**Import MySQL vs MongoDB**

* **Exporteren vanuit Neo4j**
  + Mogelijk formaten
    - **JSON**: geschikt voor document-gebaseerde databases zoals MongoDB. Ondersteunt geneste structuren, maar diepe relaties kunnen complex worden.
    - **CSV**: Geschikt voor relationele databases zoals MySQL. Nodig als je tabellen en kolommen wilt vullen.
  + Voordelen
    - Flexibel exporteren met Cypher-query’s
    - Mogelijkheid om relaties of nodes afzonderlijk te exporteren
  + Nadelen
    - JSON-export kan lastig worden bij complexen grafen (geneste relaties of veel nodes)
    - CSV-export kan veel voorbereidend werk vergen (joins in Cypher, meerdere bestanden voor nodes en relaties).
* **Importeren naar MongoDB**
  + Proces
    - JSON-export van Neo4j 🡪 mongoimport of via een script.
  + Voordelen
    - JSON is direct bruikbaar als document in mongoDB.
    - Geen strikt schema nodig, dus minder conversies.
    - Geschikt voor semi-gestructureerde data.
  + Nadelen
    - Bij diep geneste realties kan import lastig zijn (bijvoorbeeld arrays van geneste nodes).
    - Data-integriteit van relaties in niet automatisch gegarandeerd (je moet zelf referenties of embedded documents structureren).
    - Zeer grote JSON-bestanden kunnen importproblemen veroorzaken.
* **Importeren naar MySQL**
  + Proces
    - JSON 🡪 CSV (meestal per node type of per relatie) 🡪 importeren via **LOAD DATA INFILE** of script
  + Voordelen
    - CSV-import is gestandaardiseerd en goed ondersteund.
    - Relaties kunnen via foreign keys worden gemodelleerd, waardoor data-integriteit gegarandaard is
  + Nadelen
    - JSON kan niet rechtstreeks worden geïmporteerd, er zijn extra conversie stappen nodig naar CSV
    - Complexe grafstructuren vereisen meerdere tabellen (vb: nodes en realties in aparte tabellen).
    - Importscripts zijn langer en complexer vanwege mapping van relaties naar joins.
    - Minder flexibel bij wijzigingen in het datamodel.
    - Nodige import script is veel langer dan bij MongoDB

|  |  |  |
| --- | --- | --- |
| **Kenmerk** | **MongoDB** | **MySQL** |
| Input format van Neo4j export | JSON | CSV (meestal) |
| Directe import | Ja, met mongoimport | Nee, eerst converteren naar CSV |
| Relaties behouden | Alleen via embedded docs of refs | Ja, via fk (maar mapping nodig) |
| Complexiteit van import | Laag tot medium bij diepe JSON | Medium tot hoog (meerdere tabellen + scripts) |
| Schaalbaarheid van import | Goed, mits bestanden niet te groot | Goed, maar CSV-bestanden moeten splitsen bij grote datasets |

**Kortom:**

* **MongoDB:** makkelijk bij JSON, maar relaties moeten handmatig worden beheerd.
* **MySQL**: goed voor behoud van relaties en integriteit, maar JSON-import is omslachtig en vergt CSV-conversie.