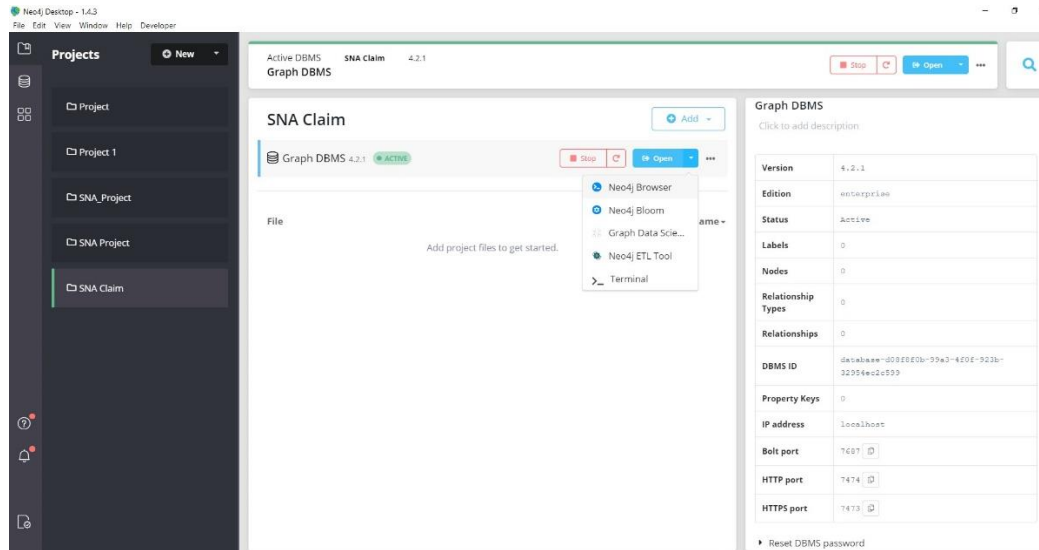
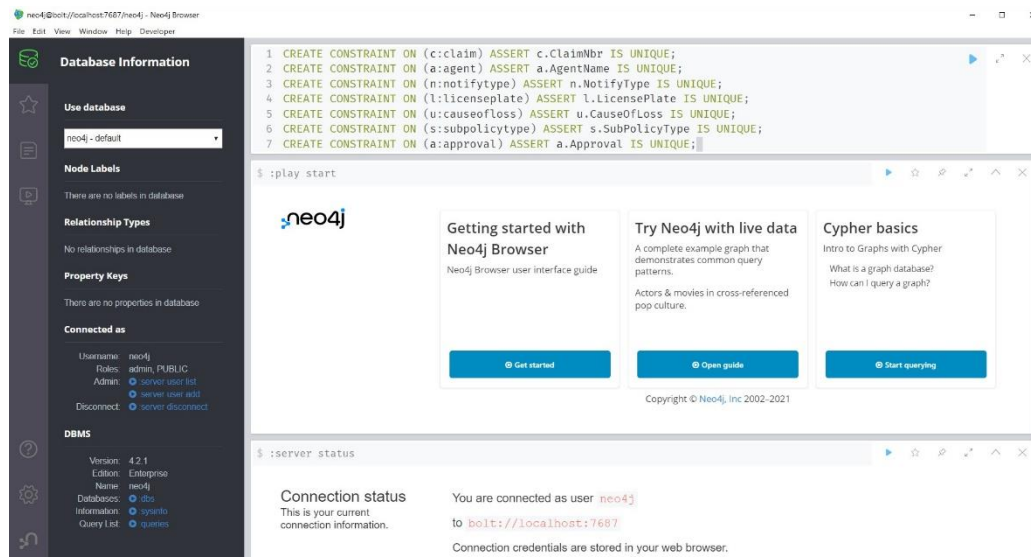


Load Dataset into local DBMS

- Use Cypher Query Language
- Use Neo4j Browser



- Load Data Claim to local DBMS
 - Create Constraint of claim data



The image shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the 'neo4j - default' database, node labels (agent, approval, causeofloss, claim, licenseplate, notifytype, subpolicytype), relationship types, property keys, and connection status. The main area displays a Cypher query to create constraints for uniqueness on various attributes. Below the query, a table shows the execution results for each constraint, all marked as successful. At the bottom, there are links for 'Getting started with Neo4j Browser', 'Try Neo4j with live data', and 'Cypher basics'.

```
neo4j$

1 CREATE CONSTRAINT ON (c:claim) ASSERT c.ClaimNbr IS UNIQUE;
2 CREATE CONSTRAINT ON (a:agent) ASSERT a.AgentName IS UNIQUE;
3 CREATE CONSTRAINT ON (n:notifytype) ASSERT n.NotifyType IS UNIQUE;
4 CREATE CONSTRAINT ON (l:licenseplate) ASSERT l.LicensePlate IS UNIQUE;
5 CREATE CONSTRAINT ON (u:causeofloss) ASSERT u.CauseOfLoss IS UNIQUE;
6 CREATE CONSTRAINT ON (s:subpolicytype) ASSERT s.SubPolicyType IS UNIQUE;
7 CREATE CONSTRAINT ON (a:approval) ASSERT a.Approval IS UNIQUE;
```

Constraint	Result
CREATE CONSTRAINT ON (c:claim) ASSERT c.ClaimNbr IS UNIQUE	✓
CREATE CONSTRAINT ON (a:agent) ASSERT a.AgentName IS UNIQUE	✓
CREATE CONSTRAINT ON (n:notifytype) ASSERT n.NotifyType IS UNIQUE	✓
CREATE CONSTRAINT ON (l:licenseplate) ASSERT l.LicensePlate IS UNIQUE	✓
CREATE CONSTRAINT ON (u:causeofloss) ASSERT u.CauseOfLoss IS UNIQUE	✓
CREATE CONSTRAINT ON (s:subpolicytype) ASSERT s.SubPolicyType IS UNIQUE	✓
CREATE CONSTRAINT ON (a:approval) ASSERT a.Approval IS UNIQUE	✓

neo4j

Getting started with Neo4j Browser
Neo4j Browser user interface guide

Try Neo4j with live data
A complete example graph that demonstrates common query patterns.
Actors & movies in cross-referenced

Cypher basics
Intro to Graphs with Cypher
What is a graph database?
How can I query a graph?

- Load Data and create property, label
- Create Relationship

The image shows the Neo4j Browser interface with a Cypher query to load data from a CSV file. The query uses the 'LOAD CSV' command to read data from 'file:///DATAMTCLAIMALL.csv' and creates nodes for 'agent', 'approval', 'causeofloss', 'licenseplate', 'notifytype', and 'subpolicytype'. It also creates relationships between these nodes based on the data in the CSV file.

```
neo4j$

1 LOAD CSV WITH HEADERS FROM
2 'file:///DATAMTCLAIMALL.csv' AS line
3 WITH line,SPLIT(line.AccidentDate,'-') as date
4
5 CREATE (claim:claim {ClaimNbr: line.ClaimNbr, NotifyType: line.NotifyType, CauseOfLoss: line.CauseOfLoss, Approval: line.Approval,
6 LicensePlate: line.LicensePlate, SubPolicyType: line.SubPolicyType, CarBrandCode: line.CarBrandCode, CarModel: line.CarModel,
7 AgentName: line.AgentName})
8
9 MERGE (agent:agent {AgentName: line.AgentName})
10 MERGE (notifytype:notifytype {NotifyType: line.NotifyType})
11 MERGE (licenseplate:licenseplate {LicensePlate: line.LicensePlate})
12 MERGE (causeofloss:causeofloss {CauseOfLoss: line.CauseOfLoss})
13 MERGE (subpolicytype:subpolicytype {SubPolicyType: line.SubPolicyType})
14 MERGE (approval:approval {Approval: line.Approval})
15
16 //Create Relationship
17 CREATE (claim)-[:agentowner]->(agent)
18 CREATE (claim)-[:vehicleclaim]->(licenseplate)
19 CREATE (claim)-[:approvedby]->(approval)
20 CREATE (claim)-[:type]->(notifytype)
21 CREATE (claim)-[:productgroup]->(subpolicytype)
22 CREATE (claim)-[:result]->(causeofloss)
23 ;
```

The image shows the Neo4j Browser interface with the same Cypher query as the previous image. The 'Database Information' sidebar is visible on the left. The main area displays the query, and the 'server status' section shows the connection status, indicating that the user is connected as 'neo4j' to 'bolt://localhost:7687'.

```
neo4j$

1 LOAD CSV WITH HEADERS FROM
2 'file:///DATAMTCLAIMALL.csv' AS line
3 WITH line,SPLIT(line.AccidentDate,'-') as date
4
5 CREATE (claim:claim {ClaimNbr: line.ClaimNbr, NotifyType: line.NotifyType, CauseOfLoss:
6 line.CauseOfLoss, Approval: line.Approval, LicensePlate: line.LicensePlate, SubPolicyType:
7 line.SubPolicyType, CarBrandCode: line.CarBrandCode, CarModel: line.CarModel, AgentName:
8 line.AgentName})
9 MERGE (agent:agent {AgentName: line.AgentName})
10 MERGE (notifytype:notifytype {NotifyType: line.NotifyType})
11 MERGE (licenseplate:licenseplate {LicensePlate: line.LicensePlate})
```

server status

Connection status
This is your current connection information.

You are connected as user **neo4j**
to **bolt://localhost:7687**
Connection credentials are stored in your web browser.

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar displays the current database as 'neo4j - default'. Under 'Node Labels', it lists 'agent', 'approval', 'causeofloss', 'claim', 'licenseplate', 'notifytype', and 'subpolicytype'. Under 'Relationship Types', it lists 'No relationships in database'. Under 'Property Keys', it lists 'AgentName', 'Approval', 'CauseOfLoss', 'ClaimNr', 'LicensePlate', 'NotifyType', and 'SubPolicyType'. Under 'Connected as', it shows 'Username: neo4j', 'Roles: admin, PUBLIC', and 'Admin: server user list, server user add, server disconnect'. The main panel shows a Cypher query: `neo4j$ LOAD CSV WITH HEADERS FROM 'file:///DATAM/CLAIMALL.csv' AS line WITH line,SPLIT(line,Acc...`. Below the query, the 'server status' section shows the connection status: 'You are connected as user neo4j to bolt://localhost:7687. Connection credentials are stored in your web browser.'

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar displays the current database as 'neo4j - default'. Under 'Node Labels', it lists '192,000 agent', 'approval', 'causeofloss', 'claim', 'licenseplate', 'notifytype', and 'subpolicytype'. Under 'Relationship Types', it lists '100,000 agentowner', 'approval', 'producttype', 'result', 'type', and 'vehicleclaim'. Under 'Property Keys', it lists 'AgentName', 'Approval', 'CarBrandCode', 'CarModel', 'CauseOfLoss', 'ClaimNr', 'LicensePlate', 'NotifyType', and 'SubPolicyType'. Under 'Connected as', it shows 'Username: neo4j', 'Roles: admin, PUBLIC', and 'Admin: server user list, server user add, server disconnect'. The main panel shows a Cypher query: `neo4j$ LOAD CSV WITH HEADERS FROM 'file:///DATAM/CLAIMALL.csv' AS line WITH line,SPLIT(line,Acc...`. Below the query, the 'server status' section shows the connection status: 'You are connected as user neo4j to bolt://localhost:7687. Connection credentials are stored in your web browser.'

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar displays the current database as 'neo4j - default'. Under 'Node Labels', it lists '192,000 agent', 'approval', 'causeofloss', 'claim', 'licenseplate', 'notifytype', and 'subpolicytype'. Under 'Relationship Types', it lists '100,000 agentowner', 'approval', 'producttype', 'result', 'type', and 'vehicleclaim'. Under 'Property Keys', it lists 'AgentName', 'Approval', 'CarBrandCode', 'CarModel', 'CauseOfLoss', 'ClaimNr', 'LicensePlate', 'NotifyType', and 'SubPolicyType'. Under 'Connected as', it shows 'Username: neo4j', 'Roles: admin, PUBLIC', and 'Admin: server user list, server user add, server disconnect'. The main panel shows a Cypher query: `1 //Car Brand
2 CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;`. Below the query, the 'server status' section shows the connection status: 'You are connected as user neo4j to bolt://localhost:7687. Connection credentials are stored in your web browser.'

- Load Data Car Brand to local DBMS
 - Create Constraint of car brand

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the 'Use database' dropdown set to 'neo4j - default'. Below this, 'Node Labels' are listed: '92,000 agent', 'approval', 'causeOfLoss', 'claim', 'licenseplate', 'notifytype', and 'subpolicytype'. 'Relationship Types' include 'type', 'agentname', 'approval', 'productgroup', 'result', 'type', and 'vehicleclass'. 'Property Keys' include 'AgentName', 'Approval', 'CarBrandCode', 'CarModel', 'CauseOfLoss', 'ClaimNr', 'LicensePlate', 'Notify Type', and 'SubPolicyType'. The 'Connected as' section shows 'Username: neo4j' and 'Role: admin, full ctrl'.

The main panel displays a Cypher query editor with the following code:

```
1 //Car Brand
2 CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;
```

Below the query editor, the execution results are shown:

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///DATAMTCLAMALL.csv' AS line WITH line,SPLIT(line.Acc...
Added 92000 labels, created 92000 nodes, set 500528 properties, created 306600 relationships, completed after 3578 ms.
```

Below the results, the 'server status' section is visible, showing the connection status:

Connection status
This is your current connection information.

You are connected as user **neo4j**
to **bolt://localhost:7687**
Connection credentials are stored in your web browser.

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the 'Use database' dropdown set to 'neo4j - default'. Below this, 'Node Labels' are listed: '92,000 agent', 'approval', 'carbrand', 'causeOfLoss', 'claim', 'licenseplate', 'notifytype', and 'subpolicytype'. 'Relationship Types' include 'type', 'agentname', 'approval', 'productgroup', 'result', 'type', and 'vehicleclass'. 'Property Keys' include 'AgentName', 'Approval', 'CarBrand', 'CarBrandCode', 'CarModel', 'CauseOfLoss', 'ClaimNr', 'LicensePlate', 'Notify Type', and 'SubPolicyType'. The 'Connected as' section shows 'Username: neo4j' and 'Role: admin, full ctrl'.

The main panel displays a Cypher query editor with the following code:

```
neo4j$ //Car Brand CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;
```

Below the query editor, the execution results are shown:

```
neo4j$ //Car Brand CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;
Added 1 constraint, completed after 26 ms.
```

Below the results, the 'server status' section is visible, showing the connection status:

Connection status
This is your current connection information.

You are connected as user **neo4j**
to **bolt://localhost:7687**
Connection credentials are stored in your web browser.

- Load Data and create property, label
- Create Relationship

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j - default

Node Labels

(92,800) agent approval
carbrand causeofloss claim
licenseplate notifytype
subpolicytype

Relationship Types

(106,800) agentowner
approvestory productgroup
result type vehicleclaim

Property Keys

AgentName Approval
CarBrand CarBrandCode
CarModel CauseOfLoss
ClaimNbr LicensePlate
NotifyType SubPolicyType

Connected as

```

1 LOAD CSV WITH HEADERS FROM
2 'file:///CARBRAND.csv' AS line
3
4 MATCH(claim:claim {ClaimNbr: line.ClaimNbr})
5 MERGE(carbrand:carbrand {CarBrandCode: line.CarBrandCode})
6
7 CREATE (claim)-[:claimbrand]-(carbrand)
8
9 ;

```

neo4j\$ //Car Brand CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;

Added 1 constraint, completed after 26 ms.

neo4j\$ LOAD CSV WITH HEADERS FROM 'file:///DATAMTCLAINALL.csv' AS line WITH line,SPLIT(line.Acc...

Added 92000 labels, created 92000 nodes, set 500526 properties, created 306600 relationships, completed after 3578 ms.

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j - default

Node Labels

(92,115) agent approval
carbrand causeofloss claim
licenseplate notifytype
subpolicytype

Relationship Types

(107,700) agentowner
approvestory claimbrand
productgroup result type
vehicleclaim

Property Keys

AgentName Approval
CarBrand CarBrandCode
CarModel CauseOfLoss
ClaimNbr LicensePlate
NotifyType SubPolicyType

neo4j\$

neo4j\$ LOAD CSV WITH HEADERS FROM 'file:///CARBRAND.csv' AS line MATCH(claim:claim {ClaimNbr: L...

Added 115 labels, created 115 nodes, set 115 properties, created 51100 relationships, completed after 2221 ms.

neo4j\$ //Car Brand CREATE CONSTRAINT ON (b:carbrand) ASSERT b.CarBrand IS UNIQUE;

Added 1 constraint, completed after 26 ms.

- Load Data Car Model to local DBMS
 - Create Constraint of car model

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database name 'neo4j - default', node labels, relationship types, and property keys. The main area displays a Cypher query editor with the following queries:

```
1 //Car Model
2 CREATE CONSTRAINT ON (m:carmodel) ASSERT m.CarModel IS UNIQUE;
```

The execution results show:

- Added 115 labels, created 115 nodes, set 115 properties, created 51100 relationships, completed after 2221 ms.
- Added 115 labels, created 115 nodes, set 115 properties, created 51100 relationships, completed after 2221 ms.
- Added 1 constraint, completed after 26 ms.
- Added 1 constraint, completed after 26 ms.

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database name 'neo4j - default', node labels, relationship types, and property keys. The main area displays a Cypher query editor with the following queries:

```
neo4j$ //Car Model CREATE CONSTRAINT ON (m:carmodel) ASSERT m.CarModel IS UNIQUE;
```

The execution results show:

- Added 1 constraint, completed after 23 ms.
- Added 1 constraint, completed after 23 ms.
- Added 115 labels, created 115 nodes, set 115 properties, created 51100 relationships, completed after 2221 ms.

- Load Data and create property, label
- Create Relationship

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database

neo4j - default

Node Labels

{82,115}

agent

approval

carbrand

carmodel

causeofloss

claim

licenseplate

notifytype

subpolicytype

Relationship Types

{137,700}

agentowner

approves

claimbrand

productgroup

resale

type

vehicleclaim

Property Keys

AgentName

Approval

CarBrand

CarBrandCode

CarModel

CauseOfLoss

ClaimNbr

LicensePlate

NotifyType

SubPolicyType

```
1 LOAD CSV WITH HEADERS FROM
2 'file:///CARMODEL.csv' AS line
3
4 MATCH (claim:claim {ClaimNbr: line.ClaimNbr})
5 MATCH (claim)-[:claimbrand]->(carbrand)
6
7 CREATE (carmodel:carmodel {CarModelCode: line.CarModel ,CarBrandCode: line.CarBrandCode })
8
9 CREATE (claim)-[:claimmodel]->(carmodel)
10 MERGE (carmodel)-[:brand]->(carbrand)
11
12 ;
```

neo4j\$ //Car Model CREATE CONSTRAINT ON (m:carmodel) ASSERT m.CarModel IS UNIQUE;

Added 1 constraint, completed after 23 ms.

neo4j\$ LOAD CSV WITH HEADERS FROM 'file:///CARBRAND.csv' AS line MATCH (claim:claim {ClaimNbr: 1...

Added 115 labels, created 115 nodes, set 115 properties, created 51100 relationships, completed after 2221 ms.