Model the following Society relations between people working in “HCL”, as a graph model, and answer the queries using Cypher. A person can be a friend of another person. A person may have siblings (brothers / sisters), A person may be a parent (mother/father) of another person. A person stays either in Pune or Mumbai or Kolhapur.

```
create(p:Person{name:"Tushar",age:24}) return p;
create(p:Person{name:"Ganesh",age:45}) return p;
create(p:Person{name:"Amar",age:26}) return p;
create(b:Brother{name:"Vinay"}) return b;
create(s:Sister{name:"Sanskruti"}) return s;
create(c:Children{name:"Amit"}) return c;
create(l:Location{name:"Pune"}) return l;
create(l:Location{name:"Mumbai"}) return l;
create(l:Location{name:"Kolhapur"}) return l;
match(p:Person),(p1:Person) where p.name="Tushar" and
p1.name="Ganesh" create (p)-[:Friend_of]->(p1) return p,p1;
match(p:Person),(b:Brother) where p.name="Ganesh" and
b.name="Vinay" create (p)-[:Brother_of]->(b) return p,b;
match(p:Person),(c:Children) where p.name="Ganesh" and
c.name="Amit" create (p)-[:Parent_of]->(c) return p,c;
match(p:Person),(l:Location) where p.name="Ganesh" and
l.name="Pune" create (p)-[:Stays_in]->(l) return p,l;
match(p:Person),(l:Location) where p.name="Tushar" and
l.name="Mumbai" create (p)-[:Stays_in]->(l) return p,l;
match(p:Person),(l:Location) where p.name="Amar" and
l.name="Mumbai" create (p)-[:Stays_in]->(l) return p,l;
```

Queries

1) Display the names of people living in Mumbai.

```
match(p:Person),(l:Location) where l.name="Mumbai" and (p)-[:Stays_in]->(l) return
p.name;
```

2) Display the nodes having age above 40.

```
match(p:Person) where p.age>40 return p.name;
```

Model the following Import Export information as a graph model, and answer the following queries using Cypher. There are countries which import and export products to each other. Products are produced across different states in a country. Production of the products is measured in %. A product can be exported if its production exceeds 60%. A product needs to be imported if its consumption percentage is more than its production percentage in a country.

```
create (c:Country{name:"India"}) return c;
create (c:Country{name:"USA"}) return c;
create (c:Country{name:"Israil"}) return c;
create (c:Country{name:"Arab"}) return c;
create (c:Country{name:"Europe"}) return c;
create (s:States{name:"Maharashtra"}) return s;
create (s:States{name:"Punjab"}) return s;
create (s:States{name:"California"}) return s;
create (p:Product{name:"Wheat",production:"75%"}) return p;
create (p:Product{name:"Oil",production:"90%"}) return p;
create (p:Product{name:"Sugar",production:"50%"}) return p;
create (p:Product{name:"GroundNuts",production:"50%"}) return p;
create (p:Product{name:"Cotton",production:"50%"}) return p;
match (c:Country),(s:States),(p:Product) where c.name="India" and s.name="Maharashtra"
and
p.name="Wheat" create (c)-[:has_States]->(s)-[:Produces]->(p) return c,s,p;
match (c:Country),(s:States),(p:Product) where c.name="USA" and s.name="California" and
p.name="Oil" create (c)-[:has_States]->(s)-[:Produces]->(p) return c,s,p;
match (c:Country),(s:States),(p:Product) where c.name="India" and s.name="Maharashtra"
and
p.name="Oil" create (c)-[:has_States]->(s)-[:Produces]->(p) return c,s,p;
match (c:Country),(s:States),(p:Product) where c.name="India" and s.name="Maharashtra"
and
p.name="GroundNuts" create (c)-[:has_States]->(s)-[:Produces]->(p) return c,s,p;
match (c:Country),(s:States),(p:Product) where c.name="India" and s.name="Punjab" and
p.name="Sugar" create (c)-[:has_States]->(s)-[:Produces]->(p) return c,s,p;
```

Queries

1) List the countries that export oil

```
match (c:Country),(p:Product) where p.name="Oil" and (c)-[:exports]->(p) return c.name;
```

2) List the products produced in “Maharashtra”

```
match (s:States),(p:Product) where s.name="Maharashtra" and (s)-[:Produces]->(p) return
p.name;
```

Model the following Clothing Brand information as a graph model, and answer the following queries using Cypher. Consider a Mall for clothing. This mall will include different sections for males, females and kids. Each section contains different types of apparels from different brands. There are many apparels with different designs, of each type. An apparel may be available in one or more standard sizes (S/M/L/XL/XXL)

```
create (s:Section {name:"Male"}) return s;
create (s:Section {name:"Female"}) return s;
create (s:Section {name:"Kids"}) return s;
create (a:Apparel {name:"Kurta"}) return a;
create (a:Apparel {name:"Saree"}) return a;
create (a:Apparel {name:"T-Shirts"}) return a;
create (a:Apparel {name:"Jackets"}) return a;
create (a:Apparel {name:"Frock"}) return a;
create (a:Apparel {name:"Shirt"}) return a;
create (s:Size {name:"S"}) return s;
create (s:Size {name:"M"}) return s;
create (s:Size {name:"L"}) return s;
match(s:Section),(a:Apparel) where s.name="Female" and a.name="Kurta" create (s)-[:Has]-
>(a) return s,a;
match(s:Section),(a:Apparel) where s.name="Female" and a.name="Saree" create (s)-[:Has]-
>(a) return s,a;
match(s:Section),(a:Apparel) where s.name="Male" and a.name="Jackets" create (s)-[:Has]-
>(a) return s,a;
match(s:Section),(a:Apparel) where s.name="Male" and a.name="Jackets" create (s)-[:Has]-
>(a) return s,a;
```

```
match(s:Section),(a:Apparel) where s.name="Kids" and a.name="Frock" create (s)-[:Has]-
>(a) return s,a;
create (ss:SalesStaff {name:"Smita"}) return ss;
create (ss:SalesStaff {name:"Geeta"}) return ss;
create (ss:SalesStaff {name:"Seeta"}) return ss;
create (ss:SalesStaff {name:"Raman"}) return ss;
match(ss:SalesStaff),(s:Section) where s.name="Kids" and ss.name="Smita" create (ss)-
[:Work_in]->(s) return ss,s;
match(ss:SalesStaff),(s:Section) where s.name="Female" and ss.name="Geeta" create (ss)-
[:Work_in]->(s) return ss,s;
match(ss:SalesStaff),(s:Section) where s.name="Male" and ss.name="Seeta" create (ss)-
[:Work_in]->(s) return ss,s;
match(ss:SalesStaff),(s:Section) where s.name="Kids" and ss.name="Raman" create (ss)-
[:Work_in]->(s) return ss,s;
```

Queries
1)List the different apparels type in female section

```
match(s:Section),(a:Apparel) where s.name="Female" and (s)-[:Has]->(a) return a.name;
```

2) List the names of sales staff in Kids section.

```
match(ss:SalesStaff),(s:Section) where s.name="Kids" and (ss)-[:Work_in]->(s) return
ss.name;
```

Model the following Hotels information as a graph model, and answer the following queries using Cypher. Consider hotels in Pune. Some hotels provide lodging facility whereas some provide only restaurant facility and some provide both. A person can rate(1-5 stars) a hotel for its facility/facilities. A person can recommend a hotel to his/her friends. A person can provide a review for a hotel after his stay/visit.

```
create (h:Hotel {name:"Marriotte", location:"Camp"}) return h;

create (h:Hotel {name:"Blue Diamond", location:"KP Road"}) return h;

create (h:Hotel {name:"Radison", location:"Kharadi"}) return h;

create (f:Facility {name:"Lodging",rating:"4 star"}) return f;

create (f:Facility {name:"Restaurant ",rating:"4 star"}) return f;

create (f:Facility {name:"Lodging Restaurant ",rating:"5 star"}) return f;

match (h:Hotel),(f:Facility) where h.name="Blue Diamond" and f.name="Lodging
Restaurant " create (h)-[:Has]->(f) return h,f;

match (h:Hotel),(f:Facility) where h.name="Radison" and f.name="Restaurant " create (h)-
[:Has]->(f) return h,f;
```

```
match (h:Hotel),(f:Facility) where h.name="Marriotte" and f.name="Lodging Restaurant "
create (h)-[:Has]->(f) return h,f;
```

Queries

1)List the names of hotels in Camp area.

```
match (h:Hotel) where h.location="Camp" return h.name;
```

2) List the names of hotels having both lodging and restaurant facility

```
match (h:Hotel),(f:Facility) where f.name="Lodging Restaurant " and (h)-[:Has]->(f) return
h.name;
```

Model the following Hospitals information as a graph model, and answer the following queries using Cypher. Consider hospitals in and around Pune. Each hospital may have one or more specializations like Pediatric, Gynaec, Orthopedic, etc. A person can recommend/provide review for a hospital. A doctor can be associated with one or more hospitals.

```
create (h:Hospital {name:"Columbia"}) return h;
create (h:Hospital {name:"Rubi"}) return h;
create (h:Hospital {name:"Sahyadri"}) return h;
create(s:Specialization{name:"Pediatric"}) return s;
create(s:Specialization{name:"Orthopedic"}) return s;
create(s:Specialization{name:"Gynaec"}) return s;
match(h:Hospital),(s:Specialization) where h.name="Columbia" and s.name="Gynaec" create
(h)-[:Specialized_in]->(s) return h,s;
match(h:Hospital),(s:Specialization) where h.name="Sahyadri" and s.name="Orthopedic"
create (h)-[:Specialized_in]->(s) return h,s;
match(h:Hospital),(s:Specialization) where h.name="Rubi" and s.name="Pediatric" create
(h)-[:Specialized_in]->(s) return h,s;
create (p:Person {name:"Vishal"}) return p;
create (r:Recommend {name:"Service"}) return r;
match(h:Hospital),(r:Recommend) where h.name="Columbia" and r.name="Service" create
(h)-[:Recommend_as]->(r) return h,r;
create (d:Doctor {name:"Aarohi"}) return d;
create (d:Doctor {name:"Rohit"}) return d;
match (d:Doctor),(h:Hospital) where d.name="Rohit" and h.name="Columbia" create (d)-
[:Associated_with]->(h) return d,h;
match (d:Doctor),(h:Hospital) where d.name="Rohit" and h.name="Rubi" create (d)-
[:Associated_with]->(h) return d,h;
match (d:Doctor),(h:Hospital) where d.name="Aarohi" and h.name="Columbia" create (d)-
[:Associated_with]->(h) return d,h;
```

Queries

1)List the names of hospitals with paediatric specialization.

```
match (h:Hospital),(r:Specialization) where r.name="Pediatric" and (h)-[:Specialized_in]->(r)
return r,h;
```

2)List the Names of doctors who are visiting “Ruby Hospital

```
match (h:Hospital),(d:Doctor) where h.name="Rubi" and (d)-[:Associated_with]->(h) return
d.name;
```