
#1.

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:"Sanskriti"}) 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;
```

#2.

Model the following Dairy Brand information as a graph model , and answer the following queries using Cypher. There are various dairy brands like Amul, Go, Britannia, Gokul etc. Their popularity varies across different states in India. The popularity is measured as %, with a high popularity defined as $\geq 90\%$, Medium Popularity between 50 to 90%, and Low popularity $< 50\%$. Each brand manufactures various types of Dairy products like milk, butter, cheese, Curd etc. The milk product can be categorized into Low fat/medium fat or high fat content type.

```
create(b:Brands{name:"Gokul",popularity:"95%"}) return b;
create(b:Brands{name:"Britannia",popularity:"45%"}) return b;
create(b:Brands{name:"Amul",popularity:"75%"}) return b;
```

```
create(p:Products{name:"Cheese"}) return p;
create(p:Products{name:"Curd"}) return p;
create(p:Products{name:"Milk"}) return p;
create(p:Products{name:"Butter"}) return p;
```

```
create(c:Category{name:"Low Fat"}) return c;
create(c:Category{name:"Medium Fat"}) return c;
create(c:Category{name:"High Fat"}) return c;
```

```
match(b:Brands),(p:Products) where b.name="Amul" and
p.name="Cheese" create (b)-[:Has_Products]->(p) return b,p;
match(b:Brands),(p:Products) where b.name="Gokul" and
p.name="Milk" create (b)-[:Has_Products]->(p) return b,p;
```

```
match(p:Products),(c:Category) where p.name="Curd" and
c.name="Medium Fat" create (p)-[:Has_Category]->(c) return p,c;
match(p:Products),(c:Category) where p.name="Milk" and
c.name="Low Fat" create (p)-[:Has_Category]->(c) return p,c;
match(p:Products),(c:Category) where p.name="Cheese" and
c.name="High Fat" create (p)-[:Has_Category]->(c) return p,c;
```

Queries

- 1) List the names of different brands considered in your graph.

```
match(b:Brands) return *;
```
- 2) List the brands that are highly popular in Maharashtra.

```
match(b:Brands) where b.popularity>="90%" return b;
```

#3.

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;
```

#4.

Model the following Furniture Showroom information as a graph model, and answer the queries using Cypher. Consider a Furniture showroom with different types of furnitures like sofas sets, tea tables, cupboards, beds, dining tables, etc. Showroom is divided into different sections, one section for each furniture type, Each section handled by a sales staff. A sales staff can handle one or more sections. Customer may enquire about furniture. An enquiry may result in a purchase by the customer.

```
create(f:Furniture{name:"Sofa Set", color:"white"}) return f;
create(f:Furniture{name:"Tea Table", color:"Black"}) return f;
create(f:Furniture{name:"Cupboards", color:"Brown"}) return f;
create(f:Furniture{name:"beds", color:"white"}) return f;
create(f:Furniture{name:"Dining Table", color:"Black"}) return f;
```

```
create(s:Staff{name:"Satish"}) return s;
create(s:Staff{name:"Suraj"}) return s;
create(s:Staff{name:"Omkar"}) return s;
create(s:Staff{name:"Pratik"}) return s;
```

```
match(s:Staff),(f:Furniture) where s.name="Satish" and
f.name="Sofa Set" create (s)-[:Handled]->(f) return s,f;
match(s:Staff),(f:Furniture) where s.name="Suraj" and
f.name="Tea Table" create (s)-[:Handled]->(f) return s,f;
match(s:Staff),(f:Furniture) where s.name="Omkar" and
f.name="Cupboards" create (s)-[:Handled]->(f) return s,f;
match(s:Staff),(f:Furniture) where s.name="Pratik" and
f.name="beds" create (s)-[:Handled]->(f) return s,f;
```

```
create(c:Customer{name:"Priyanshu"}) return c;
create(c:Customer{name:"Sahil"}) return c;
```

```
match(c:Customer),(f:Furniture) where c.name="Sahil" and
f.name="Sofa Set" create (c)-[:Enquired_about]->(f) return c,f;
match(c:Customer),(f:Furniture) where c.name="Priyanshu" and
f.name="Dining Table" create (c)-[:Enquired_about]->(f) return c,f;
```

Queries

1) List the types of furniture's available in white colour.

```
match(f:Furniture) where f.color="white" return f.name;
```

2) List the sections handled by Mr. Satish.

```
match(f:Furniture),(s:Staff) where s.name="Satish" and (s)-[:Handled]->(f)
return f.name;
```

#5.

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;

```

#6.

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:"Marriott", 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="Marriott" 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;

```