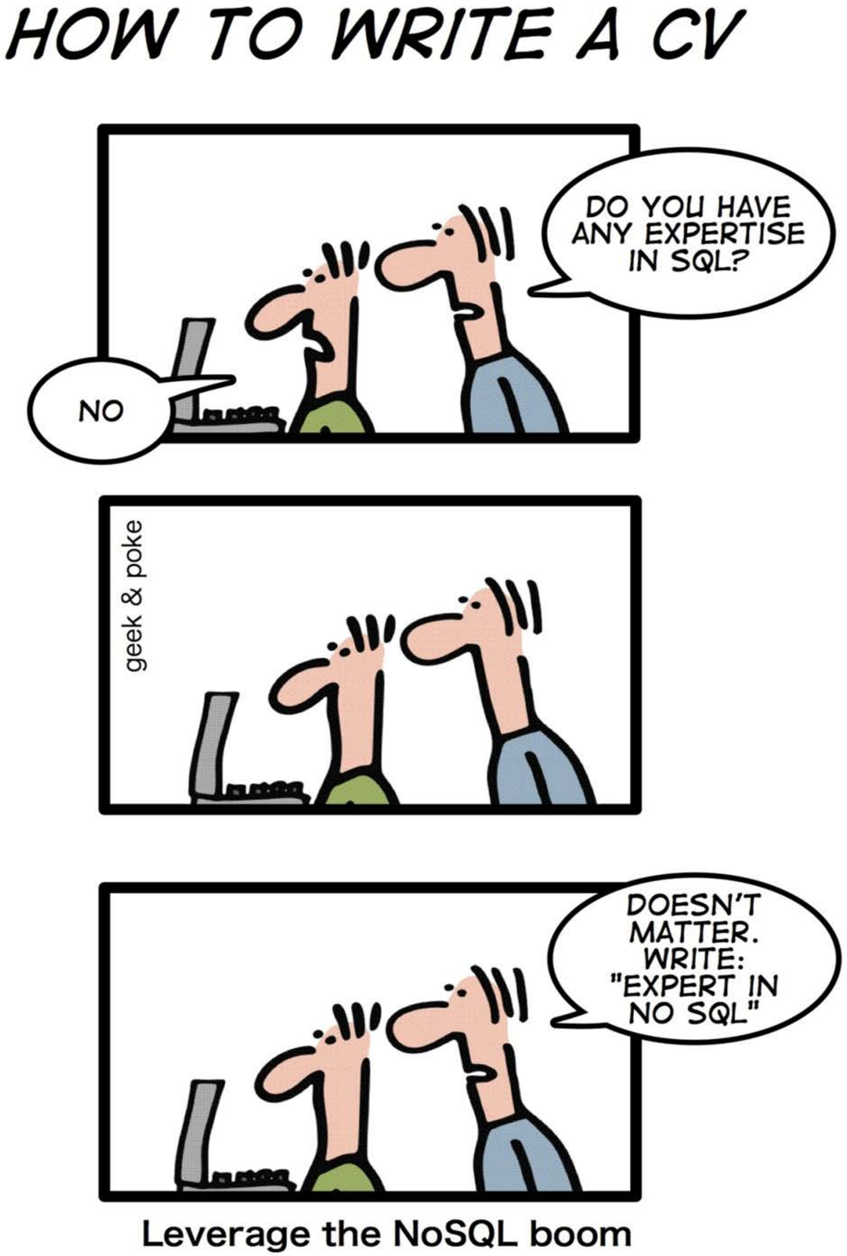
AT82.02

DATA MODELING AND MANAGEMENT

UNIT 2-1: NOSQL DATA MODEL AND MANAGEMENT

CHUTIPORN ANUTARIYA (CHUTI AT AIT DOT AC DOT TH)

Co-funded by the Erasmus+ Programme of the European Union


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom

A Little Humor…

3 DATABASE ADMINS

WALKED INTO

A NOSQL BAR …

A LITTLE WHILE LATER

THEY WALKED OUT BECAUSE

THEY COULDN’T FIND A TABLE

Fun Quiz Time!

Recap!

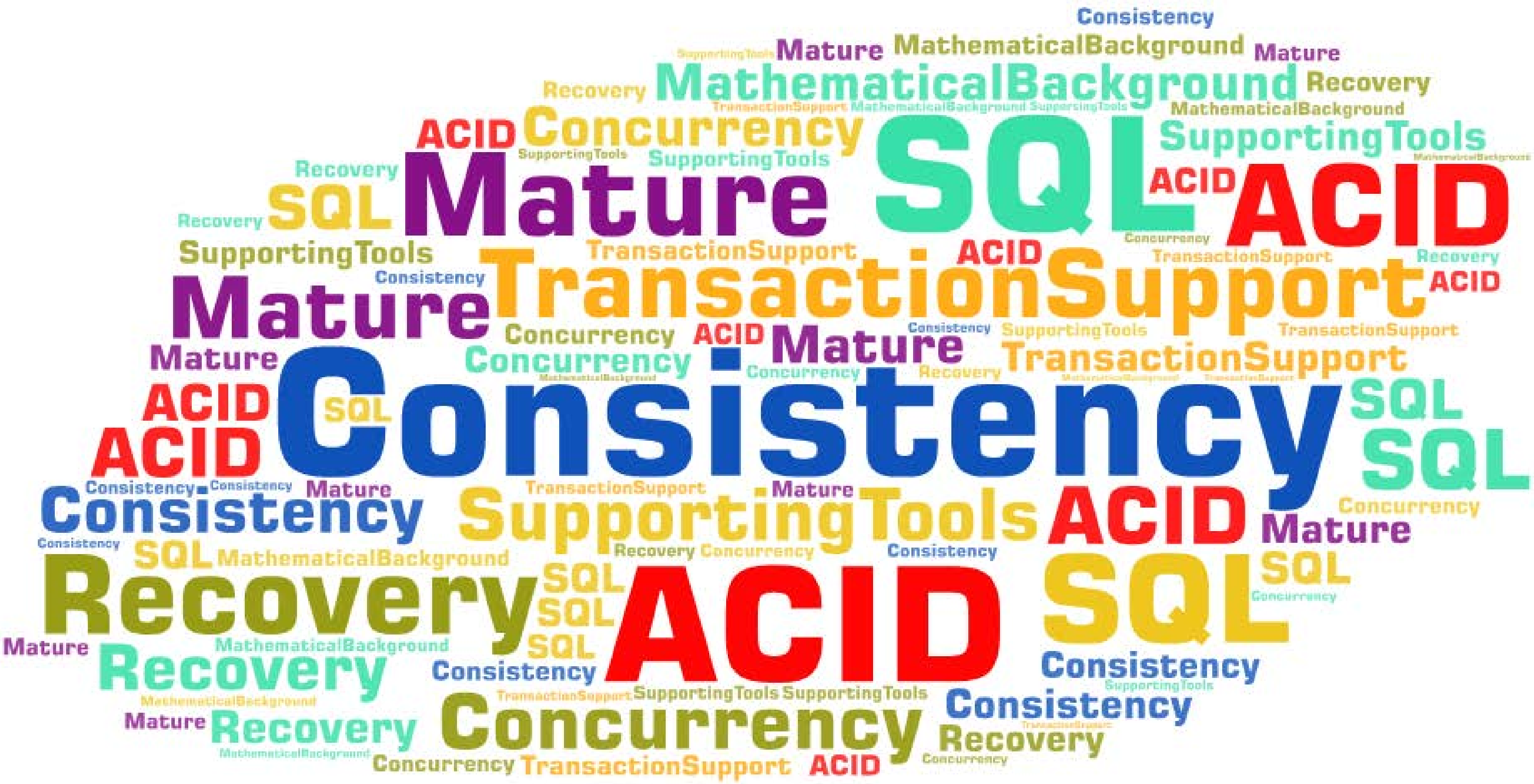
Data Model

describes data characteristics

Menti.com

Relational Database Recap!

CHARACTERISTICS, BENEFITS AND LIMITATIONS



Relational Databases: Characteristics and Benefits

RECAP

Data Model

A data model is a representation that we use to perceive and manipulate our data.

It allows us to:

- Represent the data elements under analysis, and
- How these are related to each others

This representation depends on our perception.

Data Model: Definition

RECAP

In this course we will refer “data model” as the model by which the database organize data.

RECAP

Data Model describes data characteristics

Structure

Operations

Constraints

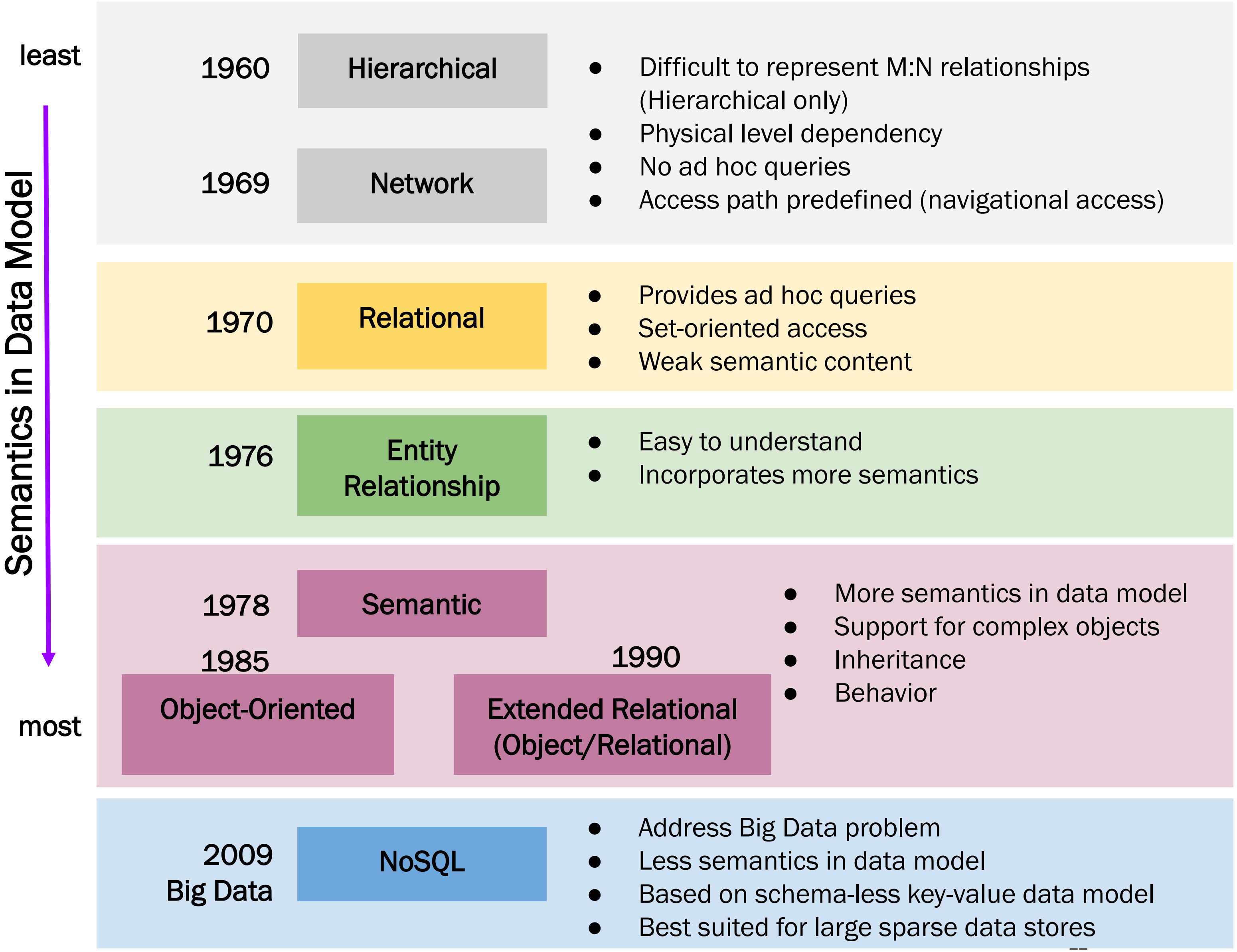
Relational Databases: Limitations

Scalability Issues

- Scale up vs. Scale out (vertical vs. horizontal)
- Not designed to run on clusters / distributed applications
- Joins are expensive

Schema-ful Databases vs. Schema-less Databases

Evolution of data models



1960 Hierarchical

1969 Network

- Difficult to represent M:N relationships (Hierarchical only)
- Physical level dependency
- No ad hoc queries
- Access path predefined (navigational access)

1970 Relational

- Provides ad hoc queries
- Set-oriented access
- Weak semantic content

1976 Entity Relationship

- Easy to understand
- Incorporates more semantics

1978 Semantic

1985 Object-Oriented

1990 Extended Relational (Object/Relational)

- More semantics in data model
- Support for complex objects
- Inheritance
- Behavior

2009 Big Data NoSQL

- Address Big Data problem
- Less semantics in data model
- Based on schema-less key-value data model
- Best suited for large sparse data stores

NoSQL Data Modeling and Management

NO SQL

CONCEPTS AND CHARACTERISTICS

NoSQL Origin

Generally newer databases solving new and different problems;

Not only SQL;

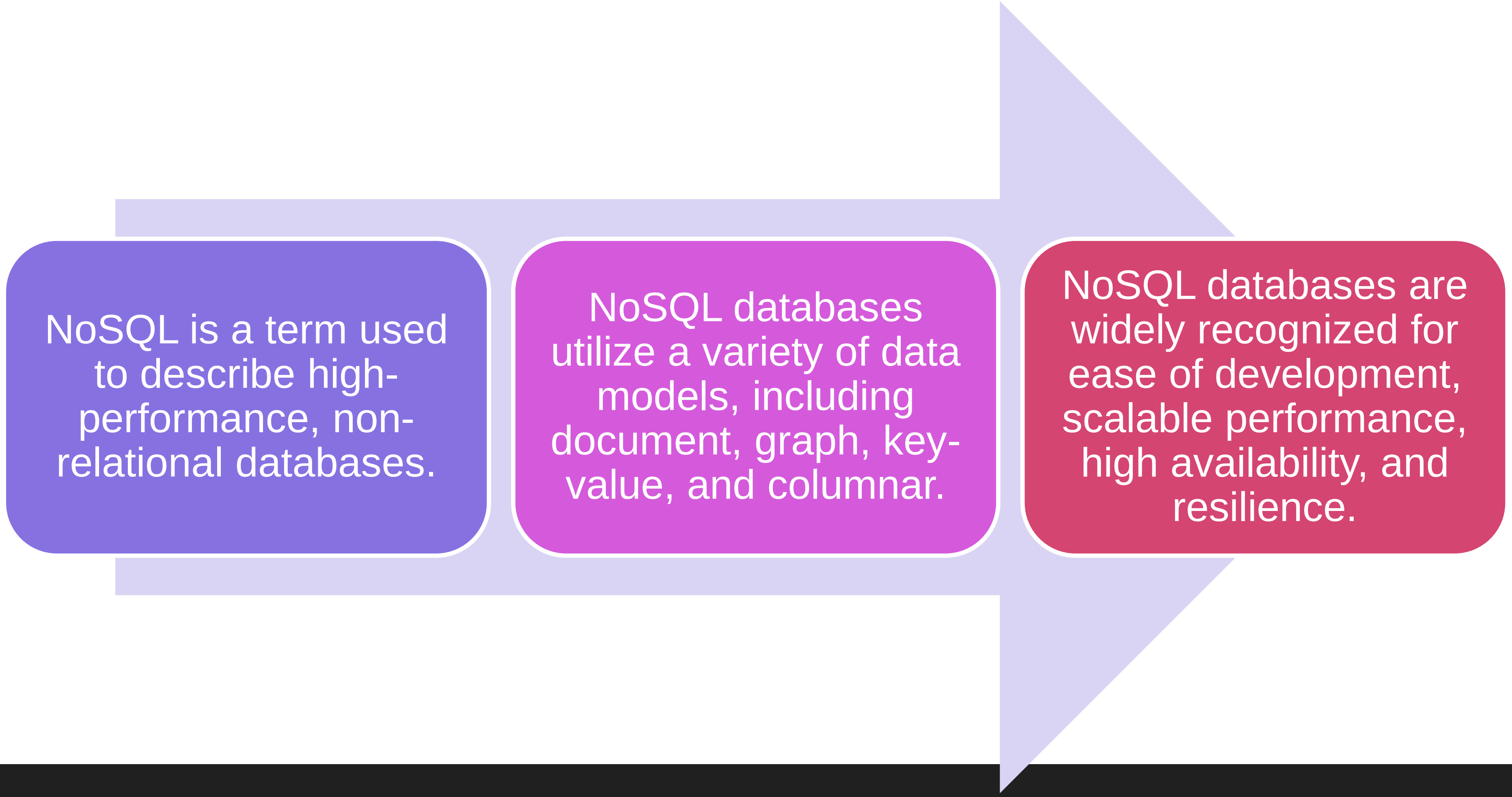
Problems not solved by RDBMSs;

Limitation of RDBMSs, not SQL;

NO SQL

NoSQL is a database technology designed to support the requirements of cloud applications and architected to overcome the scale, performance, data model, and data distribution limitations of relational databases (RDBMS’s).

What is NoSQL?

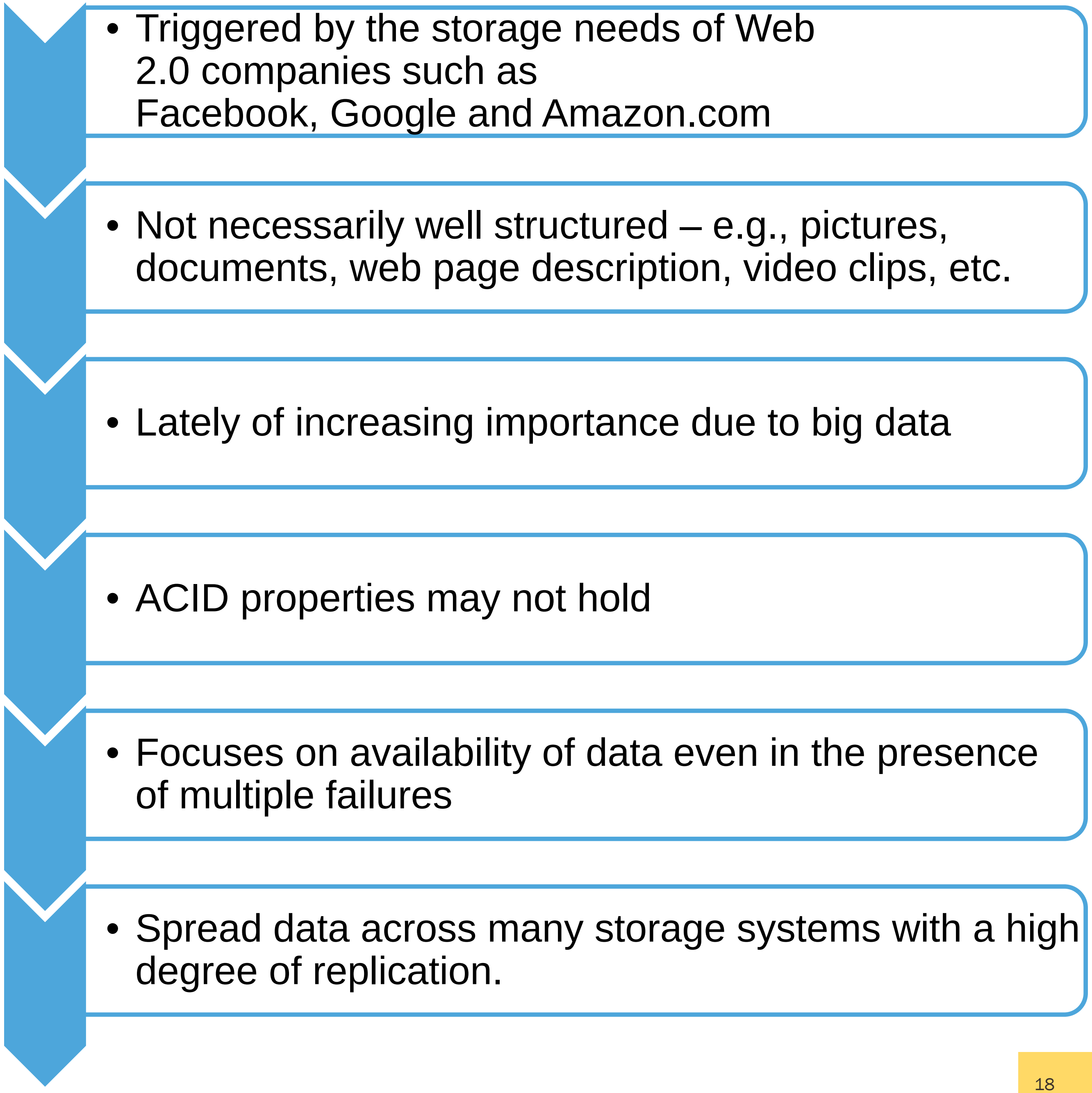


NoSQL is a term used to describe high-performance, non-relational databases.

NoSQL databases utilize a variety of data models, including document, graph, key-value, and columnar.

NoSQL databases are widely recognized for ease of development, scalable performance, high availability, and resilience.

Why NoSQL?



- Triggered by the storage needs of Web 2.0 companies such as Facebook, Google and Amazon.com
- Not necessarily well structured – e.g., pictures, documents, web page description, video clips, etc.
- Lately of increasing importance due to big data
- ACID properties may not hold
- Focuses on availability of data even in the presence of multiple failures
- Spread data across many storage systems with a high degree of replication.

NoSQL Definition

"Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontal scalable. The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge data amount, and more."

Ref: www.nosql-database.org

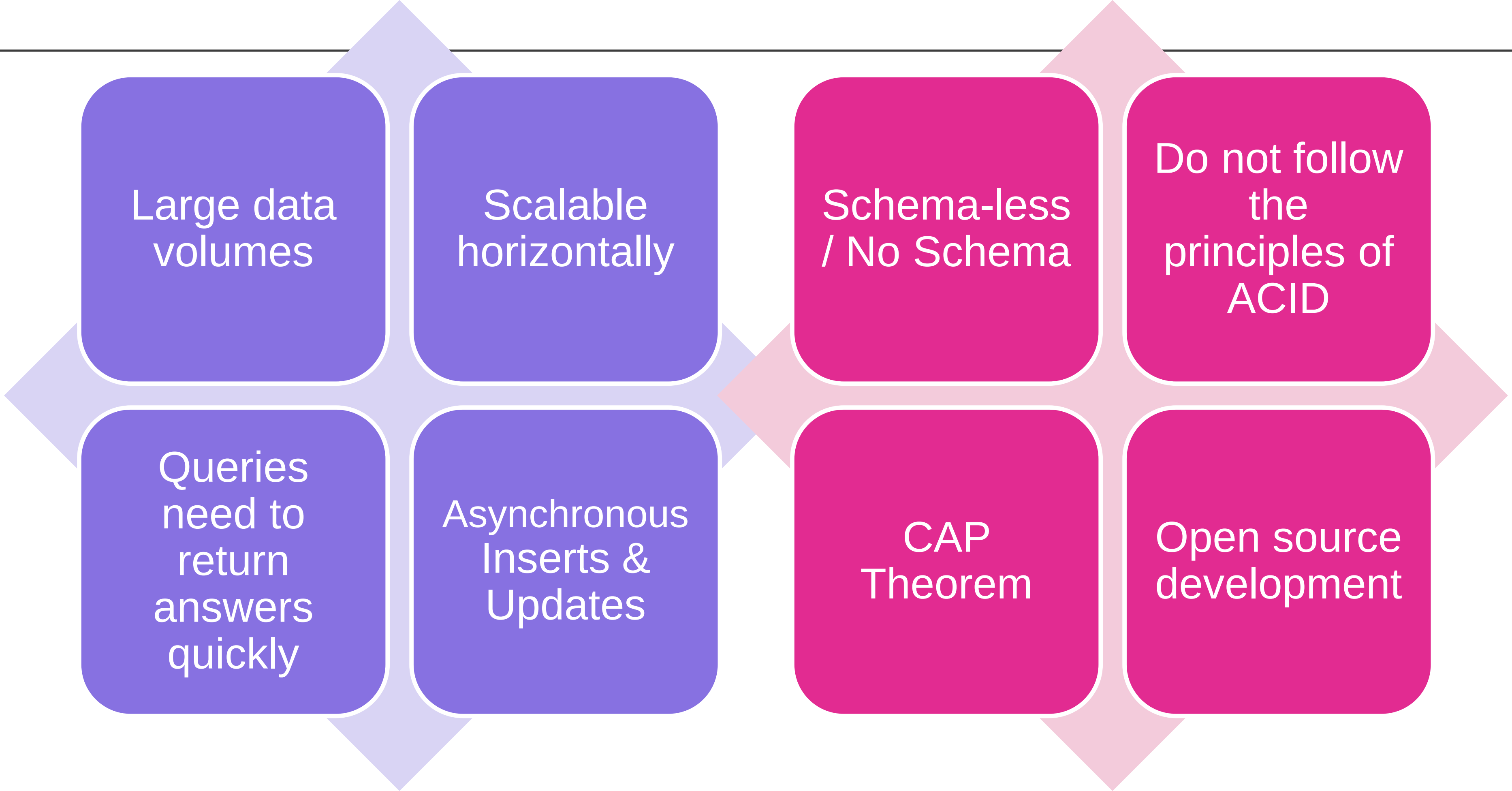
NoSQL vs Relational Model???

Data Model

Transaction Properties

Performance & Scale

NoSQL Distinguishing Characteristics



- Large data volumes
- Scalable horizontally
- Queries need to return answers quickly
- Asynchronous Inserts & Updates
- Schema-less / No Schema
- Do not follow the principles of ACID
- CAP Theorem
- Open source development

Schema-less Database: what is?

In Relational DB (schema-ful DB), there are limitations:

- Cannot add a record which does not fit a schema
- Needs to add NULL values to unused data attribute in a record
- Strong datatyping
- Composite attributes and multivalued attributes are not allowed!!

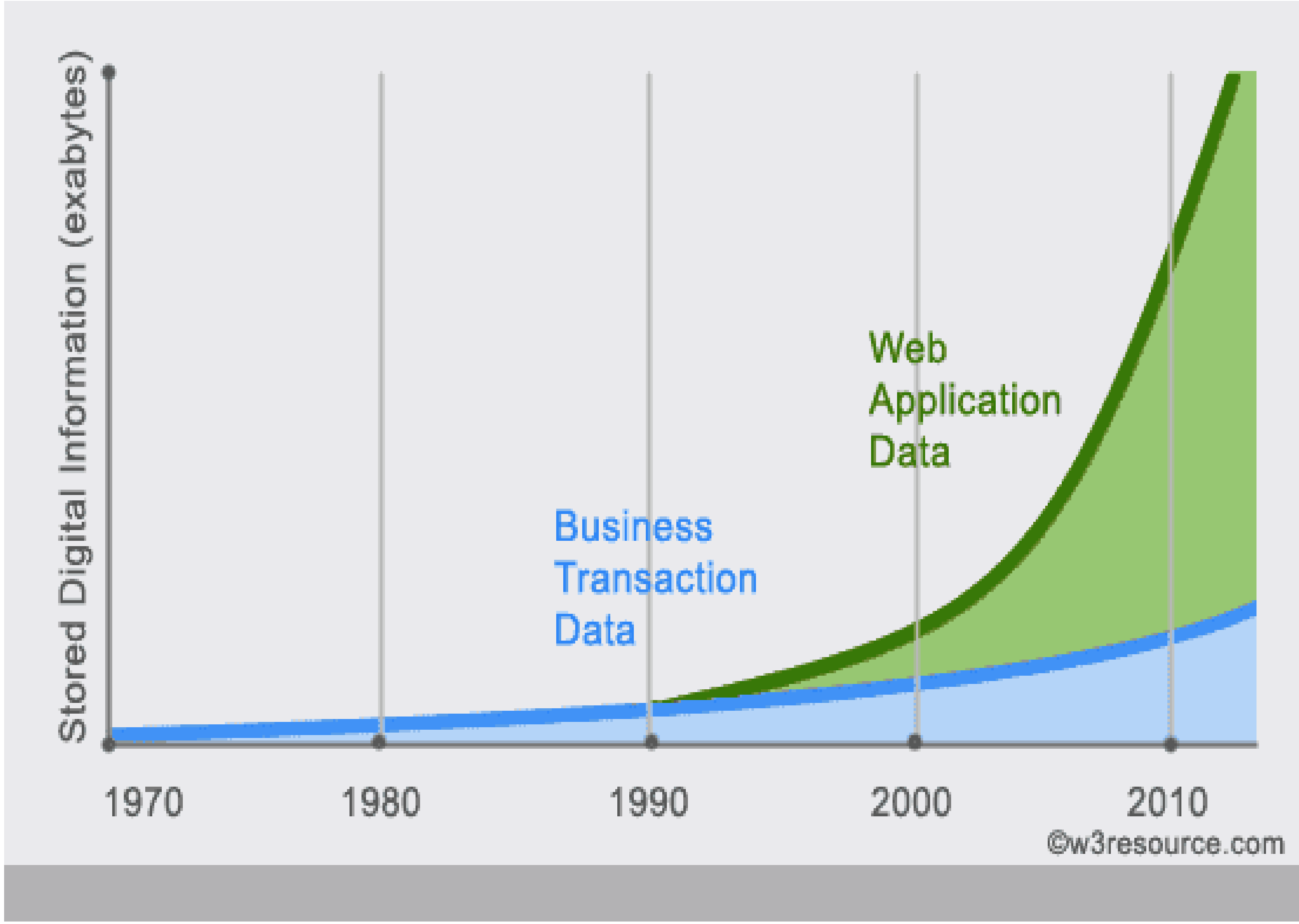
Schema-less Database: what is?

In Schema-less DB

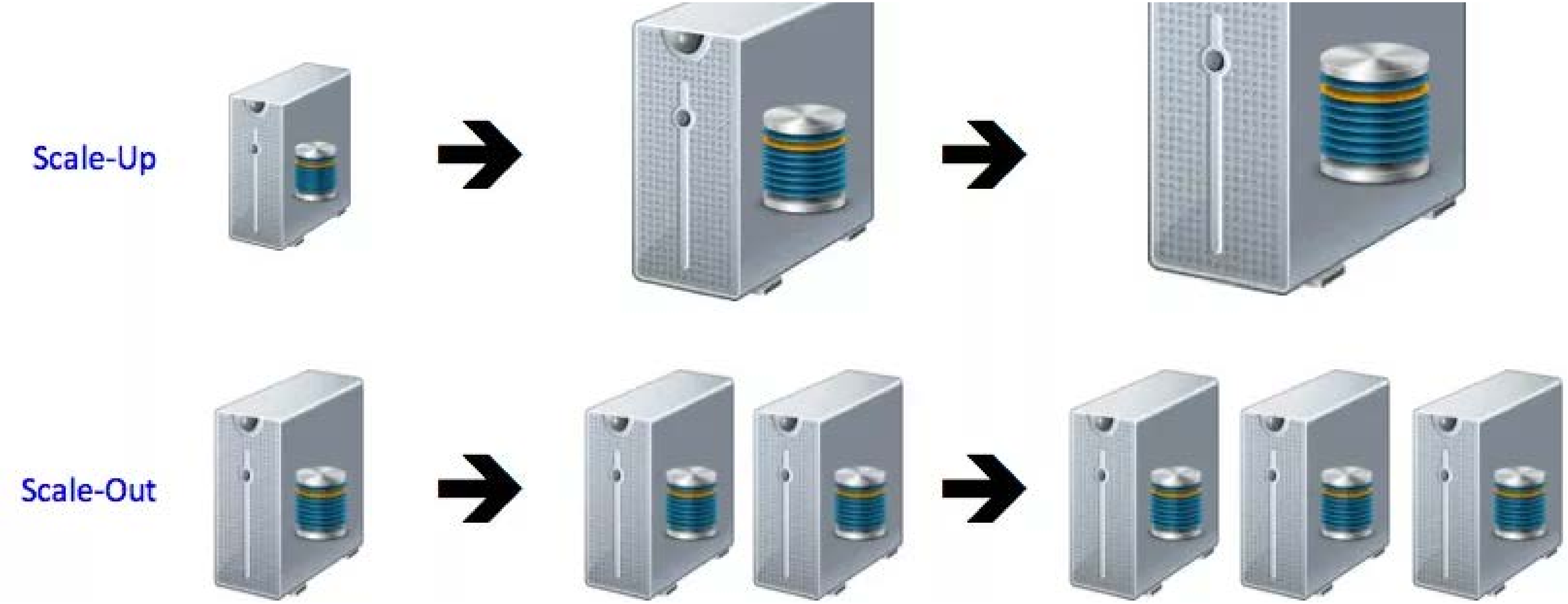
- No fixed, rigid Schema
- No NULL constraint/enforcement
- No datatyping

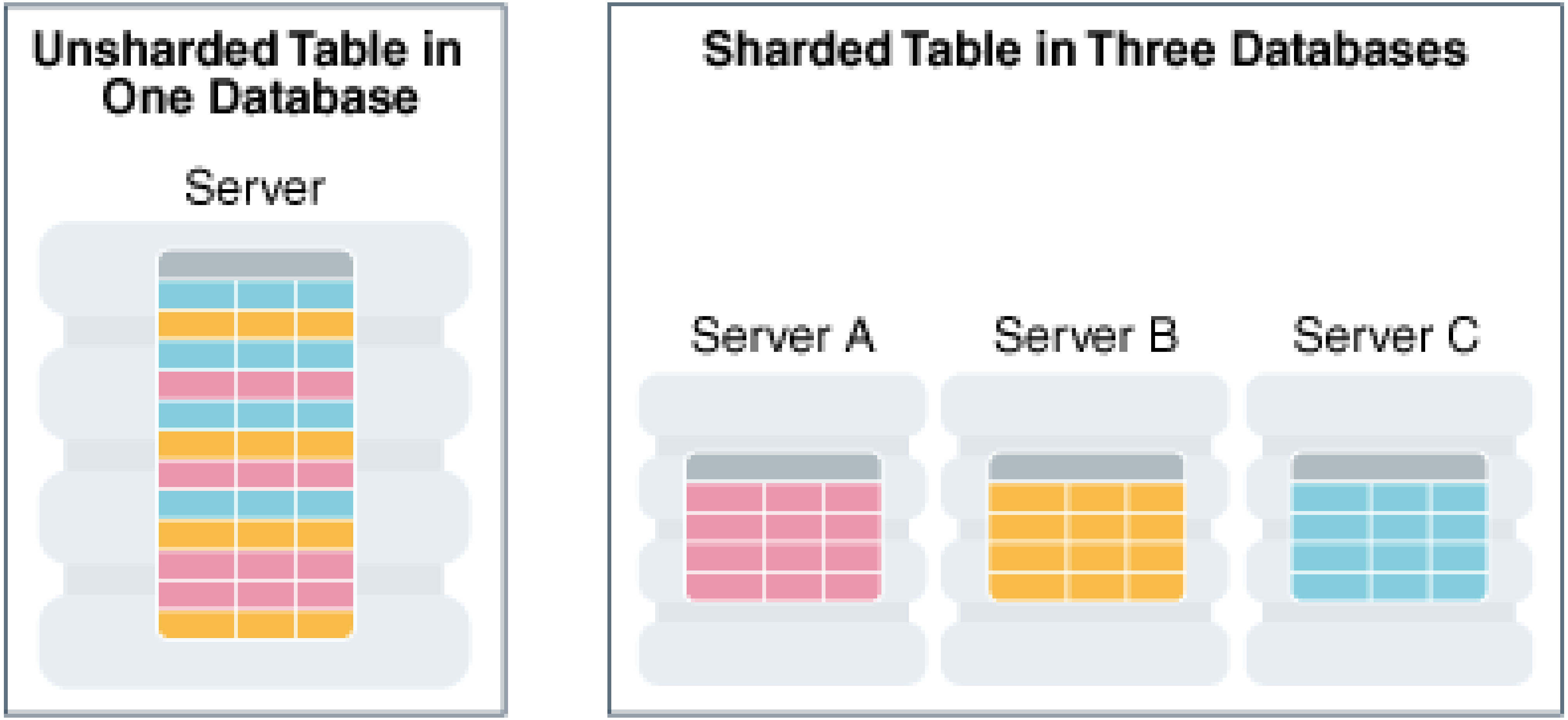
This is Schema-less Database!

Web Apps Driving Data Growth

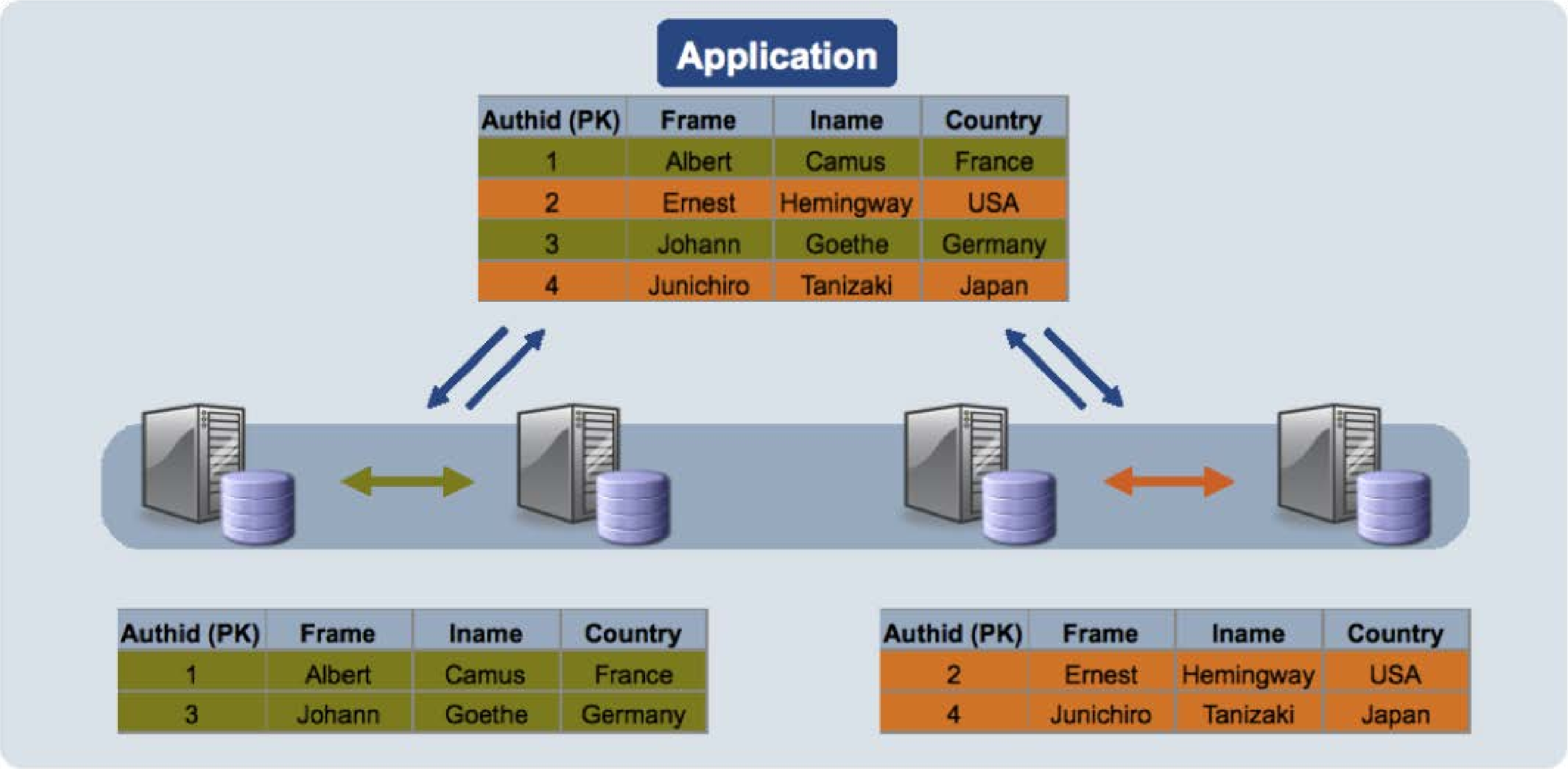


Scale UP vs. Scale OUT




Unsharded Table in One Database
Server
Sharded Table in Three Databases
Server A
Server B
Server C

Data Sharding



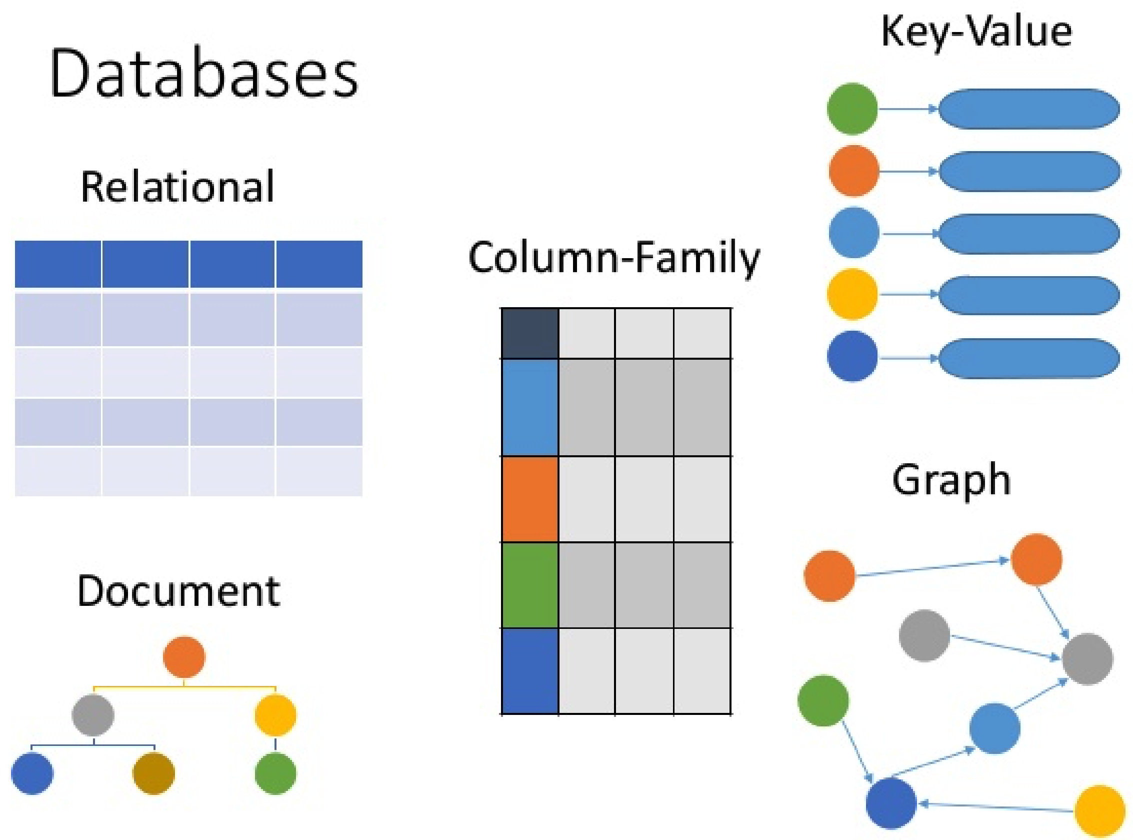
Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	USA
3	Johann	Goethe	Germany
4	Junichiro	Tanizaki	Japan

Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
3	Johann	Goethe	Germany

Authid (PK)	Frame	Iname	Country
2	Ernest	Hemingway	USA
4	Junichiro	Tanizaki	Japan

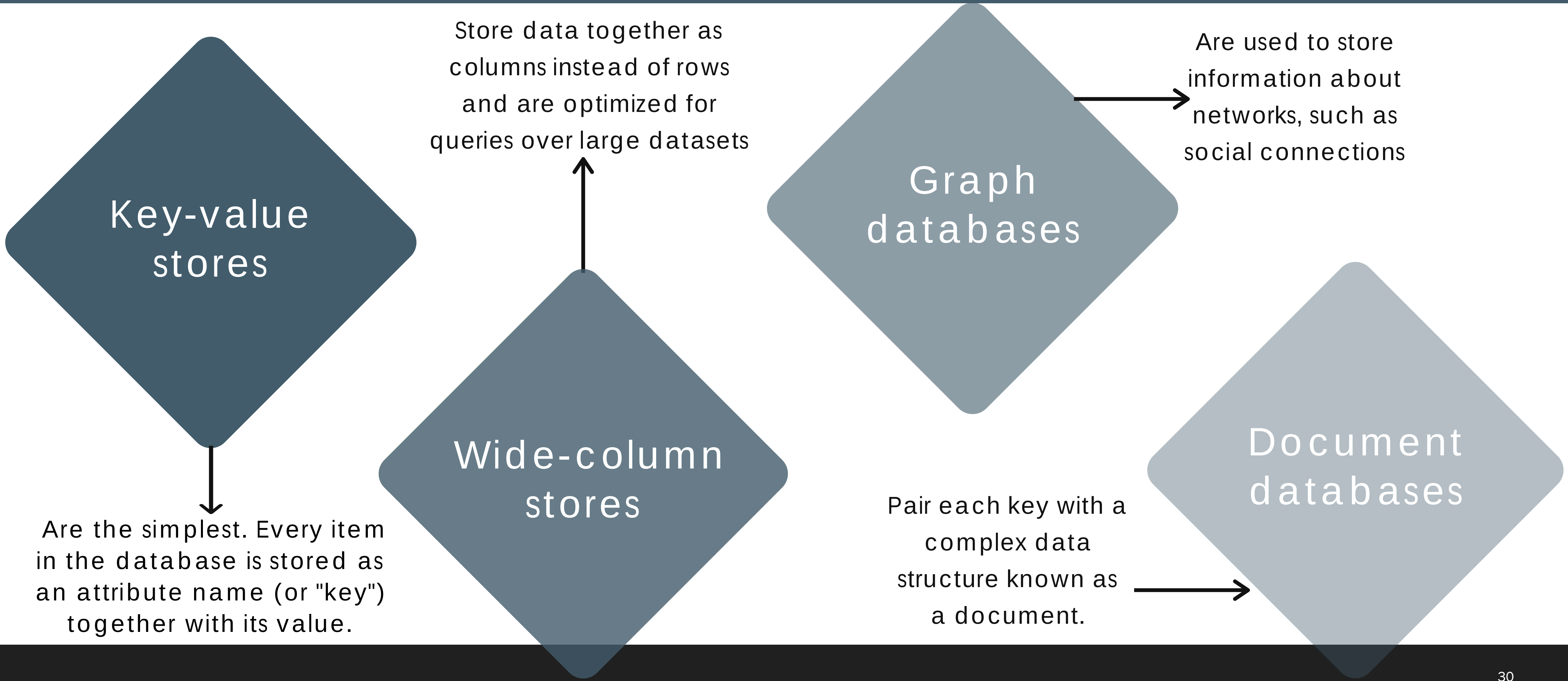
Data Sharding

Database Family



Major Categories of NOSQL Data Models

MAJOR CATEGORIES OF NOSQL DATA MODELS



Key-value stores

Are the simplest. Every item in the database is stored as an attribute name (or "key") together with its value.

Wide-column stores

Store data together as columns instead of rows and are optimized for queries over large datasets

Graph databases

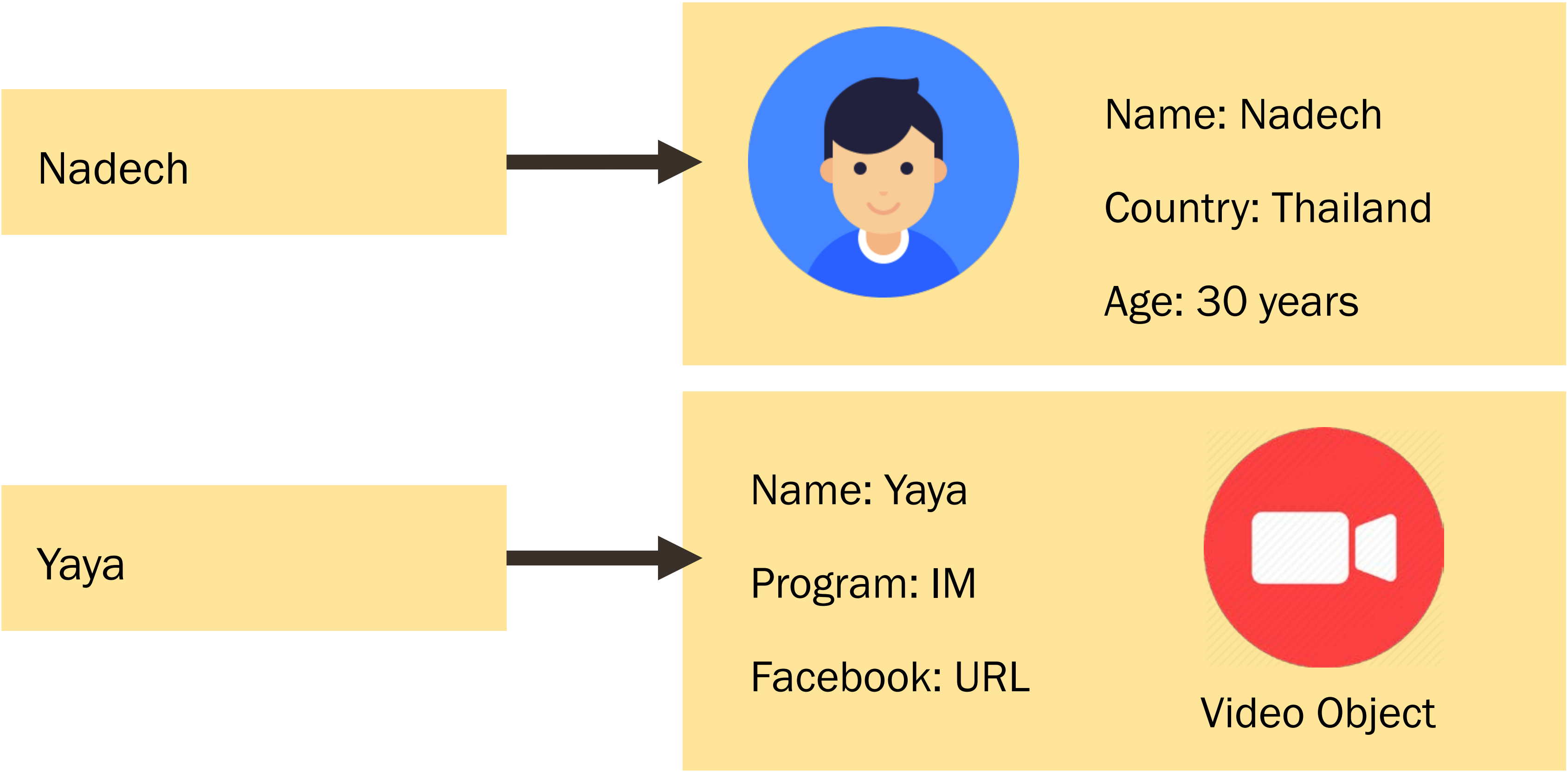
Are used to store information about networks, such as social connections

Document databases

Pair each key with a complex data structure known as a document.

Key-Value Model

Key-Value Mode



Key

Value

Nadech

Name: Nadech

Country: Thailand

Age: 30 years

Yaya

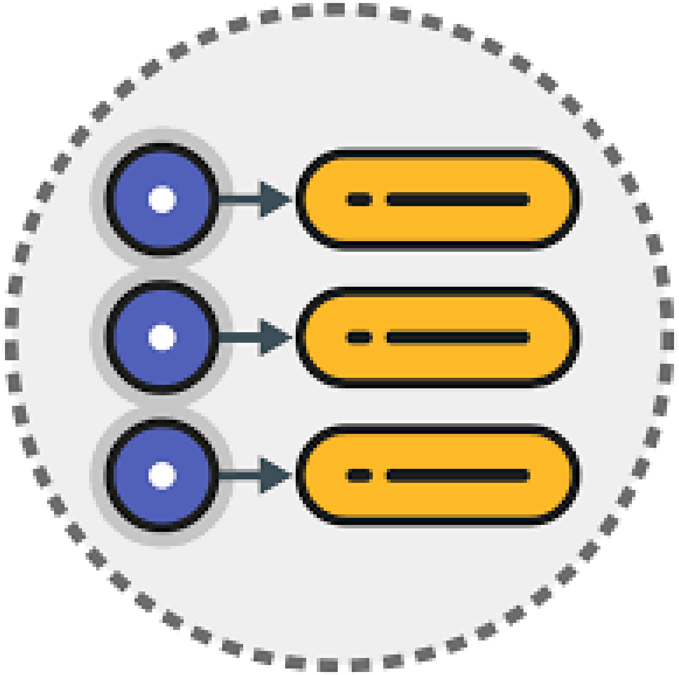
Name: Yaya

Program: IM

Facebook: URL

Video Object

Key-Value Model



The simplest model: just Keys and Values

- No Schema
- Keys: synthetic or auto-generated
- Values: any object type (e.g., String, JSON, BLOB) stored as uninterpreted block, thus the keys are the only way to retrieve stored data.

Query operations for stored objects are associated with a key:

- PUT, GET, DELETE

Benefits vs. Limitations

BENEFITS

Extremely fast retrieval using the key

Virtually no restriction on the type of data that can be stored:

- Text (for example, the HTML code for a Web page)
- Any type of multimedia binary (still images, audio, and video).

LIMITATIONS

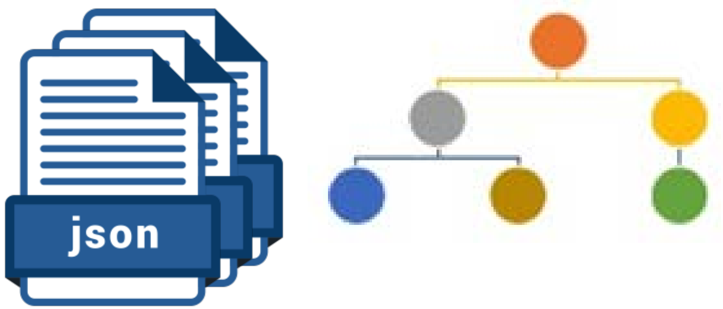
Cannot search within stored values rather than always retrieving by the key

Cannot update parts of a “value” while it’s in the database. You must replace the entire value with a new copy if modifications are needed.

Applications & Use Cases

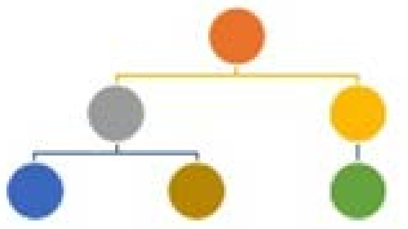
Best suited for applications where access is only through the key.

They are being used for Web sites that include thousands of pages, large image databases, and large catalogs. They are also particularly useful for keeping Web app session information.



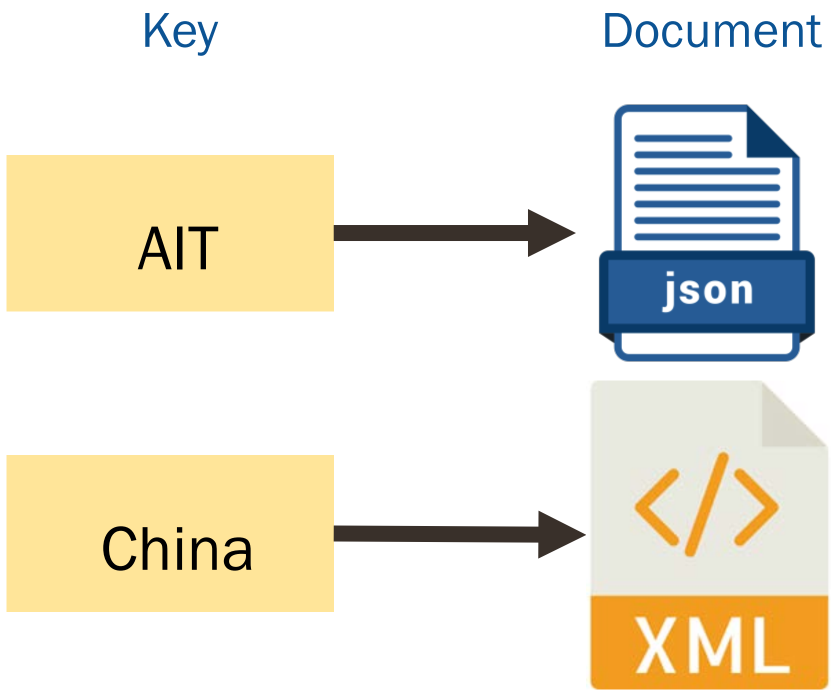
Document Model

Document Model

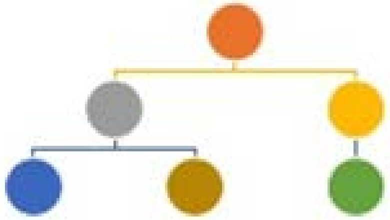


A specialized Key-value Store but rather than storing “values,” it stores “documents”, which are not adhered to schema restrictions.

Provides a way to query the documents based on the contents or metadata.



Document Model



A specialized Key-value Store

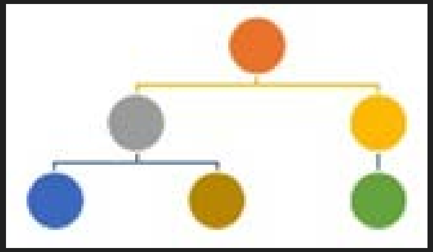
Designed for storing, retrieving and managing document-oriented information, also known as semi-structured data, such as XML, JSON, BSON

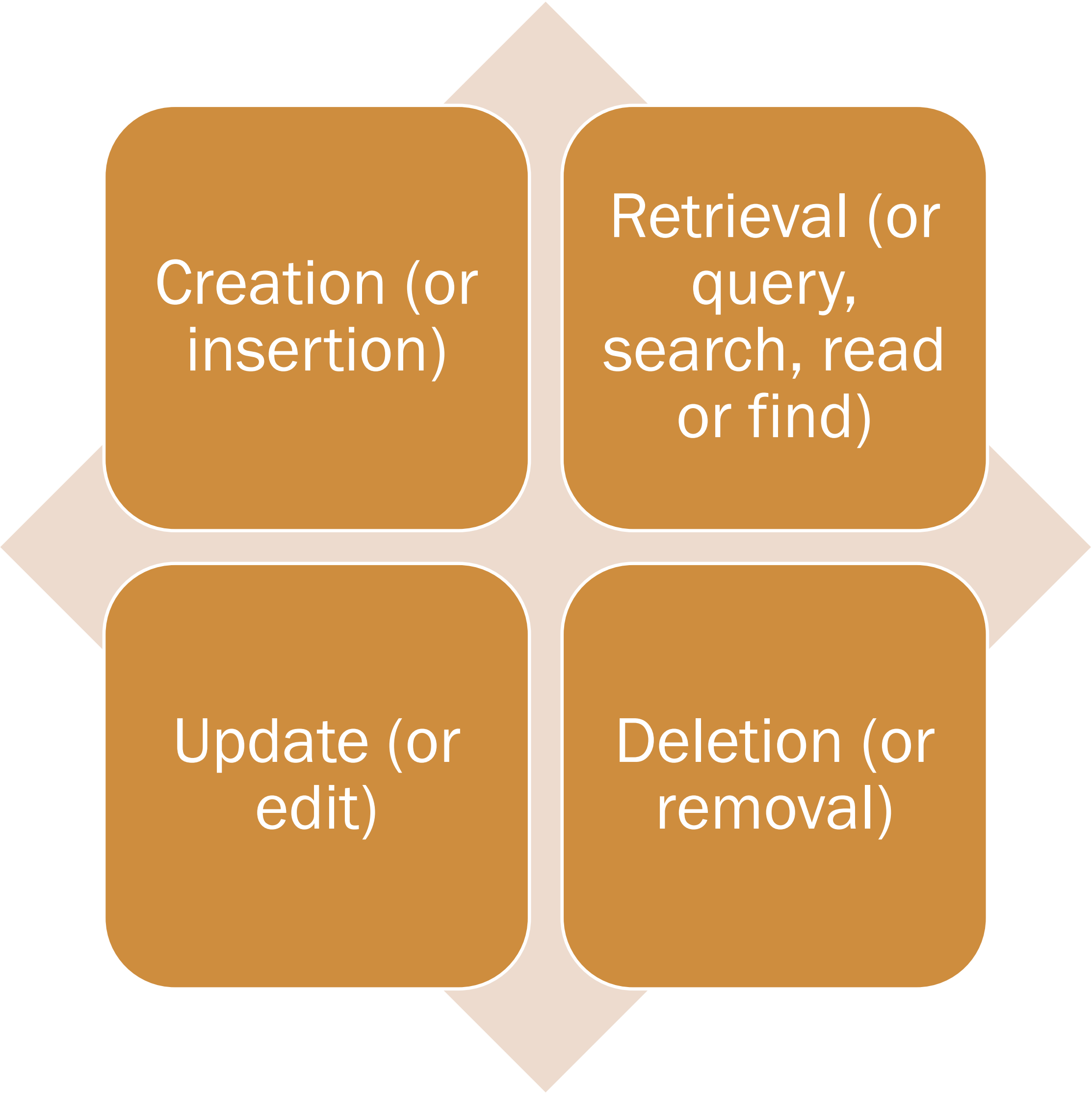
Provides APIs or a query/update language that exposes the ability to query or update based on the internal structure in the document.

```json
{
    "FirstName": "Bob",
    "Address": "5 Oak St.",
    "Hobby": "sailing"
}
```

```xml
<contact>
   <firstname>Bob</firstname>
   <lastname>Smith</lastname>
   <phone type="Cell">(123) 555-0178</phone>
   <phone type="Work">(890) 555-0133</phone>
   <address>
     <type>Home</type>
     <street1>123 Back St.</street1>
     <city>Boys</city>
     <state>AR</state>
     <zip>32225</zip>
     <country>US</country>
   </address>
 </contact>
```

CRUD Operations



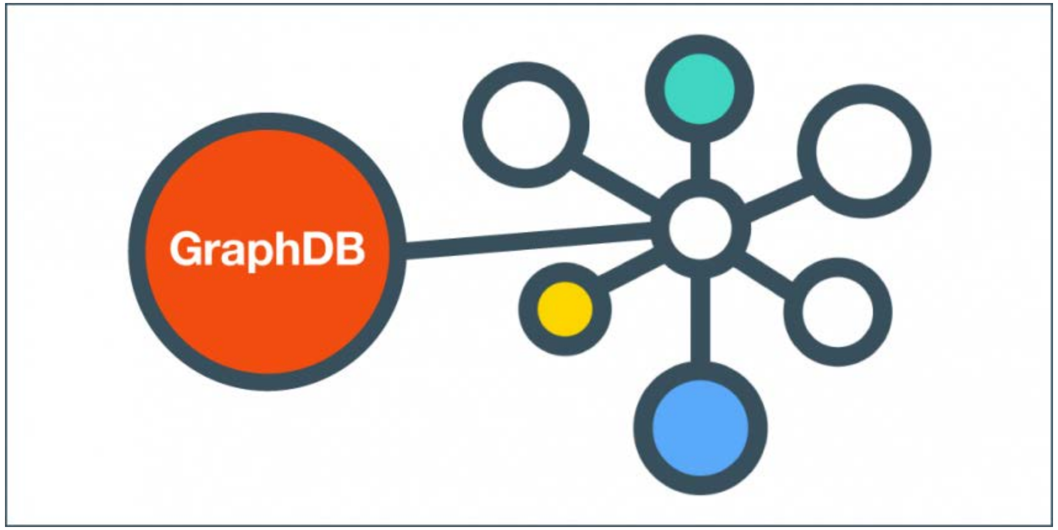


Creation (or insertion)

Retrieval (or query, search, read or find)

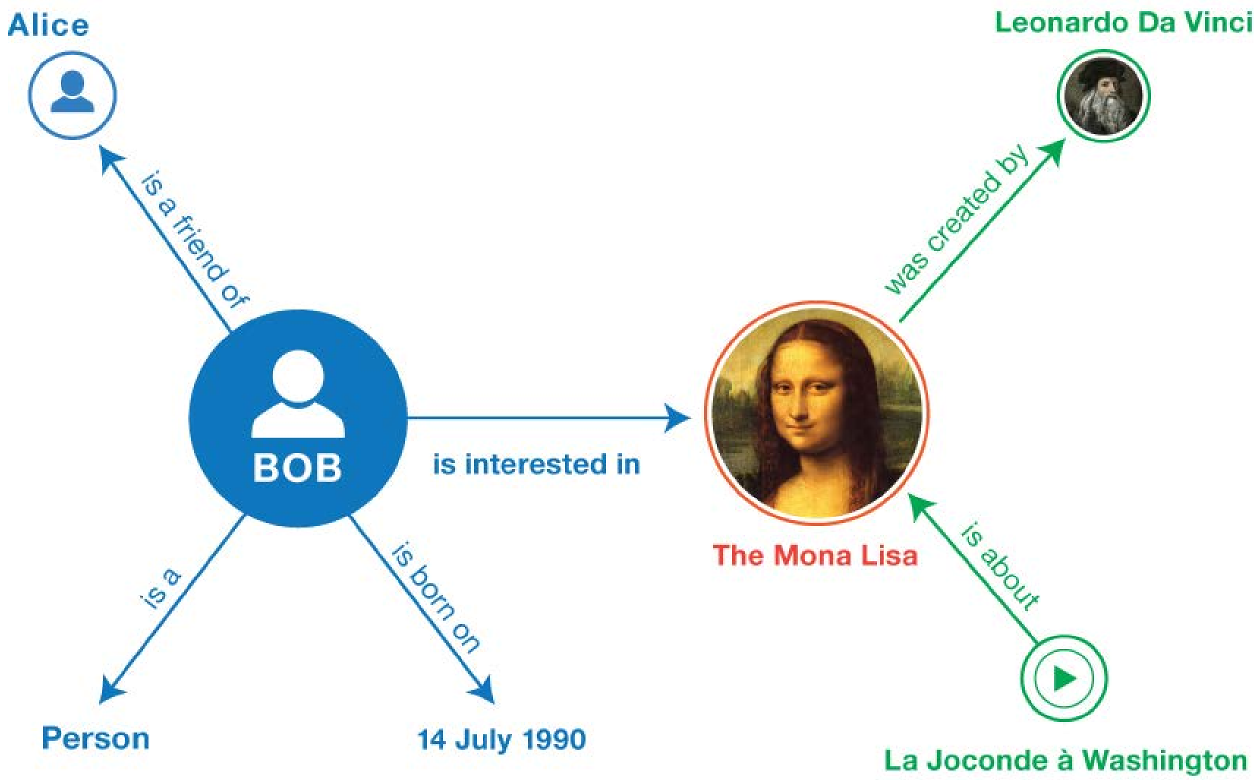
Update (or edit)

Deletion (or removal)



Graph Model

Graph Model (nodes-links-properties structure)



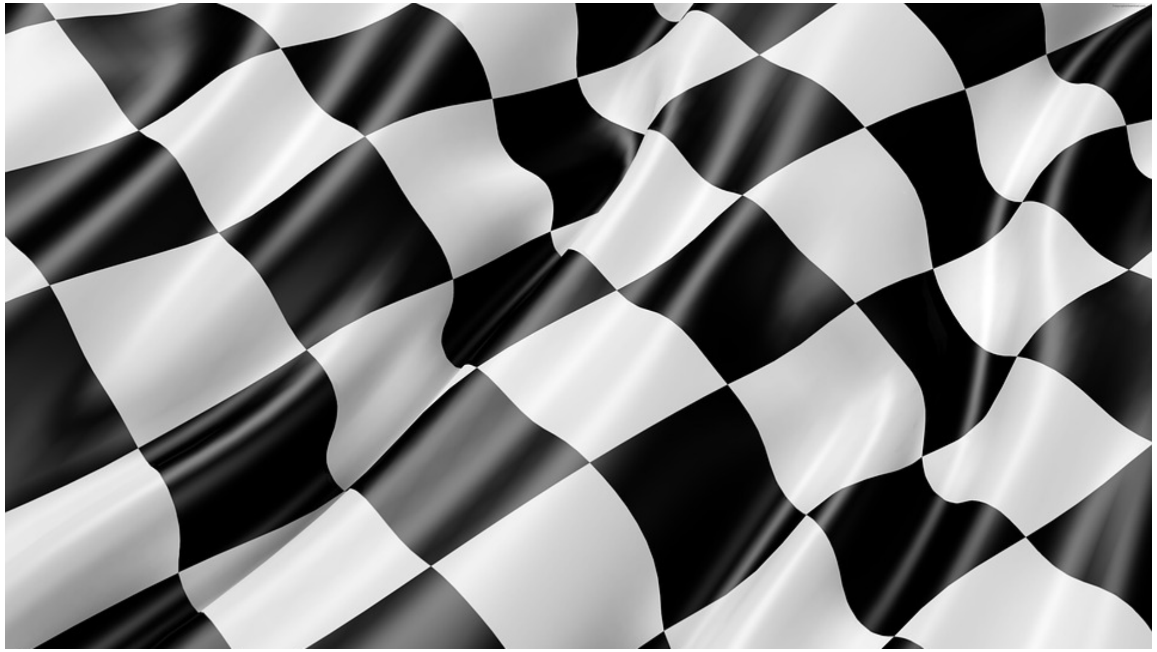
Graph Model

Graph store uses graph structures for semantic queries with nodes, edges and properties to represent and store data.

The relationships allow data in the store to be linked together directly, and in many cases retrieved with one operation.

A query on a graph is known as traversing the graph.

The biggest advantage of the graph store is that joins are not necessary.



Thank you.

Exit Slip:
Summary of What You Learned (max 100 words)
What remains unclear (if any)