match(n:Check) return case n.name STARTS WITH '肺' when TRUE then 'good' else 'bad' end as result

match(n:Check) where n.name STARTS WITH '肺' return n

match(food:Food) where food.name CONTAINS '花' return food

match (a:Disease) -[:do\_eat]-> (food:Food) return a ,food

match (a:Disease {name:"肺气肿"}) -[:do\_eat]-> (food:Food) return food

match (a:Disease {name:"肺气肿"}) -- (food:Food) return food

match (a:Disease) - [r] -> (b:Food) return a.name, type(r), b.name

match (a) - [r{name:"忌吃"}] -> (b) return a.name, type(r), b.name

match (a) - [r{name:"忌吃"}] -> (b) return count(distinct(a)), count(b)

match (a) - [r] -> (b) where r.name contains '忌' return type(r), count(\*)

match (a) - [r] -> (b) where r.name contains '忌' return type(r), collect(b.name)

match(p:Person) return p.spent, sum(p.time)

match (p:Person), (m:Person) where p.name contains 'patric' and m.name contains "mary" create (p)-[r:REL {name:"friend", year: 12}] ->(m) return p,r,m

match (p:Person), (m:Person) where p.name contains 'Mary' and m.name contains "patric" create (p)-[r:REL {name:"friend", year: 15}] ->(m) return p,r,m

match (a:Person) - [r] -> (b:Person) where r.name contains 'friend' return type(r), sum(r.year)

merge (p:Person {name: "frank"}) return p

match (p:Person) return p order by p.name

match (p:Person) where p.time > 5 with p.name as pname, p.time as psM where psM >10 return pname, psM

match (p:Person) where p.time > 5 with p.name as pname, p.time as psM where psM >10 return pname, psM order by psM limit 3

profile match (p:Person) where p.time > 5 with p.name as pname, p.time as psM where psM >10 return pname, psM order by psM limit 3

match (t1:Trace{uid:332}), (t2:Trace{uid:336}) create (t1)-[r:calls{duration:3, invocations:2 }]->(t2) return t1, r, t2

match (a:Trace) -[r:calls]-> (b:Trace) return a.name,avg(r.duration),b.name