

1 Return the titles of movies acted in by 'Kevin Bacon'.

```
Match (a) -[:ACTED_IN] -> (m:Movie) where a.name='Kevin Bacon' return m.title
```

2 Return the count of the actors that act in some movie with Kevin Bacon

```
match (a:Person{name:"Kevin Bacon"})-[:ACTED_IN]-(m:Movie)-[:ACTED_IN]-(b:Person) return count(b)
```

3 Return the actors that are at three degrees or less of separation from 'Kevin Bacon'. One degree of separation are actors who act in a movie with Kevin Bacon, and two degrees of separation are actors who act in a movie with some actor who act in a movie with Kevin Bacon, etc.

```
MATCH (a:Person {name: "Kevin Bacon"})
```

```
-[:ACTED_IN]-(m:Movie)
```

```
-[:ACTED_IN]-(b:Person)
```

```
-[:ACTED_IN]-(m2:Movie)
```

```
-[:ACTED_IN]-(b2:Person)
```

```
-[:ACTED_IN]-(m3:Movie)
```

```
-[:ACTED_IN]-(b3:Person)
```

```
RETURN b.name,b2.name,b3.name
```

4 Return the persons and movies such that the person both directed and acted in the movie

```
MATCH (a:Person)-[:ACTED_IN]-(m:Movie)-[:DIRECTED]-(b:Person) where a=b return a,m
```

5 Return the actors who have acted in 4 or more movies

```
MATCH (a:Person)-[:ACTED_IN]-(m:Movie) with a, COUNT(m) as b where b>3 RETURN a
```

6 Return the shortest path between Kevin Bacon and Emile Hirsch

```
MATCH (a:Person {name: "Kevin Bacon"}),(b:Person {name: "Emile Hirsch"}), p = shortestPath((a)-[*]-(b)) RETURN p
```

7 Return the actors that have acted in all of the movies that Emile Hirsch acts in. You may want to form a collection of 'Emile Hirsch' movies and a collection of movies acted in by another person and then use apoc.coll.containsAll to determine if one collection contains (at least) all of the movies that the other has.

```
MATCH (a:Person {name: "Emile Hirsch"})-[:ACTED_IN]-(m:Movie), (b:Person)-[:ACTED_IN]-(m2:Movie) WHERE m2 = m RETURN b
```

