'Salinger'
}
```

```
{
  _id: ObjectId('663ec31417e333091927fc3c'),
  book_code: 57,
  book_title: 'Catch 22',
  publisher: 'Scribner',
  author: 'Heller'
}
{
  _id: ObjectId('663ec31417e333091927fc3d'),
  book_code: 61,
  book_title: 'Jazz',
  publisher: 'Plume',
  author: 'Morrison'
}
{
  _id: ObjectId('663ec31417e333091927fc3e'),
  book_code: 69,
  book_title: 'Franny',
  publisher: 'LB Books',
  author: 'Salinger'
}
{
  _id: ObjectId('663ec31417e333091927fc3f'),
  book_code: 75,
  book_title: 'Fall',
  publisher: 'Vintage',
  author: 'Camus'
}
```

```

{
  _id: ObjectId('663ec31417e333091927fc40'),
  book_code: 96,
  book_title: 'Grapes',
  publisher: 'Penguin',
  author: 'Morrison'
}
{
  _id: ObjectId('663ec31417e333091927fc41'),
  book_code: 98,
  book_title: 'Catcher',
  publisher: 'LB Books',
  author: 'Salinger'
}
books >

```

- Use MongoDB to run the following queries:

- a. Show all books written by author Salinger

```

> db.books.find({author:{ $regex: /Salinger/}});
< {
  _id: ObjectId('663ec31417e333091927fc3b'),
  book_code: 37,
  book_title: 'Nine',
  publisher: 'LB Books',
  author: 'Salinger'
}
{
  _id: ObjectId('663ec31417e333091927fc3e'),
  book_code: 69,
  book_title: 'Franny',
  publisher: 'LB Books',
  author: 'Salinger'
}
{
  _id: ObjectId('663ec31417e333091927fc41'),
  book_code: 98,
  book_title: 'Catcher',
  publisher: 'LB Books',
  author: 'Salinger'
}
books > |

```

- b. Show all books published by Vintage

```
> db.books.find({publisher:{ $regex: /Vintage/}});
< {
  _id: ObjectId('663ec31417e333091927fc38'),
  book_code: 22,
  book_title: 'Stranger',
  publisher: 'Vintage',
  author: 'Camus'
}
{
  _id: ObjectId('663ec31417e333091927fc3f'),
  book_code: 75,
  book_title: 'Fall',
  publisher: 'Vintage',
  author: 'Camus'
}
books> |
```

- c. Show all books that are published by Vintage, LB Books or Plume (use IN)

```
> db.books.find({publisher: { $in: ["Vintage", "LB Books", "Plume"] }});
< {
  _id: ObjectId('663ec31417e333091927fc38'),
  book_code: 22,
  book_title: 'Stranger',
  publisher: 'Vintage',
  author: 'Camus'
}
{
  _id: ObjectId('663ec31417e333091927fc3a'),
  book_code: 18,
  book_title: 'Beloved',
  publisher: 'Plume',
  author: 'Morrison'
}
{
  _id: ObjectId('663ec31417e333091927fc3b'),
  book_code: 37,
  book_title: 'Nine',
  publisher: 'LB Books',
  author: 'Salinger'
}
{
  _id: ObjectId('663ec31417e333091927fc3d'),
  book_code: 61,
  book_title: 'Jazz',
  publisher: 'Plume',
  author: 'Morrison'
}
```

```
{
  _id: ObjectId('663ec31417e333091927fc3e'),
  book_code: 69,
  book_title: 'Franny',
  publisher: 'LB Books',
  author: 'Salinger'
}
{
  _id: ObjectId('663ec31417e333091927fc3f'),
  book_code: 75,
  book_title: 'Fall',
  publisher: 'Vintage',
  author: 'Camus'
}
{
  _id: ObjectId('663ec31417e333091927fc41'),
  book_code: 98,
  book_title: 'Catcher',
  publisher: 'LB Books',
  author: 'Salinger'
}
```

books >

- d. Show all books published by Scribner or Plume (use OR)

```
> db.books.find({$or: [{ publisher: "Scribner" }, { publisher: "Plume" }]});
< {
  _id: ObjectId('663ec31417e333091927fc39'),
  book_code: 13,
  book_title: 'Dreamcatcher',
  publisher: 'Scribner',
  author: 'King'
}
{
  _id: ObjectId('663ec31417e333091927fc3a'),
  book_code: 18,
  book_title: 'Beloved',
  publisher: 'Plume',
  author: 'Morrison'
}
{
  _id: ObjectId('663ec31417e333091927fc3c'),
  book_code: 57,
  book_title: 'Catch 22',
  publisher: 'Scribner',
  author: 'Heller'
}
{
  _id: ObjectId('663ec31417e333091927fc3d'),
  book_code: 61,
  book_title: 'Jazz',
  publisher: 'Plume',
  author: 'Morrison'
}
```

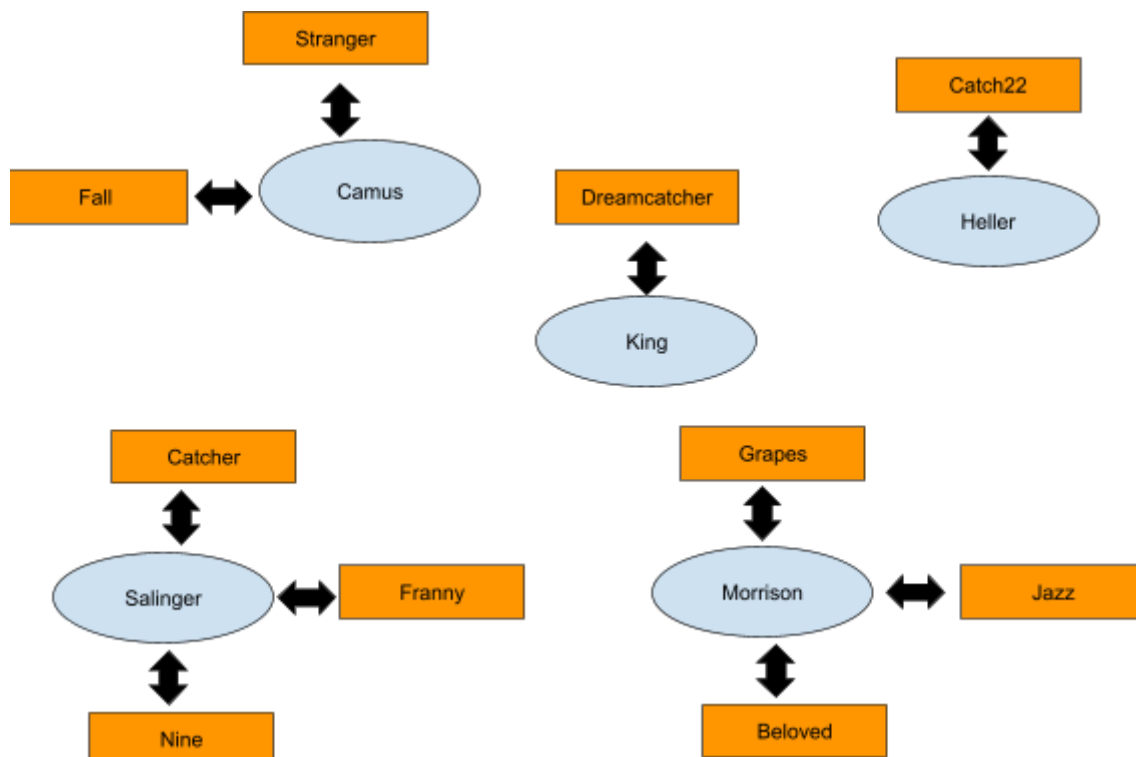


- e. Show all books published by Penguin and written by Morrison (use AND)

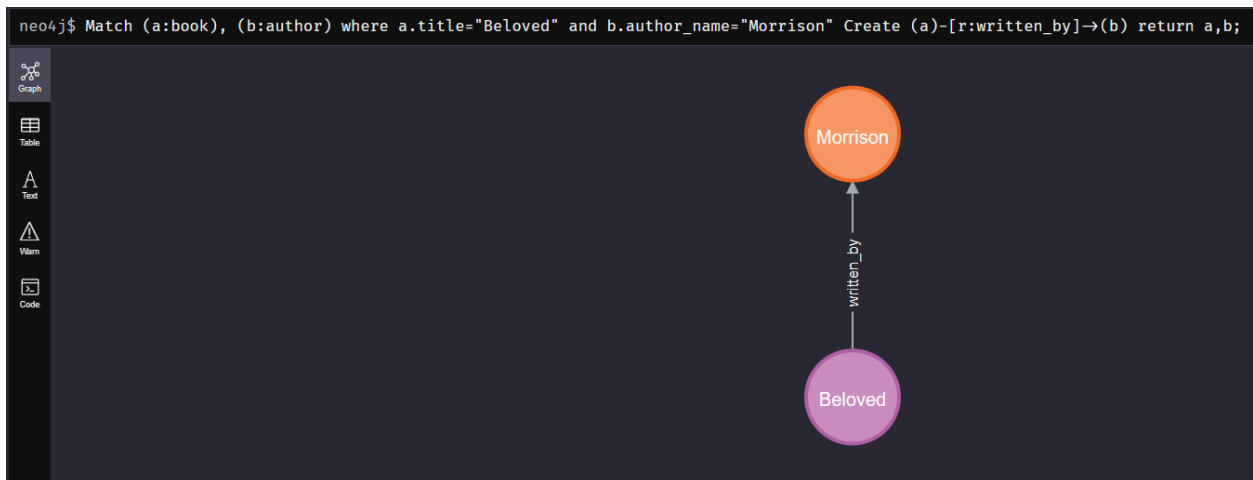
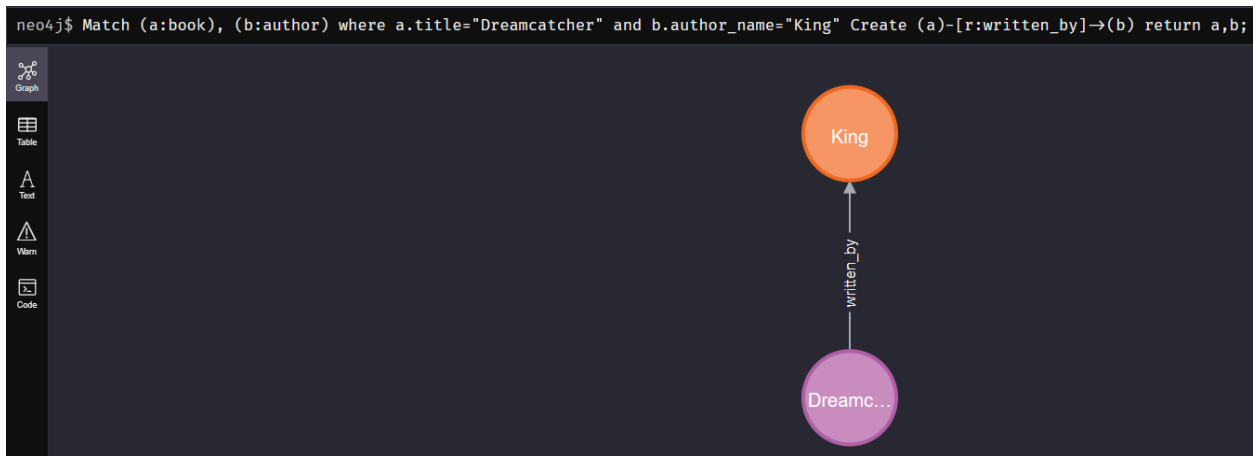
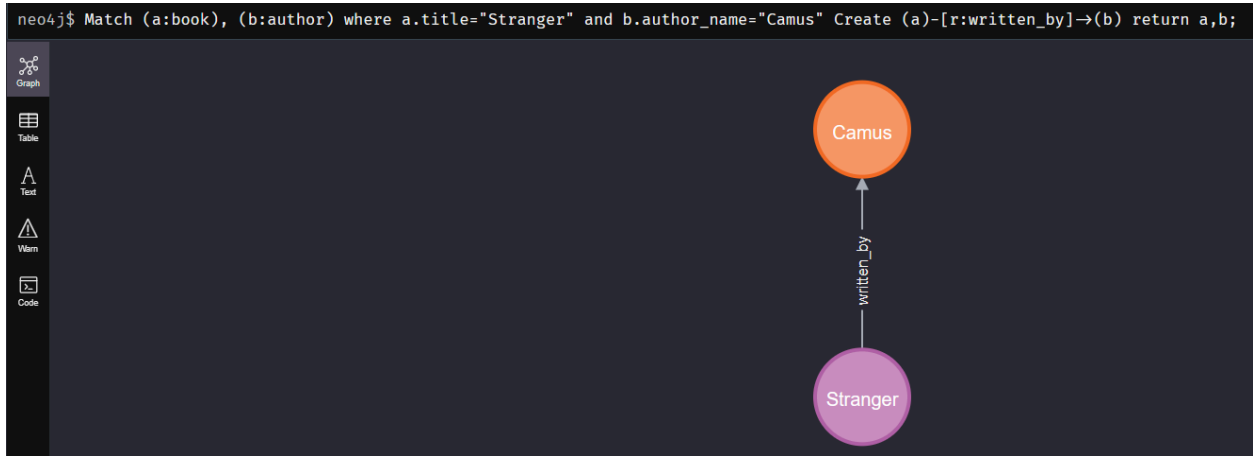
```
> db.books.find({$and: [{ publisher: "Penguin" }, { author: "Morrison" }]});
< {
  _id: ObjectId('663ec31417e333091927fc40'),
  book_code: 96,
  book_title: 'Grapes',
  publisher: 'Penguin',
  author: 'Morrison'
}
books>
```

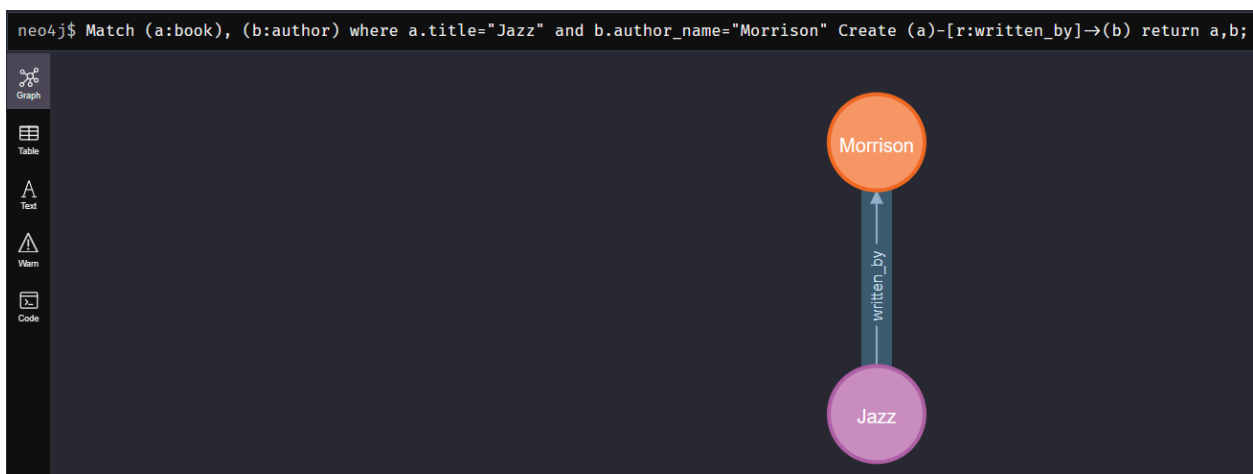
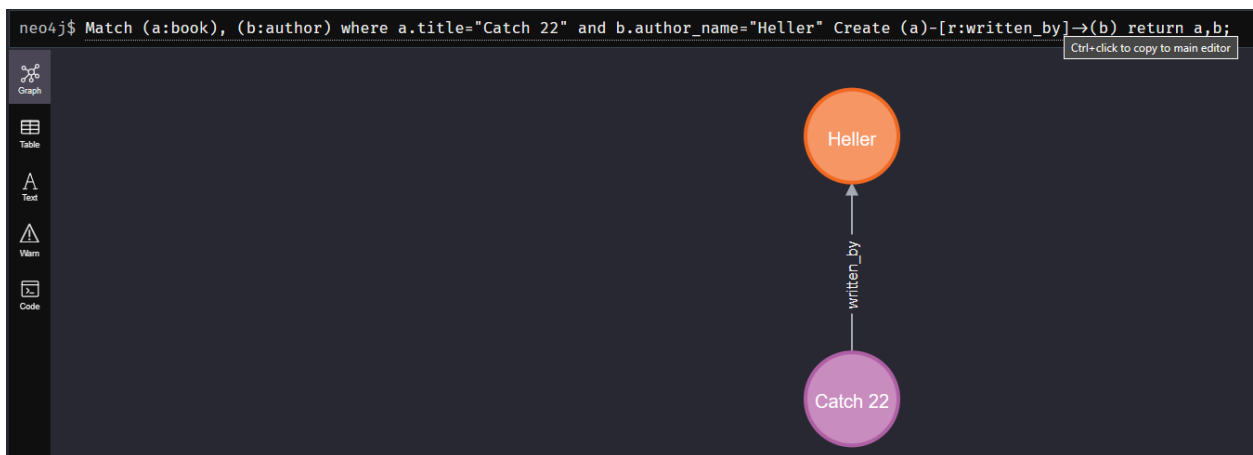
### Part 3: Graph Store – Use Neo4j

- A. Access Neo4j to complete Part 3: <https://sandbox.neo4j.com>
  - B. For each of the following steps, include screenshots of your code and results
  - C. Use the data in Table 1 below
1. Graphically represent the relationship between author and book(s) – use word or any other graphic tool



## 2. Use Neo4j code to create the relationships between author and books





```
neo4j$ Match (a:book), (b:author) where a.title="Franny" and b.author_name="Salinger" Create (a)-[r:written_by]→(b) return a,b;
```

Graph

Table

Text

Warn

Code

Salinger

written\_by

Franny

```
neo4j$ Match (a:book), (b:author) where a.title="Fall" and b.author_name="Camus" Create (a)-[r:written_by]→(b) return a,b;
```

Graph

Table

Text

Warn

Code

Camus

written\_by

Fall

```
neo4j$ Match (a:book), (b:author) where a.title="Grapes" and b.author_name="Morrison" Create (a)-[r:written_by]→(b) return a,b;
```

Graph

Table

Text

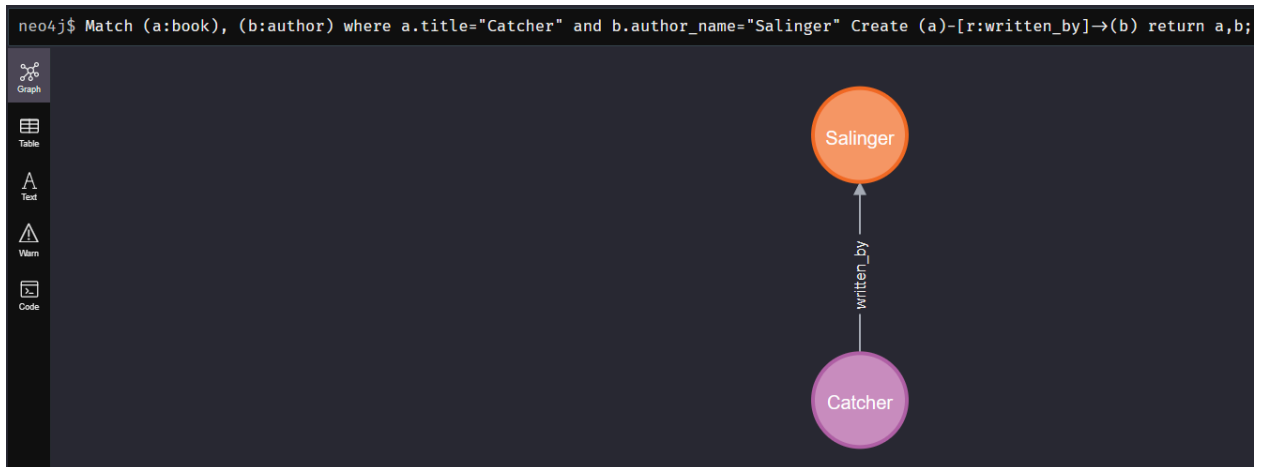
Warn

Code

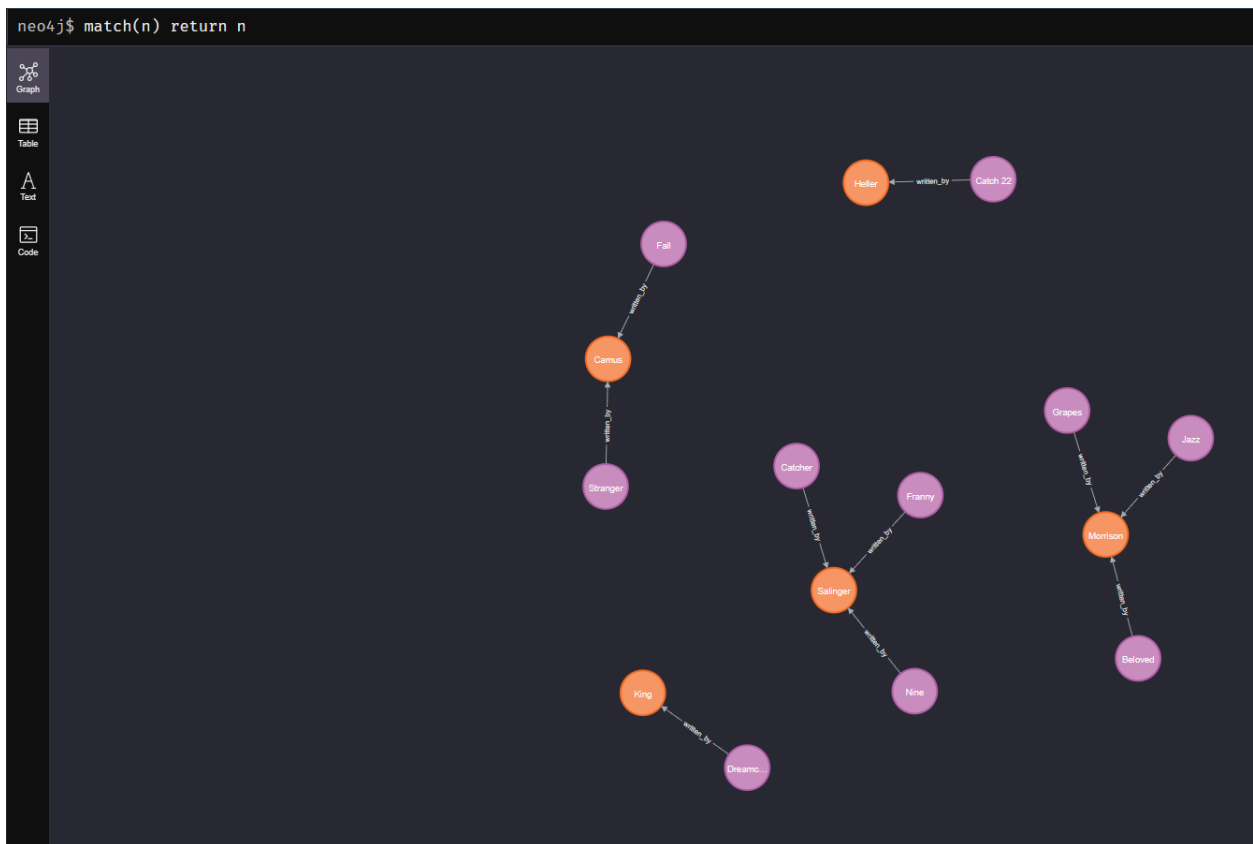
Morrison

written\_by

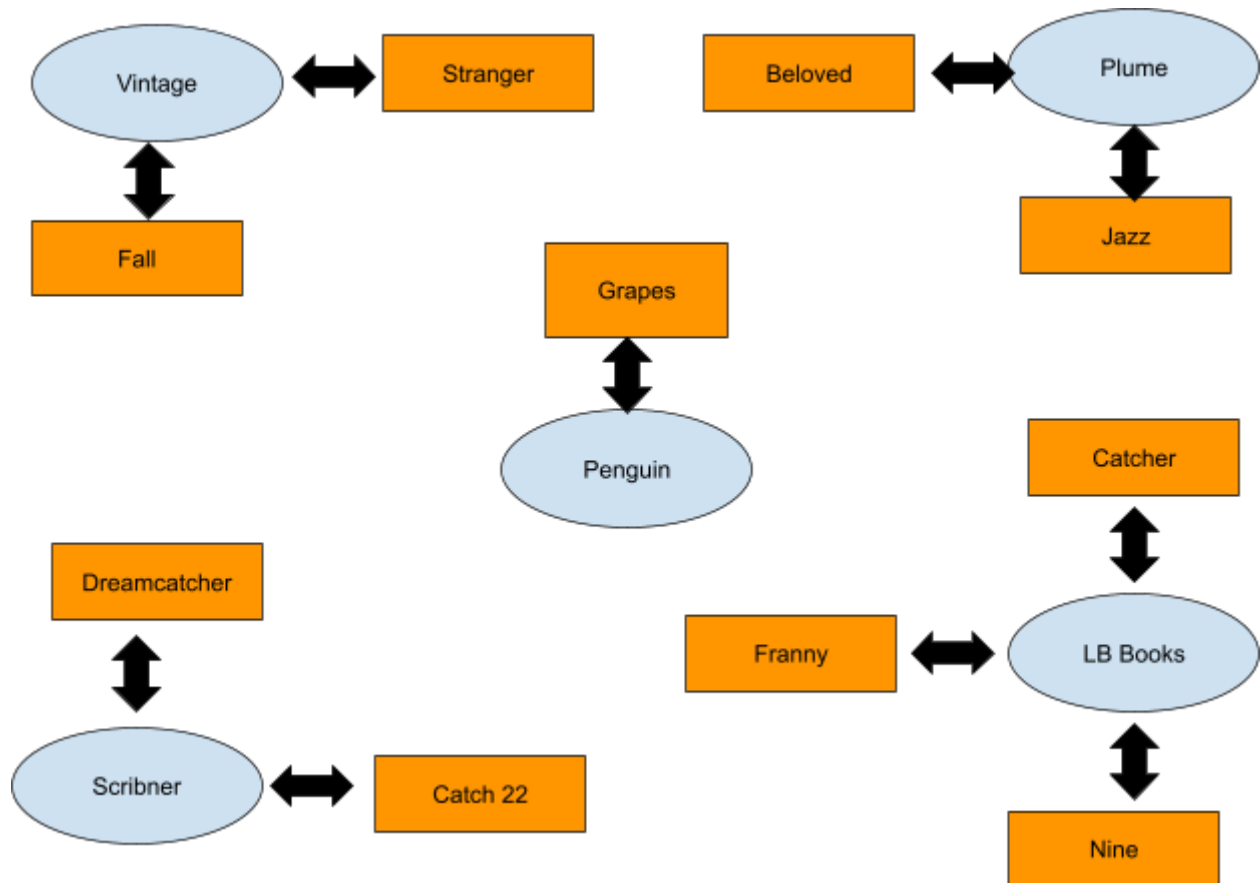
Grapes



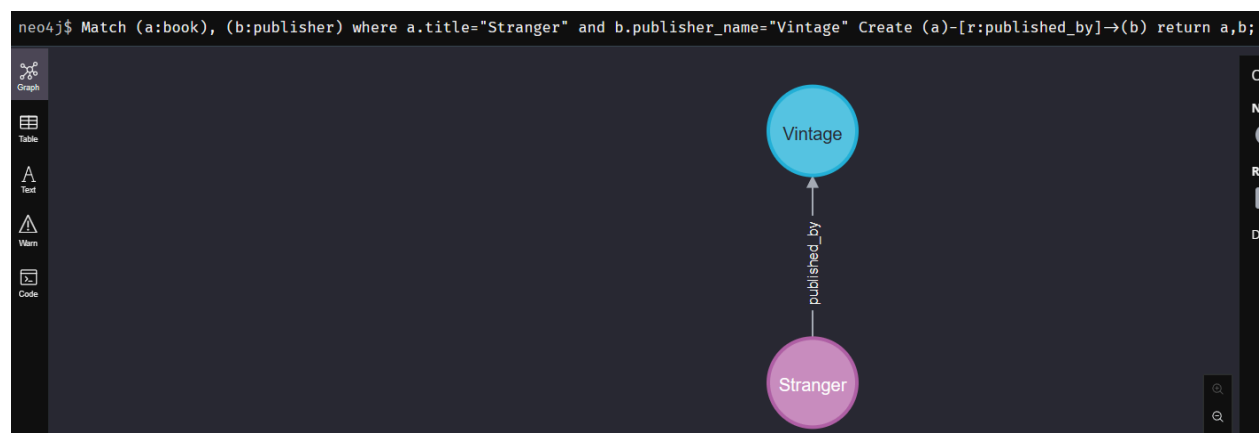
3. Show all the nodes and relationships

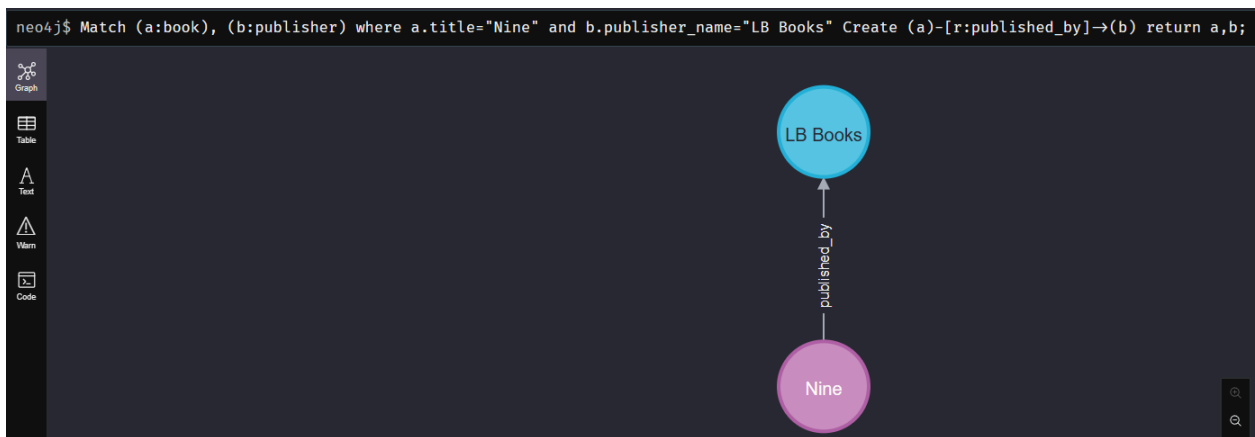
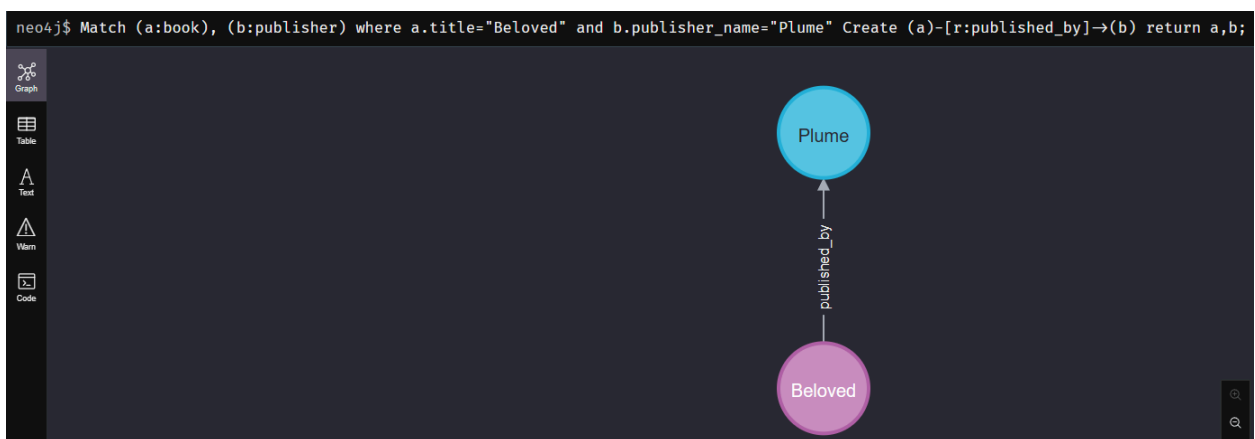
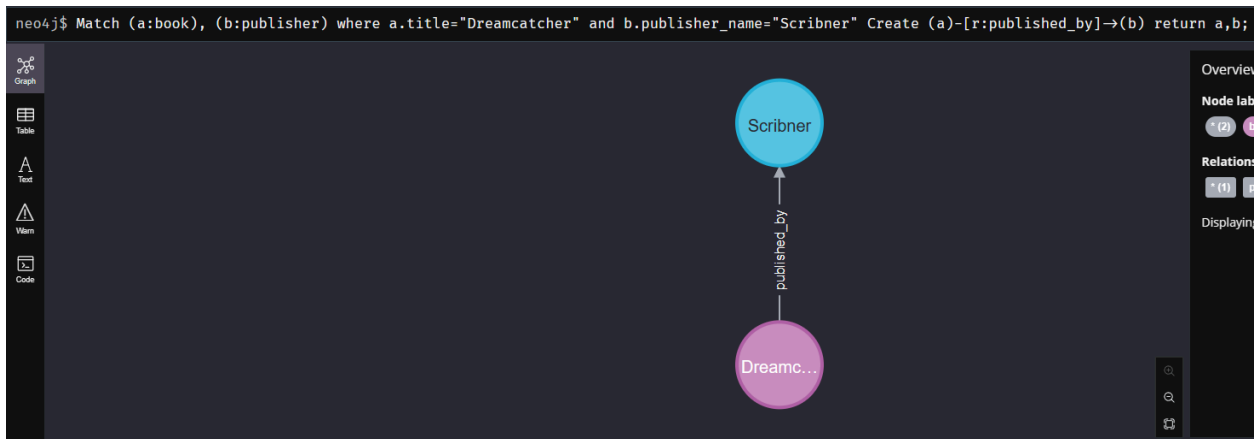


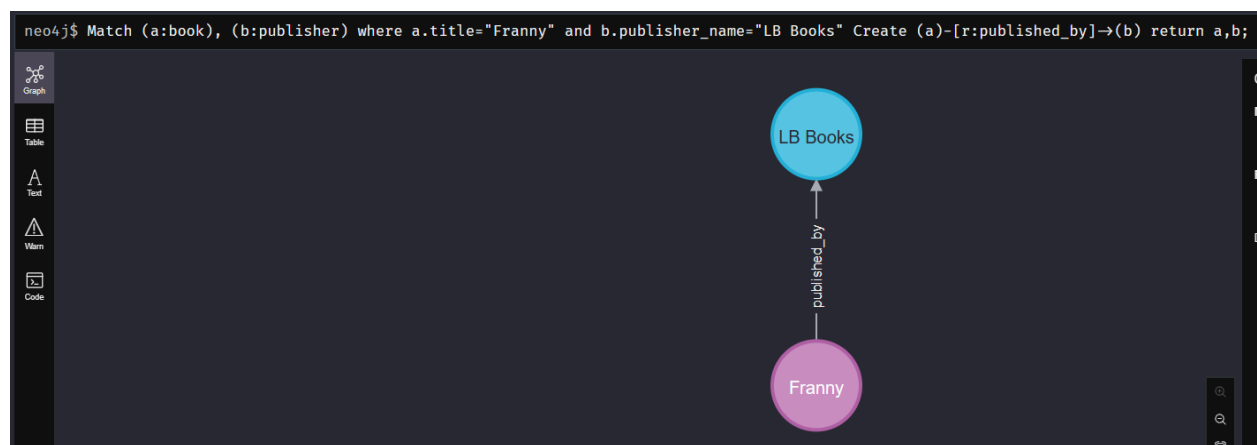
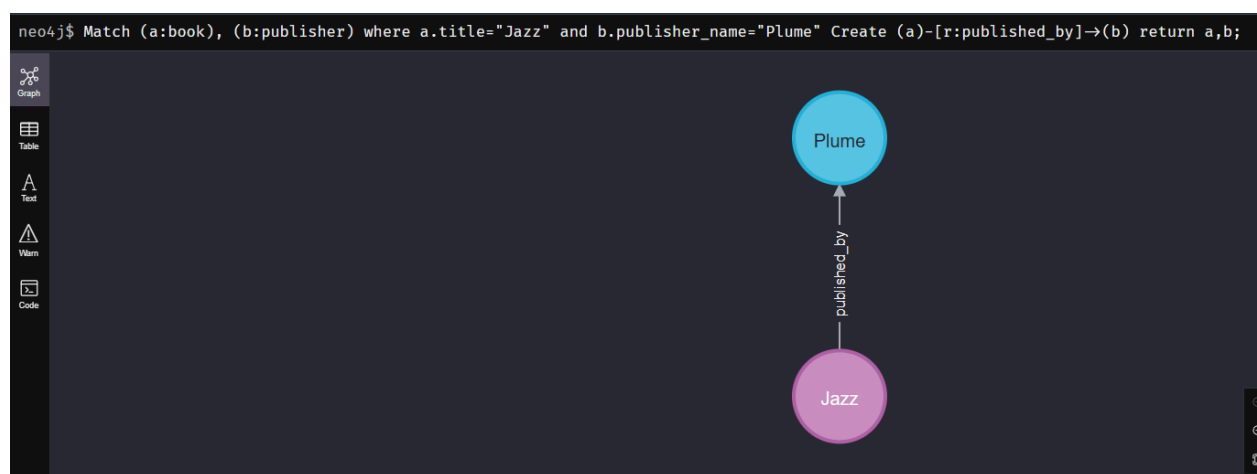
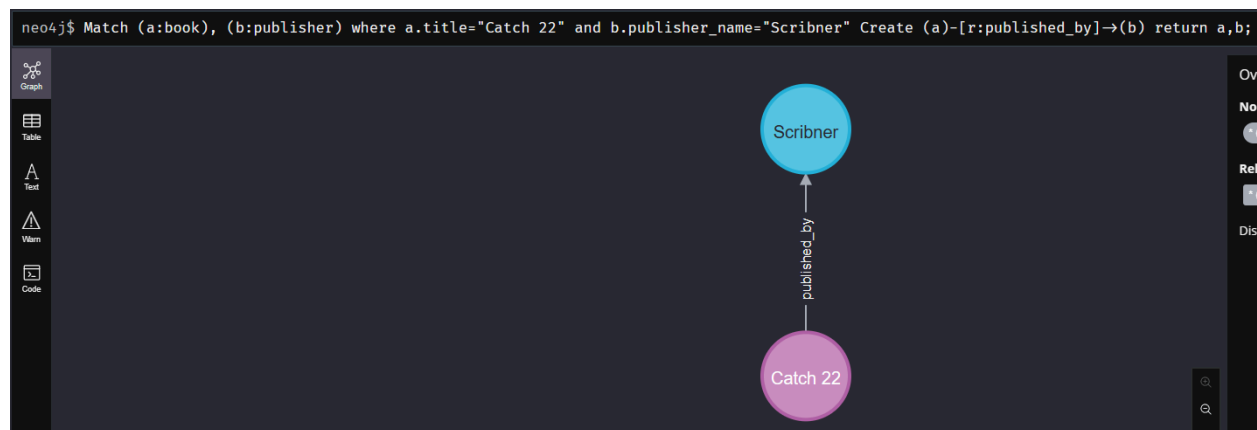
4. Graphically represent the relationship between publisher and book(s) – use word or any other graphic tool



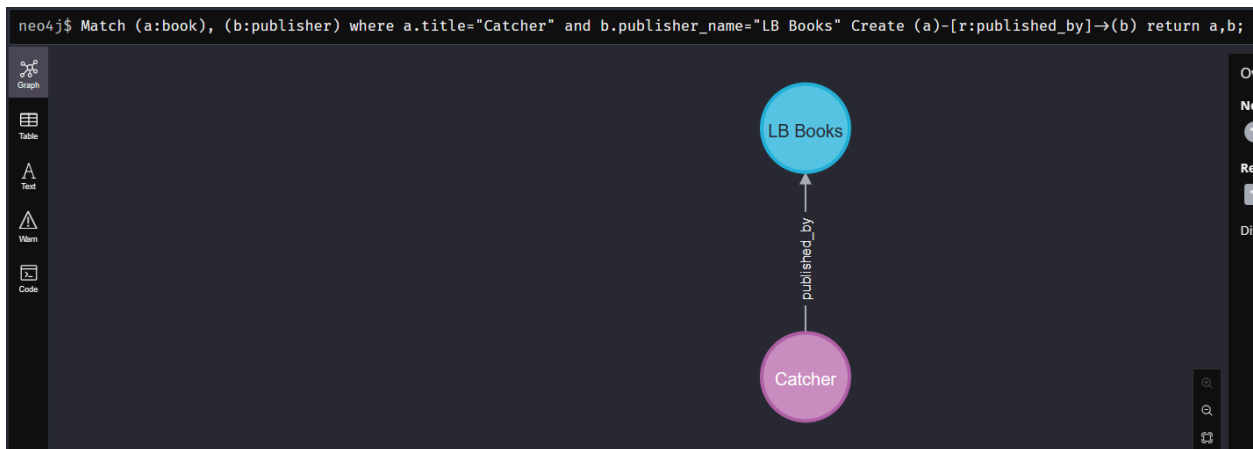
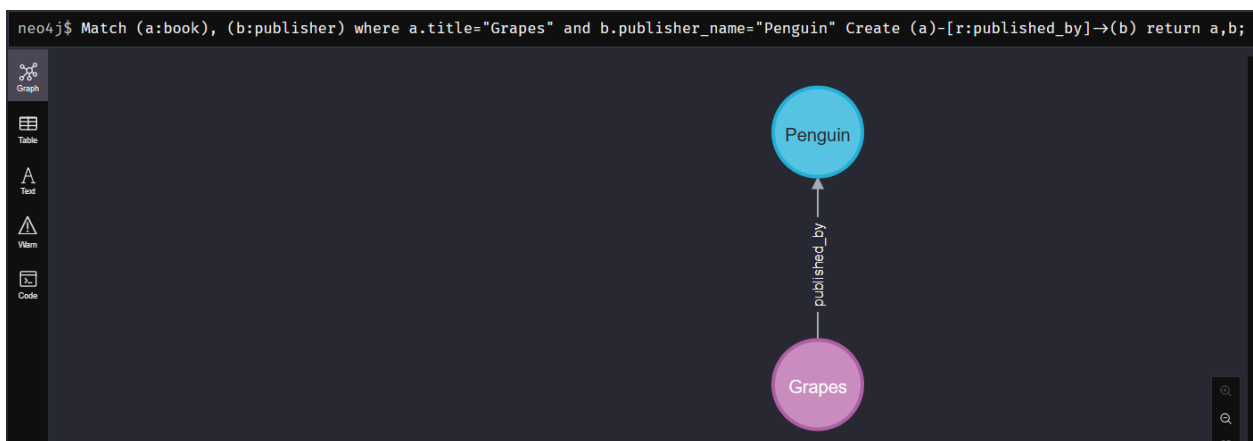
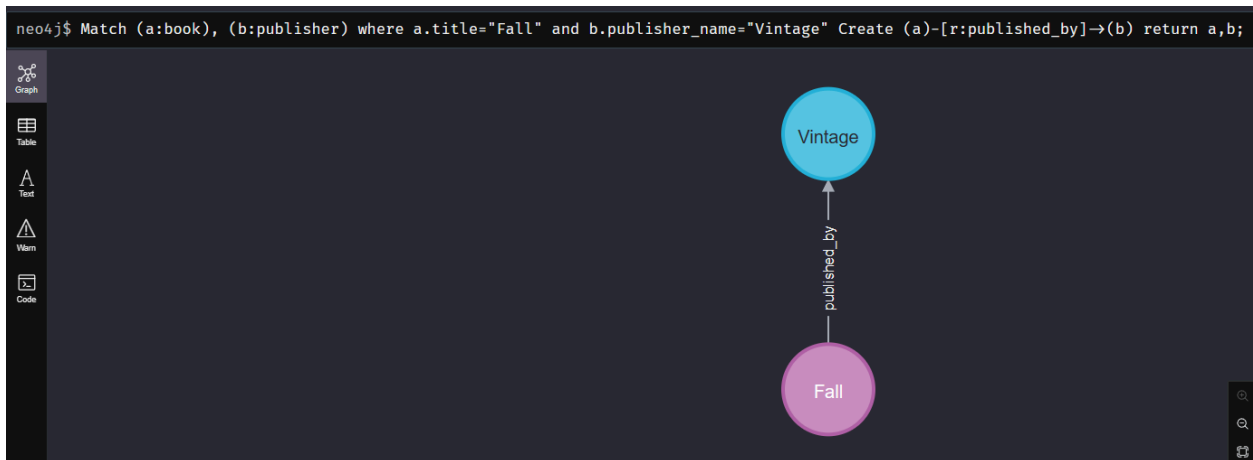
5. Use Neo4j code to create the relationships between publisher and book(s)



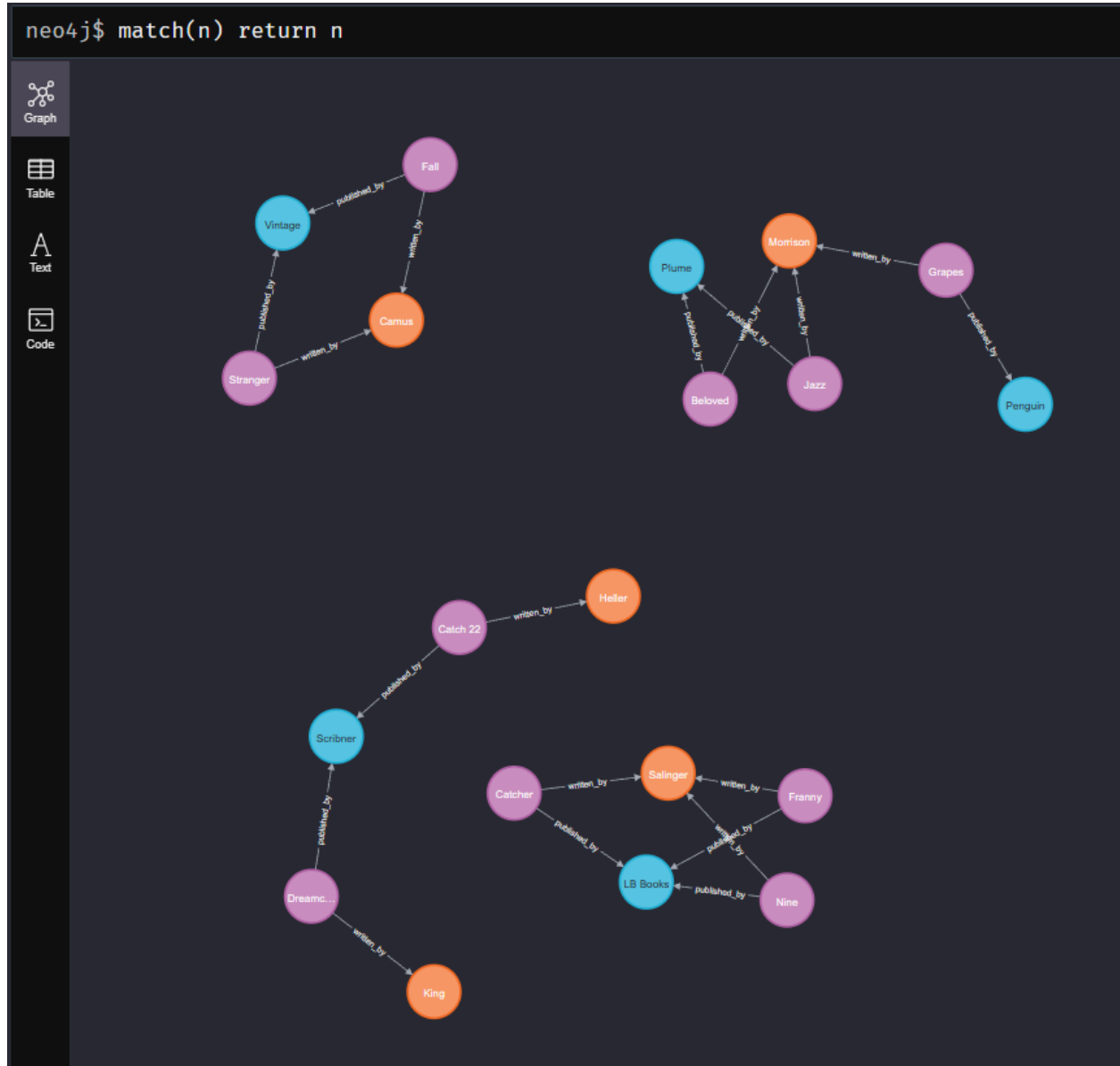




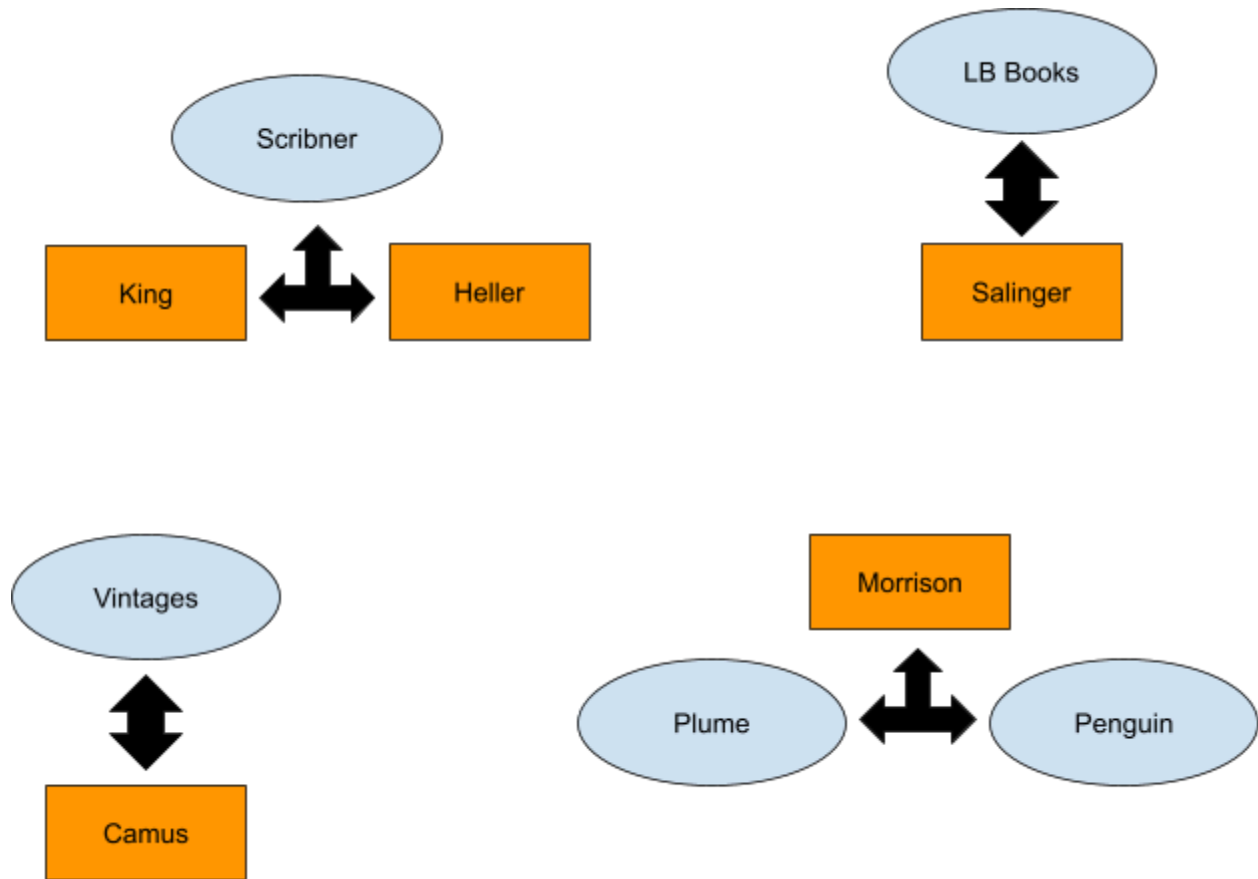




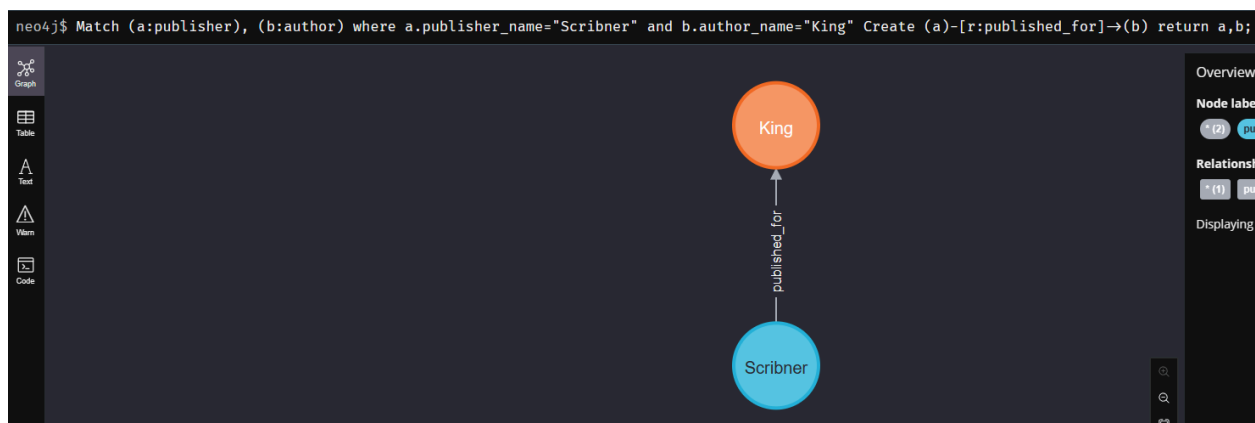
6. Show all the nodes and relationships

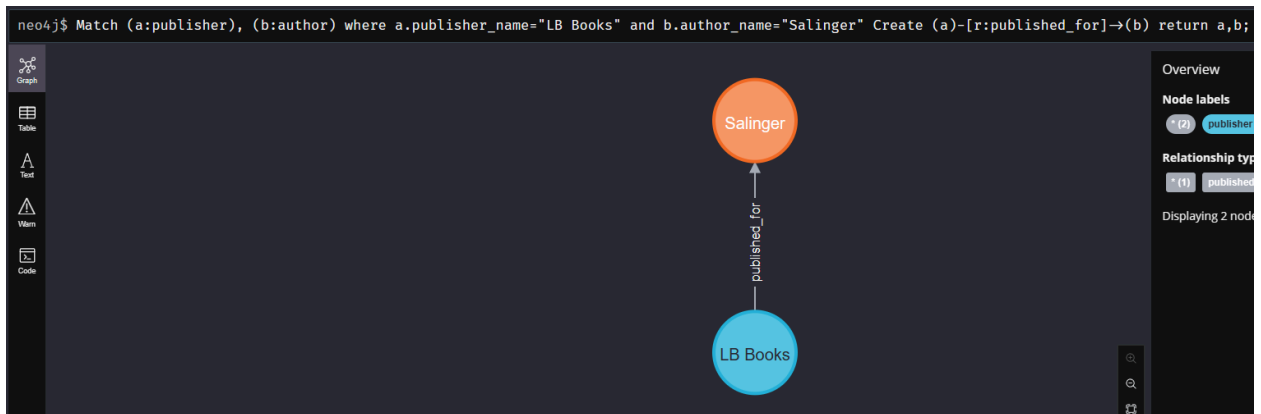
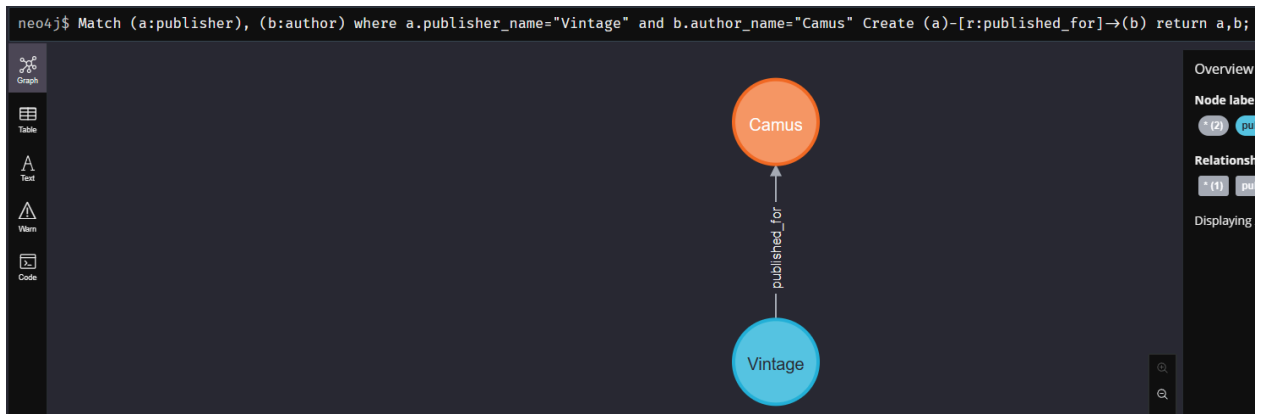
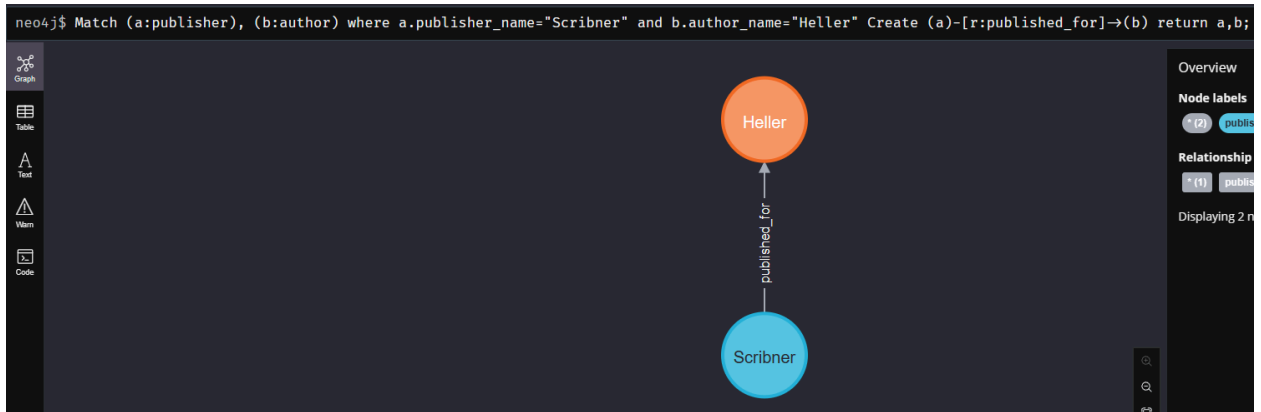


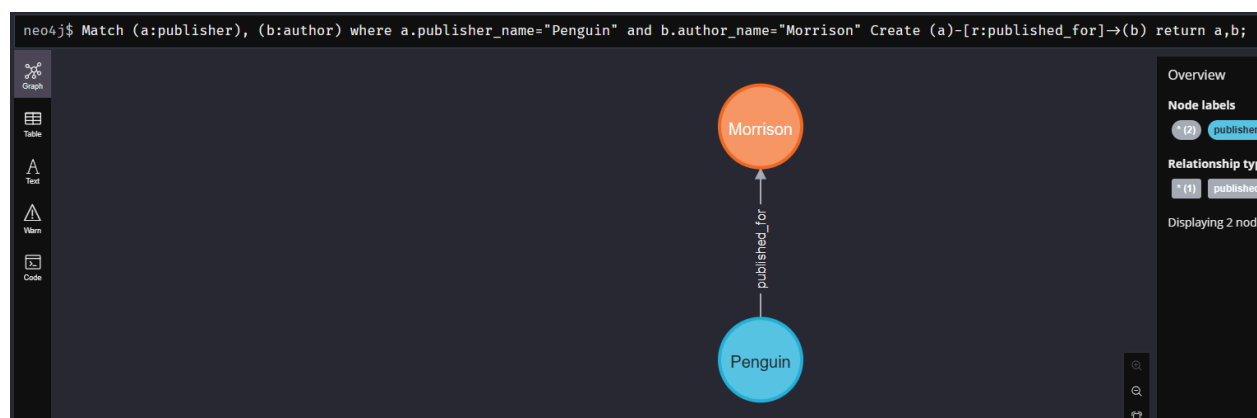
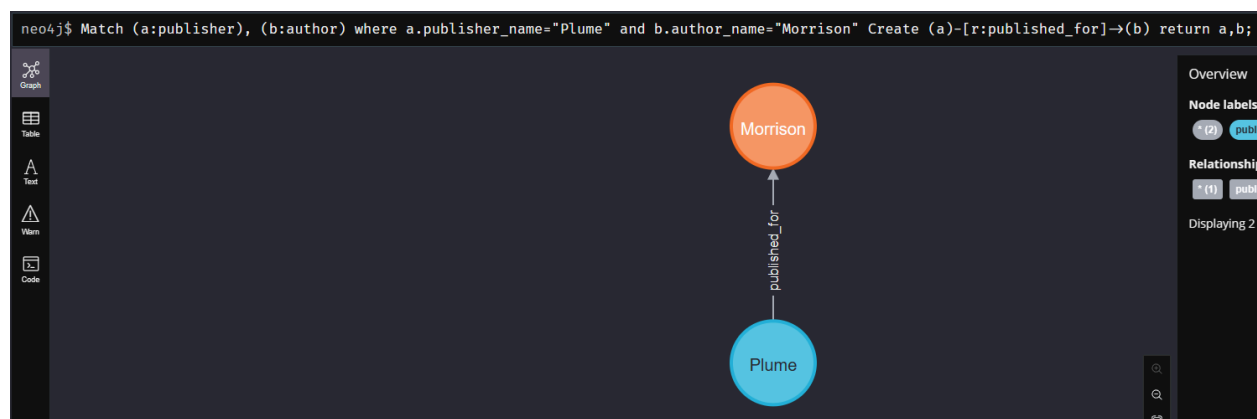
7. Graphically represent the relationship between publisher and author – use word or any other graphic tool



8. Use Neo4j code to create the relationships between publisher and author

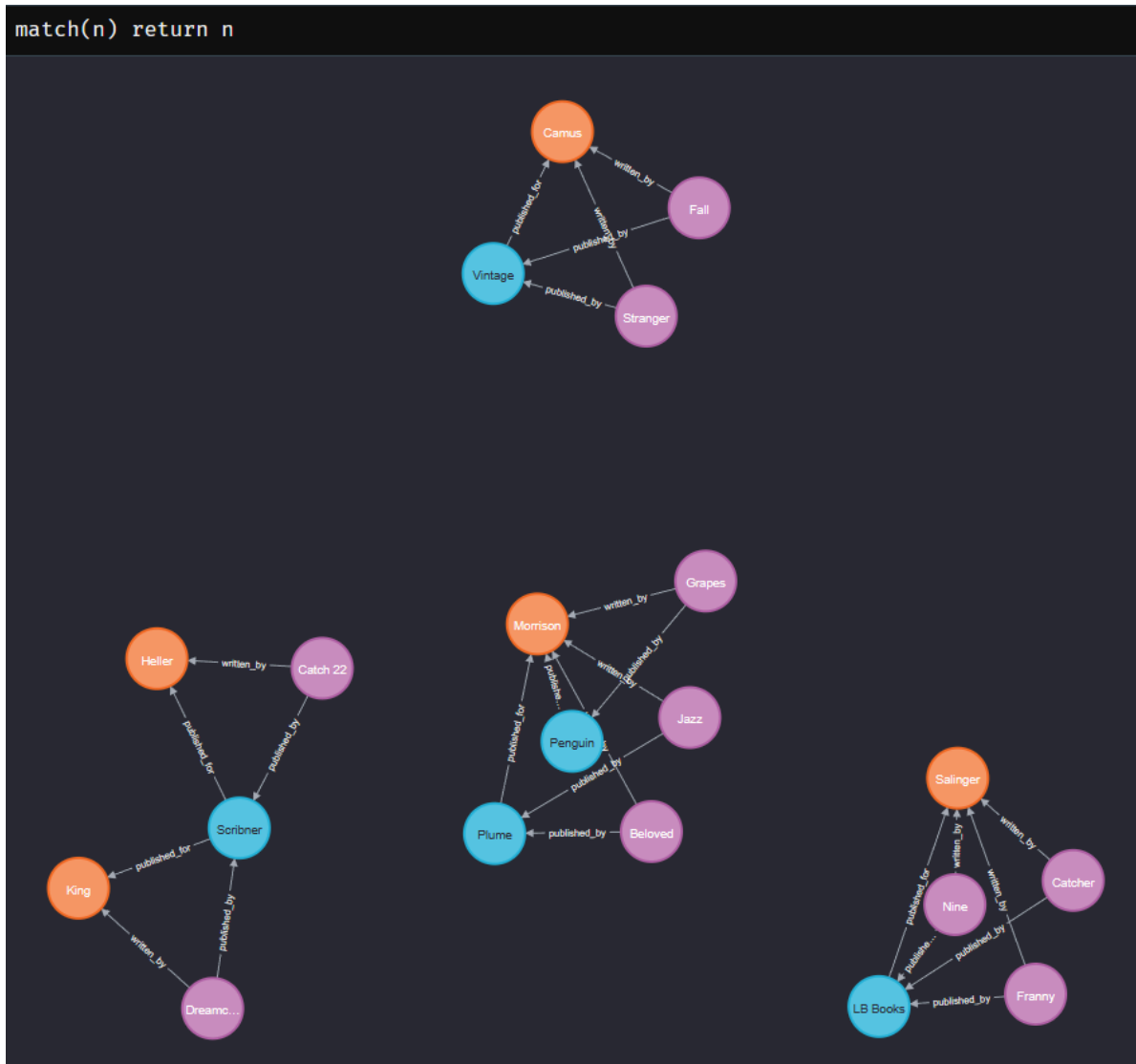






9. Show all the nodes and relationships

```
match(n) return n
```



10. Sort the names in descending order

```
neo4j$ MATCH (publisher)-[:published_for]-(author) RETURN author.author_name, publisher.publisher_name ORDER BY author.author_name DESC
```

	author.author_name	publisher.publisher_name
1	"Salingor"	"LB Books"
2	"Morrison"	"Plume"
3	"Morrison"	"Penguin"
4	"King"	"Scribner"
5	"Heller"	"Scribner"
6	"Camus"	"Vintage"

Started streaming 6 records after 8 ms and completed after 9 ms.

11. Which publisher employs the author morrison

```
neo4j$ MATCH (publisher)-[:published_for]-(author) WHERE author.author_name = "Morrison" RETURN publisher.publisher_name as Publisher
```

	Publisher
1	"Plume"
2	"Penguin"

Started streaming 2 records after 8 ms and completed after 9 ms.

## Part 4: Experience recap

Now that you have had practice implementing the same data in a relational database and two NoSQL stores,

1. What can you say about your experience using Oracle, MongoDB and Neo4j in this exercise

I found MongoDB to be really easy to use and I was able to complete the entire section relatively quickly. Even though I have the most experience using Oracle over the other systems, it took me some time to get started and code out all the data before I could run the queries. Neo4j is visually great but was very tedious on setting up and using. Neo4j was probably my least favorite system to work on for this project and took me the longest to complete. Feels very bad to not be able to view the entire session history. I tried to look it up online but could not find a way to do it. I accidentally closed the Neo4j browser I had open while working on the project and had to scrap sections and restart, which was not fun. Definitely was a problem with me more so than the system but felt like a major inconvenience for a minor mistake.

2. What can you say about the differences in use among the three systems (Oracle, MongoDB and Neo4j)

For dealing with this particular dataset I felt like Oracle did not provide enough insights for the amount of work to set it up. Oracle definitely had the most coding and preplanning involved. MongoDB was the fastest and simplest coding of all. I was able to add all the books in a single command, with all the pertinent information. Creating the collection was a single line of code with one keyword and the name of the collection. Neo4j had a lot of coding but you were able to get around it because you can reuse the commands and just change certain information. This is what made it feel tedious, unlike MongoDB where I could just insert all the documents at the same time. I didn't even create the two way relationships between the different nodes like we did in class but it still took me a long time to get everything done. In the end however, the visualization of the relationships between nodes was very nice for Neo4j.

3. Identify why you would use each of these systems:

a. Oracle

I think that Oracle would be really good for handling data that has a lot of features among all the entries (ie. same structure). Although you can get some nice insights and perform some useful queries on the dataset we used for this project, it felt like overkill because we didn't have a lot of different features to separate the data on. For me, the publisher table and the author table had only two features (id and name) and I had the id feature auto generated based on the order they were entered. I made a junction table with the author and publisher but it wasn't really necessary to handle the assignment. I think it was good practice to do so in order to deal with the many- to- many relationships present in the data but maybe not necessary for the assignment.

b. MongoDB

This seemed perfect for handling this kind of assignment. The data has pretty much the same features but it's not so many features that makes it complicated. With the MongoDB framework it wouldn't make too much of a difference anyway, since it allows you to work with a flexible schema. Querying was fast, easy, and didn't require a lot of coding. Typically, I'd use MongoDB for data that has different types or amounts of features. The strong query language will allow a user to hone in a specific detail or feature from the data for their desired purpose.

c. Neo4j

I would use Neo4j when the connections or relationships between the data points were extensive or complex. I didn't really take advantage of that feature for this particular assignment but I could definitely see how multiple relationships between nodes could be formed. The real value is in the visual created by the nodes and all



the relationships. I think Neo4j would be the best system to use when the relationships are the targeted insight.

4. What different view of the data do you get from using Oracle, MongoDB and Neo4j

It felt like I got aggregate data from using Oracle and MongoDB. From those systems you can get a nice summary of the data by focusing on different features. I got relationship/connection data from using Neo4j. You can see how different groups work together, with a bit more information you can perhaps see what kind of works or themes the author or publishers like to work with.

5. Describe how a company can benefit from using all three systems. Use what you have learned in class and by completing this exercise to validate your opinion.

An Ecommerce company can benefit from using all three systems. They could use Oracle to handle things like transaction history. They could use MongoDB for handling the different content and resources related to displaying their products, like images and product features. They could use Neo4j for promoting other products like a recommendation system.