# Load addresses:

LOAD CSV WITH HEADERS FROM 'file:///nodes-addresses.csv' AS line FIELDTERMINATOR ','

CREATE (:Address {node\_id: toInteger(line.node\_id), address: line.address, name: line.name, country: line.countries})

# Load Entities:

LOAD CSV WITH HEADERS FROM 'file:///nodes-entities.csv' AS line FIELDTERMINATOR ','

CREATE (:Entities {node\_id: toInteger(line.node\_id), name: line.name, original\_name: line.original\_name, former\_name: line.original\_name, former\_name: line.former\_name, company\_type: line.company\_type, service\_provider: line.service\_provider})

# Load Intermediaries:

LOAD CSV WITH HEADERS FROM 'file:///nodes-intermediaries.csv' AS line FIELDTERMINATOR ','

CREATE (:Intermediary {node\_id: toInteger(line.node\_id), address: line.address, name: line.name})

# Load Officers:

LOAD CSV WITH HEADERS FROM 'file:///nodes-officers.csv' AS line FIELDTERMINATOR ','

CREATE (:Officer {node\_id: toInteger(line.node\_id), country: line.countries, name: line.name, country: line.countries})

# Load Others:

LOAD CSV WITH HEADERS FROM 'file:///nodes-others.csv' AS line FIELDTERMINATOR ','

CREATE (:Others {node\_id: toInteger(line.node\_id), name: line.name, country: line.countries, type: line.type, country: line.countries})

# Create constraints (indexes auf node\_id’s warden auch nebenbei generiert -> das muss nicht mehr erledigt werden)

CREATE CONSTRAINT ON (n: Address) ASSERT n.node\_id IS UNIQUE;

CREATE CONSTRAINT ON (n: Entities) ASSERT n.node\_id IS UNIQUE;

CREATE CONSTRAINT ON (n: Officer) ASSERT n.node\_id IS UNIQUE;

CREATE CONSTRAINT ON (n: Intermediary) ASSERT n.node\_id IS UNIQUE;

CREATE CONSTRAINT ON (n: Others) ASSERT n.node\_id IS UNIQUE;

# Load Relations (Plugin APOC benötigt / in config: apoc.import.file.enabled=true):

CALL apoc.periodic.iterate(

'CALL apoc.load.csv("file:///relationships.csv", {header:true, sep:","}) yield map as line ',

'MATCH(a),(b) '+

'WHERE a.node\_id = line.\_start AND b.node\_id = line.\_end '+

'WITH a, b, line '+

'CALL apoc.create.relationship(a, line.\_type, {}, b) YIELD rel '+

'return rel',

{batchSize:1000, parallel:true}

);