Assignment 3
Graph Databases
Due Date Oct 15th
(10 points)

NAME: JAHNAVI ANGATI

You will be using the graph database from Neo4j to create the database and answer questions based on that.
Go to the website to download the graph dbms at: https://neo4j.com/docs/desktop-manual/current/installation/download-installation/

Please create the data using the following 'english language' statements.

User Nodes:
1. User ID: 1, Name: Isaac, Last Name: Newton, Age 20
2. User ID: 2, Name: Richard, Last Name: Feynman, Age 30
3. User ID: 3, Name: Chandrasekhara, Last Name: Raman, Age 40
4. User ID: 4, Name: Srinivasa, Last Name: Ramanujam, Age 25
5. User ID: 5, Name: Galileo, Last Name: Galilei, Age 78
6. User ID: 6, Name: Marie, Last Name: Curie, Age 35
7. User ID: 7, Name: Albert, Last Name: Einstein, Age 100
8. User ID: 8, Name: Oviyasri, Last Name: Aryabhata, Age 50
9. User ID: 9, Name: Rabindranath, Last Name: Tagore, Age 90
10. User ID: 10, Name: Paul, Last Name: Erdos, Age 75

Group Nodes:
11. Group ID: 101, Name: Isotope Enthusiasts
12. Group ID: 102, Name: Food Lovers
13. Group ID: 103, Name: Tech Innovators

Post Nodes:
14. Post ID: 201, Title: "Exploring Radium"
15. Post ID: 202, Title: "Best Pizza Places"
16. Post ID: 203, Title: "AI Breakthroughs"

Comment Nodes:
17. Comment ID: 301, Text: "Awesome photos!"
18. Comment ID: 302, Text: "I want to visit the moon."
19. Comment ID: 303, Text: "That pizza looks delicious!"

Relationships:
- Isaac created Post 201
- Richard created Post 202
- Chandrasekhara created Post 203
- Isaac commented on Post 202
- Richard commented on Post 201
- Chandrasekhara commented on Post 201
- Srinivasa liked Post 201
- Srinivasa liked Comment 302
- Galileo liked Comment 301
- Galileo liked Comment 303
- Marie created Group 101
- Albert created Group 102
- Oviyasri created Group 103
- Isaac is a member of Group 101
- Richard is a member of Group 102
- Chandrasekhara is a member of Group 103
- Paul is a member of Group 101
- Rabindranath is a member of Group 102

Solution:

1. **USER NODES:**

```
CREATE (u1:User {userID: 1, Name: "Isaac", LastName: "Newton", Age: 20}),
    (u2:User {userID: 2, Name: "Richard", LastName: "Feynman", Age: 30}),
    (u3:User {userID: 3, Name: "Chandrasekhara", LastName: "Raman", Age: 40}),
    (u4:User {userID: 4, Name: "Srinivasa", LastName: "Ramanujam", Age: 25}),
    (u5:User {userID: 5, Name: "Galileo", LastName: "Galilei", Age: 78}),
    (u6:User {userID: 6, Name: "Marie", LastName: "Curie", Age: 35}),
    (u7:User {userID: 7, Name: "Albert", LastName: "Einstein", Age: 100}),
    (u8:User {userID: 8, Name: "Oviyasri", LastName: "Aryabhata", Age: 50}),
    (u9:User {userID: 9, Name: "Rabindranath", LastName: "Tagore", Age: 90}),
    (u10:User {userID: 10, Name: "Paul", LastName: "Erdos", Age: 75});
```

2. **GROUP NODES:**

```
CREATE (g1:Group {groupID: 101, Name: "Isotope Enthusiasts"}),
    (g2:Group {groupID: 102, Name: "Food Lovers"}),
    (g3:Group {groupID: 103, Name: "Tech Innovators"});
```

3. **POST NODES:**

```
CREATE (p1:Post {postID: 201, Title: "Exploring Radium"}),
    (p2:Post {postID: 202, Title: "Best Pizza Places"}),
    (p3:Post {postID: 203, Title: "AI Breakthroughs"});
```

4. **COMMENT NODES:**

```
CREATE (c1:Comment {commentID: 301, Text: "Awesome photos!"}),
    (c2:Comment {commentID: 302, Text: "I want to visit the moon."}),
    (c3:Comment {commentID: 303, Text: "That pizza looks delicious!"});
```

5. **RELATIONSHIPS:**

```
// Isaac created Post 201
MATCH (isaac:User {Name: "Isaac"}), (post201:Post {postID: 201})
CREATE (isaac)-[:CREATED]->(post201);

// Richard created Post 202
MATCH (richard:User {Name: "Richard"}), (post202:Post {postID: 202})
CREATE (richard)-[:CREATED]->(post202);
```

```
// Chandrasekhara created Post 203
MATCH (chandra:User {Name: "Chandrasekhara"}), (post203:Post {postID: 203})
CREATE (chandra)-[:CREATED]->(post203);

// Isaac commented on Post 202
MATCH (isaac:User {Name: "Isaac"}), (commentOn202:Post {postID: 202})
CREATE (isaac)-[:COMMENTED_ON]->(commentOn202);

// Richard commented on Post 201
MATCH (richard:User {Name: "Richard"}), (commentOn201:Post {postID: 201})
CREATE (richard)-[:COMMENTED_ON]->(commentOn201);

// Chandrasekhara commented on Post 201
MATCH (chandra:User {Name: "Chandrasekhara"}), (commentOn201:Post {postID: 201})
CREATE (chandra)-[:COMMENTED_ON]->(commentOn201);

// Srinivasa liked Post 201
MATCH (srinivasa:User {Name: "Srinivasa"}), (likePost201:Post {postID: 201})
CREATE (srinivasa)-[:LIKED]->(likePost201);

// Srinivasa liked Comment 302
MATCH (srinivasa:User {Name: "Srinivasa"}), (likeComment302:Comment {commentID: 302})
CREATE (srinivasa)-[:LIKED]->(likeComment302);

// Galileo liked Comment 301 and 303
MATCH (galileo:User {Name: "Galileo"}), (likeComment301:Comment {commentID: 301})
CREATE (galileo)-[:LIKED]->(likeComment301);

MATCH (galileo:User {Name: "Galileo"}), (likeComment303:Comment {commentID: 303})
```

```
CREATE (galileo)-[:LIKED]->(likeComment303);

// Marie, Albert, and Oviyasri created groups
MATCH (marie:User {Name: "Marie"}), (group101:Group {groupID: 101})
CREATE (marie)-[:CREATED]->(group101);

MATCH (albert:User {Name: "Albert"}), (group102:Group {groupID: 102})
CREATE (albert)-[:CREATED]->(group102);

MATCH (oviyasri:User {Name: "Oviyasri"}), (group103:Group {groupID: 103})
CREATE (oviyasri)-[:CREATED]->(group103);

// Memberships
MATCH (isaac:User {Name: "Isaac"}), (group101:Group {groupID: 101})
CREATE (isaac)-[:MEMBER_OF]->(group101);

MATCH (richard:User {Name: "Richard"}), (group102:Group {groupID: 102})
CREATE (richard)-[:MEMBER_OF]->(group102);

MATCH (chandra:User {Name: "Chandrasekhara"}), (group103:Group {groupID: 103})
CREATE (chandra)-[:MEMBER_OF]->(group103);

MATCH (paul:User {Name: "Paul"}), (group101:Group {groupID: 101})
CREATE (paul)-[:MEMBER_OF]->(group101);

MATCH (rabindranath:User {Name: "Rabindranath"}), (group102:Group {groupID: 102})
CREATE (rabindranath)-[:MEMBER_OF]->(group102);
```

Write queries to answer the following questions.

Copy and paste the queries into a file, so that the TAs can copy and paste these comments to check for correctness.

Then make a screen shot of your result, and then paste it into the file.

You will need to submit this file into Canvas.

(Questions 4, 5, 6 8. and 9 are not part of the homework any more. Please feel free to ignore them. Thanks!)

1. Find all posts created by Isaac and list the users who liked those posts.

```
MATCH (isaac:User {Name: "Isaac"})-[:CREATED]->(p:Post)<-[:LIKED]-(likedBy:User)

RETURN p.Title AS PostTitle, collect(likedBy.Name) AS UsersWhoLiked
```

	PostTitle	UsersWhoLiked
1	"Exploring Radium"	["Srinivasa"]

2. Write a query to identify posts that received more than two comments and display the number of comments for each of these posts.

```
MATCH (post:Post)<-[:COMMENTED_ON]-(commenter:User)

WITH post, COUNT(commenter) AS NumberOfComments

WHERE NumberOfComments > 2

RETURN post.Title AS PostTitle, NumberOfComments

ORDER BY NumberOfComments DESC
```

(no changes, no records)

3. Calculate the user engagement score for each user, where the score is the sum of the likes received on their posts and comments. List users ranked by their engagement score.

```
MATCH (user:User)

OPTIONAL MATCH (user)-[:CREATED]->(post:Post)<-[:LIKED]-()

WITH user, COUNT(DISTINCT post) AS PostLikes

OPTIONAL MATCH (user)-[:CREATED]->(comment:Comment)<-[:LIKED]-()

WITH user, PostLikes, COUNT(DISTINCT comment) AS CommentLikes

RETURN user.Name AS UserName, PostLikes, CommentLikes, PostLikes + CommentLikes AS TotalEngagementScore

ORDER BY TotalEngagementScore DESC
```

	UserName	PostLikes	CommentLikes	TotalEngagementScore
1	"Isaac"	1	0	1
2	"Richard"	0	0	0
3	"Chandrasekhara"	0	0	0
4	"Srinivasa"	0	0	0
5	"Galileo"	0	0	0
6	"Marie"	0	0	0
7				

ted streaming 10 records after 13 ms and completed after 28 ms.

7	"Albert"	0	0	0
8	"Oviyasri"	0	0	0
9	"Rabindranath"	0	0	0
10	"Paul"	0	0	0

4. Calculate the average number of comments on posts created by each user. List users with their average comment counts.

```
MATCH (user:User)-[:CREATED]->(post:Post)

OPTIONAL MATCH (post)<-[:COMMENTED_ON]-(commenter:User)

WITH user, post, COUNT(commenter) AS CommentsPerPost

RETURN user.Name AS UserName, AVG(CommentsPerPost) AS
AverageComments

ORDER BY AverageComments DESC
```

	UserName	AverageComments
1	"Isaac"	2.0
2	"Richard"	1.0
3	"Chandrasekhara"	0.0

rted streaming 3 records after 10 ms and completed after 12 ms.

5. Propose a real-world use case where this social network graph database could be applied effectively, considering user engagement, content creation, and interactions. Describe the scenario and how the database would be used.

Use Case: Health and Wellness Social Network with HealthHub

Scenario:

Imagine a digital platform named "HealthHub" that serves as a comprehensive health and wellness community. The platform integrates various health-focused features such as personalized diet plans, workout regimes, meditation sessions, and community forums. Users can join specific health challenges, share their fitness journeys, create and share content related to health and wellness, and interact with health professionals and other users.Database Utility:

1. **User Profiles:**

   Metrics: Using data from health check-ups and trackers, nodes can represent metrics like weight and blood pressure.

   Goals: Users can set health targets such as weight loss or marathon training.

2. **Content & Interaction:**

   Plans: Nutritionists and trainers create workout and diet plans, allowing users to enroll, comment, and share.

   Forums: Users can initiate or join health-focused discussions.

   Articles: Users and professionals can produce and link content to health topics or challenges.

3. **Recommendation Engine:**

   Health Plans: Recommendations arise from user metrics and goals.

   Forums & Articles: Suggestions are based on user metrics, goals, and interactions.

4. **Engagement & Tracking:**

   Milestones: User progress is celebrated in the community.

   Scores: Based on user activity, engagement scores can drive motivation through gamification.

5. **Networking & Support:**

   Challenges: Users initiate or join events like "30-day yoga."

   Groups: Support groups cater to users with shared health challenges.

Summary:

By incorporating a graph database, HealthHub offers a personalized and interconnected health journey. Users benefit from tailored recommendations and a supportive community, solidifying HealthHub as a central hub for health enthusiasts and professionals.