**Lab Exercise 3**

**1.1 Exercise:**

For this exercise you are required to perform the same temporal analysis that you did in this

lab, but on a different dataset. To import the dataset run the following statements in your

Neo4J browser editor (they are the same statements as the ones in the “Environment” section

but with different URLs):

LOAD CSV WITH HEADERS FROM

"https://gist.githubusercontent.com/apogiatzis/f7793b9707f3b15f5e811acb4f9d82bb/raw/d5e769a47955fd79ca3b00efd82d48041f883a31/persons.csv" AS csvLine

MERGE (e:Email {address: csvLine.email})

MERGE (t:Phone {number: csvLine.phone})

CREATE (p:Person {obj\_id: csvLine.obj\_id, first\_name:

csvLine.first\_name, last\_name: csvLine.last\_name, date\_of\_birth:

date(csvLine.date\_of\_birth), occupation: csvLine.occupation,

nationality: csvLine.nationality, phone: csvLine.phone, ni\_number:

csvLine.ni\_number})

CREATE (p)-[:HAS\_EMAIL]->(e)

CREATE (p)-[:HAS\_PHONE]->(t)

Text

Description automatically generated

LOAD CSV WITH HEADERS FROM

"https://gist.githubusercontent.com/apogiatzis/ce3572b2671730a95ff2d96e3318764d/raw/e4d29e632f1510594f499608983c475da0143e68/accounts.csv" AS csvLine

MATCH (p:Person {obj\_id: csvLine.owners})

MERGE (b:Bank {name: csvLine.bank})

CREATE (a:Account {obj\_id: csvLine.obj\_id, balance:

toFloat(csvLine.balance), currency: csvLine.currency, owner:

csvLine.owners, opened\_at: datetime(replace(csvLine.opened\_at, ' ',

'T'))})

CREATE (p)-[:OWNS]->(a)

CREATE (a)-[:PROVIDED\_BY]->(b)

Graphical user interface, text

Description automatically generated

LOAD CSV WITH HEADERS FROM

"https://gist.githubusercontent.com/apogiatzis/e8188df98dbc8810d49f274af7e607ac/raw/c014fc9cdd1d4f024b910af4966a5e2ab5a104f9/transactions.csv"

AS csvLine

MATCH (sender:Account {obj\_id: csvLine.sender})

MATCH (recipient:Account {obj\_id: csvLine.recipient})

CREATE (t:Transaction:Transfer {obj\_id: csvLine.obj\_id, amount:

toFloat(csvLine.amount), currency: csvLine.currency, type:

csvLine.type, timestamp: datetime(replace(csvLine.timestamp, ' ',

'T'))})

CREATE (sender)-[:PERFORMED]->(t)

CREATE (t)-[:TO]->(recipient)

A screenshot of a computer

Description automatically generated with medium confidence

LOAD CSV WITH HEADERS FROM

"https://gist.githubusercontent.com/apogiatzis/04addd6990394aec5695d70e020d7ee3/raw/b63b71c0bcd5275e6c31ee7bc9a1944975be8fe2/deposits.csv" AS csvLine

MATCH (sender:Person {obj\_id: csvLine.sender})

MATCH (recipient:Account {obj\_id: csvLine.recipient})

CREATE (t:Transaction:Deposit {obj\_id: csvLine.obj\_id, amount:

toFloat(csvLine.amount), currency: csvLine.currency, type:

csvLine.type, timestamp: datetime(replace(csvLine.timestamp, ' ',

'T'))})

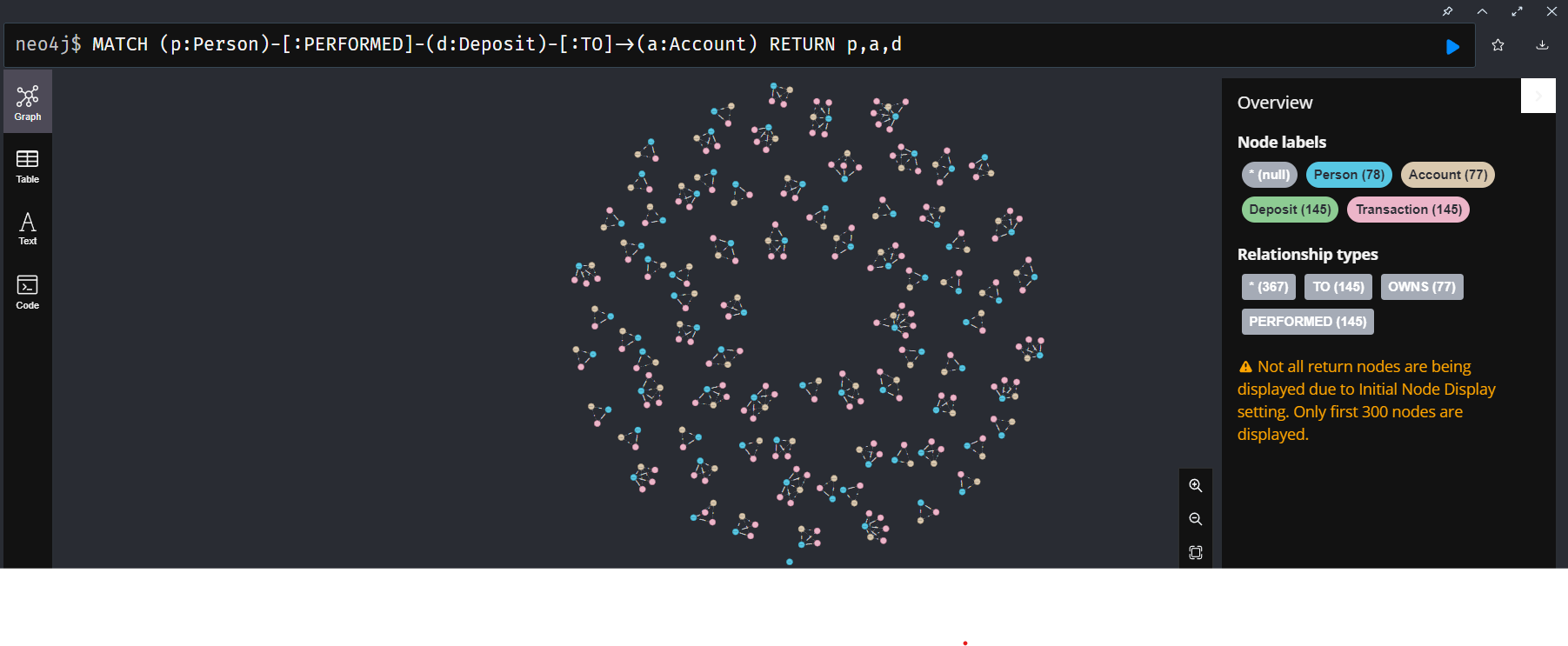
CREATE (sender)-[:PERFORMED]->(t)

CREATE (t)-[:TO]->(recipient)

Graphical user interface, text

Description automatically generated

MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account) RETURN p,a,d



MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account) RETURN

d.amount, count(DISTINCT d.amount) AS freq ORDER BY freq DESC

A screenshot of a computer

Description automatically generated with medium confidence

MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account)

WITH min(d.timestamp) AS min\_date, max(d.timestamp) AS max\_date

RETURN min\_date, min\_date.epochMillis AS min\_date\_epoch, max\_date,

max\_date.epochMillis AS max\_date\_epoch

Graphical user interface, text, application

Description automatically generated

UNWIND range(880092632000, 1644640280000, 86400000) AS t

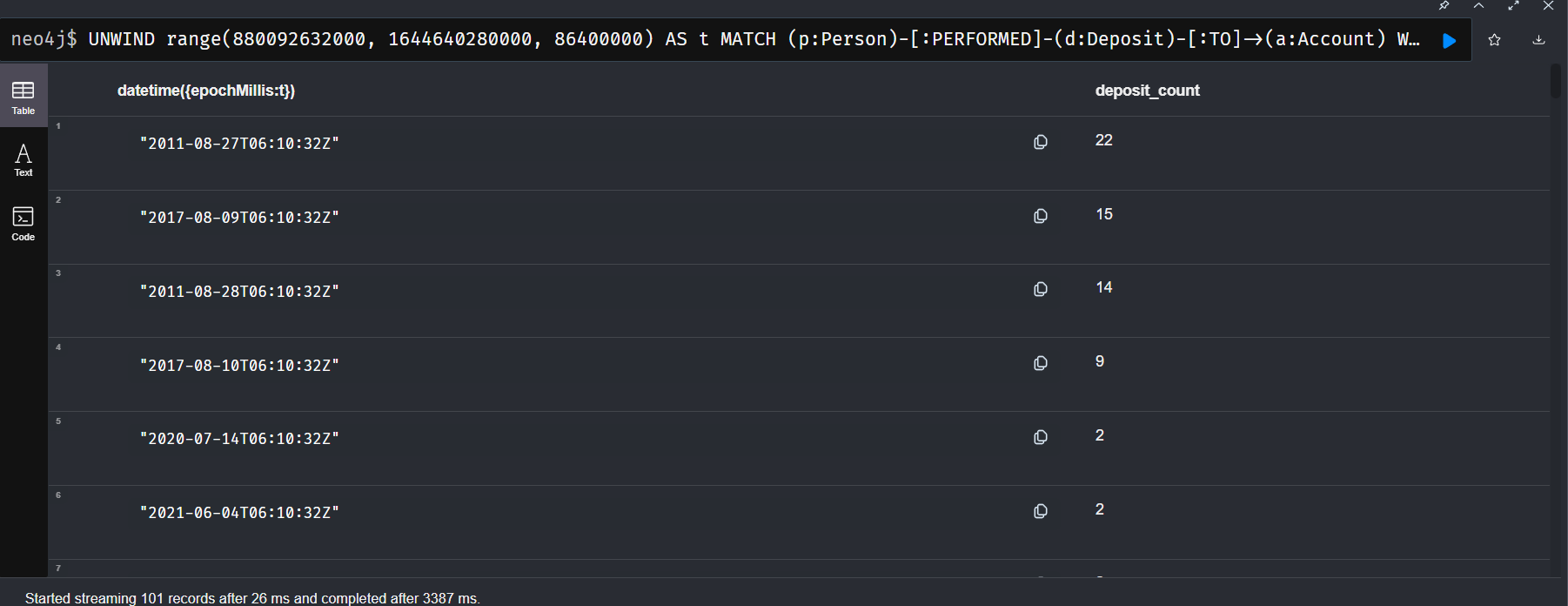
MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account)

WHERE d.timestamp.epochMillis >= t AND d.timestamp.epochMillis < t +

86400000

RETURN datetime({epochMillis:t}), COUNT(\*) as deposit\_count ORDER BY

deposit\_count desc



MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account)

WITH \*,datetime("2011-08-27T06:10:32Z") as window\_start

WHERE d.timestamp.epochMillis >= window\_start.epochMillis AND

d.timestamp.epochMillis < window\_start.epochMillis + 86400000

RETURN p,d,a

A screenshot of a computer

Description automatically generated with medium confidence

MATCH (p1:Person)-[:HAS\_EMAIL|:HAS\_PHONE]->(info)<-

[:HAS\_EMAIL|:HAS\_PHONE]-(p2:Person)

WHERE p1.obj\_id<>p2.obj\_id

WITH p1, p2, count(\*) as cnt

MERGE (p1) - [:SHARED\_IDENTIFIERS {count: cnt}] - (p2);

Graphical user interface, text

Description automatically generated

MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account)

MATCH (p)-[:SHARED\_IDENTIFIERS]-(o:Person)

WITH \*,datetime("2011-08-27T06:10:32Z") as window\_start

WHERE d.timestamp.epochMillis >= window\_start.epochMillis AND

d.timestamp.epochMillis < window\_start.epochMillis + 86400000

RETURN p,d,a,o

A picture containing diagram

Description automatically generated

MATCH (p:Person)-[:PERFORMED]-(d:Deposit)-[:TO]->(a:Account)

MATCH (p)-[:SHARED\_IDENTIFIERS]-(o:Person)

WITH \*,datetime("2011-08-27T06:10:32") as window\_start

WHERE d.timestamp.epochMillis >= window\_start.epochMillis AND

d.timestamp.epochMillis < window\_start.epochMillis + 86400000

RETURN d.obj\_id

Graphical user interface, application

Description automatically generated