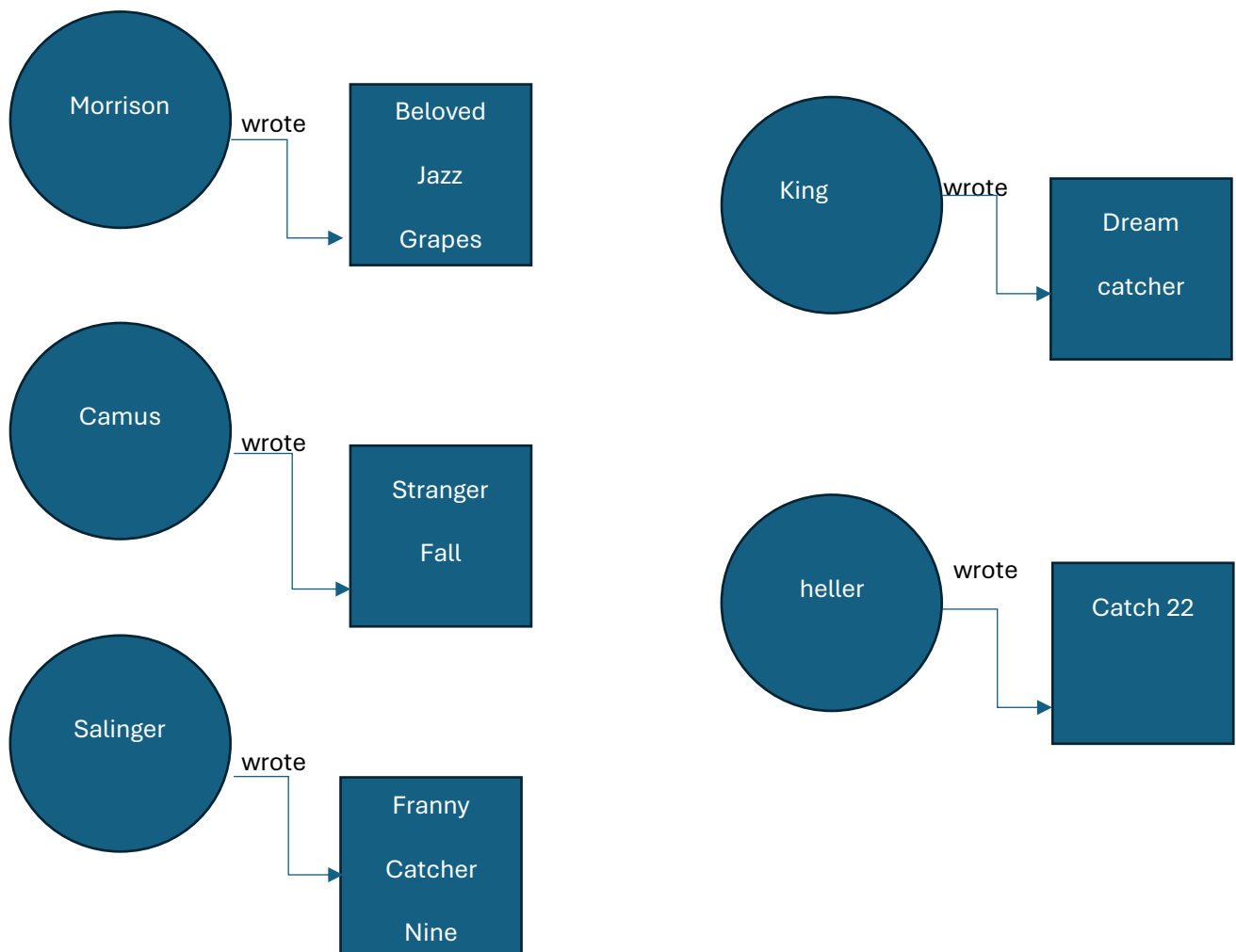


# PART 3

Project 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

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:Author {name: "Camus"}), (b:Book {title: "Stranger"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Camus"}), (b:Book {title: "Fall"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "King"}), (b:Book {title: "Dreamcatcher"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Morrison"}), (b:Book {title: "Beloved"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Morrison"}), (b:Book {title: "Jazz"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Morrison"}), (b:Book {title: "Grapes"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Salinger"}), (b:Book {title: "Nine"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Salinger"}), (b:Book {title: "Franny"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Salinger"}), (b:Book {title: "Catcher"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Heller"}), (b:Book {title: "Catch 22"}) CREATE (a)-[:WROTE]->(b)
```

## 3. Show all the nodes and relationships

```
neo4j$ MATCH (a:Author {name: "Salinger"}), (b:Book {title: "Franny"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Salinger"}), (b:Book {title: "Catcher"}) CREATE (a)-[:WROTE]->(b)
neo4j$ MATCH (a:Author {name: "Heller"}), (b:Book {title: "Catch 22"}) CREATE (a)-[:WROTE]->(b)
```

```
neo4j$ MATCH (n) RETURN n LIMIT 25
```

Overview

**Node labels**

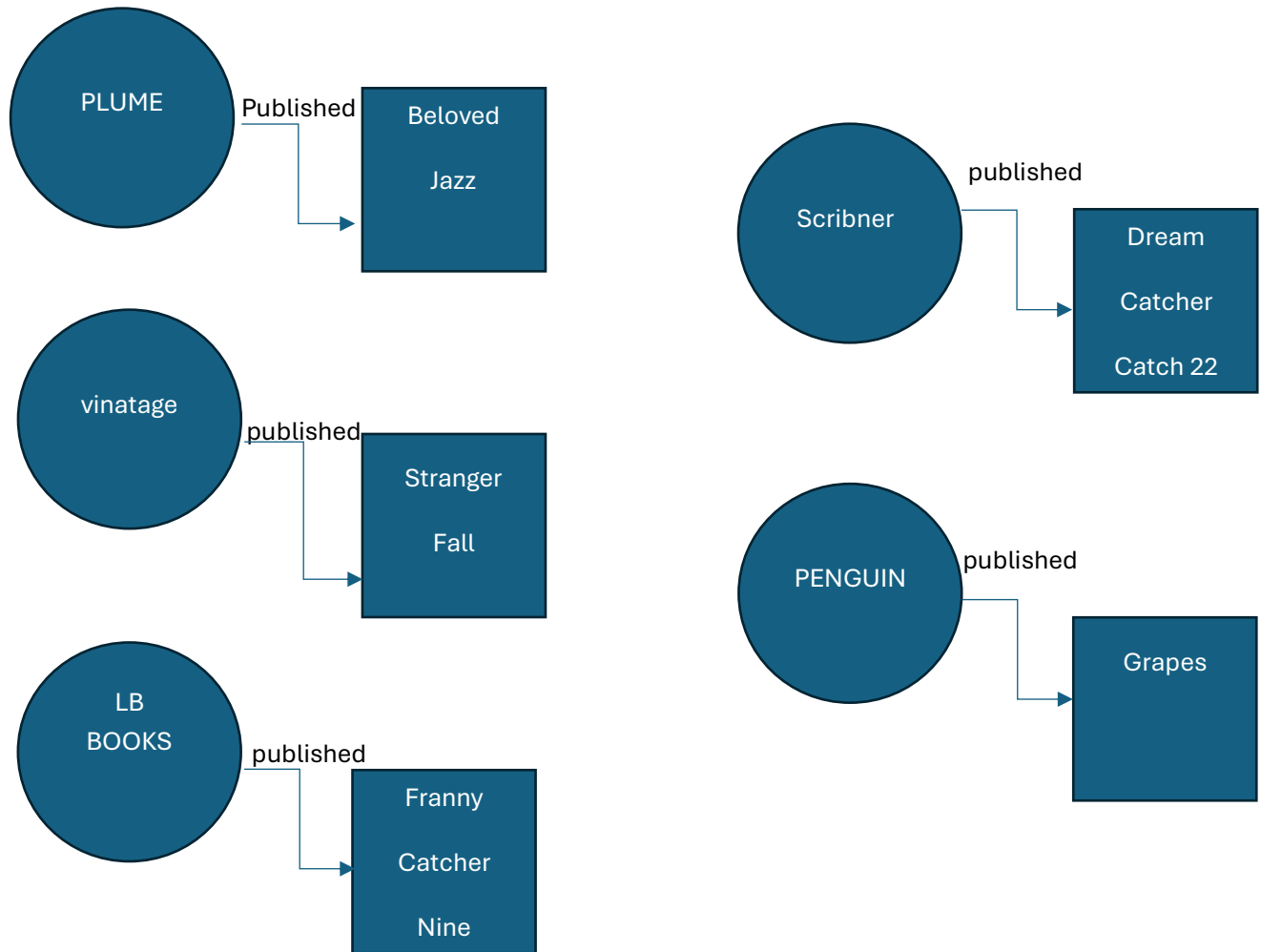
- \* (15)
- Author (5)
- Book (10)

**Relationship types**

- \* (10)
- WROTE (10)

Displaying 15 nodes, 10 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)

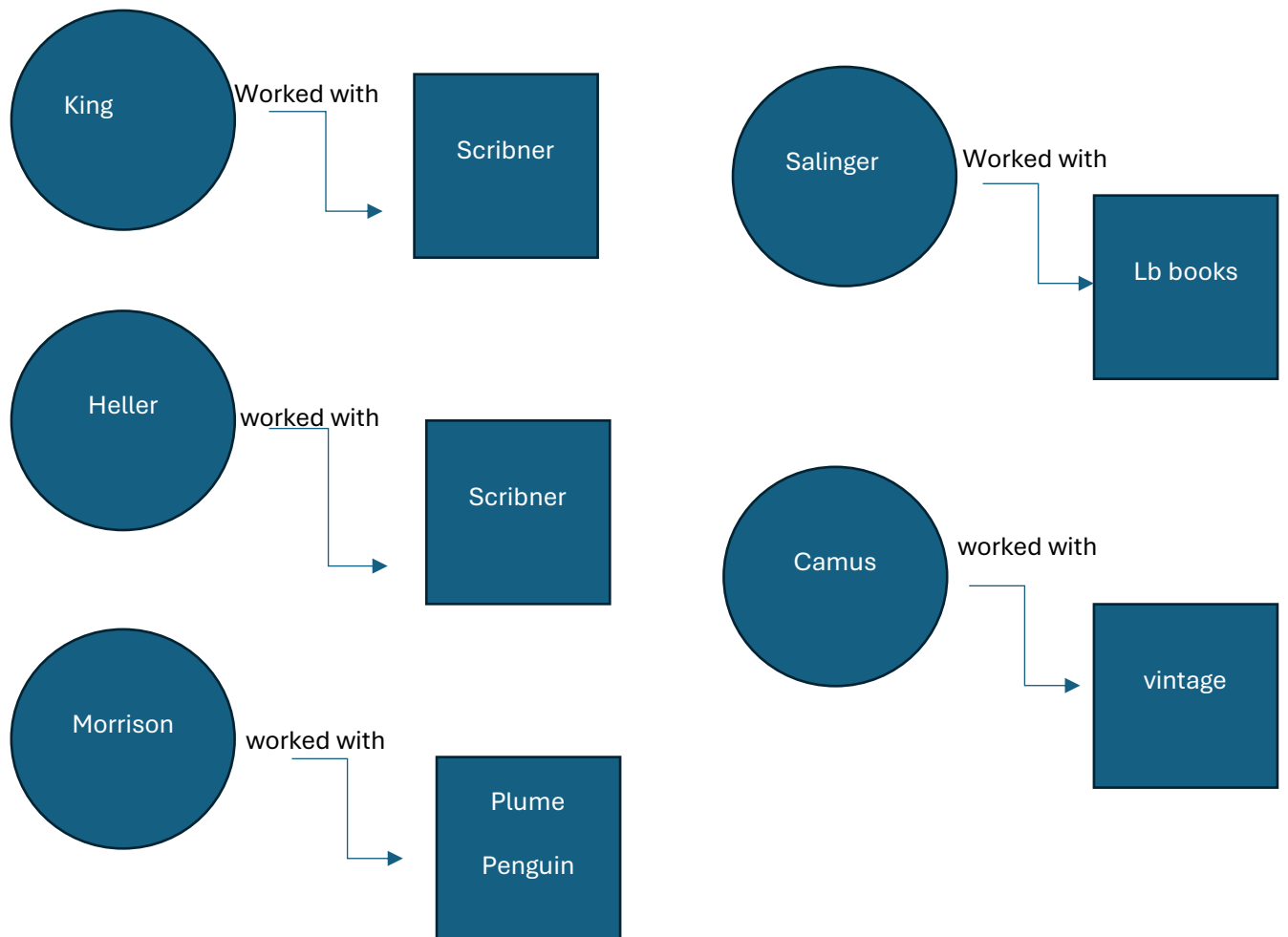
```
CREATE (:Publisher {name: "Vintage"});
CREATE (:Publisher {name: "Scribner"});
CREATE (:Publisher {name: "Plume"});
CREATE (:Publisher {name: "LB Books"});
CREATE (:Publisher {name: "Penguin"});
```

```
1 MATCH (b:Book {title: "Stranger"}), (p:Publisher {name: "Vintage"})
2 CREATE (b)-[:PUBLISHED_BY]-(p);
3
4 MATCH (b:Book {title: "Dreamcatcher"}), (p:Publisher {name: "Scribner"})
5 CREATE (b)-[:PUBLISHED_BY]-(p);
6
7 MATCH (b:Book {title: "Beloved"}), (p:Publisher {name: "Plume"})
8 CREATE (b)-[:PUBLISHED_BY]-(p);
9
10 MATCH (b:Book {title: "Nine"}), (p:Publisher {name: "LB Books"})
11 CREATE (b)-[:PUBLISHED_BY]-(p);
12
```

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

```
MATCH (a:Author {name: "Camus"}), (p:Publisher {name: "Vintage"})
CREATE (a)-[:EMPLOYED_BY]→(p);

MATCH (a:Author {name: "King"}), (p:Publisher {name: "Scribner"})
CREATE (a)-[:EMPLOYED_BY]→(p);

MATCH (a:Author {name: "Morrison"}), (p:Publisher {name: "Plume"})
CREATE (a)-[:EMPLOYED_BY]→(p);

MATCH (a:Author {name: "Salinger"}), (p:Publisher {name: "LB Books"})
CREATE (a)-[:EMPLOYED_BY]→(p);

MATCH (a:Author {name: "Heller"}), (p:Publisher {name: "Scribner"})
CREATE (a)-[:EMPLOYED_BY]→(p);
```

9. Show all the nodes and relationships



## 10. Sort the names in descending order

```
neo4j$ MATCH (a:Author) RETURN a.name AS AuthorName ORDER BY a.name DESC;
```

Table

AuthorName
"Salinger"
"Morrison"
"King"
"Heller"
"Camus"

Text

Code

MAX COLUMN WIDTH:

```
neo4j$ MATCH (n) RETURN n LIMIT 25
```

Graph

{:Book {code: 57,title: "Catch 22"}}
{:Book {code: 61,title: "Jazz"}}

```
1 MATCH (p:Publisher)
2 RETURN p.name AS PublisherName
3 ORDER BY p.name DESC;
4
```

Table

	PublisherName
1	"Vintage"
2	"Scribner"
3	"Plume"
4	"Penguin"
5	"LB Books"

Text

Code



```
neo4j$ MATCH (b:Book) RETURN b.title AS BookTitle ORDER BY b.title DESC;
```

	BookTitle
1	"Stranger"
2	"Nine"
3	"Jazz"
4	"Grapes"
5	"Franny"
6	"Fall"
7	"Dreamcatcher"
8	"Catcher"
9	"Catch 22"
10	"Beloved"

11. Which publisher employs the author Morrison

```
neo4j$ MATCH (a:Author {name: "Morrison"})-[:EMPLOYED_BY]-(p:Publisher) RETURN p.name AS Publ...
```

	Publ...
	Publisher
	"Plume"
	"Penguin"