THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

All the Databases!

Let's Discuss them all!

Amanda Kay Moran

#ossummit @AmandaK_Data

THE LINUX FOUNDATION

Agenda

Some Folks don't like Agendas, say it's too basic, but I do

- Basics and Database History
- Relational Databases
  - Postgres and MySql
- NoSQL Databases
  - Apache Cassandra and FoundationDB
- Graph Databases
  - Apache Tinkerpop
- Next Steps

Who is Amanda?

- MS in Computer Science
  - BS in Biology
- Worked in Silicon Valley for 8 years
  - Many different companies big and small
- Has worked on 5 different Databases
  - 3 Proprietary databases
  - 2 Open Source databases (Apache Trafodion)
  - And 2 different distributed systems
- Udacity Data Engineering: Data Modeling

Why Do I Care About This?

- The technology is interesting!
  - Stanford Databases by Professor Jennifer Widom
- It's all about the Data
  - Where you data is persisted
  - Where you data is analyzed
  - How data is quickly served to you
- Your job depends on it!
  - Okay, that's a little dramatic… but it's true
- Your applications depend on it!

History of Databases

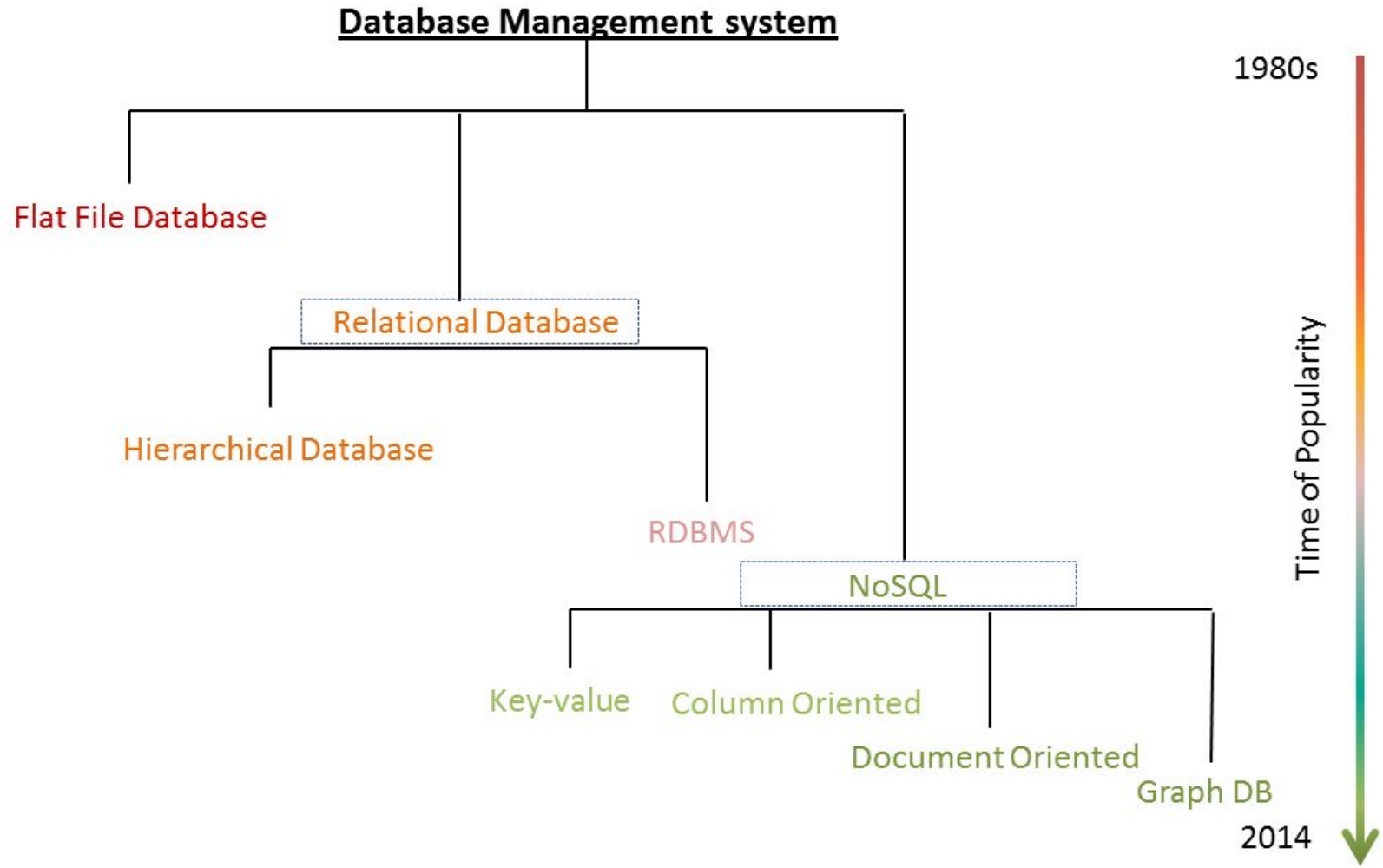


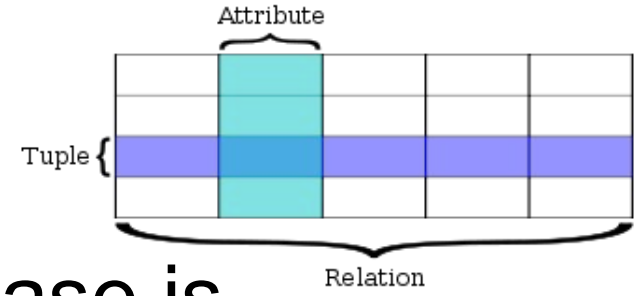
Photo Credit:
Analytics Vidhya

Focused on Open Source

- Open Source has been proven to have won
- Open Source communities
  - Contributions
  - Training
  - Tutorials
  - Docs
  - Mailing lists
  - All free :)

Relational (RDBMS)

Relational Databases



- Codd's 12 rules
  - Rule 1: All information in a relational database is represented explicitly at the logical level and in exactly one way by values in tables
- Must have ACID Transactions
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Uses SQL as its primary query language

```
SELECT * FROM myCoolTable WHERE conference = virtual
```

Relational: Examples

- PostgreSQL
  - Developed by PostgreSQL Global Development Group
  - Not just one company behind it
  - Object Relational Database
- MySQL
  - Bought by Sun Microsystems → Oracle
  - Multiple forks after the Oracle
  - More popular than PostgreSQL
    - 2019 Report: 39% of Developers use MySQL
  - More 3rd party tools

A Word on OLAP vs OLTP

- OLAP(Online Analytical Processing)
  - Complex analytical and ad-hoc queries optimized for reads
  - Not high writes (data is normally loaded in batches)
  - Lots of JOINS
- OLTP (Online Transactional Processing)
  - Less complex queries but many
  - Read, insert, update, delete

Relational: When to Use

- SQL
- Ability to do JOINS, aggrelations, analytics
- Smaller Data (not Big Data)
- Need flexibility in your queries
  - Lots of ad-hoc queries
- You need ACID transactions
  - Need consistent data
- Simplicity

Relational: When *Not* to Use

- Large Amounts of Data
- Need High Availability
  - Single point of failure → Need to hot swap
- Need Higher Read Performance
  - ACID is great, but it slows you down
- Need flexibility in schemas
  - Ability to add columns only for rows that need it
- Ability to store different types of data formats

NoSQL

NoSQL

- NoSQL a reaction to limitations of RDBMS

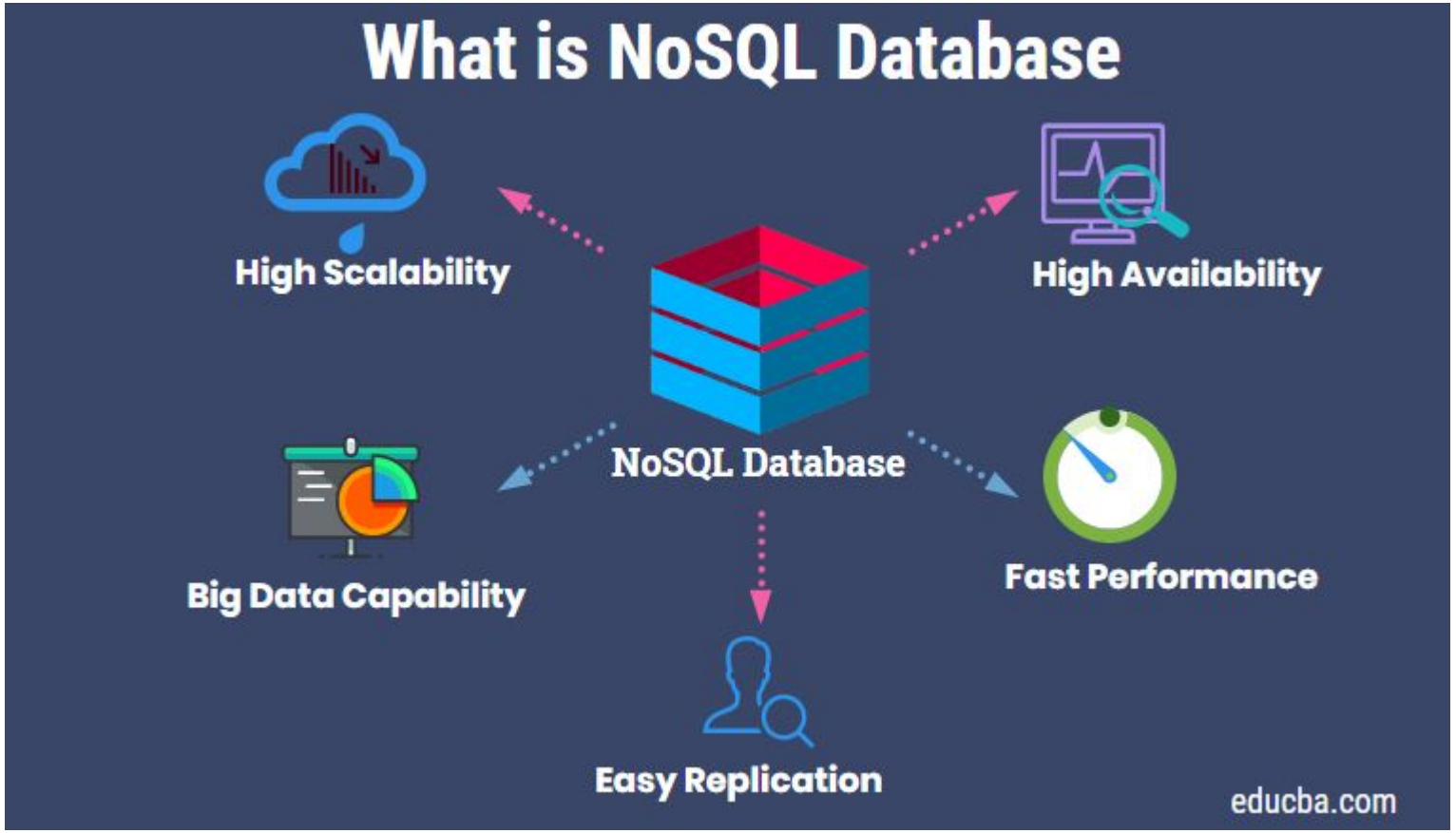


Photo Credit:
educba: What is a NoSql Database

NoSQL

- Data is not necessarily in tables
- NoSQL
  - Not Only SQL
  - Non Relational
- Many different types with different strengths
- Different data structures and modeling allow for faster operations
- Cloud Native

NoSQL

- Document
  - MongoDB
- Key Value
  - FoundationDB
- Column Family
  - Apache Cassandra

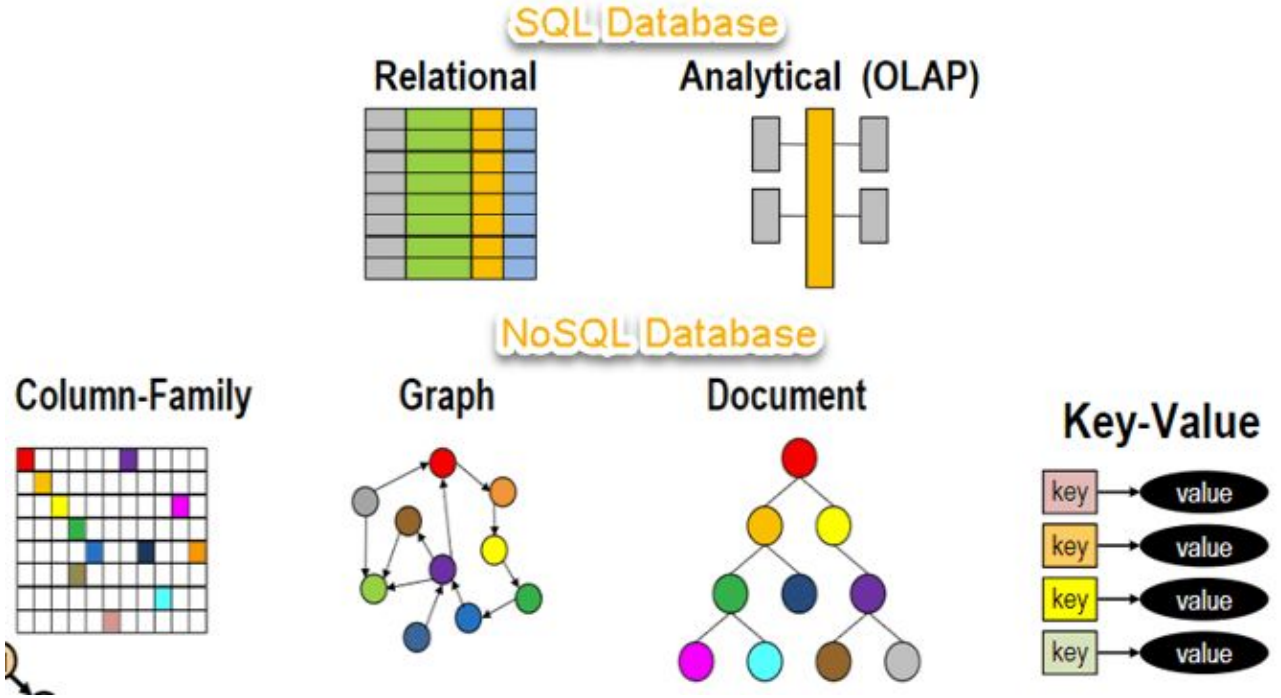


Photo Credit: Guru99

NoSQL

```
SELECT * FROM myCoolTable
```

- Each have own query language
  - And support for many drivers (Python, c++, etc)
- MongoDB

```
db.myCoolTable.find( {} )
```

- Apache Cassandra

```
SELECT * FROM coolTable WHERE state = CA
```

- FoundationDB

```
tr = db.create_transaction()
myCoolTable.unpack(k)[0] for k, v in tr[]
```

NoSQL Examples: Apache Cassandra

- Donated to the Apache Foundation
  - 10 years old
  - Supported by many different companies
- Leaderless architecture
  - High availability, easy to scale, fast reads and writes
- Uses CQL
- All the big apps use Apache Cassandra

NoSQL Examples: FoundationDB

- Key-value Database
- Open Sourced by Apple after acquisition
- Does not have a query language API instead
  - Little difficult at first
- Has a layered architecture
  - Core
  - Layers of functionality on top
- Supports ACID transactions!

FoundationDB

THE LINUX FOUNDATION

NoSQL: When to Use

- Need High Availability
- Big Data
- Need Linear Scalability
- Low latency
- Need fast reads and writes
- Flexibility with schema
- Distributed users
- Know queries in advance (applications)

NoSQL: When *Not* to Use

- Need to use SQL (ways around this)
- Need ACID transactions
  - And don't want to use FoundationDB
- Need to be able to JOIN tables
- Need flexibility
- Ability to do Ad-hoc queries
- Have small data -- don't need the headache!

NoSQL: A Word of Warning

- Beware when moving from RDBMS to NoSQL
- Can’t not be moved over as-is
- More of a learning curve
- Think queries and applications first!

Graph

Graph Databases

- Wikipedia: “uses graph structures for semantic queries with ***nodes*** and ***edges*** and properties to represent and store data”

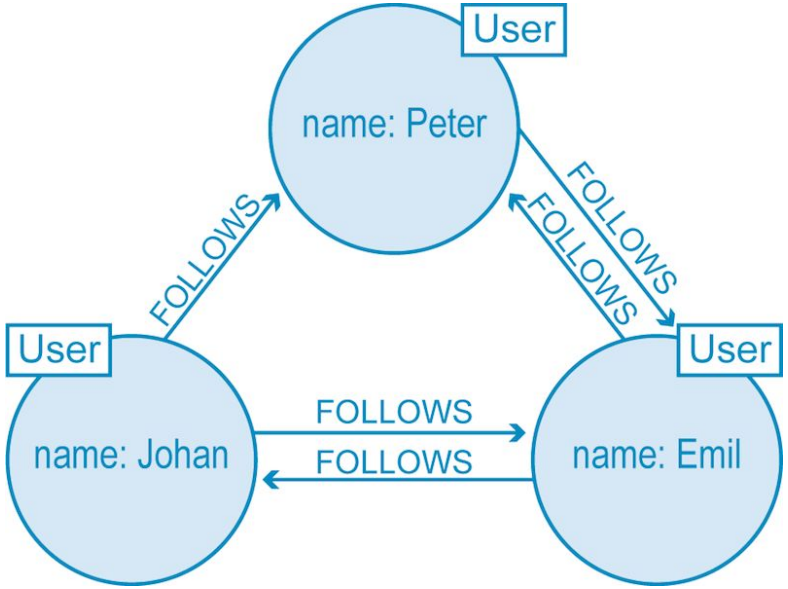


Photo Credit: Neo4j

Graph Databases

- The key is the relationships between the data
- Dependencies between data is clear
  - Not clear in Relational or other NoSQL
- Fast way to query and retrieve data (traverse)
- Uses Gremlin query language

Photo Credit: Neo4j

Graph Databases

- SQL

```
SELECT * FROM myCoolTable WHERE conference = ossumit
```

- Gremlin

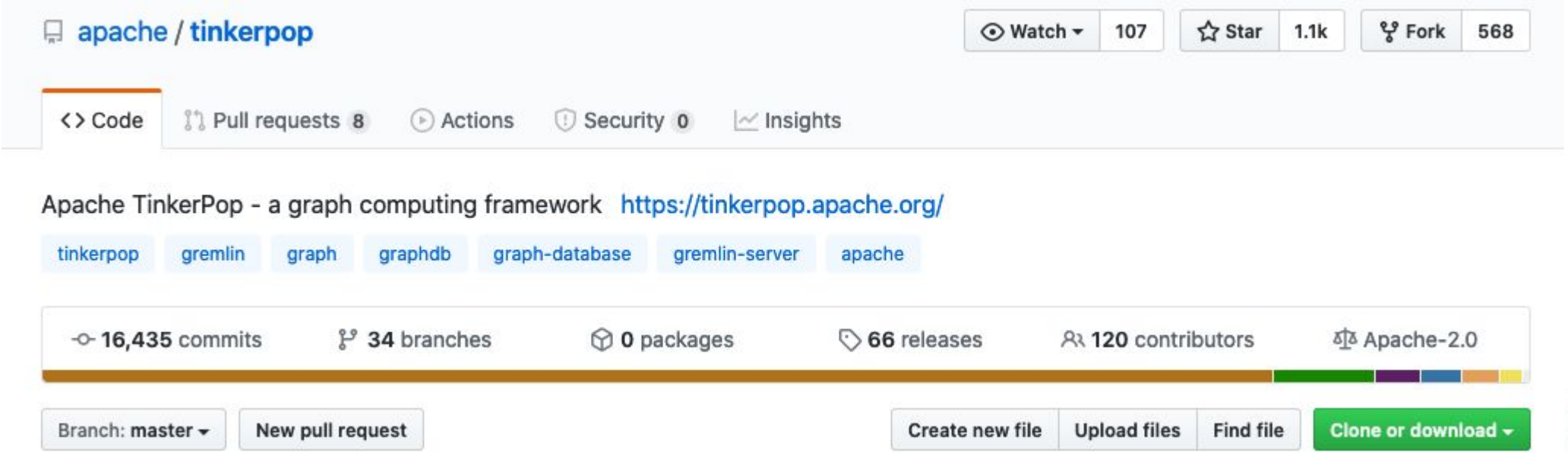
```
myCoolGraph.V().hasLabel('ossumit')
```

- Complex SQL queries can be reduced to very simple Gremlin queries

Photo Credit: Neo4j

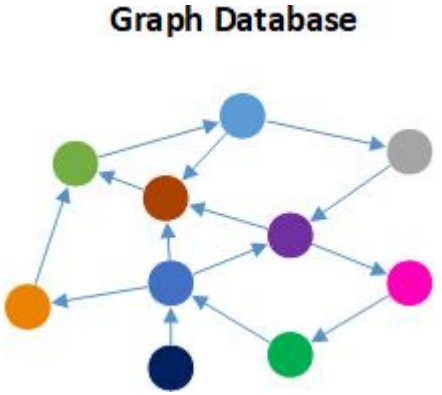
Graph Databases: Apache Tinkerpop

- Started in 2009 in Los Alamos National Lab
- Graduated to Top Level Project 2016
- Active community
- Training with Gremlin



Graph Databases: When to Use

- When trying to understand relationships
- Need better performance (for complex JOINs) can "walk" the graph instead
- Try the whiteboard test
  - Does the data naturally fit in a graph?
- Great article and talk: [Neo4J: GraphDB vs RDBMS]



Graph Databases: When to *Not* Use

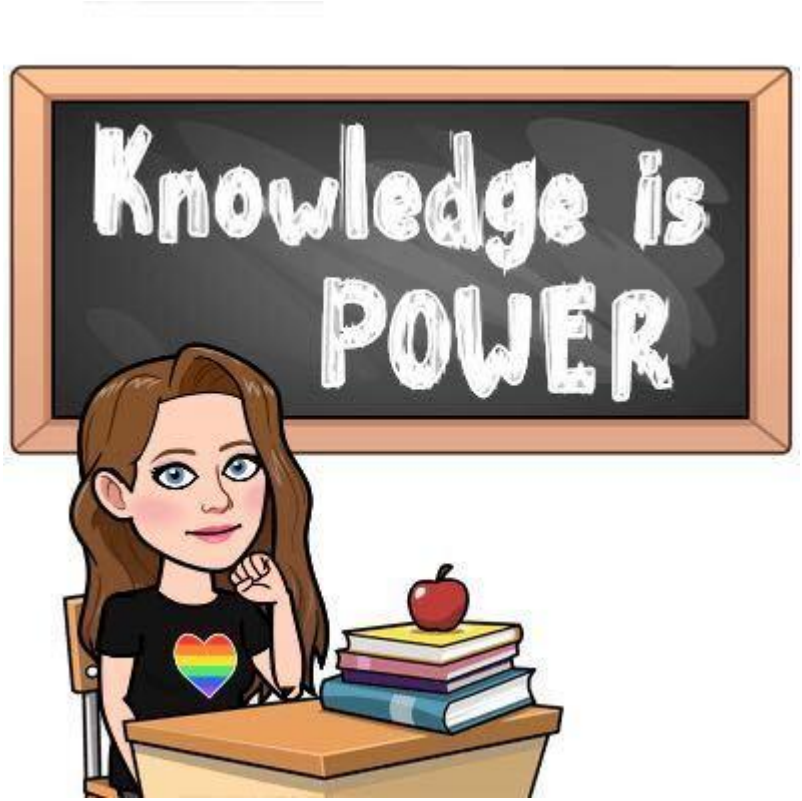
- Disconnected Data -- or relationships not important
- Write Heavy Workload
- Using it at a Key-Value store
- Data is not from a known point
  - Can’t walk the graph if don’t know where to start
- Overhead in creating graph (adding edges)

Relational vs NoSQL vs Graph

- Not either or
  - All organizations have both and multiples of each
- Each have benefits and drawbacks
  - Get informed before choosing
- All are easy to explore from your laptop
  - Benefits of Open Source
- Consider managed platforms based on oss
- Consider your uses cases
  - Reach out to other groups
  - Reach out to message boards/email lists

What to Do Next?

- Keep learning! Get hands on!
- PostgreSQL
  - Wiki
- MySQL
  - Developer Zone
- Apache Cassandra
  - Documentation
  - DataStax Academy
- FoundationDB
  - Tutorials
- Apache TinkerPop
  - Documentation



Thank you!!

THE LINUX FOUNDATION

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

NORTH AMERICA