PaxeraHealth
Transforming Healthcare Through Innovation

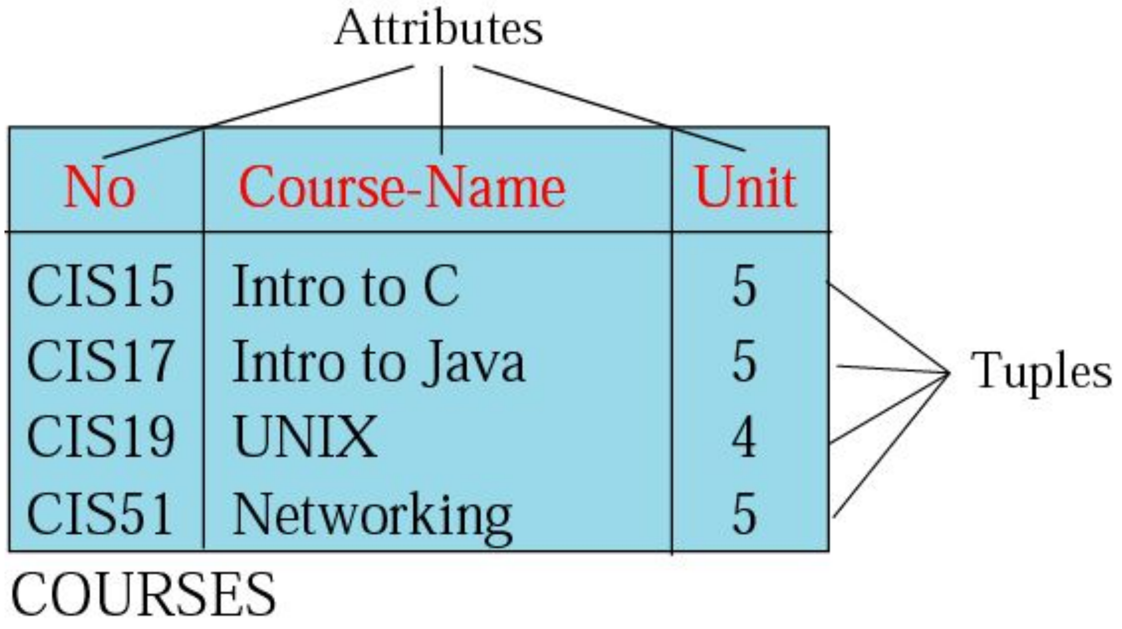
NoSQL Database

"Towards the end of RDBMS ?"

Habiba Ahmed Magdy
habiba.hamad97@eng-st.cu.edu.eg

What is RDBMS

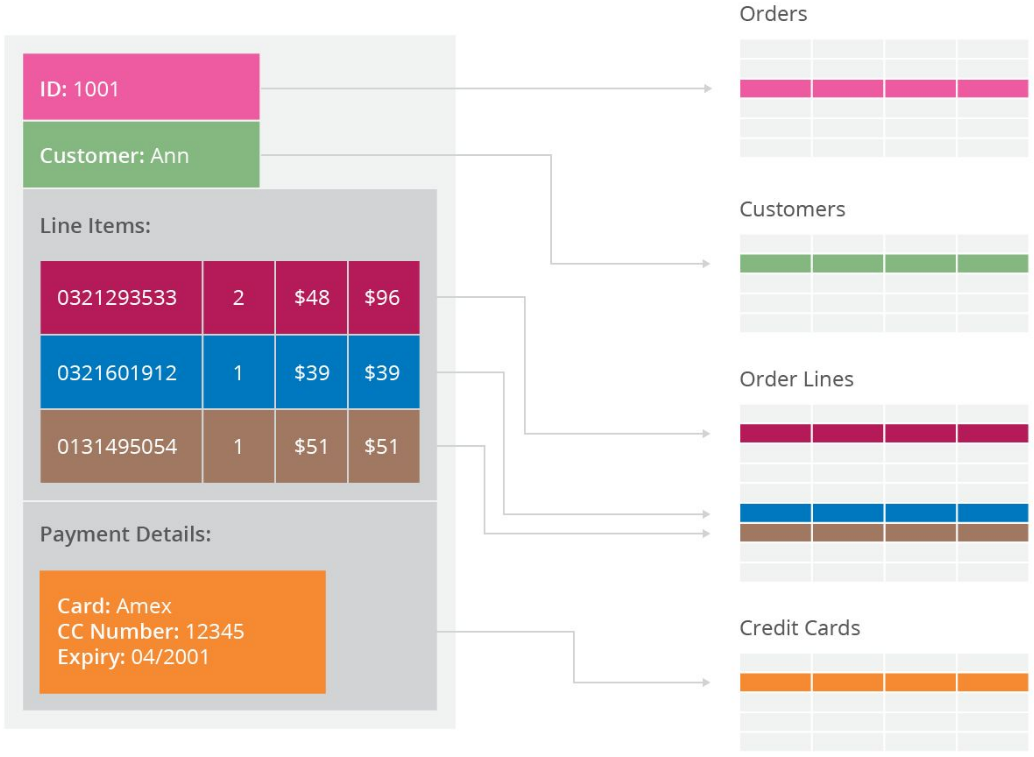
- RDBMS: the relational database management system.
- Standard Query language (SQL) Database
- Relation: a relation is a 2D table



No	Course-Name	Unit
CIS15	Intro to C	5
CIS17	Intro to Java	5
CIS19	UNIX	4
CIS51	Networking	5

Issues with RDBMS
➢ The Need for NoSQL

- ❖ Scaling up when the dataset is just too big e.g. Big Data.
- ❖ Not designed to be distributed.
- ❖ Predefined Schema
- ❖ Expensive
  - ➢ Different approaches



What is NoSQL

- ❑ Stands for Not Only SQL.

- ❑ Non-relational data storage systems.
- ❑ Recognized for their ease of development, functionality, and performance at scale.
- ❑ Use a variety of data models

NoSQL Database Features

- ❑ Flexible schemas for building modern applications.
- ❑ Does not depend on predefined tables.
- ❑ Flexibility in Data Storage: host semi-structured or unstructured data.
- ❑ Designed to be distributed, Horizontal scaling.
- ❑ The concept of joining records from multiple tables doesn’t exist (Later)

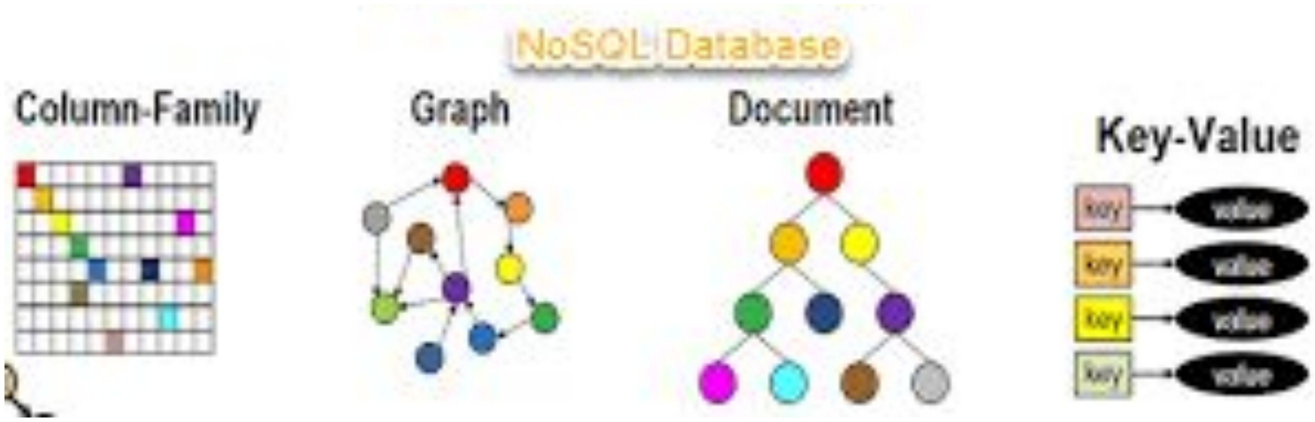
What Drives The Need of NoSQL

- ❖ Explosion of social media sites (Facebook, Twitter, Google etc.) with **large** data needs.
- ❖ Rise of **cloud-based** solutions, simple storage solution.
- ❖ Moving to **dynamically-typed data** with frequent schema changes.
- ❖ Expansion of **Open-source** community.
- ❖ Support **automatic replication:** high availability and disaster recovery **without** involving separate applications to manage these tasks

NoSQL Data Model Types

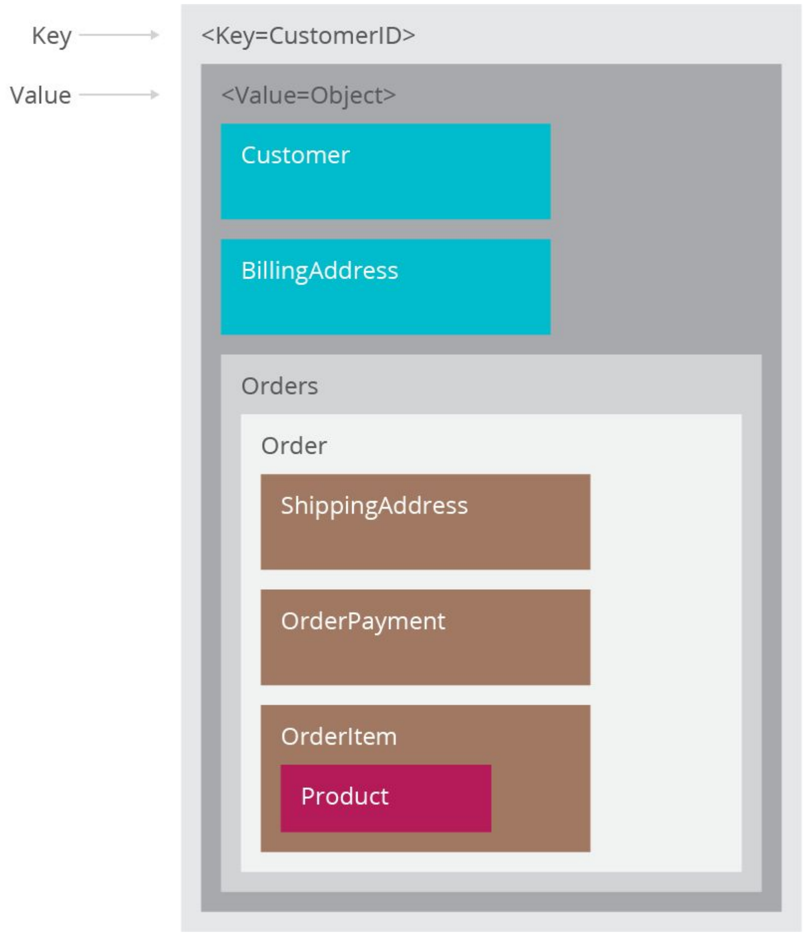
NoSQL database are classified into four types:

- ➢ Key Value pair based
- ➢ Column based
- ➢ Document based
- ➢ Graph based



Key Value Pair Based

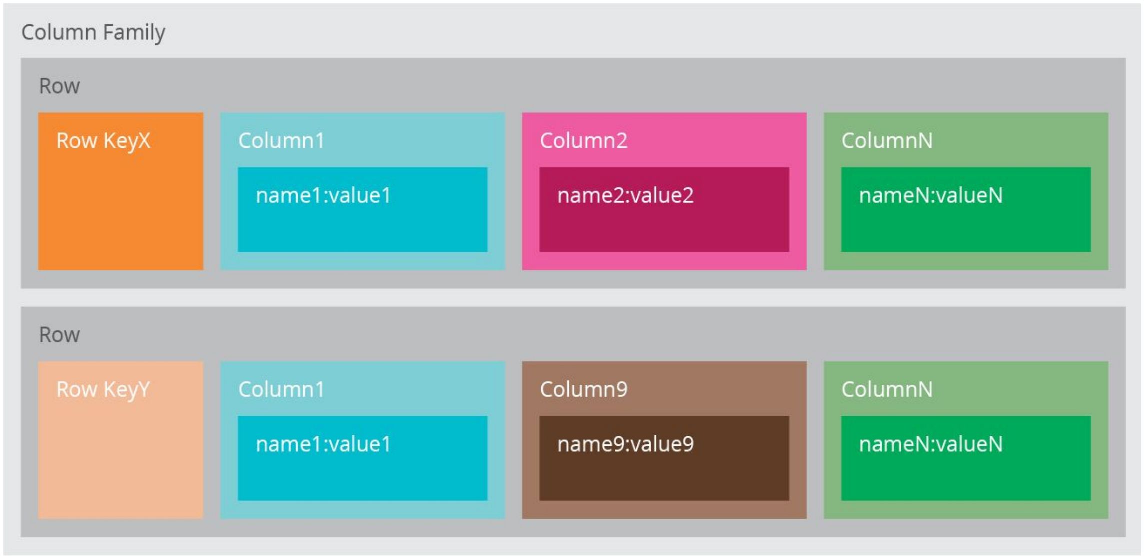
- ❑ Simplest NOSQL databases where it use a hash table to access data (values) by strings called keys
- ❑ Data has no required format data may have any format
- ❑ Use it: storing session info. , user profiles, shopping cart data.
- ❑ Avoid it: need to query data having relationships between entities.

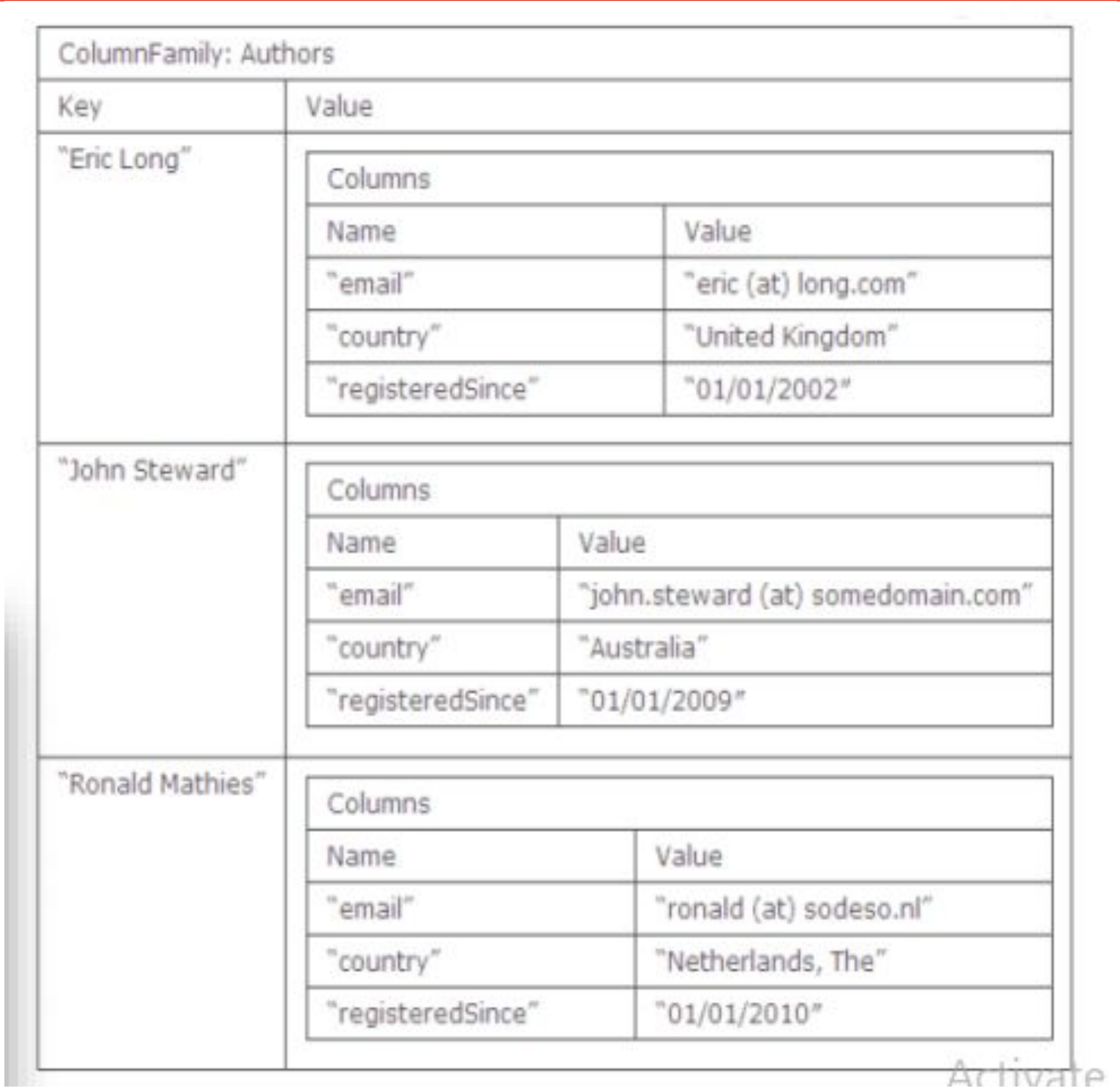


- ❑ Examples :Amazon DynamoDB, Oracle NoSQL Database, Redis, etc.

Column based

- ❑ It store data as Column families containing rows that have many columns associated with a row key. Each row can have different columns.
- ❑ Column families are groups of related data that is accessed together.
- ❑ We use it for content management systems(WordPress), blogging, etc.
- ❑ We would avoid it for systems that are in early development, changing query patterns.
- ❑ Example: Cassandra, HBas and Hypertable.





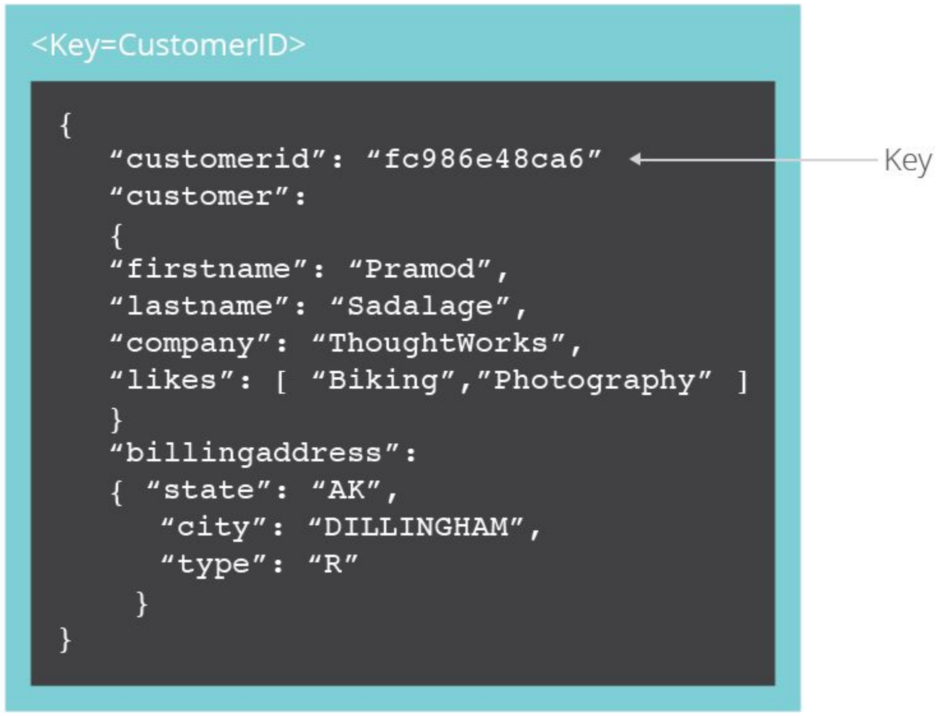
ColumnFamily: Authors

Key	Value	
	Columns	
	Name	Value
"Eric Long"	"email"	"eric (at) long.com"
	"country"	"United Kingdom"
	"registeredSince"	"01/01/2002"
"John Steward"	"email"	"john.steward (at) somedomain.com"
	"country"	"Australia"
	"registeredSince"	"01/01/2009"
"Ronald Mathies"	"email"	"ronald (at) sodeso.nl"
	"country"	"Netherlands, The"
	"registeredSince"	"01/01/2010"

Activate

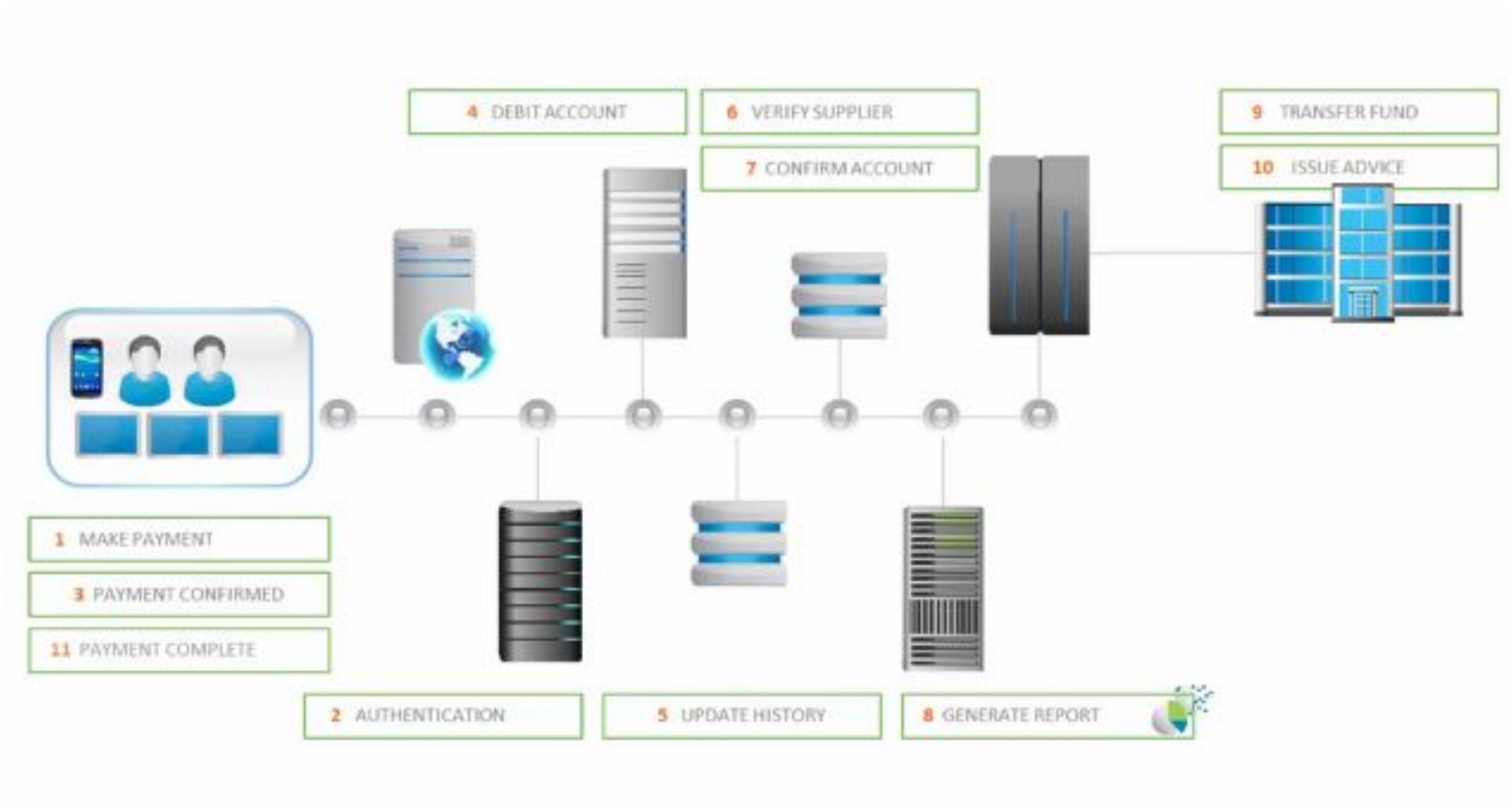
Document Based

- ❑ The database stores and retrieves documents (JSON documents).
- ❑ It stores documents in the value part of the key-value store.
- ❑ We use it for content management systems, blogging platforms, web analytics, real-time analytics.
- ❑ Examples: MongoDB, Couchbase, Orient DB, Raven DB.
- ❑ We would avoid it for systems that need complex transactions spanning multiple operations or queries against varying aggregate structures.



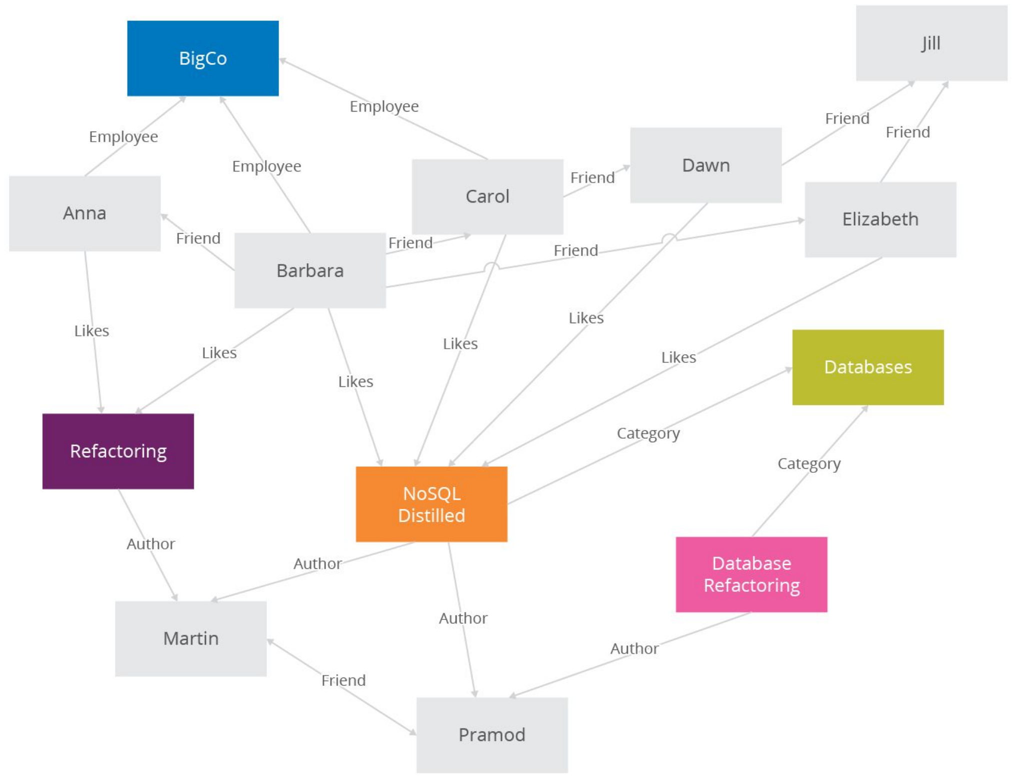
```
{
    "customerid": "fc986e48ca6"
    "customer":
    {
    "firstname": "Pramod",
    "lastname": "Sadalage",
    "company": "ThoughtWorks",
    "likes": [ "Biking","Photography" ]
    }
    "billingaddress":
    { "state": "AK",
       "city": "DILLINGHAM",
       "type": "R"
     }
}
```

Complex transactions spanning multiple operations



Graph Based

- ❑ Use components like "edges", "nodes", and "properties" to store and relate data.
- ❑ Traversing the relationships is very fast.
- ❑ It is well suited for connected data, such as social networks, and e-commerce stores.
- ❑ Example: Neo4J, Infinite Graph, FlockDB.
- ❑ Neptune are examples of graph databases.



CAP Theorem

- ❖ A distributed system has 3 properties :
  - ➢ Consistency (Write & Return)
  - ➢ Availability (Response)
  - ➢ Partitions-Tolerant
- ❖ We can have at most two of these three properties for any shared-data system
- ❖ To scale out, we have to partition. It leaves a choice between consistency and availability. (In almost all cases, we would choose availability over consistency)
- ❖ Everyone who builds big applications builds them on CAP : Google, Yahoo, Facebook, Amazon, eBay, etc.

Cassandra Vs MongoDB Vs Redis

	Cassandra	MongoDB	Redis
Data Model	Column-Family structure	JSON document format	Key Value Pair Based
Known for	leading NoSql distributed data management system	flexible & schema-less database	Redis works with an in-memory dataset
Written in	Java	C++	ANSI C.

Cassandra Vs MongoDB Vs Redis

Cassandra	MongoDB	Redis
Failure handling Best-in-class scalability and performance	High flexibility scale and performance Deploy Big apps applications	Exceptionally Fast. Easily insert huge amounts of data. Operations are atomic
Availability & Partition Tolerance	Consistency and Partition Tolerance	Consistency and Partition Tolerance
SoundCloud - Netflix - Apple	eBay - Adobe- Google - Facebook	Twitter - GitHub - Snapchat

What is not provided by NoSQL

- Joins & Group by [SQL]
- Integration with applications that are based on SQL

Where to use NoSQL

- ❑ NoSQL Data storage systems makes sense for applications that **process very large semi-structured data** –like Log Analysis, Social Networking Feeds, Time-based data.
- ❑ To **improve programmer productivity** by using a database that better matches an application's needs.
- ❑ To **improve data access** performance via some combination of handling **larger data volumes, reducing latency**, and improving throughput.

Conclusion

All the choices provided by the rise of NoSQL databases does not mean the demise of RDBMS databases as Relational databases are a powerful tool.

We are entering an era of **Polyglot** persistence, a technique that uses different data storage technologies to handle varying data storage needs. It can apply across an enterprise or within an individual application.

References

- https://www.youtube.com/watch?v=uD3p_rZPBUQ
- https://www.slideshare.net/ramakantsoni/presentation-on-no-sql
- https://www.mongodb.com/nosql-inline
- https://www.mssqltips.com/sqlservertip/5980/sql-and-nosql-database-features-and-differences/
- https://aws.amazon.com/nosql/
- https://www.youtube.com/watch?v=Jw1iFr4v58M
- https://www.youtube.com/watch?v=QlqylUeqeis&t=2s

Thank You

Habiba Ahmed Magdy
habiba.hamad97@eng-st.cu.edu.eg

PaxeraHealth
Transforming Healthcare Through Innovation