CST4724 **Final Project** **FALL 2018**

Part 1: Relational Databases – Use ORACLE

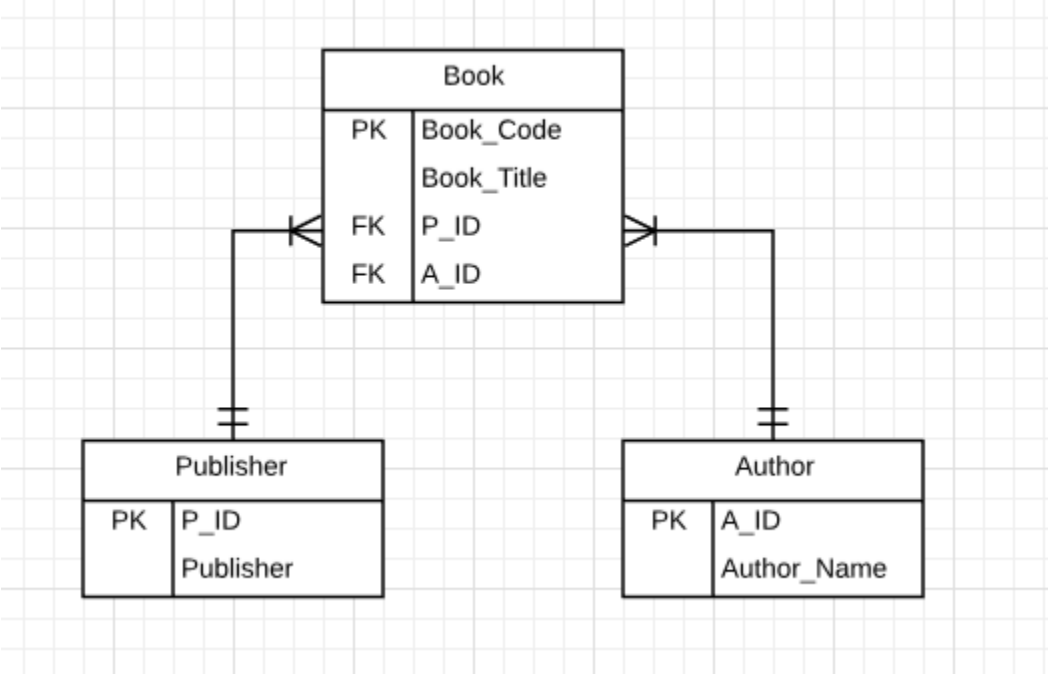
A. Normalize the data in table 1 attached – show the EER diagram for your final results

Book			
Book_Code	Book_Tittle	P_ID	A_ID
22	Stranger	P1	A1
13	Dream Catcher	P2	A2
18	Beloved	P3	A3
37	Nine	P4	A4
57	Catch 22	P2	A5
61	Jazz	P3	A3
69	Franny	P4	A4
75	Fall	P1	A1
96	Grapes	P5	A3
98	Catcher	P4	A4

Author	
A_ID	Author_Name
A1	Camus
A2	King
A3	Morrison
A4	Slinger
A5	Heller

Publisher	
P_ID	Publisher
P1	Vintage
P2	Scribner
P3	Plume
P4	LB Books
P5	Penguin

Physical Model



B. For each of the following steps, include screen shots of your code and results

1. Use SQL code to create the tables
    a. Table Book

```
SQL> create table Book
  2  (
  3  Book_Code number(3) primary key,
  4  Book_Tittle varchar2(20) NOT NULL,
  5  P_ID varchar2(3) NOT NULL,
  6  A_ID varchar2(3) NOT NULL,
  7  CONSTRAINT PID_FK FOREIGN KEY (P_ID ) REFERENCES Publisher(P_ID),
  8  CONSTRAINT AID_FK FOREIGN KEY (A_ID ) REFERENCES Author(A_ID)
  9  );

Table created.
```

b. Table Publisher

```
SQL> create table Publisher
  2  (
  3  P_ID varchar2(3) primary key,
  4  Publisher_Name varchar2(15) NOT NULL
  5  );

Table created.
```

c. Table Author

```
SQL> create table Author
  2  (
  3  A_ID varchar2(3) primary key,
  4  Author_Name varchar2(15) NOT NULL
  5  );

Table created.
```

2. Use SQL code to show that each of the tables has been created

a. Table Book

```
SQL> describe book;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_CODE                                 NOT NULL NUMBER(3)
 BOOK_TITTLE                               NOT NULL VARCHAR2(20)
 P_ID                                      NOT NULL VARCHAR2(3)
 A_ID                                      NOT NULL VARCHAR2(3)
```

b. Table Publisher

```
SQL> describe publisher;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 P_ID                                      NOT NULL VARCHAR2(3)
 PUBLISHER_NAME                            NOT NULL VARCHAR2(15)

SQL>
```

c. Table Author

```
SQL> describe Author;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 A_ID                                      NOT NULL VARCHAR2(3)
 AUTHOR_NAME                               NOT NULL VARCHAR2(15)
```

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

```
SQL> insert into Book values(22,'Stranger','P1','A1');

1 row created.

SQL> insert into Book values(13,'Dream Catcher','P2','A2');

1 row created.

SQL> insert into Book values(18,'Beloved','P3','A3');

1 row created.

SQL> insert into Book values(37,'Nine','P4','A4');

1 row created.

SQL> insert into Book values(57,'Catch 22','P2','A5');

1 row created.

SQL> insert into Book values(61,'Jazz','P3','A3');

1 row created.

SQL> insert into Book values(69,'Franny','P4','A4');

1 row created.

SQL> insert into Book values(75,'Fall','P1','A1');

1 row created.

SQL> insert into Book values(96,'Grapes','P5','A3');

1 row created.

SQL> insert into Book values(98,'Catcher','P4','A4');

1 row created.
```

    b. Table Publisher

```
SQL> insert into Publisher values('P1','Vintage');

1 row created.

SQL> insert into Publisher values('P2','Scribner');

1 row created.

SQL> insert into Publisher values('P3','Plume');

1 row created.

SQL> insert into Publisher values('P4','LB Books');

1 row created.

SQL> insert into Publisher values('P5','Penguin');

1 row created.
```

c. Table Author

```
SQL> insert into Author values('A1','Camus');

1 row created.

SQL> insert into Author values('A2','King');

1 row created.

SQL> insert into Author values('A3','Morrison');

1 row created.

SQL> insert into Author values('A4','Salinger');

1 row created.

SQL> insert into Author values('A5','Heller');

1 row created.
```

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

```
SQL> select*from book;

 BOOK_CODE BOOK_TITTLE          P_I A_I
---------- -------------------- --- ---
        22 Stranger             P1  A1
        13 Dream Catcher        P2  A2
        18 Beloved              P3  A3
        37 Nine                 P4  A4
        57 Catch 22             P2  A5
        61 Jazz                 P3  A3
        69 Franny               P4  A4
        75 Fall                 P1  A1
        96 Grapes               P5  A3
        98 Catcher              P4  A4

10 rows selected.
```

b. Table Publisher

```
SQL> select * from Publisher;

P_I PUBLISHER_NAME
--- ---------------
P1  Vintage
P2  Scribner
P3  Plume
P4  LB Books
P5  Penguin

SQL>
```

c. Table Author

```
SQL> select * from Author;

A_I AUTHOR_NAME
--- ---------------
A1  Camus
A2  King
A3  Morrison
A4  Salinger
A5  Heller
```

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.

```
SQL> select Author_Name,Book_Tittle
  2  from Author, Book
  3  where Author.A_ID=Book.A_ID
  4  order by Author_Name desc;

AUTHOR_NAME     BOOK_TITTLE
--------------- --------------------
Salinger        Nine
Salinger        Catcher
Salinger        Franny
Morrison        Grapes
Morrison        Beloved
Morrison        Jazz
King            Dream Catcher
Heller          Catch 22
Camus           Stranger
Camus           Fall

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.

```
SQL> select Publisher_Name,Book_Tittle
  2  from Publisher, Book
  3  where Publisher.P_ID=Book.P_ID
  4  order by Publisher_Name asc;

PUBLISHER_NAME  BOOK_TITTLE
--------------- --------------------
LB Books        Franny
LB Books        Nine
LB Books        Catcher
Penguin         Grapes
Plume           Beloved
Plume           Jazz
Scribner        Dream Catcher
Scribner        Catch 22
Vintage         Stranger
Vintage         Fall

10 rows selected.

SQL>
```

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.

```
SQL> select A.Author_Name, P.Publisher_Name
  2  From Author A, Publisher P,Book B
  3  where A.A_ID=B.A_ID
  4  AND P.P_ID=B.P_ID
  5  Order by 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

10 rows selected.
```

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

```
SQL> select B.Book_Tittle
  2  From Book B,Author A
  3  where A.A_ID = B.A_ID
  4  AND Author_Name ='Salinger';

BOOK_TITTLE
--------------------
Nine
Franny
Catcher
```

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

```
SQL> select B.Book_Tittle
  2  From Publisher P, Book B
  3  where P.P_ID = B.P_ID
  4  AND Publisher_Name ='Vintage';

BOOK_TITTLE
--------------------
Stranger
Fall

SQL>
```

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)

```
SQL> select B.Book_Tittle, P.Publisher_Name
  2  From Book B, Publisher P
  3  where B.P_ID = P.P_ID
  4  AND Publisher_Name IN ('Vintage','LB Books','Plume');

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

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**

```
SQL> select B.Book_Tittle, P.Publisher_Name
  2  from Book B, Publisher P
  3  where P.P_ID = B.P_ID
  4  AND (P.Publisher_Name ='Scribner'
  5  OR P.Publisher_Name ='Plume');

BOOK_TITTLE          PUBLISHER_NAME
-------------------- --------------
Dream Catcher        Scribner
Beloved              Plume
Catch 22             Scribner
Jazz                 Plume

SQL>
```

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

```
SQL> select B.Book_Tittle, P.Publisher_Name
  2  from Book B, Publisher P, Author A
  3  where B.P_ID = P.P_ID
  4  AND B.A_ID = A.A_ID
  5  AND Author_Name ='Morrison'
  6  AND Publisher_Name ='Penguin';

BOOK_TITTLE          PUBLISHER_NAME
-------------------- --------------
Grapes               Penguin

SQL>
```

Part 2: Document Stores – Use MongoDB

A. Create the data in table 1 in MongoDB

B. For each of the following steps, include screen shots of your code and results

1. Use MongoDB to create the collection(s)

```
> use Books;
```

2. Use MongoDB to show the code used to create the collection(s)

```
> use Books;
switched to db Books
>
```

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

```
> db.Books.insert({"Book_Code":22,"Book_Title":"Stranger","Publisher":"Vintage","Author":"Camus"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":13,"Book_Title":"Dreamcatcher","Publisher":"Scribner","Author":"King"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":18,"Book_Title":"Beloved","Publisher":"Plume","Author":"Morrison"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":37,"Book_Title":"Nine","Publisher":"LB Books","Author":"Salinger"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":57,"Book_Title":"Catch 22","Publisher":"Scribner","Author":"Heller"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":61,"Book_Title":"Jazz","Publisher":"Plume","Author":"Morrison"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":69,"Book_Title":"Franny","Publisher":"LB Books","Author":"Salinger"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":75,"Book_Title":"Fall","Publisher":"Vintage","Author":"Camus"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":96,"Book_Title":"Grapes","Publisher":"Penguin","Author":"Morrison"})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"Book_Code":98,"Book_Title":"Catcher","Publisher":"LB Books","Author":"Salinger"})
WriteResult({ "nInserted" : 1 })
>
```

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

```
> db.Books.find().forEach(printjson);
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d40"),
        "Book_Code" : 22,
        "Book_Title" : "Stranger",
        "Publisher" : "Vintage",
        "Author" : "Camus"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d41"),
        "Book_Code" : 13,
        "Book_Title" : "Dreamcatcher",
        "Publisher" : "Scribner",
        "Author" : "King"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d42"),
        "Book_Code" : 18,
        "Book_Title" : "Beloved",
        "Publisher" : "Plume",
        "Author" : "Morrison"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d43"),
        "Book_Code" : 37,
        "Book_Title" : "Nine",
        "Publisher" : "LB Books",
        "Author" : "Salinger"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d44"),
        "Book_Code" : 57,
        "Book_Title" : "Catch 22",
        "Publisher" : "Scribner",
        "Author" : "Heller"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d47"),
        "Book_Code" : 75,
        "Book_Title" : "Fall",
        "Publisher" : "Vintage",
        "Author" : "Camus"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d48"),
        "Book_Code" : 96,
        "Book_Title" : "Grapes",
        "Publisher" : "Penguin",
        "Author" : "Morrison"
}
{
        "_id" : ObjectId("5c0f331f9b38eaf3ba479d49"),
        "Book_Code" : 98,
        "Book_Title" : "Catcher",
        "Publisher" : "LB Books",
        "Author" : "Salinger"
}
>
```

5. Use MongoDB to run the following queries:
   a. Show all books written by author Salinger

```
> db.Books.find({Author:"Salinger"});
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d43"), "Book_Code" : 37, "Book_Title" : "Nine", "Publisher" : "LB Books", "Author" : "Salinger" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d46"), "Book_Code" : 69, "Book_Title" : "Franny", "Publisher" : "LB Books", "Author" : "Salinger" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d49"), "Book_Code" : 98, "Book_Title" : "Catcher", "Publisher" : "LB Books", "Author" : "Salinger" }
>
```

b. Show all books published by Vintage

```
> db.Books.find({Publisher:"Vintage"});
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d40"), "Book_Code" : 22, "Book_Title" : "Stranger", "Publisher" : "Vintage", "Author" : "Camus" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d47"), "Book_Code" : 75, "Book_Title" : "Fall", "Publisher" : "Vintage", "Author" : "Camus" }
>
```

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

```
> db.Books.find( { Publisher: { $in: [ "Scribner", "LB Books","Plume"] } } );
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d41"), "Book_Code" : 13, "Book_Title" : "Dreamcatcher", "Publisher" : "Scribner", "Author" : "King" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d42"), "Book_Code" : 18, "Book_Title" : "Beloved", "Publisher" : "Plume", "Author" : "Morrison" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d43"), "Book_Code" : 37, "Book_Title" : "Nine", "Publisher" : "LB Books", "Author" : "Salinger" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d44"), "Book_Code" : 57, "Book_Title" : "Catch 22", "Publisher" : "Scribner", "Author" : "Heller" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d45"), "Book_Code" : 61, "Book_Title" : "Jazz", "Publisher" : "Plume", "Author" : "Morrison" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d46"), "Book_Code" : 69, "Book_Title" : "Franny", "Publisher" : "LB Books", "Author" : "Salinger" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d49"), "Book_Code" : 98, "Book_Title" : "Catcher", "Publisher" : "LB Books", "Author" : "Salinger" }
>
```

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

```
> db.Books.find({$or:[{Publisher:"Scribner"},{Publisher:"Plume"}]});
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d41"), "Book_Code" : 13, "Book_Title" : "Dreamcatcher", "Publisher" : "Scribner", "Author" : "King" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d42"), "Book_Code" : 18, "Book_Title" : "Beloved", "Publisher" : "Plume", "Author" : "Morrison" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d44"), "Book_Code" : 57, "Book_Title" : "Catch 22", "Publisher" : "Scribner", "Author" : "Heller" }
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d45"), "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({Publisher:"Penguin",Author:"Morrison"});
{ "_id" : ObjectId("5c0f331f9b38eaf3ba479d48"), "Book_Code" : 96, "Book_Title" : "Grapes", "Publisher" : "Penguin", "Author" : "Morrison" }
>
```

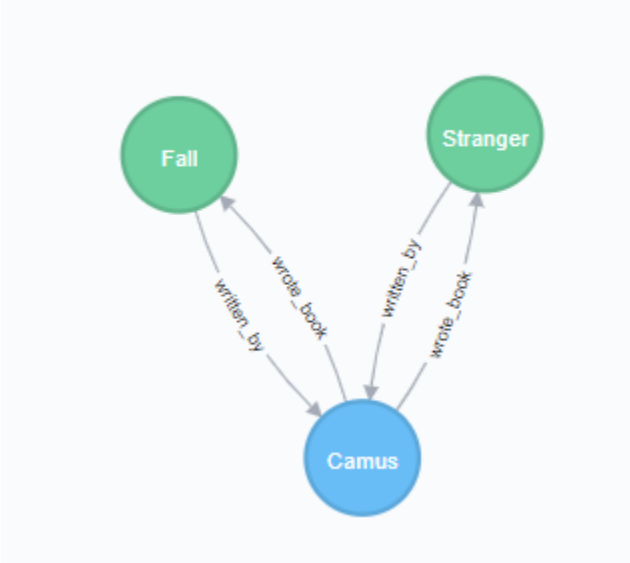
Part 3: Graph Store – Use Neo4j

A. For each of the following steps, include screen shots of your code and results
B. Use the data in Table 1
   1. Graphically represent the relationship between author and book(s) – use word or any other graphic tool

Author	Books
Camus	Stranger Fall
King	Dreamcatcher
Morrison	Beloved jazz Grapes
Slinger	Nine Franny Catcher
Heller	Catch 22

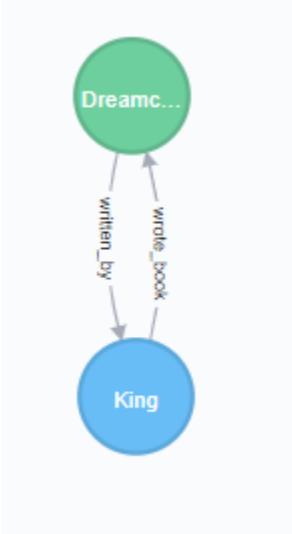
   2. Use Neo4j code to create the relationships between author and books. Show all the nodes and relationships
      - **Relationship Between author Camus and books**

```
MATCH (a:author),(b:book),(c:book)
WHERE a.name = "Camus" AND b.Book_Title = "Stranger" AND c.Book_Title = "Fall"
CREATE (a)-[ra1:wrote_book]->(b), (a)-[ra2:wrote_book]->(c),(b)-[rm1:written_by]->(a),(c)-[rm2:written_by]->
(a)
RETURN a,b,c
```



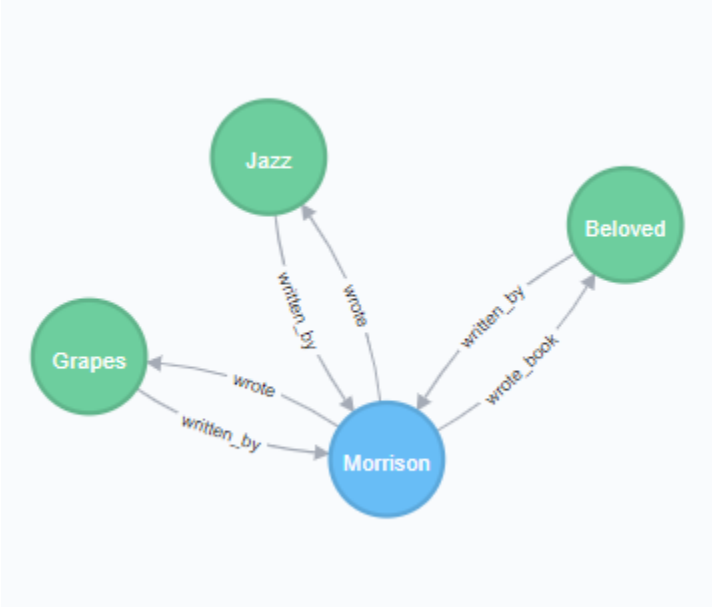
      - **Relationship Between author King and books**

```
MATCH (a:author),(b:book)
WHERE a.name = "King" AND b.Book_Title= "Dreamcatcher"
CREATE (a)-[ra1:wrote_book]->(b),(b)-[rm1:written_by]->(a)
RETURN a,b
```



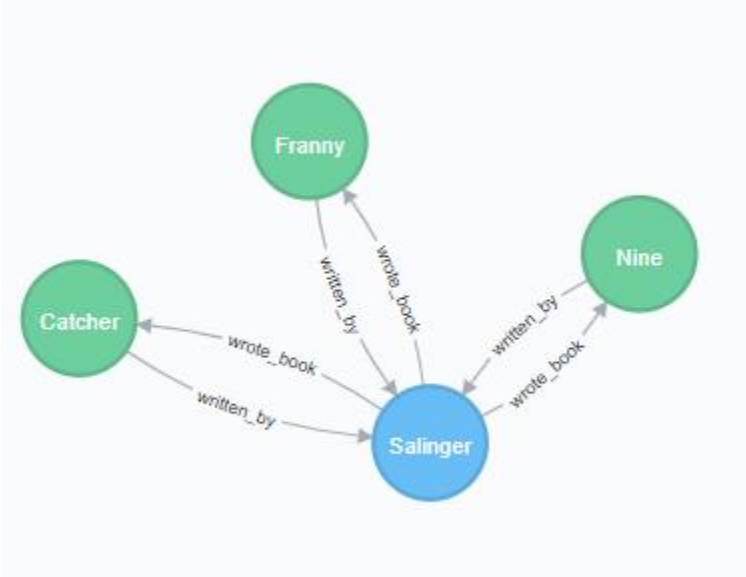
- **Relationship Between author Morrison and books**

```
MATCH (a:author),(b:book),(c:book),(d:book)
WHERE a.name = "Morrison" AND b.Book_Title = "Beloved" AND c.Book_Title= "Jazz" AND d.Book_Title = "Grapes"
CREATE (a)-[ra1:wrote_book]->(b), (a)-[ra2:wrote]->(c),(a)-[ra3:wrote]->(d),(b)-[rm1:written_by]->(a),(c)-
[rm2:written_by]->(a),(d)-[rm3:written_by]->(a)
RETURN a,b,c,d
```



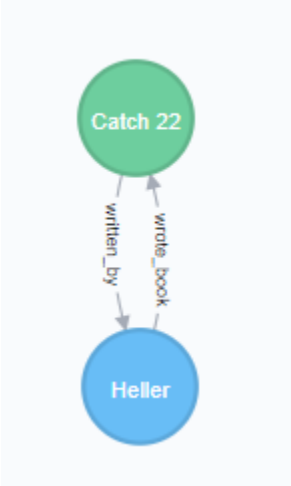
- **Relationship Between author Salinger and books**

```
MATCH (a:author),(b:book),(c:book),(d:book)
WHERE a.name = "Salinger" AND b.Book_Title = "Nine" AND c.Book_Title = "Franny" AND d.Book_Title = "Catcher"
CREATE (a)-[ra1:wrote_book]->(b), (a)-[ra2:wrote_book]->(c),(a)-[ra3:wrote_book]->(d),(b)-[rm1:written_by]->
(a),(c)-[rm2:written_by]->(a),(d)-[rm3:written_by]->(a)
RETURN a,b,c,d
```



- **Relationship Between author Heller and books**

```
MATCH (a:author),(b:book)
WHERE a.name = "Heller" AND b.Book_Title = "Catch 22"
CREATE (a)-[ra1:wrote_book]->(b),(b)-[rm1:written_by]->(a)
RETURN a,b
```

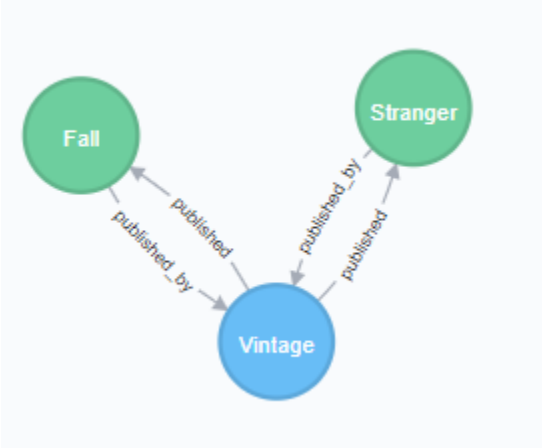


3. Graphically represent the relationship between publisher and book(s) – use word or any other graphic tool

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

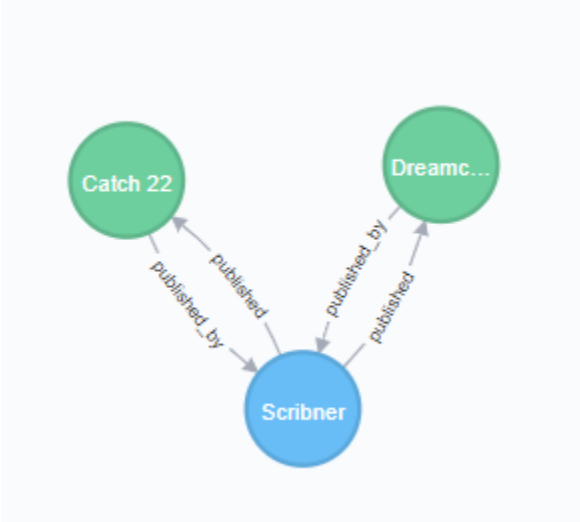
4. Use Neo4j code to create the relationships between publisher and book(s)
   - **Relationship between publisher Vintage and books**

```
MATCH (a:publisher),(b:book),(c:book)
WHERE a.Publisher_Name = "Vintage" AND b.Book_Title = "Stranger" AND c.Book_Title = "Fall"
CREATE (a)-[ra1:published]->(b), (a)-[ra2:published]->(c),(b)-[rm1:published_by]->(a),(c)-[rm2:published_by]->
(a)
RETURN a,b,c
```



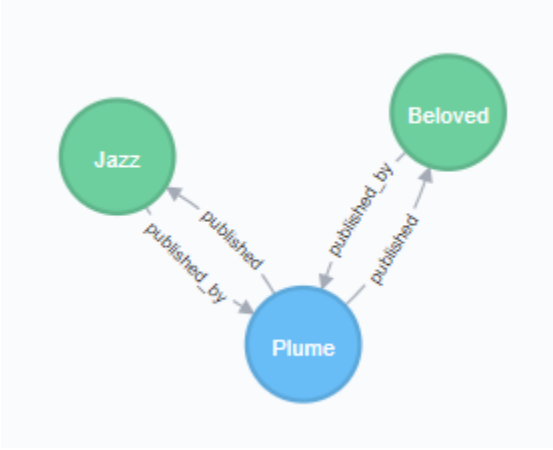
   - **Relationship between publisher Scribner and books**

```
MATCH (a:publisher),(b:book),(c:book)
WHERE a.Publisher_Name= "Scribner" AND b.Book_Title= "Dreamcatcher" AND c.Book_Title = "Catch 22"
CREATE (a)-[ra1:published]->(b), (a)-[ra2:published]->(c),(b)-[rm1:published_by]->(a),(c)-[rm2:published_by]->
(a)
RETURN a,b,c
```



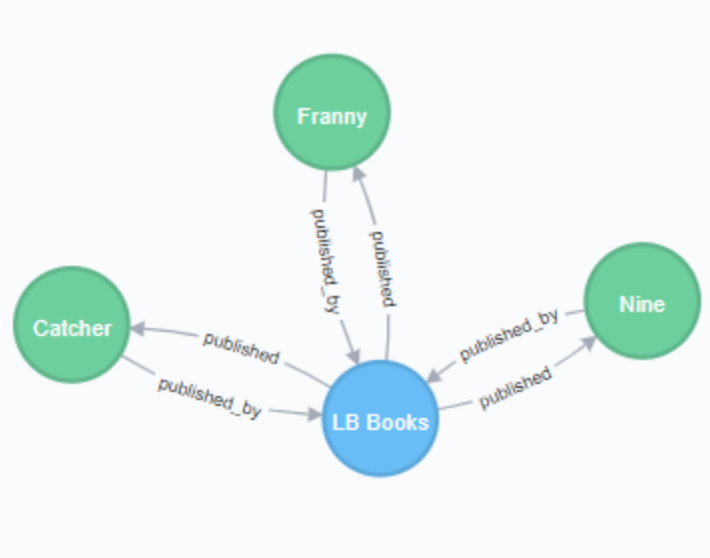
- **Relationship between publisher Plume and books**

```
MATCH (a:publisher),(b:book),(c:book)
WHERE a.Publisher_Name= "Plume" AND b.Book_Title = "Beloved" AND c.Book_Title = "Jazz"
CREATE (a)-[ra1:published]->(b), (a)-[ra2:published]->(c),(b)-[rm1:published_by]->(a),(c)-[rm2:published_by]->
(a)
RETURN a,b,c
```



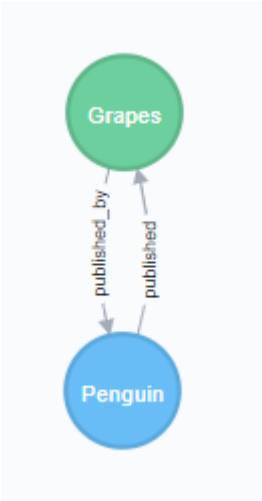
- **Relationship between publisher LB Books and books**

```
MATCH (a:publisher),(b:book),(c:book),(d:book)
WHERE a.Publisher_Name = "LB Books" AND b.Book_Title = "Nine" AND c.Book_Title = "Franny" AND d.Book_Title =
"Catcher"
CREATE (a)-[ra1:published]->(b), (a)-[ra2:published]->(c), (a)-[ra3:published]->(d),(b)-[rm1:published_by]->
(a),(c)-[rm2:published_by]->(a),(d)-[rm3:published_by]->(a)
RETURN a,b,c,d
```



- **Relationship between publisher Penguin and books**

```
MATCH (a:publisher),(b:book)
WHERE a.Publisher_Name = "Penguin" AND b.Book_Title = "Grapes"
CREATE (a)-[ra1:published]->(b),(b)-[rm1:published_by]->(a)
RETURN a,b
```



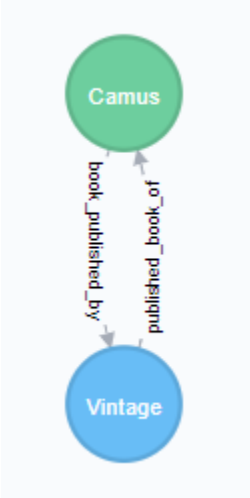
5. Graphically represent the relationship between publisher and author – use word or any other graphic tool

Publisher	Author
Vintage	Camus
Scribner	King
Plume	Morrison
LB Books	Slinger

Penguin	Heller

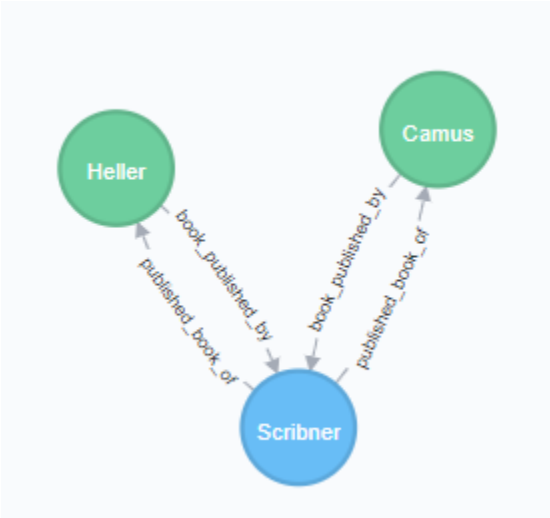
6. Use Neo4j code to create the relationships between publisher and author
   - **Relationship between publisher Vintage and authors**

```
MATCH (a:publisher),(b:author)
WHERE a.Publisher_Name = "Vintage" AND b.name = "Camus"
CREATE (a)-[ra1:published_book_of]->(b),(b)-[rm1:book_published_by]->(a)
RETURN a,b
```



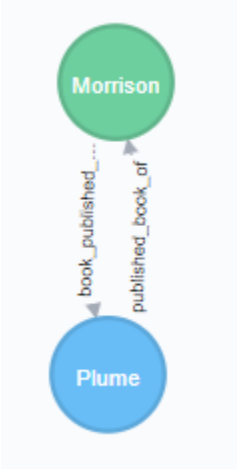
   - **Relationship between publisher Scribner and authors.**

```
MATCH (a:publisher),(b:author),(c:author)
WHERE a.Publisher_Name= "Scribner" AND b.name = "Camus"  AND c.name = "Heller"
CREATE (a)-[ra1:published_book_of]->(b),(a)-[ra2:published_book_of]->(c),(b)-[rm1:book_published_by]->(a),(c)-
[rm2:book_published_by]->(a)
RETURN a,b,c
```



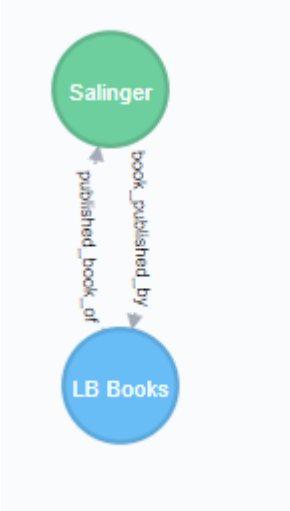
- **Relationship between publisher Plume and authors.**

```
MATCH (a:publisher),(b:author)
WHERE a.Publisher_Name = "Plume" AND b.name = "Morrison"
CREATE (a)-[ra1:published_book_of]->(b), (b)-[rm1:book_published_by]->(a)
RETURN a,b
```



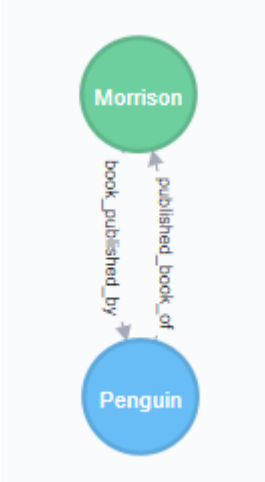
- **Relationship between publisher LB Books and authors.**

```
MATCH (a:publisher),(b:author)
WHERE a.Publisher_Name = "LB Books" AND b.name = "Salinger"
CREATE (a)-[ra1:published_book_of]->(b),(b)-[rm1:book_published_by]->(a)
RETURN a,b
```



- **Relationship between publisher Penguin and authors.**

```
MATCH (a:publisher),(b:author)
WHERE a.Publisher_Name = "Penguin" AND b.name = "Morrison"
CREATE (a)-[ra1:published_book_of]->(b),(b)-[rm1:book_published_by]->(a)
RETURN a,b
```



7. Sort the names in descending order
   - **Books name in descending order.**

```
MATCH (book)
WHERE book.Book_Title IS NOT NULL
RETURN book.Book_Title
ORDER BY book.Book_Title DESC
```

book.Book_Title
"Stranger"
"Nine"
"Jazz"
"Grapes"
"Franny"
"Fall"
"Dreamcatcher"
"Catcher"
"Catch 22"
"Beloved"

   - **Author names in descending order.**

```
MATCH (author)
WHERE author.name IS NOT NULL
RETURN author.name
ORDER BY author.name DESC
```

author.name
"Salinger"
"Morrison"
"King"
"Heller"
"Camus"

- **Publishers names in descending order.**

```
1 MATCH (publisher)
2 WHERE publisher.Publisher_Name IS NOT NULL
3 RETURN publisher.Publisher_Name
4 ORDER BY publisher.Publisher_Name DESC
```

publisher.Publisher_Name
"Vintage"
"Scribner"
"Plume"
"Penguin"
"LB Books"

8. Which publisher employs the author Morrison

```
1 MATCH (x:publisher)-[:published_book_of]->(y:author)
2 WHERE y.name = "Morrison"
3 RETURN x
```

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'm very familiar with Oracle. I have been learning Oracle for last two and half years. In this semester, I learned Mongodb and Neo4j in the data on the web class. At first both were difficult for me but after doing the assignments and final project, I have better understanding in Mongodb and Neo4j. I'm very glad I took the class and learned Mongodb and Neo4j.
2. What can you say about the differences in use among the three systems (Oracle, MongoDB and Neo4j)?
   I don't see much difference between Oracle, Mongodb, and Neo4j. I used oracle for relational database, Mongodb for document stores, and Neo4j for graph stores. But after using all these three in this exercise, we got same types of output. Only difference I can see is the commands are different.
3. Identify why you would use each of these systems:
   a. Oracle: Oracle database is the collection of data. The purpose of oracle database is store data and retrieve related information from the database. Oracle is the most advance partitioning and sub partitioning implementation by far. Way more options for customizing business solutions. The most advance optimizer that can handle every sophisticated way to model data. This includes trivial things like nested sub queries.
   b. MongoDB: MongoDB's document data model maps naturally to objects in application code, making it simple for developers to learn and use. Documents give us the ability to represent hierarchical relationships to store arrays and other more complex structures easily.
   c. Neo4j: Graph databases help to find relationships between data and extract their true value. Many companies have data that are of little use because they are unstructured, and they do not know the relationship between them. Neo4j uses graphs to represent data and the relationships between them. Neo4j uses property graphs to extract added value of data of any company with great performance and in an agile, flexible and scalable way.
4. What different view of the data do you get from using Oracle, MongoDB and Neo4j
   Using Oracle, we get relational database where we can store and retrieve data. Mongodb and Neo4j both are NoSQL databases. Document databases like MongoDB are designed to store documents and retrieve them with lightning speed. Documents are stored in BSON format (Binary JavaScript Object Notation) and are schema less. This allows you to store and retrieve data blocks really fast but does not store relationships. Graph database is that they are used to analyze the nature of the relationships between data. Think of Facebook "Friend Suggestions" is the examples of relationship analysis.

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.

   Any companies will be successful if they use all of these three. In relational database, we normalize data and store in database. We can easily retrieve data from this database. Companies can also use Mongodb and Neo4j to store data. Mongodb use large non-relational database to store data and retrieve data more quickly than relational database and it is schema free and Neo4j use to show the natural relationship between data.