

001-create-database

neo4j\$ **CREATE DATABASE testload WAIT;**

The screenshot shows the Neo4j browser interface. On the left, the sidebar displays 'Database information' with nodes (0), relationships (0), and property keys (id, name, position, uri). In the main panel, the command 'neo4j\$ CREATE DATABASE testload WAIT;' is entered in the query input field. Below it, a table shows the result of the command:

address	state	message	success
localhost:7687	"CaughtUp"	"caught up"	true

At the bottom right of the main panel, a message states: "Started streaming 1 record after 9,486 ms and completed after 9,487 ms."

The screenshot shows the Neo4j browser interface. The sidebar remains the same. In the main panel, the command 'neo4j\$ SHOW DATABASES;' is entered. Below it, a table lists all databases:

name	type	aliases	access	address	role	writer	requestedStatus
1 "chapter01"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
2 "neo4j"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
3 "system"	"system"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
4 "testload"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"

002-create-constraint

testload\$

CREATE CONSTRAINT FOR (n:Track) REQUIRE n.id IS NODE KEY;

The screenshot shows the Neo4j browser interface. The sidebar shows nodes (0) and relationships (0). In the main panel, the command 'testload\$ CREATE CONSTRAINT FOR (n:Track) REQUIRE n.id IS NODE KEY;' is entered. A message 'Added 1 constraint' appears below the command. Below it, a table shows the databases again:

name	type	aliases	access	address	role	writer	requestedStatus
1 "chapter01"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
2 "neo4j"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
3 "system"	"system"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"
4 "testload"	"standard"	[]	"read-write"	"localhost:7687"	"primary"	true	"online"

A message at the bottom right indicates: "Completed after 118 ms".

003-load-tracks

testload\$

```
LOAD CSV WITH HEADERS FROM "file:///medium/sample_tracks_medium.csv" AS row
MERGE (track:Track {id: row.track_id})
SET track.uri = row.track_uri,
track.name = row.track_name;
```

The screenshot shows the Neo4j Browser interface with the following details:

- Database Information:** Nodes (69,878), Relationships (0), Property keys (id, name, uri). The "Track" node is selected.
- Logs:**
 - testload\$ LOAD CSV WITH HEADERS FROM "file:///medium/sample_tracks_medium.csv" AS row
 - testload\$ CALL(row){
 - testload\$ MERGE (track:Track {id: row.track_id})
 - testload\$ SET track.uri = row.track_uri,
 - testload\$ track.name = row.track_name
 - testload\$ } IN TRANSACTIONS OF 10_000 ROWS;
 - Created 69,878 nodes, set 257,724 properties, added 69,878 labels (Completed after 4,732 ms)
 - testload\$ CREATE CONSTRAINT FOR (n:Track) REQUIRE n.id IS NODE KEY; (Added 1 constraint) (Completed after 118 ms)
 - neo4j\$ SHOW DATABASES;
- Graph View:** A large circular cluster of nodes representing the imported tracks.
- Results Overview:** Nodes (1,000), Track (1000).
- Notes:** Initial display limit hit at 1,000 nodes. Edit settings. Fetch limit hit at 5,000 records. Started streaming after 25 ms and completed after 105 ms.

004-call-in-transactions

```
testload$ LOAD CSV WITH HEADERS FROM "file:///medium/sample_tracks_medium.csv" AS row
CALL(row){
  MERGE (track:Track {id: row.track_id})
  SET track.uri = row.track_uri,
  track.name = row.track_name
} IN TRANSACTIONS OF 10_000 ROWS;

// CALL { ... }: Executes a subquery for each row
// IN TRANSACTIONS OF 10_000 ROWS: Processes in batches of 10,000 rows
```

Instance: neo4j://localhost:7687 Database: testload User: neo4j

Database information

- Nodes (69,878)
 - * Track
- Relationships (0)
- Property keys
- `id` `name` `uri`

```
testload$
```

```

1 LOAD CSV WITH HEADERS FROM "file:///medium/sample_tracks_medium.csv" AS row
2 CALL(row) {
3   MERGE (track:Track {id: row.track_id})
4   SET track.uri = row.track_uri,
5   track.name = row.track_name
6 } IN TRANSACTIONS OF 10_000 ROWS;

```

Set 187,846 properties

```
testload$ MATCH (n) RETURN n ;
```

005-create-database-locking

neo4j\$ CREATE DATABASE locking WAIT;

Instance: neo4j://localhost:7687 Database: neo4j User: neo4j

Database information

- Nodes (0)
- * Playlist Track User
- Relationships (0)
- * HAS_TRACK
- Property keys
- `id` `name` `position` `uri`

```
neo4j$
```

```
neo4j$ CREATE DATABASE locking WAIT;
```

Table	RAW		
address	state	message	success
"localhost:768	"CaughtUp"	"caught up"	true
7"			

006-merge-wait-spinning

locking\$

```

MERGE (n:Track {id: 1})
WITH n
// the transaction will be paused for 60 seconds
CALL apoc.util.sleep(60000)
// the transaction now continues
SET n.name = 'Creep'
RETURN n

```

Instance: neo4j://localhost:7687 Database: locking User: neo4j

Database information

- Nodes (1)
- * Track
- Relationships (0)
- *
- Property keys
- `id` `name`

```
locking$
```

```

1 MERGE (n:Track {id: 1})
2 WITH n
3 // the transaction will be paused for 60 seconds
4 CALL apoc.util.sleep(60000)
5 // the transaction now continues
6 SET n.name = 'Creep'
7 RETURN n

```

Results overview

Nodes (1)

* (1) Track (1)



007-create-node-1

locking\$ **CREATE (t:Track {id: 1})**

The screenshot shows the Neo4j browser interface. On the left, the 'Database information' sidebar indicates there are 2 nodes and 0 relationships. It also lists property keys 'id' and 'name'. The main workspace displays a single green circular node labeled '1'. Below it, another green circular node labeled 'Creep' is shown. To the right, the 'Results overview' panel shows 'Nodes (2)' and 'Track (2)'. At the bottom, the command history shows 'locking\$ CREATE (t:Track {id: 1})' and a message indicating 'Created 1 node, set 1 property, added 1 label'. A status bar at the bottom right says 'Started streaming 2 records after 3 ms arc'.

008-tab-1-1

```
locking$  
    MATCH (t:Track {id: 1})  
    SET t.name = 'Creep'  
    WITH t  
    CALL apoc.util.sleep(60000)  
    RETURN t
```

The screenshot shows the Neo4j browser interface with a transaction tab open. The tab title is 'locking\$'. The main workspace contains the same two nodes as before. The command history in the bottom panel shows the entire query from the previous step. The status bar at the bottom right indicates 'Started streaming 2 records after 3 ms arc'.

009-tab-1-2

locking\$

```
MATCH (t:Track {id: 1})
SET t.name = 'Creep from transaction 2'
RETURN t
```

The screenshot shows the Neo4j Browser interface. On the left, the sidebar displays 'Database information' with 'Nodes (2)' and 'Relationships (0)'. Under 'Nodes (2)', there is one node labeled 'Track'. Below the sidebar, the main area has a title 'locking\$' and contains a code block with the Cypher query: 'MATCH (t:Track {id: 1}) SET t.name = 'Creep from transaction 2' RETURN t'. Below the code, there are three green circular nodes representing the results. The first node has the text 'Creep fr...'. The second node has the text 'om tran...'. The third node has the text 'om tran...'. At the bottom of the main area, it says 'Set 2 properties'. On the right side, there is a 'Results overview' section showing 'Nodes (2)' and 'Track (2)'. At the bottom right, it says 'Started streaming 2 records after 56 ms'.

010-transactions-info

locking\$

```
SHOW TRANSACTIONS YIELD *
RETURN transactionId, status, currentQueryId, currentQuery,
resourceInformation.lockMode AS lockMode,
resourceInformation.resourceType AS lockOnResource
```

The screenshot shows the Neo4j Browser interface. On the left, the sidebar displays 'Database information' with 'Nodes (2)' and 'Relationships (0)'. Under 'Nodes (2)', there is one node labeled 'Track'. Below the sidebar, the main area has a title 'locking\$' and contains a code block with the Cypher query: 'SHOW TRANSACTIONS YIELD * RETURN transactionId, status, currentQueryId, currentQuery, resourceInformation.lockMode AS lockMode, resourceInformation.resourceType AS lockOnResource'. Below the code, there is a table with the following data:

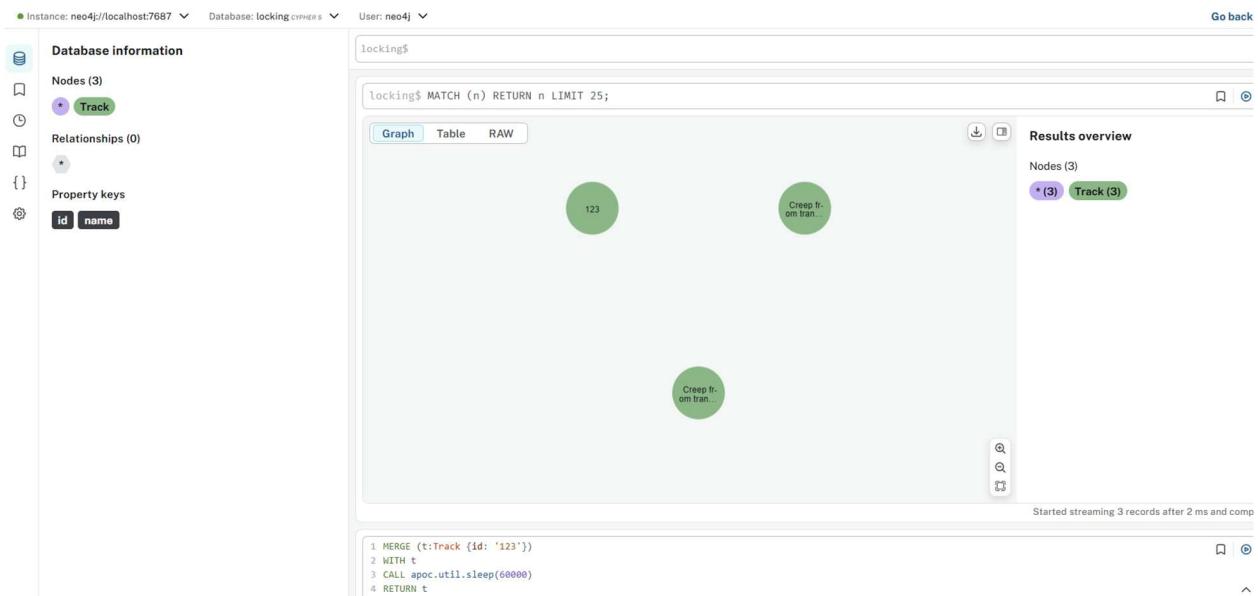
transactionId	status	currentQueryId	currentQuery	lockMode	lockOnResource
"locking-transaction-19"	"Running"	"query-82"	"SHOW TRANSACTIONS YIELD * RETURN transactionId, status, currentQueryId, currentQuery, resourceInformation.lockMode AS lockMode, resourceInformation.resourceType AS lockOnResource"	null	null

At the bottom right, it says 'Started streaming 1 record after 80 ms and completed after 102 ms.'

011-tab-2-1

locking\$

```
MERGE (t:Track {id: '123'})  
WITH t  
CALL apoc.util.sleep(60000)  
RETURN t
```



012-tab-2-2

locking\$

```
MERGE (t1:Track {id: '123'})  
MERGE (t2:Track {id: '234'})  
MERGE (t1)-[:SIMILAR_TO]->(t2)
```

Instance: neo4j://localhost:7687 Database: locking User: neo4j Go back

Database information

- Nodes (4)
- Relationships (1)
- Property keys
- id name

locking\$

```
locking$ MATCH (n) RETURN n LIMIT 25;
```

Graph Table RAW

Results overview

Nodes (4) * (4) Track (4)

Started streaming 4 records after 2 ms and c

```
1 MERGE (t1:Track {id: '123'})  
2 MERGE (t2:Track {id: '234'})  
3 MERGE (t1)-[:SIMILAR_TO]->(t2)
```

✓ Created 1 node, created 1 relationship, set 1 property, added 1 label

Instance: neo4j://localhost:7687 Database: locking User: neo4j Go back

Database information

- Nodes (4)
- Relationships (1)
- Property keys
- id name

locking\$

```
locking$ MATCH (n) RETURN n LIMIT 25;
```

Graph Table RAW

Results overview

Nodes (4) * (4) Track (4)

Relationships (1) * (1) SIMILAR_TO (1)

Started streaming 1 record after 47 ms and comp

```
1 MERGE (t1:Track {id: '123'})  
2 MERGE (t2:Track {id: '234'})  
3 MERGE (t1)-[:SIMILAR_TO]->(t2)
```

Started streaming 4 records after 2 ms and com

013-unique-constraint

locking\$ MATCH (n) DETACH DELETE n;

The screenshot shows the Neo4j Browser interface. On the left, there's a sidebar titled "Database information" with sections for Nodes (0), Relationships (0), and Property keys (id, name). The main area has a search bar with "locking\$". Below it, a log window shows three entries:

- locking\$ MATCH (n) RETURN n LIMIT 25; No changes, no records Completed a few seconds ago
- locking\$ MATCH (n) DETACH DELETE n; Deleted 4 nodes, deleted 1 relationship Completed after 1 ms
- locking\$ MATCH (n) RETURN n LIMIT 25; No changes, no records Completed after 1 ms

locking\$

CREATE CONSTRAINT track_uk FOR (t:Track) REQUIRE t.id IS UNIQUE;

The screenshot shows the Neo4j Browser interface. The sidebar shows "Nodes (0)" with a single node labeled "Track" highlighted in green. The main area has a search bar with "locking\$". Below it, a log window shows two entries:

- locking\$ show CONSTRAINTS Table RAW
- locking\$ CREATE CONSTRAINT track_uk FOR (t:Track) REQUIRE t.id IS UNIQUE; Added 1 constraint

A table view shows the constraint details:

id	name	type	entityType	labelsOrTypes	properties
1	"track_uk"	"UNIQUENESS"	"NODE"	["Track"]	[{"id"}]

Completed after 62 ms

014-tab-3-1

locking\$

```
MATCH (t1:Track {id: 1})
SET t1.popularity = 0.9
WITH t1
CALL apoc.util.sleep(60000)
```

The screenshot shows the Neo4j Browser interface. The sidebar shows "Nodes (0)" with a single node labeled "Track" highlighted in green. The main area has a search bar with "locking\$". Below it, a log window shows two entries:

- 1 MATCH (t1:Track {id: 1})
2 SET t1.popularity = 0.9
3 WITH t1
4 CALL apoc.util.sleep(60000)
No changes, no records Completed after 83 ms
- locking\$ MATCH (n) RETURN n LIMIT 25;
No changes, no records Completed after 20 ms

015-tab-3-2

locking\$

```
MATCH (t1:Track {id: 1})  
MATCH (t2:Track {id: 2})  
CREATE (t1)-[r:SIMILAR]->(t2)
```

Instance: neo4j://localhost:7687 Database: locking User: neo4j

Go back to old Browse

Database information

Nodes (0)	lockings	...
Track	1 MATCH (t1:Track {id: 1}) 2 MATCH (t2:Track {id: 2}) 3 CREATE (t1)-[r:SIMILAR]->(t2)	undo redo ^ ↵ ×
Relationships (0)	No changes, no records	Completed after 99 m: