SQL & No-SQL (Hadoop?)

- Big Data Adoption & Success in the Enterprise

MYTHS ABOUT HADOOP

Myth	Actual
Hadoop Is A Single Product	Hadoop is an ecosystem, not a single product
Apache Hadoop open source	Multiple vendors have implementations of Hadoop as well to add usability tools, ease of use functionality
Hadoop is a Data management system	Hadoop is actually file-system based
Hive resembles SQL	Is not standard SQL. Instead, Hadoop uses Apache Hive and HiveQL, a SQL-like language
Hadoop requires MapReduce	Hadoop can survive by itself. They complement each but are not required

MYTHS ABOUT HADOOP - II

Myth	Actual
Hadoop and Big Data are synonymous	Hadoop isn’t the only answer. products from Teradata, Sybase IQ (now owned by SAP) and Vertica (now owned by Hewlett-Packard).
Hadoop is used for Data Volume	Hadoop is for data diversity, not just data volume. Go back to the V^3 or V^4 (Volume, Variety, Velocity, Veracity)
Hadoop replaces the Data Warehouse	It is a complement to the data warehouse, another tool in the arsenal, not a replacement
Hadoop is for Web Analytics	Hadoop enables many types of analytics, not just Web analytics
Big Data Requires Hadoop	Big data does not require Hadoop, there are other options

SQL DATABASES & NOSQL

Traditional OLAP/OLTP Limitations:

1. E.g., A structured database needs to know what is being stored in advance.
2. The Agile development approach does not work well. Each time new features are added, the schema of the database requires changes.
3. If the database is large, the process is slow.
4. Rapid iterations and frequent data changes result in frequent downtime.

SQL VS. NOSQL

5. Relational databases are not designed to cope with the scale and agility challenges of modern applications. They are not built to take advantage of today’s available cheap storage and processing power.

NOSQL ADVANTAGES

1. NoSQL databases allow insertion of data without a predefined schema.
2. Application changes in real-time are easier, resulting in faster development.
3. Code integration is more reliable, and less database administration is needed.
4. NoSQL provides the ability to handle a variety of database technologies. It was developed in response to handling volume of data, frequency in which this data is accessed, performance and processing needs.

NO-SQL ADVANTAGES - II

5. NoSQL databases are more scalable and provide superior performance. The data model addresses issues not addressed by traditional relational model databases such as:

- Ability to handle large volumes of structured, semi-structured, and unstructured data
- Ability to handle Agile sprints, quick iterations, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture.

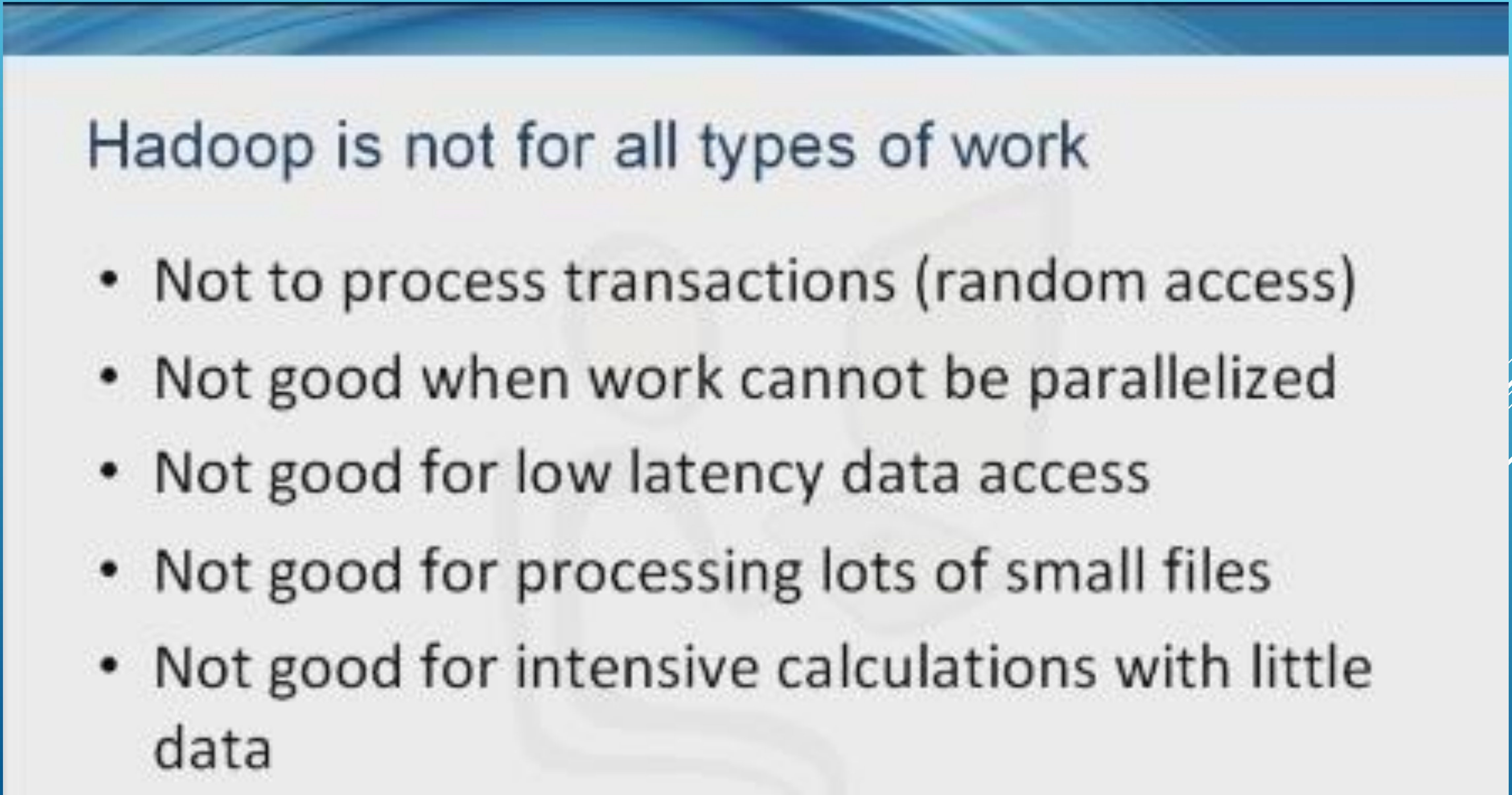
WHEN TO USE BIG DATA TOOLING

- Users want to interact with their data: totality, exploration and frequency. *Totality* refers to the increased desire to process and analyze all available data, rather than analyzing a sample of data and extrapolating the results.

However:

- Apache Hadoop does not replace the data warehouse and NoSQL databases do not replace transactional relational databases.
- Neither do MapReduce, nor streaming analytics Hive—Apache’s data warehousing application is used to query Hadoop data stores

WHEN NOT TO USE HADOOP



Hadoop is not for all types of work

- Not to process transactions (random access)
- Not good when work cannot be parallelized
- Not good for low latency data access
- Not good for processing lots of small files
- Not good for intensive calculations with little data

SAMPLE TOOLING USAGE -- MONGODB

Boxed Ice is an example of a company working with big data technology— a NoSQL database called MongoDB.

Headquartered in London, Boxed Ice offers a hosted software product called Server Density that monitors the health of cloud computing deployments, servers and websites for about 1,000 clients around the globe—a task that requires copious amounts of data processing.

> “We monitor quite a few websites and servers for customers like EA, Intel and *The New York Times*,” said David Mytton, Boxed Ice’s CEO and founder. “We are processing about 12 terabytes of data every month with MongoDB, and that equates to about 1 billion documents each month.”

ZION'S BANCORPORATION SOLUTION

- Zion's Bancorporation (Zion's), launched a fraud analytics program nine years ago. Cracking the big data code is a moving target requiring advanced technology and keen intellect.

Problem Statement

1. With exponential growth of data volume over the last decade, finding needles of useful information in haystacks of data was a formidable task.
2. Data needed to be captured while in motion, in order to be truly effective.

Solution Approach

- Used to handle "big data" challenge for fraud analytics: Zion used non-traditional data tools such as Hadoop, NoSQL Database, MapReduce for Analytics, for defining big data computing.

Outcome

- Zion's bank fraud and security analytics team has continually built and refined statistical models that have helped bank executives predict, identify, evaluate and, when necessary, react to suspicious activity

Characteristics of NoSQL

non-relational

open-source

cluster-friendly

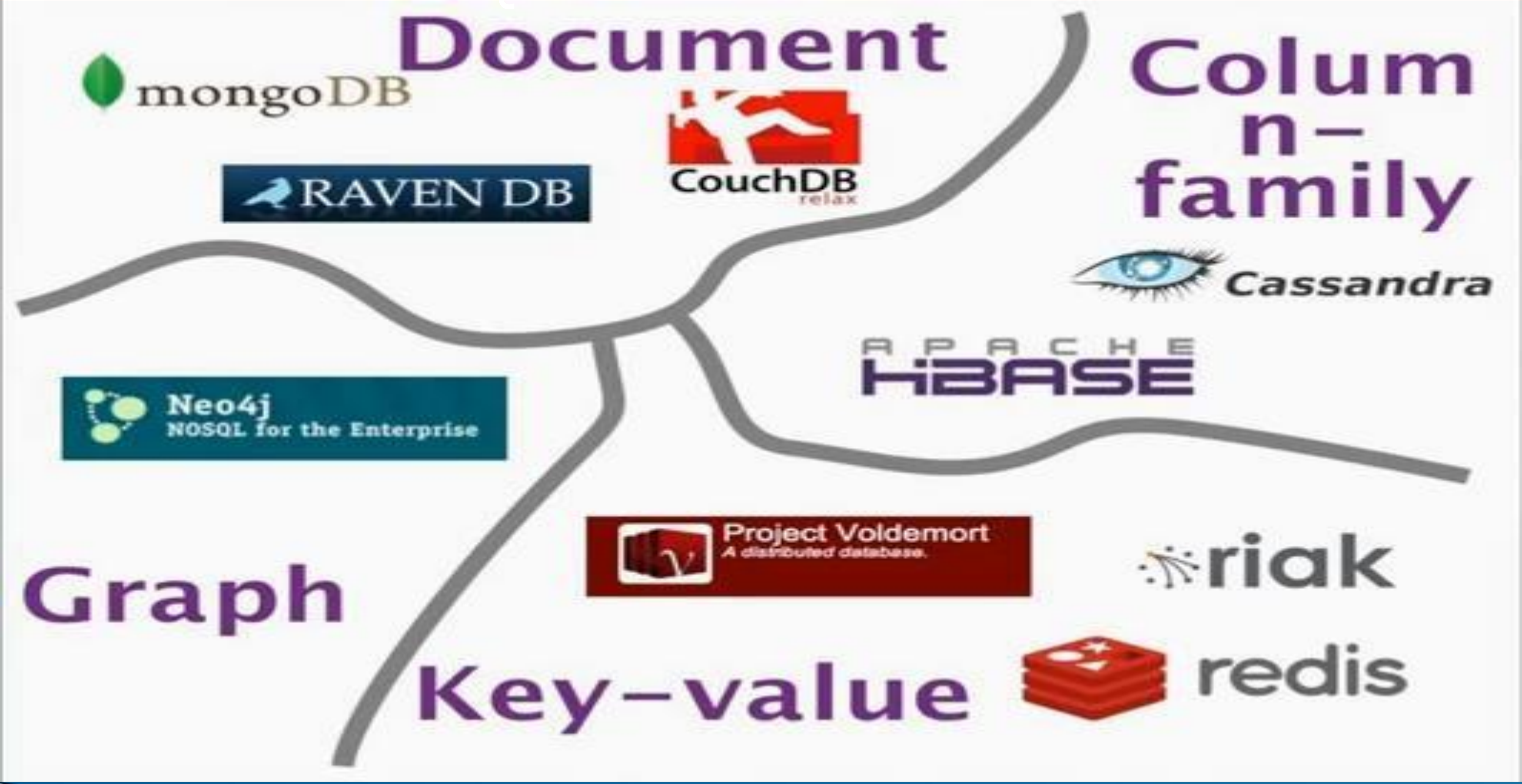
21st Century Web

schema-less

WHY DIFFERENT TYPES OF NOSQL DATABASES?

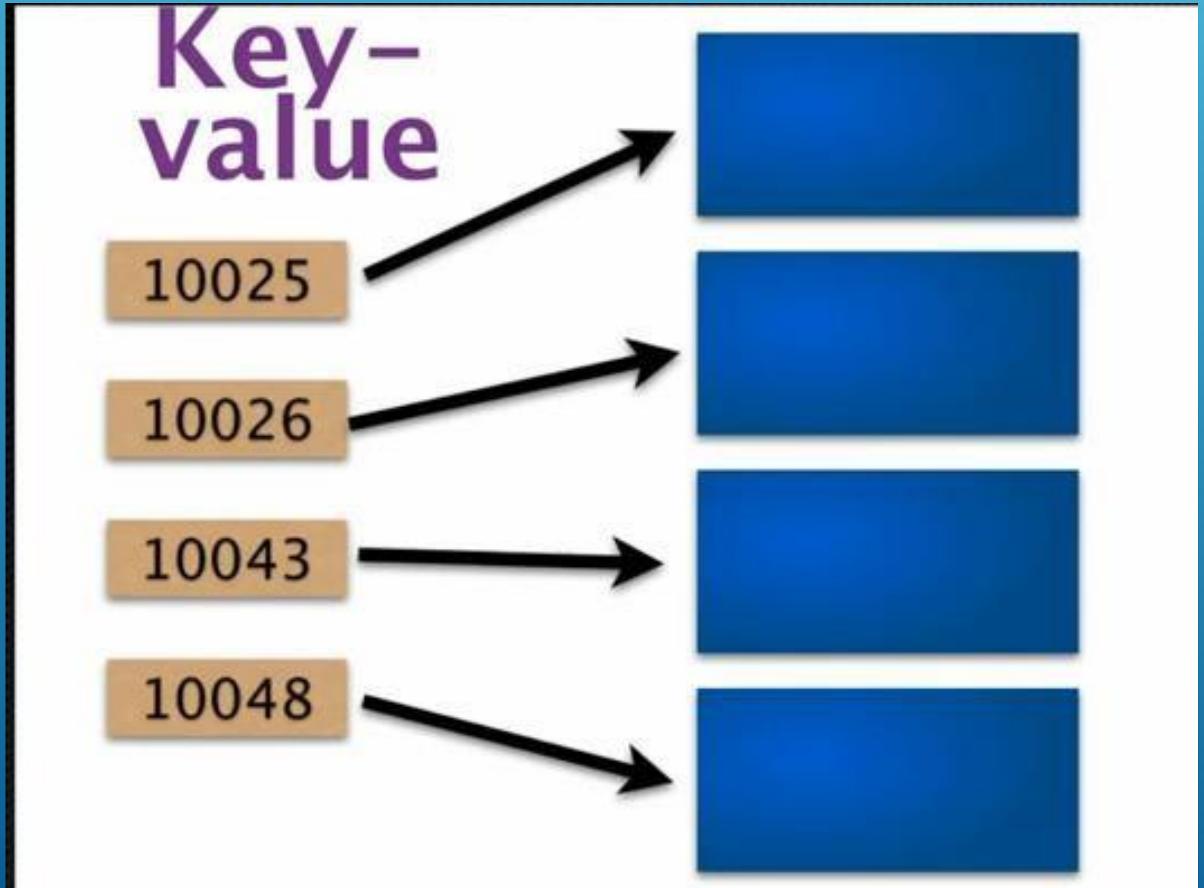
1. CAP theorem, aka Brewer's Theorem.
   You can provide only two out of the following three characteristics: **consistency**, **availability**, and **partition tolerance**.
   - Different datasets and different runtime rules require trade-offs.
   - Different database technologies focus on different trade-offs.
   - The complexity of the data and the scalability of the system also come into play.
2. Basic computer science or even more basic mathematics.
   - Some datasets can be mapped easily to key-value pairs. I.e., Flattening the data does not make it less meaningful. No reconstruction of relationships is necessary.
3. There are datasets where the relationship to other items of data is as important as the items of data themselves.

SAMPLE NO-SQL DATABASES BY DB TYPE



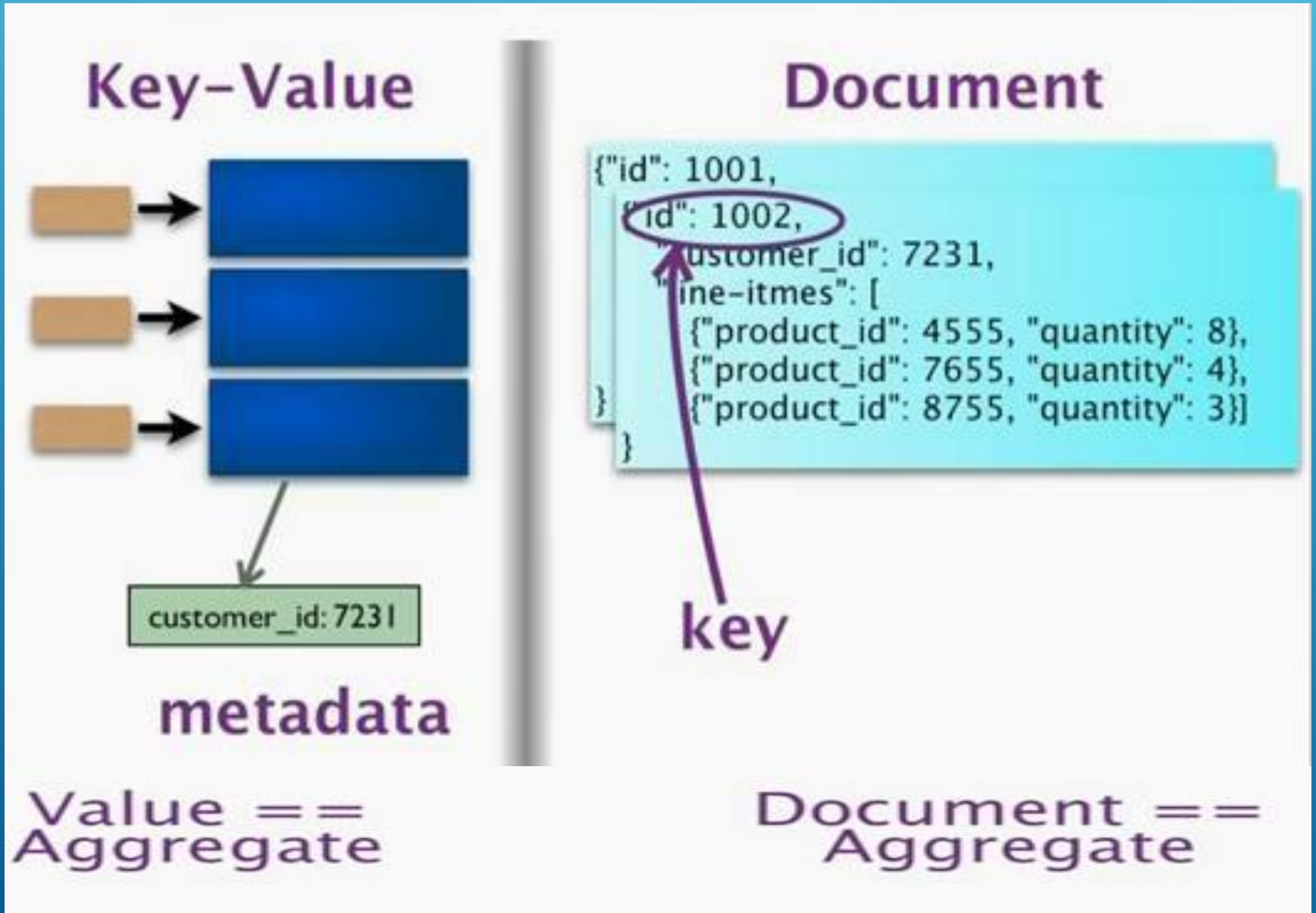
NO-SQL (AGGREGATE) DATA MODEL: KEY VALUE

Key Value are the simplest NoSQL databases. Every item in the database is stored as an attribute name (or "key"), together with its value. Some key-value stores, such as Redis, allow each value to have a type, such as "integer", which adds functionality.



NO-SQL (AGGREGATE) DATA MODEL: DOCUMENT

Pairs each key with a complex data structure known as a document. Documents can contain many different key-value pairs, key-array pairs, or even nested documents.



```
{"id": 1001,
  {"id": 1002,
   "customer_id": 7231,
   "line-itmes": [
     {"product_id": 4555, "quantity": 8},
     {"product_id": 7655, "quantity": 4},
     {"product_id": 8755, "quantity": 3}]
}
  }
```

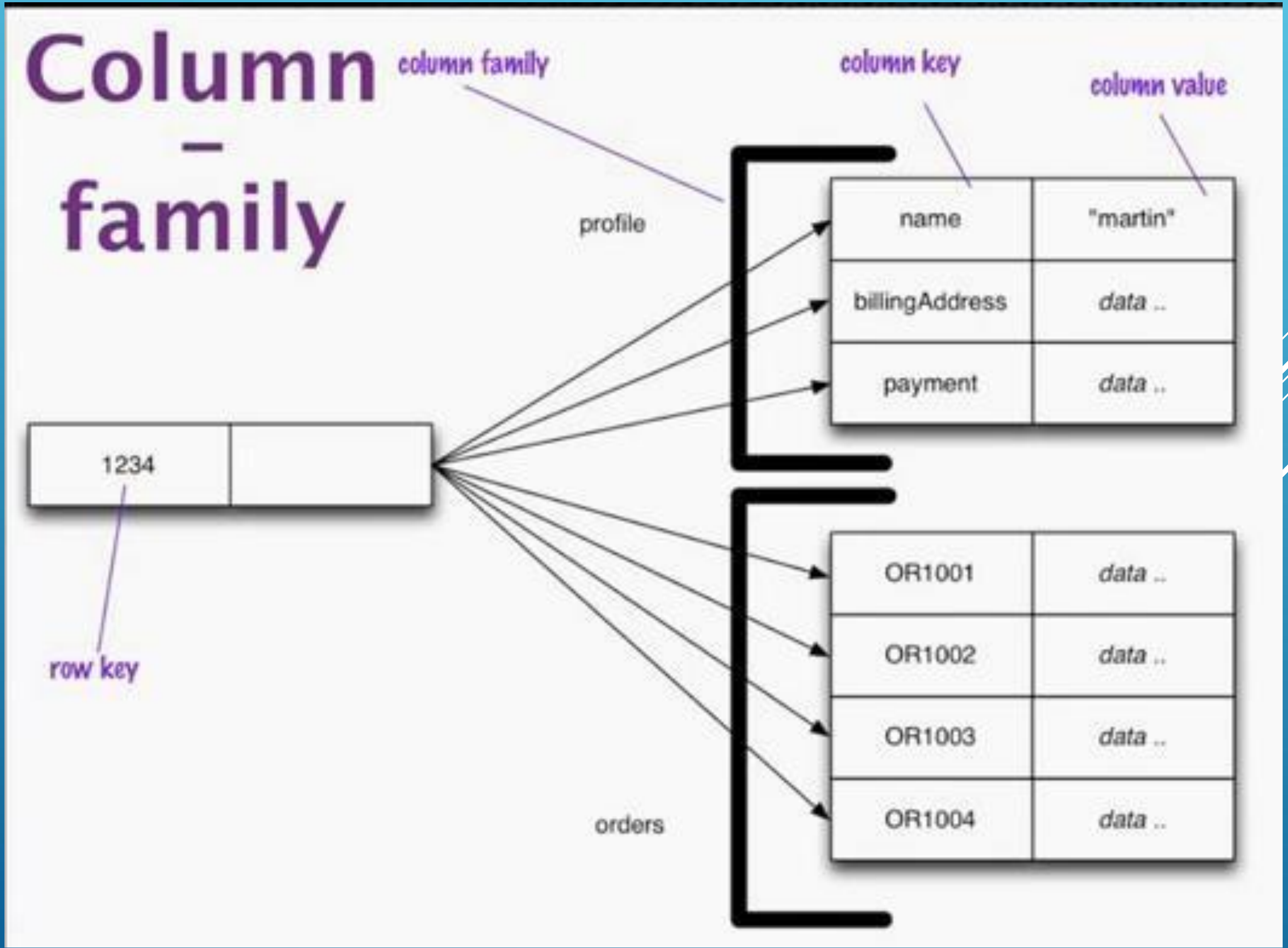
Value == Aggregate

Document == Aggregate

NO-SQL (AGGREGATE) DATA MODEL: COLUMN N FAMILY

Wide-Column stores are optimized for queries over large datasets, and store columns of data together, instead of rows. A hash map crossed with a multidimensional array. Each column contains a row of data ...

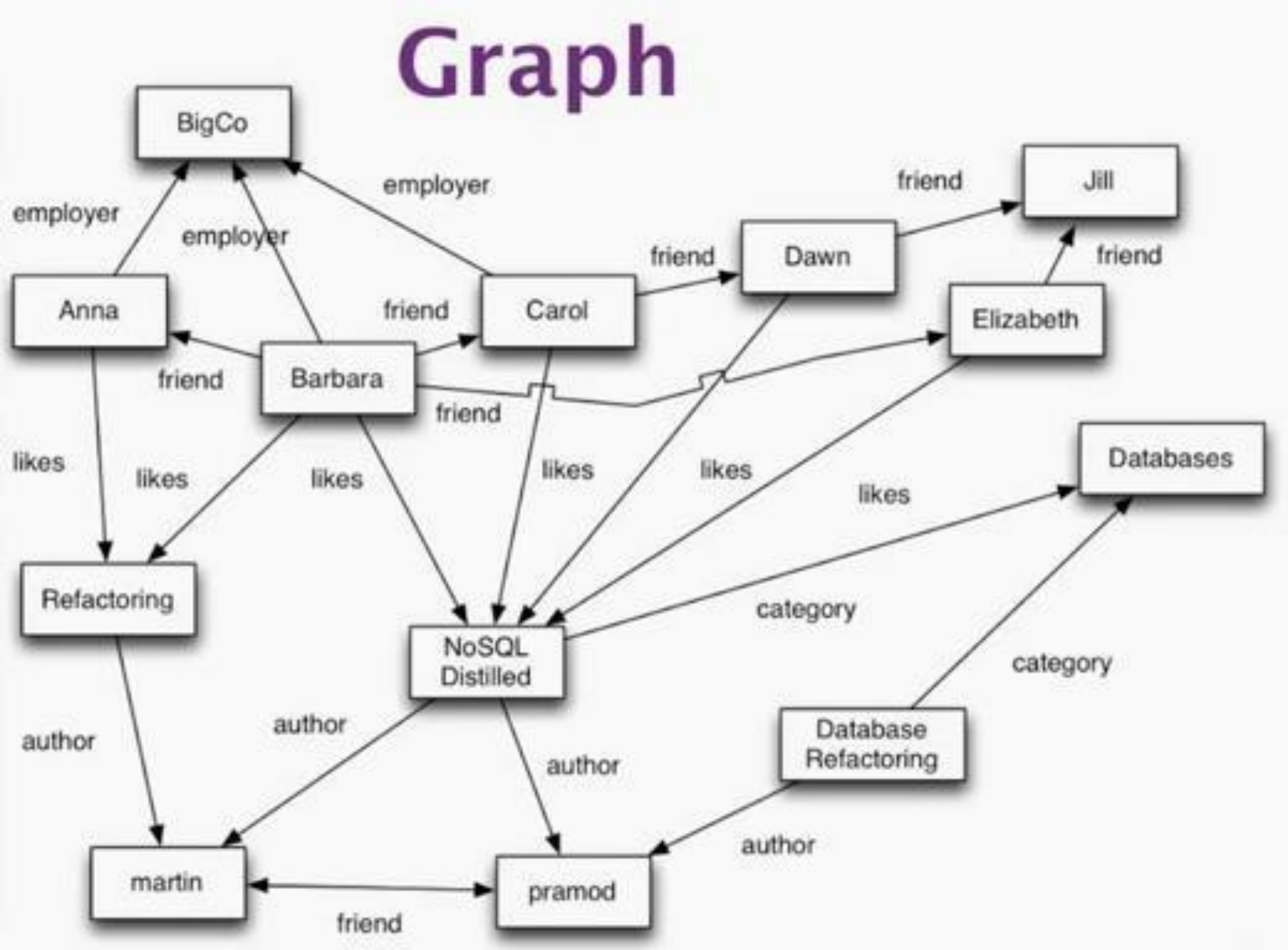
Eg., Row Key, Grouped by a row key, a customer profile, and orders for the customer



NO-SQL DATA MODEL: GRAPH STORES

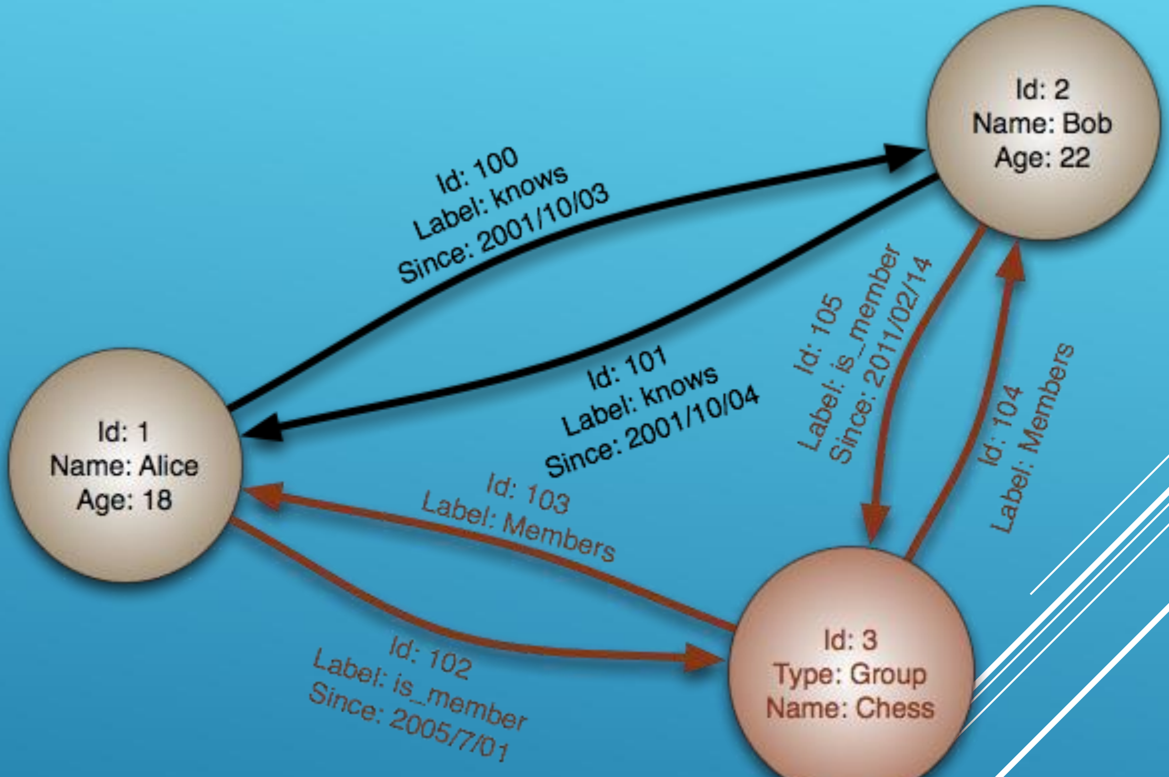
Used to store information about networks, such as social connections. Graph stores include Neo4J and HyperGraphDB. Outlier.

- Complex inter-relationships
- Not easily handled in multi-table joins
- This model is very good for managing across relationships. Very good at moving across relationships between things.
- Just like SQL databases, Graph databases do ACID. Other noSQL data models do not use the ACID concept, however, Graph databases do.



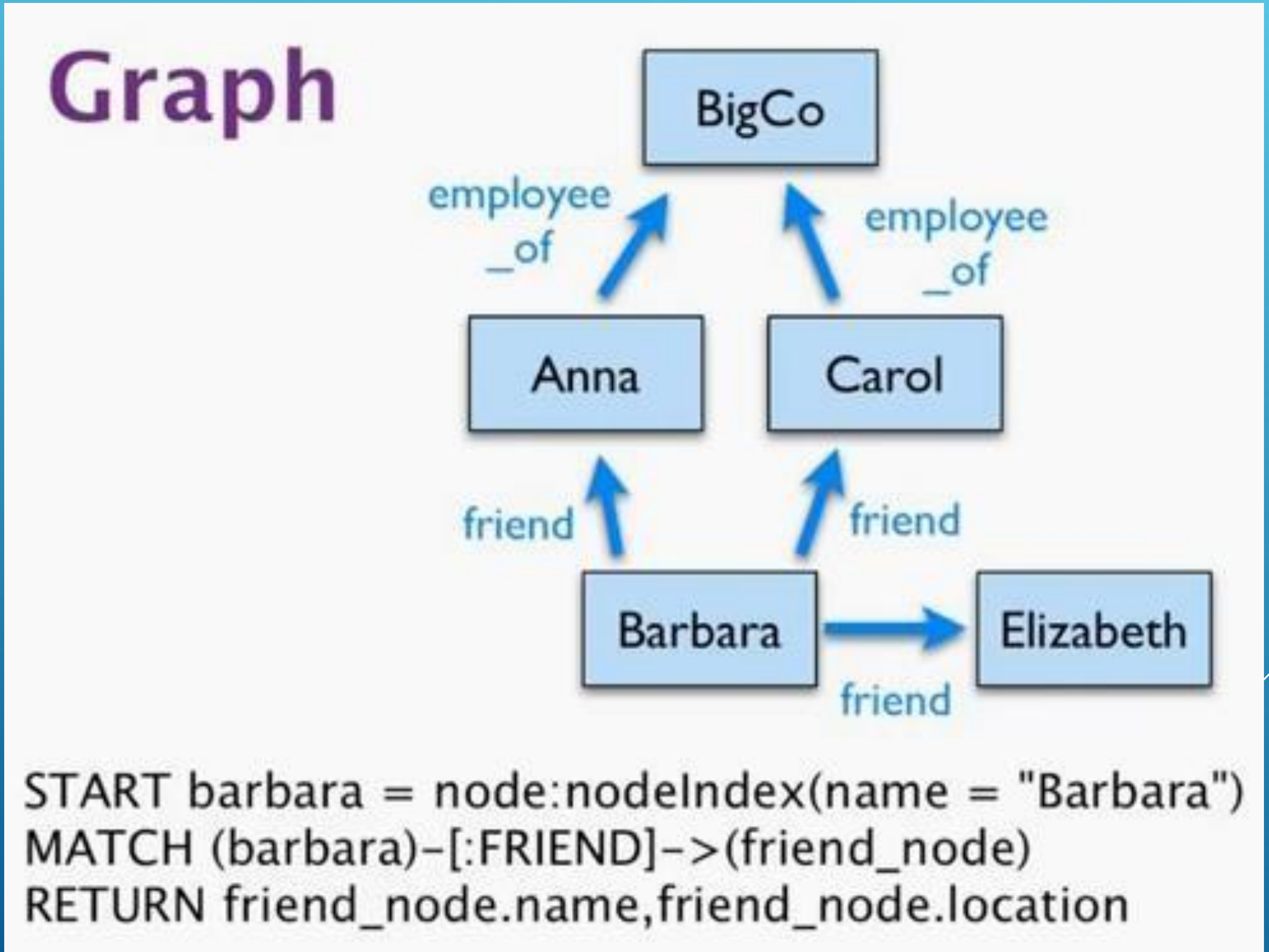
GRAPH DATABASES - EXPLAINED

- Graph databases are based on graph theory. Graph databases employ nodes, properties, and edges.
- Nodes represent entities such as people, businesses, accounts, or any other item you might want to keep track of.
- Properties -- relate to nodes. E.g., "Wikipedia" as one of the nodes, might tie to properties such as "website", "reference material", or "word that starts with the letter 'w'", depending on aspects of "Wikipedia" pertinent to the particular database.[25]



GRAPH DATABASES

- Able to do some interesting queries with Graph databases.

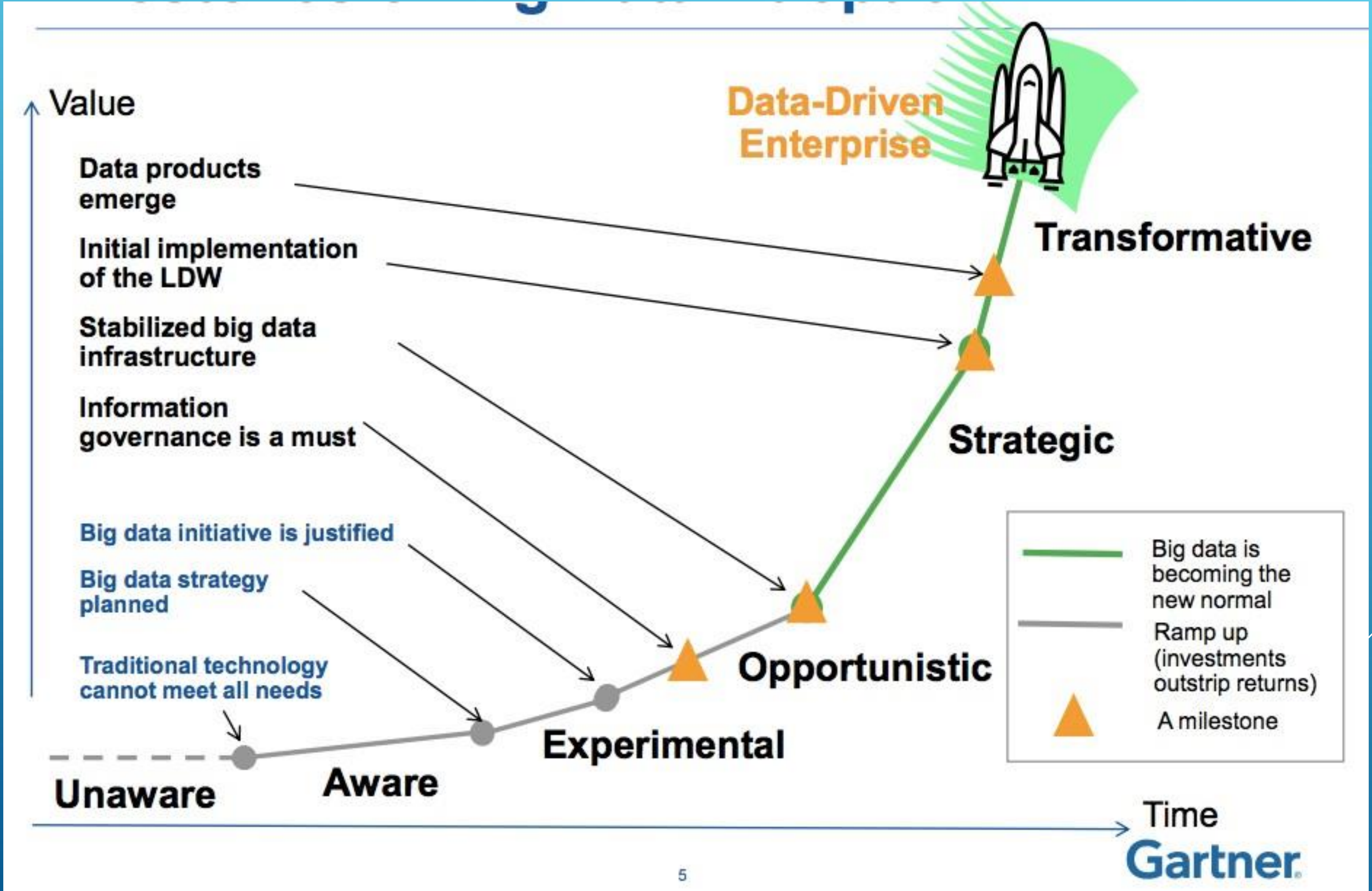


Graph

```
START barbara = node:nodeIndex(name = "Barbara")
MATCH (barbara)-[:FRIEND]->(friend_node)
RETURN friend_node.name,friend_node.location
```

Enterprise Roadmap & Defining Success

THE ROADMAP: TYPICAL STAGES & MILESTONES TO BIG DATA ADOPTION IN THE ENTERPRISE


Value
Data products emerge
Initial implementation of the LDW
Stabilized big data infrastructure
Information governance is a must
Big data initiative is justified
Big data strategy planned
Traditional technology cannot meet all needs
Data-Driven Enterprise
Transformative
Strategic
Opportunistic
Experimental
Aware
Unaware
Big data is becoming the new normal
Ramp up (investments outstrip returns)
A milestone
Time
Gartner
5

DEFINING SUCCESS IN A BIG DATA PROGRAM

- According to Zions and others who use Hadoop, NoSQL databases and similar tools that have come to define the era of big data computing.
- Achieving a great return on investment (ROI) is about
  1. Creating the right team
  2. Putting a solid business strategy in place
  3. Being agile and testing—lots and lots of testing

BIG DATA – WHAT DEFINES SUCCESS II

1. Organizations evaluating big data technologies should remember to test them in conjunction with their own data sets or applications, as opposed to using some arbitrary data set or app a salesperson has on hand.
2. Getting a Big Data environment to work right requires testing, tuning, keeping up with documentation and paying attention to what the open source community has to say.
3. With the right team and strategy in place, big data technologies like Hadoop, Hive, Pig, Cassandra, Mahout and others can open up a world of predictive possibilities