Cypher Sleuthing:

Tips and Tricks for Querying a Graph

Jennifer Reif
Email: jennifer.reif@neo4j.com
Twitter: @JMHReif
LinkedIn: linkedin.com/in/jmhreif
Github: GitHub.com/JMHReif
Website: jmhreif.com

Who Am I?

- Developer, Advocate
- Continuous learner
- Conference speaker
- Blogger
- Geek

Jennifer Reif
Email: jennifer.reif@neo4j.com
Twitter: @JMHReif
LinkedIn: linkedin.com/in/jmhreif
Github: GitHub.com/JMHReif
Website: jmhreif.com

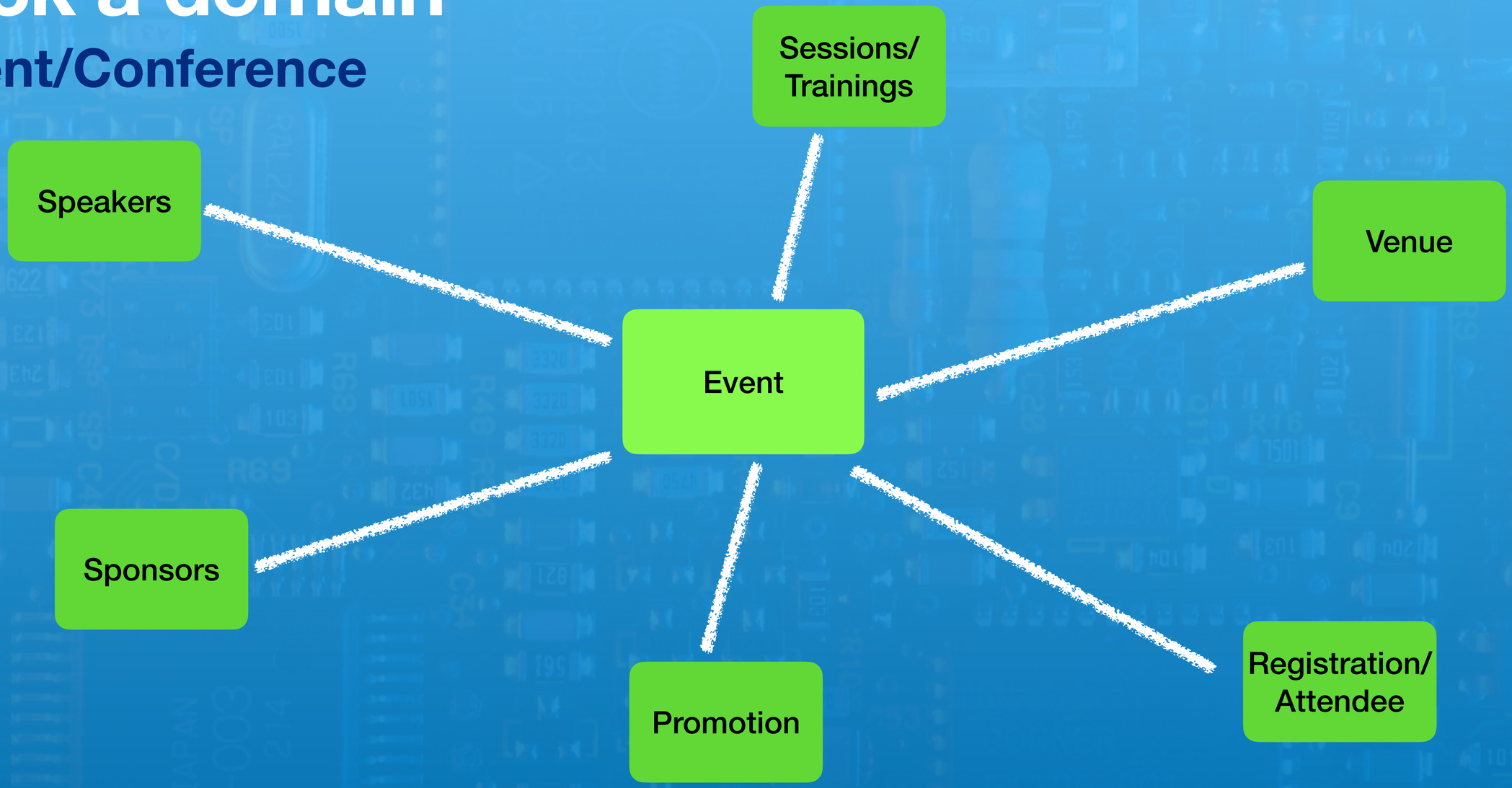
What is a graph?

Data storage with relationships!

- Stores relationships with entities
- Faster read queries
- Easily connect multiple entities together
- Mimic real-world data organization

Pick a domain

Event/Conference



Solution: why graph?

- **Understand** -> by relating known to unknown
- **Connect** -> disparate domains
- **Mirror** -> real-life domain
- **Increase** -> business agility
- **Reduce** -> complexity
- **Address** -> shortcomings in existing models

Photo by Javier Ezpeleta on Unsplash

Disclaimer!

- Graph is not a silver bullet
- Some use cases do not fit
  - That’s a separate session!

Photo by Julius Drost on Unsplash

Learn FAST!

And break things?

Tackling a solution…

- Trial-and-error
- At the mercy of the language
- Whack-a-mole fixes

- What if….

Photo by Michal Vrba on Unsplash

Understanding is the key

- How to wield the tool brings mastery
- Knowing when and how —>
  - Efficiency
  - Better results

Credit: unsplash.com

Cypher query language

What is Cypher?

- Graph query language
- Created by Neo4j ~10yrs ago
  - Open sourced
  - Widely supported
- Part of GQL standardization
  - opencypher.org

Photo by Eugene Chystiakov on Unsplash

What does it do?

- Follow path of connected data
- Help understand data patterns
- Surface hidden connections

Photo by ConvertKit on Unsplash

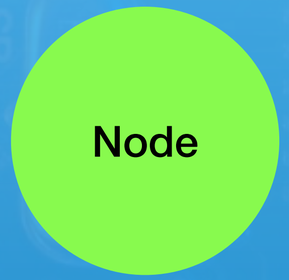
Quick lesson on Cypher syntax

Functional and visual

- Designed to store and retrieve data in a graph
- Based on ASCII-art
- Declarative query language
  - Focus on what to retrieve, not how


```
MATCH (A)-[:LIKES]->(B)
```

Cypher in 20 seconds…

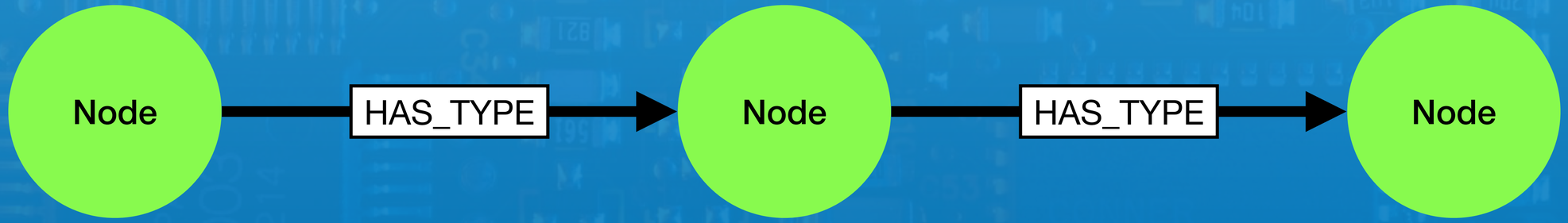


(var:Label)

(var:Label {key: value})

-[var:REL_TYPE]->

-[var:REL_TYPE {key: value}]->



(n1:Label)-[r1:REL_TYPE]->(n2:Label)-[r2:REL_TYPE]->(n3:Label)

Here’s what I’ve learned

Cypher CASE

Pitfalls (mine, at least)

- ✗ Logic like other languages (e.g. Java)
- ✗ Variable-setting
- ✗ Complex conditionals

Photo by Wander Fleur on Unsplash

Examples

```
//Bad syntax
LOAD CSV…AS row
MERGE (c:Company {companyId: row.Id})
WITH row, c,
 CASE row.type
  WHEN 'P' THEN row.type = 'Public'
  WHEN 'R' THEN row.type = 'Private'
  WHEN 'G' THEN row.type = 'Government'
  ELSE row.type = 'Other' END
SET c.businessType = row.type
RETURN *
```

```
//Bad syntax 2
LOAD CSV…AS row
WITH row,
 CASE row.BusinessType
  WHEN 'P' THEN type = 'Public'
  WHEN 'R' THEN type = 'Private'
  WHEN 'G' THEN type = 'Government'
  ELSE type = 'Other' END
RETURN row.CompanyId, row.CompanyName, type
```

Correct Logic

- 1. Expression comparison against multiple values

```
MATCH (p:Person)-[r:IS_MANAGED_BY]->(m:Manager)-[r2:OVERSEES]->(d:Department)
RETURN p.name,
 CASE p.role
   WHEN 'management' THEN d.departmentPhone
   WHEN 'business' THEN p.businessPhone
   WHEN 'technical' THEN p.emailAddress
   ELSE d.departmentEmail END as personContact
```

- 2. Multiple conditional statements expressed

```
MATCH (p:Person)
RETURN p.name,
 CASE
   WHEN dateHired is null THEN 'candidate'
   WHEN dateHired > date(‘2020-09-17') THEN 'newHire'
   ELSE 'employee' END as personStatus,
 CASE
   WHEN dateFired is null THEN dateHired
   WHEN dateHired is null THEN entryDate
   ELSE 'n/a' END as leadDate
```

In Action!

Cypher Temporals

Pitfalls

- Date format
  - ISO 8601
    - Date: 2022-02-24
    - Datetime: 2022-02-24T13:00:15Z
  - Pesky literal 'T'
- Durations
  - Precision
  - Groups

Credit: unsplash.com

Translating to ISO 8601

- Cypher accepts:
  - ISO 8601
  - Strings in ISO 8601 format
- What if you have ANYTHING else?
  - Error!

APOC to the rescue!

What is APOC?

- Utility library for Neo4j
- Widely used, broadly applicable
- 550+ procedures and functions

Photo by Alexis Fauvet on Unsplash

Epoch time

- Unix system date/time
  - Processes, logs, etc.
- Seconds since 1970-01-01T00:00:00Z
- Example: 1645729200 (end time for this session)
- Handled with apoc.date.format() or apoc.date.toISO8601()

https://www.epochconverter.com/

In Action!

Date Strings

- Process:
  - Temporal string -> ISO 8601 string -> Neo4j temporal
- Any temporal string with specified format
- Handled with apoc.date.convertFormat() or apoc.temporal.toZonedTemporal()

In Action!

Cypher Temporal Durations

Durations

- Distance in time measurements
- Literal ‘P’ and ’T’ in syntax
  - P3M5DT2H17M0S
- Groups - months, days, seconds
- Duration functions (between, inMonths, inDays, inSeconds)

Photo by Randy Fath on Unsplash

In Action!

Duration conversions

- Duration functions: inMonths, inDays, inSeconds
- Most components -> whole values only, no remainders
- Convert between units: functions + components
  - Component groups

Component group	Constituent components
Months	*Years*, *Quarters* and *Months*
Days	*Weeks* and *Days*
Seconds	*Hours*, *Minutes*, *Seconds*, *Milliseconds*, *Microseconds* and *Nanoseconds*

In Action!

Eager operator

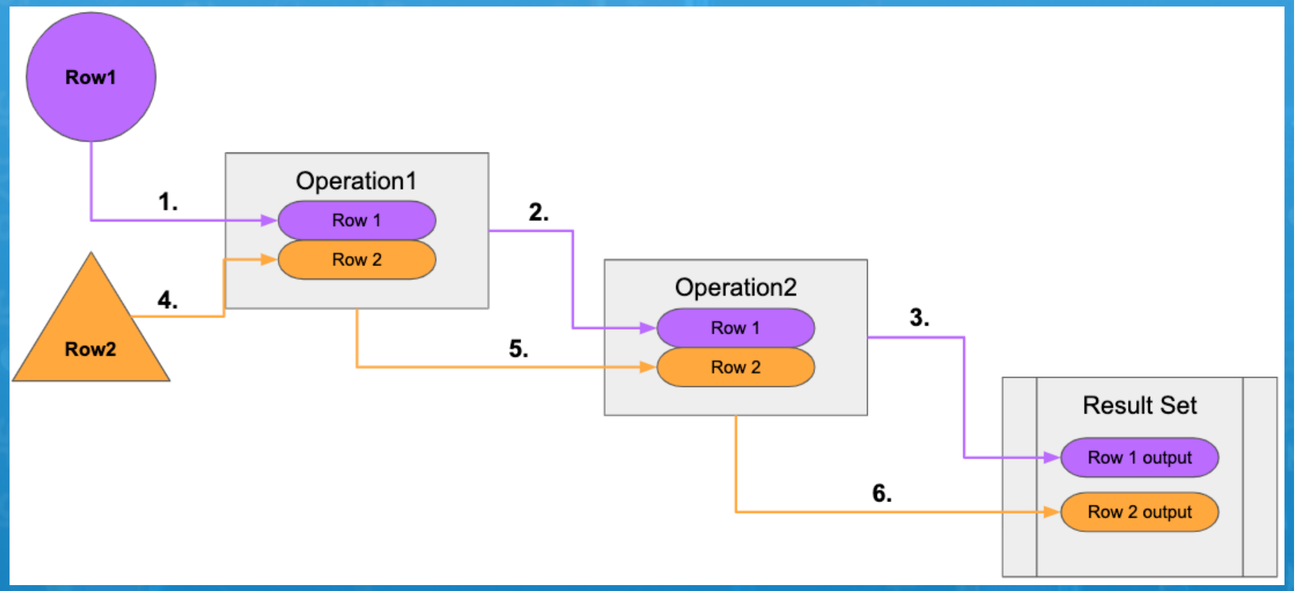
What is Eager?

- Eager loading
  - Consistency and conflicts
  - Operations occur to all rows before continuing
  - Avoid read/write conflicts

Credit: unsplash.com

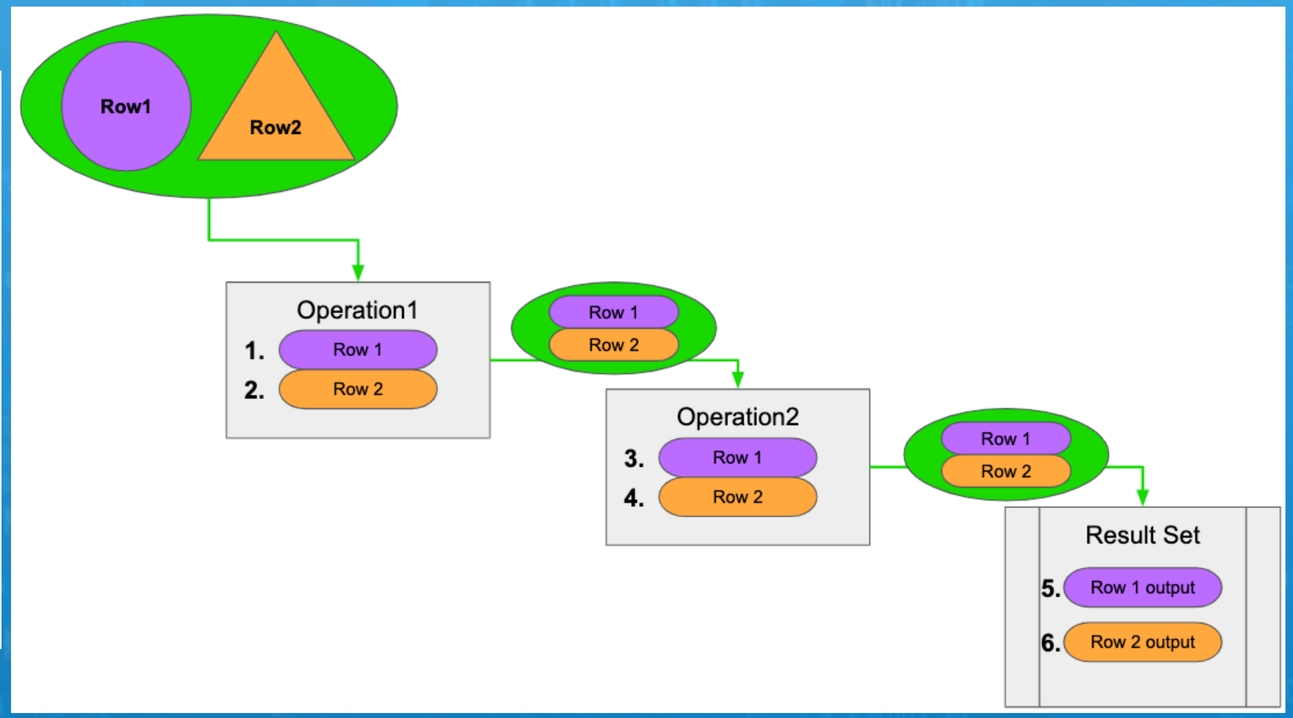
What's the difference?

Non-Eager



Row-by-row

Eager



Operation-by-operation

Does it matter?

- Low memory/heap
- Large dataset
- Query performance

Credit: unsplash.com

In Action!

How to avoid eager?

- Might not be an issue for smaller operations/data sets
- Separate operations
  - Avoids situations where conflicts might occur
- Use PROFILE/EXPLAIN on queries
- Having trouble? Ask for help!
  - dev.neo4j.com/forum -> Cypher channel

Recap!

- CASE statement
  - Not a programming language
  - Compare values OR multiple conditionals
- Cypher temporal types
  - Format, transformation
  - Durations and groups
- Eager operator
  - Separate operations
  - Inspect queries with PROFILE/EXPLAIN

Resources

- Blog posts: CASE, temporals, eager, and more!
  - jmhreif.com/blog/
- Repository:
  - github.com/JMHReif/cypher-sleuthing
- Ask for help!
  - dev.neo4j.com/forum -> Cypher channel

Credit: unsplash.com

Jennifer Reif
Email: jennifer.reif@neo4j.com
Twitter: @JMHReif
LinkedIn: linkedin.com/in/jmhreif
Github: GitHub.com/JMHReif
Website: jmhreif.com