```
create (n1:CostCenter {id:1, Code:'CC1'});
create (n2:CostCenter {id:2, Code:'CC2'});
create (n3:Employee {id:3, Name:'Nathan', Surname:'Davies'});
create (n4:Employee {id:4, Name:'Rose', Surname:'Taylor'});
create (n5:Employee {id:5, Name:'Heather', Surname:'Underwood'});
create (n6:Employee {id:6, Name:'John', Surname:'Smith'});
create (n7:VacantPost {id:7});

match (from {id:3}), (to {id:1}) create (from)-[r1:BELONGS_TO]->(to)
set r1.FROM='2011-01-10';
match (from {id:3}), (to {id:4}) create (from)-[:REPORTS_TO]->(to);
match (from {id:4}), (to {id:1}) create (from)-[r3:MANAGER_OF]->(to)
set r3.FROM='2010-02-8';
match (from {id:6}), (to {id:2}) create (from)-[r4:BELONGS_TO]->(to)
set r4.FROM='2008-09-20';
match (from {id:6}), (to {id:5}) create (from)-[:REPORTS_TO]->(to);
match (from {id:5}), (to {id:2}) create (from)-[:BELONGS_TO]->(to);
match (from {id:5}), (to {id:7}) create (from)-[:REPORTS_TO]->(to);
match (from {id:7}), (to {id:2}) create (from)-[:BELONGS_TO]->(to);

//Find all employees, list names
match (e:Employee)
return e.Name, e.Surname

//Find all CostCenter 1 employees, list names
match (e:Employee)-[r]->(cc:CostCenter {Code:'CC1'})
return e.Name, e.Surname, r.FROM, cc.Code

match (e:Employee)-[r:BELONGS_TO]->(cc:CostCenter {Code:'CC1'})
return e.Name, e.Surname, r.FROM, cc.Code

//Find indirect links
match (e:Employee {Surname:'Smith'})-[*2]->(neighborhood)
return e.Name, e.Surname, neighborhood

//Range of hops
match (e:Employee {Surname:'Smith'})-[r*0..3]->(neighborhood)
return r, neighborhood

//Find all employees, list their possible reporting partners
match (e:Employee)
optional match (e)<-[r:REPORTS_TO]-(m:Employee)
return e.Surname, m.Surname

//Find a path
match p=(a {Surname:'Davies'})-[*]-(b {Surname:'Taylor'})
return p
```

```
//Find the shortest path
match (a {Surname:'Davies'}), (b {Surname:'Taylor'})
return allShortestPaths((a)-[*]-(b)) as path
```