**Neo4j Database Solution**

**Database1 = Song**

**1. Consider a Song database, with labels as Artists, Song, Recording\_company,**

**Recoding\_studio, song author etc.**

**Relationships can be as follows:**

**Artist → [Performs] → Song → [Written by] → Song\_author.**

**Song → [Recorded in] → Recording Studio →[managed by] → Recording\_**

**Company**

**Recording Company → [Finances] → Song**

**Solve the Following Queries:**

**1. List the names of artist performing the song “....”**

**2. Name the songs recorded by the studio “.......”**

**3. List the names of songs written by “ .....”**

**4. List the names of record companies who have financed for the song“....”**

**Creating Label Arsist**

create(Artist1:a1{name:'Shreya Ghoshal',age:25,addr:'Pune'}) return((Artist1))

create(Artist2:a2{name:'Aasha Bhosale',age:55,addr:'Mumbai'}) return((Artist2))

create(Artist3:a3{name:'Abhijeet Sawant',age:35,addr:'Nagar'}) return((Artist3))

create(Artist4:a4{name:'Aarya Ambekar',age:24,addr:'Nagar'}) return((Artist4))

**Creating Label Song**

create(Song1:s1{name:'Song1',lyrics:'Abcd..'}) return(Song1)

create(Song2:s2{name:'Song2',lyrics:'xyz..'}) return(Song2)

create(Song3:s3{name:'Song3',lyrics:'pqrs..'}) return(Song3)

create(Song4:s4{name:'Song4',lyrics:'tlys'}) return(Song4)

**Creating Label Recording\_Company**

create(Recording\_Company1:rc1{name:'T\_Series',Est:2010}) return(Recording\_Company1)

create(Recording\_Company2:rc2{name:'HMV',Est:2012}) return(Recording\_Company2)

create(Recording\_Company3:rc3{name:'Sangeet',Est:2011}) return(Recording\_Company3)

create(Recording\_Company4:rc4{name:'Sureel',Est:2013}) return(Recording\_Company4)

**Creating Label Recording\_Studio**

create(Recording\_Studio1:rs1{name:'R K',location:'Pune'}) return(Recording\_Studio1)

create(Recording\_Studio2:rs2{name:'R J',location:'Delhi'}) return(Recording\_Studio2)

create(Recording\_Studio3:rs3{name:'Saa',location:'Pune'}) return(Recording\_Studio3)

create(Recording\_Studio4:rs4{name:'Sunshine',location:'Mumbai'}) return(Recording\_Studio

4)

**Creating Label Song\_Author**

create(Song\_Author1:sa1{name:'Nitin',addrs:'Pune'}) return(Song\_Author1)

create(Song\_Author2:sa2{name:'Seema',addrs:'Pune'}) return(Song\_Author2)

create(Song\_Author3:sa3{name:'Resham',addrs:'Nagpur'}) return(Song\_Author3)

create(Song\_Author4:sa4{name:'sameer',addrs:'Delhi'}) return(Song\_Author4)

**Creating Relationships:**

**Artist → [Performs] → Song → [Written by] → Song\_author**

**Song → [Recorded in] → Recording Studio →[managed by] → Recording\_Company**

**Recording Company → [Finances] → Song**

1. **Artist – Performs 🡪 Song – Written\_By 🡪 Author**

Match(a:a1{name:'Shreya Ghoshal'}),(b:s1{name:'Song1'}),(c:sa1{name:'Nitin'}) create (a)-[r1:Performs]->(b)-[r2:Written\_By]->(c) return r1,r2

Match(a:a2{name:'Aasha Bhosale'}),(b:s2{name:'Song2'}),(c:sa2{name:'Seema'}) create (a)-[r3:Performs]->(b)-[r4:Written\_By]->(c) return r3,r4

Match(a:a2{name:'Aasha Bhosale'}),(b:s2{name:'Song2'}),(c:sa3{name:'Resham'}) create (a)-[r5:Performs]->(b)-[r6:Written\_By]->(c) return r5,r6

1. **Song → [Recorded in] → Recording Studio →[managed by] → Recording\_Company**

Match (a:s1{name:'Song1'}),(b:rs1{name:'R K'}), (c:rc1{name:'T\_Series'}) create (a)-[r1:Recorded\_in]->(b)-[r2:Managed\_By]->(c) return r1,r2

Match (a:s2{name:'Song2'}),(b:rs1{name:'R K'}), (c:rc2{name:'HMV'}) create (a)-[r1:Recorded\_in]->(b)-[r2:Managed\_By]->(c) return r1,r2

Match (a:s3{name:'Song3'}),(b:rs2{name:'R J'}), (c:rc3{name:'Sangeet'}) create (a)-[r1:Recorded\_in]->(b)-[r2:Managed\_By]->(c) return r1,r2

1. **Recording Company → [Finances] → Song**

Match (a:rc1{name:'T\_Series'}),(b:s1{name:'Song1'}) create (a)-[r1:Finances]->(b)  return r1

Match (a:rc1{name:'T\_Series'}),(b:s2{name:'Song2'}) create (a)-[r1:Finances]->(b)  return r1

Match (a:rc2{name:'HMV'}),(b:s3{name:'Song3'}) create (a)-[r1:Finances]->(b)  return r1

Match (a:rc4{name:'Sureel'}),(b:s4{name:'Song4'}) create (a)-[r1:Finances]->(b)  return r1

**Queries**

1. List the names of artist performing the song “Song1”

Match (a)-[:Performs]->(:s1{name:'Song1'}) return a

1. Name the songs recorded by the studio “R K”

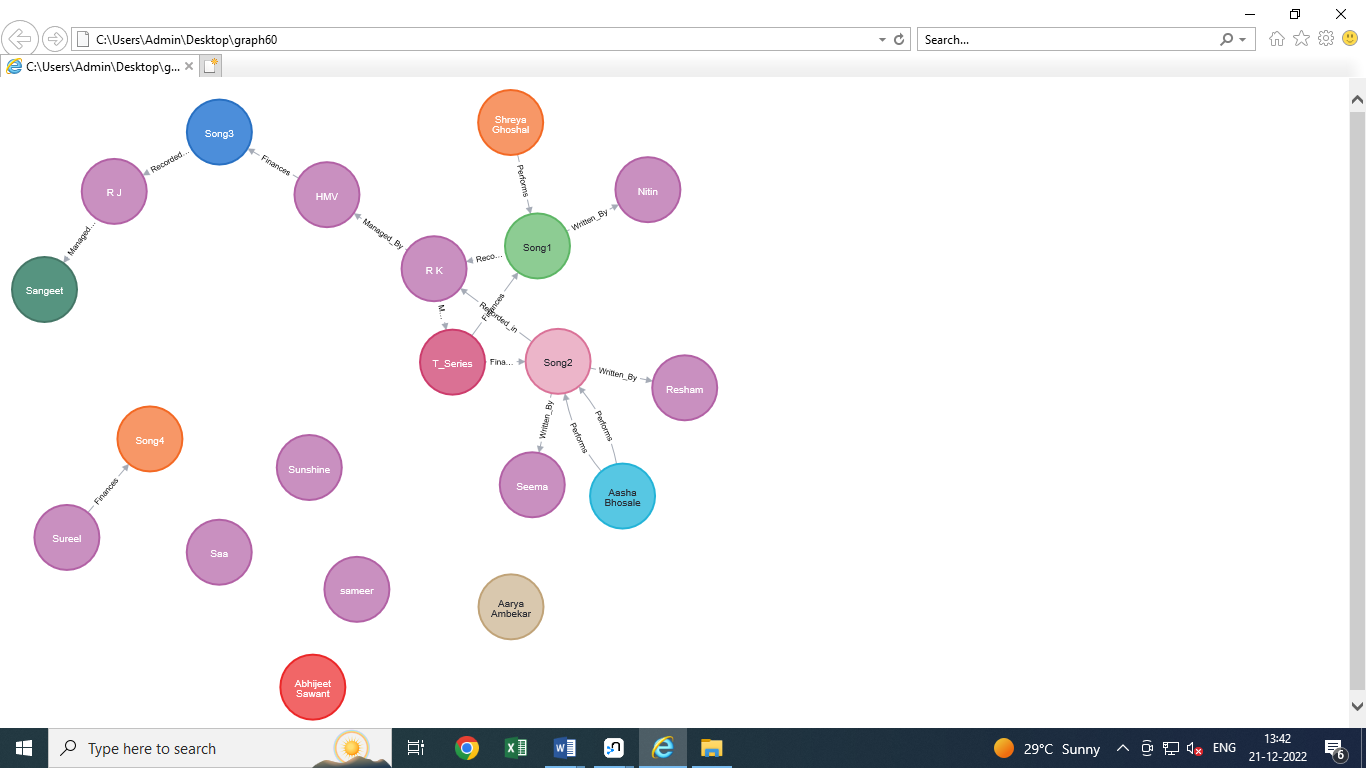
Match (a)-[:Recorded\_in]->(b:rs1{name:'R K'}) return a

1. List the names of songs written by “ Nitin”

Match (a)-[:Written\_By]->(b:sa1{name:'Nitin'}) return a

1. List the names of record companies who have financed for the song “Song1”

Match (a)-[:Finances]->(:s1{name:'Song1'}) return a



**Database2 = Employee**

**2. Solve the following database using Neo4j.**

**Consider an Employee database, with a minimal set of labels as follows**

**Employee: denotes a person as an employee of the organization**

**Department: denotes the different departments, in which employees work.**

**Skillset: A list of skills acquired by an employee**

**Projects: A list of projects in which an employee works.**

**A minimal set of relationships can be as follows:**

**Works\_in : employee works in a department**

**Has\_acquired: employee has acquired a skill**

**Assigned\_to : employee assigned to a project**

**Controlled\_by: A project is controlled by a department**

**Project\_manager: Employee is a project\_manager of a Project**

**Solve the Following Queries:**

**1. List the names of employees in department “... .................. ”**

**2. List the projects along with their properties, controlled by department “......”**

**3. List the skillset for an employee “ ”**

**4. List the names of the projects managed by employee “ ”**

**5. List the departments along with the count of employees in it.**

**Creating Label Employee**

create(Employee:e1{name:'Sameer',age:28,addrs:'Pune'}) return(Employee)

create(Employee:e2{name:'Seema',age:25,addrs:'Pune'}) return(Employee)

create(Employee:e3{name:'Nisha',age:26,addrs:'Mumbai'}) return(Employee)

create(Employee:e4{name:'Neeta',age:27,addrs:'Nagar'}) return(Employee)

**Creating Label Department**

create(Department:d1{id:1,name:'Computer Science',location:'First Floor'}) return Department

create(Department:d2{id:2,name:'Account',location:'First Floor'}) return Department

create(Department:d3{id:3,name:'Mathematics',location:'M1'}) return Department

create(Department:d4{id:4,name:'Chemistry',location:'Second Floor'}) return Department

**Creating Label Skillset**

create(Skillset:s1{name:'s1',skills:['Communication','Reasoning','Verbal']}) return Skillset

create(Skillset:s2{name:'s2',skills:['Communication','Reasoning']}) return Skillset

create(Skillset:s3{name:'s3',skills:['Communication','Reasoning','Logical']}) return Skillset

create(Skillset:s4{name:'s4',skills:['Communication','Logical']}) return Skillset

**Creating Label Project**

create(Project:p1{name:'Robotics',duration:3}) return Project

create(Project:p2{name:'Alexa',duration:4}) return Project

create(Project:p3{name:'BMS',duration:2}) return Project

create(Project:p4{name:'HMS',duration:3}) return Project

**Creating Relationships:**

**Works\_in : employee works in a department**

Match(a:e1{name:'Sameer'}),(b:d1{id:1}) create (a)-[r:Works\_in]->(b) return r

Match(a:e2{name:'Seema'}),(b:d2{id:2}) create (a)-[r:Works\_in]->(b) return r

Match(a:e3{name:'Nisha'}),(b:d2{id:2}) create (a)-[r:Works\_in]->(b) return r

Match(a:e3{name:'Nisha'}),(b:d3{id:3}) create (a)-[r:Works\_in]->(b) return r

Match(a:e4{name:'Neeta'}),(b:d4{id:4}) create (a)-[r:Works\_in]->(b) return r

**Has\_acquired: employee has acquired a skill**

Match(a:e4{name:'Neeta'}),(b:s1{name:'s1'}) create (a)-[r:Has\_acquired]->(b) return r

Match(a:e3{name:'Nisha'}),(b:s3{name:'s3'}) create (a)-[r:Has\_acquired]->(b) return r

Match(a:e4{name:'Neeta'}),(b:s4{name:'s4'}) create (a)-[r:Has\_acquired]->(b) return r

Match(a:e2{name:'Seema'}),(b:s2{name:'s2'}) create (a)-[r:Has\_acquired]->(b) return r

**Assigned\_to : employee assigned to a project**

Match(a:e1{name:'Sameer'}),(b:p2{name:'Alexa'}) create (a)-[r:AssignedTo]->(b) return r

Match(a:e1{name:'Sameer'}),(b:p1{name:'Robotics'}) create (a)-[r:AssignedTo]->(b) return r

Match(a:e3{name:'Nisha'}),(b:p3{name:'BMS'}) create (a)-[r:AssignedTo]->(b) return r

Match(a:e4{name:'Neeta'}),(b:p4{name:'HMS'}) create (a)-[r:AssignedTo]->(b) return r

**Controlled\_by: A project is controlled by a department**

Match (a:p1{name:'Robotics'}),(b:d1{name:'Computer Science' }) create (a)-[r:ControlledBy]->(b) return r

Match (a:p2{name:'Alexa'}),(b:d3{name:'Mathematics' }) create (a)-[r:ControlledBy]->(b) return r

Match (a:p3{name:'BMS'}),(b:d3{name:'Mathematics' }) create (a)-[r:ControlledBy]->(b) return r

Match (a:p4{name:'HMS'}),(b:d2{name:'Account' }) create (a)-[r:ControlledBy]->(b) return r

**Project\_manager: Employee is a project\_manager of a Project**

Match(a:e1{name:'Sameer'}),(b:p2{name:'Alexa'}) create (a)-[r:Project\_Manager\_of]->(b) return r

Match(a:e1{name:'Sameer'}),(b:p1{name:'Robotics'}) create (a)-[r:Project\_Manager\_of]->(b) return r

Match(a:e3{name:'Nisha'}),(b:p3{name:'BMS'}) create (a)-[r:Project\_Manager\_of]->(b) return r

Match(a:e4{name:'Neeta'}),(b:p4{name:'HMS'}) create (a)-[r:Project\_Manager\_of]->(b) return r

**Quries :**

1. List the names of employees in department “Account”

Match (a)-[:Works\_in]->(:d2{name:'Account'}) return a.name

1. List the projects along with their properties, controlled by department “Mathematics”

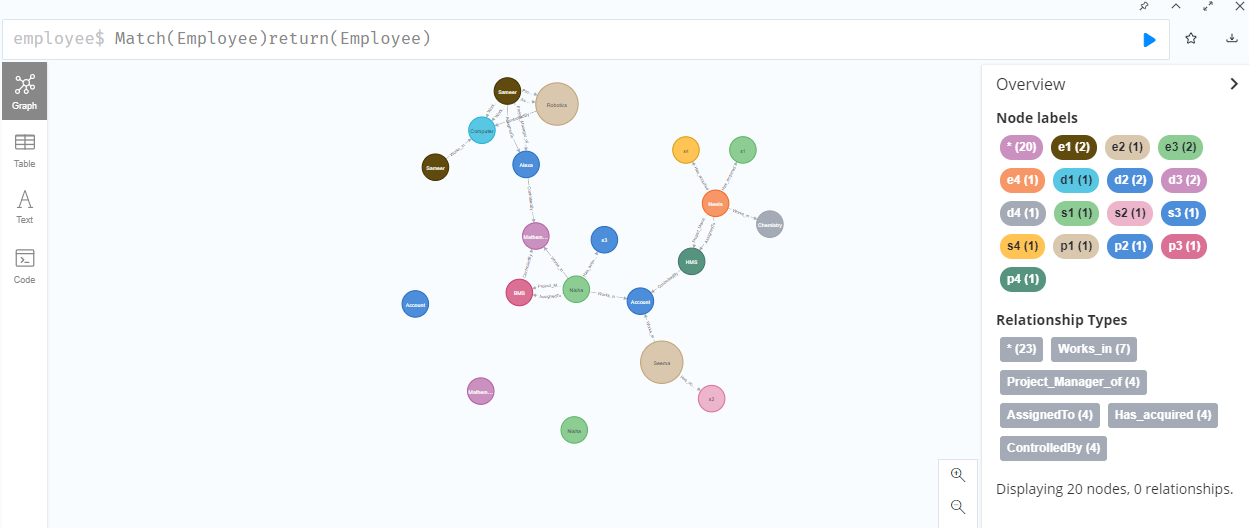
Match (a)-[:ControlledBy]->(b:d3{name:'Mathematics'}) return a

1. List the skillset for an employee “Nisha”
2. Match (a:e3{name:'Nisha'})-[:Has\_acquired]->(b) return b.skills
3. List the names of the projects managed by employee “Sameer”

Match(a:e1{name:'Sameer'})-[:Project\_Manager\_of]->(b) return b.name

1. List the departments along with the count of employees in it.

Match (a)-[:Works\_in]->(b) return b.name,count(a)



**Database 3 : Library**

1. **Solve the following database using Neo4j.**

**Create a library database, as given below.**

**There are individual books, readers, and authors that are**

**Present in the library data model.**

**A minimal set of labels are**

**as follows:**

**Book: This label includes all the books**

**Person: This label includes authors, Readers, Suppliers and so on**

**Publisher: This label includes the publishers of books in the database**

**A set of basic relationships are as follows:**

**Published By:**

**Issued By:**

**Returned By:**

**Person Label**

create(author:Person{name:"John Le Carre",born:19-10-1932})

create(reader:Person{name:"lan"})

create(reader:Person{name:"Alan"})

create(supplier:Person{name:"Rahul"})

**Book Label**

create(book1:Book{name:"Story Book",title:["Tinker","Tailor","Soldier","spy"],published:1974,city:"Pune"})

create(book2:Book{name:"Suspens Story Book",title:"One Man in Havana",published:1958,city:'Mumbai'})

**Publisher Label**

create(publisher1:Publisher{name:'Mr.Joshi',City:'Pune'})

create(publisher2:Publisher{name:'Mr.Mohan',City:'Mumbai'})

match(n) return n

**Relationships:**

match(a:Publisher{name:"Mr.Joshi" }), (b:Book{name:"Story Book"})  create (a)-[r1:published\_by]->(b)

match(a:Publisher{name:"Mr.Joshi" }),(b:Book{name:"Suspens Story Book" })  create (a)-[r2:published\_by]->(b)

match(a:Publisher{name:'Mr.Mohan'  }), (b:Book{name:"Story Book"}) create (a)-[r3:published\_by]->(b)

match(a:Person{name: "John Le Carre" }), (b:Book{name: "Story Book" }) create (a)-[r1:WROTE]->(b)

match(a:Person{name: "Alan" }), (b:Book{name: "Story Book" }) create (a)-[r4:Issued\_by]->(b) return a,b

match(a:Person{name:"lan"}),(b:Book{name:"Suspens Story Book"} )  create (a)-[r5:Issued\_by]->(b)

match(a:Person{name:"Alan}),(b:Book{name:"Story Book"})  create (a)-[r2:RECOMMENDED{date:5-7-2011}]->(b)

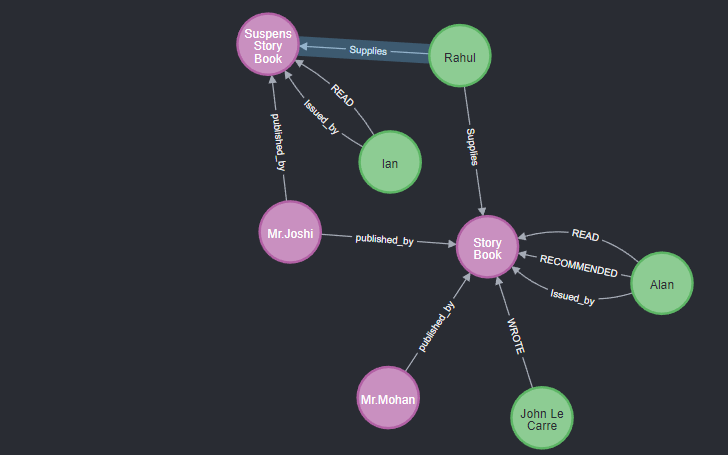
match(a:Person),(b:Book) where a.name:"Alan" and b.name:"Story Book" create (a)-[r3:READ{date:9-9-2011}]->(b)

match(a:Person),(b:Book) where a.name:"lan" and b.name:"Suspens Story Book" create (a)-[r3:READ{date:9-9-2011}]->(b)

match(a:Person),(b:Book) where a.name:"Rahul" and b.name:"Story Book" create (a)-[r3:Supplies]->(b)

match(a:Person),(b:Book) where a.name:"Rahul" and b.name:"Suspens Story Book" create (a)-[r3:Supplies]->(b)

match (Library) return Library



**Queries**

**1] List all people, who have issued a book “Our Man in Havana”.**

MATCH (rd:Person)-[r:Issued\_by]->(b:Book)

WHERE b.title='One Man in Havana'

RETURN rd.name

OUTPUT :

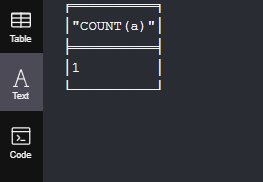
**2] Count the number of people who have read “….”**

MATCH (a:Person)-[r:READ]->(b:Book)

WHERE b.title="One Man in Havana"

RETURN COUNT(a)

OUTPUT :



**3] Add a property**

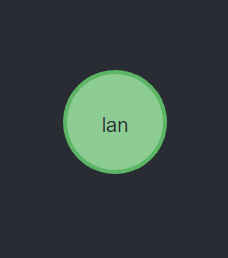
**“Number of books issued “ for Mr. Joshi and set its value as the count**

MATCH (ir:Person{name:'lan'})

SET ir.No\_of\_Issued=1

RETURN ir

OUTPUT :

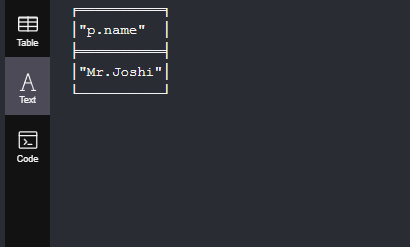


**4] List the names of publishers from pune city**

match(p:Publisher{City:'Pune'})

return p.name

OUTPUT :



**5:List all readers who have recommended either book "Tinker","Tailor","Soldier","spy"] or "One Man in Havana"**

MATCH (a:Person)-[r:RECOMMENDED]->(b:Book) WHERE b.title=["Tinker","Tailor","Soldier","spy"] OR b.title="One Man in Havana" RETURN a

OUTPUT :



**6: List the readers who haven’t recommended any book**

MATCH (a:Person) WHERE NOT (a:Person)-[:RECOMMENDED]->(:Book) return a

**7: List the reader names who have read/ issued books and display their count.**

MATCH (a:Person)-[r:Issued\_by]->(b:Book) RETURN a.name,b.title,COUNT(b)

**8: List the names of voracious readers in our library**

MATCH (a:Person) WHERE NOT(a:Person)-[:Issued\_by]->(:Book) RETURN a.name