**LOADING CSV files**

1. Внесување CSV фајлови е функција која е достапна во Neo4j за внесување на податоци.
2. Формат на CSV

CSV треба да има header (прва линија со имиња на колони), на пример:

name,age,city

Ана,25,Скопје -ова се редовите во фајлот

Бојан,30,Битола

3.Имаме 2 начини на внесување CSV фајлови во Neo4j

3.1 Со користење на моќната Cypher  команда **LOAD CSV**. Овој метод е корисен кога имаме мали до средно големи податочни множества.

**Прв чекор**: Фајлот треба да биде достапен од Neo4j серверот. Најлесно е да го ставиме во import фолдерот каде што е инсталиран Neo4j (на пример: /var/lib/neo4j/import )

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

* **WITH HEADERS** :означува дека првиот ред од CSV-фајлот содржи хедери (имиња на колони), кои понатаму може да се користат како имиња на полињата.
* **AS row**: На секој ред од CSV-фајлот му доделува променлива со име row, преку која се пристапува до вредностите од тој ред.
* file:///your\_file.csv – фајлот мора да биде во import директориумот! (your\_file е името на вашиот фајл)
* row.column1, row.column2 се колоните.
* **CREATE или MERGE**: Се користат за креирање (CREATE) или ажурирање (MERGE) на јазли и релации врз основа на податоците од CSV-фајлот.

Пример:

Нека датотеката friends.csv е следната

from,to,since

Ана,Бојан,2015

Бојан,Гоце,2018

LOAD CSV WITH HEADERS FROM 'file:///friends.csv' AS row

CREATE (a:Person {name: row.from})

CREATE (b:Person {name: row.to})

CREATE (a)-[:FRIEND\_WITH ]->(b) -се креира релација од јазол а до јазол б

Но доколку веќе постојат јазли со тие имиња,место да ги креираме **CREATE** може да искористиме **MERGE** за да избегнеме дупликати.

**MERGE** проверува дали таков јазол веќе постои – ако постои, го користи него; ако не, создава нов а CREATE секогаш креира нов јазол

MERGE (a:Person {name: row.from})

MERGE (b:Person {name: row.to})

CREATE (a)-[:FRIEND\_WITH]->(b)

Креирање релација со **MATCH**

MATCH (a:Person {name: row.from}), (b:Person {name: row.to})

**MATCH** е команда за пребарување на јазли или релации во графот. Оваа команда ја наоѓа двојката лица во графот, едниот по името од колоната from, другиот по името од колоната to и се креира релација меѓу нив

**UNWIND**

Доколку имаш податоци во JSON ,можеш да ги префрлиш со UNWIND:

A screenshot of a computer code

AI-generated content may be incorrect.

**ВТОР НАЧИН НА ВНЕСУВАЊЕ CSV фајлови**

Користење на neo4j-admin database import команда

* Овој метод е наменет за внесување на многу големи сетови на податоци (над 10 милиони записи) и бара базата да биде офлајн за време на внесот.Подготовка:
* CSV-датотеките треба да бидат специјално структурирани за оваа алатка — одделни датотеки за јазли и релации, и специјални колони како :ID, :START\_ID, :END\_ID, и :TYPE.

Извршување:

* Командата се стартува од терминал, при што се специфицираат патеките до CSV-датотеките за јазли и релации.

Пример команда:

neo4j-admin database import full --nodes=Person=import/persons.csv --relationships=FRIEND\_WITH=import/friends.csv --database=graph.db

Практична работа

Внеси податоци за луѓе,пријателства и градови.

1. Пронајди ги сите пријатели на Ана кои живеат во ист град како Ана, но се постари од неаA black background with white text

   AI-generated content may be incorrect.
2. Пронајди го најкраткиот пат помеѓу двајца корисници и испиши ги сите имиња на јазлите низ кои минува патот:

A black background with white text

AI-generated content may be incorrect.

За секој човек, колку пријатели има што живеат во различен град од него?

A black background with white text

AI-generated content may be incorrect.

Пронајди ги сите лица што немаат пријатели:

A black background with white text

AI-generated content may be incorrect.

Пронајди го човекот кој има најмногу пријатели

A black screen with white text

AI-generated content may be incorrect.