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

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

Task 1:

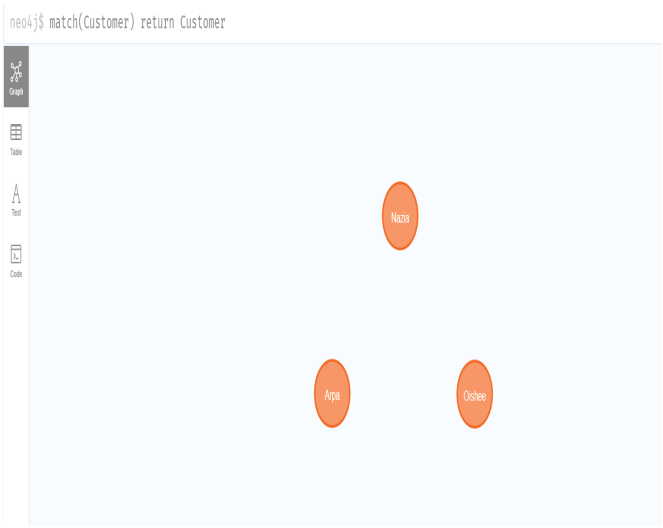
Create necessary nodes and relations with properties.

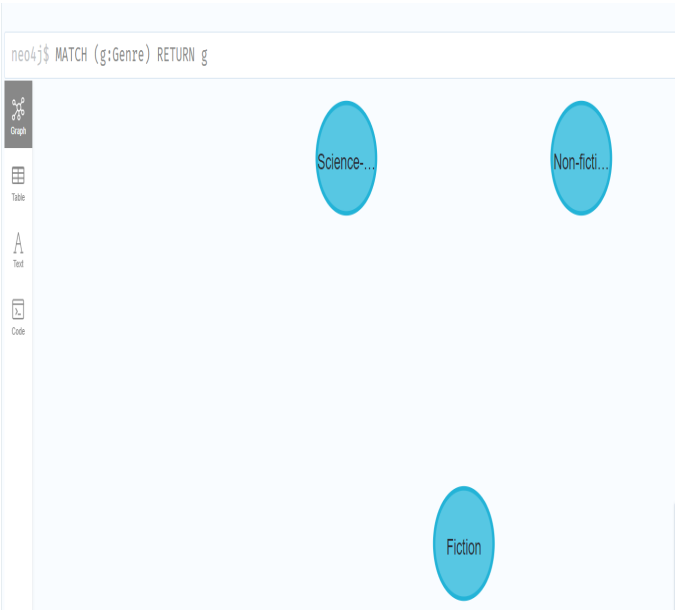
```
CREATE (:Customer {customer_id: '1', name: 'Nazia', phone_no:
'1234567890', age: 20, gender: 'female', country: 'BD'})
CREATE (:Customer {customer_id: '2', name: 'Oishee', phone_no:
'1234567890', age: 20, gender: 'female', country: 'BD'})
CREATE (:Customer {customer_id: '3', name: 'Arpa', phone_no:
'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{wriring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK2'}), (a:Author {author_id: 'AUTHOR1'})
CREATE (b)-[:Writes{wriring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK3'}), (a:Author {author_id: 'AUTHOR3'})
CREATE (b)-[:Writes{wriring_year:1990}]->(a)
MATCH (b:Book {book_id: 'BOOK4'}), (a:Author {author_id: 'AUTHOR4'})
CREATE (b)-[:Writes{wriring_year:1990}]->(a)
```

Image:

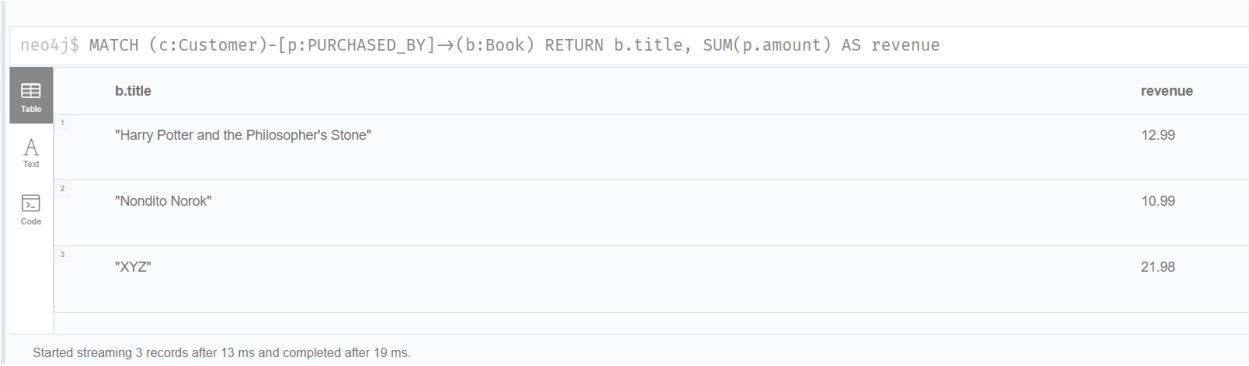




2. Cypher Query

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

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



(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
```

(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
```

```
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
```

	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.

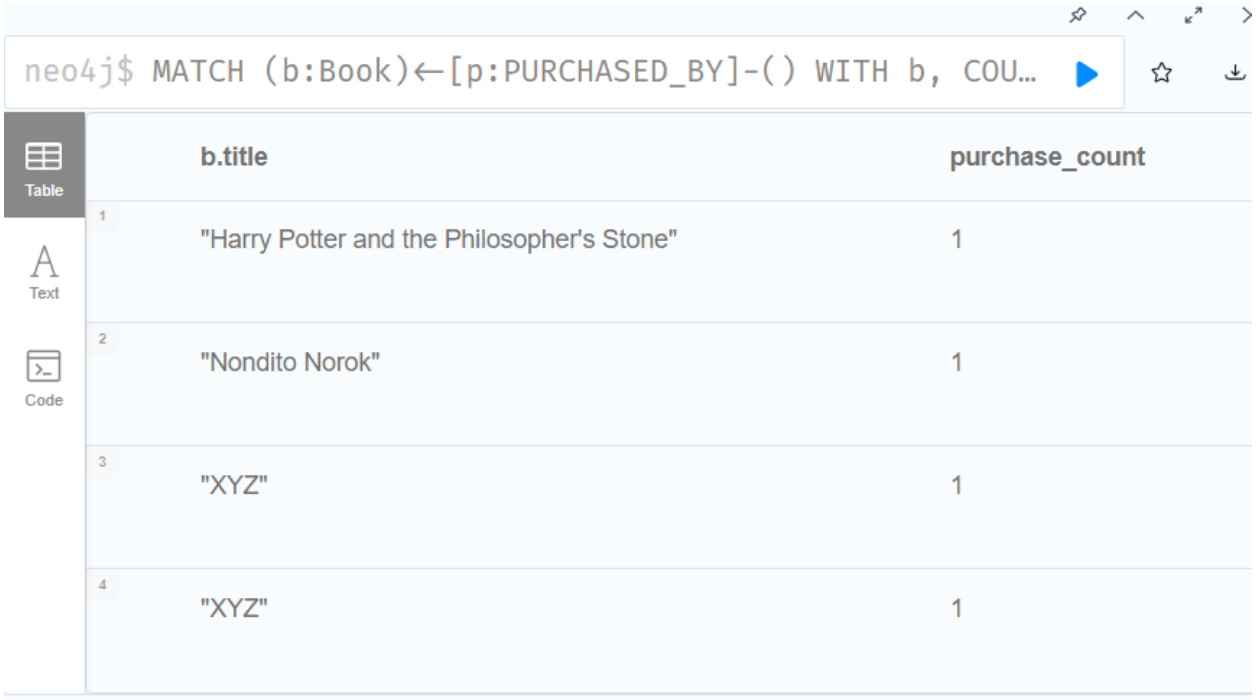
```
MATCH (c:Customer)-[p:PURCHASED_BY]->(b:Book)
WITH c.customer_id AS customer_id, c.name AS name, COUNT(p) AS book_count
ORDER BY book_count DESC
LIMIT 1
RETURN customer_id,name
```

```
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
```

	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
```



	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
```

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

c.name
"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
```

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

c.name	b.title
"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
```