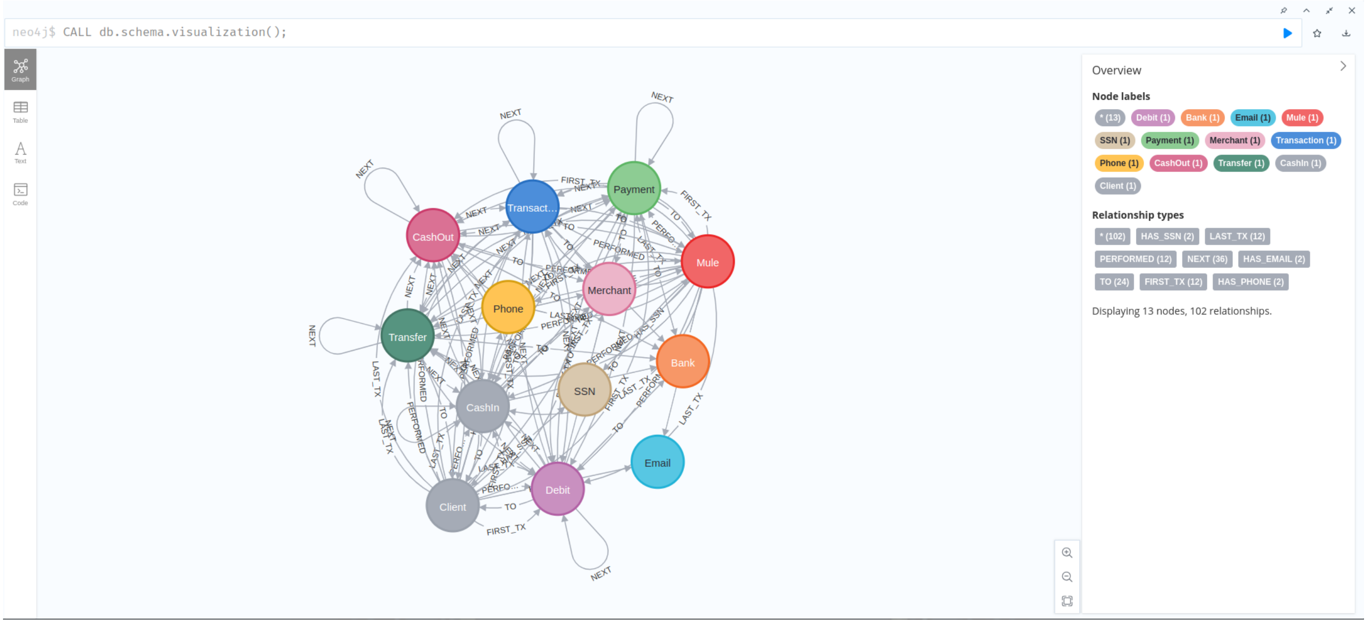
Fraud Detection

Schema Visualization

```
CALL db.schema.visualization();
```



```
neo4j$ CALL db.schema.visualization();
```

Overview

Node labels

* (13) Debit (1) Bank (1) Email (1) Mule (1) SSN (1) Payment (1) Merchant (1) Transaction (1) Phone (1) CashOut (1) Transfer (1) CashIn (1) Client (1)

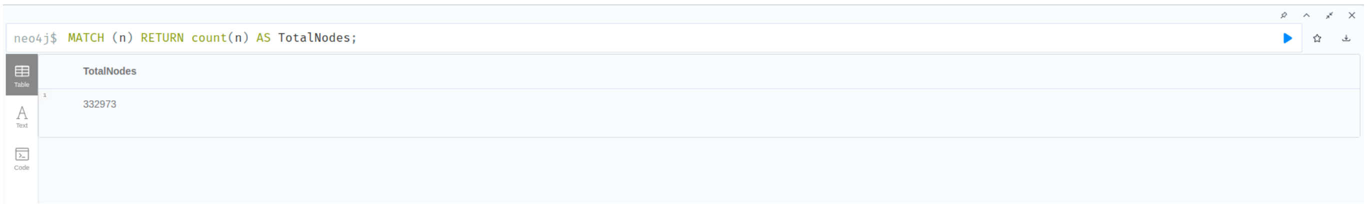
Relationship types

* (102) HAS_SSN (2) LAST_TX (12) PERFORMED (12) NEXT (36) HAS_EMAIL (2) TO (24) FIRST_TX (12) HAS_PHONE (2)

Displaying 13 nodes, 102 relationships.

Nodes Count (332973)

```
MATCH (n) RETURN count(n) AS TotalNodes;
```

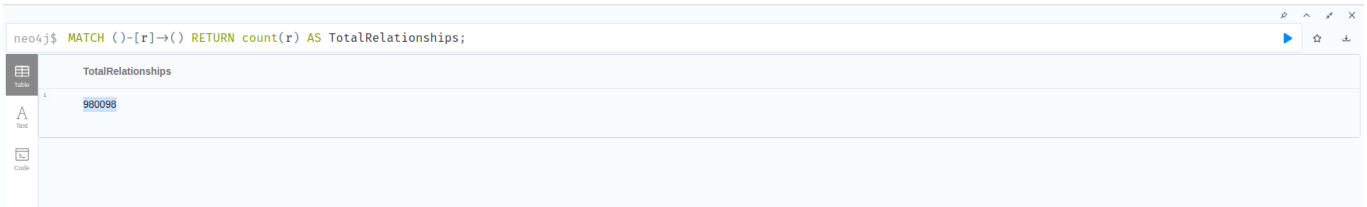


```
neo4j$ MATCH (n) RETURN count(n) AS TotalNodes;
```

	TotalNodes
1	332973

Relationships Count (980098)

```
MATCH ()-[r]->() RETURN count(r) AS TotalRelationships;
```

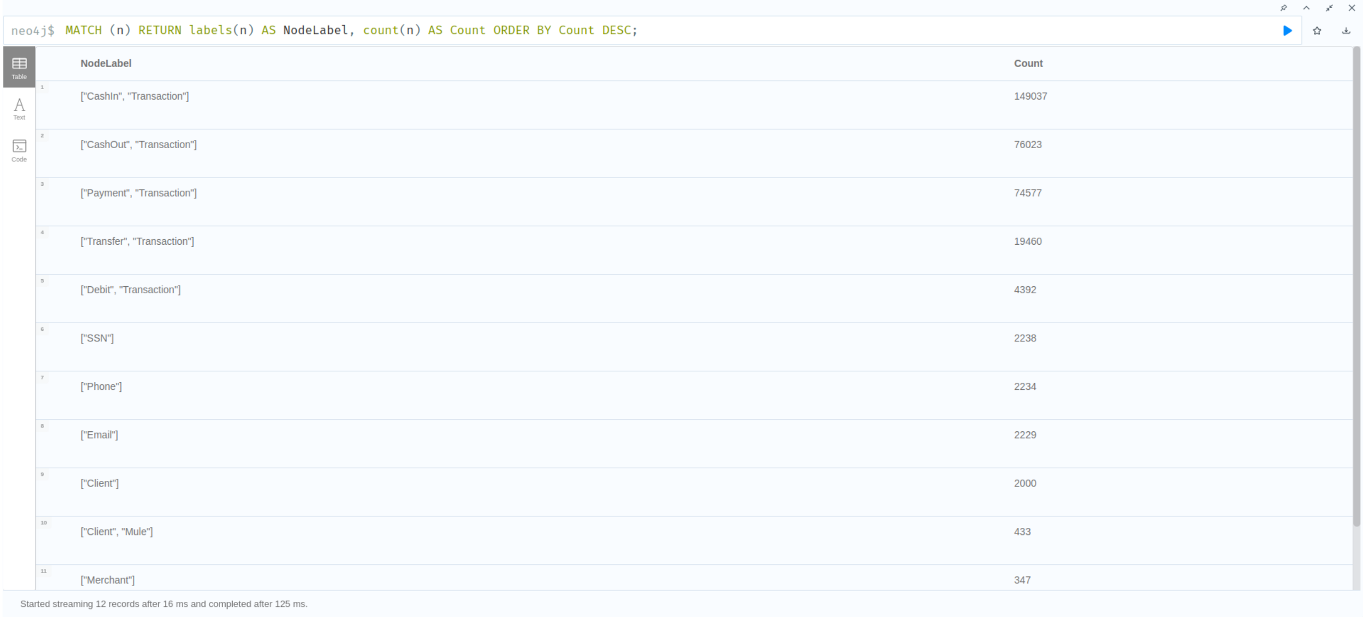


```
neo4j$ MATCH ()-[r]→() RETURN count(r) AS TotalRelationships;
```

	TotalRelationships
1	980098

Node Labels and their Count

```
MATCH (n) RETURN labels(n) AS NodeLabel, count(n) AS Count ORDER BY Count
DESC;
```



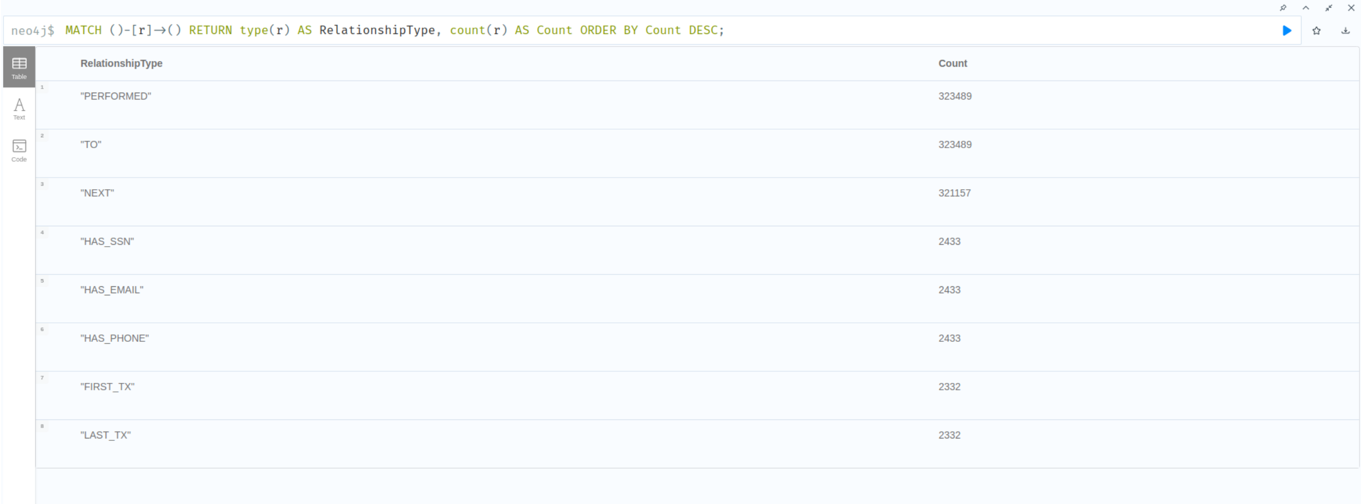
```
neo4j$ MATCH (n) RETURN labels(n) AS NodeLabel, count(n) AS Count ORDER BY Count DESC;
```

	NodeLabel	Count
1	["CashIn", "Transaction"]	149037
2	["CashOut", "Transaction"]	76023
3	["Payment", "Transaction"]	74577
4	["Transfer", "Transaction"]	19460
5	["Debit", "Transaction"]	4392
6	["SSN"]	2238
7	["Phone"]	2234
8	["Email"]	2229
9	["Client"]	2000
10	["Client", "Mule"]	433
11	["Merchant"]	347

Started streaming 12 records after 16 ms and completed after 125 ms.

Relationship Types and their Count

```
MATCH ()-[r]->() RETURN type(r) AS RelationshipType, count(r) AS Count
ORDER BY Count DESC;
```

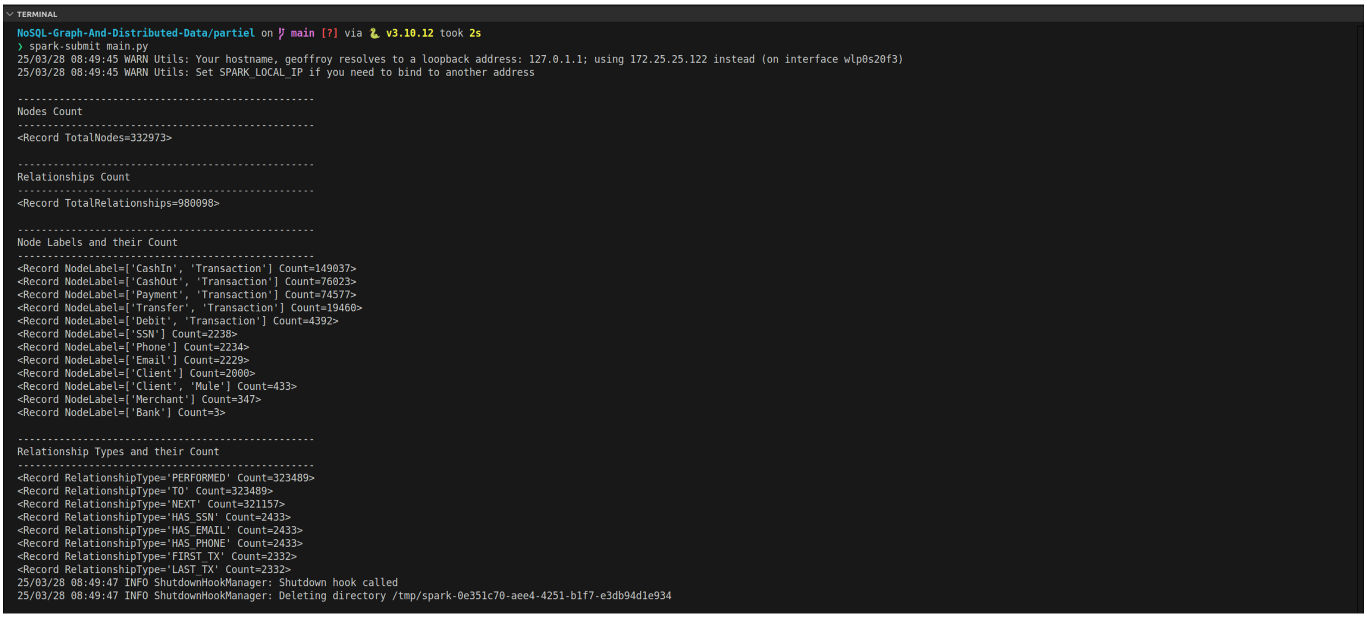


```
neo4j$ MATCH ()-[r]→() RETURN type(r) AS RelationshipType, count(r) AS Count ORDER BY Count DESC;
```

	RelationshipType	Count
1	"PERFORMED"	323489
2	"TO"	323489
3	"NEXT"	321157
4	"HAS_SSN"	2433
5	"HAS_EMAIL"	2433
6	"HAS_PHONE"	2433
7	"FIRST_TX"	2332
8	"LAST_TX"	2332

Running all commands using Python (whithout Spark)

```
spark-submit main.py
```

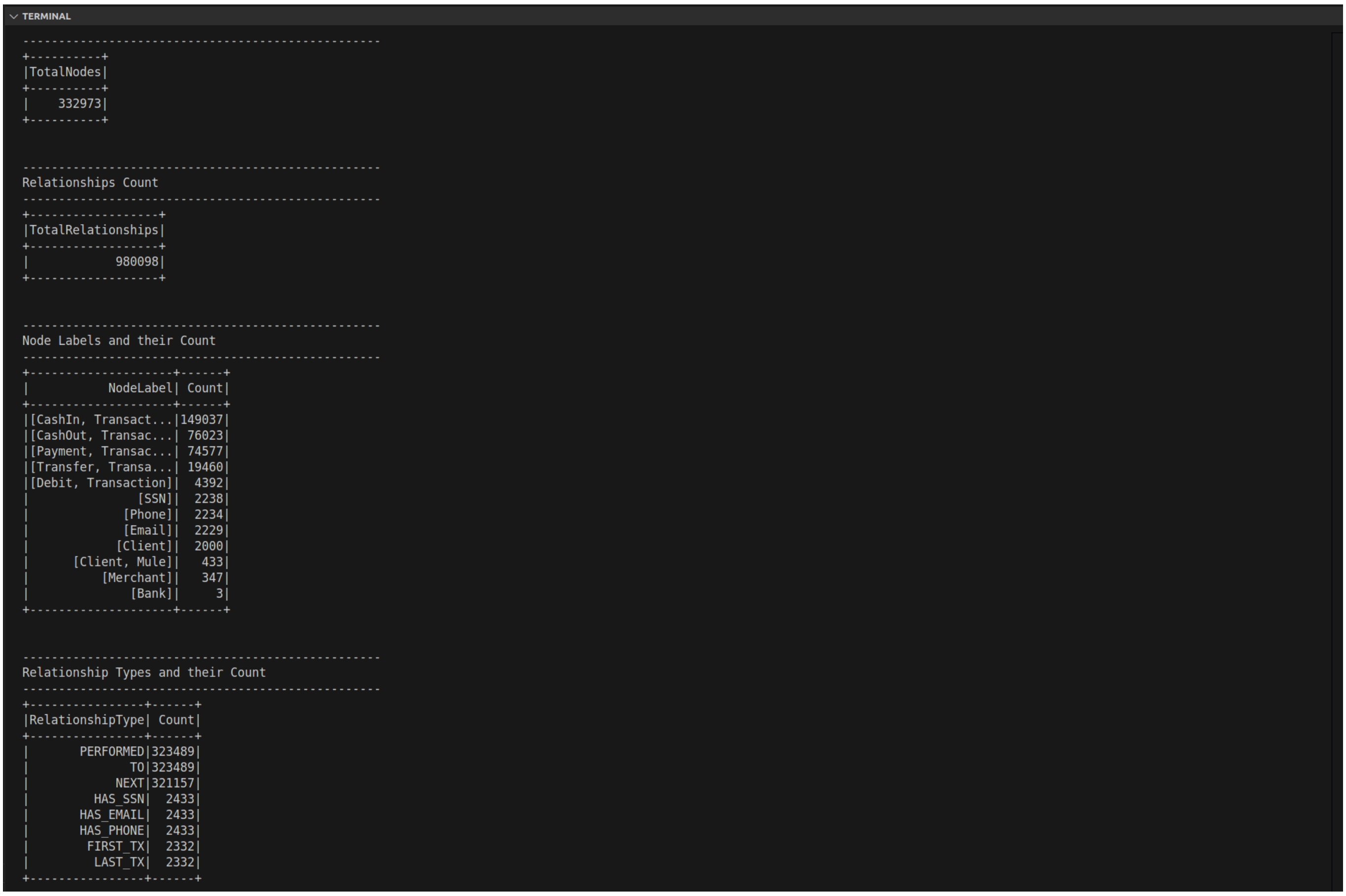


```
TERMINAL
NoSQL-Graph-And-Distributed-Data/partiel on  main [?] via 🐍 v3.10.12 took 2s
❯ spark-submit main.py
25/03/28 08:49:45 WARN Utils: Your hostname, geoffroy resolves to a loopback address: 127.0.1.1; using 172.25.25.122 instead (on interface wlp0s20f3)
25/03/28 08:49:45 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address

--------------------------------------------------
Nodes Count
--------------------------------------------------
<Record TotalNodes=332973>

--------------------------------------------------
Relationships Count
--------------------------------------------------
<Record TotalRelationships=980098>

--------------------------------------------------
Node Labels and their Count
--------------------------------------------------
<Record NodeLabel=['CashIn', 'Transaction'] Count=149037>
<Record NodeLabel=['CashOut', 'Transaction'] Count=76023>
<Record NodeLabel=['Payment', 'Transaction'] Count=74577>
<Record NodeLabel=['Transfer', 'Transaction'] Count=19460>
<Record NodeLabel=['Debit', 'Transaction'] Count=4392>
<Record NodeLabel=['SSN'] Count=2238>
<Record NodeLabel=['Phone'] Count=2234>
<Record NodeLabel=['Email'] Count=2229>
<Record NodeLabel=['Client'] Count=2000>
<Record NodeLabel=['Client', 'Mule'] Count=433>
<Record NodeLabel=['Merchant'] Count=347>
<Record NodeLabel=['Bank'] Count=3>

--------------------------------------------------
Relationship Types and their Count
--------------------------------------------------
<Record RelationshipType='PERFORMED' Count=323489>
<Record RelationshipType='TO' Count=323489>
<Record RelationshipType='NEXT' Count=321157>
<Record RelationshipType='HAS_SSN' Count=2433>
<Record RelationshipType='HAS_EMAIL' Count=2433>
<Record RelationshipType='HAS_PHONE' Count=2433>
<Record RelationshipType='FIRST_TX' Count=2332>
<Record RelationshipType='LAST_TX' Count=2332>
25/03/28 08:49:47 INFO ShutdownHookManager: Shutdown hook called
25/03/28 08:49:47 INFO ShutdownHookManager: Deleting directory /tmp/spark-0e351c70-aee4-4251-b1f7-e3db94d1e934
```

Running all commands using Python (with Spark)

> [!NOTE] The neo4j-connector-apache-spark_2.12-5.3.5_for_spark_3.jar is required to run the spark job.

```
spark-submit --jars neo4j-connector-apache-spark_2.12-5.3.5_for_spark_3.jar
main-spark.py
```



```
TERMINAL
--------------------------------------------------
+----------+
|TotalNodes|
+----------+
|    332973|
+----------+


--------------------------------------------------
Relationships Count
--------------------------------------------------
+------------------+
|TotalRelationships|
+------------------+
|            980098|
+------------------+


--------------------------------------------------
Node Labels and their Count
--------------------------------------------------
+--------------------+------+
|           NodeLabel| Count|
+--------------------+------+
|[CashIn, Transact...|149037|
|[CashOut, Transac...| 76023|
|[Payment, Transac...| 74577|
|[Transfer, Transa...| 19460|
|[Debit, Transaction]|  4392|
|               [SSN]|  2238|
|             [Phone]|  2234|
|             [Email]|  2229|
|            [Client]|  2000|
|      [Client, Mule]|   433|
|          [Merchant]|   347|
|              [Bank]|     3|
+--------------------+------+


--------------------------------------------------
Relationship Types and their Count
--------------------------------------------------
+----------------+------+
|RelationshipType| Count|
+----------------+------+
|       PERFORMED|323489|
|              TO|323489|
|            NEXT|321157|
|         HAS_SSN|  2433|
|       HAS_EMAIL|  2433|
|       HAS_PHONE|  2433|
|        FIRST_TX|  2332|
|         LAST_TX|  2332|
+----------------+------+
```

Running using Kubernetes (k3d)

Setting up the Kubernetes Cluster

1. Create a Kubernetes Cluster with k3d

```
k3d cluster create my-spark-cluster --servers 1 --agents 2
```

- Creates a Kubernetes cluster named `my-spark-cluster` with `1 server` node and `2 agent` nodes.

2. Grant the necessary permissions to the Kubernetes Cluster

```
kubectl create clusterrolebinding spark-role --clusterrole=edit --
serviceaccount=default:default --namespace=default
```

3. Install the Spark Operator with Helm

```
helm repo add spark-operator https://kubeflow.github.io/spark-operator
helm repo update
helm install spark-operator spark-operator/spark-operator \
    --namespace spark-operator \
    --create-namespace
```

Deploying the Spark Application

1. Build the Docker Image

```
docker build -t spark:partiel .
```

2. Import the Docker Image to Kubernetes

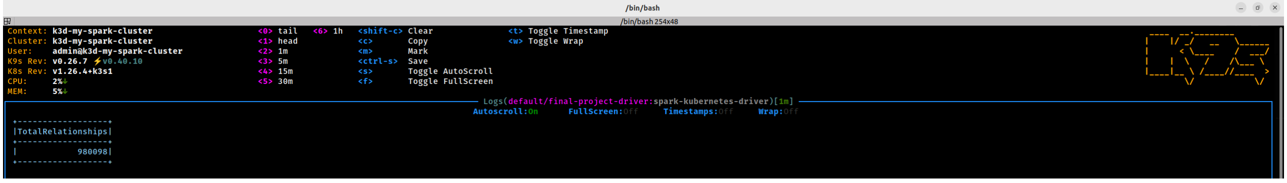
```
k3d image import -c my-spark-cluster spark:partiel
```

3. Deploy the application

```
kubectl apply -f ./kube/spark.yaml
```

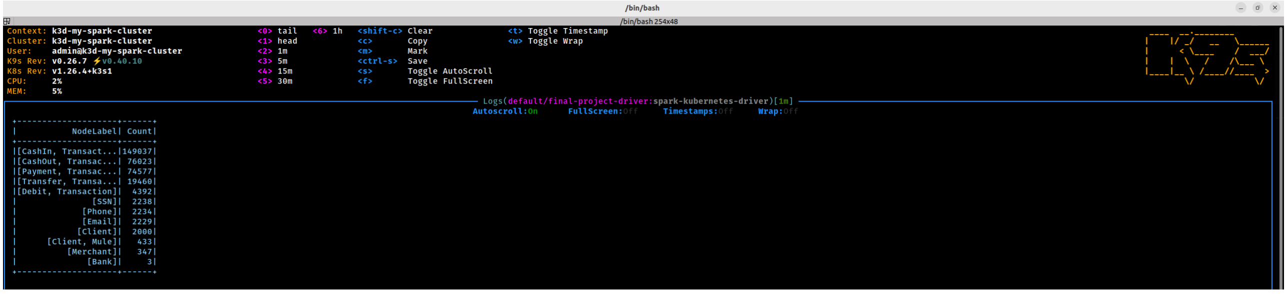
Kubernetes Output

1. Relationship



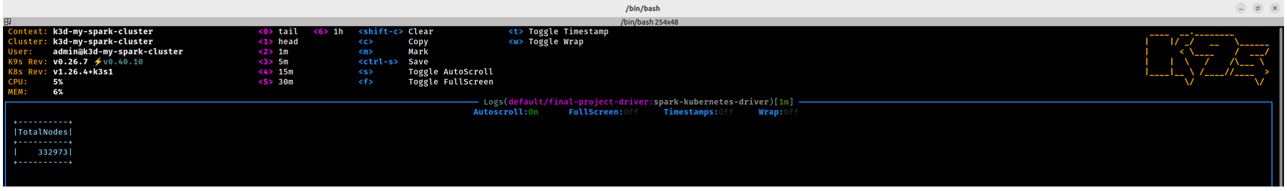
```
Context: k3d-my-spark-cluster                   <0> tail   <6> 1h   <shift-c> Clear            <t> Toggle Timestamp
Cluster: k3d-my-spark-cluster                   <1> head            <c>       Copy             <w> Toggle Wrap
User:    admin@k3d-my-spark-cluster             <2> 1m              <m>       Mark
K9s Rev: v0.26.7 ⚡v0.40.10                     <3> 5m              <ctrl-s>  Save
K8s Rev: v1.26.4+k3s1                           <4> 15m             <s>       Toggle AutoScroll
CPU:     2%↓                                    <5> 30m             <f>       Toggle FullScreen
MEM:     5%↓
                          Logs(default/final-project-driver:spark-kubernetes-driver)[1m]
                       Autoscroll:On      FullScreen:Off      Timestamps:Off      Wrap:Off

+------------------+
|TotalRelationships|
+------------------+
|            980098|
+------------------+
```

2. Node Labels



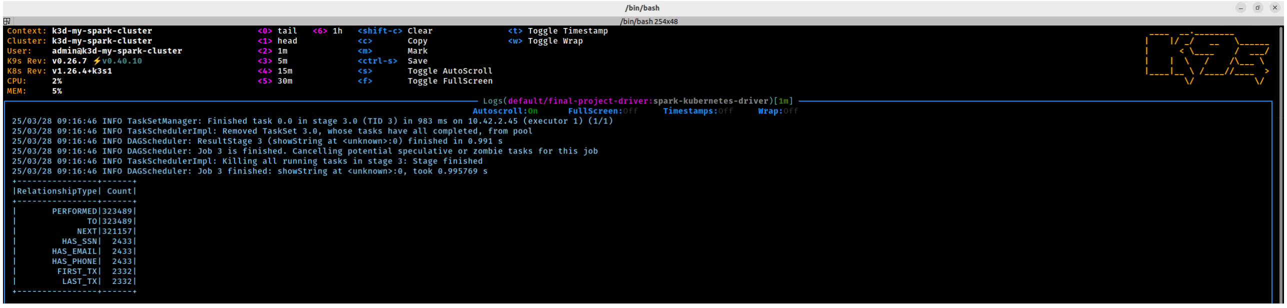
```
Context: k3d-my-spark-cluster                   <0> tail   <6> 1h   <shift-c> Clear            <t> Toggle Timestamp
Cluster: k3d-my-spark-cluster                   <1> head            <c>       Copy             <w> Toggle Wrap
User:    admin@k3d-my-spark-cluster             <2> 1m              <m>       Mark
K9s Rev: v0.26.7 ⚡v0.40.10                     <3> 5m              <ctrl-s>  Save
K8s Rev: v1.26.4+k3s1                           <4> 15m             <s>       Toggle AutoScroll
CPU:     2%                                     <5> 30m             <f>       Toggle FullScreen
MEM:     5%
                          Logs(default/final-project-driver:spark-kubernetes-driver)[1m]
                       Autoscroll:On      FullScreen:Off      Timestamps:Off      Wrap:Off

+--------------------+------+
|           NodeLabel| Count|
+--------------------+------+
|[CashIn, Transact...|149037|
|[CashOut, Transac...| 76023|
|[Payment, Transac...| 74577|
|[Transfer, Transa...| 19460|
|[Debit, Transaction]|  4392|
|               [SSN]|  2238|
|             [Phone]|  2234|
|             [Email]|  2229|
|            [Client]|  2000|
|      [Client, Mule]|   433|
|          [Merchant]|   347|
|              [Bank]|     3|
+--------------------+------+
```

3. Node Count



```
Context: k3d-my-spark-cluster                   <0> tail   <6> 1h   <shift-c> Clear            <t> Toggle Timestamp
Cluster: k3d-my-spark-cluster                   <1> head            <c>       Copy             <w> Toggle Wrap
User:    admin@k3d-my-spark-cluster             <2> 1m              <m>       Mark
K9s Rev: v0.26.7 ⚡v0.40.10                     <3> 5m              <ctrl-s>  Save
K8s Rev: v1.26.4+k3s1                           <4> 15m             <s>       Toggle AutoScroll
CPU:     5%                                     <5> 30m             <f>       Toggle FullScreen
MEM:     6%
                          Logs(default/final-project-driver:spark-kubernetes-driver)[1m]
                       Autoscroll:On      FullScreen:Off      Timestamps:Off      Wrap:Off

+----------+
|TotalNodes|
+----------+
|    332973|
+----------+
```

4. Relationship Types

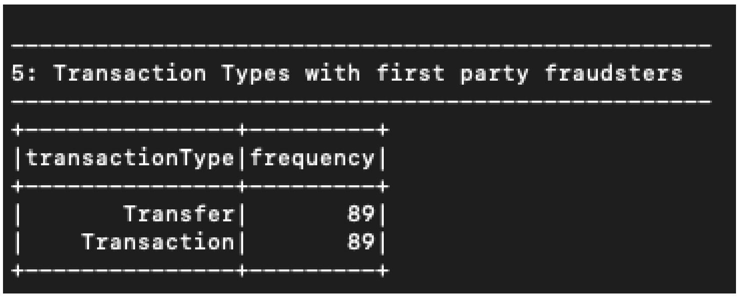


```
Context: k3d-my-spark-cluster                   <0> tail   <6> 1h   <shift-c> Clear            <t> Toggle Timestamp
Cluster: k3d-my-spark-cluster                   <1> head            <c>       Copy             <w> Toggle Wrap
User:    admin@k3d-my-spark-cluster             <2> 1m              <m>       Mark
K9s Rev: v0.26.7 ⚡v0.40.10                     <3> 5m              <ctrl-s>  Save
K8s Rev: v1.26.4+k3s1                           <4> 15m             <s>       Toggle AutoScroll
CPU:     2%                                     <5> 30m             <f>       Toggle FullScreen
MEM:     5%
                          Logs(default/final-project-driver:spark-kubernetes-driver)[1m]
                       Autoscroll:On      FullScreen:Off      Timestamps:Off      Wrap:Off
25/03/28 09:16:46 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 983 ms on 10.42.2.45 (executor 1) (1/1)
25/03/28 09:16:46 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
25/03/28 09:16:46 INFO DAGScheduler: ResultStage 3 (showString at <unknown>:0) finished in 0.991 s
25/03/28 09:16:46 INFO DAGScheduler: Job 3 is finished. Cancelling potential speculative or zombie tasks for this job
25/03/28 09:16:46 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
25/03/28 09:16:46 INFO DAGScheduler: Job 3 finished: showString at <unknown>:0, took 0.995769 s
+----------------+------+
|RelationshipType| Count|
+----------------+------+
|       PERFORMED|323489|
|              TO|323489|
|            NEXT|321157|
|         HAS_SSN|  2433|
|       HAS_EMAIL|  2433|
|       HAS_PHONE|  2433|
|        FIRST_TX|  2332|
|         LAST_TX|  2332|
+----------------+------+
```

Exercices

1. Find out what types of transactions do these Clients perform with first party fraudsters?

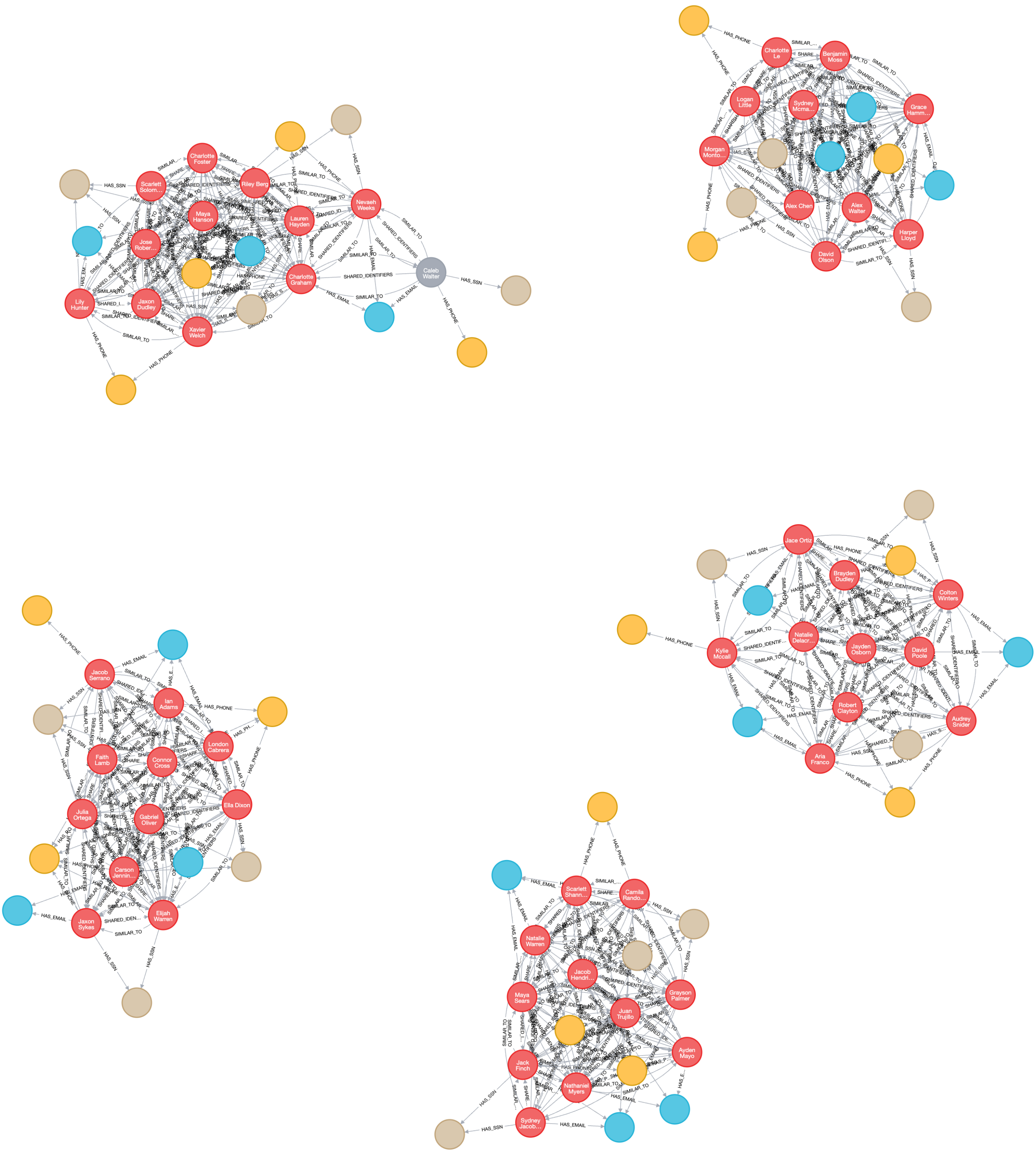
```
MATCH (:Client:FirstPartyFraudster)-[]-(txn:Transaction)-[]-(c:Client)
WHERE NOT c:FirstPartyFraudster
UNWIND labels(txn) AS transactionType
RETURN transactionType, count(*) AS freq;
```



```
--------------------------------------------------
5: Transaction Types with first party fraudsters
--------------------------------------------------
+---------------+---------+
|transactionType|frequency|
+---------------+---------+
|       Transfer|       89|
|    Transaction|       89|
+---------------+---------+
```

1. How many clusters of FraudRings with greater than 9 client nodes. (THIS IS A GRAPH VISUALIZATION OUTPUT TO BE ADDED IN YOUR REPORT) -- THIS IS RELATED TO NEO4J CYPHER TASK

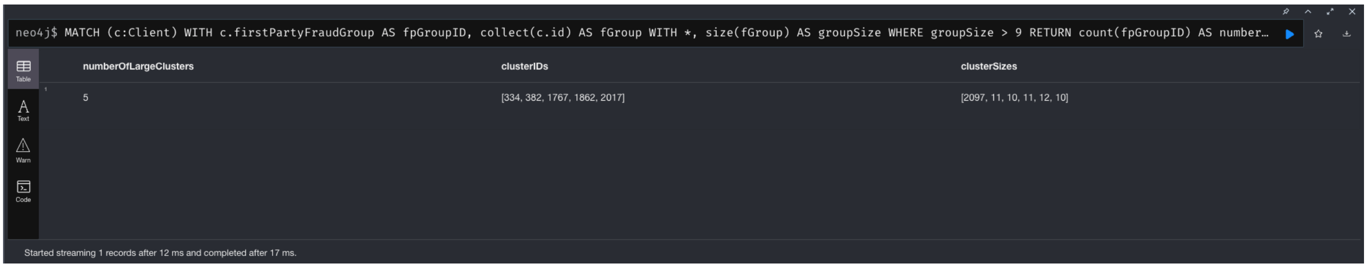
```
MATCH (c:Client)
WITH c.firstPartyFraudGroup AS fpGroupID, collect(c.id) AS fGroup
WITH *, size(fGroup) AS groupSize
WHERE groupSize > 9
WITH collect(fpGroupID) AS fraudRings
MATCH p=(c:Client)-[:HAS_SSN|HAS_EMAIL|HAS_PHONE]->()
WHERE c.firstPartyFraudGroup IN fraudRings
RETURN p
pour la 6
```



3. How many clusters of SecondPartyFraudsters with more than 10 client nodes. (THIS IS A GRAPH VISUALIZATION OUTPUT TO BE ADDED IN YOUR REPORT) -- THIS IS RELATED TO NEO4J CYPHER TASK

```
MATCH (c:Client)
WITH c.firstPartyFraudGroup AS fpGroupID, collect(c.id) AS fGroup
WITH *, size(fGroup) AS groupSize
WHERE groupSize > 10
WITH collect(fpGroupID) AS fraudRings
MATCH p=(c:Client)-[:HAS_SSN|HAS_EMAIL|HAS_PHONE]->()
```

```
WHERE c.firstPartyFraudGroup IN fraudRings
RETURN p;
```



```
neo4j$ MATCH (c:Client) WITH c.firstPartyFraudGroup AS fpGroupID, collect(c.id) AS fGroup WITH *, size(fGroup) AS groupSize WHERE groupSize > 9 RETURN count(fpGroupID) AS number…
```

	numberOfLargeClusters	clusterIDs	clusterSizes
1	5	[334, 382, 1767, 1862, 2017]	[2097, 11, 10, 11, 12, 10]

Started streaming 1 records after 12 ms and completed after 17 ms.