Cyphercode zum Import der Tabellendaten in neo4j

Arbeitsstand 12.04.2024

```
MATCH (n) DETACH DELETE n;
CALL apoc.schema.assert({},{},true) YIELD label, key
RETURN *;

//index Knotentyp(property)
CREATE INDEX IF NOT EXISTS FOR (n:Person) ON (n.name);
CREATE INDEX IF NOT EXISTS FOR (n:Person) ON (n.pid);
CREATE INDEX IF NOT EXISTS FOR (n:Quelle) ON (n.label);
CREATE INDEX IF NOT EXISTS FOR (n:Gemeindedienst) ON (n.name);
CREATE INDEX IF NOT EXISTS FOR (n:Ort) ON (n.ortsId);
CREATE INDEX IF NOT EXISTS FOR (n:Ort) ON (n.name);
CREATE INDEX IF NOT EXISTS FOR (n:Taetigkeit) ON (n.name);
CREATE INDEX IF NOT EXISTS FOR (n:Titel) ON (n.label);


// Personen-Tabelle importieren
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=1307737667&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Name IS NOT NULL
WITH line
CREATE (p:Person {pid:line.ID,nachname:line.`Nachname`, name:line.`Name`,
namensvariation:line.`Namensvariation`, g:line.`G`, schutzherr:line.`Schutzherr`,
schutzerteilung:line.`Schutzerteilung`, gemeindedienst:line.`Gemeindedienst`,
verheiratetMit:line.`verheiratetMit`, kindVon1:line.`kindVon1`, kindVon2:line.`kindVon2`,
geschwister:line.`Geschwister`, schwiegervater:line.`schwiegervater`,
Taetigkeit:line.`Taetigkeit`, religion:line.`Religionszugehörigkeit`, todNorm:line.`TodNorm`,
todesjahr:line.`Todesjahr`, todesort:line.`Todesort`, geburtsjahr:line.`Geburtsjahr`,
geburtNorm:line.`GeburtNorm`, begraebnisort:line.`Begräbnisort`, alterTod:line.`Alter
(Tod)`, geburtsort:line.`Geburtsort`});

// Orte importieren
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=1967277549&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Ort IS NOT NULL
WITH line
```

```
CREATE (o:Ort {ortsId:line.OrtsId, name:line.Ort, latitude:line.Latitude,
longitude:line.Longitude, wikidataUrl:line.WikidataUrl, wikipediaUrl:line.WikipediaUrl});
// Anzeigename erstellen
MATCH (p:Person)
WHERE p.name IS NOT NULL
AND p.nachname IS NOT NULL
SET p.anzeigename = p.nachname + ', ' + p.name
RETURN p.anzeigename;

MATCH (p:Person)
WHERE p.name IS NOT NULL
AND p.nachname IS NULL
SET p.anzeigename = p.name
RETURN p.anzeigename;

MATCH (p:Person)
WHERE p.name IS NULL
AND p.nachname IS NOT NULL
SET p.anzeigename = p.nachname
RETURN p.anzeigename;

LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=867639424&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Gemeindedienst IS NOT NULL
WITH line
MATCH (p:Person {pid:line.PersonenId})
MERGE (g:Gemeindedienst {name:line.Gemeindedienst})
CREATE (r:Rolle {name:line.Gemeindedienst, von:line.GemeindedienstAb,
bis:line.GemeindedienstBis, genannt:line.Genannt})
CREATE (p)-[:GEMEINDEDIENST]->(r), (r)-[:ROLLENTYP]->(g)
WITH line, r WHERE line.Ortsname IS NOT NULL
MATCH (o:Ort {name:line.Ortsname})
MERGE (r)-[:ORT]->(o);

//Taetigkeiten importieren
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=1641076151&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Taetigkeit IS NOT NULL
WITH line
MATCH (p:Person {pid:line.PersonenId})
MERGE (t:Taetigkeit {name:line.Taetigkeit})
```

```
CREATE (r:Rolle {name:line.Taetigkeit, von:line.TaetigkeitAb, bis:line.TaetigkeitBis,
genannt:line.Genannt})
CREATE (p)-[:TAETIGKEIT]->(r), (r)-[:ROLLENTYP]->(t)
WITH line, r WHERE line.Ortsname IS NOT NULL
MATCH (o:Ort {name:line.Ortsname})
MERGE (r)-[:ORT]->(o);


// Todesort erstellen und verknüpfen
MATCH (p:Person)
WHERE p.todesort IS NOT NULL
MERGE (o:Ort {name:p.todesort})
MERGE (o)<-[:VERSTORBEN_IN]-(p)
RETURN *;

// Begräbnisort erstellen und verknüpfen
MATCH (p:Person)
WHERE p.begraebnisort IS NOT NULL
MERGE (o:Ort {name:p.begraebnisort})
MERGE (o)<-[:BEGRABEN_IN]-(p)
RETURN *;

// Geburtsort erstellen und verknüpfen
MATCH (p:Person)
WHERE p.geburtsort IS NOT NULL
MERGE (o:Ort {name:p.geburtsort})
MERGE (o)<-[:GEBOREN_IN]-(p)
RETURN *;


//Schutzherr erstellen
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=972249366&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Schutzherr IS NOT NULL
AND line.Herrschaftstyp IS NOT NULL
WITH line
MATCH (p:Person {pid:line.PersonenID})
MERGE (sb:Schutzbrief {name:line.Schutzherr,
typ:line.Herrschaftstyp})
ON CREATE SET
sb.obrigkeit = line.behördlicheObrigkeit,
sb.von = line.SchutzerteilungAb,
sb.bis = line.SchutzerteilungBis,
sb.genannt = line.Genannt
MERGE (sh:Schutzherr {name:line.Schutzherr})
```

```
MERGE (p)-[:SCHUTZBRIEF]->(sb)
MERGE (sb)-[:SCHUTZHERR]->(sh)
WITH line, sb WHERE line.Ortsname IS NOT NULL
MATCH (o:Ort {name:line.Ortsname})
MERGE (sb)-[:SCHUTZORT]->(o)
RETURN *;


//Wohnorte importieren
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=113830897&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line WHERE line.Ort IS NOT NULL
WITH line
MATCH (p:Person {pid:line.PersonenId})
MATCH (o:Ort {ortsId:line.OrtsId})
MERGE (p)-[r:WOHNORT]->(o)
ON CREATE SET r.startDate = line.WohntInAb,
r.enddate = line.WohntInBis,
r.date = line.Genannt,
r.kommentar = line.line.Kommentar,
r.hochzeit = line.Hochzeit;

// Wohnort aus Geburtsort erschließen
match (o:Ort)<-[:GEBOREN_IN]-(p:Person)
where not (o)<-[:WOHNORT]-(p)
merge (o)<-[:WOHNORT {type:'erschlossen'}]-(p)
return count(*);

// Kommentare extrahieren
MATCH (n:Person)
WHERE n.verheiratetMit CONTAINS '['
WITH n
UNWIND apoc.text.regexGroups(n.verheiratetMit, "(\\d+)\\[(.*)\\]") as k
SET n.verheiratetMit = k[1]
SET n.verheiratetMitKommentar = k[2]
RETURN *;


// Verheiratet mit erstellen
MATCH (n:Person)
WHERE n.verheiratetMit IS NOT NULL
WITH n
FOREACH ( j in split(n.verheiratetMit, ",") |
MERGE (n2:Person {pid:trim(j)})
MERGE (n)-[:VERHEIRATET_MIT]->(n2)
MERGE (n)<-[:VERHEIRATET_MIT]-(n2))
```

```
RETURN *;

// Kind von erstellen
MATCH (n:Person)
WHERE n.kindVon1 IS NOT NULL
WITH n
MATCH (n2:Person {pid:n.kindVon1})
MERGE (n)-[:KIND_VON]->(n2);
MATCH (n:Person)
WHERE n.kindVon2 IS NOT NULL

WITH n
MATCH (n2:Person {pid:n.kindVon2})
MERGE (n)-[:KIND_VON]->(n2);

// Tätigkeitsliste importieren
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=940017382&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line
WHERE line.Gesamtname IS NOT NULL
AND line.Unterkategorie IS NOT NULL
WITH line
MATCH (r:Rolle)-[:ROLLENTYP]->(t:Taetigkeit {name:line.Gesamtname})
MERGE (r1:Unterkategorie {name:line.Unterkategorie})
MERGE (r2:Oberkategorie {name:line.Oberkategorie})
MERGE (r)-[:ROLLE]->(r1)
MERGE (r1)-[:SUB_OF]->(r2);
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=940017382&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line
WHERE line.Gesamtname IS NOT NULL
AND line.Unterkategorie IS NULL
WITH line
MATCH (r:Rolle)-[:ROLLENTYP]->(t:Taetigkeit {name:line.Gesamtname})
MERGE (r2:Oberkategorie {name:line.Oberkategorie})
MERGE (r)-[:ROLLE]->(r2);
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
// Quellen erstellen
LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=1961926803&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line
WHERE line.ID IS NOT NULL
MATCH (p:Person {pid:line.ID})
FOREACH ( j in split(line.Quellennachweise, ";") |
MERGE (q:Quelle {label:trim(j)})
MERGE (p)-[:QUELLE]->(q))
RETURN *;


// Familienverband erstellen

LOAD CSV WITH HEADERS FROM
"https://docs.google.com/spreadsheets/d/e/2PACX-
1vRpp_pj9wiVn3qbZ2FFIOhTOk9zGcvNqXYi0-cH3mfO2XoT1QnPPnLHPP58LNNYYF9uQHe-
UGtxwMPp/pub?gid=266265472&single=true&output=csv"
AS line FIELDTERMINATOR ','
WITH line
WHERE line.Relevanz IS NOT NULL
AND line.Familienverband IS NOT NULL
WITH line
MATCH (n1:Person {pid:line.pid})
SET n1.familienverband = line.Familienverband
SET n1.relevanz = toInteger(line.Relevanz)
SET n1.pathLength = '0'
MERGE (s:Familienverband {label:line.Familienverband, relevanz:line.Relevanz})
MERGE (n1)-[:FAMILIENVERBAND]->(s)
WITH n1, s, line
MATCH p=(n1)<-[r:KIND_VON*]-(n2)
SET n2.pathLength = toString(length(p))
SET n2.familienverband = line.Familienverband
MERGE (n2)-[:FAMILIENVERBAND {pathLength:length(p),
familienverband:line.Familienverband}]->(s);
```