Islamic University of Technology

CSE-4410
Database Management Systems - II Lab

Lab Report - 9

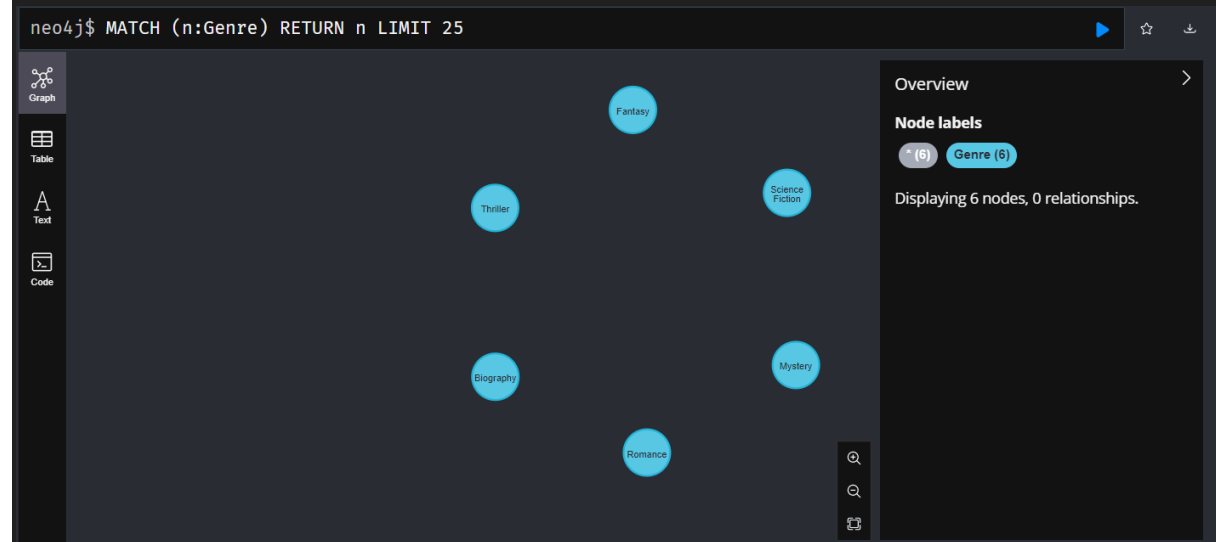**Name: Mukit Mahdin**
ID: 200042170
Department: CSE
Program: SWE

Task-1 :

Create necessary nodes and relations with properties.

```
CREATE (:Genre {name: 'Romance'})
CREATE (:Genre {name: 'Mystery'})
CREATE (:Genre {name: 'Science Fiction'})
CREATE (:Genre {name: 'Fantasy'})
CREATE (:Genre {name: 'Thriller'})
CREATE (:Genre {name: 'Biography'})
```

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

Graph
Table
Text
Code

Fantasy
Science Fiction
Thriller
Biography
Mystery
Romance

Overview

Node labels

* (6)   Genre (6)

Displaying 6 nodes, 0 relationships.

```
CREATE (:Author {name: 'Agatha Christie', country: 'UK', date_of_birth: '15
September 1890'})
CREATE (:Author {name: 'J.K. Rowling', country: 'UK', date_of_birth: '31 July
1965'})
CREATE (:Author {name: 'George Orwell', country: 'UK', date_of_birth: '25 June
1903'})
```

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

Graph
Table
Text
Code

George Orwell
J.K. Rowling
Agatha Christie

Overview

Node labels

* (3)   Author (3)

Displaying 3 nodes, 0 relationships.

```
CREATE (:Book {title: 'Harry Potter and the Philosopher\'s Stone',
published_year: 1997, language: 'English', page_count: 223, price: 10.99})
CREATE (:Book {title: '1984', published_year: 1949, language: 'English',
page_count: 328, price: 8.99})
CREATE (:Book {title: 'Murder on the Orient Express', published_year: 1934,
language: 'English', page_count: 347, price: 9.99})
CREATE (:Book {title: 'The Murder of Roger Ackroyd', published_year: 1926,
language: 'English', page_count: 299, price: 7.99})
CREATE (:Book {title: 'Animal Farm', published_year: 1945, language: 'English',
page_count: 112, price: 6.99})
CREATE (:Book {title: 'The Hobbit', published_year: 1937, language: 'English',
page_count: 310, price: 12.99})
CREATE (:Book {title: 'Becoming', published_year: 2018, language: 'English',
page_count: 448, price: 14.99})
CREATE (:Book {title: 'Sapiens: A Brief History of Humankind', published_year:
2011, language: 'English', page_count: 443, price: 11.99})
```

neo4j$ MATCH (n:Book) RETURN n LIMIT 25

Overview

Node labels

* (8)   Book (8)

Displaying 8 nodes, 0 relationships.

```
//Customer nodes
CREATE (:Customer {customer_id: 'C001', name: 'John Smith', phone_no:
'555-1234', age: 30, gender: 'Male', country: 'USA'})
CREATE (:Customer {customer_id: 'C002', name: 'Jane Doe', phone_no: '555-5678',
age: 25, gender: 'Female', country: 'Canada'})
CREATE (:Customer {customer_id: 'C003', name: 'Bob Williams', phone_no:
'555-9012', age: 45, gender: 'Male', country: 'UK'})
```

Graph

Table

Text

Code

Overview >

Node labels

* (3)   Customer (3)

Displaying 3 nodes, 0 relationships.

Bob Williams

Jane Doe

John Smith

```
//Relationship between customers and books
MATCH (c:Customer {customer_id: 'C001'}), (b:Book {title: 'Harry Potter and the
Philosopher\'s Stone'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-01', amount: 10.99}]→(b)


MATCH (c:Customer {customer_id: 'C001'}), (b:Book {title: '1984'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-01', amount: 8.99}]→(b)


MATCH (c:Customer {customer_id: 'C002'}), (b:Book {title: 'Murder on the Orient
Express'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-02', amount: 9.99}]→(b)


MATCH (c:Customer {customer_id: 'C002'}), (b:Book {title: 'The Murder of Roger
Ackroyd'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-03', amount: 7.99}]→(b)


MATCH (c:Customer {customer_id: 'C003'}), (b:Book {title: 'Animal Farm'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-04', amount: 6.99}]→(b)


MATCH (c:Customer {customer_id: 'C003'}), (b:Book {title: 'The Hobbit'})
CREATE (c)-[:PURCHASED {purchasing_date: '2022-01-04', amount: 12.99}]→(b)
```
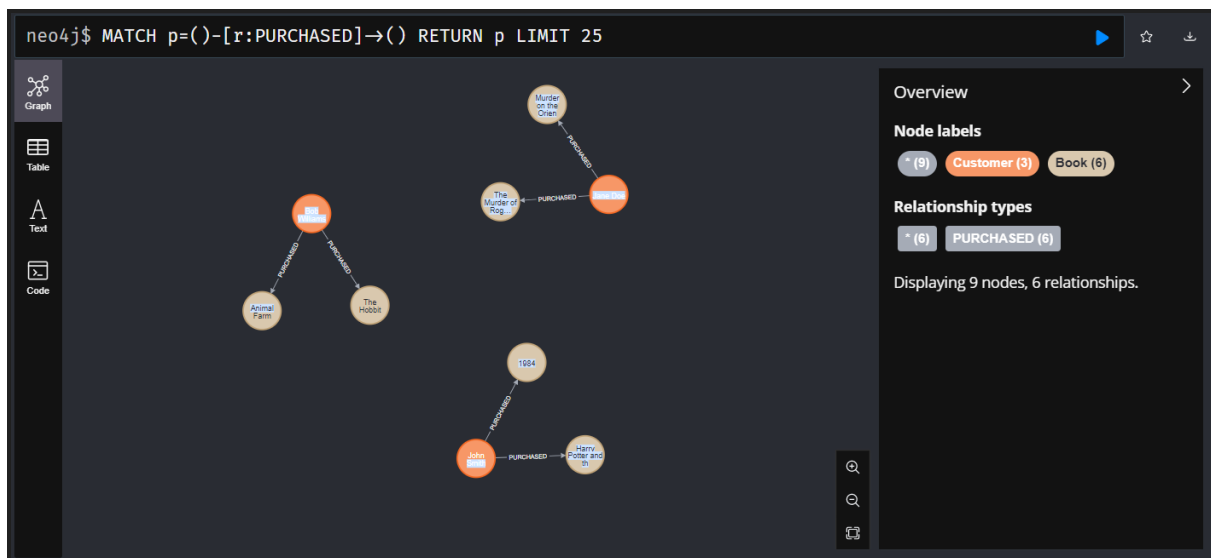
Graph

Table

Text

Code

Overview

Node labels

* (9)   Customer (3)   Book (6)

Relationship types

* (6)   PURCHASED (6)

Displaying 9 nodes, 6 relationships.

```
//Books and Genres
MATCH (b:Book {title: '1984'}), (g:Genre {name: 'Science Fiction'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'Murder on the Orient Express'}), (g:Genre {name:
'Mystery'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'The Murder of Roger Ackroyd'}), (g:Genre {name:
'Mystery'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'Animal Farm'}), (g:Genre {name: 'Politics'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'The Hobbit'}), (g:Genre {name: 'Fantasy'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'Becoming'}), (g:Genre {name: 'Biography'})
CREATE (b)-[:BELONGS_TO]→(g)

MATCH (b:Book {title: 'Sapiens: A Brief History of Humankind'}), (g:Genre
{name: 'Science'})
CREATE (b)-[:BELONGS_TO]→(g)
```
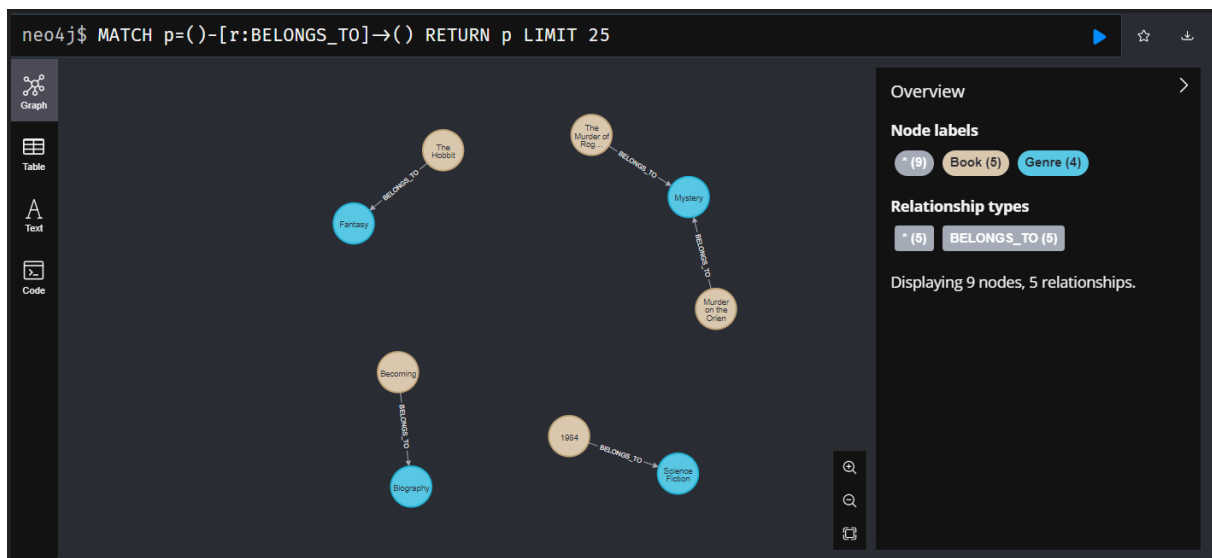
```
neo4j$ MATCH p=()-[r:BELONGS_TO]→() RETURN p LIMIT 25
```


Overview >

Node labels
* (9)   Book (5)   Genre (4)

Relationship types
* (5)   BELONGS_TO (5)

Displaying 9 nodes, 5 relationships.

```
MATCH (b:Book {title: 'Murder on the Orient Express'}), (a:Author {name:
'Agatha Christie'})
CREATE (a)-[:WROTE {writing_year: 1934}]→(b)

MATCH (b:Book {title: 'The Murder of Roger Ackroyd'}), (a:Author {name: 'Agatha
Christie'})
CREATE (a)-[:WROTE {writing_year: 1926}]→(b)

MATCH (b:Book {title: 'Animal Farm'}), (a:Author {name: 'George Orwell'})
CREATE (a)-[:WROTE {writing_year: 1945}]→(b)

MATCH (b:Book {title: '1984'}), (a:Author {name: 'George Orwell'})
CREATE (a)-[:WROTE {writing_year: 1949}]→(b)
```
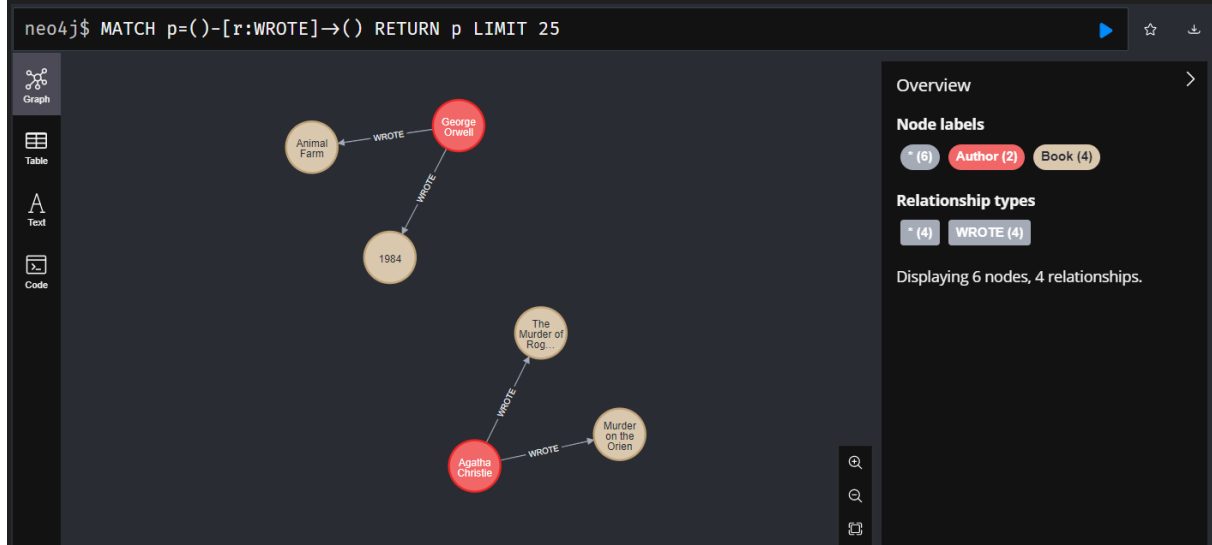
```
neo4j$ MATCH p=()-[r:WROTE]→() RETURN p LIMIT 25
```


Overview >

Node labels
* (6)   Author (2)   Book (4)

Relationship types
* (4)   WROTE (4)

Displaying 6 nodes, 4 relationships.

Task-2. Cypher Queries:

(a) Find the total revenue generated by each book.

```
MATCH (b:Book)-[p:PURCHASED]→()
RETURN b.title, SUM(p.amount) AS total_revenue
```

(b) Find the average rating for each genre.

```
MATCH (g:Genre)←[:BELONGS_TO]-(b:Book)-[r:RATED]→()
RETURN g.name, AVG(r.rating) AS average_rating
```

(c) Find books purchased by a customer 'N' within a specific time range.

```
MATCH (c:Customer)-[p:PURCHASED]→(b:Book)
WHERE c.name = 'N' AND p.purchasing_date ≥ '2022-01-01' AND
p.purchasing_date ≤ '2022-12-31'
RETURN b.title
```

(d) Find the customer who buys the maximum number of books.

```
MATCH (c:Customer)-[p:PURCHASED]→()
RETURN c.name, COUNT(p) AS total_purchases
ORDER BY total_purchases DESC
LIMIT 1
```

(e) Find the best-seller books by the number of purchases.

```
MATCH (b:Book)-[p:PURCHASED]→()
RETURN b.title, COUNT(p) AS total_purchases
ORDER BY total_purchases DESC
```

(f) Find the customer who bought or rated a certain book. for example 'A'

```
MATCH (c:Customer)-[:PURCHASED|:RATED]→(b:Book {title: 'The Hobbit'})
RETURN c.name
```

(g) Find the customer who bought the books of a certain author. for example 'X'

```
MATCH (c:Customer)-[:PURCHASED]→(b:Book)←[:WROTE]-(a:Author {name:
'Agatha Christie'})
RETURN c.name
```

(h) Find books frequently purchased together.

```cypher
MATCH (b1:Book)←[p1:PURCHASED]-(c:Customer)-[p2:PURCHASED]→(b2:Book)
WHERE b1 <> b2
WITH b1, b2, COUNT(DISTINCT c) AS num_customers
RETURN b1.title, b2.title, num_customers
ORDER BY num_customers DESC
```