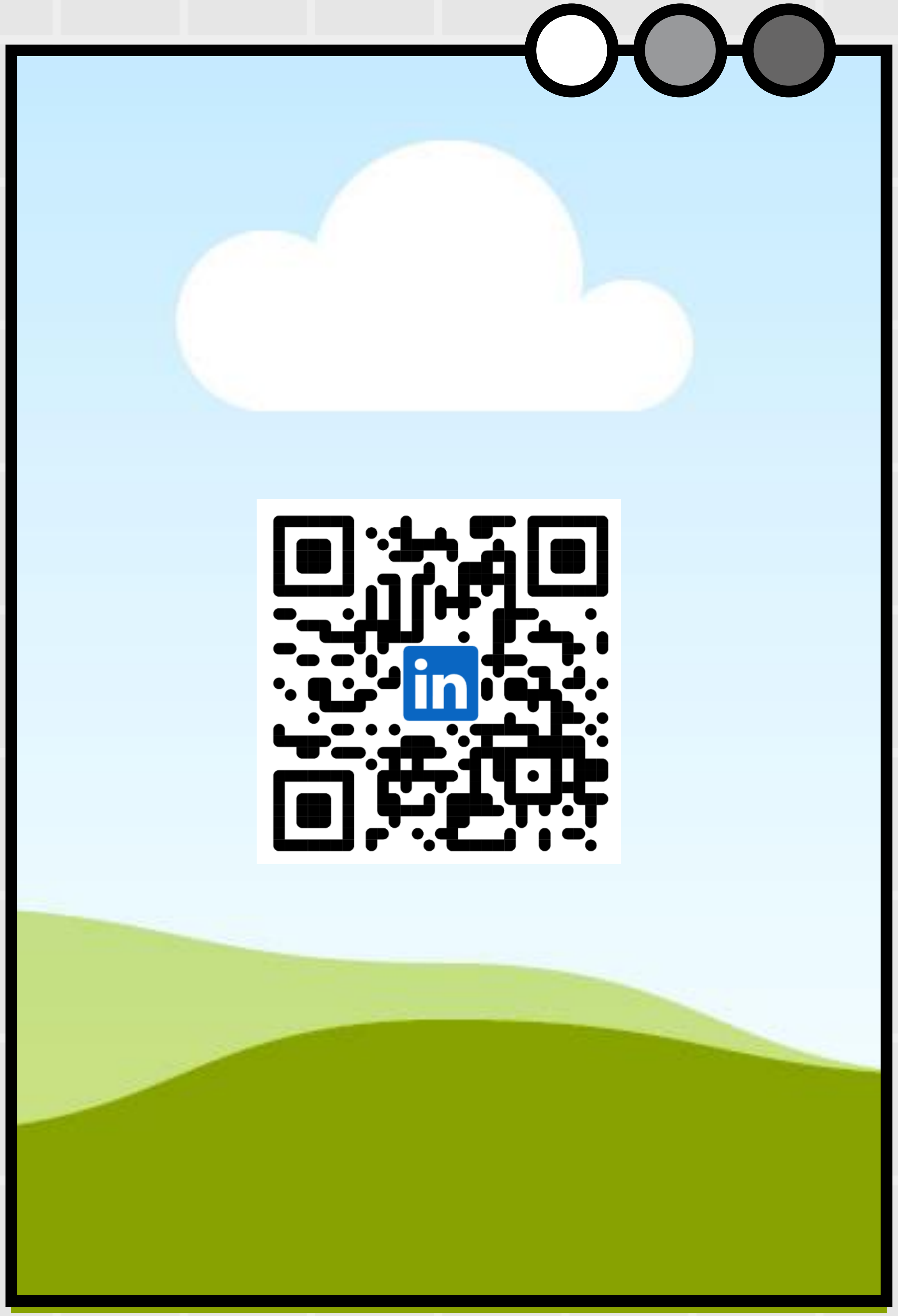
R CONFERENCE

ORGANIZED BY MALAYSIAN R USER GROUP (MYRUG)

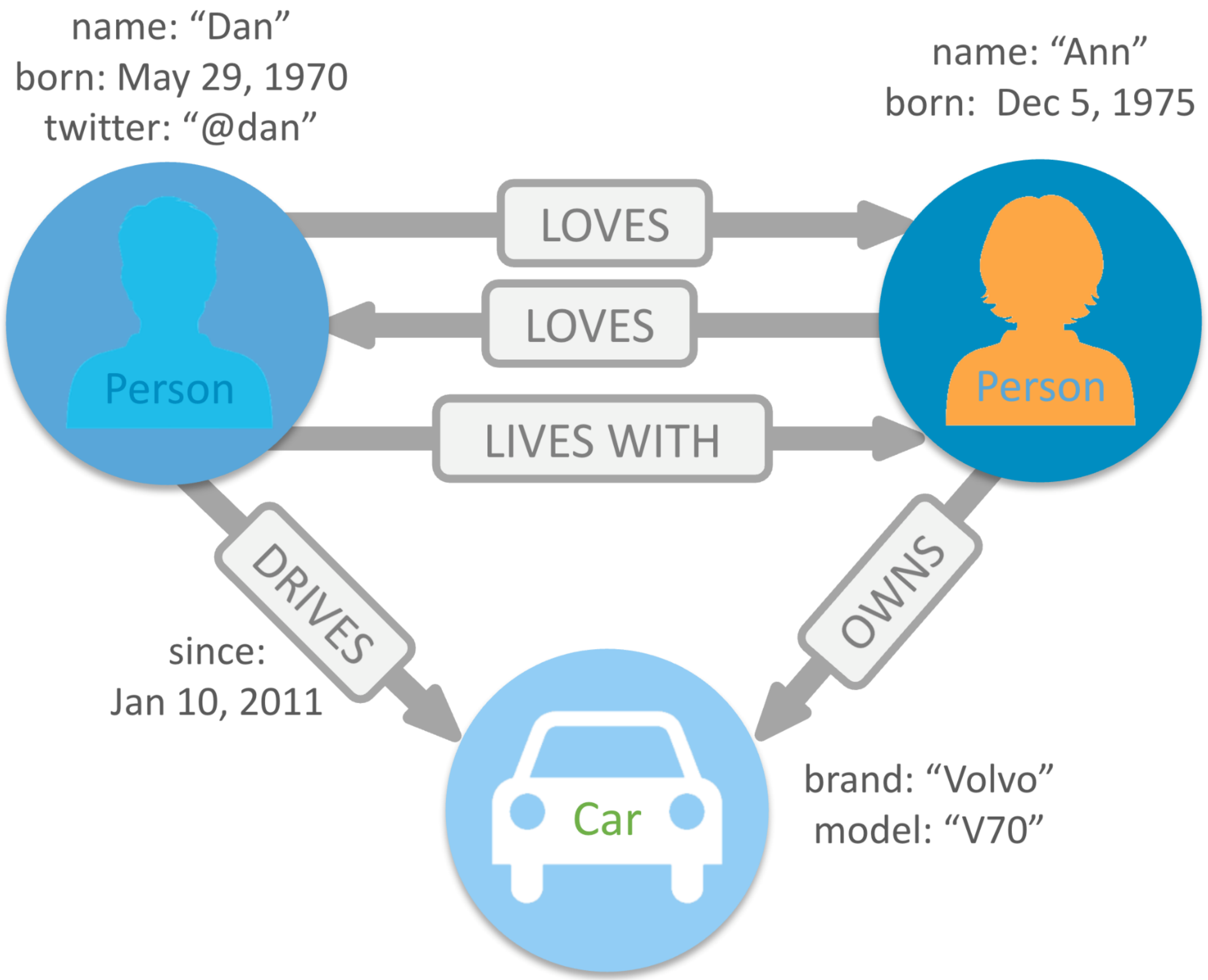
Dr Tan Yan Bin

- Data Engineer ; Data Science trainer
- Specializations :
  - graph databases (Neo4j, TigerGraph)
  - SIEM (Splunk)
  - dashboards and reporting
  - machine learning



Neo4j in R

Graph Fundamentals



Node

- an entity in the graph

Relationship

- connection between nodes

Node Label

- grouping of similar nodes, e.g. Person, Car

Properties

- description (key-value pairs) of a node or relationship, e.g. name, born, brand, model

Why graph database?

- Nodes and relationships instead of table
- Flexible schema
- Deal with connected data, i.e. the ‘connection’ itself brings meaning
- Traverse multiple hops
- Discover patterns or hidden connections

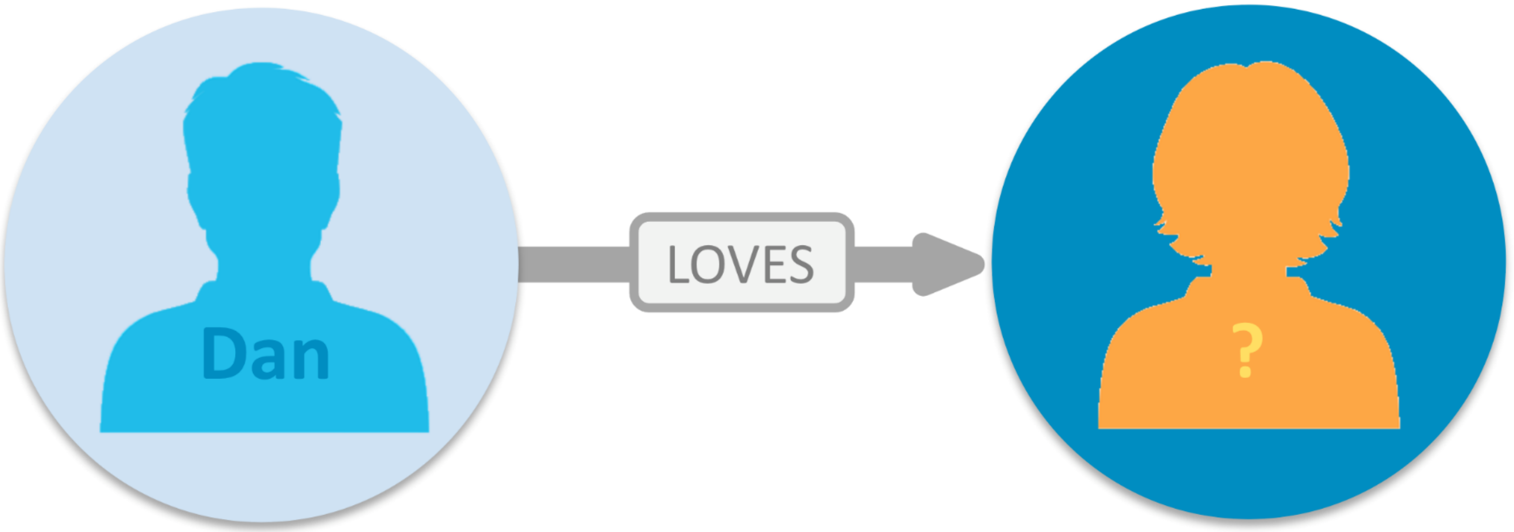
Typical use cases :

- social media network, fraud detection, recommendation system, supply chain, …

Conceptual comparison

RDBMS		Graph DB
Table	⟷	Node Label
Row	⟷	Node
Column	⟷	Properties
JOINs	⟷	Relationships

Neo4j Cypher



NODE Relationship NODE

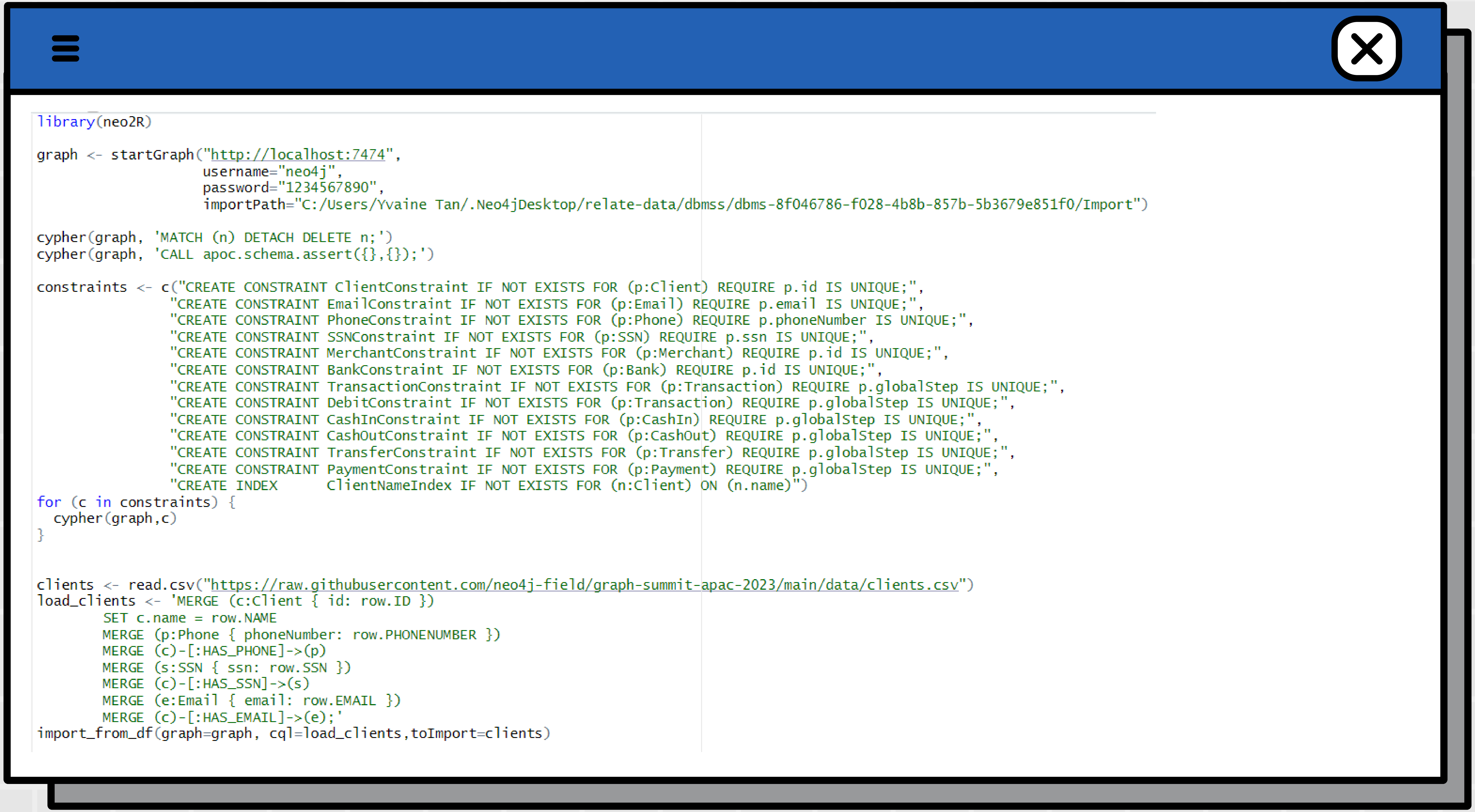
```
MATCH (:Person { name:"Dan"} ) -[:LOVES]-> ( whom ) RETURN whom
```

LABEL PROPERTY VARIABLE

Connecting to R

Available libraries :

- neo2R
- neo4jshell
- neo4r



```r
library(neo2R)

graph <- startGraph("http://localhost:7474",
                    username="neo4j",
                    password="1234567890",
                    importPath="C:/Users/Yvaine Tan/.Neo4jDesktop/relate-data/dbmss/dbms-8f046786-f028-4b8b-857b-5b3679e851f0/Import")

cypher(graph, 'MATCH (n) DETACH DELETE n;')
cypher(graph, 'CALL apoc.schema.assert({},{});')

constraints <- c("CREATE CONSTRAINT ClientConstraint IF NOT EXISTS FOR (p:Client) REQUIRE p.id IS UNIQUE;",
                "CREATE CONSTRAINT EmailConstraint IF NOT EXISTS FOR (p:Email) REQUIRE p.email IS UNIQUE;",
                "CREATE CONSTRAINT PhoneConstraint IF NOT EXISTS FOR (p:Phone) REQUIRE p.phoneNumber IS UNIQUE;",
                "CREATE CONSTRAINT SSNConstraint IF NOT EXISTS FOR (p:SSN) REQUIRE p.ssn IS UNIQUE;",
                "CREATE CONSTRAINT MerchantConstraint IF NOT EXISTS FOR (p:Merchant) REQUIRE p.id IS UNIQUE;",
                "CREATE CONSTRAINT BankConstraint IF NOT EXISTS FOR (p:Bank) REQUIRE p.id IS UNIQUE;",
                "CREATE CONSTRAINT TransactionConstraint IF NOT EXISTS FOR (p:Transaction) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE CONSTRAINT DebitConstraint IF NOT EXISTS FOR (p:Transaction) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE CONSTRAINT CashInConstraint IF NOT EXISTS FOR (p:CashIn) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE CONSTRAINT CashOutConstraint IF NOT EXISTS FOR (p:CashOut) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE CONSTRAINT TransferConstraint IF NOT EXISTS FOR (p:Transfer) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE CONSTRAINT PaymentConstraint IF NOT EXISTS FOR (p:Payment) REQUIRE p.globalStep IS UNIQUE;",
                "CREATE INDEX      ClientNameIndex IF NOT EXISTS FOR (n:Client) ON (n.name)")
for (c in constraints) {
  cypher(graph,c)
}


clients <- read.csv("https://raw.githubusercontent.com/neo4j-field/graph-summit-apac-2023/main/data/clients.csv")
load_clients <- 'MERGE (c:Client { id: row.ID })
        SET c.name = row.NAME
        MERGE (p:Phone { phoneNumber: row.PHONENUMBER })
        MERGE (c)-[:HAS_PHONE]->(p)
        MERGE (s:SSN { ssn: row.SSN })
        MERGE (c)-[:HAS_SSN]->(s)
        MERGE (e:Email { email: row.EMAIL })
        MERGE (c)-[:HAS_EMAIL]->(e);'
import_from_df(graph=graph, cql=load_clients,toImport=clients)
```

Graph Data Science

Neo4j GDS

. . . . use graph algorithms, i.e. set of instructions that traverse across the graph to analyse relationships and patterns in connected data.

Graph Algorithm Family

Pathfinding & Search

Centrality & Importance

Community Detection

Supervised Machine Learning

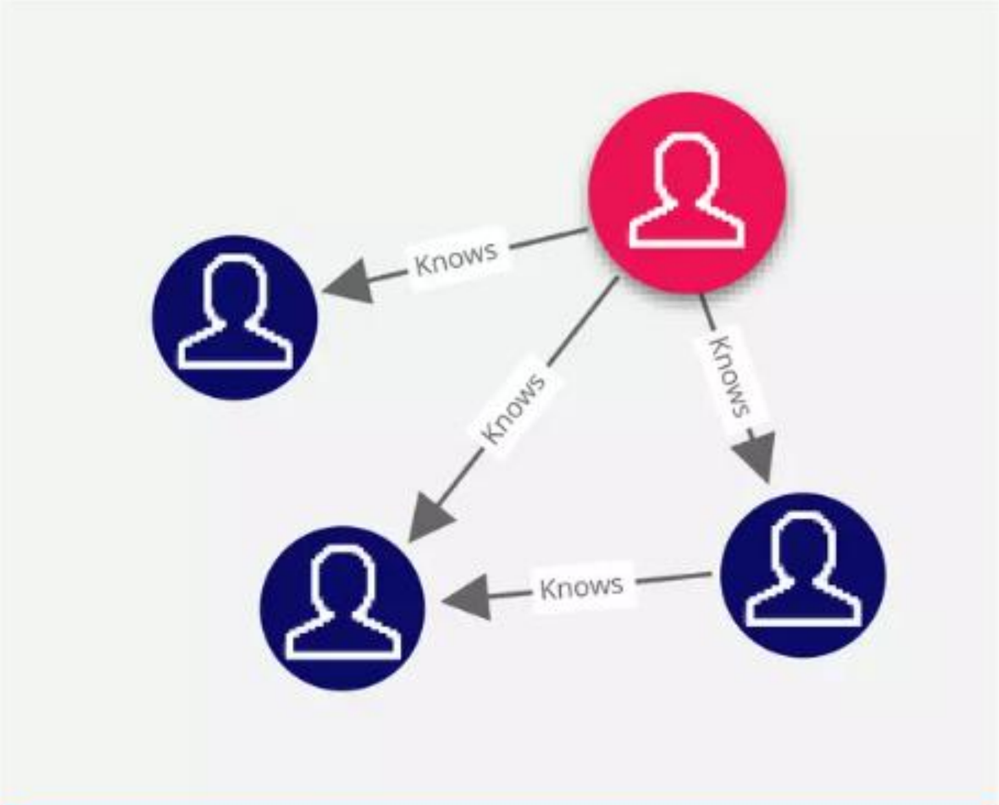
Heuristic Link Prediction

Similarity

Graph Embeddings

...and more

Graph Structure Improves Data Science Outcomes



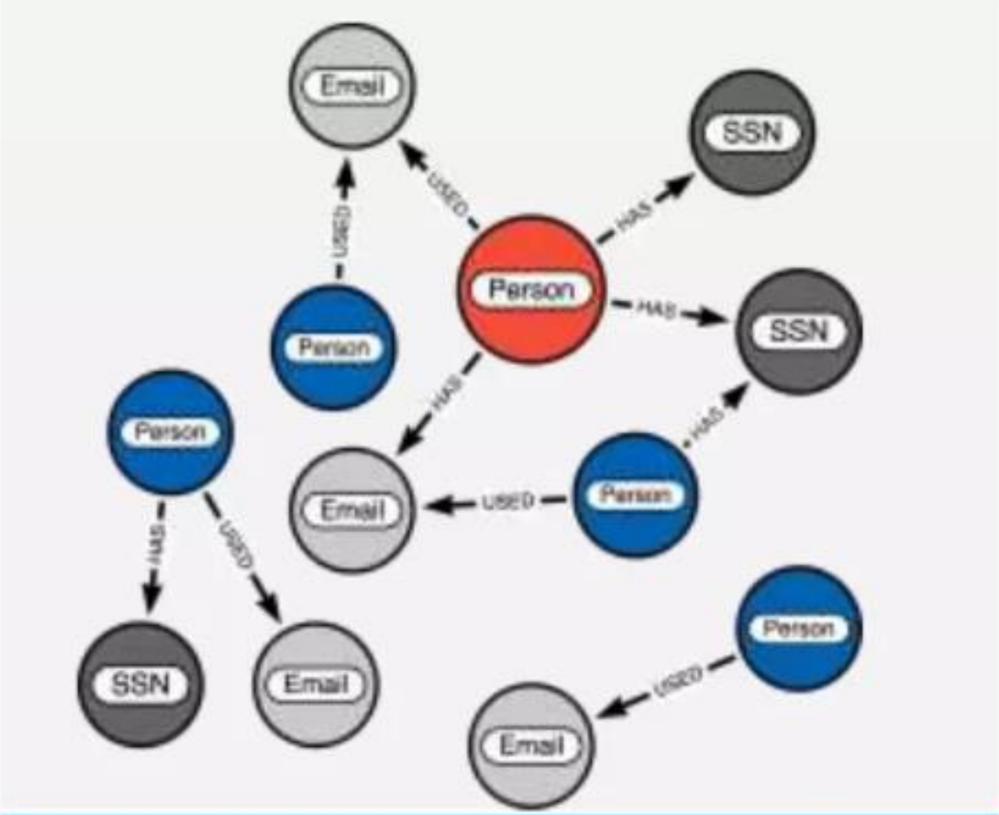
What's important?

Prioritization

Who has the most connections?

Who has the highest page rank?

Who is an influencer?

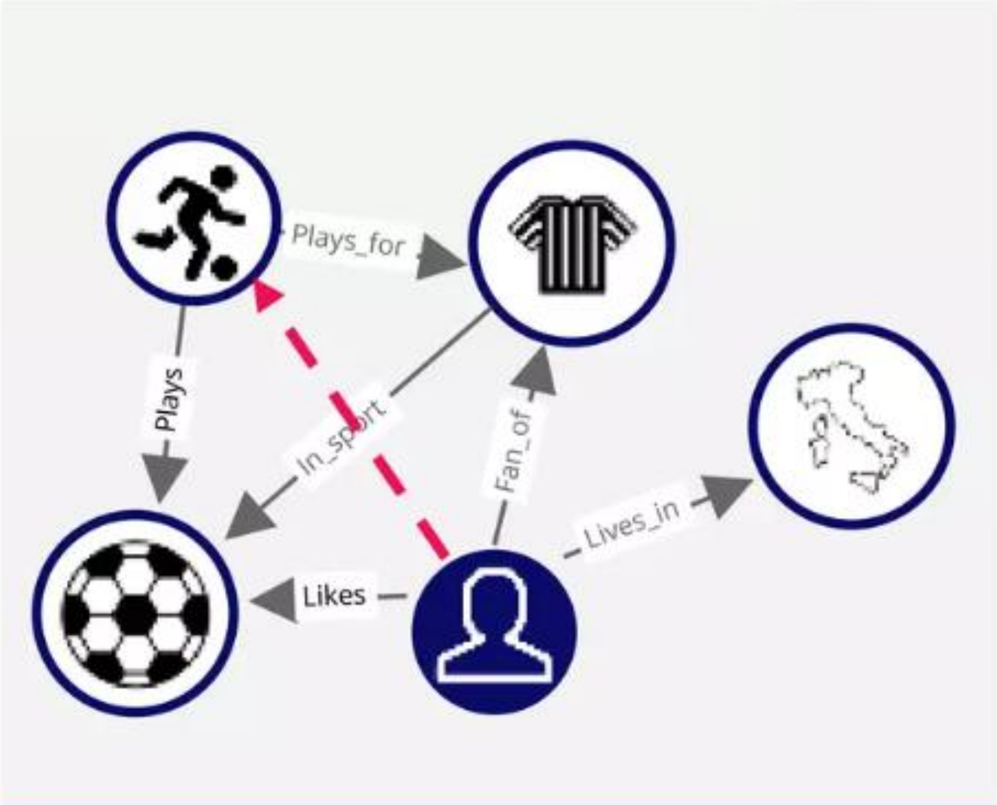


What's unusual?

Anomaly & Fraud Detection

Where is a community forming?
What are the group dynamics?

What's unusual about this data?



What's next?

Predictions

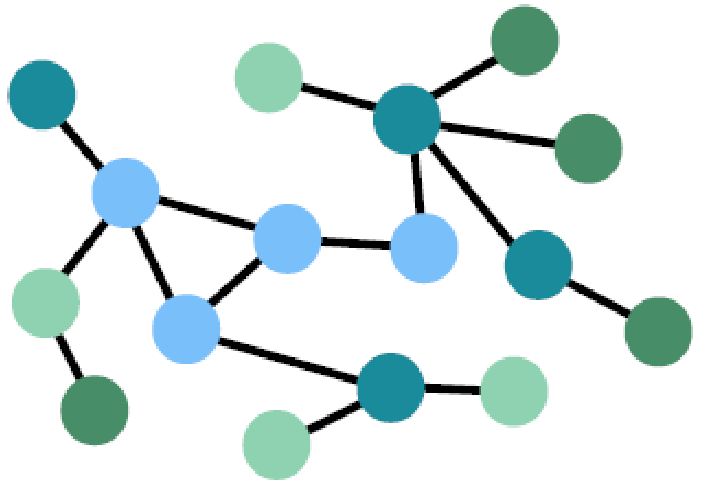
What's the most common path?

Who is in the same community?

What relationship will form?

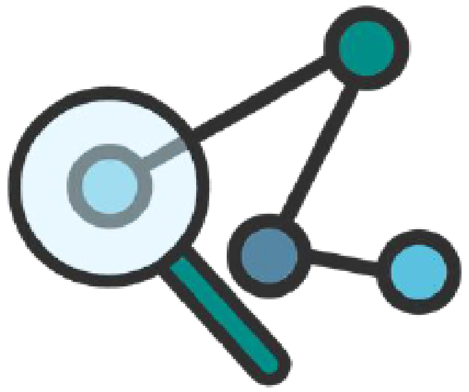
Graph and Data Science

Knowledge Graphs



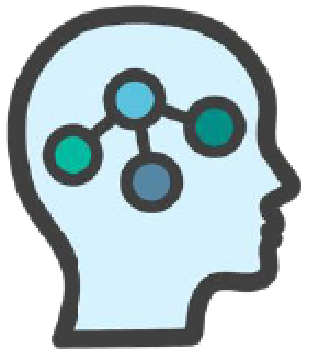
Find the patterns you're looking for in connected data

Graph Algorithms



Use unsupervised machine learning techniques to identify associations, anomalies, and trends.

Graph Native Machine Learning



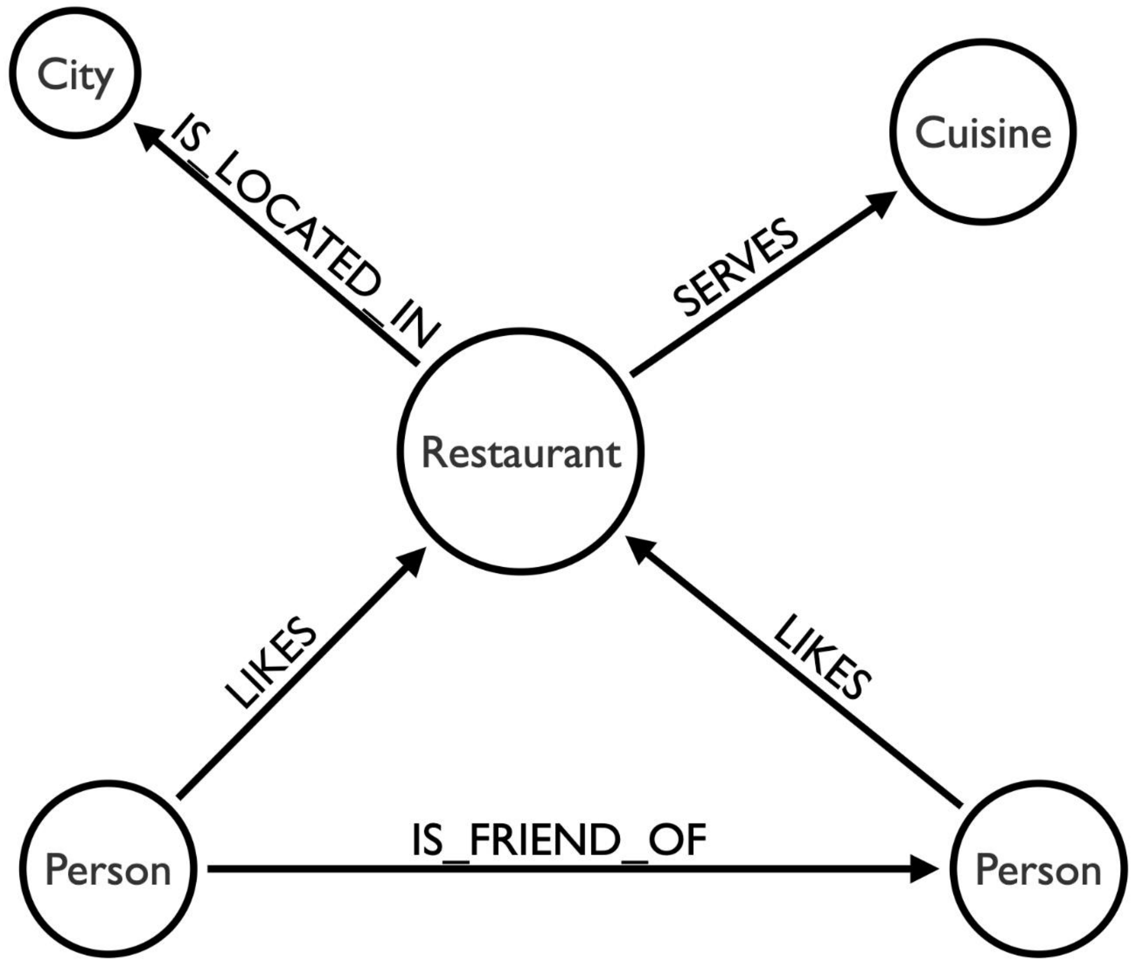
Use embeddings to learn the features in your graph that you don't even know are important yet.

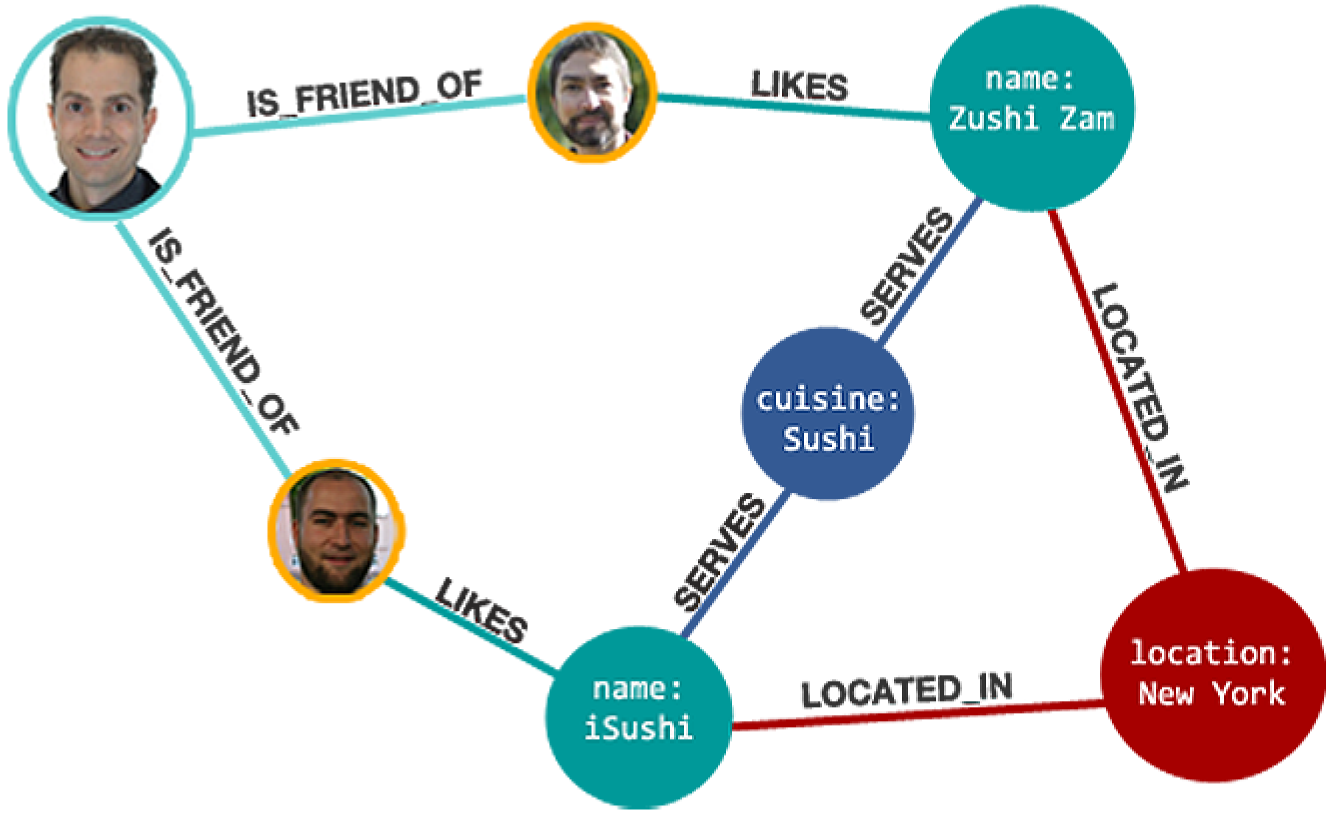
Train in-graph supervised ML models to predict links, labels, and missing data.

neo4j

Use case example :

Recommend Philip some sushi restaurants in New York that his friends like.





Algorithm :

1. Find Philip and his friends
2. Find restaurants that serve sushi in New York
3. Find restaurants that Philip's friends like

Read more

1. neo4r User Guide
2. 10 Things You Can Do with Cypher
3. Neo4j GraphGist – use cases and examples

Thank you !

Q & A

```
MERGE (:R)-[:LOVES]->(: Neo4j)
```