Neo4j Work: Importing a CSV file into Neo4j

Prepared by: Mert OLÇAMAN

Week: 4

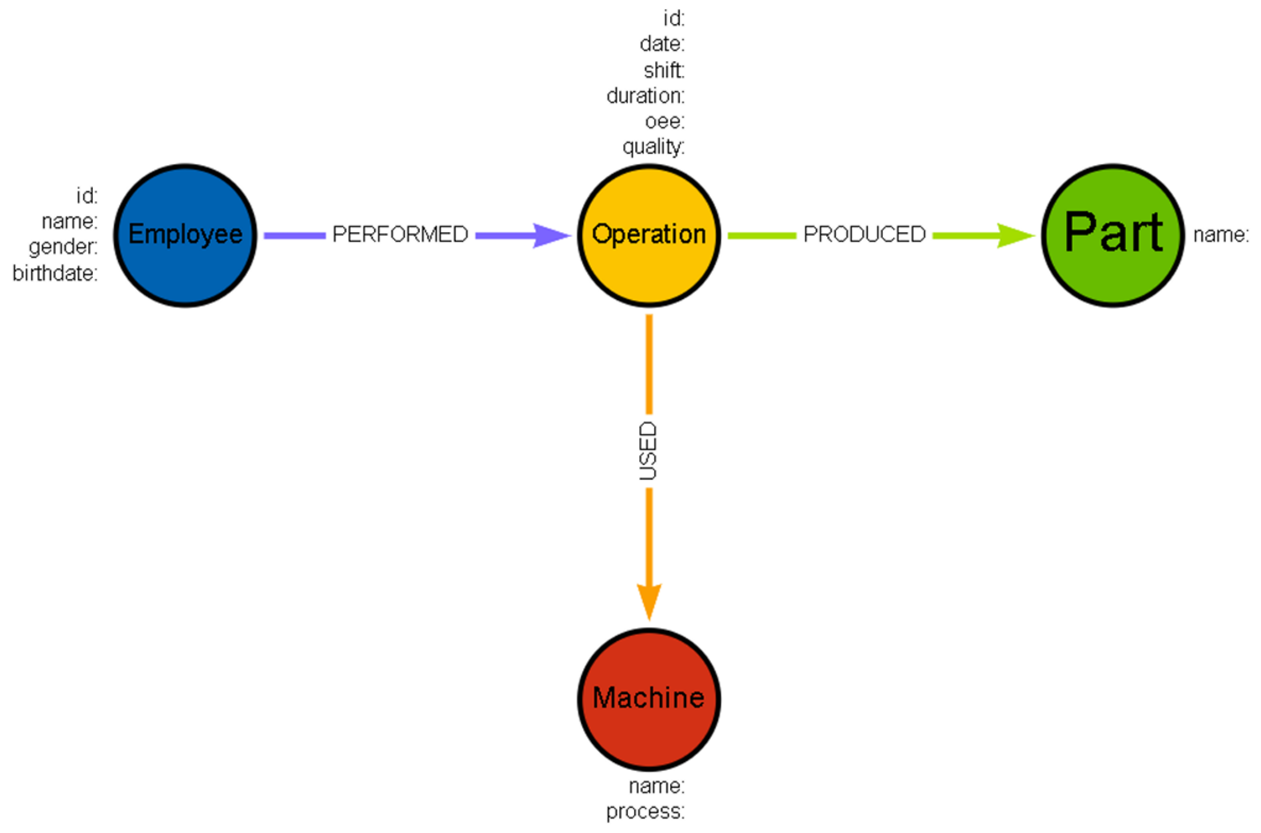
patika.dev

NewMind AI Bootcamp

INTRODUCTION

In this work, how to import a csv file into Neo4j Aura (cloud version) was shown in 2 ways. The dataset and the graph database design can be seen below respectively.

EmployeeID	Name	Gender	DateOfBirth	OperationID	Part	Process	Shift	Machine	Date	Duration	OEE	Quality
Employee1	Danielle Johnson	M	17/01/1983	Operation1	Piston	Milling	Day	M2	26/04/2025	64	0.68	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation19	Piston	Assembly	Morning	M3	30/04/2025	38	0.73	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation25	Gearbox	Drilling	Morning	M1	25/04/2025	31	0.61	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation32	Shaft	Drilling	Morning	M1	29/04/2025	85	0.77	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation41	Bearing	Painting	Night	M4	25/04/2025	179	0.63	Low
Employee1	Danielle Johnson	M	17/01/1983	Operation60	Valve	Milling	Day	M2	25/04/2025	31	0.83	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation78	Piston	Painting	Morning	M4	29/04/2025	171	0.7	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation81	Bearing	Painting	Night	M4	28/04/2025	126	0.63	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation106	Valve	Milling	Morning	M2	29/04/2025	176	0.79	High
Employee1	Danielle Johnson	M	17/01/1983	Operation169	Shaft	Assembly	Morning	M3	30/04/2025	106	0.88	Low
Employee1	Danielle Johnson	M	17/01/1983	Operation170	Gearbox	Milling	Day	M2	24/04/2025	146	0.68	Medium
Employee1	Danielle Johnson	M	17/01/1983	Operation171	Valve	Milling	Morning	M2	25/04/2025	40	0.75	Low
Employee1	Danielle Johnson	M	17/01/1983	Operation234	Shaft	Milling	Day	M2	30/04/2025	89	0.82	Low
Employee2	John Taylor	M	22/04/1971	Operation3	Bearing	Painting	Morning	M4	26/04/2025	176	0.73	High
Employee2	John Taylor	M	22/04/1971	Operation33	Piston	Milling	Morning	M2	28/04/2025	178	0.69	Medium
Employee2	John Taylor	M	22/04/1971	Operation36	Bearing	Painting	Night	M4	25/04/2025	45	0.94	Low
Employee2	John Taylor	M	22/04/1971	Operation77	Piston	Milling	Day	M2	27/04/2025	165	0.8	High
Employee2	John Taylor	M	22/04/1971	Operation115	Valve	Milling	Night	M2	30/04/2025	65	0.9	Low
Employee2	John Taylor	M	22/04/1971	Operation125	Gearbox	Painting	Day	M4	24/04/2025	60	0.64	Medium



a) through the link

At first, the generated csv file was uploaded to github directory. Secondly, it was directly imported by using function.

Step1 - Adding Constraints: Before starting everything, constraints should be specified. Otherwise, there might be some improper nodes, which causes wrong results.

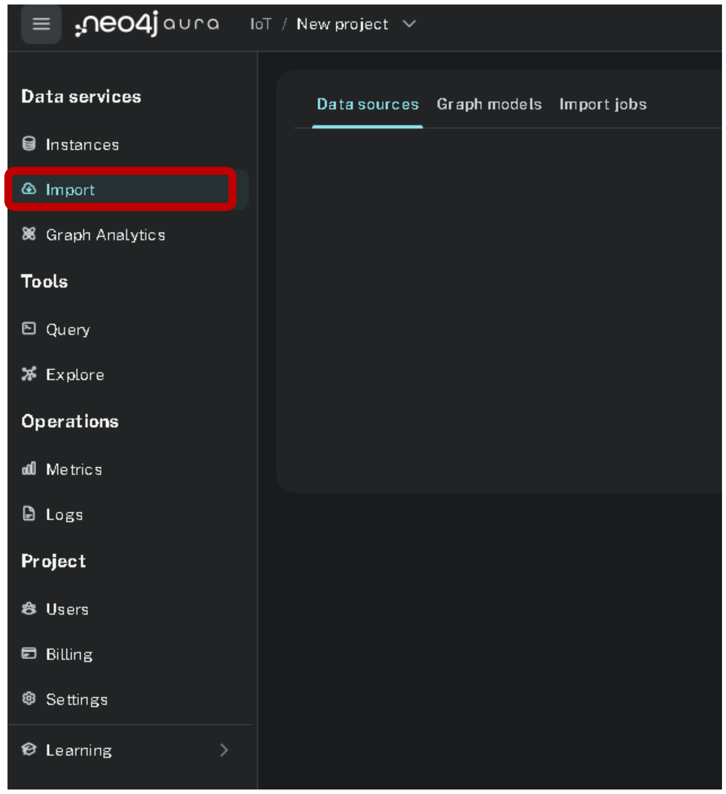
```
// --- ADDING CONSTRAINTS --- \\

// Employee constraint
CREATE CONSTRAINT employee_id IF NOT EXISTS
FOR (e:Employee)
REQUIRE e.employee_id IS UNIQUE;

// Operation constraint
CREATE CONSTRAINT operation_id IF NOT EXISTS
FOR (o:Operation)
REQUIRE o.operation_id IS UNIQUE;

// Part constraint
CREATE CONSTRAINT part_name IF NOT EXISTS
FOR (p:Part)
REQUIRE p.part_name IS UNIQUE;

// Machine constraint
CREATE CONSTRAINT machine_name IF NOT EXISTS
FOR (m:Machine)
REQUIRE m.machine_name IS UNIQUE;
```

Step2 – Loading CSV Files & Creating Nodes with Properties & Adding Relationships: While loading data from the link, an alias should be specified to use in the other parts of the cypher query. It was specifed as **data** in this work.
For each node created, the unique property has to be specified in the merge line as a property. As for Employee node, only employee_id was set as unique property, so it was the only property which was provided in the merge, as it can be seen in the code cell below. If there had been multiple unique properties, they would have been specifed as well. Additionally, data alias was used to call the relevant data from the imported dataset by using the same column name as the imported dataset had. What it is meant here that EmployeeID was the name of the column in the dataset. It is valid for the other variables which start with the alias of data such as data.Name, data.Gender, and data.Shift.
On the other hand, the numerical values like duration must be converted into float by toFloat() function while calling the data from the csv file. The column of date was converted to date, as the format was only data (time not included).

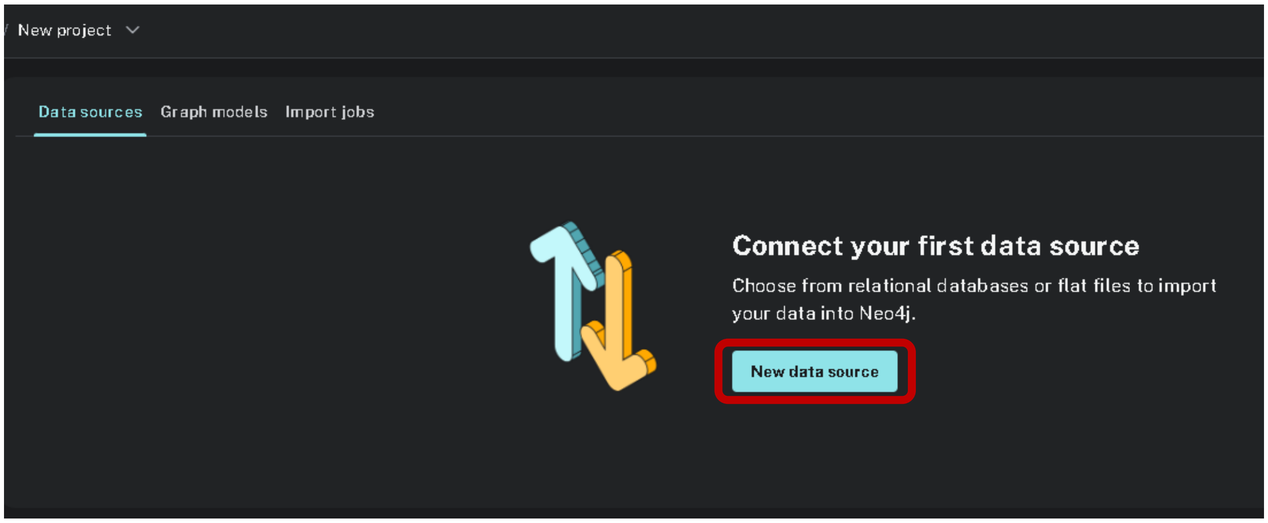
```
// --- LOADING CSV FILE --- \\
LOAD CSV WITH HEADERS FROM
'https://media.githubusercontent.com/media/mertolcaman/patika_newmind_ai_bootcamp/refs/head
s/main/4.week/mydata.csv' as data

// --- CREATING NODES WITH PROPERTIES--- \\

// Creating Employee Node
MERGE (e:Employee {employee_id: data.EmployeeID})
SET e.name = data.Name,
e.gender = data.Gender,
e.birthdate = data.DateOfBirth

// Creating Operation Node
MERGE (o:Operation {operation_id: data.OperationID})
SET o.date = date(data.Date),
o.shift = data.Shift,
o.duration = toFloat(data.Duration),
o.oee = toFloat(data.OEE),
o.quality = data.Quality

// Creating Machine Node
MERGE (m:Machine {machine_name: data.Machine})
SET m.process = data.Process

// Creating Part Node
MERGE (p:Part {part_name: data.Part})

// --- CREATING RELATIONSHIPS --- \\

// Creating PERFORMED Relationship: Employee -> Operation
MERGE (e)-[:PERFORMED]->(o)

// Creating USED Relationship: Operation -> Machine
MERGE (o)-[:USED]->(m)

// Creating PRODUCED Relationship: Operation -> Person
MERGE (o)-[:PRODUCED]->(p);
```

b)through the import section in Neo4j Aura

The second way to import the data into Neo4j is to use the import section in the menu left-hand side as it is shown below.



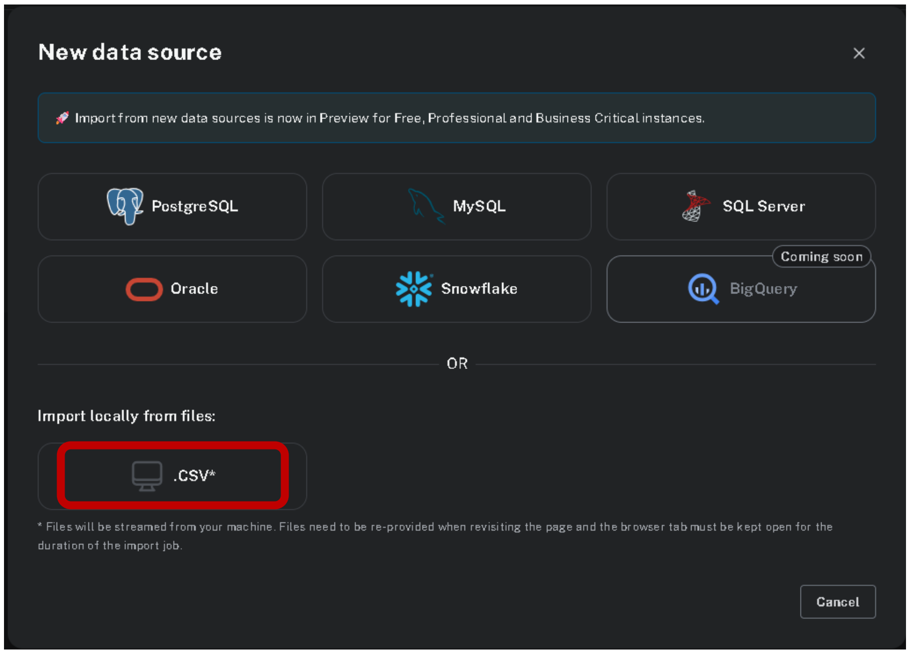
Step-1: After choosing import section, new data source is chosen.



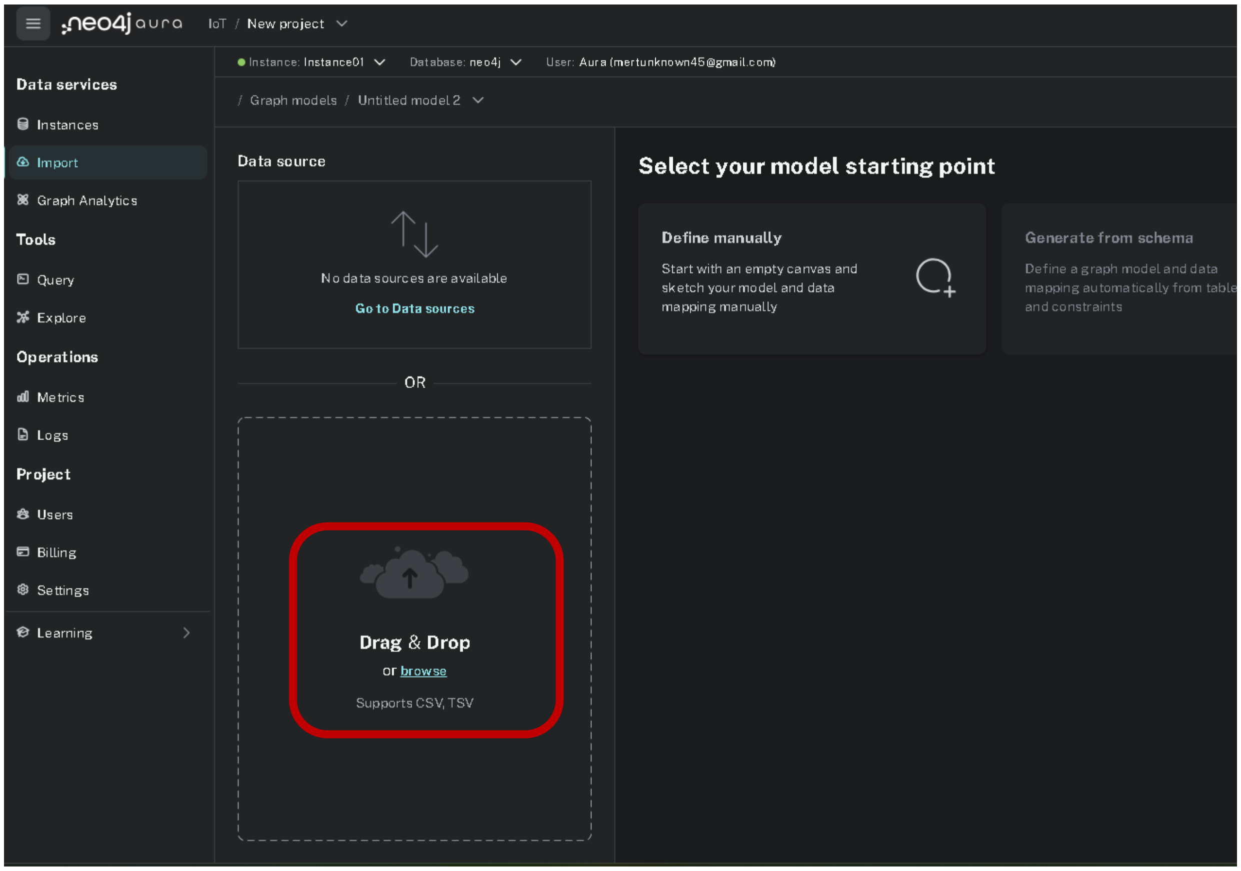
Connect your first data source

Choose from relational databases or flat files to import your data into Neo4j.

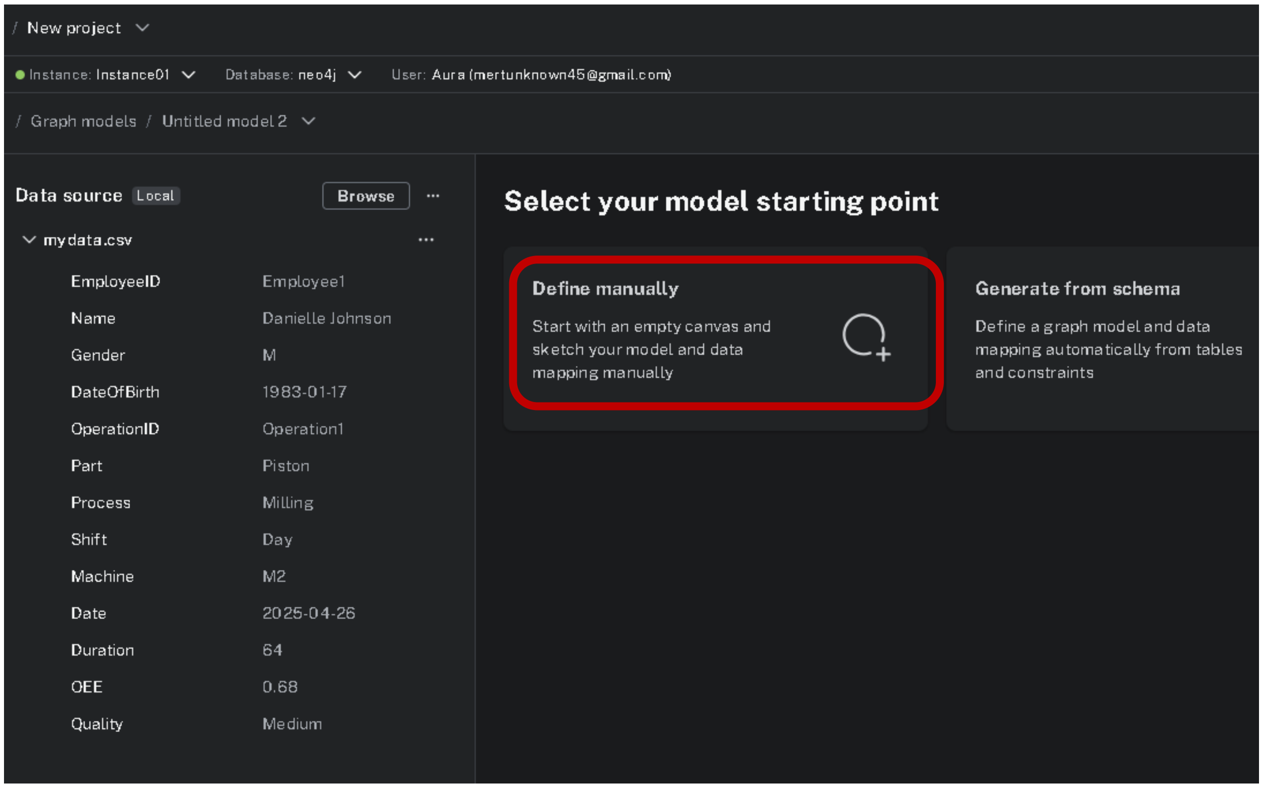
Step-2: Since the csv file would be uploaded, .csv* option is chosen.


New data source
Import from new data sources is now in Preview for Free, Professional and Business Critical instances.
PostgreSQL
MySQL
SQL Server
Oracle
Snowflake
Coming soon
BigQuery
OR
Import locally from files:
.CSV*
* Files will be streamed from your machine. Files need to be re-provided when revisiting the page and the browser tab must be kept open for the duration of the import job.
Cancel

Step-3: Csv file which would like to be uploaded is found by either browsing or dragging into the window.


neo4j aura
IoT / New project
Instance: Instance01
Database: neo4j
User: Aura (mertunknown45@gmail.com)
/ Graph models / Untitled model 2
Data services
Instances
Import
Graph Analytics
Tools
Query
Explore
Operations
Metrics
Logs
Project
Users
Billing
Settings
Learning
Data source
No data sources are available
Go to Data sources
OR
Drag & Drop
or browse
Supports CSV, TSV
Select your model starting point
Define manually
Start with an empty canvas and sketch your model and data mapping manually
Generate from schema
Define a graph model and data mapping automatically from table and constraints

Step-4: Since all of the nodes, properties, and relationship would be specified manually, define manually option is clicked.



/ New project

Instance: Instance01 Database: neo4j User: Aura (mertunknown45@gmail.com)

/ Graph models / Untitled model 2

Data source Local Browse

mydata.csv

EmployeeID	Employee1
Name	Danielle Johnson
Gender	M
DateOfBirth	1983-01-17
OperationID	Operation1
Part	Piston
Process	Milling
Shift	Day
Machine	M2
Date	2025-04-26
Duration	64
OEE	0.68
Quality	Medium

Select your model starting point

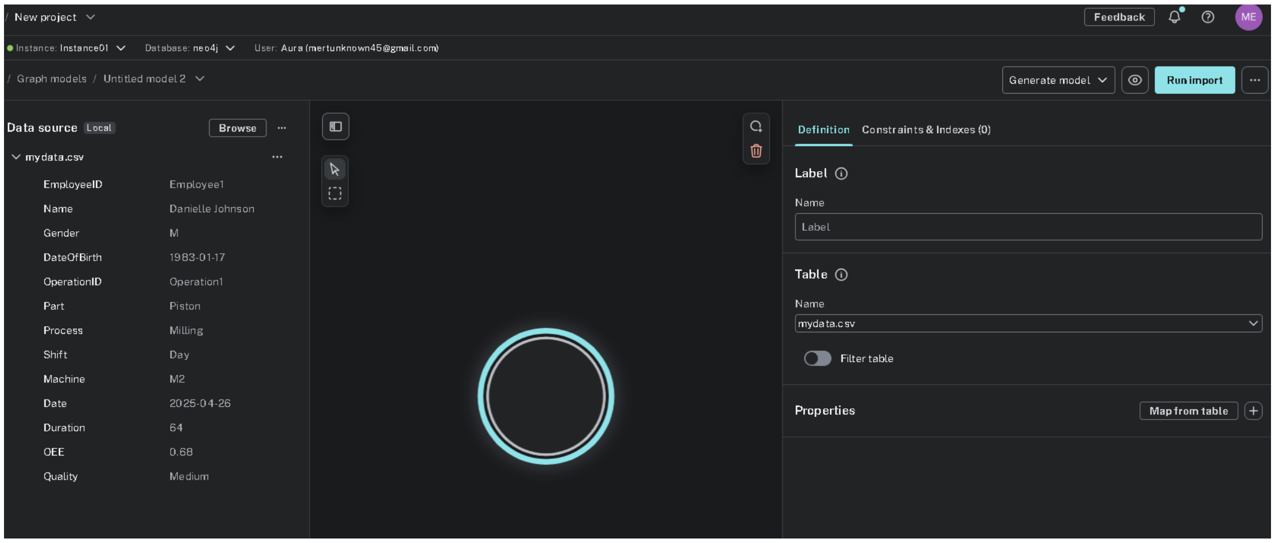
Define manually

Start with an empty canvas and sketch your model and data mapping manually

Generate from schema

Define a graph model and data mapping automatically from tables and constraints

Step-5: A screen like below appears. A sample row is shown in the left-hand side. Label name is provided through the window positioned on the right. Because of having only 1 csv file, there is no need of changing the table name by drop-down menu.



Instance: Instance01 Database: neo4j User: Aura (mertunknown45@gmail.com)

/ Graph models / Untitled model 2

Data source Local Browse

mydata.csv

EmployeeID	Employee1
Name	Danielle Johnson
Gender	M
DateOfBirth	1983-01-17
OperationID	Operation1
Part	Piston
Process	Milling
Shift	Day
Machine	M2
Date	2025-04-26
Duration	64
OEE	0.68
Quality	Medium

Label

Name: Label

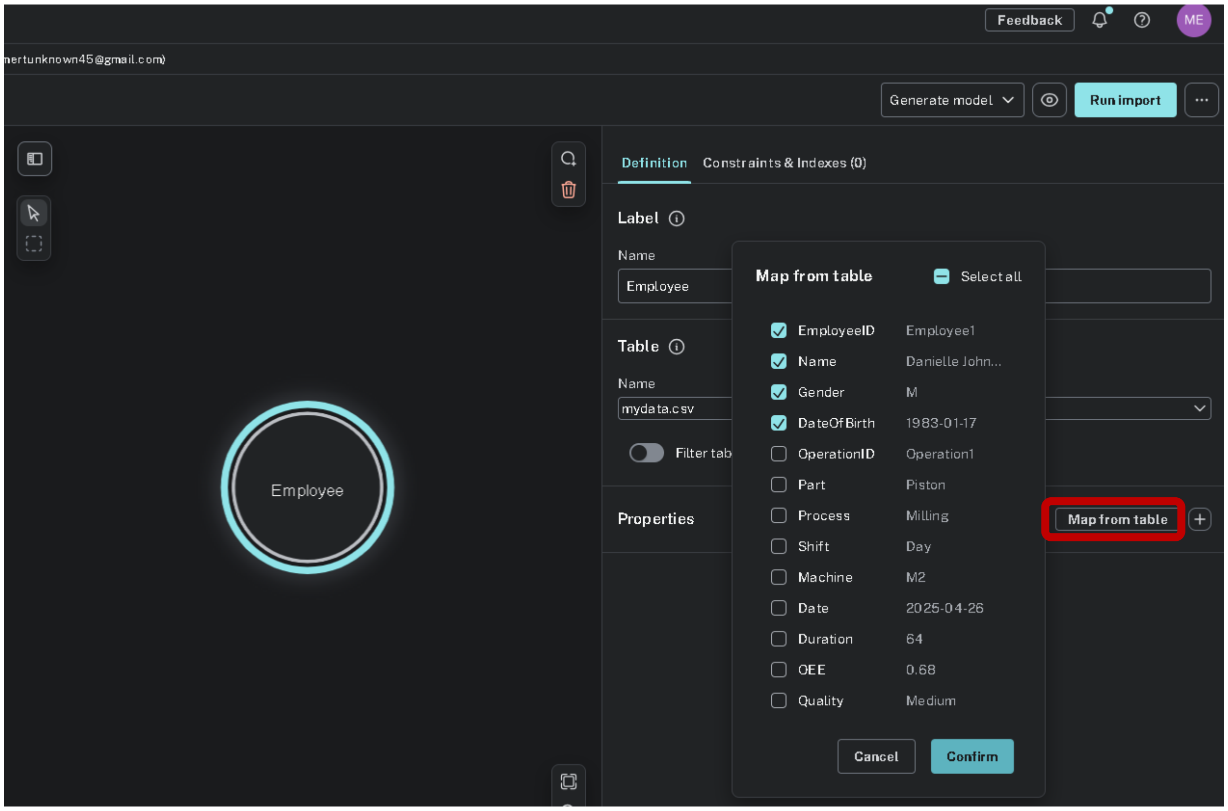
Table

Name: mydata.csv

Filter table

Properties Map from table

Step-6: When it comes to the properties of the node, they can be chosen from pop-up window, after clicking Map from table button. All of the column names related to Employee are checked.

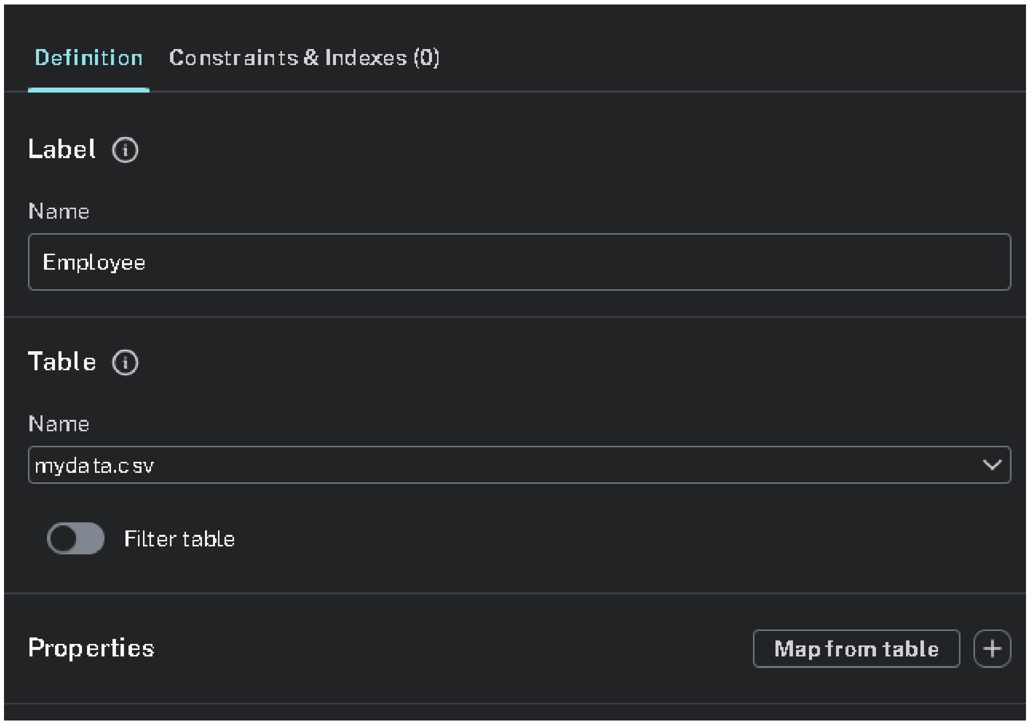


Map from table ☐ Select all

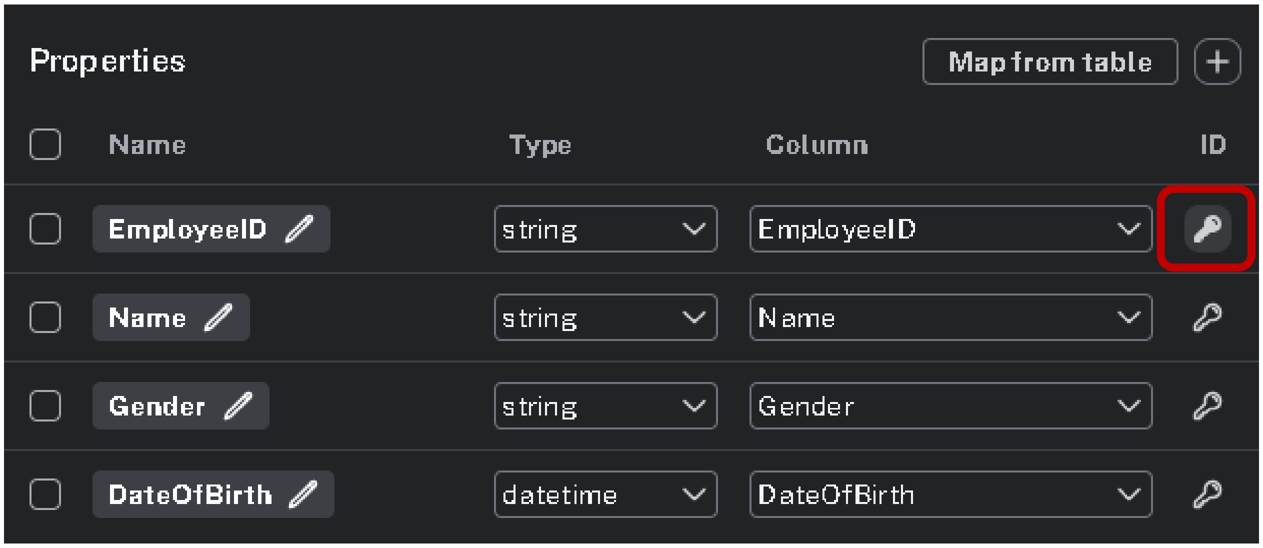
✓	EmployeeID	Employee1
✓	Name	Danielle John...
✓	Gender	M
✓	DateOfBirth	1983-01-17
☐	OperationID	Operation1
☐	Part	Piston
☐	Process	Milling
☐	Shift	Day
☐	Machine	M2
☐	Date	2025-04-26
☐	Duration	64
☐	OEE	0.68
☐	Quality	Medium

Cancel Confirm

Step-7: Label name is chosen as relevant keyword, in other words, Employee here.



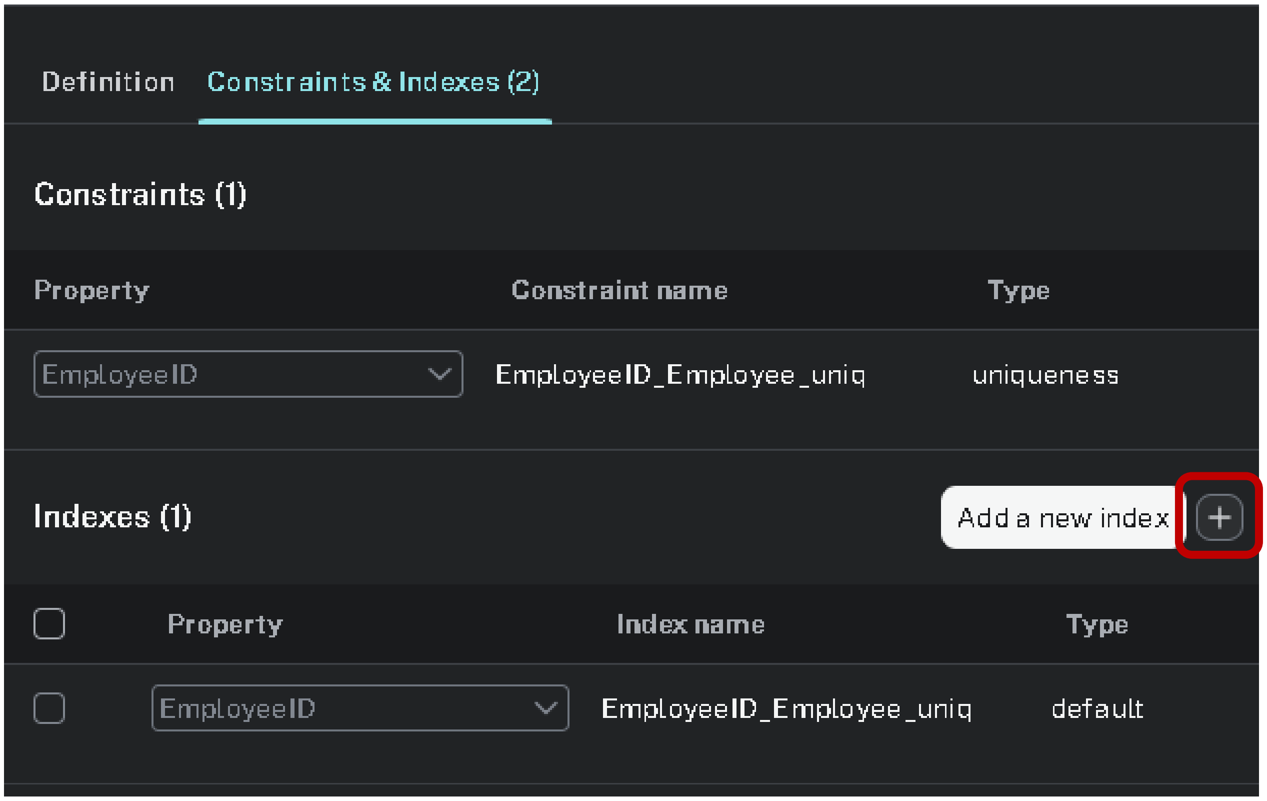
Step-8: After that, each property type is checked. If there is any wrong ones, it is corrected. Also, ID should be unique, so EmployeeID is selected by clicking the key button on the right.



Properties

Name	Type	Column	ID
EmployeeID	string	EmployeeID	🔑
Name	string	Name	
Gender	string	Gender	
DateOfBirth	datetime	DateOfBirth	

Step-9: In Constraints & Indexes tab, plus button (Add a new index) is clicked to add another index as name for employees.



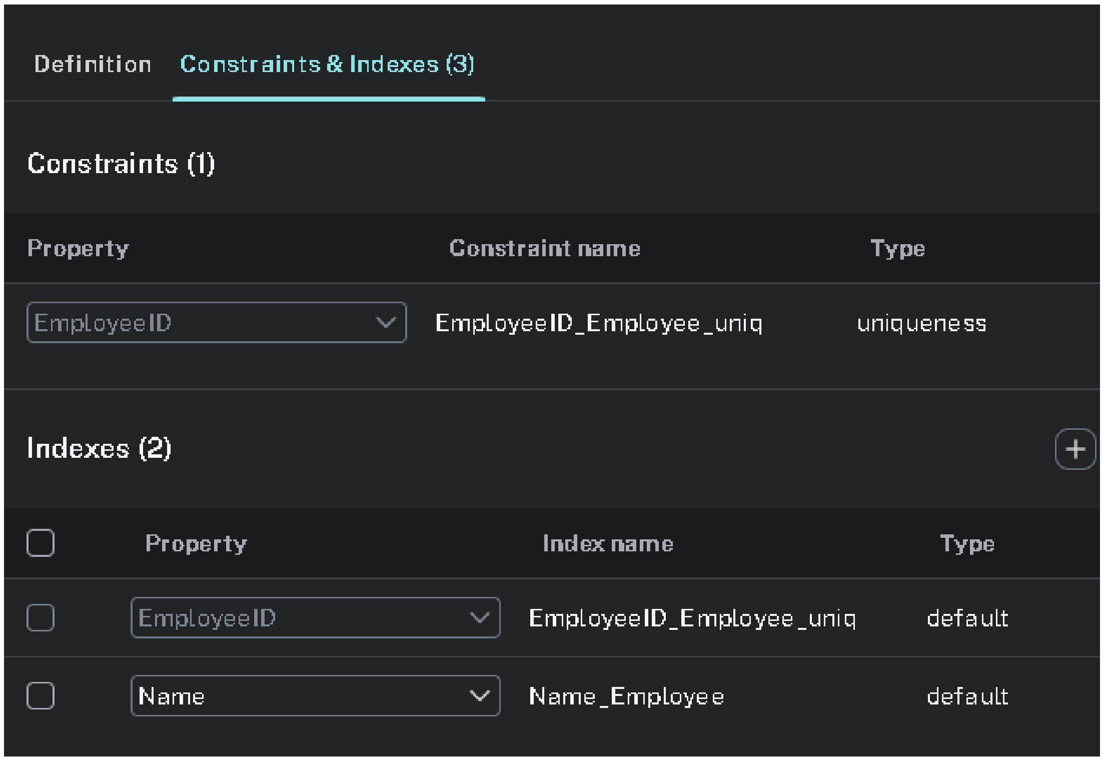
Constraints (1)

Property	Constraint name	Type
EmployeeID	EmployeeID_Employee_uniq	uniqueness

Indexes (1)

Property	Index name	Type
EmployeeID	EmployeeID_Employee_uniq	default

Step-10: Relevant column name is chosen, name here.



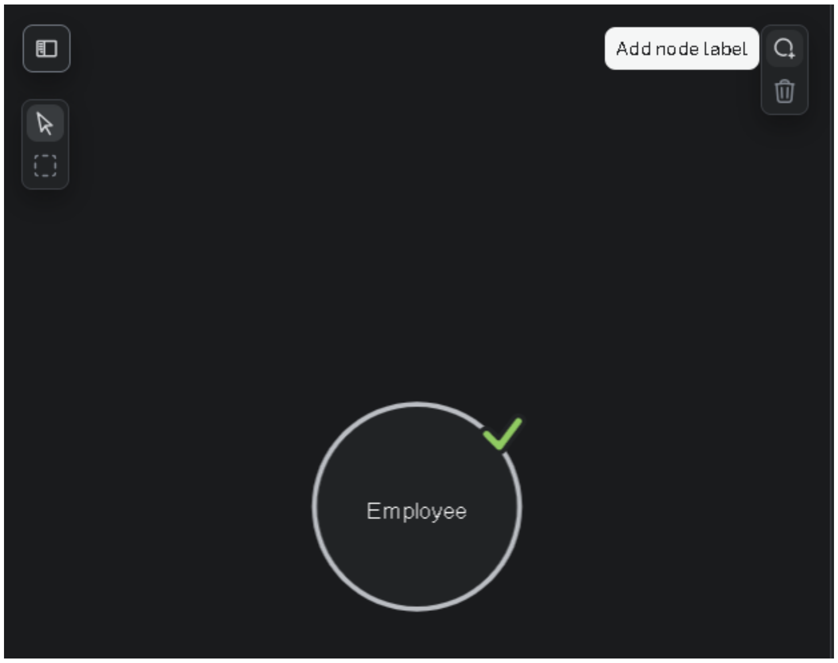
Constraints (1)

Property	Constraint name	Type
EmployeeID	EmployeeID_Employee_uniq	uniqueness

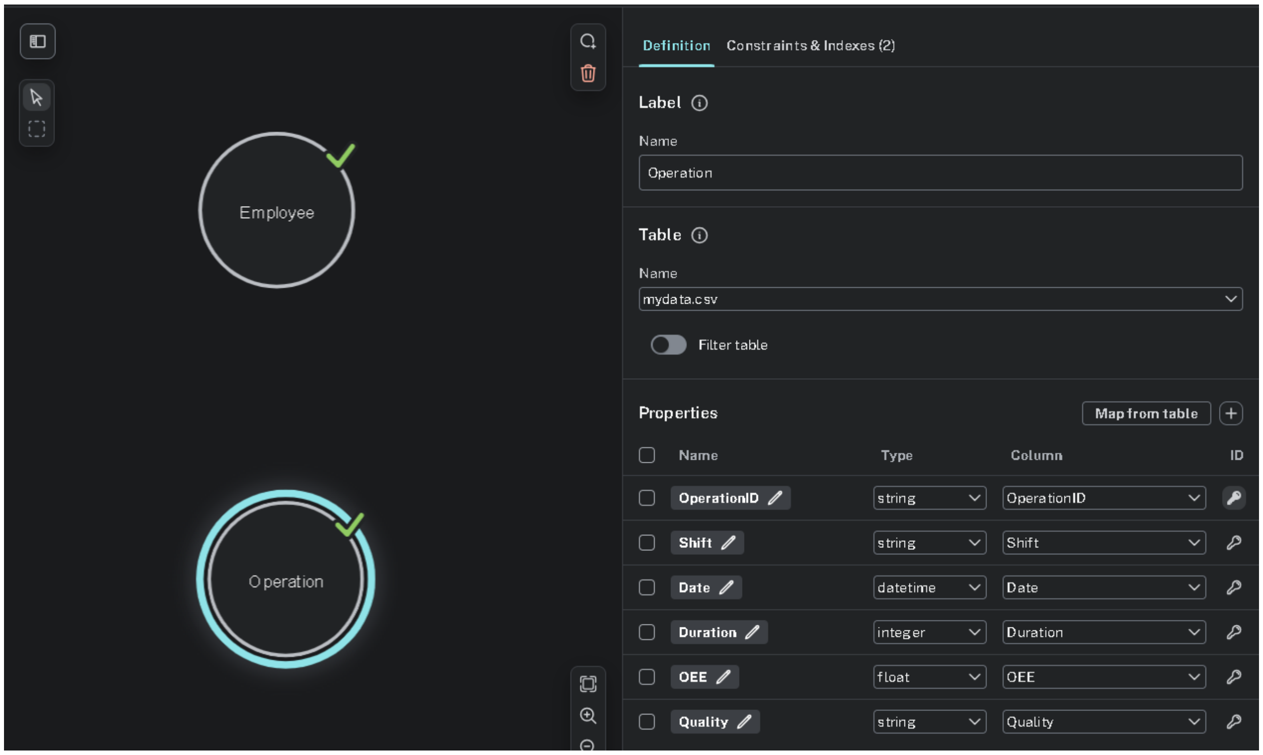
Indexes (2)

	Property	Index name	Type
☐	EmployeeID	EmployeeID_Employee_uniq	default
☐	Name	Name_Employee	default

Step-11: After all, Employee node is created.



Step-8: Similar steps are applied for Operation node.



Definition | Constraints & Indexes (2)

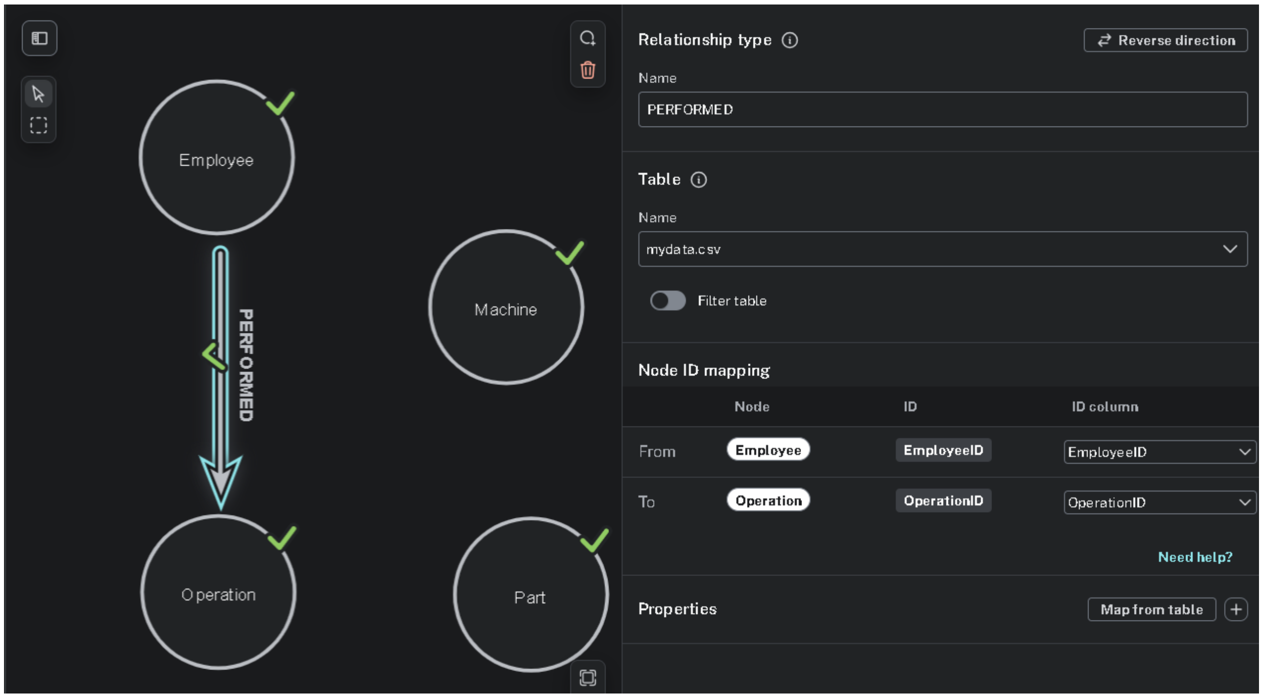
Label
Name: Operation

Table
Name: mydata.csv
Filter table

Properties — Map from table +

Name	Type	Column	ID
OperationID	string	OperationID	✓
Shift	string	Shift	
Date	datetime	Date	
Duration	integer	Duration	
OEE	float	OEE	
Quality	string	Quality	

Step-9: After creating all of the nodes, the relationships are created by clicking and dragging from one node edge to another. Relationship name is specified as name. If there is any property which should be attached to relationship, it can be added through the plus button near map from table. However, there is no property to add for this dataset.



Relationship type — Reverse direction
Name: PERFORMED

Table
Name: mydata.csv
Filter table

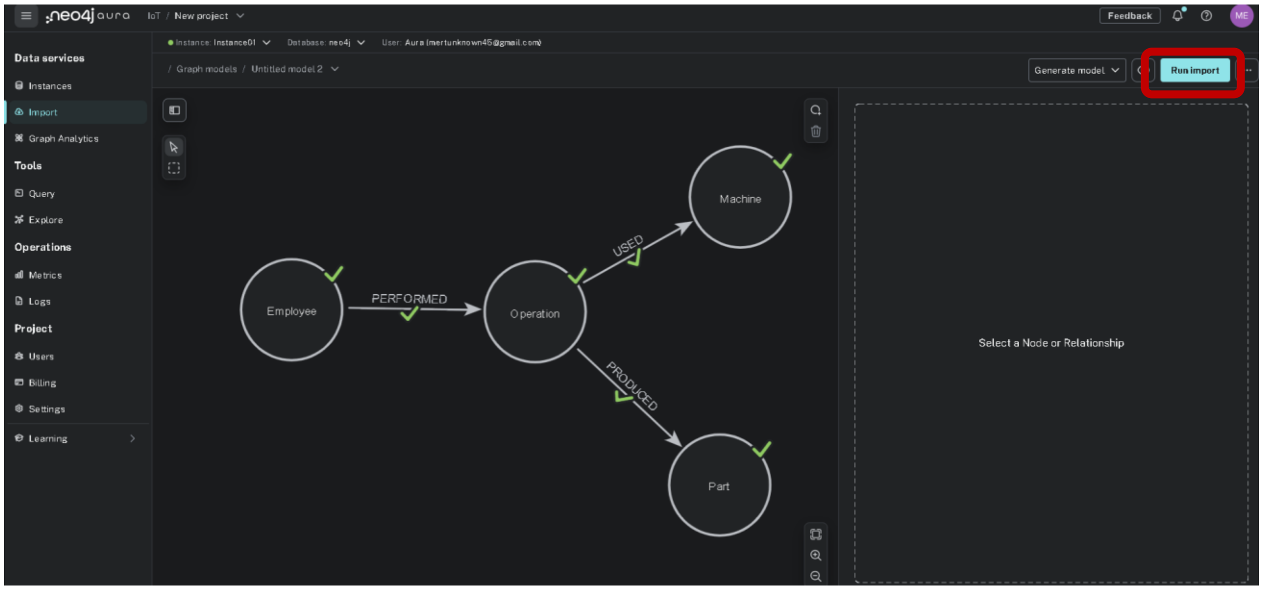
Node ID mapping

	Node	ID	ID column
From	Employee	EmployeeID	EmployeeID
To	Operation	OperationID	OperationID

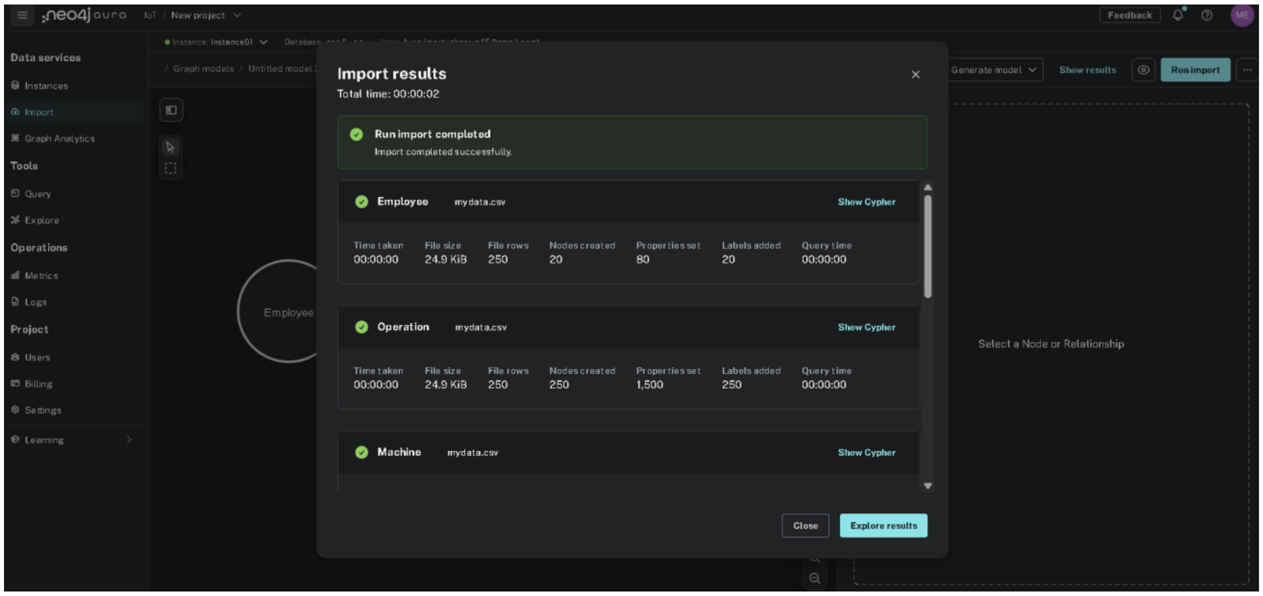
Need help?

Properties — Map from table +

Step-12: All of them are created as below. At the end, Run Import button is clicked to perform all of the settings.



Step-13: The result is seen in the windown below. Here, how many nodes, properties, and labels were created can be seen for each node type. Also, the total time is visible.



Import results

Total time: 00:00:02

Run import completed

Import completed successfully.

Employee mydata.csv Show Cypher

Time taken	File size	File rows	Nodes created	Properties set	Labels added	Query time
00:00:00	24.9 KiB	250	20	80	20	00:00:00

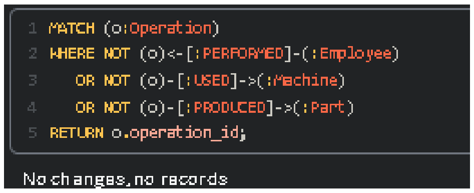
Operation mydata.csv Show Cypher

Time taken	File size	File rows	Nodes created	Properties set	Labels added	Query time
00:00:00	24.9 KiB	250	250	1,500	250	00:00:00

Machine mydata.csv Show Cypher

TEST

After loading a dataset, it should be checked if it has any errors. According to the query, there is no record found, which is related to not linked nodes.



```
MATCH (o:Operation)
WHERE NOT (o)<-[:PERFORMED]-(:Employee)
   OR NOT (o)-[:USED]->(:Machine)
   OR NOT (o)-[:PRODUCED]->(:Part)
RETURN o.operation_id;

No changes, no records
```

SUMMARY

In this task, a CSV dataset was imported into Neo4j Aura through both the Cypher LOAD CSV method and the manual interface provided in the platform. Throughout the process, data integrity was maintained by defining appropriate constraints, assigning correct data types, and establishing meaningful relationships between nodes.

The work demonstrated how structured operational data can be effectively modeled in a graph format to allow more flexible and insightful querying. Emphasis was placed on the importance of schema design, accurate property mapping, and validation during the import process.

As a result, a well-structured graph model was created, enabling rich analysis of operations, employee activities, machine utilization, and part production, which provides a solid foundation for further analytical tasks in future assignments.