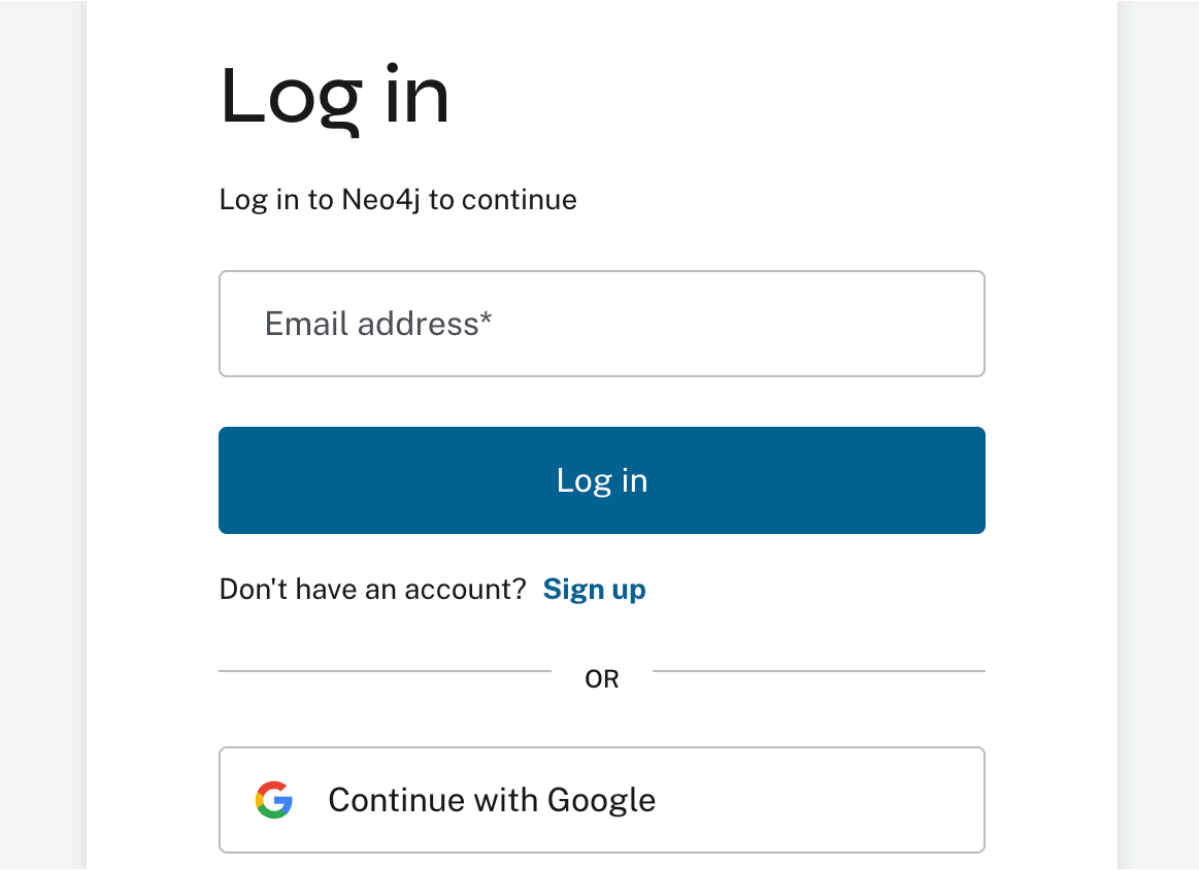
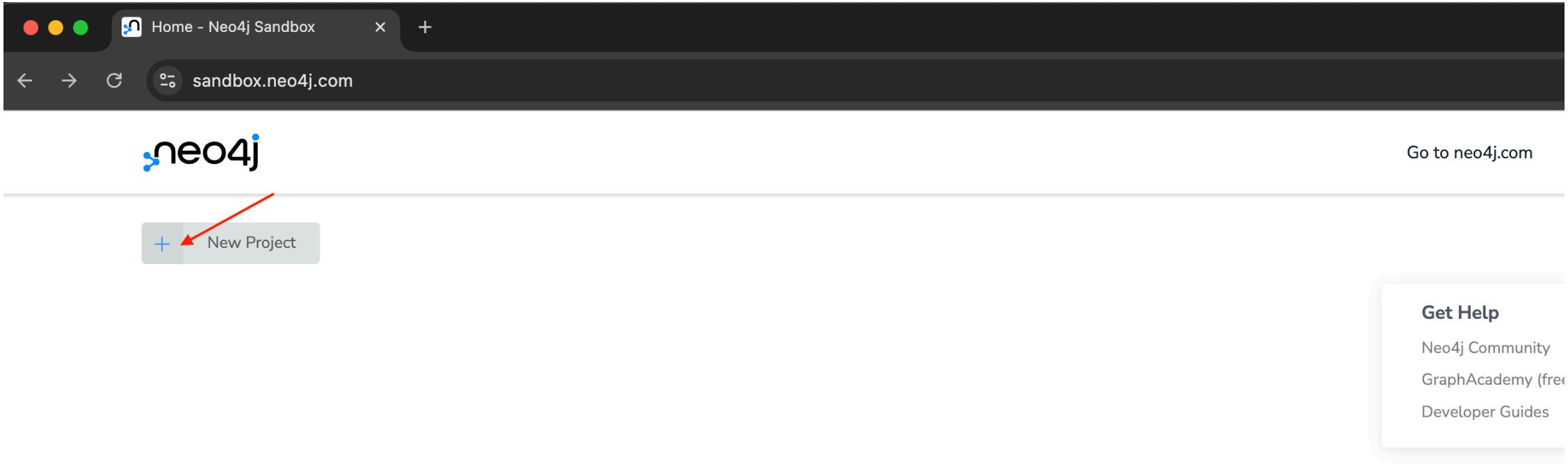
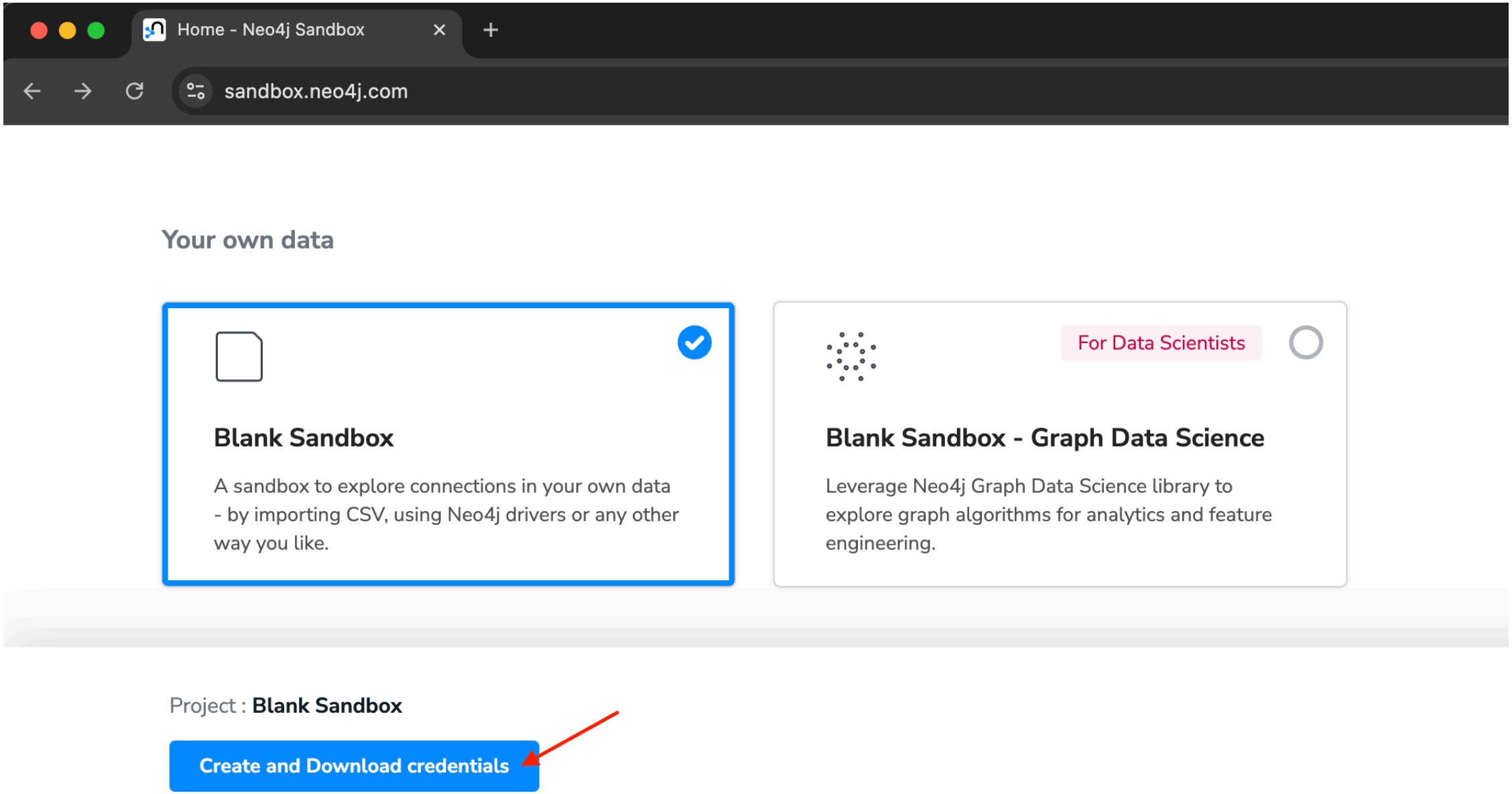
Neo4j Sandbox 구축

단계1: Neo4j Sandbox 접속

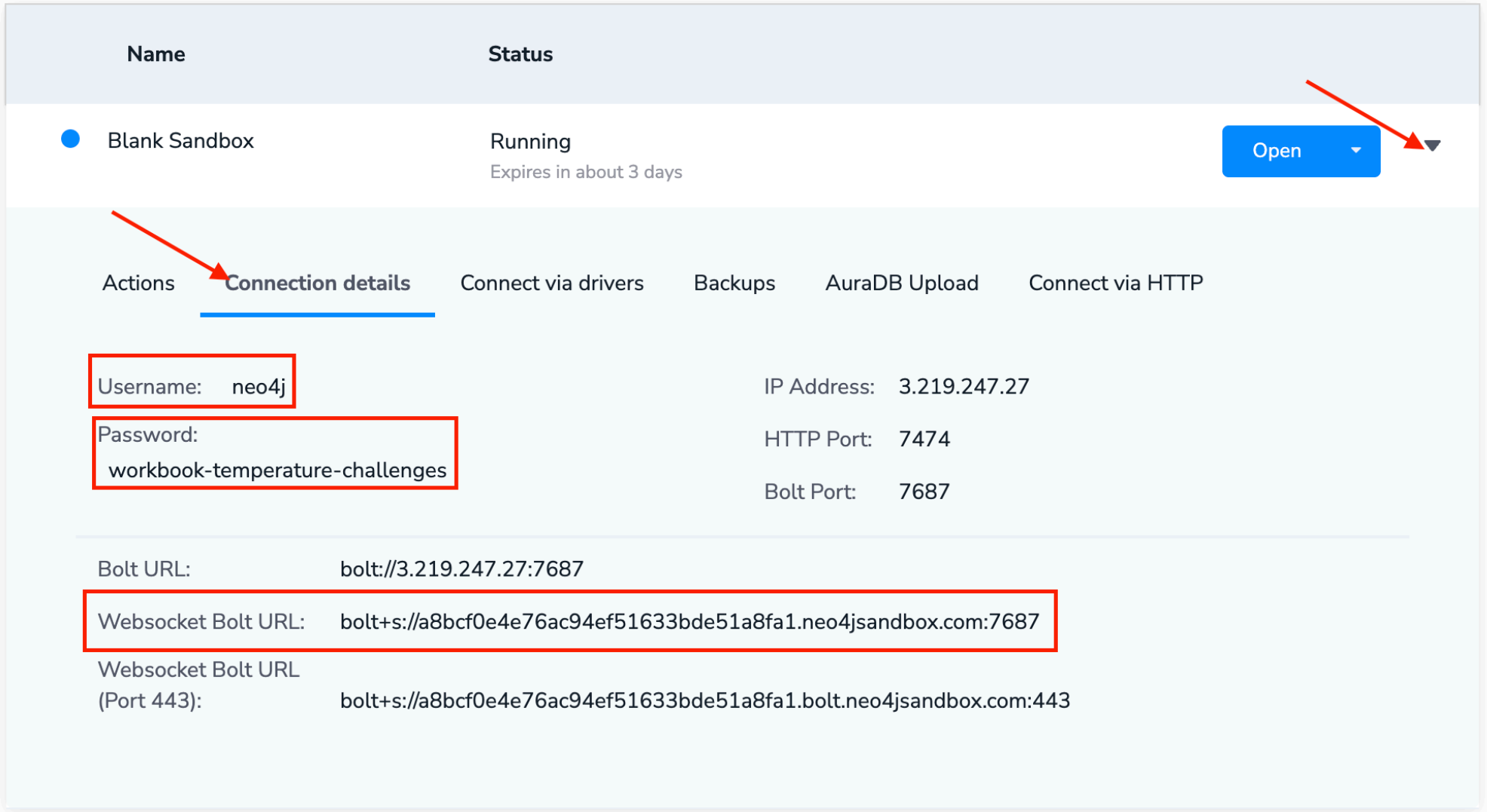


단계2: New Project




Home - Neo4j Sandbox
sandbox.neo4j.com
Your own data
Blank Sandbox
A sandbox to explore connections in your own data - by importing CSV, using Neo4j drivers or any other way you like.
For Data Scientists
Blank Sandbox - Graph Data Science
Leverage Neo4j Graph Data Science library to explore graph algorithms for analytics and feature engineering.
Project : Blank Sandbox
Create and Download credentials

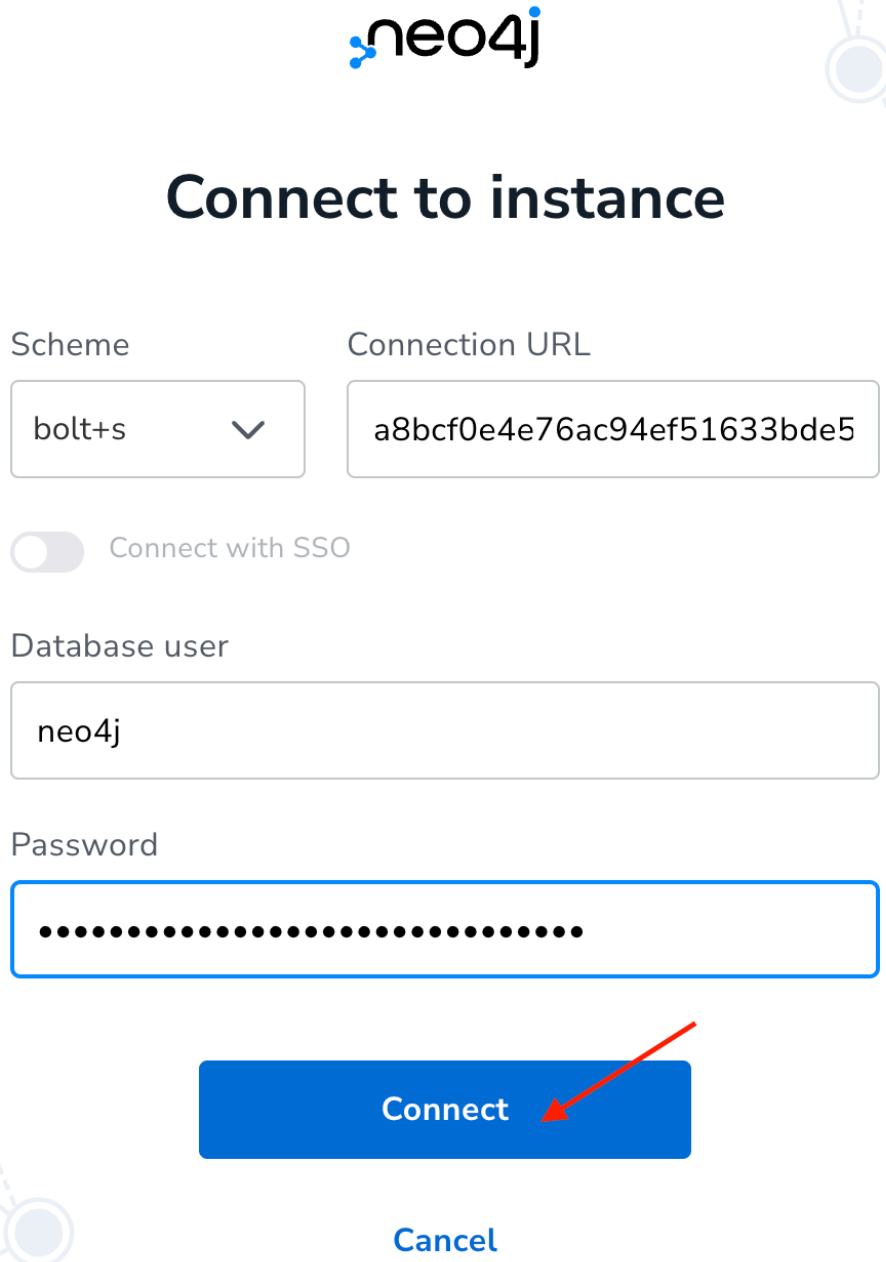
단계3: 생성된 DB의 접속 정보 확인



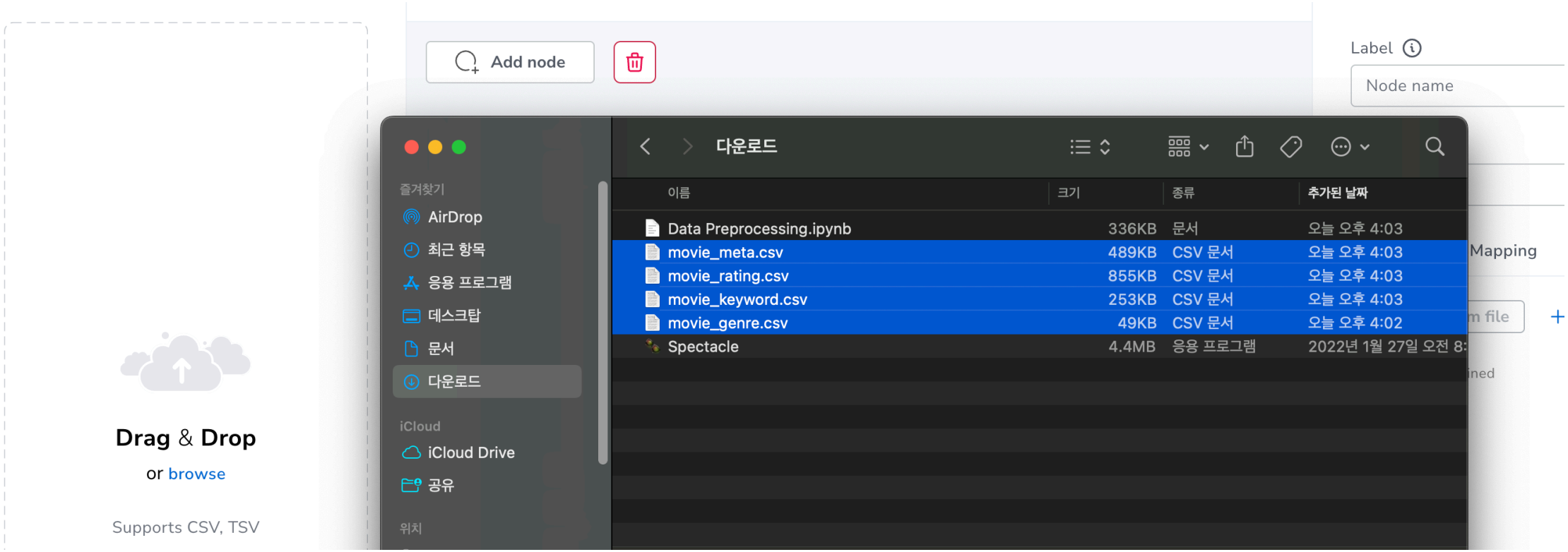
데이터셋 임포트

단계1: data-importer 접속

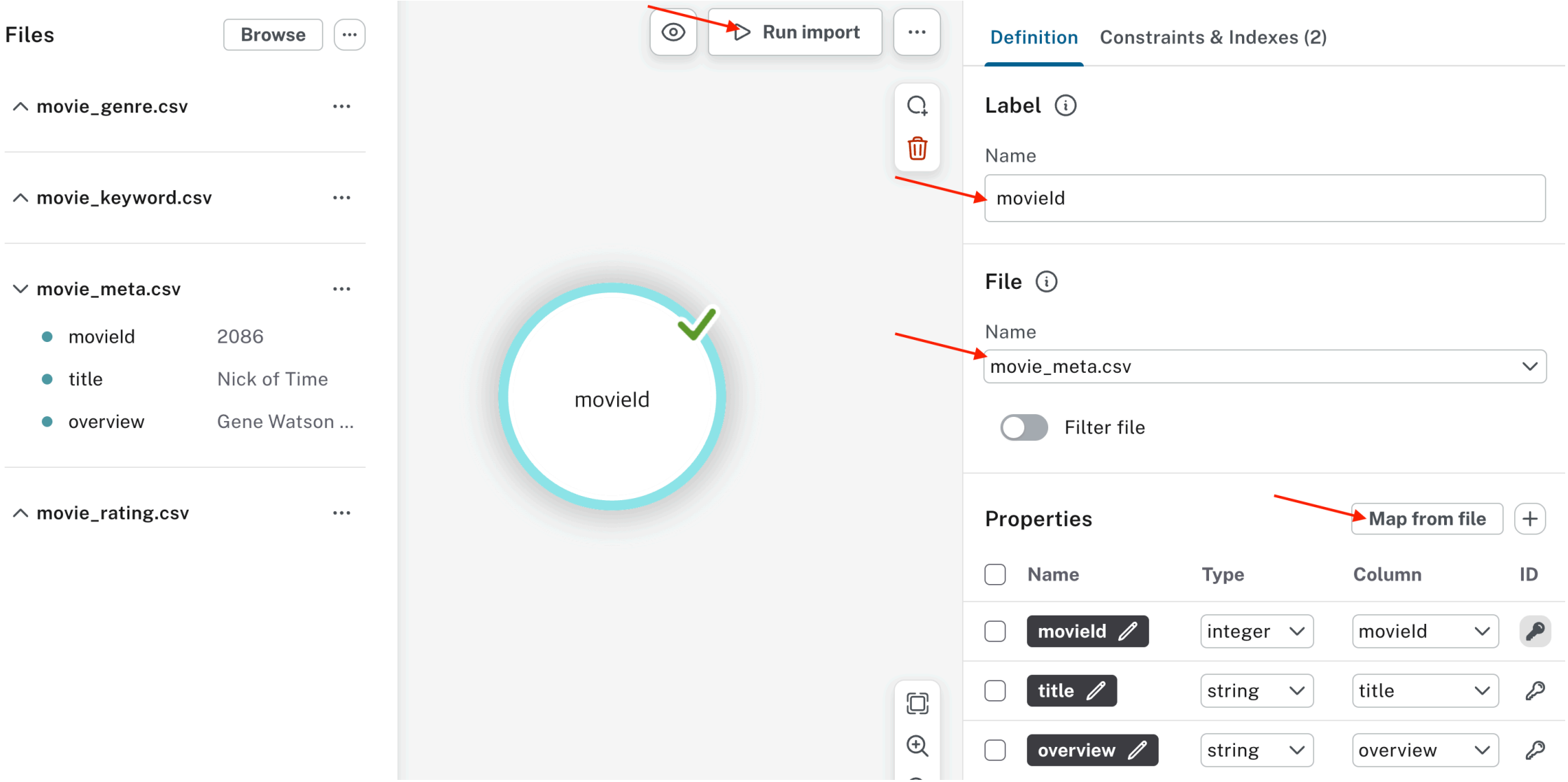
- 생성된 DB의 접속 정보 사용



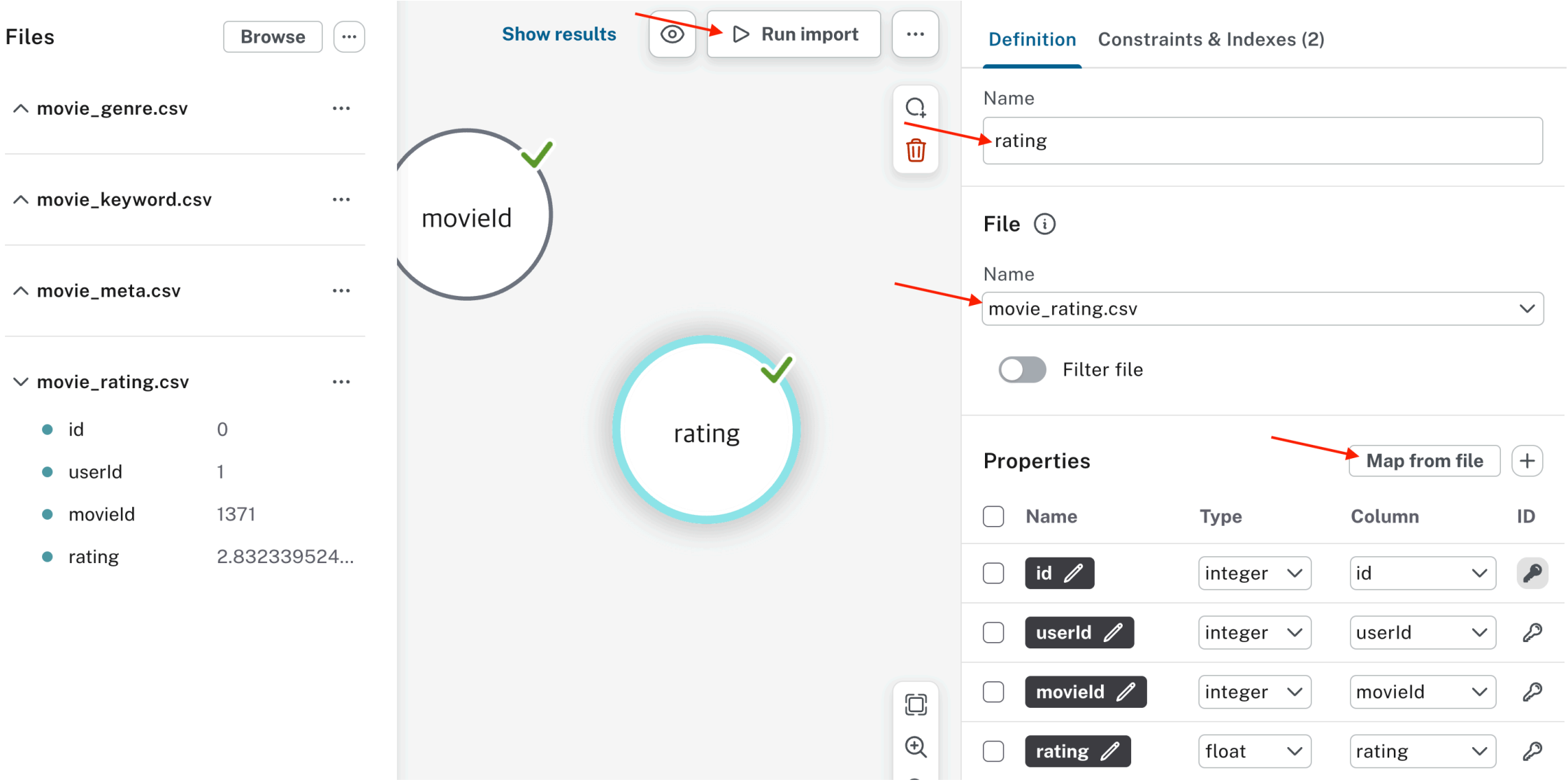
단계2: 파일 업로드


Add node
Label
Node name
Drag & Drop
or browse
Supports CSV, TSV
다운로드
즐겨찾기
AirDrop
최근 항목
응용 프로그램
데스크탑
문서
다운로드
iCloud
iCloud Drive
공유
위치
이름
크기
종류
추가된 날짜
Data Preprocessing.ipynb 336KB 문서 오늘 오후 4:03
movie_meta.csv 489KB CSV 문서 오늘 오후 4:03
movie_rating.csv 855KB CSV 문서 오늘 오후 4:03
movie_keyword.csv 253KB CSV 문서 오늘 오후 4:03
movie_genre.csv 49KB CSV 문서 오늘 오후 4:02
Spectacle 4.4MB 응용 프로그램 2022년 1월 27일 오전 8:
Mapping

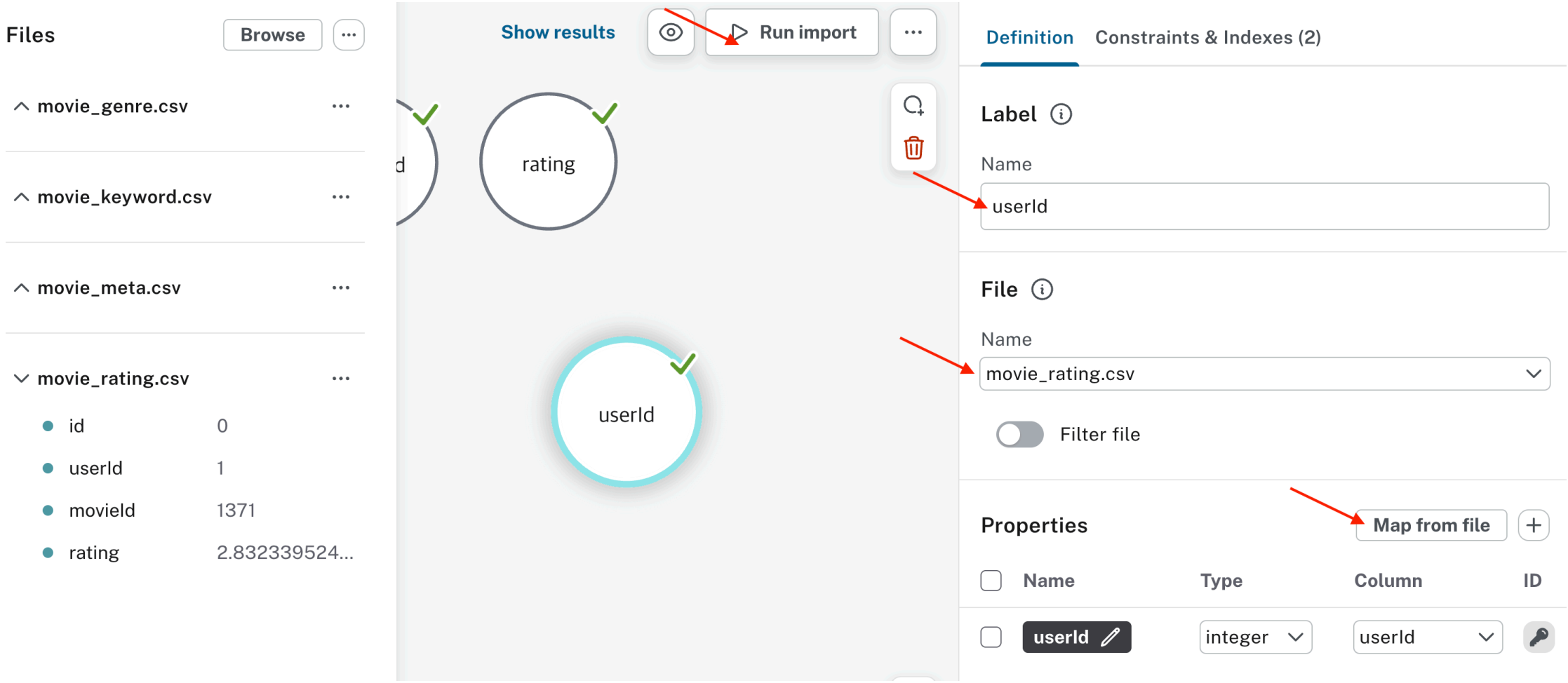
단계3: movieId 노드 생성



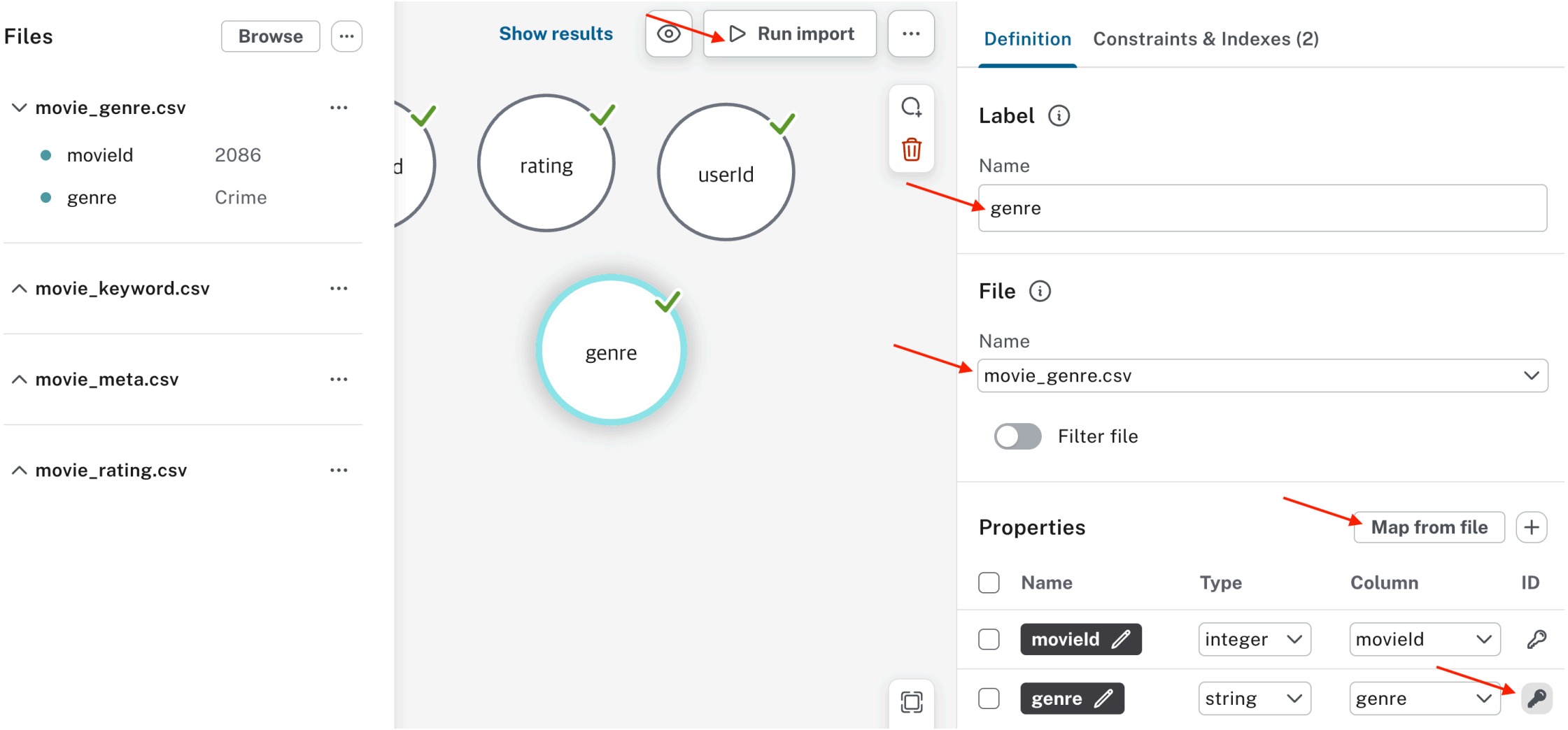
단계4: rating 노드 생성



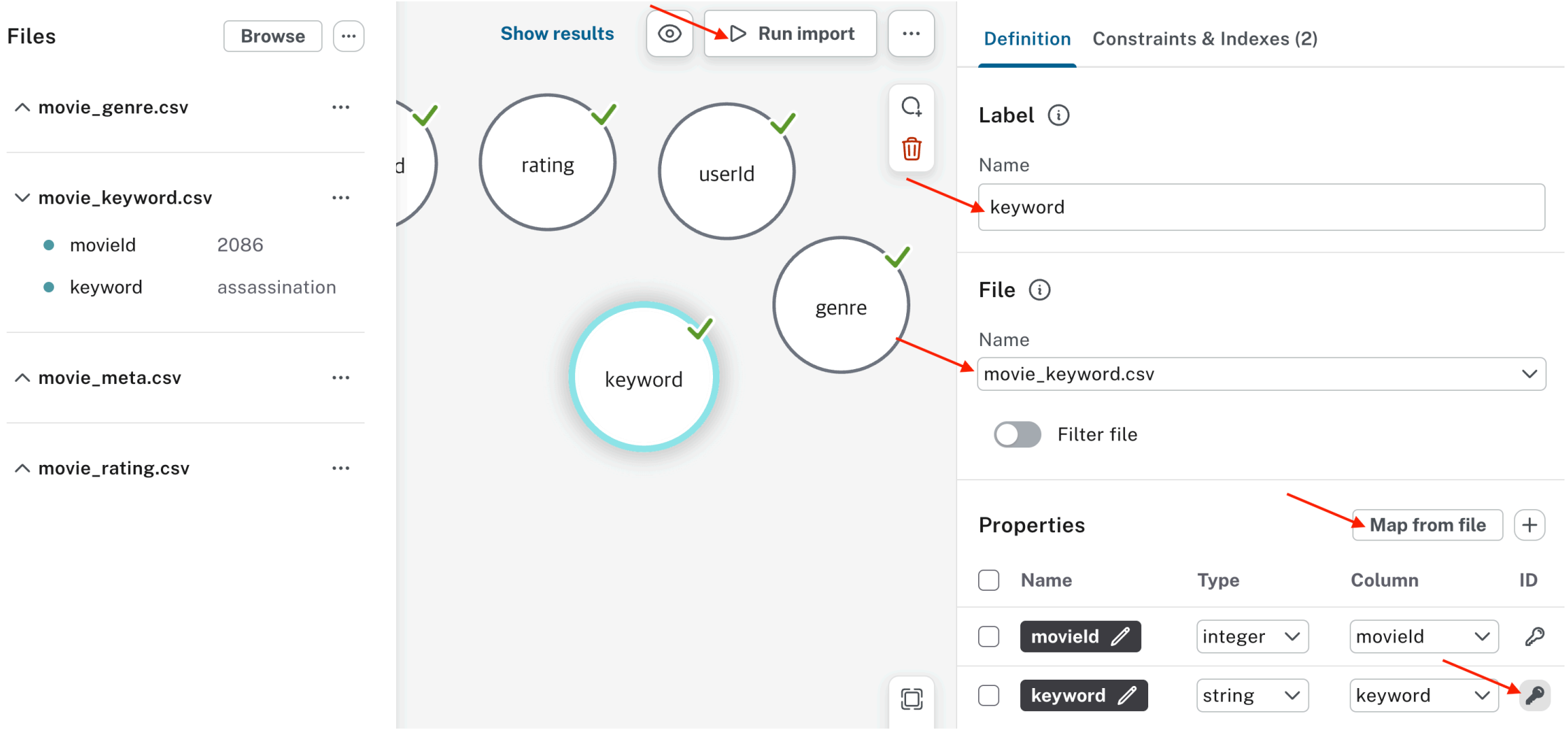
단계5: userId 노드 생성



단계6: genre 노드 생성

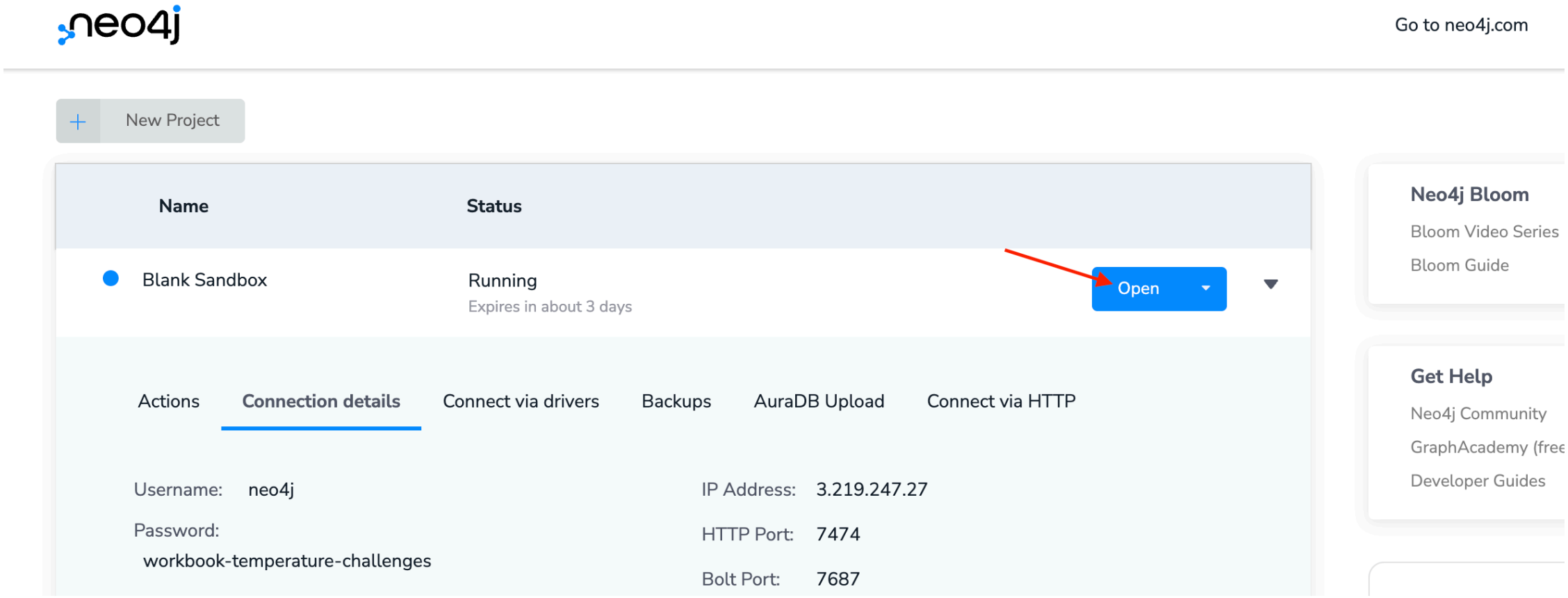


단계7: keyword 노드 생성

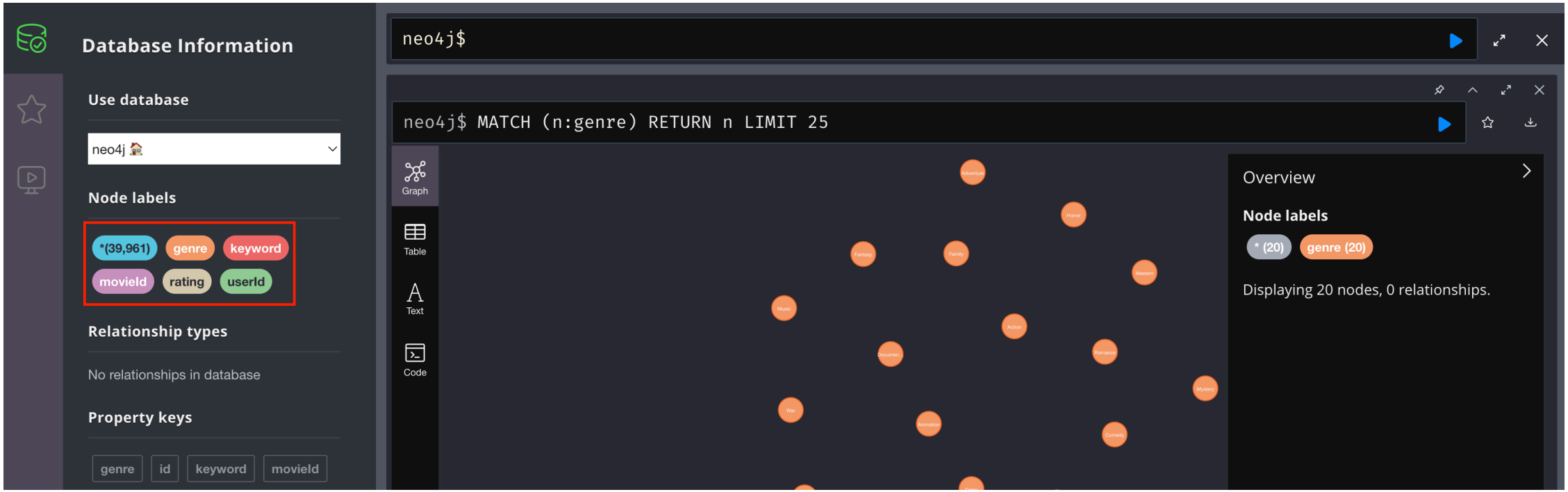


관계성 생성

단계1: Neo4j Sandbox 접속

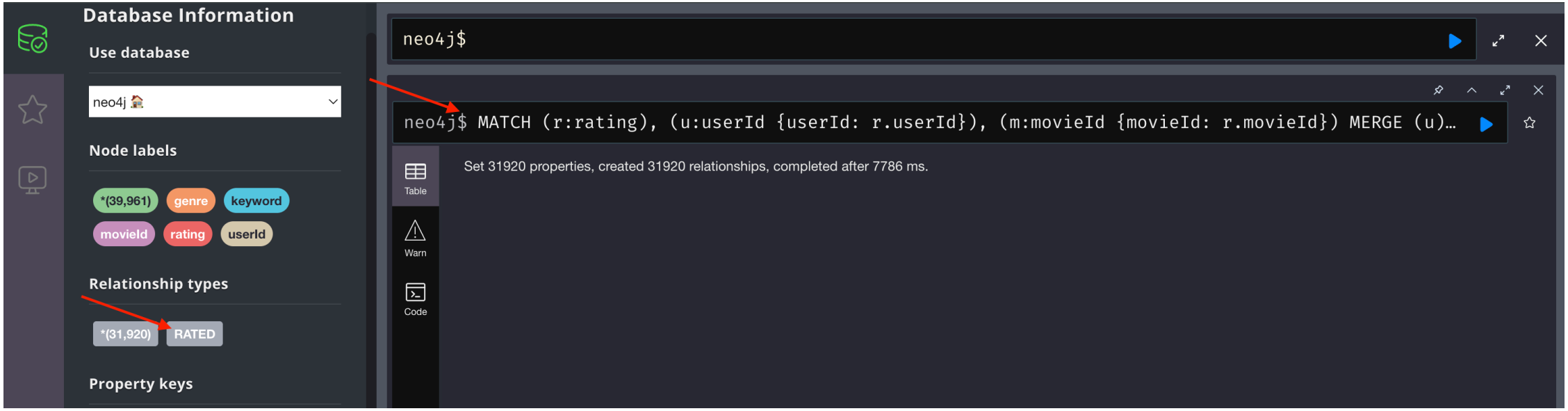


단계2: Nodes 확인

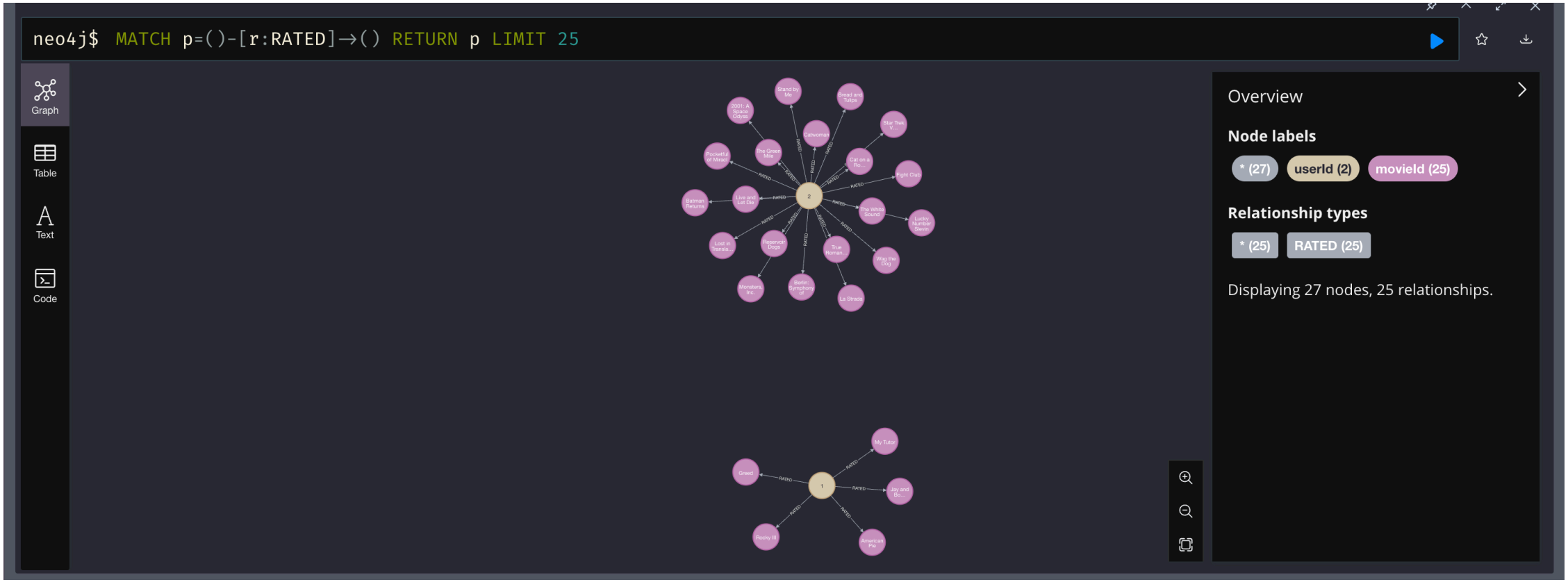

Database Information
Use database
neo4j
Node labels
*(39,961)
genre
keyword
movieId
rating
userId
Relationship types
No relationships in database
Property keys
genre
id
keyword
movieId
neo4j$
neo4j$ MATCH (n:genre) RETURN n LIMIT 25
Graph
Table
Text
Code
Overview
Node labels
* (20)
genre (20)
Displaying 20 nodes, 0 relationships.

단계3: RATED 생성

```
MATCH (r:rating), (u:userId {userId: r.userId}), (m:movieId {movieId: r.movieId})
MERGE (u)-[:RATED {score: r.rating}]->(m)
```

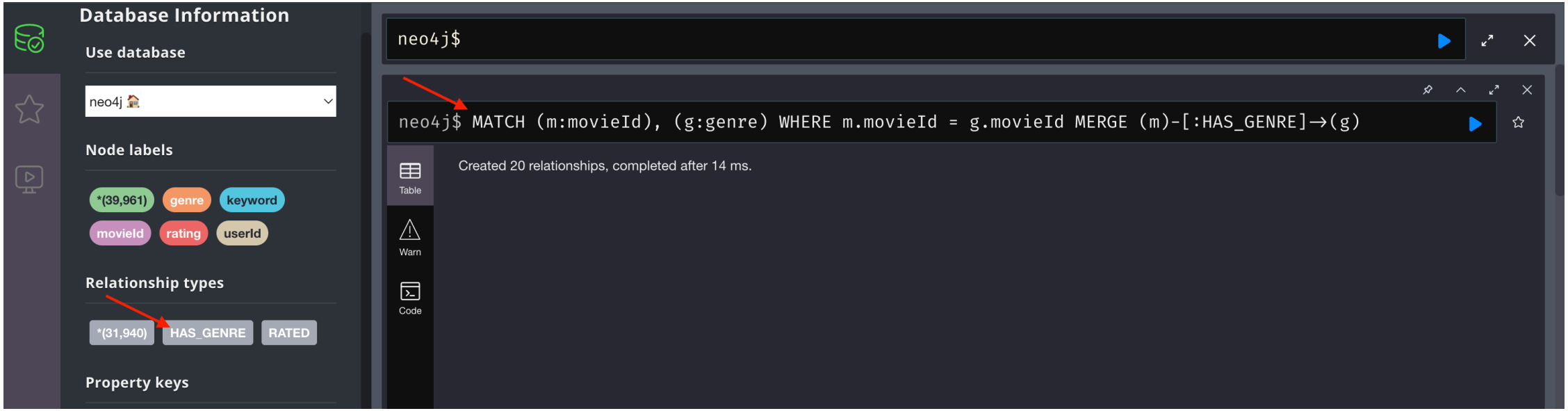


```
MATCH p=()-[r:RATED]->() RETURN p LIMIT 25
```

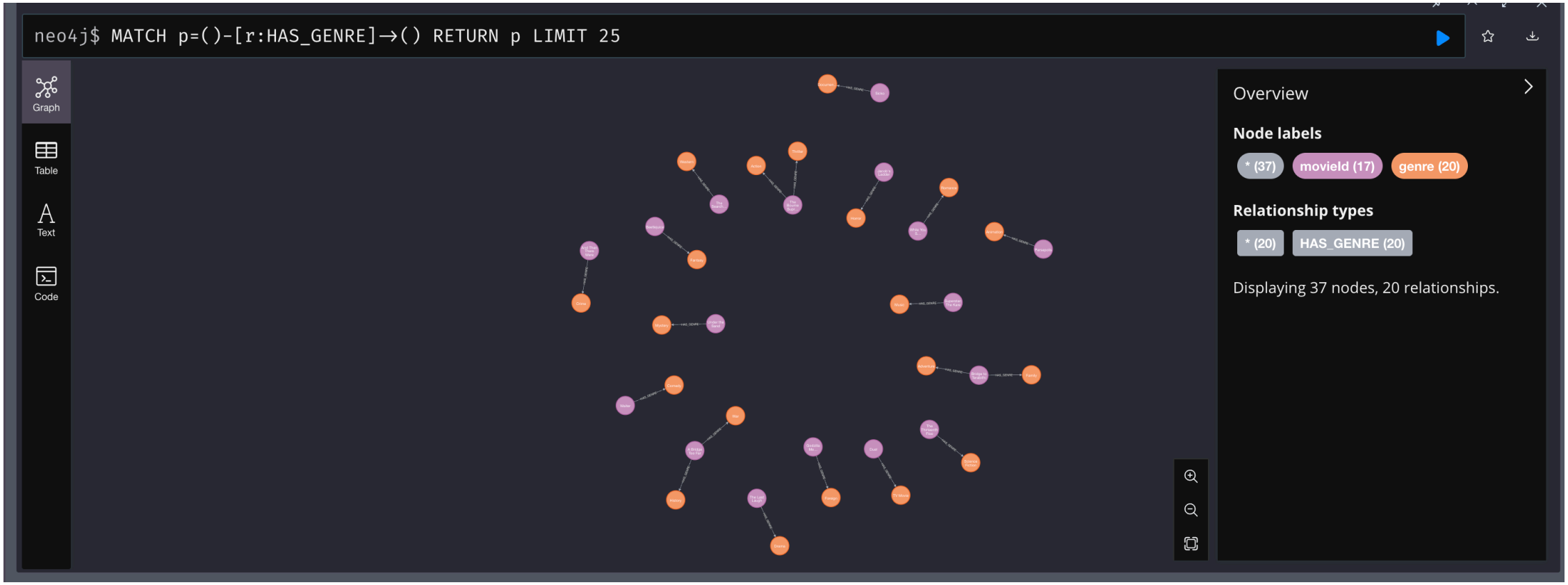


단계4: HAS_GENRE 생성

```
MATCH (m:movieId), (g:genre)
WHERE m.movieId = g.movieId
MERGE (m)-[:HAS_GENRE]->(g)
```

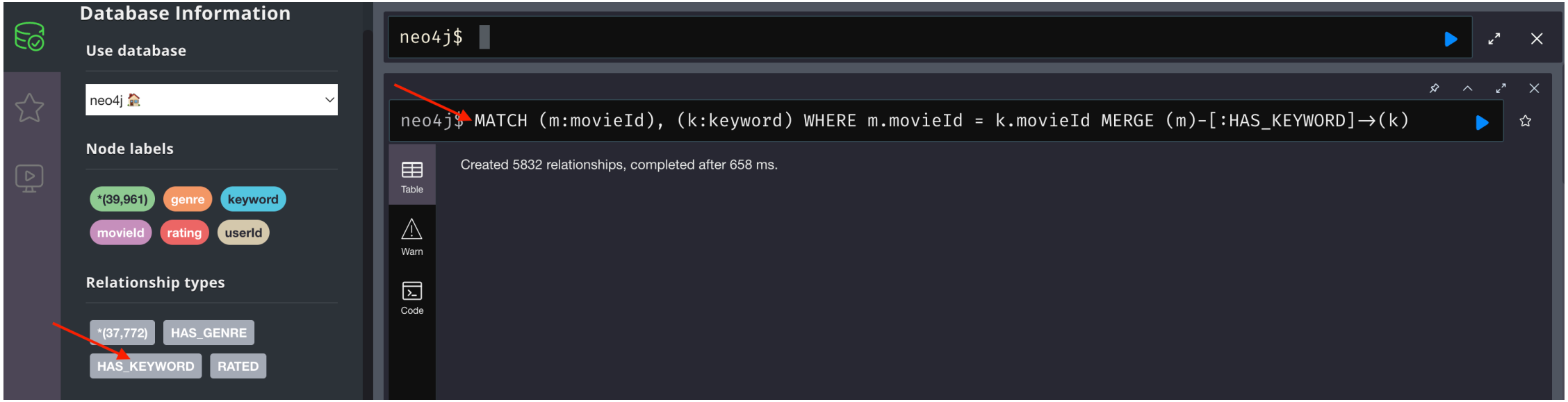


```
MATCH p=()-[r:HAS_GENRE]->() RETURN p LIMIT 25
```



단계5: HAS_KEYWORD 생성

```
MATCH (m:movieId), (k:keyword)
WHERE m.movieId = k.movieId
MERGE (m)-[:HAS_KEYWORD]->(k)
```



```
MATCH p=()-[r:HAS_KEYWORD]->() RETURN p LIMIT 25
```