

Ex No 1.**Design own social media application****Date:****Aim :** To Design own social media application**Procedure :****How to Build a Social Network Website****1. Media sharing websites**

When you think about how to build a social network website from scratch, this type of social media platform usually comes to mind, as it is the most popular now. These sites focus on the display and exchange of multimedia content. Platforms like this allow users to share content in the form of photos, videos, audio, etc. Instagram is a great example of this type of social media site, just like YouTube and Flickr.

2. Business and professional sites

These are designed to help establish business connections and create a brand image. LinkedIn and Glassdoor are the most popular social network examples of this type. They provide users with the ability to communicate, create groups by professional interest, exchange professional expertise, and even find new jobs.

3. Informational sites

When looking for an answer to the question “how to create a social media website,” you have probably stumbled upon this type of social media platform. Their main task is to provide information and help people find answers to relevant questions. For example, you can try looking up the answer on Reddit or Quora. These platforms are often created in the form of forums and review sites.

4. Educational websites

Educational platforms are designed to help students communicate with each other as well as to look up answers, get help from teaching staff, conduct research, and much more. If you are

wondering how to make a social media website of this type, look to the Student Room and the Math forum, and other social media platforms for inspiration.

5. Hobby networks

This type of social media platform is quite popular these days, as users can actively search for communities that share their interests. If you are wondering how to build a social network website from scratch that will be of interest to hobbyists, examples are numerous, as sites can be dedicated to almost any hobby that is currently popular.

6. Academic websites

These social media platforms unite scientists and researchers who are willing to share knowledge, present the results of their work, and find out about the most vibrant research. ResearchGate and Academia.edu give an idea of how to build a social network website from scratch that would belong to this category.

7. Social networks

If you want to know how to make a website like Facebook, you should research this type of social networking platform. The main idea behind this model is to allow users to communicate with each other directly, establish personal connections, and unite in groups. As you have probably guessed, Facebook and Twitter are the most remarkable examples.

8. Dating platforms

These networking websites are quite popular today, as they allow users to search for their soulmates. If you are wondering how to build a social network website from scratch in order to have a similar product, you can consult with our company, as we have experience creating these solutions.

How to Make a Social Network Website Step-by-Step



1. Idea and concept

Social network development should start with an original idea that will be conveyed into a concept. It is important to think this part through, as your network should be able to win users over. The following questions will help you understand how to make a social media website that will stand out from the crowd:

What unique features will make your network different from competitors?

What is the main idea behind the website?

What value can you offer to potential users?

Care2, Dogster, and Untappd are examples of niche networks that offer value to users, are unique, and unite users by interests. If you create something like one of these, you have a higher chance that your own social network will become popular.

2. Target audience

As you research an idea for your future product and decide how to build a social network website from scratch in the best way, you should also determine who your target audience is – what potential users might be interested in your product. It is not enough to know their age, locations, financial status, and hobbies; you need to think about the target audience in terms of numbers.

Just imagine that you are looking for an answer regarding how to create a social media website for collectors of water guns. How many people out there that collect them? It does not make sense to invest in something that will not return on your investment.

3. Strategy

By now you probably know a thing or two about how to create a social media website of your choice, but there are a few more important points to mention. It is time to move on to another step that is dedicated to building a business strategy for your website.

The wants and needs of the community you will be targeting should determine how to make a social media website and what features you should offer. What do your users expect from you? How do you want them to use your social network? When you know the answers to these questions, it is not difficult to determine what features to include in social network web development.

You can start from an initial smaller version of your app and later expand your network by adding new features and adopting innovative technology. If your resources allow, you may go bigger and launch a more feature-rich network or a full-fledged product all at once. The less risky strategy is to start small, but it is up to you to decide how to create a social media website.

4. Web development vendor

You cannot decide how to build a platform for social networking and make it successful without choosing the right company providing web development services. Unless you are a developer yourself and are ready to take on the task alone, you will need someone who will be able to provide professional networking web site development services and who knows how to make a social network website with all the features you need. A good option is to opt for IT outsourcing in Eastern Europe due to a great price-quality ratio.

5. Website project

After you have decided on the software development vendor, you can move to the first stage – the discovery stage. During this stage, business analysis is performed to determine functional specifications. The results of the business analysis are used to create the UX/UI social network design, which will later be used to build a network from scratch.

6. Development

This stage is the most time consuming, as it may take upwards of a few months to develop a social network website, even if we are talking about simple forums, and years of continuous

development and iterations if we are talking about complex and custom social networks. However, a company with experienced professionals will be able to provide you with a plan for how to create a social media website in the shortest time possible.

7. Marketing and promotion

Marketing activities are an important part of making a social media website, as without promotion, no one has any way of knowing about a new social network. Promotion can be done in different ways, one of which is advertising through existing social networks like Instagram, Facebook, etc. Other practices include email marketing, SEO practices, blogging, paid ads, etc. As you can clearly see, it is not enough to decide how to build a social network website from scratch, as you also need to keep in mind post-development activities like promotion and advertising.

8. Ongoing maintenance and support

Knowing how to make a social network website is not enough. You also need to know how to organize the ongoing maintenance of your project to guarantee its effective functioning in the future. As a rule, a good web development vendor can consult you about this and provide relevant services as well.

Result

Ex No 2.**Create a Network model using Neo4j****Date:****Aim :** To create Network model using Neo4j**Procedure :**

1. First, you need to install and set up Neo4j. You can download it from the official website and follow the installation instructions.
2. Once Neo4j is installed and running, you can access the Neo4j Browser to interact with the database.
3. In the Neo4j Browser, you can create nodes and relationships using the Cypher query language. Here's how you can create a simple social network:

Neo4j Code

// Create users

```
CREATE (:User {id: 1, name: 'Alice'}),
(:User {id: 2, name: 'Bob'}),
(:User {id: 3, name: 'Charlie'}),
(:User {id: 4, name: 'David'});
```

// Users following each other

```
MATCH (a:User {name: 'Alice'}), (b:User {name: 'Bob'})
CREATE (a)-[:FOLLOWS]->(b);
```

```
MATCH (b:User {name: 'Bob'}), (c:User {name: 'Charlie'})
CREATE (b)-[:FOLLOWS]->(c);
```

```
MATCH (c:User {name: 'Charlie'}), (d:User {name: 'David'})
CREATE (c)-[:FOLLOWS]->(d);
```

// Retrieving users and their followers

```
MATCH (u:User)-[:FOLLOWS]->(f)
RETURN u.name AS User, COLLECT(f.name) AS Followers;
```

Output

```
neo4j$ CREATE (:User {id: 1, name: 'Alice'}), (:User {id: 2, name: 'Bob'}), (:User {i... ✓
neo4j$ MATCH (a:User {name: 'Alice'}), (b:User {name: 'Bob'}) CREATE (a)-[:FOLLOWS]→... ✓
neo4j$ MATCH (b:User {name: 'Bob'}), (c:User {name: 'Charlie'}) CREATE (b)-[:FOLLOWS]... ✓
neo4j$ MATCH (c:User {name: 'Charlie'}), (d:User {name: 'David'}) CREATE (c)-[:FOLLOW... ✓
neo4j$ MATCH (u:User)-[:FOLLOWS]→(f) RETURN u.name AS User, COLLECT(f.name) AS Follo... ✓
```

Visualization Procedure

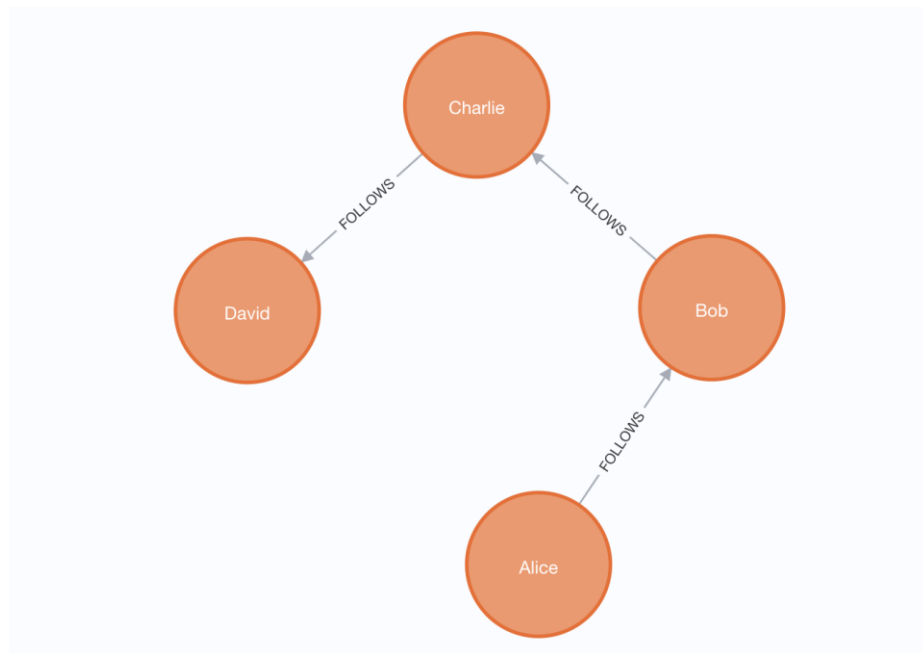
1. Open Neo4j Bloom: Launch Neo4j Bloom from your Neo4j Desktop or web interface.
2. Connect to Your Database: Connect Neo4j Bloom to your Neo4j database.
3. Create Perspective: Create a new perspective or select an existing one.

4. Explore Graph: Once in the perspective, you can explore the graph by clicking on nodes, expanding relationships, and applying layouts to visualize the data in various ways.

Visualization Code

```
MATCH p = ()-[:FOLLOWS]->()  
RETURN p;
```

Output



Result

Ex No 3.**Read and write Data from Graph Database****Date:****Aim :** To Read and write Data from Graph Database**Procedure :****CSV files in the import folder**

Open a finder window on your system by hovering over the three dots to the right started DBMS and selecting **Open folder**, and then **Import**:

Copy or move the three CSV files into this **import** folder on your system:

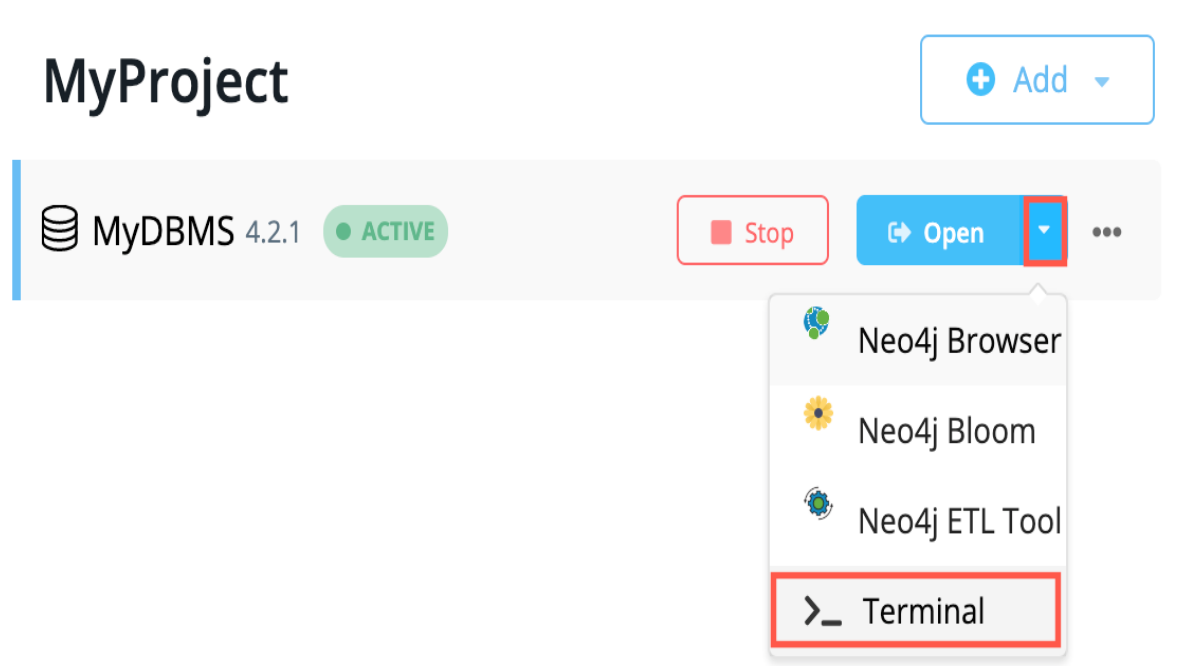
Download file for the below link

<https://s3.amazonaws.com/dev.assets.neo4j.com/wp-content/uploads/desktop-csv-import.zip>

LOAD CSV

To enter and run Cypher statements on a started DBMS, you can:

1. Use Neo4j Browser:
 - a. Click the **Open** button for the started DBMS
 - b. Type or copy Cypher statements into the edit pane.
 - c. Execute the Cypher with the **play** button on the right.
2. Use cypher-shell:
 - a. Click the drop-down menu to the right of the **Open** button and select **Terminal**.



- a. Enter bin/cypher-shell.
- b. Enter neo4j for the user.
- c. Enter the password you specified for the DBMS.
- d. All Cypher statements must end with ";"
- e. Use :exit to quit.

```
//count data rows in products.csv (no headers)
LOAD CSV FROM 'file:///products.csv' AS row
RETURN count(row);
```

```
neo4j$ //count data rows in products.csv (no headers) LOAD CSV FROM 'file:///products.
```

count(row)
78

```
//count data rows in orders.csv (headers)
LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row
RETURN count(row);
```

```
neo4j$ //count data rows in orders.csv (headers) LOAD CSV WITH HEADERS FROM 'file:///or... ▶ ☆ ⬇
```

count(row)
2155

Started streaming 1 records in less than 1 ms and completed after 34 ms.

```
//count data rows in order-details.csv (headers)
LOAD CSV WITH HEADERS FROM 'file:///order-details.csv' AS row
RETURN count(row);
```

```
neo4j$ //count data rows in order-details.csv (headers) LOAD CSV WITH HEADERS FROM 'fil... ▶ ☆ ⬇
```

count(row)
2155

Started streaming 1 records in less than 1 ms and completed after 18 ms.


```
//view data rows in products.csv
LOAD CSV FROM 'file:///products.csv' AS row
RETURN row
LIMIT 3;
```

row	["ProductID", "ProductName", "SupplierID", "CategoryID", "QuantityPerUnit", "UnitPrice", "UnitsInStock", "UnitsOnOrder", "ReorderLevel", "Disconti"]
1	["1", "Chai", "8", "1", "10 boxes x 30 bags", "18", "39", "0", "10", "1"]
2	["2", "Chang", "1", "1", "24 - 12 oz bottles", "19", "17", "40", "25", "1"]

Started streaming 3 records after 1 ms and completed after 40 ms.

```
//count data rows in orders.csv (headers)
LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row
RETURN row
LIMIT 5;
```

row	{ \"ShipPostalCode\": \"51100\", \"ShippedDate\": \"1996-07-16\", \"OrderDate\": \"1996-07-04\", \"Quantity\": \"12\", \"ShipRegion\": null, \"OrderID\": \"10248\", \"Freight\": \"32.38\", \"ShipCity\": \"Reims\", \"RequiredDate\": \"1996-08-01\", \"ProductID\": \"11\", \"UnitPrice\": \"14\", \"ShipCountry\": \"France\", \"EmployeeID\": \"5\", \"ShipVia\": \"3\", \"CustomerID\": \"VINET\", \"\"}
1	

Started streaming 5 records in less than 1 ms and completed after 9 ms.

```
//count data rows in order-details.csv (headers)
LOAD CSV WITH HEADERS FROM 'file:///order-details.csv' AS row
RETURN row
LIMIT 8;
```

neo4j\$ //count data rows in order-details.csv (headers) LOAD CSV WITH HEADERS FROM 'fil...

	row
1	{ "orderId": "10248", "quantity": "12", "productId": "11" }
2	{ "orderId": "10248", "quantity": "10", "productId": "42" }
3	{ "orderId": "10248", "quantity": "5", "productId": "72" }

Started streaming 8 records in less than 1 ms and completed after 6 ms.

```
LOAD CSV FROM 'file:///products.csv' AS row
WITH toInteger(row[0]) AS productId, row[1] AS productName, toFloat(row[2]) AS
unitCost
RETURN productId, productName, unitCost
LIMIT 3;
```

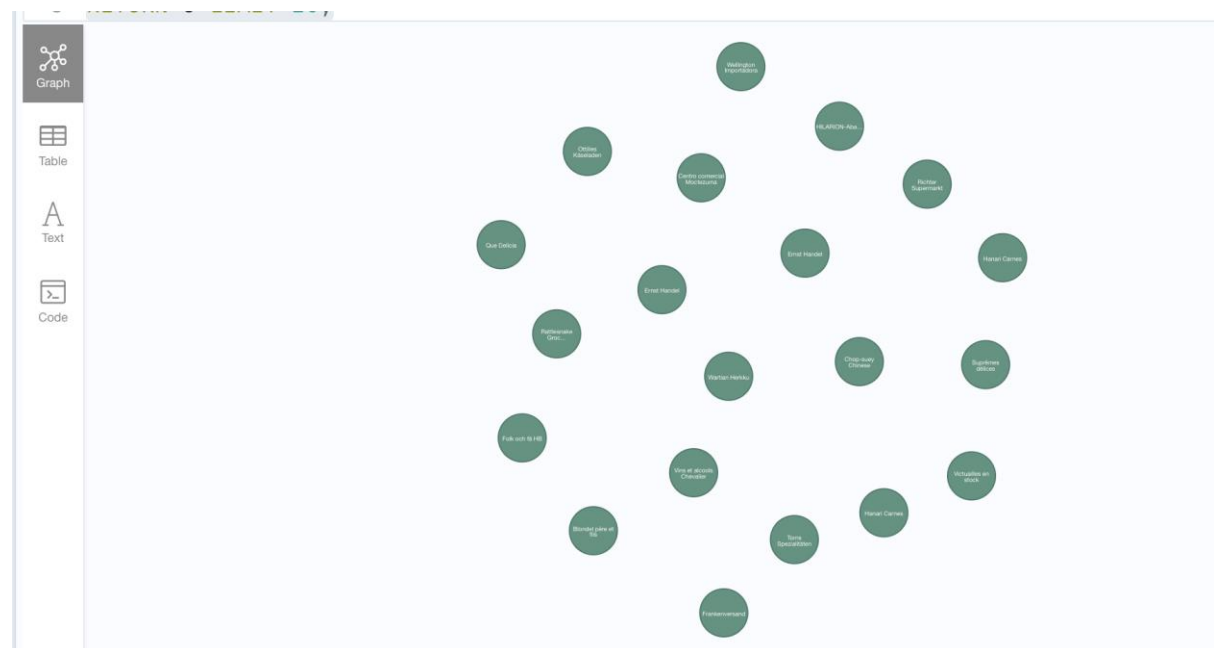
neo4j\$ LOAD CSV FROM 'file:///products.csv' AS row WITH toInteger(row[0]) AS productId,...

	productId	productName	unitCost
1	null	"ProductName"	null
2	1	"Chai"	8.0
3	2	"Chang"	1.0

Started streaming 3 records after 32 ms and completed after 44 ms.

```
LOAD CSV WITH HEADERS FROM 'file:///order-details.csv' AS row
WITH toInteger(row.productId) AS productId, toInteger(row.orderID) AS orderId,
toInteger(row.quantity) AS quantityOrdered
RETURN productId, orderId, quantityOrdered
LIMIT 8;
```

```
//validate orders loaded correctly
MATCH (o:Order)
RETURN o LIMIT 20;
```



Result

Ex No 4.**Find “Friend of Friends” using Neo4j****Date:****Aim :** To Find “Friend of Friends” using Neo4j**Procedure :****Dataset Link**
<https://gist.github.com/jexp/cdaca7e32d8fca630016#file-people-csv>

Jim	Mike
Jim	Billy
Anna	Jim
Anna	Mike
Sally	Anna
Joe	Sally
Joe	Bob
Bob	Sally

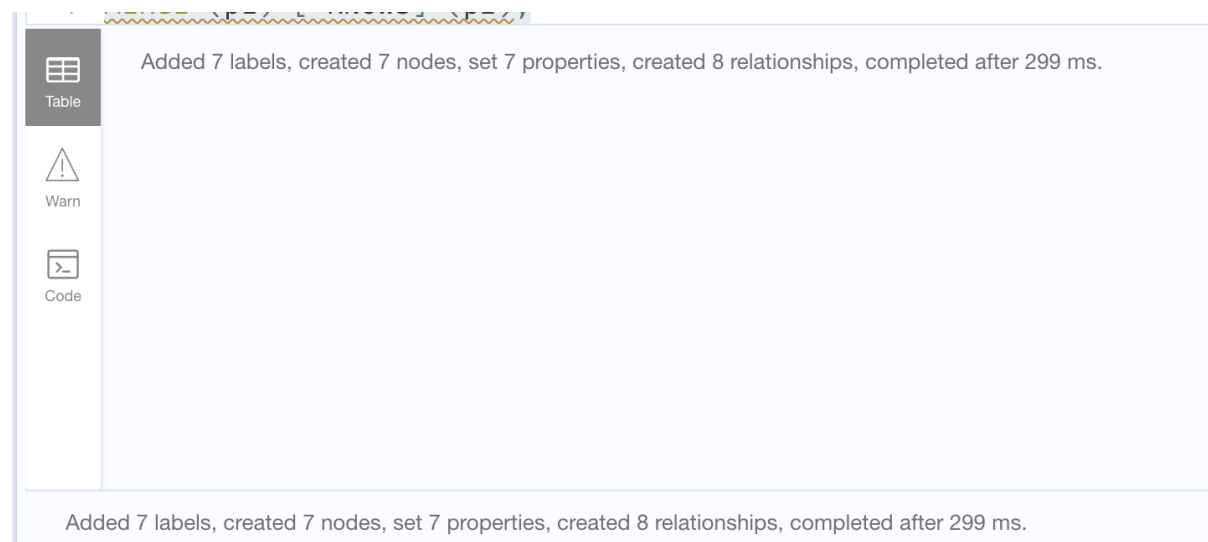
Data Setup with LOAD CSV

LOAD CSV FROM 'file:///people.csv' as row

MERGE (p1:Person {name: row[0]})

MERGE (p2:Person {name: row[1]})

MERGE (p1)-[:KNOWS]-(p2);

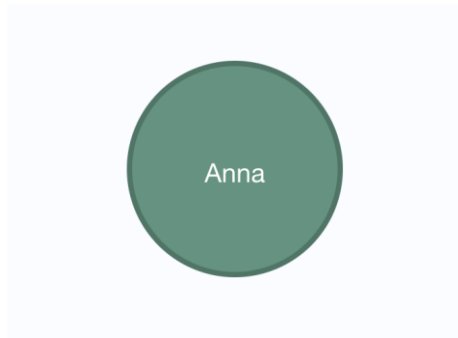
**Friend of Friends query**

MATCH (person:Person)-[:KNOWS]-(friend)-[:KNOWS]-(foaf)

WHERE person.name = "Joe"

AND NOT (person)-[:KNOWS]-(foaf)

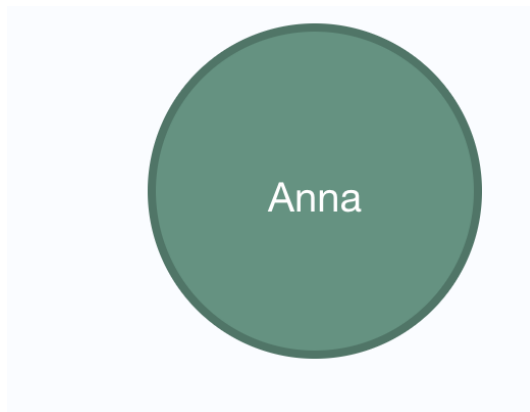
RETURN DISTINCT foaf



Friends of Friends

Find all of Joe's second-degree friends.

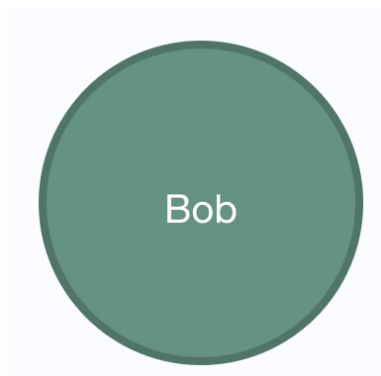
```
MATCH (user:Person)-[:KNOWS]-(friend)-[:KNOWS]-(foaf)
WHERE user.name = "Joe" AND NOT(user)-[:KNOWS]-(foaf)
RETURN foaf
```



Common Friends

Find all friends that Joe has in common with Sally.

```
MATCH (user:Person)-[:KNOWS]-(friend)-[:KNOWS]-(foaf:Person)
WHERE user.name = "Joe" AND foaf.name = "Sally"
RETURN friend
```

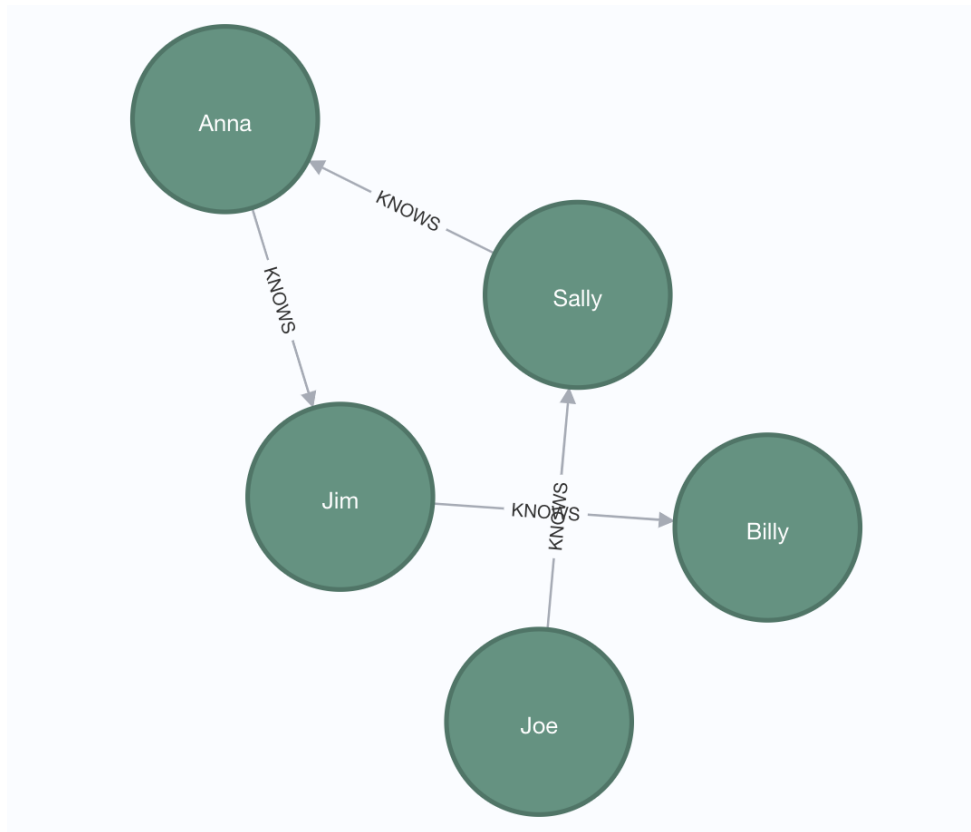


Connecting Paths

Find all friends on the shortest path that connects Joe to Billy.

```
MATCH path = shortestPath((p1:Person)-[:KNOWS*..6]-(p2:Person))
```

WHERE p1.name = "Joe" AND p2.name = "Billy"
RETURN path



Result

Ex No 5.**Implement secure search in social media****Date:****Aim :** To Implement secure search in social media using Neo4j**Procedure :**

1. Design the data model in Neo4j to represent the social media network.
2. Implement role-based access control (RBAC) to manage user access to the Neo4j database.
3. Encrypt sensitive data at rest using Neo4j's encryption features or external encryption tools to protect user data from unauthorized access.
4. Develop a secure search algorithm that ensures the privacy of user data while allowing users to search for posts, comments, or other content within the social media platform.
5. Implement query filtering mechanisms to prevent unauthorized access to user data.
6. Enable audit logging in Neo4j to track user activities, including search queries, access attempts, and modifications to the database schema or permissions.
7. Conduct regular security audits of the Neo4j database and the application infrastructure to identify and remediate security vulnerabilities.

```

CREATE (smartphones:Category {title: 'Smartphones'}),
(notebooks:Category {title: 'Notebooks'}),
(cameras:Category {title: 'Cameras'})

// Smartphones
CREATE (sony_xperia_z22:Product {title: 'Sony Xperia Z22', price: 765.00,
shippability: true, availability: true})
CREATE (samsung_galaxy_s8:Product {title: 'Samsung Galaxy S8', price:
784.00, shippability: true, availability: true})
CREATE (sony_xperia_xa1:Product {title: 'Sony Xperia XA1 Dual G3112',
price: 229.50, shippability: true, availability: false})
CREATE (iphone_8:Product {title: 'Apple iPhone 8 Plus 64GB', price: 874.20,
shippability: true, availability: false})
CREATE (xiaomi_mi_mix_2:Product {title: 'Xiaomi Mi Mix 2', price: 420.87,
shippability: true, availability: true})
CREATE (huawei_p8:Product {title: 'Huawei P8 Lite', price: 191.00,
shippability: true, availability: true})

MERGE (sony_xperia_z22)-[:IS_IN]->(smartphones)
MERGE (samsung_galaxy_s8)-[:IS_IN]->(smartphones)
MERGE (sony_xperia_xa1)-[:IS_IN]->(smartphones)
MERGE (iphone_8)-[:IS_IN]->(smartphones)
MERGE (xiaomi_mi_mix_2)-[:IS_IN]->(smartphones)
MERGE (huawei_p8)-[:IS_IN]->(smartphones)

// Notebooks
CREATE (acer_swift_3:Product {title: 'Acer Swift 3 SF314-51-34TX', price:
595.00, shippability: true, availability: false})
CREATE (hp_pro_book:Product {title: 'HP ProBook 440 G4', price: 771.30,
shippability: true, availability: true})
CREATE (dell_inspiron_15:Product {title: 'Dell Inspiron 15 7577', price:
1477.50, shippability: true, availability: true})
CREATE (apple_macbook:Product {title: "Apple MacBook A1534 12' Rose Gold",
price: 1293.00, shippability: false, availability: true})

MERGE (acer_swift_3)-[:IS_IN]->(notebooks)
MERGE (hp_pro_book)-[:IS_IN]->(notebooks)
MERGE (dell_inspiron_15)-[:IS_IN]->(notebooks)
MERGE (apple_macbook)-[:IS_IN]->(notebooks)

// Cameras

```

```
CREATE (canon_eos_6d:Product {title: 'Canon EOS 6D Mark II Body', price:
1794.00, shippability: true, availability: false})
CREATE (nikon_d7500:Product {title: 'Nikon D7500 Kit 18-105mm VR', price:
1612.35, shippability: true, availability: true})
```

```
MERGE (canon_eos_6d)-[:IS_IN]->(cameras)
MERGE (nikon_d7500)-[:IS_IN]->(cameras)
```

```
neo4j$ CREATE (smartphones:Category {title: 'Smartphones'}), (notebooks:Category {title...
```



Added 15 labels, created 15 nodes, set 51 properties, created 12 relationships, completed after 78 ms.

Added 15 labels, created 15 nodes, set 51 properties, created 12 relationships, completed after 78 ms.

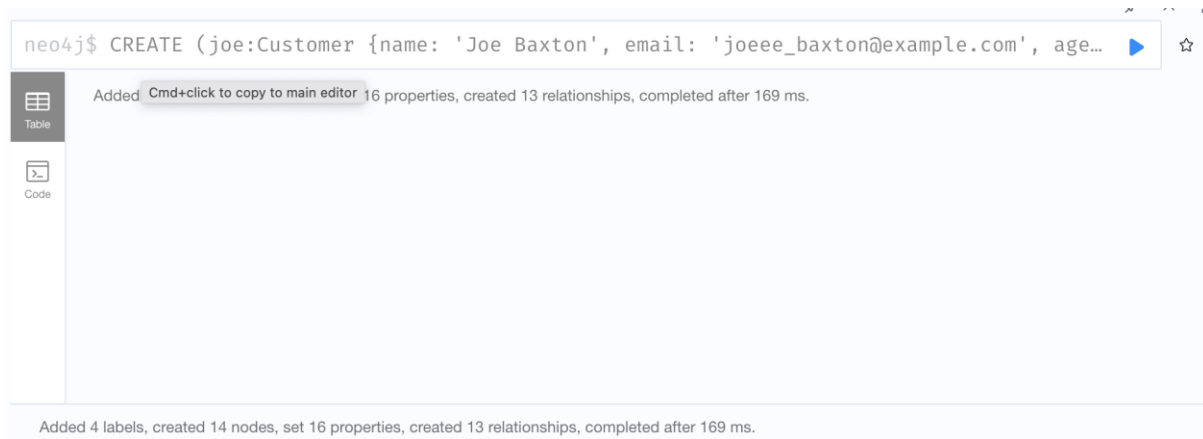
```
CREATE (joe:Customer {name: 'Joe Baxton', email:
'joeee_baxton@example.com', age: 25})
CREATE (daniel:Customer {name: 'Daniel Johnston', email:
'dan_j@example.com', age: 31})
CREATE (alex:Customer {name: 'Alex McGyver', email: 'mcgalex@example.com',
age: 22})
CREATE (alisson:Customer {name: 'Allison York', email:
'ally_york1@example.com', age: 24})

MERGE (joe)-[:VIEWED {views_count: 15}]->(nikon_d7500)
MERGE (joe)-[:ADDED_TO_WISH_LIST]->(iphone_8)
MERGE (joe)-[:BOUGHT]->(apple_macbook)

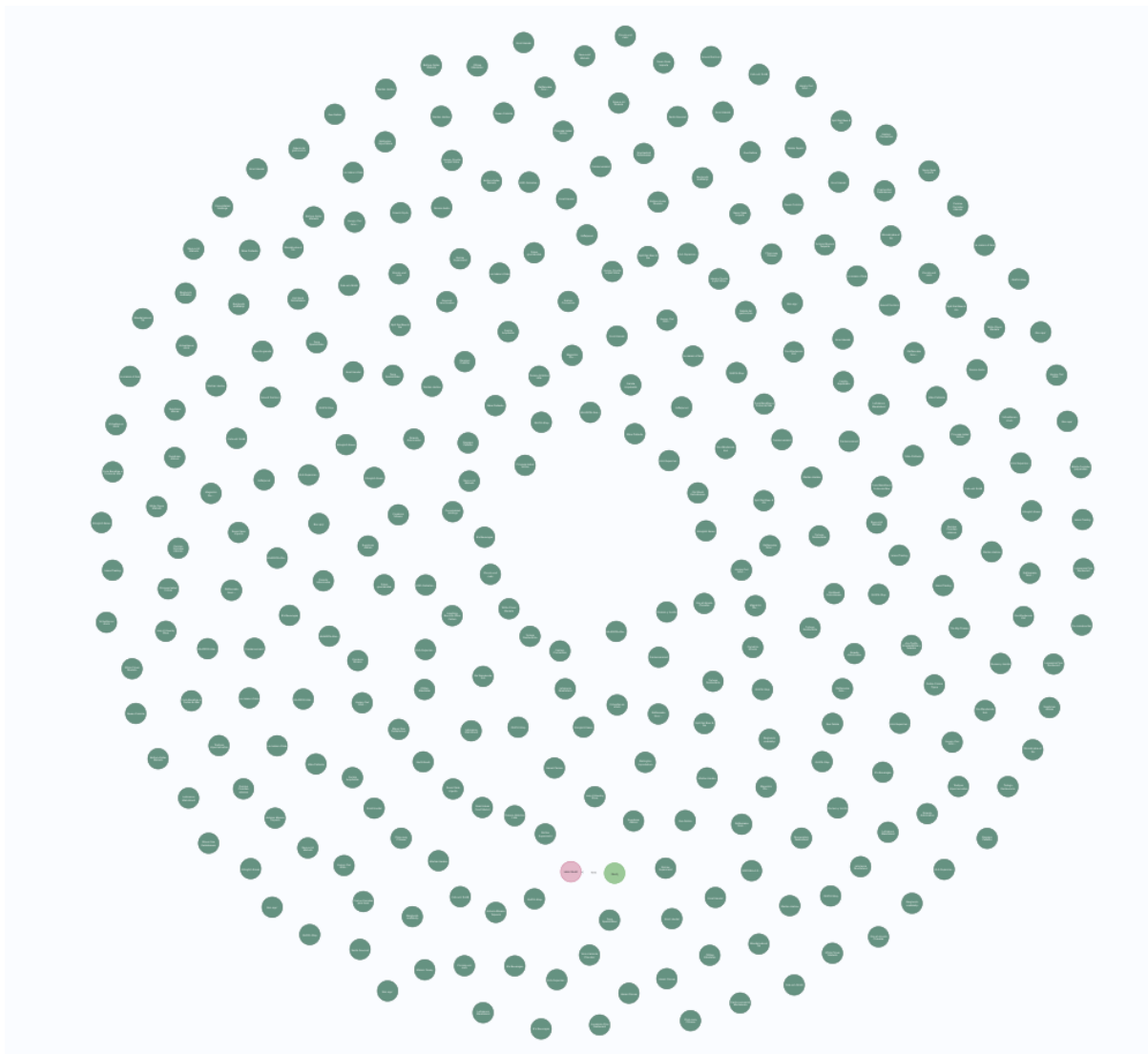
MERGE (daniel)-[:VIEWED {views_count: 10}]->(sony_xperia_z22)
MERGE (daniel)-[:VIEWED {views_count: 20}]->(dell_inspiron_15)
MERGE (daniel)-[:ADDED_TO_WISH_LIST]->(dell_inspiron_15)

MERGE (alex)-[:VIEWED {views_count: 20}]->(canon_eos_6d)
MERGE (alex)-[:ADDED_TO_WISH_LIST]->(sony_xperia_xa1)
MERGE (alex)-[:ADDED_TO_WISH_LIST]->(nikon_d7500)
MERGE (alex)-[:BOUGHT]->(xiaomi_mi_mix_2)

MERGE (alisson)-[:ADDED_TO_WISH_LIST]->(acer_swift_3)
MERGE (alisson)-[:ADDED_TO_WISH_LIST]->(hp_pro_book)
MERGE (alisson)-[:BOUGHT]->(huawei_p8)
MERGE (alisson)-[:BOUGHT]->(sony_xperia_xa1);
```



MATCH (n) RETURN n




Result:

Ex No 6**Create a simple Security & Privacy detector****Date:****Aim :** To Create a simple Security & Privacy detector using**Port Scanning****Procedure :**

- The first test we'll execute looks for different ways to infiltrate the target system. We do this by scanning for open ports using a script like portScanner.py.
- An open port can provide an ingress point if we can determine what type of traffic the target machine is expecting on that port.
- We'll be using sockets to test for connection, and a threaded model to speed up the process.
- set my thread pool at 500 concurrent requests and checked every port up to 10,000. I include these as constants to make them easy to change.

Program :

```
import socket
from concurrent import futures
def check_port(targetIp, portNumber, timeout):
    TCPsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    TCPsock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    TCPsock.settimeout(timeout)
    try:
        TCPsock.connect((targetIp, portNumber))
        return (portNumber)
    except:
        return
def port_scanner(targetIp, timeout):
    threadPoolSize = 500
    portsToCheck = 10000
    executor = futures.ThreadPoolExecutor(max_workers=threadPoolSize)
    checks = [
        executor.submit(check_port, targetIp, port, timeout)
        for port in range(0, portsToCheck, 1)
    ]
    for response in futures.as_completed(checks):
        if (response.result()):
            print('Listening on port: {}'.format(response.result()))
def main():
    targetIp = input("Enter the target IP address: ")
    timeout = int(input("How long before the connection times out: "))
    port_scanner(targetIp, timeout)
if __name__ == "__main__":
    main()
```

Output : Enter the target IP address: 127.0.0.1
How long before the connection times out: 1
Listening on port: 3453
Listening on port: 8080**Result :**