

//Build a graph model:

// Create nodes: Job / Employee

// load jobs data: create each job as a node, with property: jobID,degree,location,technicalSkills,preferredSkills, jobTitles

load csv with headers from

'file:///job_data.csv' as line

merge(j:Job

{jobID:line.jobID,degree:line.degree,location:line.location,technicalSkills:line.technicalSkills,preferredSkills:line.preferredSkills,jobTitle:line.jobTitle})

return j

// load employee data: create each employee as a node, with property:employeeID,degree,location,technicalSkills

load csv with headers from

'file:///employee_data.csv' as line

merge(e:Employee {employeeID: line.employeeID,degree:line.degree,location:line.location,technicalSkills:line.technicalSkills})

return e

// Create relationships:

// create applied_to relationship

load csv with headers from

'file:///appliedJobs.csv' as line

match (e:Employee {employeeID: line.employeeID})

match (j:Job {jobID: line.jobID})

merge (e)-[op:APPLIED_TO]->(j)

return e,j,op

// create work_at relationship

load csv with headers from

'file:///previousJobs.csv' as line

match (e:Employee {employeeID: line.employeeID})

match (j:Job {jobID: line.jobID})

merge (e)-[op:WORKED_AT]->(j)

return e,j,op

```
// create searched_for relationship
load csv with headers from
'file:///searchedJobs.csv' as line
match (e:Employee {employeeID: line.employeeID})
match (j:Job {jobID: line.jobID})
merge (e)-[op:SEARCHED_FOR]->(j)
return e,j,op

// create connected relationship
load csv with headers from
'file:///network.csv' as line
match (e0:Employee {employeeID: line.employeeID})
match (e1:Employee {employeeID: line.connectedEmployeeID})
merge (e0)-[:CONNECTED_TO {score:line.score}]->(e1)
return e0,e1
```