**Neo4j programs**

**HCL database**

create(p:person {name:"Sayali",add:"Pune"})

create(p:person {name:"Vinay",add:"Pune"})

create(p:person {name:"Riya",add:"Pune"})

create(p:person {name:"Shreyas",add:"Pune"})

create(p:person {name:"Nisha",add:"Mumbai"})

create(p:person {name:"Manoj",add:"Mumbai"})

create(p:person {name:"Aadhya",add:"Mumbai"})

create(p:person {name:"Ninad",add:"Kolhapur"})

create(p:person {name:"Vidya",add:"Pune"})

create(p1:place {name:"Pune"})

create(p1:place {name:"Mumbai"})

create(p1:place {name:"Kolhapur"})

create(p2:project {name:"Finnace",add:"HCL"})

create(p2:project {name:"Inventory",add:"HCL"})

create(p2:project {name:"Sales",add:"HCL"})

create relation

-sayli and nisha friends

match (a:person),(b:person)where a.name="Sayali" and b.name="Nisha" create(a)-[:Friend\_of]->(b) return a,b

-sayli and ninad siblings

match (a:person),(b:person)where a.name="Sayali" and b.name="Ninad" create(a)-[:Sister\_of]->(b) return a,b

-sayli mother of riya and shreyas

match (a:person),(b:person)where a.name="Sayali" and b.name="Riya" create(a)-[:Mother\_of]->(b) return a,b

match (a:person),(b:person)where a.name="Sayali" and b.name="Shreyas" create(a)-[:Mother\_of]->(b) return a,b

-vinay of riya and shreyas

match (a:person),(b:person)where a.name="Vinay" and b.name="Riya" create(a)-[:Father\_of]->(b) return a,b

match (a:person),(b:person)where a.name="Vinay" and b.name="Shreyas" create(a)-[:Father\_of]->(b) return a,b

-nisha mother of aadhya

match (a:person),(b:person)where a.name="Nisha" and b.name="Aadhya" create(a)-[:Mother\_of]->(b) return a,b

-manoj father of aadhya

match (a:person),(b:person)where a.name="Manoj" and b.name="Aadhya" create(a)-[:Father\_of]->(b) return a,b

-nisha sister of vidya

match (a:person),(b:person)where a.name="Nisha" and b.name="Vidya" create(a)-[:Sister\_of]->(b) return a,b

/\* to delete node \*/

match(p:person {name:"Aadhya",add:"Mumbai"}) delete p

\*\*\*\*\*\*Queries\*\*\*\*\*\*

1)

match (a:person),(b:person)where (a)-[:Father\_of]->(b) return distinct a.name AS name1 UNION ALL

match (a:person),(b:person)where (a)-[:Mother\_of]->(b) return distinct a.name AS name1

2)

match (a:person),(b:project)where b.name="Finnace" and (a)-[:Works\_in]->(b) return distinct a.name

3)

match (a:person),(b:place)where b.name IN["Pune","Mumbai"] and (a)-[:stays\_in]->(b) return distinct a.name

4)

match (a:person),(b:person)where (a)-[:Mother\_of]->(b) return distinct a.name,b.name

**University database**

neo4j university database

create(u:university{name:"SPPU",location:"Pune"})

create(d:department{name:"comp-sci"})

create(d:department{name:"maths"})

create(d:department{name:"chemistry"})

create(d:department{name:"biology"})

match(u:university),(d:department) where u.name="SPPU" and d.name="comp-sci" create(u)-[:has]->(d) return u,d

match(u:university),(d:department) where u.name="SPPU" and d.name="maths" create(u)-[:has]->(d) return u,d

match(u:university),(d:department) where u.name="SPPU" and d.name="chemistry" create(u)-[:has]->(d) return u,d

match(u:university),(d:department) where u.name="SPPU" and d.name="biology" create(u)-[:has]->(d) return u,d

create(c:course{name:"extracredit"})

create(c:course{name:"bba"})

create(c:course{name:"analyse"})

create(c:course{name:"sports"})

match(c:course),(d:department) where d.name="comp-sci" and c.name="bba" create(d)-[:conducts]->(c) return c,d

match(c:course),(d:department) where d.name="maths" and c.name="bba" create(d)-[:conducts]->(c) return c,d

match(c:course),(d:department) where d.name="maths" and c.name="extracredit" create(d)-[:conducts]->(c) return c,d

match(c:course),(d:department) where d.name="biology" and c.name="extracredit" create(d)-[:conducts]->(c) return c,d

match(c:course),(d:department) where d.name="comp-sci" and c.name="extracredit" create(d)-[:conducts]->(c) return c,d

create(p:people{name:"tanu"})

create(p:people{name:"manu"})

create(p:people{name:"sonu"})

match(p:people),(c:course) where p.name="tanu" and c.name="extracredit" create(p)-[:recommends]->(c) return p,c

match(p:people),(c:course) where p.name="sonu" and c.name="extracredit" create(p)-[:recommends]->(c) return p,c

match(p:people),(c:course) where p.name="manu" and c.name="extracredit" create(p)-[:recommends]->(c) return p,c

match(p:people),(c:course) where p.name="tanu" and c.name="bba" create(p)-[:recommends]->(c) return p,c

match(p:people),(c:course) where p.name="manu" and c.name="analyse" create(p)-[:recommends]->(c) return p,c

QUERIES

with["comp-sci","maths"]as names match(d:drpartment) where d.name in names with collect(d) as s match(c:course) where all(d in s where(d)-[:conducts]->(c)) return c

match(c:course),(d:department) where d.name="maths" and(d)-[:conducts]->(c) return d,c,c.name

match(c:course),(d:department) where d.name="chemistry" and(d)-[:conducts]->(c) return d,c,c.name

match(u:university),(d:department),(c:course) where u.name="SSPU" and(d)-[:conducts]->(c) return d.name

**Mongodb databases**

**Students Database**

db.students.insertMany([

{ Studid: 1, Studname: 'Omakar', Course: 'BCA', Marks: 25 },

{ Studid: 2, Studname: 'Siddhi', Course: 'BBA', Marks: 88 },

{ Studid: 3, Studname: 'Sakshi', Course: 'Mcs', Marks: 81 },

{ Studid: 4, Studname: 'Pritee', Course: 'Mca', Marks: 32 },

{ Studid: 5, Studname: 'Swarup', Course: 'MSc', Marks: 98 },

{ Studid: 6, Studname: 'Tanaya', Course: 'Bsc', Marks: 28 },

{ Studid: 7, Studname: 'Satvik', Course: 'BA', Marks: 87 },

{ Studid: 8, Studname: 'Om', Course: 'Bcom', Marks: 95 },

{ Studid: 9, Studname: 'Pratik', Course: 'Bvoc', Marks: 35 },

{ Studid: 10, Studname: 'Geeta', Course: 'Mcs', Marks: 85 },

]);

db.students.find({ Marks: {$gt: 80} }).count()

db.students.find({ Marks: {$lt: 40} }).count()

**Product Database**

db.Products.insertMany([

{ Productid: 1, Pname:'Iphone 15', Brand: 'Apple', Price: 150000 },

{ Productid: 2, Pname:'Samsung 23', Brand: 'Samsung', Price: 90000 },

{ Productid: 3, Pname:'Oneplus 6', Brand: 'Oneplus', Price: 75000 },

{ Productid: 4, Pname:'Nothing phone 2', Brand: 'Nothing', Price: 60000 },

{ Productid: 5, Pname:'Asus 3', Brand: 'Asus', Price: 120000 },

{ Productid: 6, Pname:'Oppo reno', Brand: 'Oppo', Price: 6000 },

{ Productid: 7, Pname:'Vivo v15', Brand: 'Vivo', Price: 5000 },

{ Productid: 8, Pname:'Redmi Note 5', Brand: 'RedMi', Price: 10000 },

{ Productid: 9, Pname:'Jio541',Brand: 'Jio', Price: 7500 },

{ Productid: 10, Pname:'Moto 7',Brand: 'Motorola', Price: 80000 },

]);

db.Order.insertMany([

{ Orderid: 100, Odate: '12-05-2023', custName: 'Mr.Swarup', Productid: 1, price: 150000 },

{ Orderid: 101, Odate: '18-06-2023', custName: 'Mr.Ritik', Productid: 2, price: 90000 },

{ Orderid: 102, Odate: '19-08-2023', custName: 'Mr.Rafiq', Productid: 3, price: 75000 },

{ Orderid: 103, Odate: '21-04-2023', custName: 'Mr.Jayad', Productid: 4, price: 60000 },

{ Orderid: 104, Odate: '01-05-2023', custName: 'Mr.Sahil', Productid: 7, price: 5000 },

{ Orderid: 105, Odate: '02-09-2023', custName: 'Mr.Aditya', Productid: 5, price: 120000 },

{ Orderid: 106, Odate: '10-12-2023', custName: 'Mr.Anish', Productid: 6, price: 6000 },

{ Orderid: 107, Odate: '11-01-2023', custName: 'Mr.Jagdish', Productid: 9, price: 7500 },

{ Orderid: 108, Odate: '15-02-2023', custName: 'Mrs.Sakshi', Productid: 10, price: 80000 },

{ Orderid: 109, Odate: '29-07-2023', custName: 'Mrs.Pritee', Productid: 8, price: 10000 },

]);

db.Invoice.insertMany([

{ Orderid: 100, custName: 'Mr.Swarup', Productid: 1, Qyt: 2, Total: 300000 },

{ Orderid: 101, custName: 'Mr.Ritik', Productid: 2, Qyt: 1, Total: 90000 },

{ Orderid: 102, custName: 'Mr.Rafiq', Productid: 3, Qyt: 1, Total: 75000 },

{ Orderid: 103, custName: 'Mr.Jayad', Productid: 4, Qyt: 5, Total: 300000 },

{ Orderid: 104, custName: 'Mr.Sahil', Productid: 7, Qyt: 20, Total: 100000 },

{ Orderid: 105, custName: 'Mr.Aditya', Productid: 5, Qyt: 1, Total: 120000 },

{ Orderid: 106, custName: 'Mr.Anish', Productid: 6, Qyt: 5, Total: 30000 },

{ Orderid: 107, custName: 'Mr.Jagdish', Productid: 9, Qyt: 10, Total: 750000 },

{ Orderid: 108, custName: 'Mrs.Sakshi', Productid: 10, Qyt: 5, Total: 400000 },

{ Orderid: 109, custName: 'Mrs.Pritee', Productid: 8, Qyt: 4, Total: 40000 },

]);

db.Products.find({},{Productid:1, Pname:1});

db.Order.find({price:{$gt:10000}});

**Publisher Database**

Assignment 3

db.Book.insertMany([

{ Bookid: 100, BookName: 'DAA', Publisher: 'Vision', PublishedYear: 2017, Writer: 'Mr. Anuj', Cost: 500, Lang: 'English' },

{ Bookid: 101, BookName: 'AI', Publisher: 'Nirali', PublishedYear: 2023, Writer: 'Mrs. Meenal', Cost: 1200, Lang: 'English' },

{ Bookid: 102, BookName: 'Data Science', Publisher: 'Vision', PublishedYear: 2019, Writer: 'Mr. Sameer', Cost: 1800, Lang: 'English' },

{ Bookid: 103, BookName: 'DBT', Publisher: 'Nirali', PublishedYear: 2018, Writer: 'Mrs. Dipali', Cost: 1500, Lang: 'English' },

{ Bookid: 104, BookName: 'Cloud Computing', Publisher: 'Nirali', PublishedYear: 2019, Writer: 'Mr. Satish', Cost: 250, Lang: 'English' },

{ Bookid: 105, BookName: 'Python', Publisher: 'Vision', PublishedYear: 2022, Writer: 'Mr. Harry', Cost: 160, Lang: 'English' },

{ Bookid: 106, BookName: 'Java', Publisher: 'Nirali', PublishedYear: 2017, Writer: 'Mr. Aditya', Cost: 1400, Lang: 'English' },

{ Bookid: 107, BookName: 'C language', Publisher: 'oReily', PublishedYear: 2017, Writer: 'Mr. Jason', Cost: 410, Lang: 'English' },

{ Bookid: 108, BookName: 'PPL', Publisher: 'Vision', PublishedYear: 2017, Writer: 'Mr. Ranjana', Cost: 200, Lang: 'English' },

{ Bookid: 109, BookName: '.Net', Publisher: 'Mehta', PublishedYear: 2017, Writer: 'Mr. sumit', Cost: 320, Lang: 'English' },

])

db.Publisher.insertMany([

{ Pubid: 200, PubName: 'Nirali', add: 'Pune', age: 50 },

{ Pubid: 201, PubName: 'Vision', add: 'Mumbai', age: 75 },

{ Pubid: 202, PubName: 'Mehta', add: 'Mumbai', age: 10 },

{ Pubid: 203, PubName: 'oReily', add: 'Pune', age: 20 },

])

db.Publisher.find({ add: 'Mumbai' }, { \_id: 0, PubName: 1 });

db.Book.find({ Cost: { $gt: 1000} }, { \_id: 0, BookName: 1 });

**Hospital Database**

,specialist:['Allergist','Orthopedist'],visits : {monday : ['Rakshak Hospital','Ruby hall Hospital'],tuesday : ['Aditya Birla Hospital']}},

{dID : 216,dname : 'Dr. Jagtap' ,specialist:['Orthopedist'],visits : {monday : ['Jahangir Hospital'],tuesday : ['Dinanath Hospital']}},

{dID : 217,dname : 'Dr. Deshmukh' ,specialist:['Paediatric'],visits :{friday : ['Aditya Birla Hospital'],wednesday : ['Jupiter Hospital','Umang Hospital'],thursday:['Sahyadri Hospital']}},

{dID : 218,dname : 'Dr. Ambike' ,specialist:['Orthopedist'],visits : {thursday : ['Poona Hospital'],wednesday : ['Rakshak Hospital','Aditya Birla Hospital']}},

{dID : 219,dname : 'Dr. Modi' ,specialist:['Orthopedist'],visits : {monday : ['Aditya Birla Hospital'],wednesday : ['Yash Hospital','Umang Hospital'],thursday:['Sahyadri Hospital']}},

{dID : 220,dname : 'Dr. Pratik' ,specialist:['Paediatric'],visits : {tuesday : ['Poona Hospital'],wednesday : ['Dinanath Hospital','Aditya Birla Hospital']}}

])

db.hospital.find({Spec:'Paediatric'},{Hname:1,\_id:0});

db.doctor.find({'visits.monday': 'Jahangir Hospital'}, { dname: 1, \_id: 0 });