



**Department of Computer Science and Engineering**  
**Islamic University of Technology (IUT)**  
A subsidiary organ of OIC

**Laboratory Report**

**CSE 4410: DATABASE MANAGEMENT SYSTEMS II LAB**

|                           |                                 |
|---------------------------|---------------------------------|
| <b>Name</b>               | <b>:Nazia Karim Khan Oishee</b> |
| <b>Student ID</b>         | <b>:200042137</b>               |
| <b>Section</b>            | <b>:1A</b>                      |
| <b>Semester</b>           | <b>:Summer</b>                  |
| <b>Academic Year</b>      | <b>:2021-2022</b>               |
| <b>Date of Submission</b> | <b>:10.04.2023</b>              |
| <b>Lab No</b>             | <b>:09</b>                      |

### Task 1:

Create necessary nodes and relations with properties.

```
CREATE (:Customer {customer_id: '1', name: 'Nazia', phone_no:
'1234567890', age: 20, gender: 'female', country: 'BD'})
CREATE (:Customer {customer_id: '2', name: 'Oishee', phone_no:
'1234567890', age: 20, gender: 'female', country: 'BD'})
CREATE (:Customer {customer_id: '3', name: 'Arpa', phone_no:
'1234567890', age: 20, gender: 'female', country: 'BD'})

CREATE (:Genre {genre_id: 'GENRE1', name: 'Fiction'})
CREATE (:Genre {genre_id: 'GENRE2', name: 'Non-fiction'})
CREATE (:Genre {genre_id: 'GENRE3', name: 'Science-fiction'})

CREATE (:Author {author_id: 'AUTHOR1', name: 'Humayen Ahmed',
country: 'BD', date_of_birth: date('1965-07-31')})
CREATE (:Author {author_id: 'AUTHOR2', name: 'J.K. Rowling', country:
'USA', date_of_birth: date('1965-09-21')})
CREATE (:Author {author_id: 'AUTHOR3', name: 'Jafor Iqbal', country:
'BD', date_of_birth: date('1965-07-31')})
CREATE (:Author {author_id: 'AUTHOR4', name: 'Anisul Haque', country:
'USA', date_of_birth: date('1965-09-21')})

CREATE (:Book {book_id: 'BOOK1', title: 'Harry Potter and the
Philosopher\'s Stone', published_year: 1997, language: 'English',
page_count: 223, price: 12.99, volume: 1})
CREATE (:Book {book_id: 'BOOK2', title: 'Nondito Norok',
published_year: 1977, language: 'English', page_count: 447, price:
10.99, volume: 1})
CREATE (:Book {book_id: 'BOOK3', title: 'XYZ', published_year: 1977,
language: 'English', page_count: 447, price: 10.99, volume: 1})
CREATE (:Book {book_id: 'BOOK4', title: 'XYZ', published_year: 1977,
language: 'English', page_count: 447, price: 10.99, volume: 2})

// customer and book
MATCH (c:Customer {customer_id: '1'}), (b:Book {book_id: 'BOOK1'})
```

```
CREATE (c)-[:PURCHASED_BY {purchasing_date: date('2022-01-01'),  
amount: 12.99}]->(b)  
MATCH (c:Customer {customer_id: '2'}), (b:Book {book_id: 'BOOK2'})  
CREATE (c)-[:PURCHASED_BY {purchasing_date: date('2022-02-15'),  
amount: 10.99}]->(b)  
MATCH (c:Customer {customer_id: '3'}), (b:Book {book_id: 'BOOK3'})  
CREATE (c)-[:PURCHASED_BY {purchasing_date: date('2022-02-15'),  
amount: 10.99}]->(b)  
MATCH (c:Customer {customer_id: '3'}), (b:Book {book_id: 'BOOK4'})  
CREATE (c)-[:PURCHASED_BY {purchasing_date: date('2022-02-15'),  
amount: 10.99}]->(b)
```

*// customer and author*

```
MATCH (c:Customer {customer_id: '1'}), (a:Author {author_id:  
'AUTHOR1'})  
CREATE (c)-[:RATED_BY {rating: 4}]->(b)  
MATCH (c:Customer {customer_id: '2'}), (a:Author {author_id:  
'AUTHOR2'})  
CREATE (c)-[:RATED_BY {rating: 4}]->(b)  
MATCH (c:Customer {customer_id: '3'}), (a:Author {author_id:  
'AUTHOR3'})  
CREATE (c)-[:RATED_BY {rating: 4}]->(b)
```

*// book and genre*

```
MATCH (b:Book {book_id: 'BOOK1'}), (g:Genre {genre_id: 'GENRE1'})  
CREATE (b)-[:BELONGS_TO]->(g)  
  
MATCH (b:Book {book_id: 'BOOK2'}), (g:Genre {genre_id: 'GENRE2'})  
CREATE (b)-[:BELONGS_TO]->(g)
```

*//Books and volume*

```
MATCH (b:Book {book_id: 'BOOK3'}), (b2:Book {book_id: 'BOOK4'})  
CREATE (b)-[:Writes]->(b2)
```

*//Books and Author*

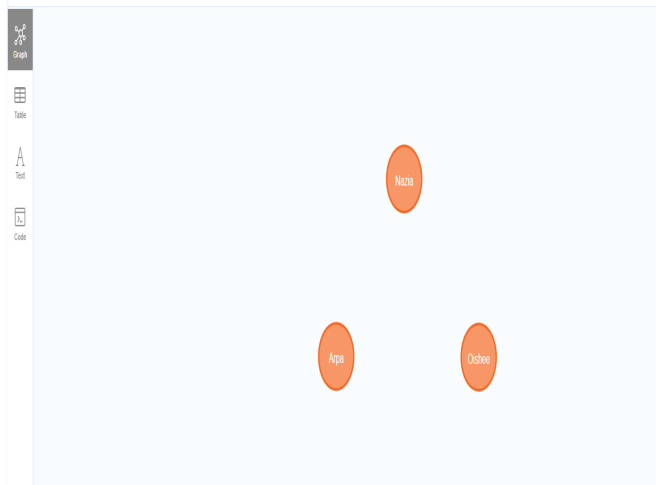
```

MATCH (b:Book {book_id: 'BOOK1'}), (a:Author {author_id: 'AUTHOR2'})
CREATE (b)-[:Writes{wiring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK2'}), (a:Author {author_id: 'AUTHOR1'})
CREATE (b)-[:Writes{wiring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK3'}), (a:Author {author_id: 'AUTHOR3'})
CREATE (b)-[:Writes{wiring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK4'}), (a:Author {author_id: 'AUTHOR4'})
CREATE (b)-[:Writes{wiring_year:1990}]->(a)

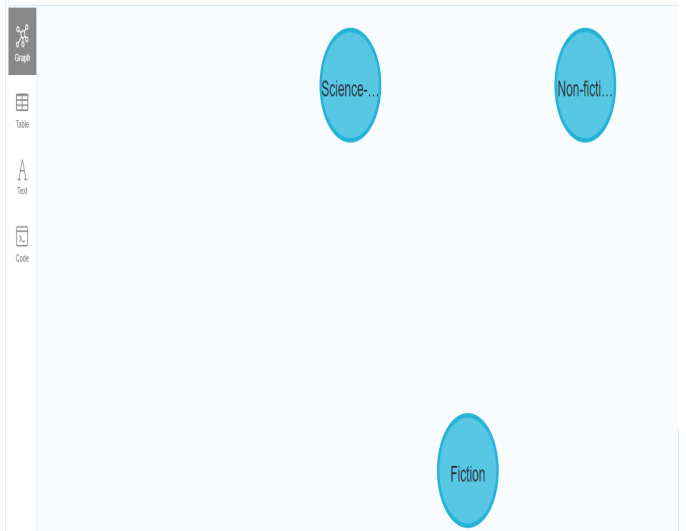
```

### Image:

```
neo4j$ match(Customer) return Customer
```



```
neo4j$ MATCH (g:Genre) RETURN g
```





## 2. Cypher Query

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

```
MATCH (c:Customer)-[p:PURCHASED]->(b:Book)
RETURN b.title, SUM(p.amount) AS revenue
```

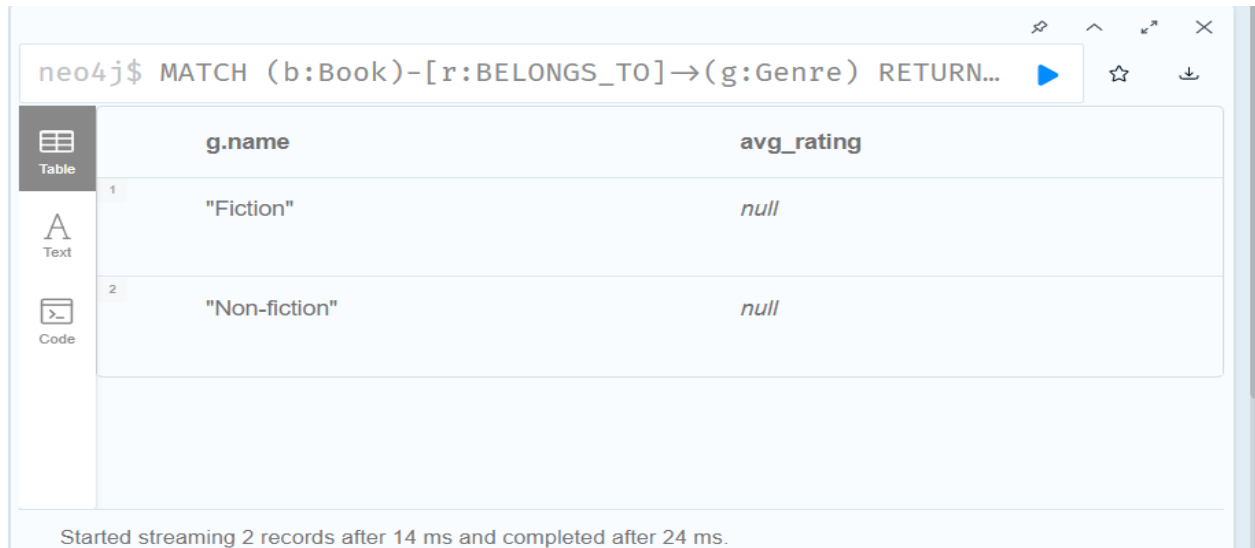
neo4j\$ MATCH (c:Customer)-[p:PURCHASED\_BY]→(b:Book) RETURN b.title, SUM(p.amount) AS revenue

|   | b.title                                    | revenue |
|---|--|---------|
| 1 | "Harry Potter and the Philosopher's Stone" | 12.99   |
| 2 | "Nondito Norok"                            | 10.99   |
| 3 | "XYZ"                                      | 21.98   |

Started streaming 3 records after 13 ms and completed after 19 ms.

**(b) Find the average rating for each genre.**

```
MATCH (b:Book)-[r:BELONGS_TO]->(g:Genre)
RETURN g.name, AVG(b.rating) AS avg_rating
```



|   | g.name        | avg_rating  |
|---|---------------|-------------|
| 1 | "Fiction"     | <i>null</i> |
| 2 | "Non-fiction" | <i>null</i> |

Started streaming 2 records after 14 ms and completed after 24 ms.

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

```
MATCH (c:Customer {customer_id: '1'})-[p:PURCHASED_BY]->(b:Book)
WHERE p.purchasing_date >= date('2022-01-01') AND p.purchasing_date
<= date('2022-01-10')
RETURN b.title,c.name
```

```

1 MATCH (c:Customer {customer_id: '1'})-
  [p:PURCHASED_BY]→(b:Book)
2 WHERE p.purchasing_date ≥ date('2022-01-01') AND
  p.purchasing_date ≤ date('2022-01-10')
3 RETURN b.title,c.name
4

```

|   | b.title                                    | c.name  |
|---|--|---------|
| 1 | "Harry Potter and the Philosopher's Stone" | "Nazia" |

Started streaming 1 records after 2 ms and completed after 4 ms.

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

```

MATCH (c:Customer)-[p:PURCHASED_BY]→(b:Book)
WITH c.customer_id AS customer_id, c.name AS name, COUNT(p) AS
book_count
ORDER BY book_count DESC
LIMIT 1
RETURN customer_id,name

```

```

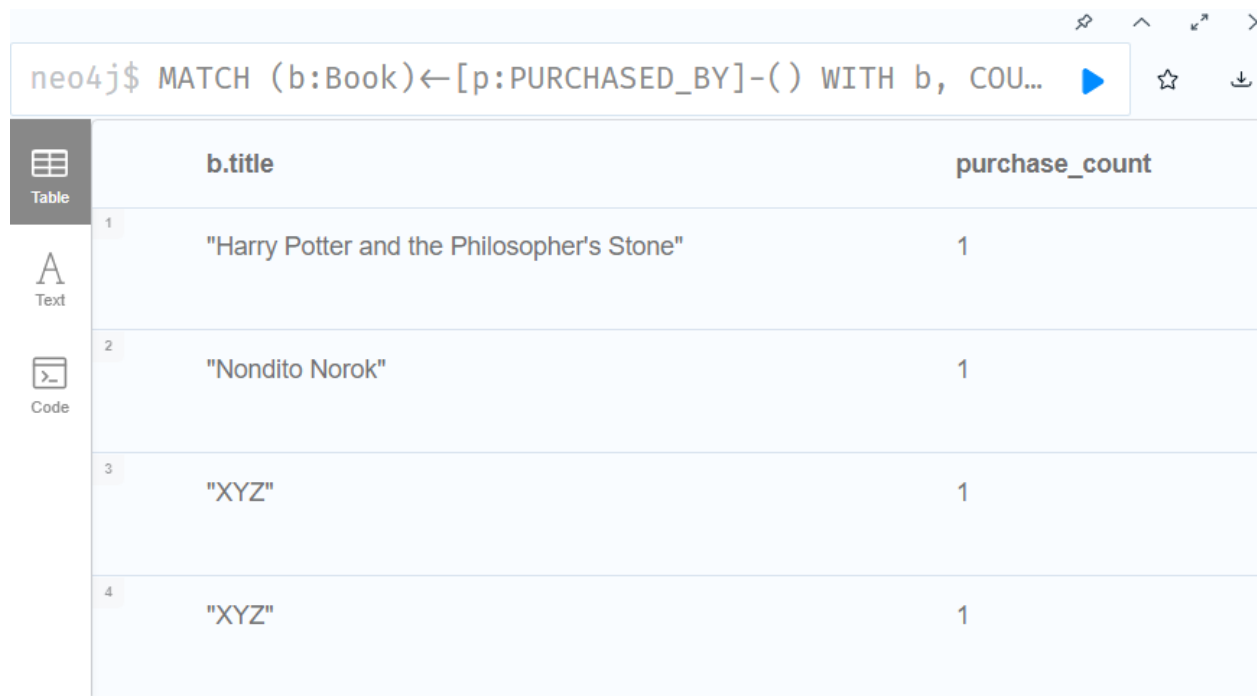
1 MATCH (c:Customer)-[p:PURCHASED_BY]→(b:Book)
2 WITH c.customer_id AS customer_id, c.name AS name,
  COUNT(p) AS book_count
3 ORDER BY book_count DESC
4 LIMIT 1
5 RETURN customer_id,name
6

```

|   | customer_id | name   |
|---|-------------|--------|
| 1 | "3"         | "Arpa" |

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

```
MATCH (b:Book)←-[p:PURCHASED_BY]-(  
WITH b, COUNT(p) AS purchase_count  
ORDER BY purchase_count DESC  
RETURN b.title, purchase_count
```



The image shows a screenshot of the Neo4j query interface. At the top, a query bar contains the Cypher query: `neo4j$ MATCH (b:Book)←-[p:PURCHASED_BY]-( ) WITH b, COU...`. Below the query bar, there are three tabs: 'Table' (selected), 'Text', and 'Code'. The 'Table' tab displays the results of the query in a table with two columns: 'b.title' and 'purchase\_count'. The results are as follows:

|   | b.title                                    | purchase_count |
|---|--|----------------|
| 1 | "Harry Potter and the Philosopher's Stone" | 1              |
| 2 | "Nondito Norok"                            | 1              |
| 3 | "XYZ"                                      | 1              |
| 4 | "XYZ"                                      | 1              |

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

```
MATCH (c:Customer)-[pr:PURCHASED_BY|RATED_BY]->(b:Book {title:  
'Nondito Norok'})  
RETURN c.name
```



|   |   |
|---|---|
| 1 | MATCH (c:Customer)-[pr:PURCHASED_BY RATED_BY]→<br>(b:Book {title: 'Nondito Norok'}) |
| 2 | RETURN c.name   |
| 3 |   |

| c.name     |
|------------|
| 1 "Oishee" |

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

```
MATCH (c:Customer)-[:PURCHASED_BY]->(b:Book)-[:Writes]->(a:Author
{name: 'J.K. Rowling'})
RETURN c.name, b.title
```

|   |   |
|---|---|
| 1 | MATCH (c:Customer)-[:PURCHASED_BY]→(b:Book)-<br>[:Writes]→(a:Author {name: 'J.K. Rowling'}) |
| 2 | RETURN c.name, b.title  |
| 3 |   |

| c.name    | b.title                                    |
|-----------|--|
| 1 "Nazia" | "Harry Potter and the Philosopher's Stone" |

(h) Find books frequently purchased together.

```
MATCH
(b1:Book)<-[:PURCHASED_BY]-(c:Customer)-[:PURCHASED_BY]->(b2:Book)
WHERE b1 <> b2
WITH b1, b2, COUNT(DISTINCT c) AS num_customers
```

```
ORDER BY num_customers DESC
LIMIT 10
RETURN b1.title AS book1, b2.title AS book2, num_customers
```

neo4j\$ MATCH (b1:Book)←[:PURCHASED\_BY]-(c:Customer)-[:PURCHASED\_BY]→(b2:Book) WHERE b1 <> b2 WITH b1, b2, COUNT(DISTINCT c) ...

|   | book1                                      | book2                                      | num_customers |
|---|--|--|---------------|
| 1 | "Nondito Norok"                            | "Harry Potter and the Philosopher's Stone" | 1             |
| 2 | "XYZ"                                      | "Nondito Norok"                            | 1             |
| 3 | "XYZ"                                      | "XYZ"                                      | 1             |
| 4 | "XYZ"                                      | "XYZ"                                      | 1             |
| 5 | "Harry Potter and the Philosopher's Stone" | "Nondito Norok"                            | 1             |
| 6 | "Nondito Norok"                            | "XYZ"                                      | 1             |

Started streaming 6 records after 27 ms and completed after 50 ms.