

## Kiểm tra cuối kì

2274802010922 – Ô Bảo Trí

### Câu 1: Tạo dữ liệu mẫu

Note that Cypher queries must end with a semicolon.

neo4j@neo4j>

neo4j@neo4j>

>....

```
CREATE (:Person {name: "Robert Zemeckis", birthYear: 1952, country: "USA", yearsWorked: 42});
CREATE (:Person {name: "Natalie Portman", birthYear: 1981, country: "Israel", yearsWorked: 25});
CREATE (:Person {name: "Christopher Nolan", birthYear: 1970, country: "UK", yearsWorked: 35});
CREATE (:Person {name: "Emma Watson", birthYear: 1990, country: "UK", yearsWorked: 15});
```

// Tạo các nút Movie

```
CREATE (:Movie {title: "Forrest Gump", year: 1994, type: "Drama", budget: 50});
CREATE (:Movie {title: "Sleepless in Seattle", year: 1993, type: "Romance", budget: 20});
CREATE (:Movie {title: "Saving Private Ryan", year: 1998, type: "War", budget: 70});
CREATE (:Movie {title: "Titanic", year: 1997, type: "Romance", budget: 200});
CREATE (:Movie {title: "The Terminal", year: 2004, type: "Drama", budget: 60});
CREATE (:Movie {title: "Catch Me If You Can", year: 2002, type: "Crime", budget: 52});
CREATE (:Movie {title: "Inception", year: 2010, type: "Sci-Fi", budget: 160});
CREATE (:Movie {title: "Harry Potter 1", year: 2001, type: "Fantasy", budget: 125});
CREATE (:Movie {title: "Interstellar", year: 2014, type: "Sci-Fi", budget: 165});
CREATE (:Movie {title: "Black Swan", year: 2010, type: "Thriller", budget: 13});
```

// Quan hệ ACTED\_IN

```
MATCH (p:Person {name: "Tom Hanks"}), (m:Movie {title: "Forrest Gump"})
CREATE (p)-[:ACTED_IN {role: "Forrest", salary: 10}]->(m);
```

```
MATCH (p:Person {name: "Meg Ryan"}), (m:Movie {title: "Sleepless in Seattle"})
CREATE (p)-[:ACTED_IN {role: "Annie", salary: 4}]->(m);
```

// (Thêm các quan hệ tương tự cho các diễn viên và phim khác dựa theo bảng đề bài)

// Quan hệ DIRECTED

```
MATCH (p:Person {name: "Steven Spielberg"}), (m:Movie {title: "Saving Private Ryan"})
CREATE (p)-[:DIRECTED {yearsSpent: 3}]->(m);
```

```
CREATE (p)-[:ACTED_IN {role: "Annie", salary: 4}]->(m);
```

// (Thêm các quan hệ tương tự cho các diễn viên và phim khác dựa theo bảng đề bài)

// Quan hệ DIRECTED

```
MATCH (p:Person {name: "Steven Spielberg"}), (m:Movie {title: "Saving Private Ryan"})
CREATE (p)-[:DIRECTED {yearsSpent: 3}]->(m);
```

// Quan hệ WORKED\_WITH

```
MATCH (p1:Person {name: "Tom Hanks"}), (p2:Person {name: "Steven Spielberg"})
CREATE (p1)-[:WORKED_WITH {projects: 3, lastYear: 2004}]->(p2);
```

```
0 rows
ready to start consuming query after 258 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 21 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 44 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 17 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 15 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 30 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 15 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 15 ms, results consumed after another 0 ms
```

```

Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 37 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 12 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 14 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 14 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 14 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 13 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 13 ms, results consumed after another 0 ms
Added 1 nodes, Set 4 properties, Added 1 labels
0 rows
ready to start consuming query after 308 ms, results consumed after another 0 ms
Created 1 relationships, Set 2 properties
0 rows
ready to start consuming query after 24 ms, results consumed after another 0 ms
Created 1 relationships, Set 2 properties
0 rows
ready to start consuming query after 77 ms, results consumed after another 0 ms
Created 1 relationships, Set 1 properties
0 rows
ready to start consuming query after 93 ms, results consumed after another 0 ms
Created 1 relationships, Set 2 properties

```

## Câu 2: Tìm Phim và Người

1.

```

neo4j@neo4j> MATCH (p:Person {name: "Tom Hanks"})-[:ACTED_IN|WORKED_WITH*]-(m:Movie)
RETURN m.title AS title, m.year AS year, m.type AS type,
CASE WHEN (p)-[:ACTED_IN]->(m) THEN "Actor" ELSE "Collaborator" END AS role;

+-----+
| title          | year | type  | role  |
+-----+
| "Forrest Gump" | 1994 | "Drama" | "Actor" |
+-----+

1 row
ready to start consuming query after 290 ms, results consumed after another 16 ms

```

2.

```

neo4j@neo4j>
neo4j@neo4j> MATCH (p:Person)-[:DIRECTED]->(m:Movie)
WHERE p.yearsWorked > 40
WITH p, COLLECT(m.title) AS movies, COUNT(m) AS movieCount
WHERE movieCount >= 2
RETURN p.name AS name, p.country AS country, movies;

+-----+
| name | country | movies |
+-----+
+-----+

0 rows
ready to start consuming query after 224 ms, results consumed after another 17 ms

```

## Câu 3: Tìm bạn diễn và phim

1.

```
neo4j@neo4j> MATCH (p:Person {name: "Tom Hanks"})-[:ACTED_IN]->(m:Movie)-[:ACTED_IN]-(coActor)
WHERE m.budget > 50
WITH coActor, m
MATCH (p)-[:WORKED_WITH]->(coActor)
WHERE p.projects >= 2
RETURN coActor.name AS name, m.title AS title, m.budget AS budget, p.projects AS projects
ORDER BY m.budget DESC;

+-----+
| name | title | budget | projects |
+-----+
+-----+

0 rows
ready to start consuming query after 460 ms, results consumed after another 5 ms
neo4j@neo4j>
```

2.

```
neo4j@neo4j> MATCH (m:Movie {type: "Sci-Fi"})<-[:ACTED_IN]-(p:Person)
WHERE m.budget > 100
WITH p, m
MATCH (p)-[:WORKED_WITH]->(:Person {name: "Leonardo DiCaprio"})
RETURN p.name AS name, m.title AS title, m.budget AS budget;

+-----+
| name | title | budget |
+-----+
+-----+

0 rows
ready to start consuming query after 134 ms, results consumed after another 1 ms
neo4j@neo4j>
```

3.

```
neo4j@neo4j> MATCH (p:Person {name: "Kate Winslet"})-[:ACTED_IN]->(m:Movie)
WHERE m.type = "Romance" AND m.budget > 150
RETURN COUNT(m) AS romanceMovieCount;

+-----+
| romanceMovieCount |
+-----+
| 0 |
+-----+

1 row
ready to start consuming query after 96 ms, results consumed after another 3 ms
neo4j@neo4j>
```

## Câu 4: Đường đi và phân tích

1.

```
neo4j@neo4j> MATCH path = shortestPath((p1:Person {name: "Tom Hanks"})-[:ACTED_IN|DIRECTED|WORKED_WITH*]-(p2:Person {name: "James Cameron"}))
WITH path
MATCH (p1:Person {name: "Tom Hanks"})-[:ACTED_IN]->(m:Movie)
WITH path, SUM(m.budget) AS totalSalary
RETURN path, totalSalary;

+-----+
| path | totalSalary |
+-----+
+-----+

0 rows
ready to start consuming query after 192 ms, results consumed after another 14 ms
neo4j@neo4j>
```

2.

```
neo4j@neo4j> MATCH path = shortestPath((p1:Person {name: "Steven Spielberg"})-[:ACTED_IN|DIRECTED|WORKED_WITH]-(p2:Person {name: "Natalie Portman"}))
WITH path
MATCH (p2:Person {name: "Natalie Portman"})-[:ACTED_IN]->(m:Movie)
WITH path, SUM(m.salary) AS totalSalary, COLLECT(m.type) AS types
RETURN path, totalSalary, types;
```

```
+-----+
| path | totalSalary | types |
+-----+
```

```
0 rows
ready to start consuming query after 120 ms, results consumed after another 2 ms
neo4j@neo4j>
```

3.

```
neo4j@neo4j> MATCH (p:Person)-[:WORKED_WITH]->(colleague)
WITH p, COUNT(colleague) AS connections
WHERE connections >= 3
RETURN COUNT(p) AS countOfPeople;
```

```
+-----+
| countOfPeople |
+-----+
| 0 |
+-----+
```

```
1 row
ready to start consuming query after 63 ms, results consumed after another 2 ms
```