CALL {

MATCH (n)

RETURN labels(n) AS labels, keys(n) AS properties

}

UNWIND labels AS label

WITH label, properties

UNWIND properties AS property

WITH label, COLLECT(DISTINCT property) AS uniqueProperties

RETURN label, uniqueProperties ORDER BY label;

CALL {

MATCH ()-[r]->()

RETURN type(r) as relationshipType, keys(r) AS properties

}

UNWIND properties AS property

WITH relationshipType, COLLECT(DISTINCT property) AS uniqueProperties

RETURN relationshipType, uniqueProperties ORDER BY relationshipType;

MATCH (a:Person)-[:ACTED\_IN]->(m:Movie)

WHERE (a)-[:DIRECTED]->(m)

RETURN a.name, collect(m.title) as movies, count(\*) as count

ORDER BY count DESC LIMIT 5;

MATCH (a:Person)-[:ACTED\_IN]->(m:Movie)

RETURN m.title, count(a.name) as count

ORDER BY count DESC LIMIT 5;

MATCH (m:Movie) RETURN min(m.year) ;

MATCH (m1:Movie)<-[:ACTED\_IN]-(a:Actor)-[:ACTED\_IN]->(m2:Movie)

WHERE id(m1) < id(m2)

WITH m1, m2, COUNT(a) AS sharedActors

ORDER BY sharedActors DESC

RETURN m1.title, m2.title, sharedActors

LIMIT 6;

MATCH (u:User)-[:RATED]->(m:Movie)<-[:ACTED\_IN]-(a:Actor),

(a)-[:ACTED\_IN]->(rec:Movie)

WHERE NOT (u)-[:RATED]->(rec) AND m <> rec

WITH u, rec, COUNT(a) AS actorCount

ORDER BY actorCount DESC

RETURN u.name, rec.title, actorCount

LIMIT 6;

MATCH (d1:Director)-[:DIRECTED]->(:Movie)-[:IN\_GENRE]->(g:Genre)<-[:IN\_GENRE]-(:Movie)<-[:DIRECTED]-(d2:Director)

WHERE id(d1) < id(d2)

WITH d1, d2, COLLECT(DISTINCT g.name) AS sharedGenres

RETURN d1.name, d2.name, sharedGenres, SIZE(sharedGenres) AS genreOverlap

ORDER BY genreOverlap DESC

LIMIT 6;

MATCH (d:Director)-[:DIRECTED]->(m:Movie)<-[:ACTED\_IN]-(a:Actor)

WITH d, a, COUNT(m) AS moviesTogether

ORDER BY moviesTogether DESC

RETURN d.name AS Director, a.name AS Actor, moviesTogether

LIMIT 6;

MATCH (m:Movie)-[:IN\_GENRE]->(g:Genre)

WHERE m.year IS NOT NULL

WITH g.name AS Genre, m.year AS Year, COUNT(m) AS MoviesCount

ORDER BY Year

RETURN Genre, collect({Year: Year, Count: MoviesCount}) AS MoviesPerYear;

MATCH (a:Person)-[:ACTED\_IN]->(m:Movie)-[:IN\_GENRE]->(g:Genre)

WHERE g.name = 'Comedy'

RETURN DISTINCT a.name AS Actor, a.born

ORDER BY a.born ASC

LIMIT 5;

MATCH (a:Person)-[:ACTED\_IN]->(:Movie {title: "Jumanji"})

MATCH (a)-[ACTED\_IN]->(m:Movie)

RETURN m.title

LIMIT 5;

MATCH (a:Person)-[:ACTED\_IN]->(m:Movie)

RETURN a.name AS Actors, COLLECT(m.title) AS movies;

MATCH (a:Person)-[:ACTED\_IN]->(m:Movie)-[:IN\_GENRE]->(g:Genre)

WHERE g.name ="Drama"

RETURN DISTINCT  m.title AS Movie;

MATCH (u:User)-[r:RATED]->(m:Movie)<-[:DIRECTED]-(d:Director)

WITH u, d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount

WHERE ratingsCount > 5

RETURN u.name AS User, d.name AS Director, avgRating

ORDER BY avgRating DESC

LIMIT 6;

MATCH (a:Actor)-[:ACTED\_IN]->(m:Movie)

WITH a, COUNT(m) AS moviesCount

MATCH (a)-[:ACTED\_IN]->()<-[:ACTED\_IN]-(coActor)

WITH a, moviesCount, COUNT(DISTINCT coActor) AS coActorCount

RETURN a.name, moviesCount, coActorCount

ORDER BY coActorCount DESC, moviesCount DESC

LIMIT 10;

MATCH (a:Actor)-[:ACTED\_IN]->(m:Movie)<-[:ACTED\_IN]-(coActor:Actor)

WITH a, coActor, COUNT(m) AS moviesTogether

WHERE moviesTogether > 3 AND id(a) < id(coActor)

RETURN a.name, coActor.name, moviesTogether

ORDER BY moviesTogether DESC

LIMIT 10;

MATCH (g:Genre)<-[:IN\_GENRE]-(m:Movie)

WITH g, COUNT(m) AS moviesCount

MATCH (g)<-[:IN\_GENRE]-(m)<-[:ACTED\_IN]-(a:Actor)

WITH g, moviesCount, COUNT(DISTINCT a) AS actorsCount

ORDER BY moviesCount DESC, actorsCount DESC

RETURN g.name AS Genre, moviesCount, actorsCount

LIMIT 10;

MATCH (a:Actor)-[:ACTED\_IN]->(m:Movie)-[:IN\_GENRE]->(g:Genre)

WITH a, COUNT(DISTINCT g) AS genresCount

ORDER BY genresCount DESC

RETURN a.name AS Actor, genresCount

LIMIT 10;

MATCH (u:User)-[:RATED]->(m:Movie)

WITH u, AVG(m.runtime) AS avgLength

RETURN u.name AS User, avgLength

ORDER BY avgLength DESC

LIMIT 10;

MATCH (d:Director)-[:DIRECTED]->(m:Movie)<-[r:RATED]-(u:User)

WITH d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount

ORDER BY avgRating DESC, ratingsCount DESC

RETURN d.name AS Director, avgRating, ratingsCount

LIMIT 10;

MATCH (a:Actor)-[:ACTED\_IN]->(m:Movie)<-[r:RATED]-(u:User),

      (d:Director)-[:DIRECTED]->(m)

WITH a, d, AVG(r.rating) AS avgRating, COUNT(r) AS ratingsCount

ORDER BY avgRating DESC, ratingsCount DESC

RETURN a.name AS Actor, d.name AS Director, avgRating, ratingsCount

LIMIT 10;

MATCH path=shortestPath((a1:Actor)-[:ACTED\_IN\*]-(a2:Actor))

WHERE a1.name = 'Bruce Willis' AND a2.name = 'Robert De Niro'

RETURN path;

MATCH path = (p:Person)-[\*3..5]-(p)

RETURN path

LIMIT 1;