

GraphAware

world's #1 Neo4j consultancy

 graphaware.com

 [@graph_aware](https://twitter.com/graph_aware)



GraphAware®

Day 1

Graphical Data Modelling

Session 1

What is a Graph? What is Neo4j?

Session 2

Introduction to Data Modelling

Session 3

Introduction to Cypher

Session 4

Hands-On: Let's Play with Cypher



What is a Graph?



GraphAware®

Welcomes and Introductions



GraphAware®

GraphAware

- Founded in **2013**
- HQ in **London, UK**
- Growing all the time!
- Clients **all over the world**
- Focus on **graphs** and **Neo4j** in particular



GraphAware®

David Simons

- Senior Consultant at GraphAware
- Based in London, UK
- Neo4j Certified Professional
- Author of the Neo4j Expire module
- Experience with Java, JavaScript and .NET
- Worked on projects with BBC and News International



GraphAware®

As a Group...

- What do you know about graphs?
- What do you know about NoSQL?
- What do you know about Neo4j?
- What do you want from this course?



GraphAware®



Why are we looking to **NoSQL Solutions**



GraphAware®



Why are we moving away from SQL now?



GraphAware®

My thoughts...



Semi-Structured
Data



More Connected
Data



Bigger Data

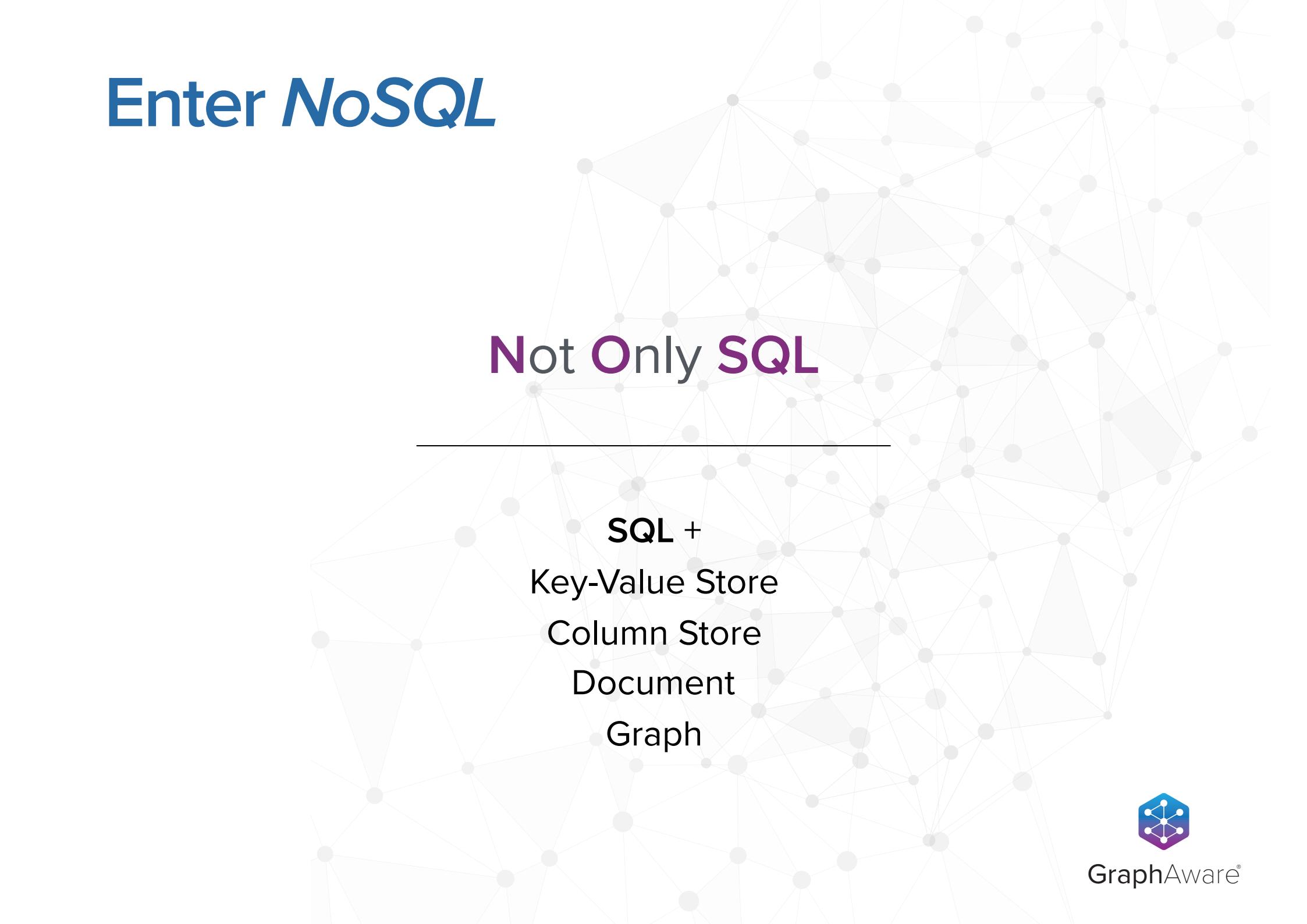


Microservices
unlocking more
niche databases



GraphAware®

Enter *NoSQL*



Not Only SQL

SQL +
Key-Value Store
Column Store
Document
Graph



GraphAware®

A Brief History of **Graphs**



GraphAware®

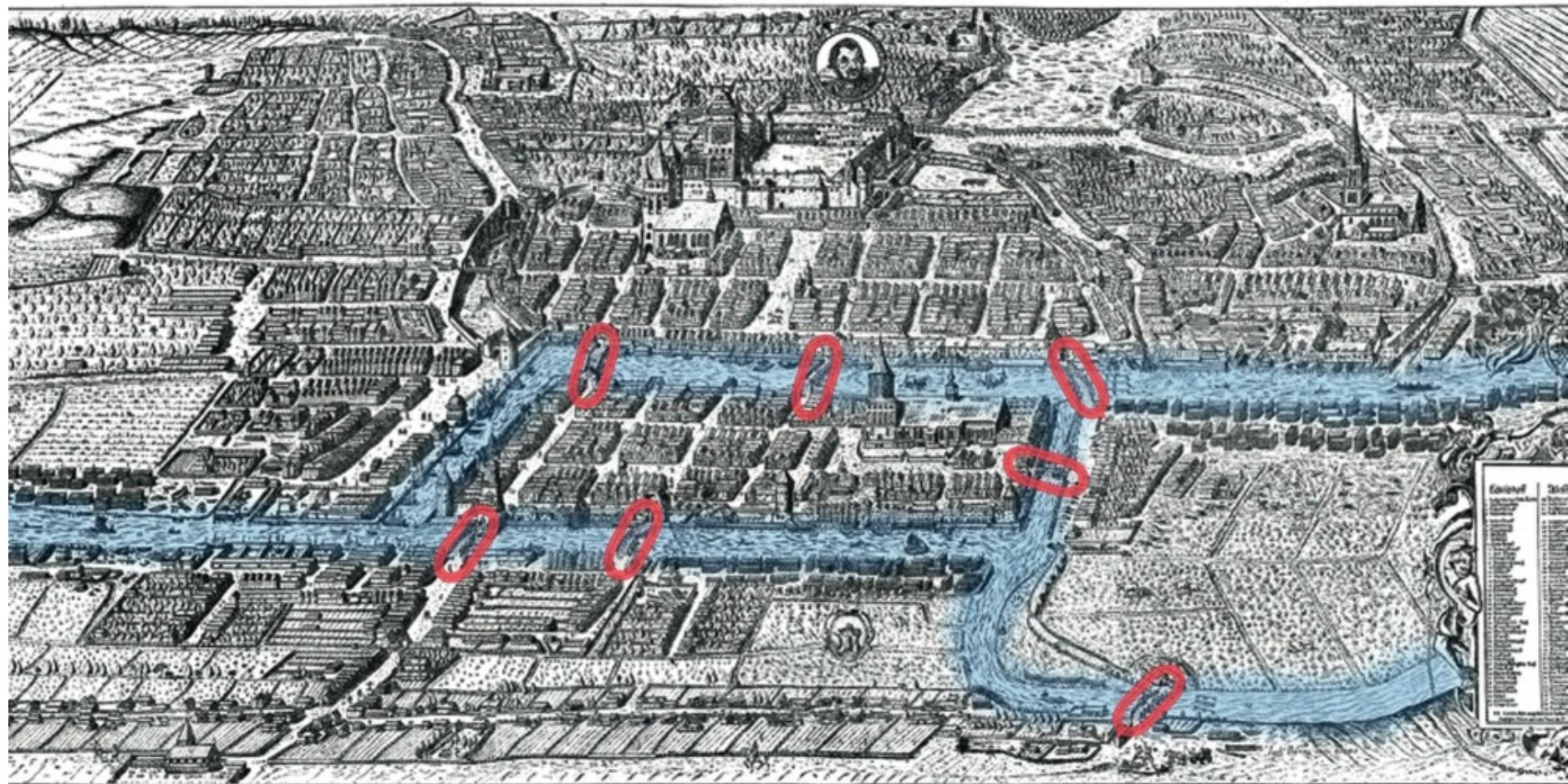
Our Graph Hero....



GraphAware®

The Seven Bridges of Koenigsberg

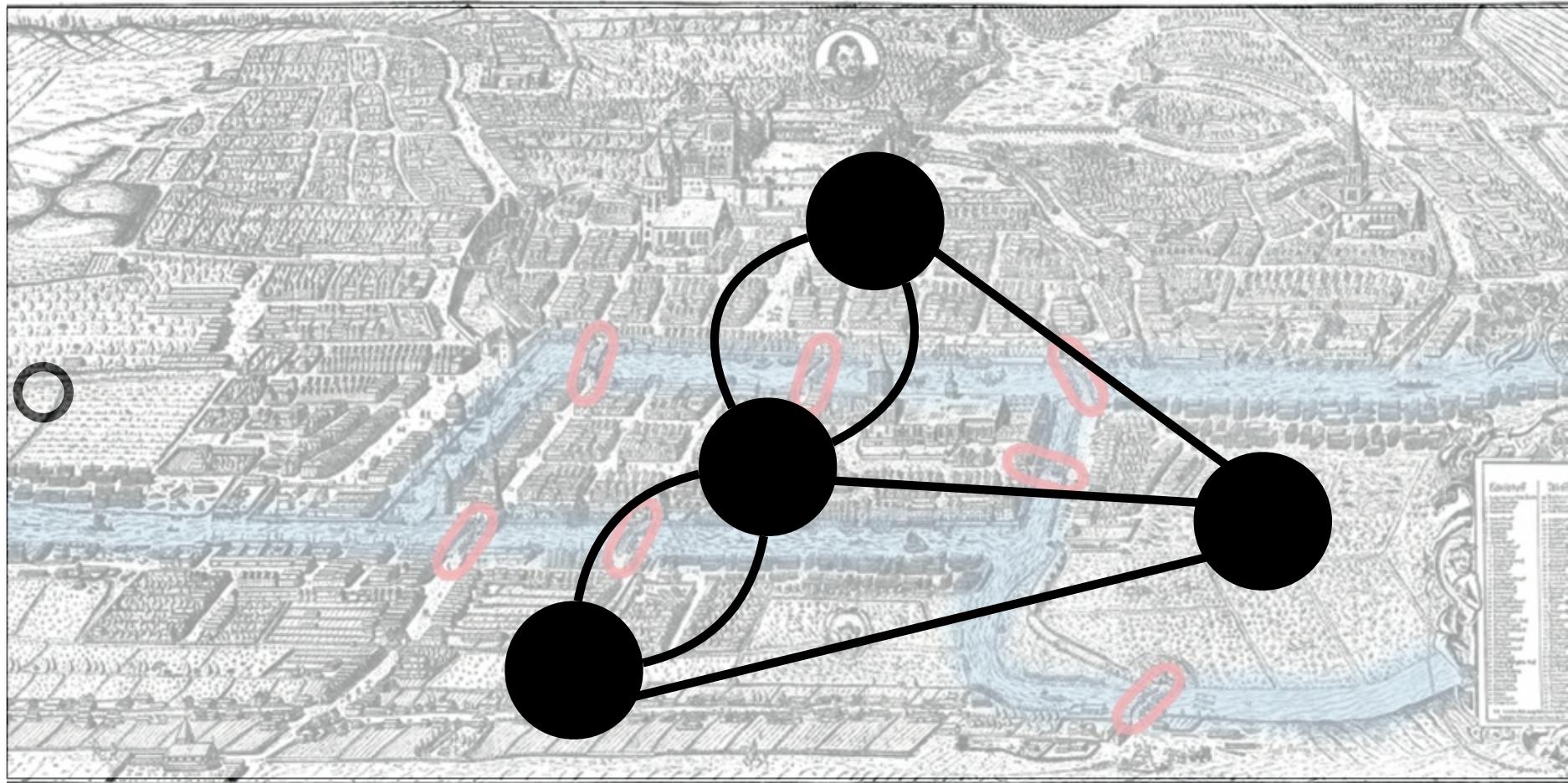
Gedenkblatt zur sechshundert jährigen Jubelfeier der Königlichen Haupt und Residenz-Stadt Königsberg in Preußen.



GraphAware®

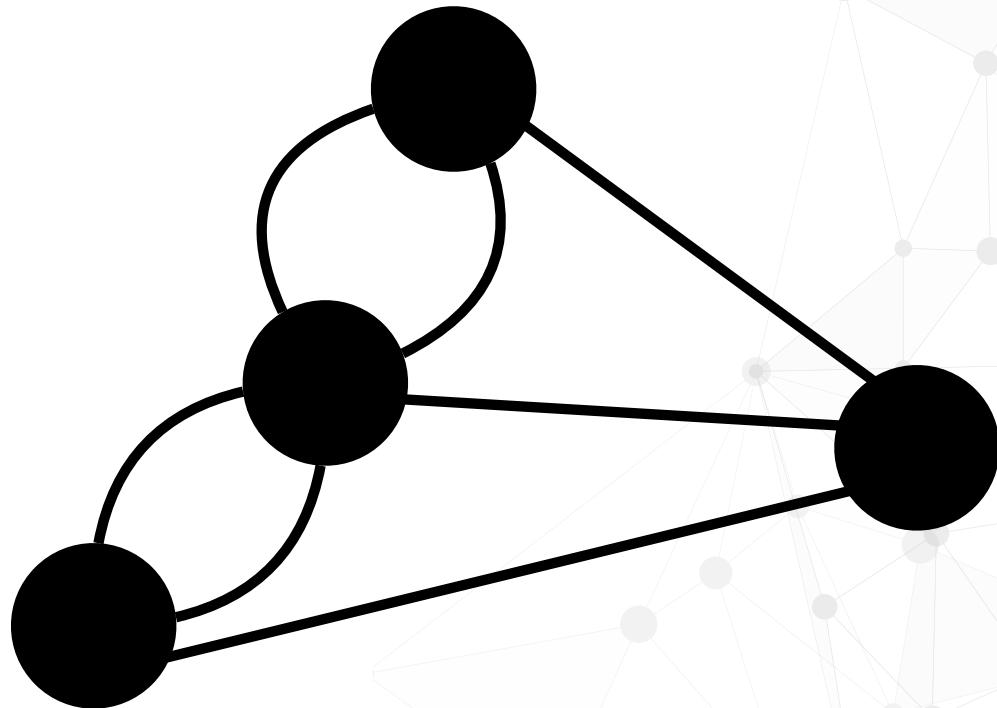
The Seven Bridges of Koenigsberg

Gedenkblatt zur sechshundert jährigen Jubelfeier der Königlichen Haupt und Residenz-Stadt Königsberg in Preußen.



GraphAware®

The Seven Bridges of Koenigsberg



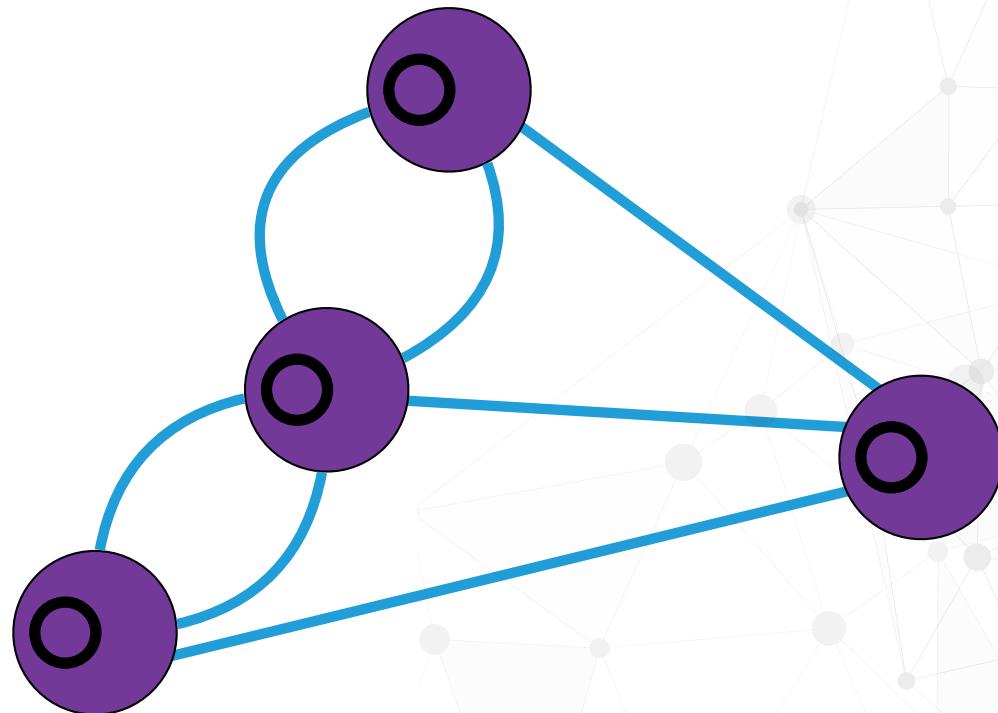
You can cross
every bridge
exactly once if and
only if at most two
have an odd
degree

"This question is so banal, but seemed to me worthy of attention in that neither geometry, nor algebra, nor even the art of counting was sufficient to solve it"



GraphAware®

So what is a graph?



A set of nodes
(representing things)

A set of edges
(representing the relationships between the things)



Why are Graphs Useful?

Our Aim:

What pets does Alice own?



GraphAware®

Why are Graphs Useful?

Our Aim:

What pets does Alice own?



ID	Name
1	Alice
2	Bob
3	Colin

PersonID	PetID
1	A
1	B
1	C



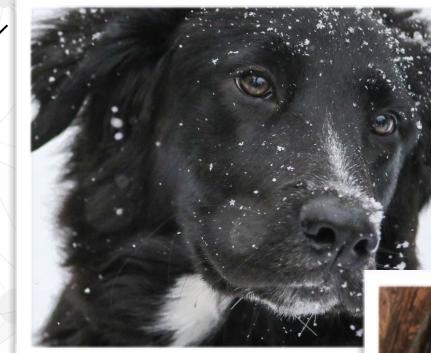
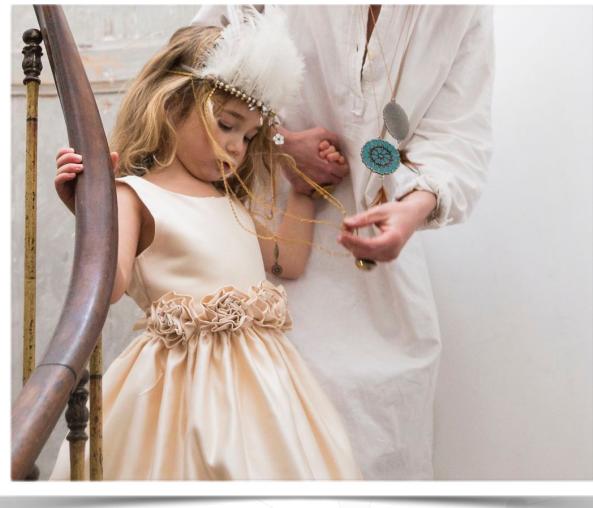
ID	Species	Name
A	Dog	Fluff
B	Iguana	Scales
C	Parrot	Feather



Why are Graphs Useful?

Our Aim:

What pets does Alice own?



GraphAware®

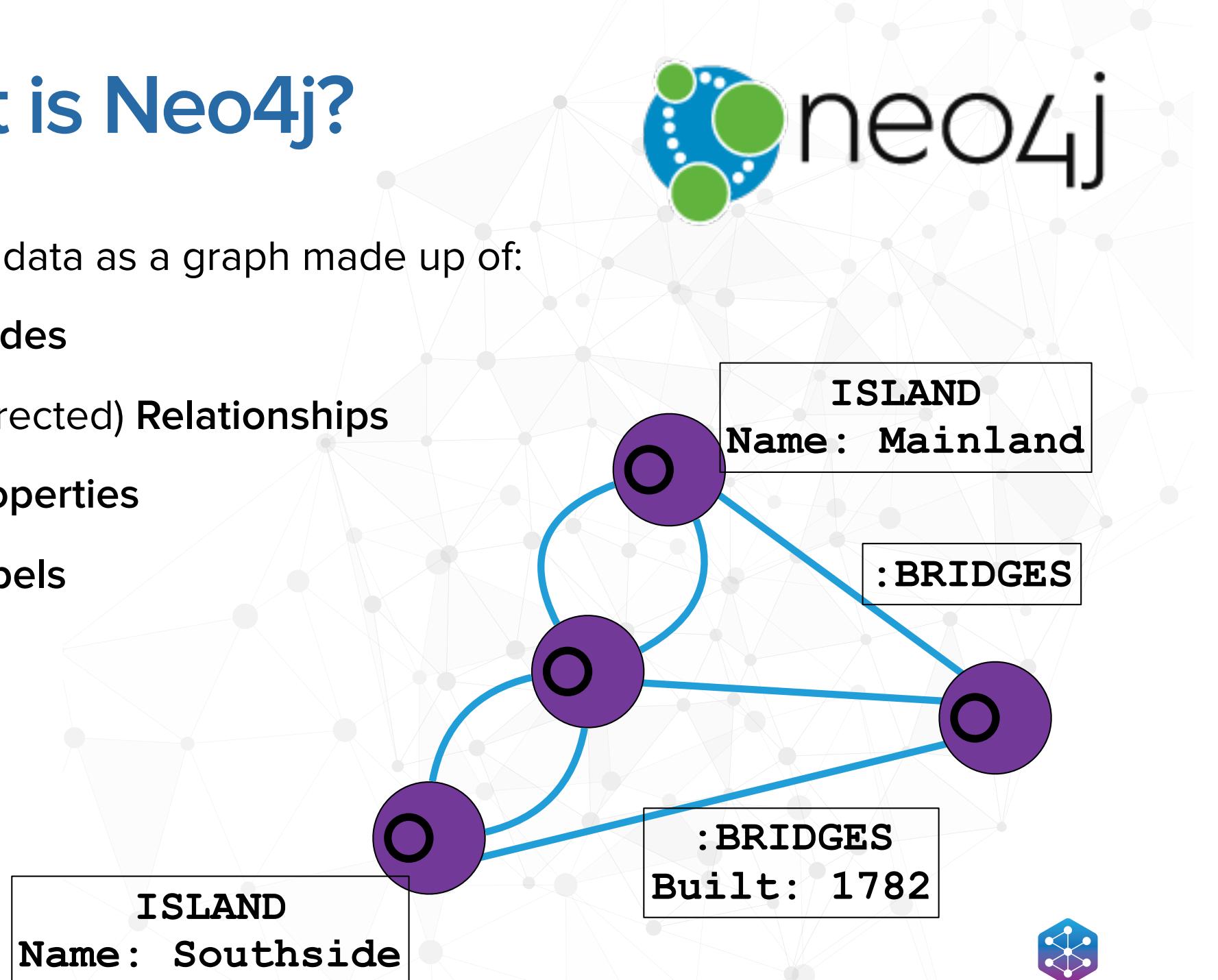
So what is... **Neo4j**



GraphAware®

What is Neo4j?

- Stores data as a graph made up of:
 - **Nodes**
 - **(Directed) Relationships**
 - **Properties**
 - **Labels**



GraphAware®

What is Neo4j?

- Embeddable in a Java service, or as a separate server
- Full ACID transactions
- Enforces a graph model, but beyond that has a flexible schema
- Used by many companies: currently on Version 3.0
- Free ‘Community’ edition, and AGPL ‘Enterprise’ edition



GraphAware®

Look at Neo4j Use Cases



GraphAware®

Recommendations



GraphAware®

Social Network



GraphAware®

Impact Analysis



GraphAware®

Routing and Logistics



GraphAware®

Intelligence & Fraud Detection



GraphAware®

Data Modelling



GraphAware®

Adding

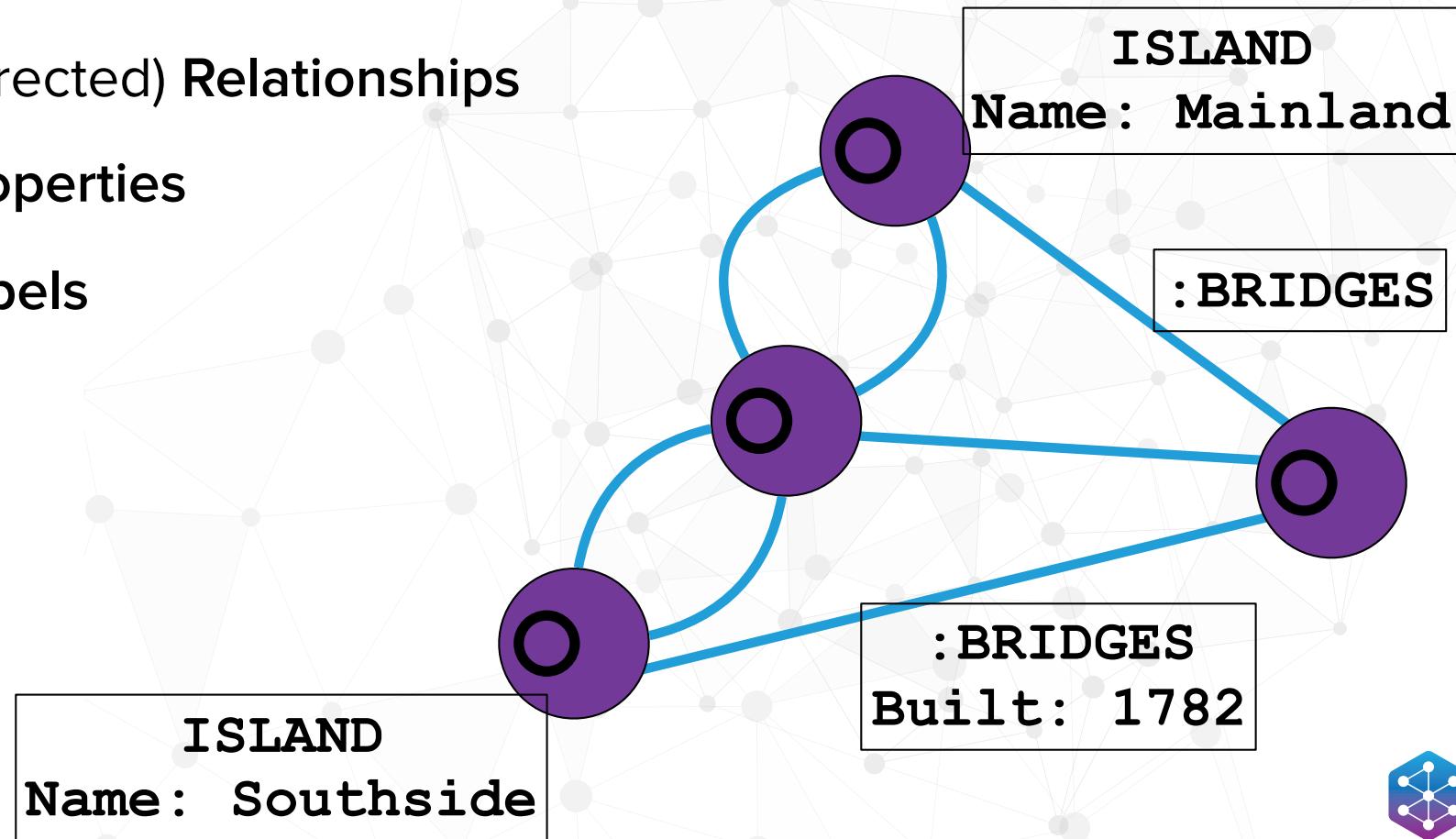
Data in Neo4j



GraphAware®

Recall: What is Neo4j?

- Stores data as a graph made up of:
 - **Nodes**
 - **(Directed) Relationships**
 - **Properties**
 - **Labels**



Data Modelling

- Neo4j enforces the graph-like structure behind the scene
- You do not **have** to define the shape of your model - it's **Schema Flexible**.
- **But** we recommend thinking about how to model your graph to avoid problems down the line



Data Modelling through **Whiteboarding**



GraphAware®

Data Modelling



- Graphs are **very whiteboard friendly**:

- Represent nodes by circles
- Represent edges by line.

- Example: Social Networking



GraphAware®

Data Modelling

- Use a whiteboard
- Start with an understanding of your current data model and requirements
- Drive your model:
 - **nodes** represent the nouns of your model
 - **labels** representing types (i.e. Enums)
 - **relationships** representing verbs
 - **properties** representing adjectives and adverbs



Data Modelling



Another Example: Movies

- Films
- Directors
- Actors
- Roles



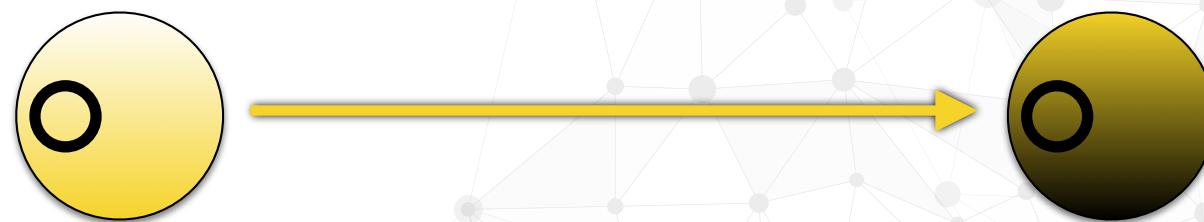
GraphAware®

Some Common **Pitfalls**



GraphAware®

Bi-Directional Relationships



Remember
Edges always have a
direction in Neo4j



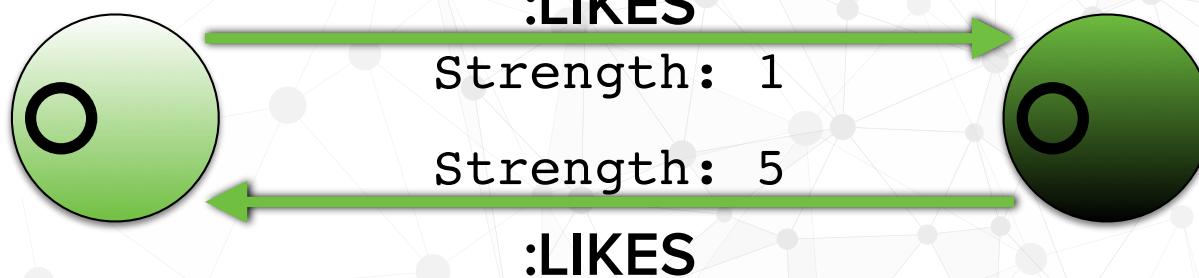
GraphAware®

Bi-Directional Relationships

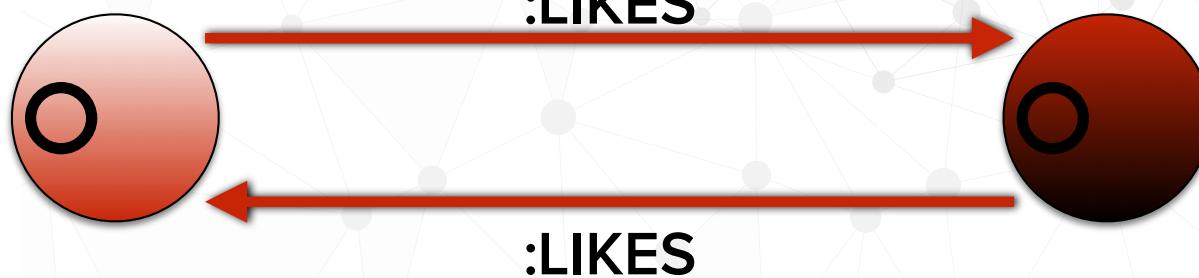
Yes!



Yes!

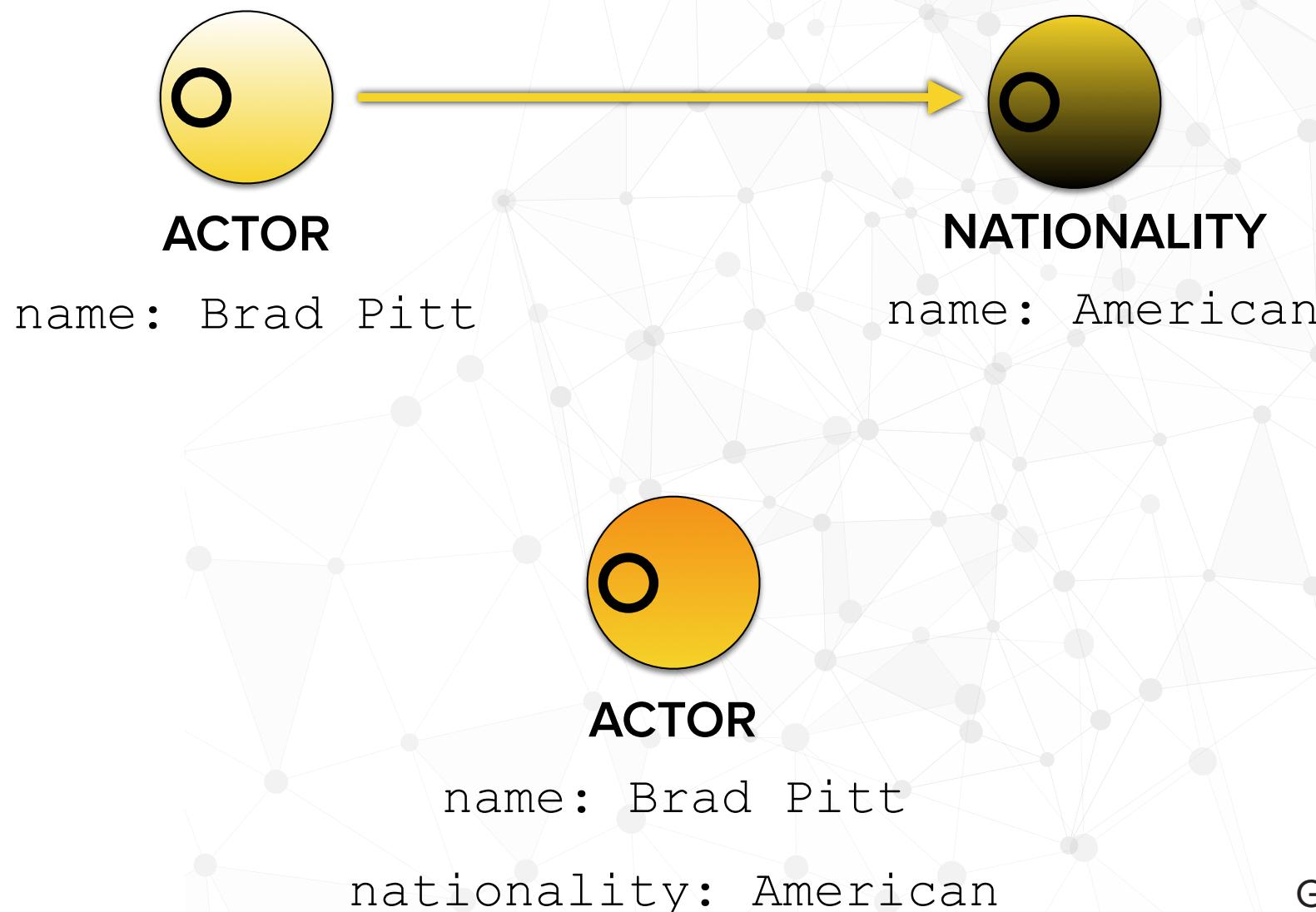


No!



GraphAware®

Properties vs. Nodes



GraphAware®

Properties vs. Nodes

An Aside:

How does Neo4j store things on the drive?



GraphAware®

Step One:

Get Neo4j working



GraphAware®

Load Data

:play movies



GraphAware®

Cypher Basics



GraphAware®

What is **Cypher**



GraphAware®

What is Cypher?

- Declarative Pattern-Matching Language
- SQL-like Syntax
- **Except:** It's defined for graphs



GraphAware®

How do we identify

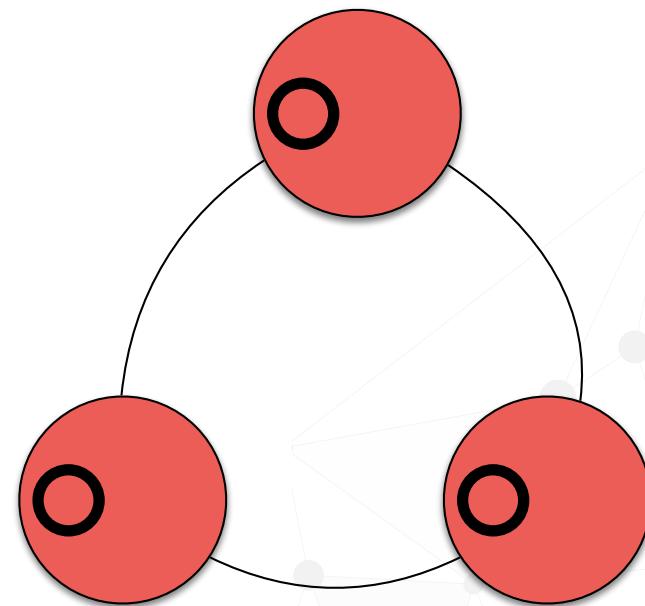
Patterns



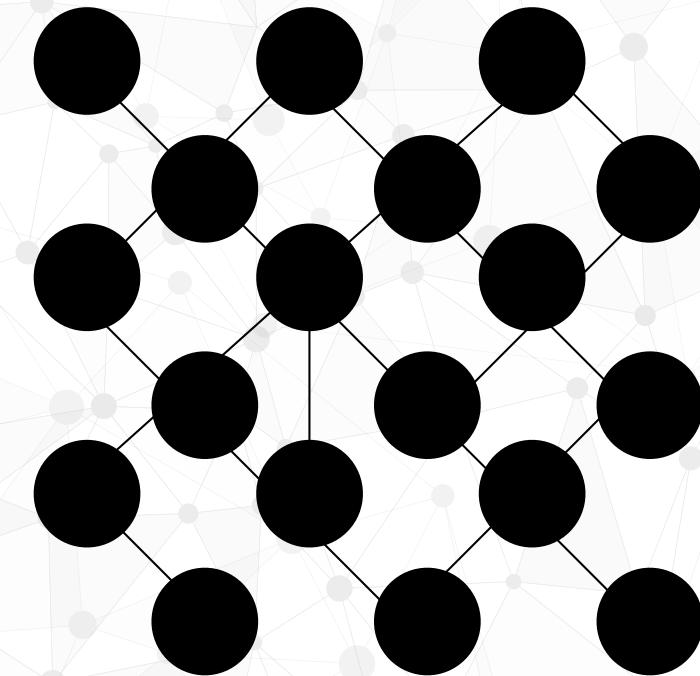
GraphAware®

Pattern Matching

Given a pattern:



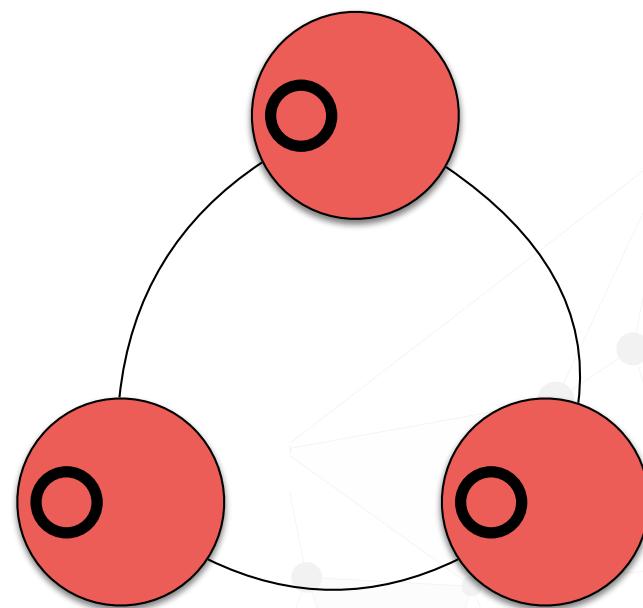
It will match in this graph



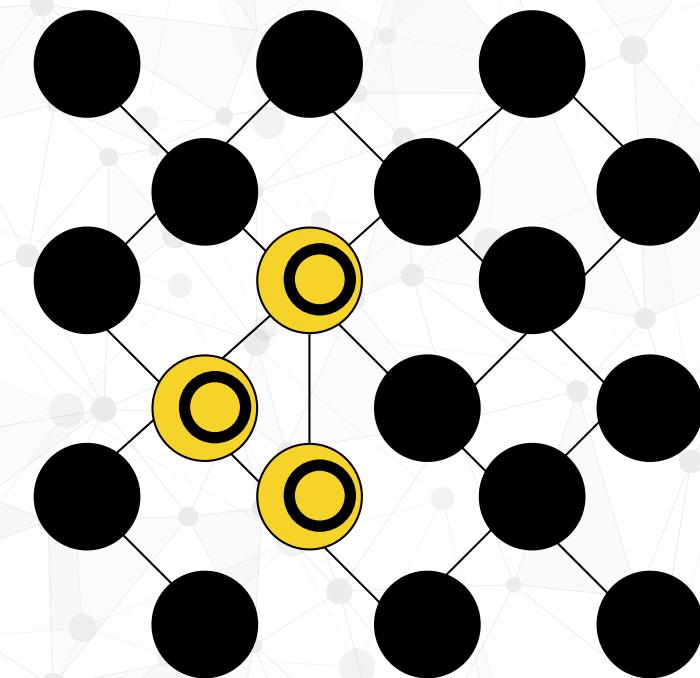
GraphAware®

Pattern Matching

Given a pattern:



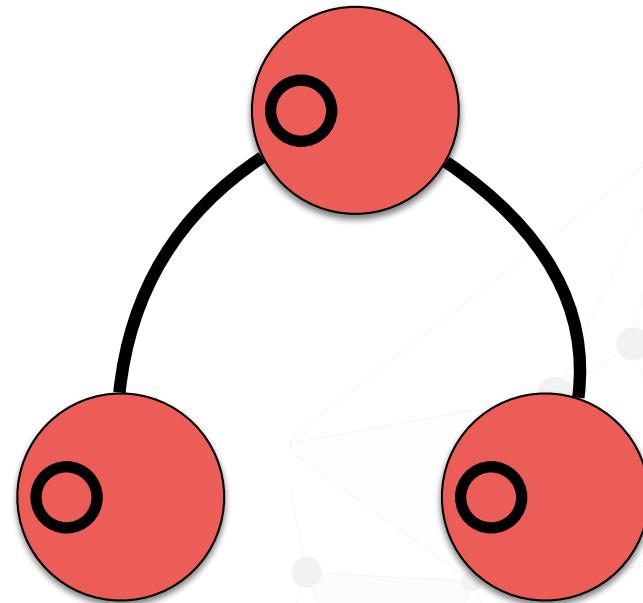
It will match in this graph



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



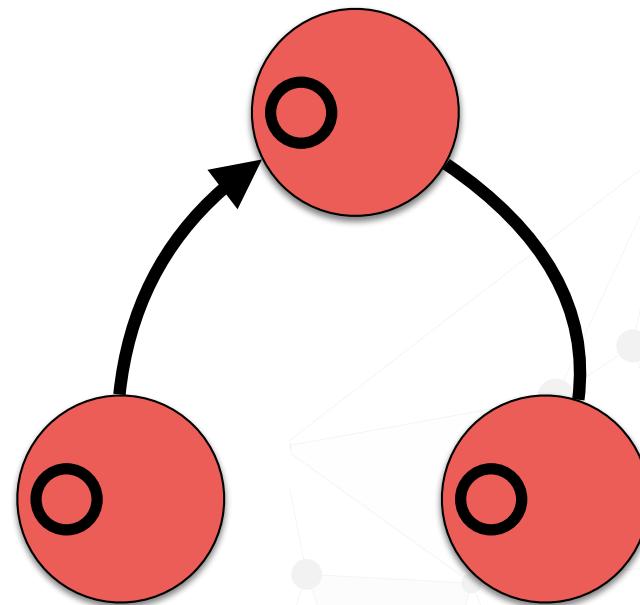
(a) -- (b) -- (c)



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



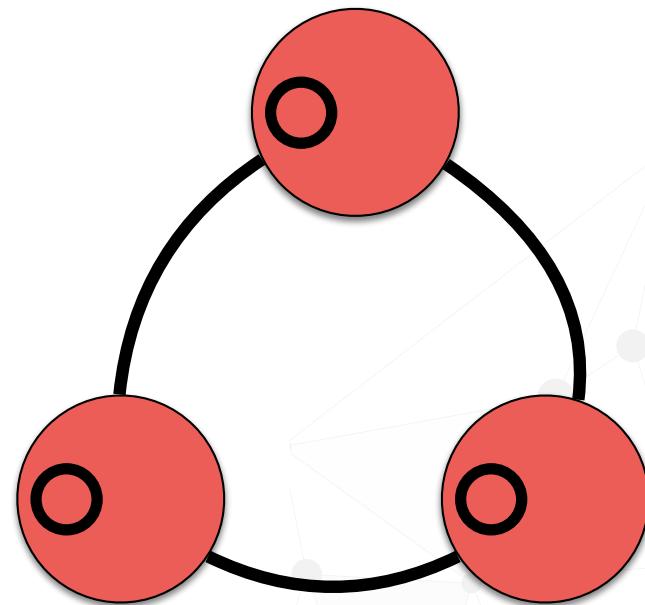
(a) --> (b) -- (c)



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



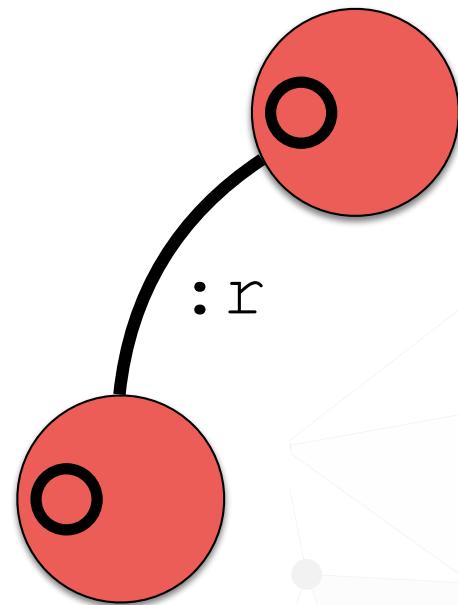
(a) -- (b) -- (c) ,
(c) -- (a)



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



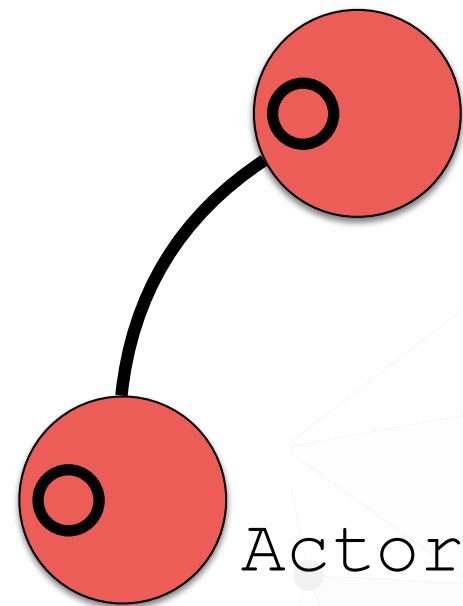
(a) - [: r] - (c)



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



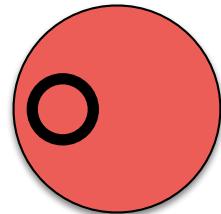
**(a:Actor)--
(c:Character)**



GraphAware®

Patterns are written as ASCII Art

Given a pattern:



name: Brad Pitt

(a {name: "Brad Pitt"})



GraphAware®

The “MATCH” Keyword

- This is what turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```



A Brief...

A Chance To Play



GraphAware®

What if we need

Filtering and Ordering



GraphAware®

The RETURN keyword

What can we return?

MATCH (a)--(b)

- Identifier
 - **RETURN p**
- Aggregation Function
 - **RETURN count(p)**
- Attributes
 - **RETURN p.name**

The WHERE keyword

```
MATCH (a)--(b)
```

WHERE clauses filter the results, such as:

```
WHERE NOT (a)<-(b)
```

```
WHERE NOT a.runtime > 180
```

```
WHERE NOT a.name = "David"
```

Ordering and Skipping

Order isn't guaranteed by default, but you can enforce it:

ORDER BY, LIMIT, SKIP

Example:

```
MATCH (a)- - (b)
```

```
RETURN a, COUNT(b) AS count
```

```
ORDER BY count DESC
```

```
LIMIT 5
```

Aggregation

Aggregation queries:

COUNT(x) – add up the number of occurrences

MIN(x) – get the lowest value

MAX(x) – get the highest value

AVG(x) – get the average value

SUM(x) – add up all the values

COLLECT(x) – collect all the occurrences into an array

Using data with **Exercises**



GraphAware®

Questions to Answer

- Which movie has the tagline “The Game Has Changed”?
- How many movies are there in the dataset?
- How many of these movies star Charlize Theron?
- Have any directors acted in their own movies?
- Who has acted in the most movies since 2000?
- How many people have costarred with Al Pacino AND Keanu Reeves?
- How many people have costarred with Al Pacino but NOT with Keanu Reeves?
- Which two actors have costarred most often?



Creating **Nodes**



GraphAware®

CREATE

CREATE (a:Label)

CREATE (a:Label {prop: "properties"})

CREATE

```
MATCH (a:User {name: 'David'}), (b:Movie  
{name: 'Titanic'})
```

```
CREATE (a)-[:LIKES]-(b)
```

Questions to Answer

- Add Multiple Users
- Add movies that they like
- Can you write a query that recommends a user what movies they may like?
 - Consider the director of the movies they already like?
 - What similar people like?
 - **Extension:** Add friends relationships between the users. How can we use this?
 - **Extension:** Add a ‘rating out of 5’ property to the like relationship. How can we use this?



Day 2

Using Neo4j at Cablevision

Session 1

Introduction and Use Cases of Neo4j

Session 2

Investigation: What data is there?

Session 3

Investigation: What would we like out of
this exercise?

Session 4

Why Neo4j?



GraphAware®



Why are we looking to **NoSQL Solutions**



GraphAware®

What is SQL?



GraphAware®



Why are we moving away from SQL now?

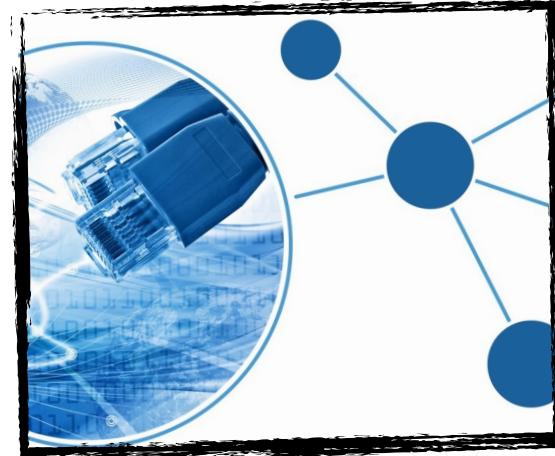


GraphAware®

My thoughts...



Semi-Structured
Data



More Connected
Data



Bigger Data



Realtime Data



GraphAware®

A Brief History of **Graphs**



GraphAware®

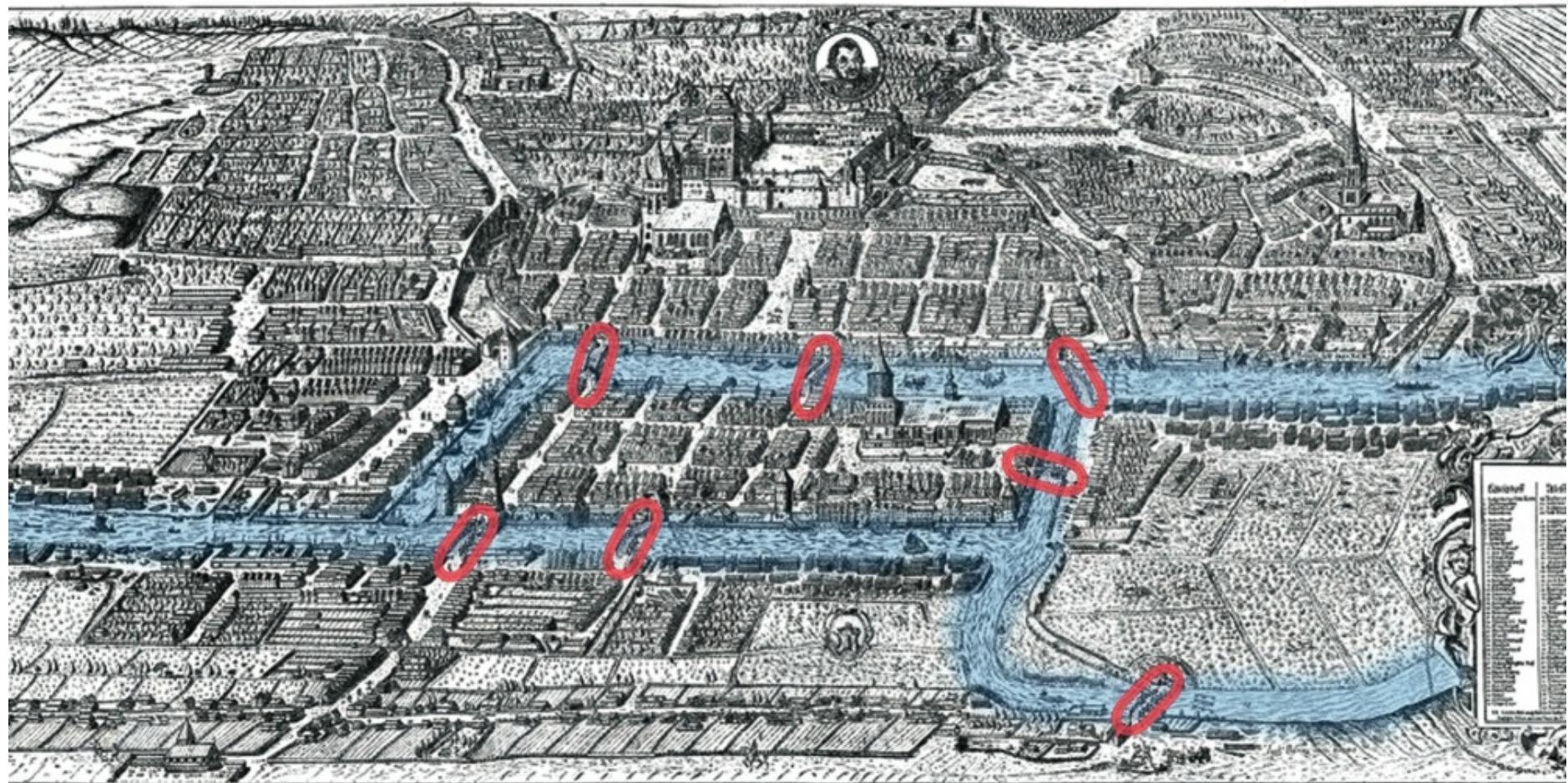
Our Graph Hero....



GraphAware®

The Seven Bridges of Koenigsberg

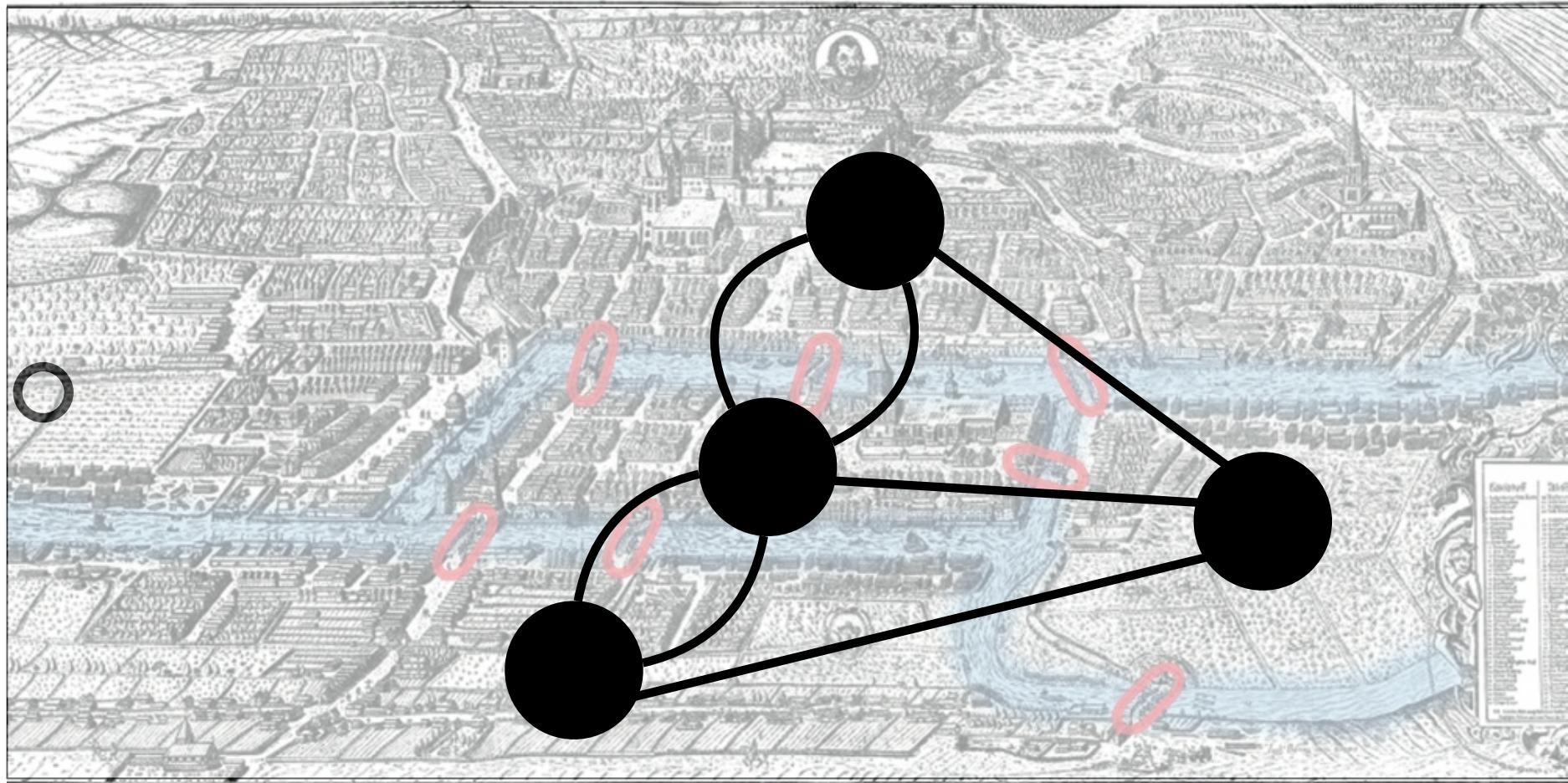
Gedenkblatt zur sechshundert jährigen Jubelfeier der Königlichen Haupt und Residenz-Stadt Königsberg in Preußen.



GraphAware®

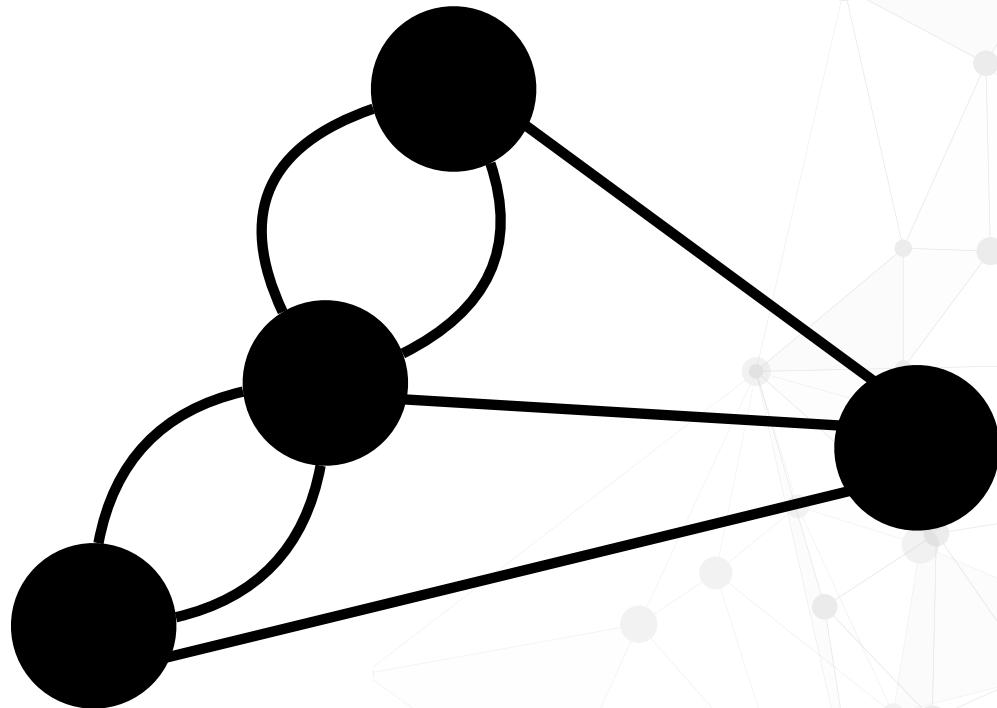
The Seven Bridges of Koenigsberg

Gedenkblatt zur sechshundert jährigen Jubelfeier der Königlichen Haupt und Residenz-Stadt Königsberg in Preußen.



GraphAware®

The Seven Bridges of Koenigsberg

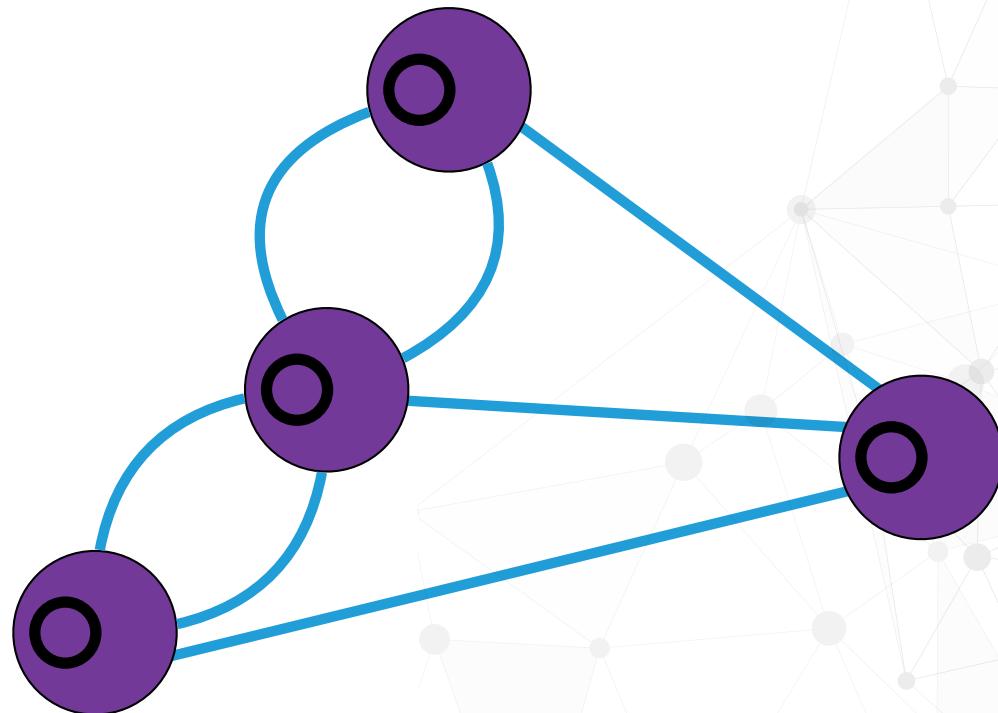


You can cross
every bridge
exactly once if and
only if at most two
have an odd
degree

"This question is so banal, but seemed to me worthy of attention in that neither geometry, nor algebra, nor even the art of counting was sufficient to solve it"



So what is a graph?



A set of nodes
(representing things)

A set of edges
(representing the relationships between the things)



Why are Graphs Useful?

Our Aim:

What pets does Alice own?



GraphAware®

Why are Graphs Useful?

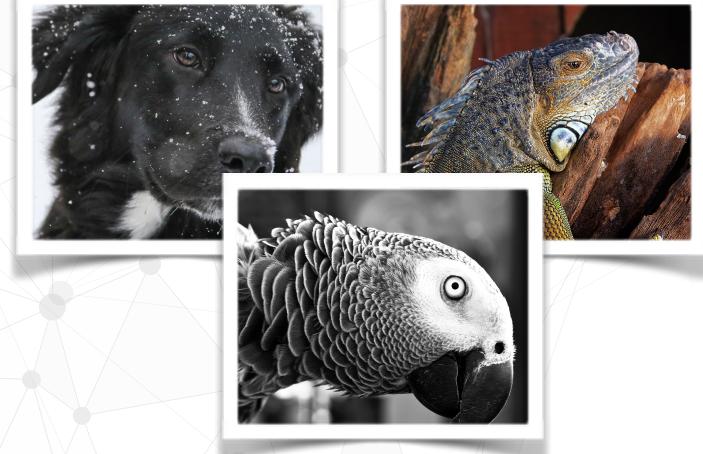
Our Aim:

What pets does Alice own?



ID	Name
1	Alice
2	Bob
3	Colin

PersonID	PetID
1	A
1	B
1	C



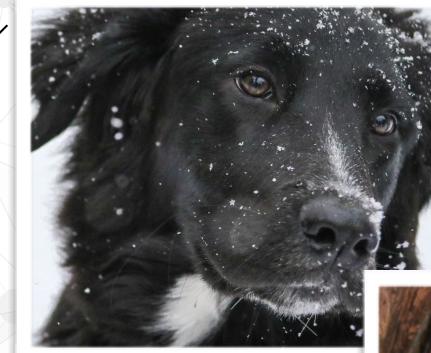
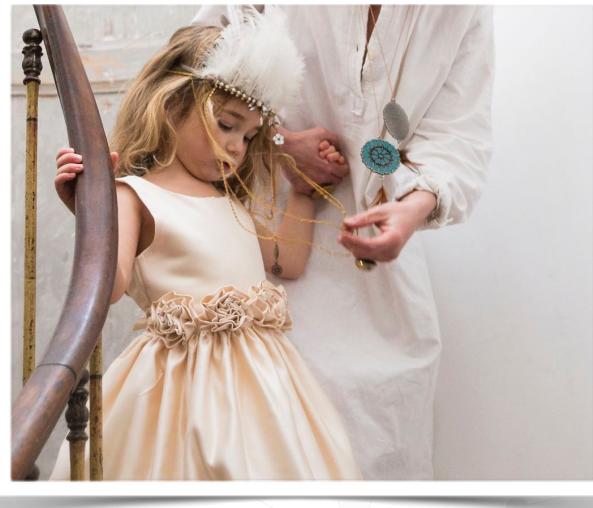
ID	Species	Name
A	Dog	Fluff
B	Iguana	Scales
C	Parrot	Feather



Why are Graphs Useful?

Our Aim:

What pets does Alice own?



GraphAware®

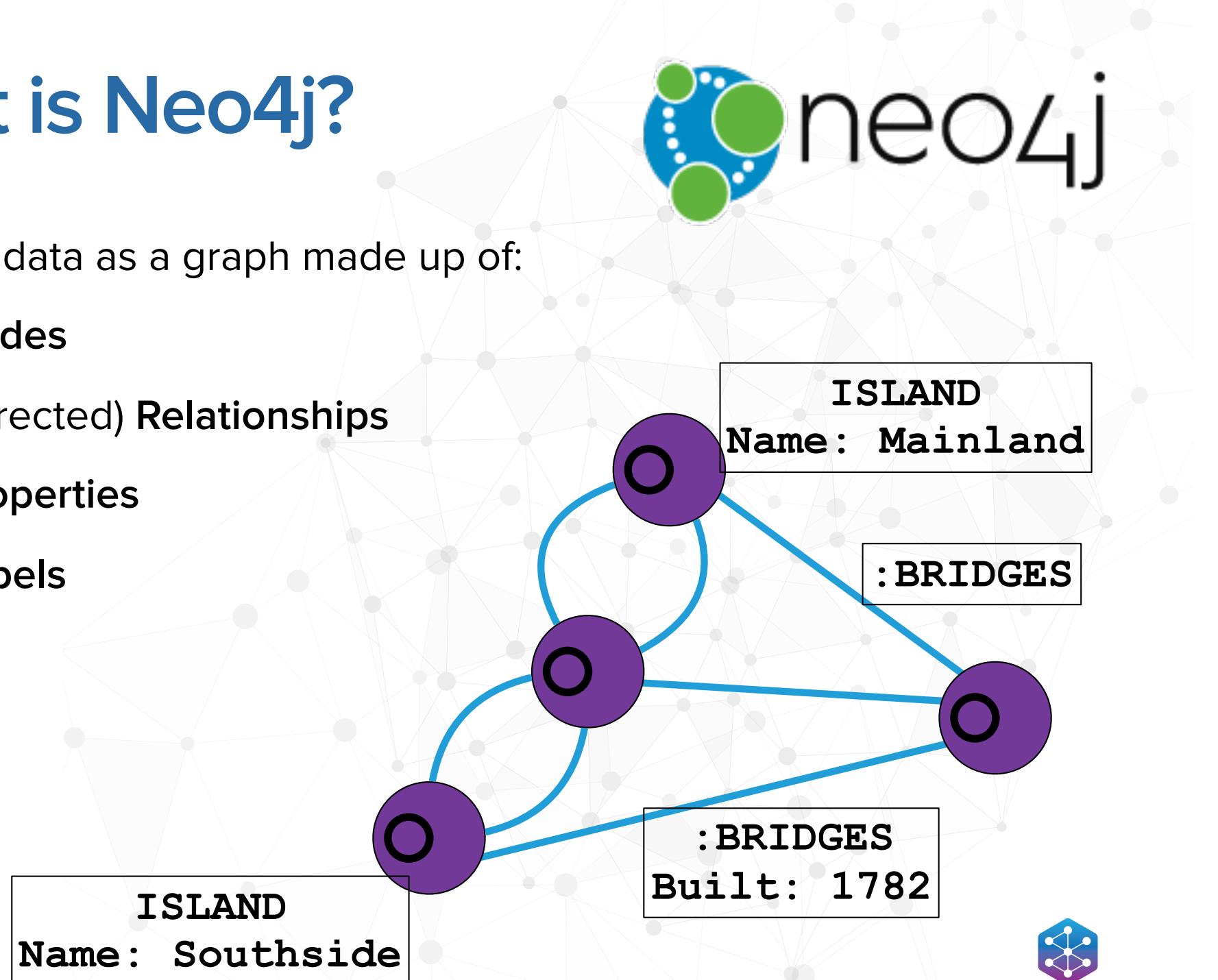
So what is... **Neo4j**



GraphAware®

What is Neo4j?

- Stores data as a graph made up of:
 - **Nodes**
 - **(Directed) Relationships**
 - **Properties**
 - **Labels**



GraphAware®

Adding

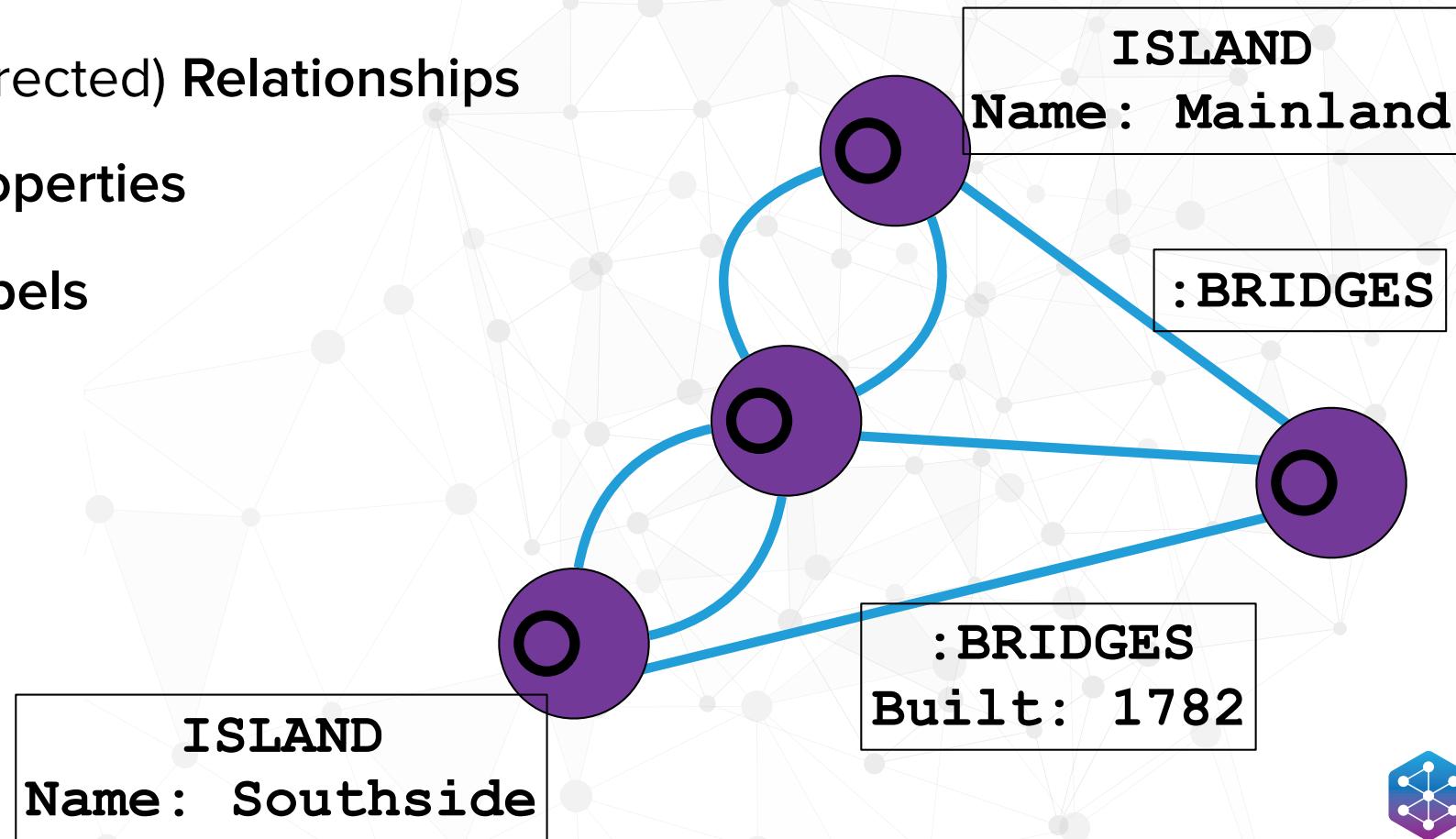
Data in Neo4j



GraphAware®

Recall: What is Neo4j?

- Stores data as a graph made up of:
 - **Nodes**
 - **(Directed) Relationships**
 - **Properties**
 - **Labels**



Data Modelling



- Graphs are **very whiteboard friendly**:

- Represent nodes by circles
- Represent edges by line.

- Example: Social Networking



GraphAware®

Look at Neo4j Use Cases



GraphAware®

Recommendations



GraphAware®

Social Network



GraphAware®

Impact Analysis



GraphAware®

Routing and Logistics



GraphAware®

Intelligence & Fraud Detection



GraphAware®

Day 3

Data Import with Neo4j

Session 1

Importing Data

Session 2

Whiteboard: Data Modelling

Session 3

Hands-On: Let's Import the Data

Session 4



Importing Data



GraphAware®

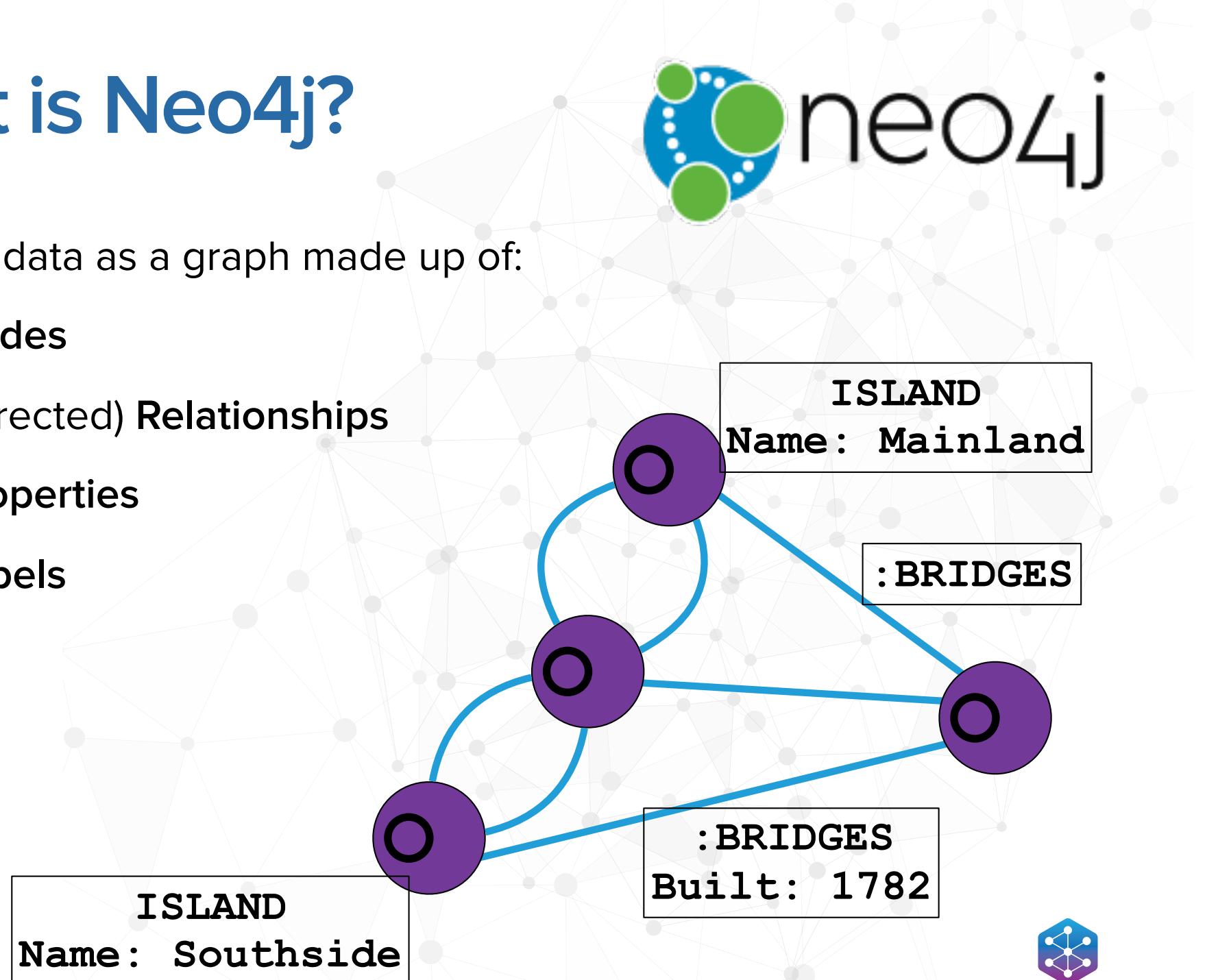
Creating Data with **Cypher Queries**



GraphAware®

What is Neo4j?

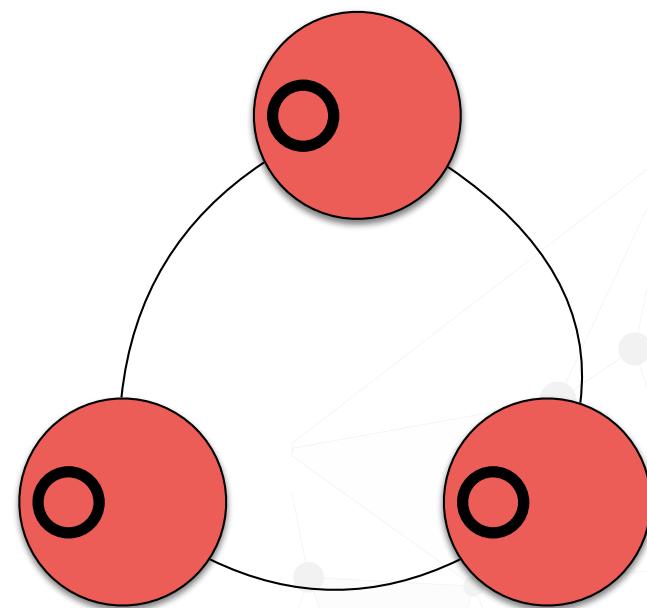
- Stores data as a graph made up of:
 - **Nodes**
 - **(Directed) Relationships**
 - **Properties**
 - **Labels**



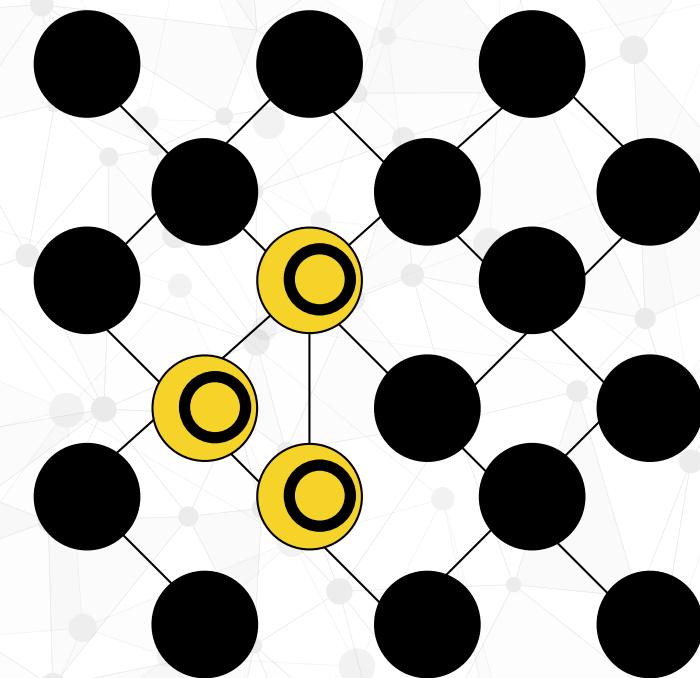
GraphAware®

Pattern Matching

Given a pattern:



It will match in this graph



GraphAware®

The “MATCH” Keyword

- This is what turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```



The “MATCH” Keyword

- This is what turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```

Keywords



GraphAware®

The “MATCH” Keyword

- This is what turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```

Identifiers



The “MATCH” Keyword

- This is what gives turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```

Pattern



What else can MATCH do?

- This is what gives turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
DELETE a, c
```



CREATE

CREATE (a:Label)

CREATE (a:Label {prop: "properties"})

CREATE

```
MATCH (a:User {name: 'David'}), (b:Movie  
{name: 'Titanic'})
```

```
CREATE (a)-[:LIKES]->(b)
```

CREATE

But what happens if we do:

CREATE

(a:User {name: 'David'})

- [:LIKES] ->

(b:Movie {name: 'Titanic'})

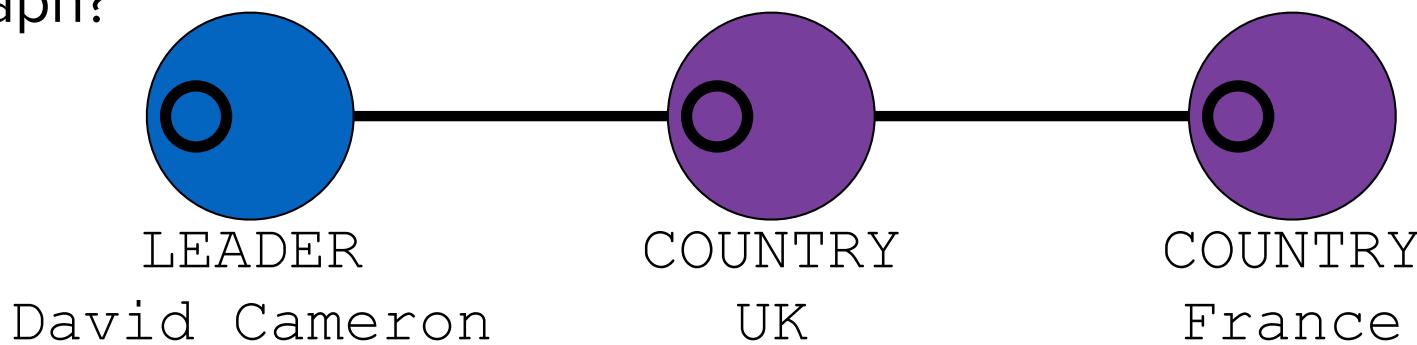
without the “MATCH”

MERGE

- Merge can either be on its own, **or** follow a ‘MATCH’ clause
- Merge takes a pattern
 - **If** this pattern exists in the graph, MERGE does nothing.
 - **If** this pattern does not exist, it will create the whole pattern *except for any ‘bound identifiers’*
- **A bound identifier is an identifier used in a previous expression**

CREATE vs. MERGE: Exercises

- Create a ‘Country’ node for USA, and create another node for Canada. Create a relationship between them.
- Write a query that creates a ‘Leader’ node for Barack Obama, and connects him to USA.
 - What happens if you run this query more than once? Make sure that only one relationship is created.
- Do the same for David Cameron (Leader of the UK). This time, your query should create the UK node if it doesn't exist.
- Write a query that creates this pattern. How does it work in our graph?



MERGE (dc:Leader {name: 'David Cameron'})

MERGE (uk:Country {name: 'UK'})

MERGE (france:Country {name: 'France'})

MERGE (dc)-[:LEADER_OF]->(uk)-[:BORDERS]->(france)

Importing through

LOAD CSV



GraphAware®

LOAD CSV

LOAD CSV is a cypher command to load data from a file.

The syntax is very simple:

```
LOAD CSV FROM {URI} AS line
```

```
{MORE COMMANDS}
```



LOAD CSV

Example:

```
LOAD CSV FROM "http://data.neo4j.com/  
examples/person.csv" AS line
```

```
MERGE (n:Person {id:toInt(line[0])})
```

```
RETURN n
```



GraphAware®

Some Options

Example:

```
LOAD CSV WITH HEADERS FROM "http://  
data.neo4j.com/examples/person.csv" AS line  
  
MERGE (n:Person {id:toInt(line.ID)})  
  
RETURN n
```



GraphAware®

Some Options

Example:

```
LOAD CSV FROM "http://data.neo4j.com/  
examples/person.csv" AS line
```

FIELDTERMINATOR ' | '

```
MERGE (n:Person {id:toInt(line[0])})
```

```
RETURN n
```



GraphAware®

Some Options

Example:

USING PERIODIC COMMIT

```
LOAD CSV FROM "http://data.neo4j.com/  
examples/person.csv" AS line
```

```
MERGE (n:Person {id:toInt(line[0])})
```

```
RETURN n
```



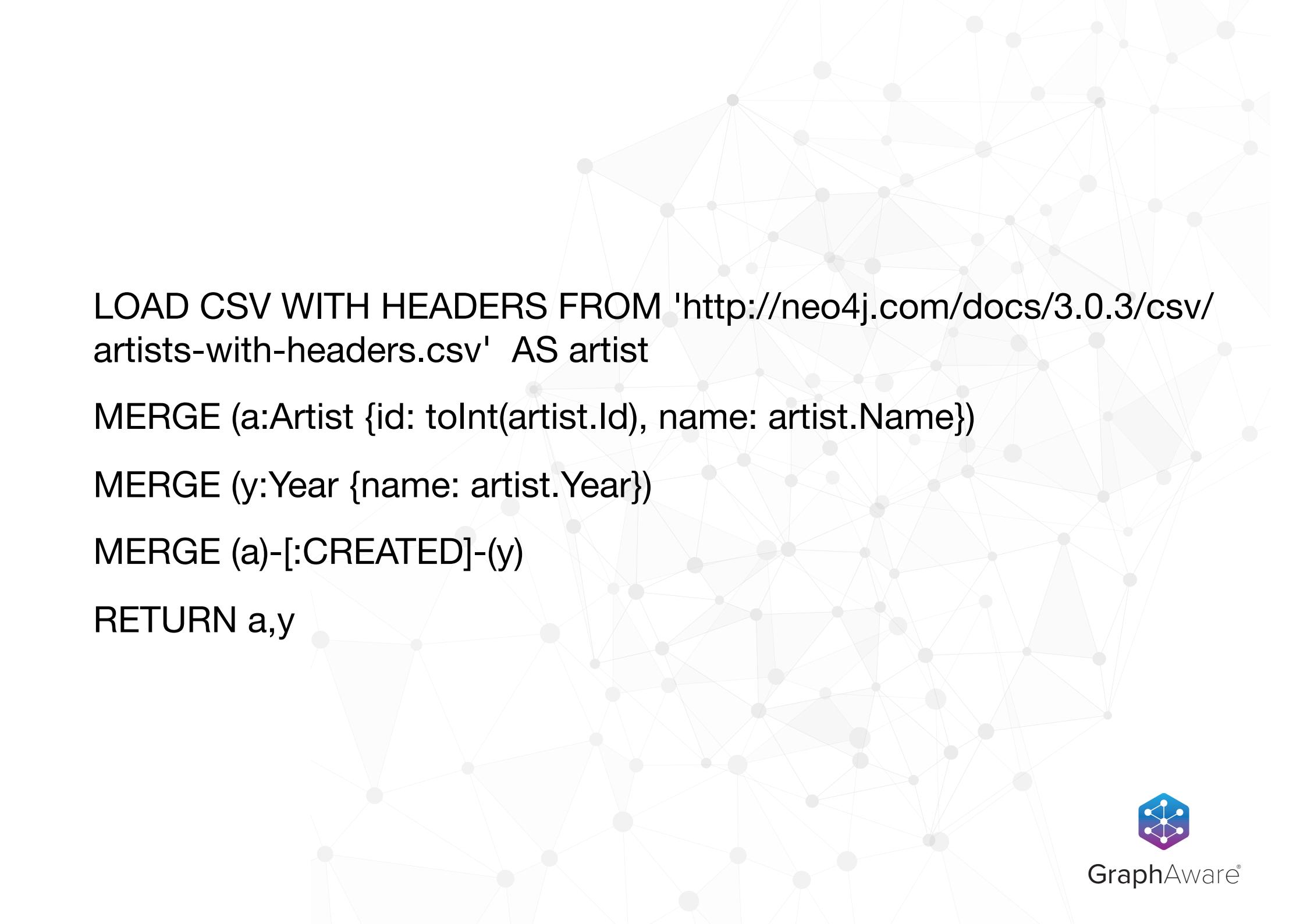
GraphAware®



```
LOAD CSV FROM "http://data.neo4j.com/examples/person.csv" AS line  
MERGE (n:CSV_Person {id: toInt(line[0])})  
RETURN n
```



GraphAware®



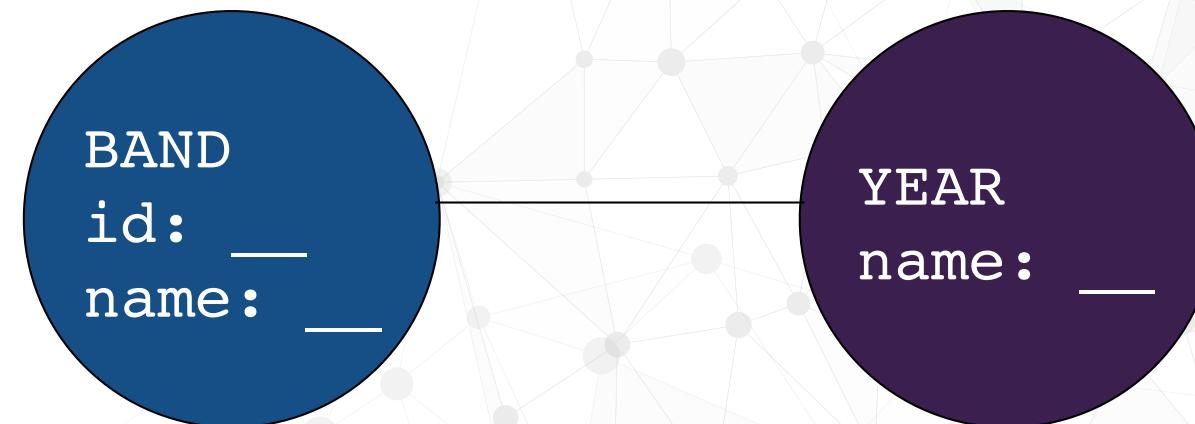
```
LOAD CSV WITH HEADERS FROM 'http://neo4j.com/docs/3.0.3/csv/artists-with-headers.csv' AS artist  
MERGE (a:Artist {id:toInt(artist.Id)}, name: artist.Name}  
MERGE (y:Year {name: artist.Year})  
MERGE (a)-[:CREATED]-(y)  
RETURN a,y
```



GraphAware®

LOAD CSV Exercises

Load the following files into Neo4j with the data model:



<http://neo4j.com/docs/3.0.3/csv/artists.csv>

<http://neo4j.com/docs/3.0.3/csv/artists-with-headers.csv>

<http://neo4j.com/docs/3.0.3/csv/artists-fieldterminator.csv>



GraphAware®

Importing through

Programmatic Tools



GraphAware®

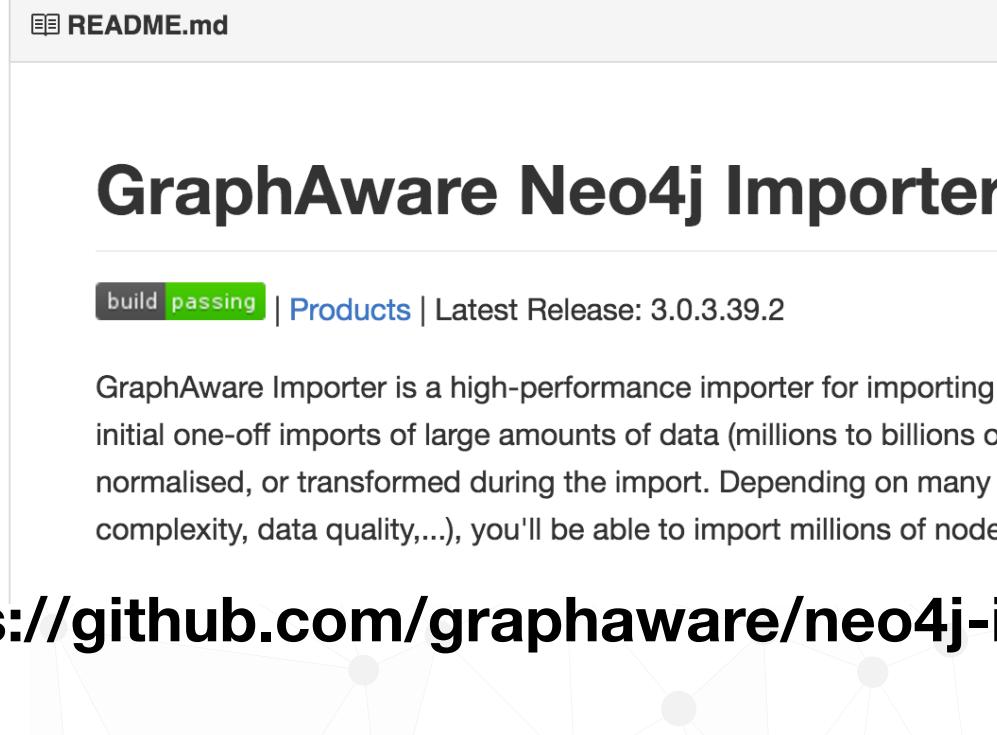
LOAD CSV

LOAD CSV is a great tool for fast data import, but:

- It can be slow for very large data sets
- It requires you to have data in a CSV format
- It doesn't provide you all the customisation you want

GraphAware Neo4j Importer

Open source library that connects JDBC to Neo4j



The screenshot shows the GitHub README page for the GraphAware Neo4j Importer. At the top left is a file icon followed by the text "README.md". Below this is a large heading "GraphAware Neo4j Importer". Underneath the heading is a button with the text "build passing" and a green bar indicating success. To the right of the button are links to "Products" and "Latest Release: 3.0.3.39.2". The main content area contains a paragraph of text describing the importer's purpose and capabilities.

GraphAware Importer is a high-performance importer for importing data. It can handle initial one-off imports of large amounts of data (millions to billions of records) and can merge, filter, normalised, or transformed during the import. Depending on many things (data size, complexity, data quality,...), you'll be able to import millions of nodes.

<https://github.com/graphaware/neo4j-importer>



GraphAware®

Neo4j-Import

Terminal command that will import CSV on
the database command line



neo4j

[http://neo4j.com/docs/operations-manual/
current/#import-tool-usage](http://neo4j.com/docs/operations-manual/current/#import-tool-usage)



GraphAware®

GraphAware DataBridge

Commercial tool that provides declarative (in JSON) data import functionality from JDBC or CSV

Example node entry: The following JSON snippet shows a complete node entry

```
{  
  "type": "Satellite",  
  "labels": [  
    { "name": "Satellite" }  
  ],  
  "properties": [  
    { "name": "satellite", "column": "Object" },  
    { "name": "manned", "column": "Manned" },  
    { "name": "active", "column": "Status" }  
  ],  
  "identity": [ "Object" ],  
  "update_strategy": "unique"  
}
```

<https://bitbucket.org/vbickers/neo4j-databridge/wiki/>



GraphAware®

Bespoke Script

You can communicate with Neo4j
using REST API or BOLT Protocol
(to be covered later)



GraphAware®

Exercises



GraphAware®

Day 4

Wrap-Up & Next Steps

Session 1

Advanced Cypher

Session 2

Performance Profiling and Bottlenecks

Session 3

Hands-On: Delivery of Cypher Queries

Session 4



Advanced Cypher



GraphAware®

A brief Cypher

Recap



GraphAware®

The “MATCH” Keyword

- This is what turns your pattern into a query.
- Syntax:

```
MATCH {pattern}  
{VERB} {variable}
```

- Example

```
MATCH (a:Actor) - - (c:Character)  
RETURN a, c
```



Indexes

- Indexes are created by default on labels.
- If you are doing lots of querying on a property, you should create an index manually

CREATE INDEX ON :Label (property)

- You can drop indexes as well

DROP INDEX ON :Label (property)



The WHERE keyword

MATCH (a)--(b)

WHERE clauses filter the results, such as:

WHERE NOT (a)<-(b)

WHERE NOT a.runtime > 180

WHERE NOT a.name = "David"

Ordering and Skipping

Order isn't guaranteed by default, but you can enforce it:

ORDER BY, LIMIT, SKIP

Example:

```
MATCH (a)- - (b)
```

```
RETURN a, COUNT(b) AS count
```

```
ORDER BY count DESC
```

```
LIMIT 5
```

Aggregation

Aggregation queries:

COUNT(x) – add up the number of occurrences

MIN(x) – get the lowest value

MAX(x) – get the highest value

AVG(x) – get the average value

SUM(x) – add up all the values

COLLECT(x) – collect all the occurrences into an array

CREATE

```
MATCH (a:User {name: 'David'}), (b:Movie  
{name: 'Titanic'})
```

```
CREATE (a)-[:LIKES]->(b)
```

LOAD CSV

Example:

```
LOAD CSV FROM "http://data.neo4j.com/  
examples/person.csv" AS line
```

```
MERGE (n:Person {id:toInt(line[0])})
```

```
RETURN n
```



GraphAware®

Working with Arrays



GraphAware®

Arrays

Arrays are supported in Neo4j. You can use them as constants or properties on nodes:

```
CREATE (n:Person {property: [1, 2, 3]})
```

You can use nested/mixed collections in Cypher, but **not** in properties.



GraphAware®

Accessing Arrays

A number of ‘predicates’ (i.e. functions that return TRUE or FALSE) are defined for arrays.

These are all **TRUE**

1 IN [1, 2, 3]

ALL (x IN [1, 2, 3] WHERE x > 0)

ANY (x IN [1, 2, 3] WHERE x > 2)

NONE (x IN [1, 2, 3] WHERE x < 0)

SINGLE (x IN [1, 2, 3] WHERE x < 2)



Accessing Arrays

There are other functions to find out about arrays

- **array[i]** returns the i^{th} element of the array. Negative numbers count backwards from the end
- **array[i..j]** returns the elements between i and j. (Including i, not including j)
- **size(array)**
- **UNWIND (array)** turns an array back into a series of rows



Arrays

COLLECT is an aggregation function that turns the results into an array

```
MATCH (n:Category)--(g:Genre) RETURN n.name,  
COLLECT(g.name)
```

n.name	COLLECT(g.name)
Internacionales	[Tecnología, Variedades, Especial, Turismo, Concierto, Infantil, Serie, Negocios, Humorístico, Religioso, Agro, Musical, Deportivo, Entretenimientos, Cultural, Magazine, Cocina, Económica, Entrevista, Noticiero, Periodístico, Interés General]
Ciclo	[Erotica, Variedades, Película, Documental, Serie]
Series	[Romántica, Musical, C.Ficción, Biográfico, C.Dramática, Bélica, Suspensión, Humorístico, Cocina, Erótica, Infantil, Thriller, Policial, Interés General, Telenovela, Acción, D.Animados, Comedia, Drama, Serie, Documental, Terror, Animación, Aventura, Miniserie]
Cine	[Miniserie, Ficción, Suspensión, Bélica, Cortometrajes, Musical, Corto, Biográfico, Crimen, Especial, Policial, C.Musical, C.Romántica, C.Dramática, Romántica, Familiar, Terror, C.Ficción, D.Animados, Misterio, Infantil, Drama, Thriller, Animación, Acción, Western, Comedia, Aventura, Erótica, Película, Fantasía, Documental]



GraphAware®

Arrays

FOREACH lets you run a statement for each statement in an array

FOREACH(x in [1, 2, 3] | STATEMENT)

For example:

FOREACH(x in [1, 2, 3] | CREATE (n:Test {prop: x})



Creating Arrays

There's many ways to create arrays

- ‘Array Literal’ notation - i.e. writing out the array
 - `[1, 2, 3]`
- COLLECT up a group of returned results
 - `MATCH (a) - - (b) RETURN a, COLLECT(b)`
- Some functions - e.g. SPLIT return an array
 - `SPLIT("a,b,c,d", ",", ",")`
 - `//Returns ['a', 'b', 'c', 'd']`



Array Exercises

- Find the **sum**, **average**, **max** and **min** of the array [1, 2, 3, 4] using Cypher
- Which predicate query would you use to work out:
 - Are any of them bigger than 4?
 - Is at least one bigger than 4?
 - Are all of them bigger than 4?
- Using ‘COLLECT’, can you work out which category has the most genres associated with it?
- **And finally:** Using ‘LOAD CSV’ ‘SPLIT’ and ‘FOREACH’ import the Actors and Directors in to the data base



How to work with Data

Mutation



GraphAware®

Some ways we know to mutate data

CREATE

MERGE

DELETE



GraphAware®

SET

Set a property on n:

```
MATCH (n:Node)  
SET n.name = "foo"  
RETURN n
```

Add properties to n:

```
MATCH (n:Node)  
SET n.name += {prop1: 1, prop2: 2}  
RETURN n
```



GraphAware®

SET

Set a label on n:

```
MATCH (n:Node)
```

```
SET n:AnotherLabel
```

```
RETURN n
```



GraphAware®

REMOVE

Remove a label from n:

```
MATCH (n:Node)
```

```
REMOVE n:Node
```

```
RETURN n
```

Remove a property from n:

```
MATCH (n:Node)
```

```
REMOVE n.property
```

```
RETURN n
```



GraphAware®

Exercise

- Find all people that have acted in a TV Show. Give these people an Actor Label
- Find all people that have directed a TV Show. Give these people a Director Label.



And what's left of the Cypher

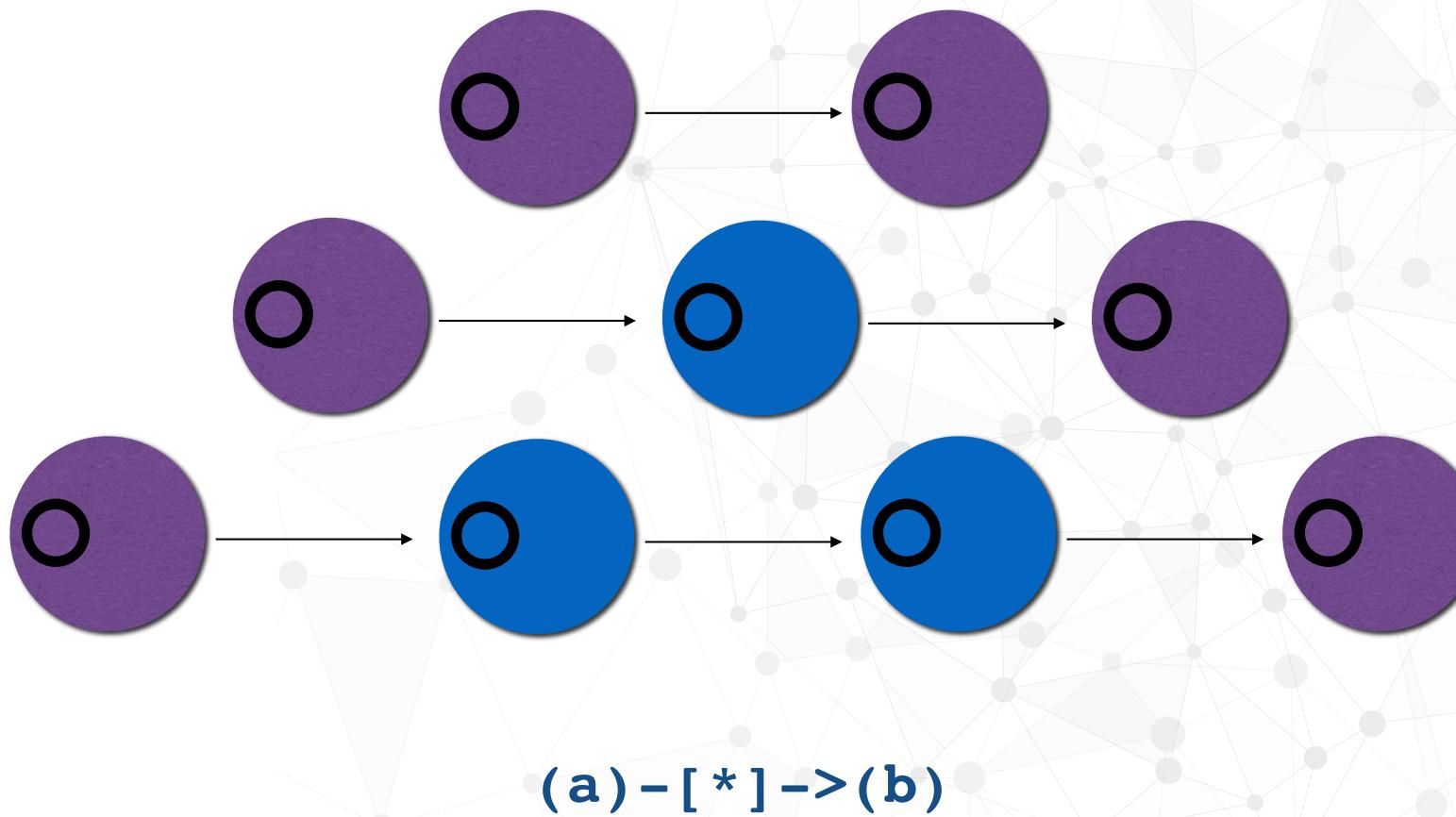
Odds and Ends



GraphAware®

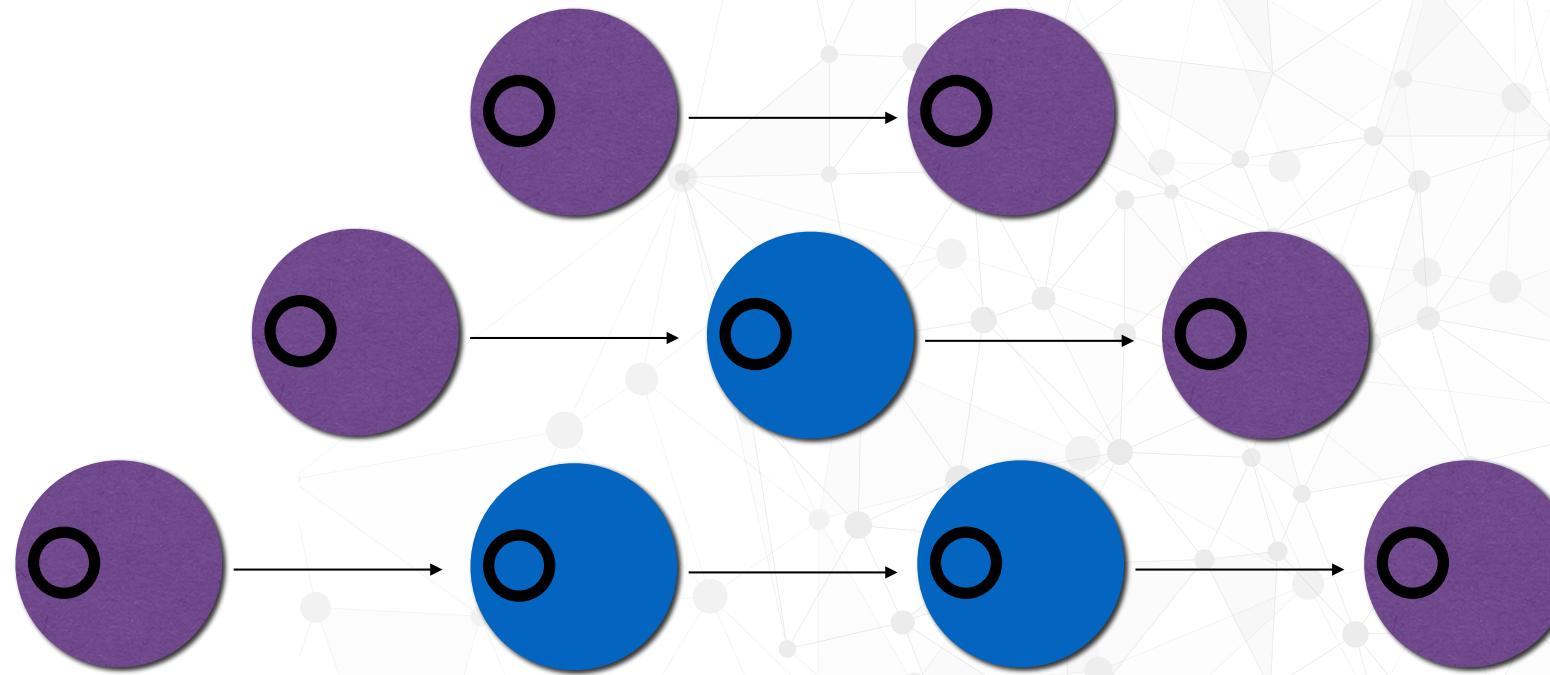
Variable Length Paths

Sometimes you want to match paths of any length:



Variable Length Paths

WARNING: (Almost) never use this: **(a)-[*]->(b)**



INSTEAD say how long the path can be: **(a)-[*1..3]->(b)**



GraphAware®

Paths as Variables

You can ask about paths later in the query:

```
MATCH p=(a)-[*]->(b)  
RETURN p
```



GraphAware®

Path Functions

Paths have functions you can use with them:

- **length(path)**
- **nodes(path)**
- **relationships(path)**
- **extract(x in nodes(path) | x.property)**



CASE/WHEN

Case works just like in SQL

CASE val

WHEN 1 THEN ...

WHEN 2 THEN ...

ELSE ...

END

CASE

WHEN val=1 THEN ...

WHEN val=2 THEN ...

ELSE ...

END

CASE/WHEN

Case works just like in SQL

CASE

WHEN line.PAIS <> "" THEN [1]

ELSE []

END

Neo4j RefCard

<http://neo4j.com/docs/cypher-refcard/current>



GraphAware®

Performance



GraphAware®

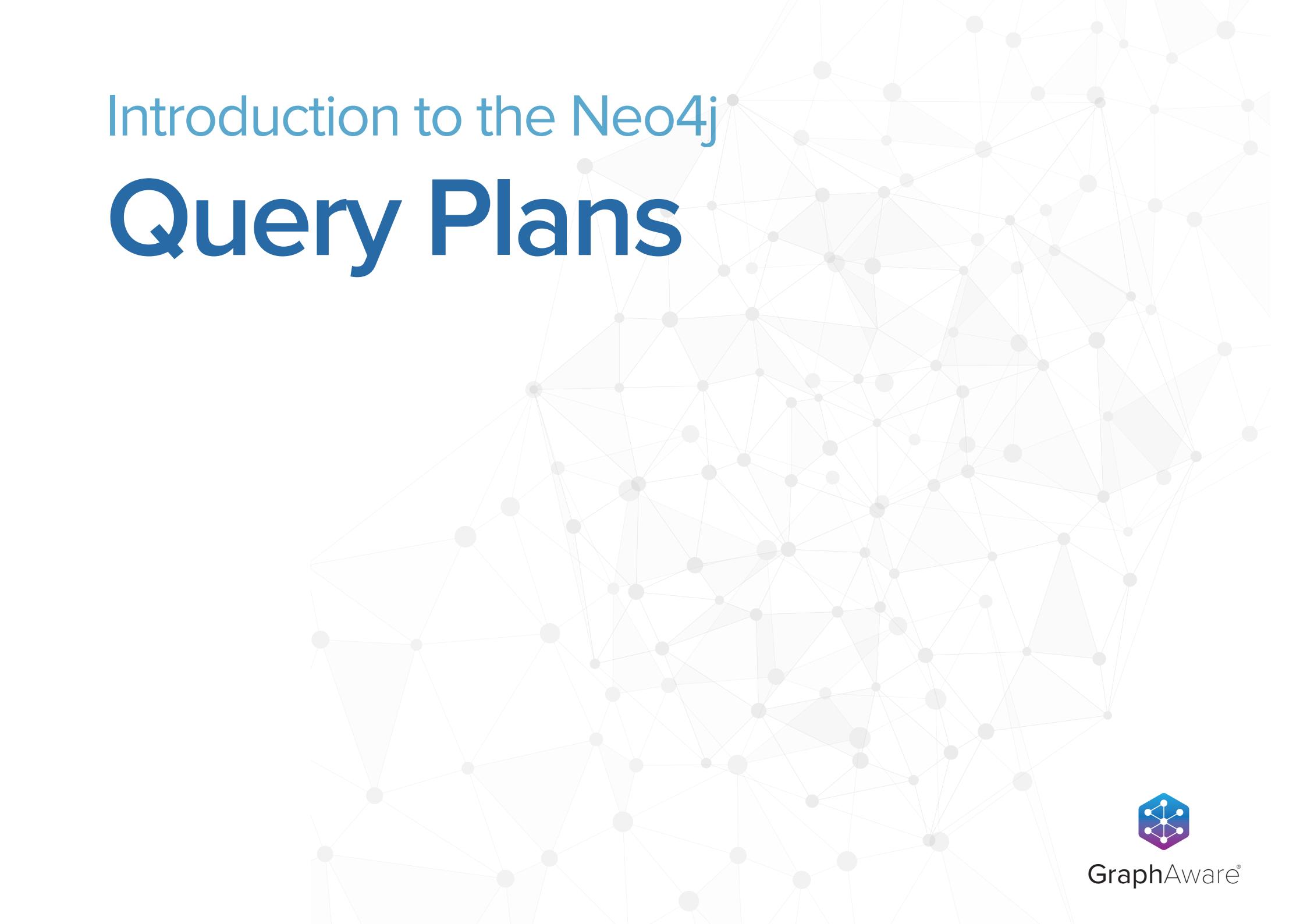
Proviso

Today is about writing efficient queries using Cypher. Other topics about performance are:

- **12th July, 2:30-6pm:** Installation, configuration and scalability
- **11th July 2:30-6pm:** Non-Cypher querying



GraphAware®



Introduction to the Neo4j

Query Plans



GraphAware®

How do I view a query plan?

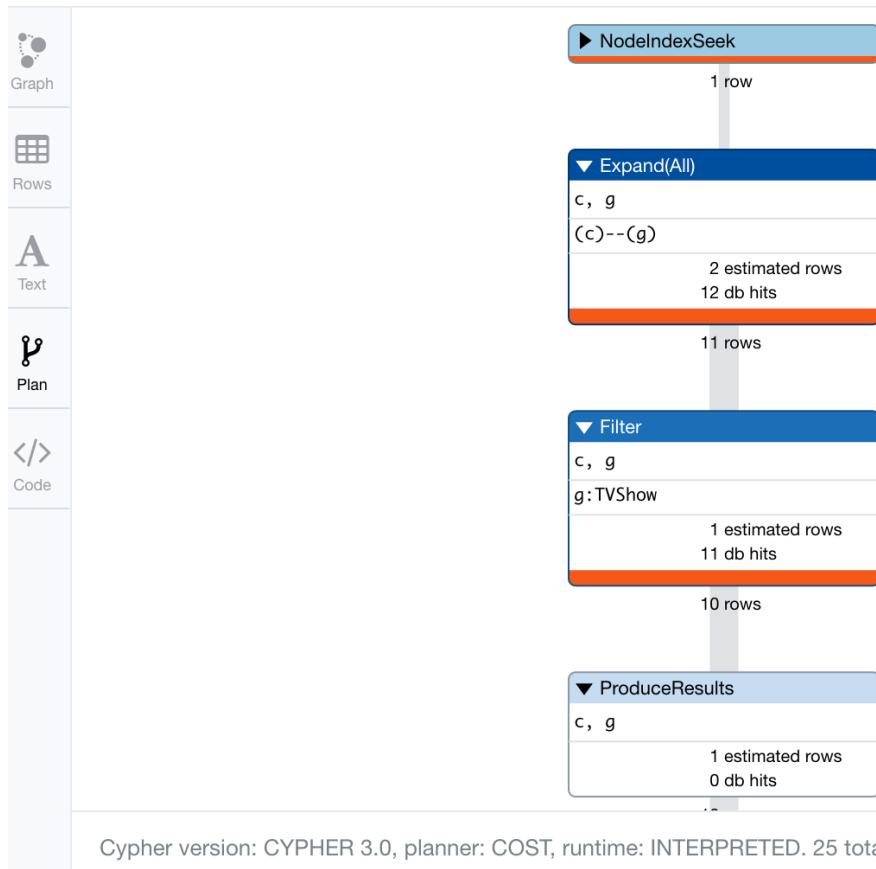
- EXPLAIN
 - This shows the execution plan without executing it or returning results
- PROFILE
 - This shows the statement and returns the results along with profiling information



How do I view a query plan?

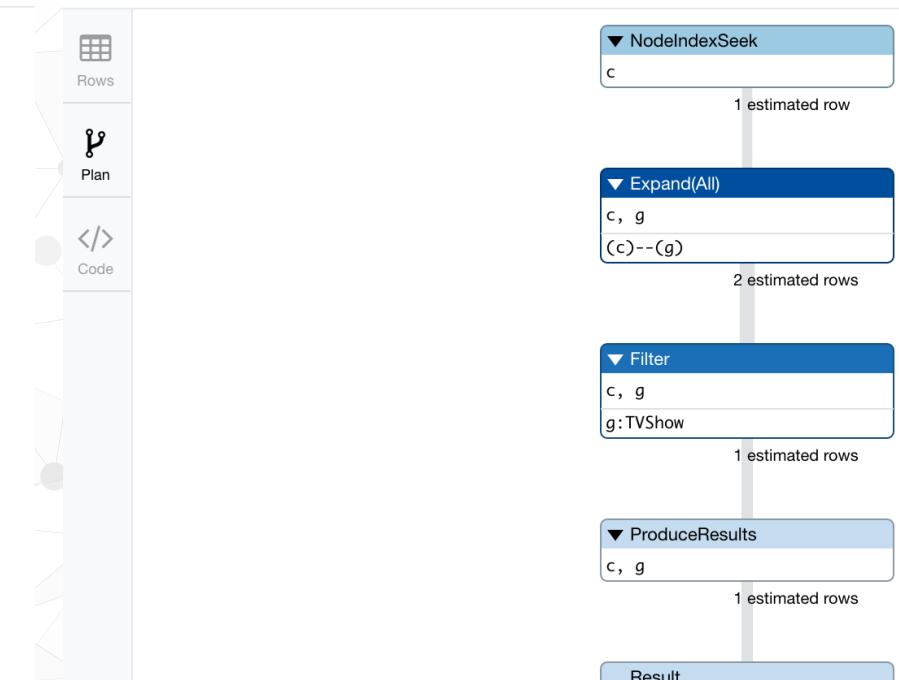
PROFILE

```
$ PROFILE MATCH (c:Device {mac: '0007815d3c'})--(g:TVShow) RETURN c,g
```



EXPLAIN

```
$ EXPLAIN MATCH (c:Device {mac: '0007815d3c'})--(g:TVShow) RETURN c,g
```



GraphAware®

Some Common Operators

All Nodes Scan	Very expensive
Label Scan	Cheaper
Node Index Seek	Cheapest



What are common **Bottlenecks**



GraphAware®

Global Graph Scans

Asking questions of the whole graph is very slow.

For example:

- What is the average age of every user?
- Which nodes (out of any of the nodes in the database) have a given property?



GraphAware®

Cartesian Products

In **set theory** a cartesian product is the result of enumerating all two-element pairs from both sets:

$$\begin{aligned}\{1, 2\} \times \{A, B, C\} &= \\ \{1, A\}, \{1, B\}, \{1, C\}, \\ \{2, A\}, \{2, B\}, \{2, C\}\end{aligned}$$

If (in Neo4j) you ask for unrelated patterns, then you can get a cartesian product returned which is slow.



Querying through dense nodes

Dense nodes are expensive:

- Looking up many relationships requires multiple db hits
- Going through a dense node will lead to a lot of nodes loaded into memory



GraphAware®

How to unlock the power of **Indexing**



GraphAware®

Indexes

- Indexes are created by default on labels.
- If you are doing lots of querying on a property, you should create an index manually

CREATE INDEX ON :Label (property)

- You can drop indexes as well

DROP INDEX ON :Label (property)



An Aside

- There is something called the ‘legacy’ index which is available from Java and REST APIs.
- **These are not available in Cypher**
- If you see reference to ‘legacy index’ you can probably ignore it - by Neo4j 3.0 they are mostly out of date.



No Indexes: Exercises

- Find a device with a Mac address (e.g. 0007815d3c) and see what TV shows they watch.
- DROP the index on the Mac address. How performant is this query?
- What happens if we don't use labels?



GraphAware®

The Tip

- **Put indexes on what you'll be querying frequently**



GraphAware®

More Exercises

Answer these questions, and think about the profiling, and performance:

- Which TV shows are watched by the most people?
- Which TV shows were watched by 20 to 30 people?
- Which other TV shows did these people watch?



Exercises



GraphAware®

First Data Queries

- How many people watched ‘El lobo de Wall Street’?
- What TV Programme did the most people watch?
- How many documentaries were there?
- What genres exist under the Categoria ‘Cine’?
- What kind of sports “Deportes” game did people most often watch?
- **Cannot answer yet:**
- What is the earliest that a show was produced with Isabella Rossellini in?
- What genre was Tom Hanks most frequently starring in?



Starting a Recommendation Engine

- Choose a user that has watched a lot of shows...
- **Content-based filtering**
 - What genre did the user most often watch? What are the most watched shows of this genre? Remember to filter out the ones the user has seen.
 - Which directors did the user most often watch? Can we do a similar thing.
- **Collaborative filtering**
 - Which users had the most shows in common with this user? Are there any common shows within these users?
 - I am more similar to a user if I watch a *less* popular show. Can we make a ‘similarity’ score based on this?



Day 5

Wrap-Up & Next Steps

Session 1	Share Results of Cypher Queries
Session 2	Commercials of Neo4j and Business Q&A
Session 3	Software Architecture with Neo4j
Session 4	Whiteboarding: PoC Architecture

Neo4j Commercials



GraphAware®

Where will you find

Neo4j in your Business



GraphAware®

Differing Versions and Neo4j Licensing



GraphAware®

Bootstrapping Neo4j with

GraphAware



GraphAware®

Architecture



GraphAware®

How to Communicate between **Neo4j & Your App**



GraphAware®

Means of **Extending Neo4j**



GraphAware®

Third Party Libraries in the **Graph Ecosystem**



GraphAware®

Extending your Application with **Polyglot Persistence**



GraphAware®

Day 6

Operations with Neo4j

Session 1

Practical: Development of Proof of Concept

Session 2

Operations and Neo4j

Session 3

Whiteboarding: Physical Architecture

Session 4

Operations



GraphAware®

Subtitle

Deployment Options



GraphAware®

Subtitle

Clustering



GraphAware®

Subtitle

Backups



GraphAware®

Subtitle

Monitoring



GraphAware®

Subtitle

Performance



GraphAware®

Day 7

Recommendation Engines

Session 1

Introduction to Recommendation Engines

Session 2

Hands-On: Playing with Reco

Session 3

Whiteboarding: Making a
Recommendation Engine

Session 4



Recommendations



GraphAware®

What are

Recommendations



GraphAware®

Recommendations in **Theory**



GraphAware®

How do we solve this problem

In Reco



GraphAware®

Day 8

Hands-On Development

Session 1

Session 2

Session 3

Session 4

Practical: Development of Proof of Concept



Day 9

Wrap-Up & Next Steps

Session 1

Practical: Development of Proof of Concept

Session 2

Demonstration of Product

Session 3

Questions & Answers

Session 4

