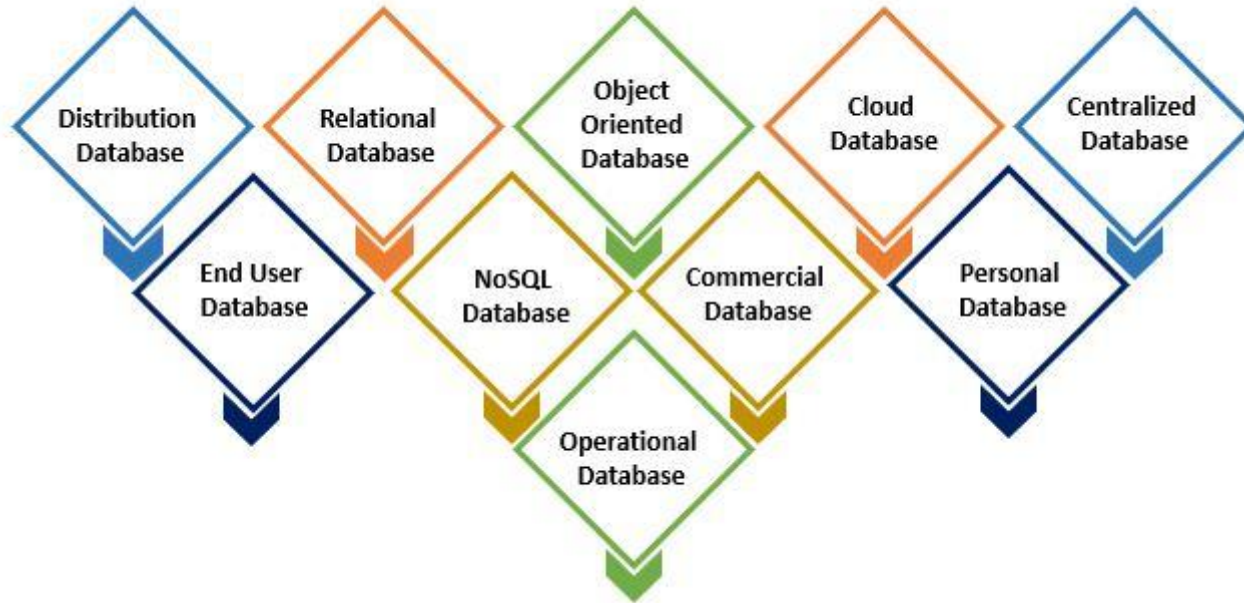
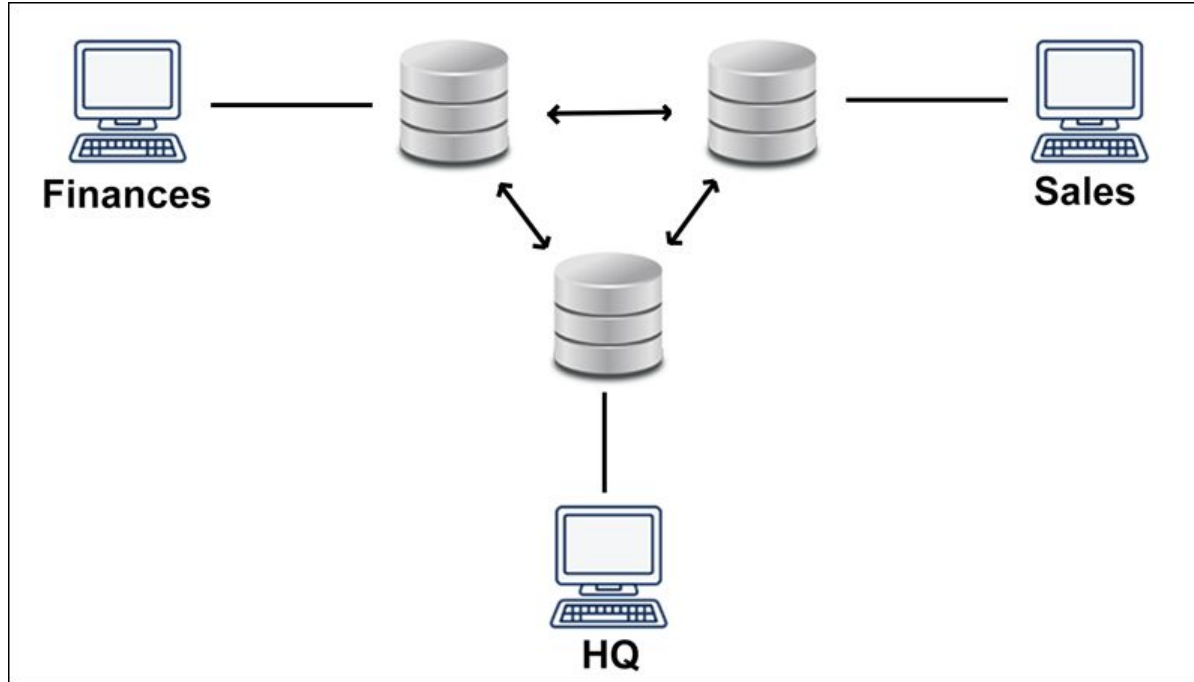


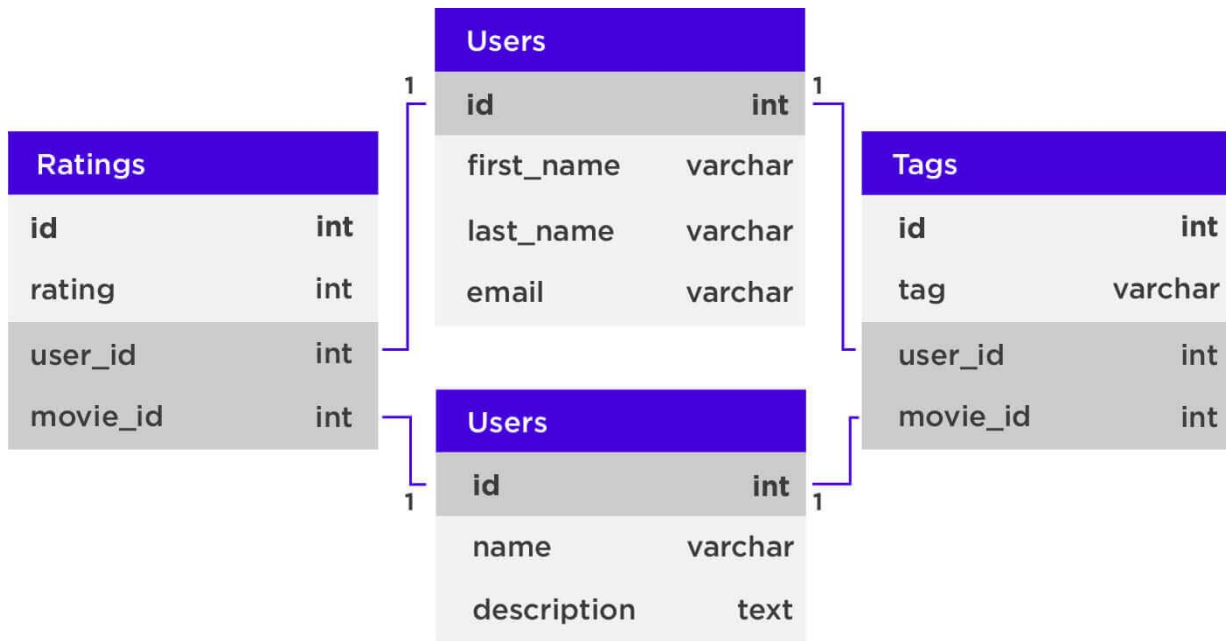
Intro to DBMS



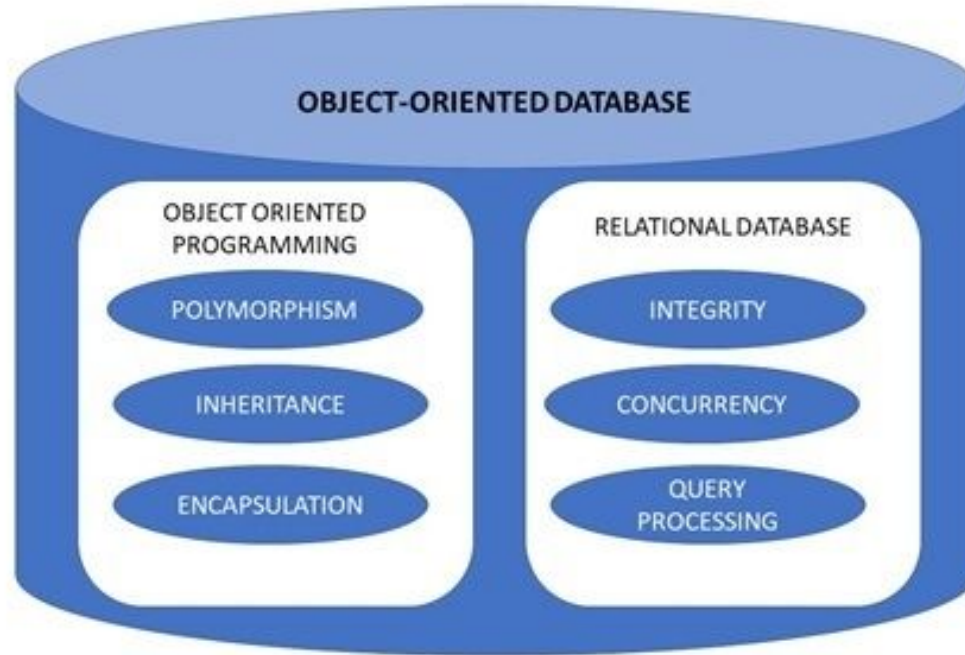
Distributed Database



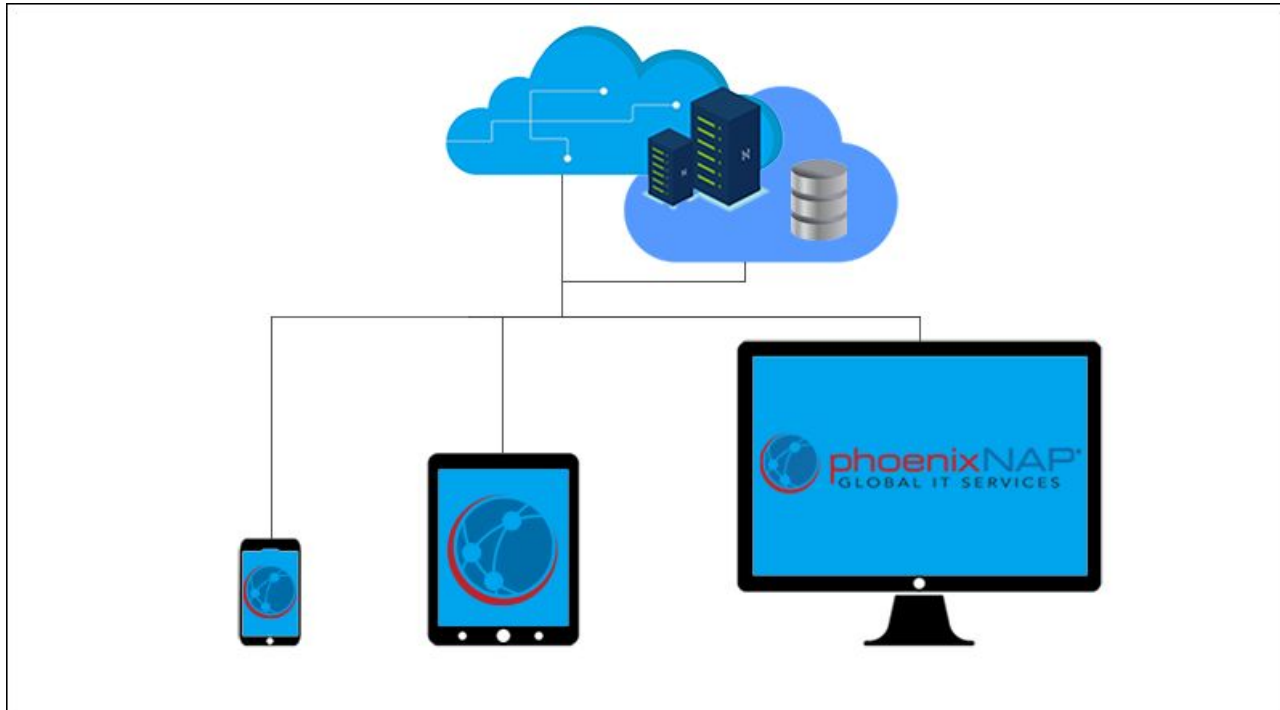
Relational Database



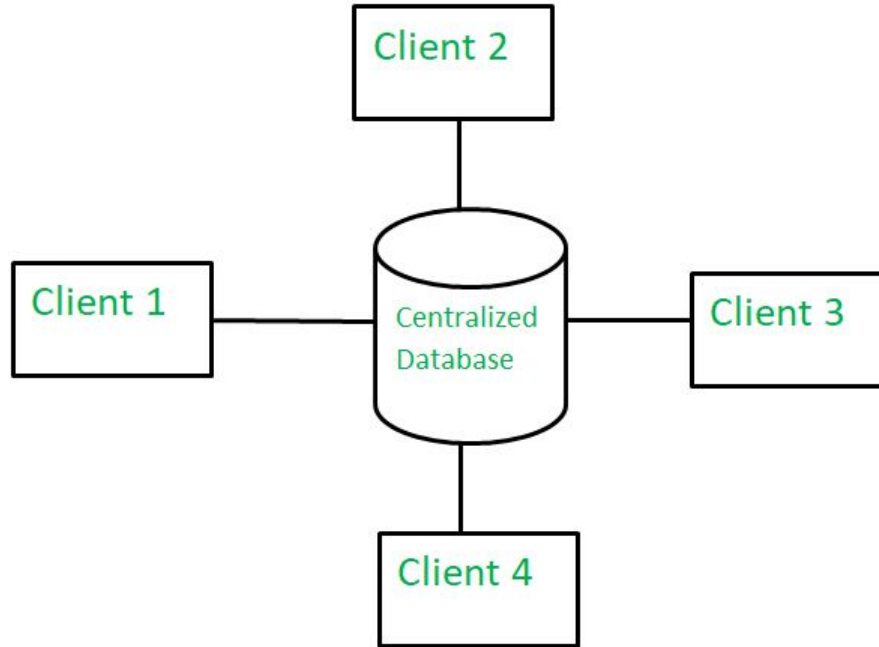
Object Oriented Database



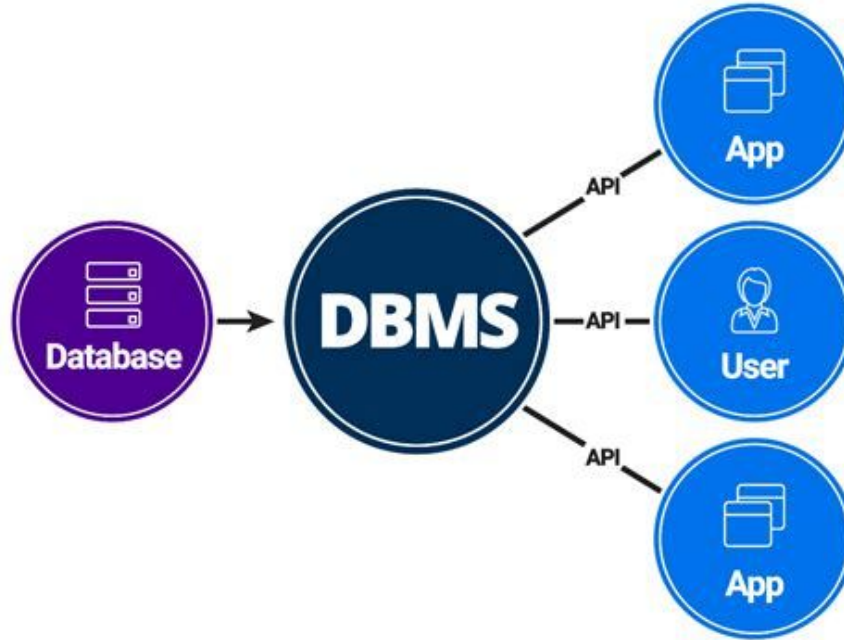
Cloud Database



Centralized Database



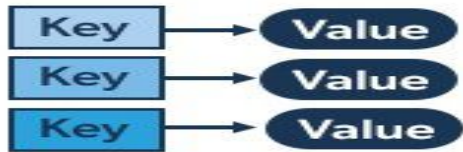
End user Database



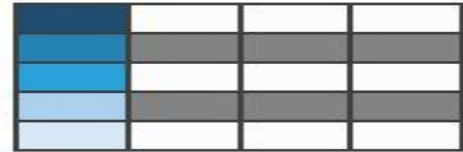
NoSQL Database

NoSQL

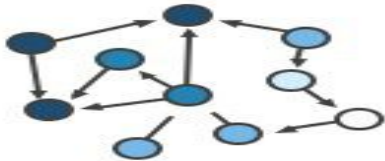
Key-Value



Column-Family



Graph



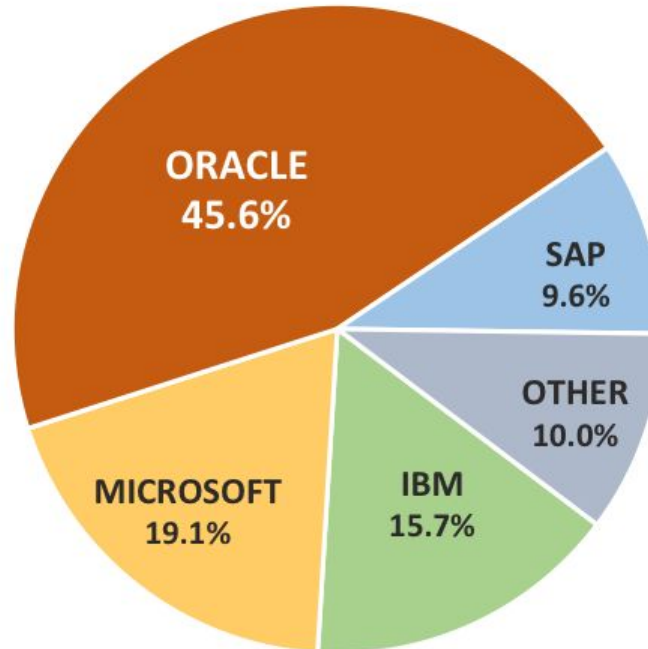
Document



Commercial Database

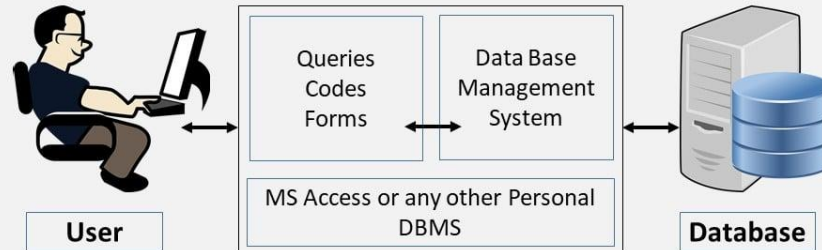
2016 COMMERCIAL DATABASE MARKET SHARE

(Source: Gartner, Inc. 2016)



Personal Database

PERSONAL DATABASE



Operational Database



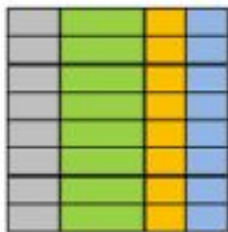
Intro to NoSQL

Definition

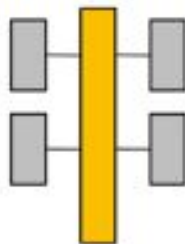
- NoSQL Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for **distributed data** stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook and Google collect terabytes of user data

SQL Database

Relational

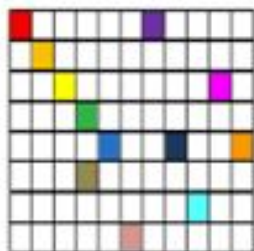


Analytical (OLAP)

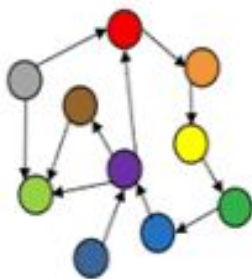


NoSQL Database

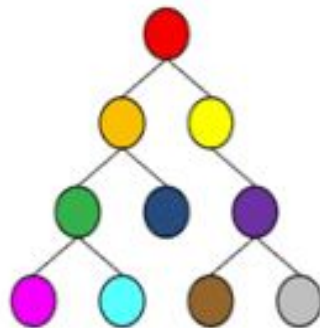
Column-Family



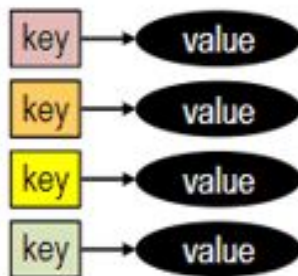
Graph



Document

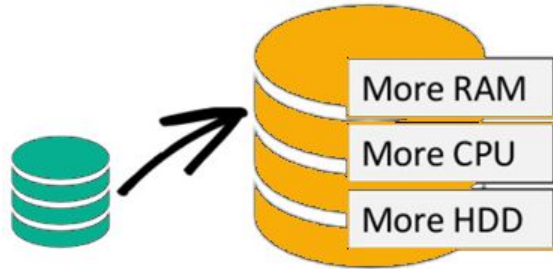


Key-Value

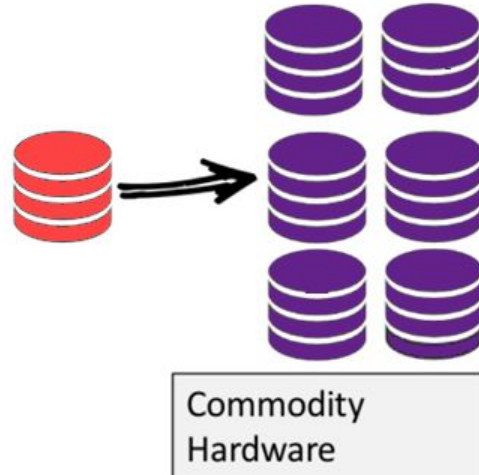


Why NoSQL

Scale-Up (*vertical scaling*):



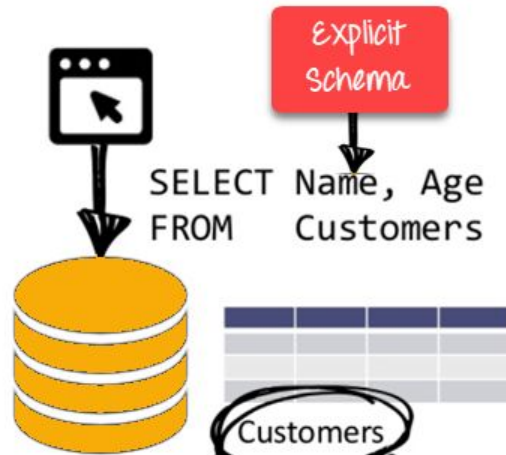
Scale-Out (*horizontal scaling*):



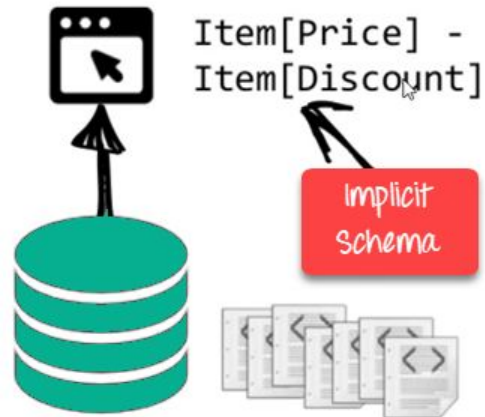
Schema-free

- NoSQL databases are either schema-free or have relaxed schemas
- Do not require any sort of definition of the schema of the data
- Offers heterogeneous structures of data in the same domain

RDBMS:



NoSQL DB:



NoSQL is Schema-Free

Types of NoSQL Databases:

- Key-value Pair Based
- Column-oriented Graph
- Graphs based
- Document-oriented

Key Value



Example:
Riak, Tokyo Cabinet, Redis
server, Memcached,
Scalaris

Document-Based



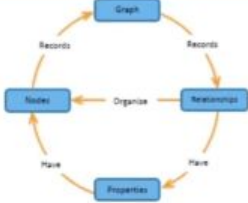
Example:
MongoDB, CouchDB,
OrientDB, RavenDB

Column-Based



Example:
BigTable, Cassandra,
Hbase,
Hypertable

Graph-Based



Example:
Neo4J, InfoGrid, Infinite
Graph, Flock DB

Key Value Pair Based

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.

Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

For example, a key-value pair may contain a key like “Website” associated with a value like “Guru99”.

Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

Column-based

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.

ColumnFamily			
Row Key	Column Name		
	Key	Key	Key
	Value	Value	Value
	Column Name		
	Key	Key	Key
	Value	Value	Value

Column based NoSQL database

Document-Oriented:

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.

Col1	Col2	Col3	Col4
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data

Document 1

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```

Document 2

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```

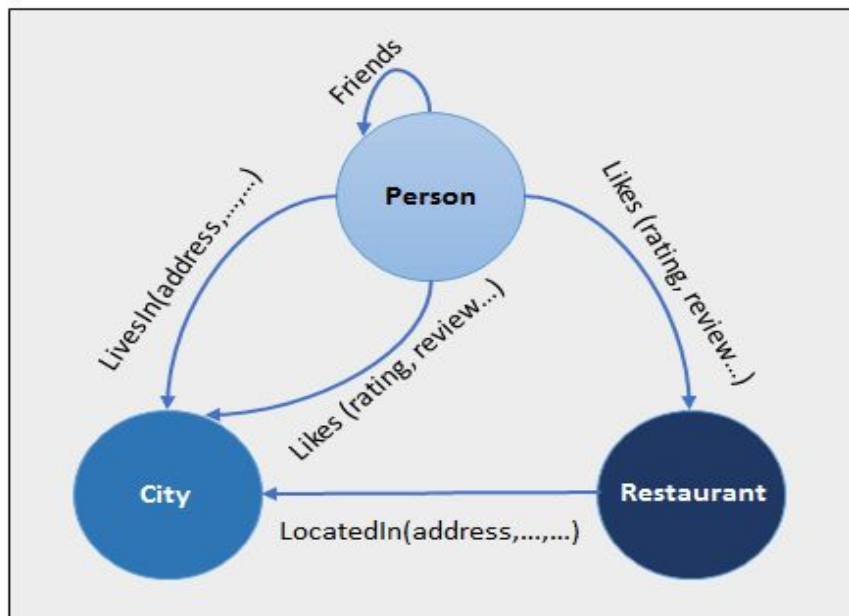
Document 3

```
{  
  "prop1": data,  
  "prop2": data,  
  "prop3": data,  
  "prop4": data  
}
```

Relational Vs. Document

Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.



The Value of Relational Databases

Relational databases have become such an embedded part of our computing culture that it's easy to take them for granted. It's therefore useful to revisit the benefits they provide

- 1) Persistent Data
- 2) Concurrency
- 3) Integrations
- 4) Standard Model

Impedance Mismatch

Impedance mismatch is the term used to refer to the problems that occurs due to differences between the database model and the programming language model.

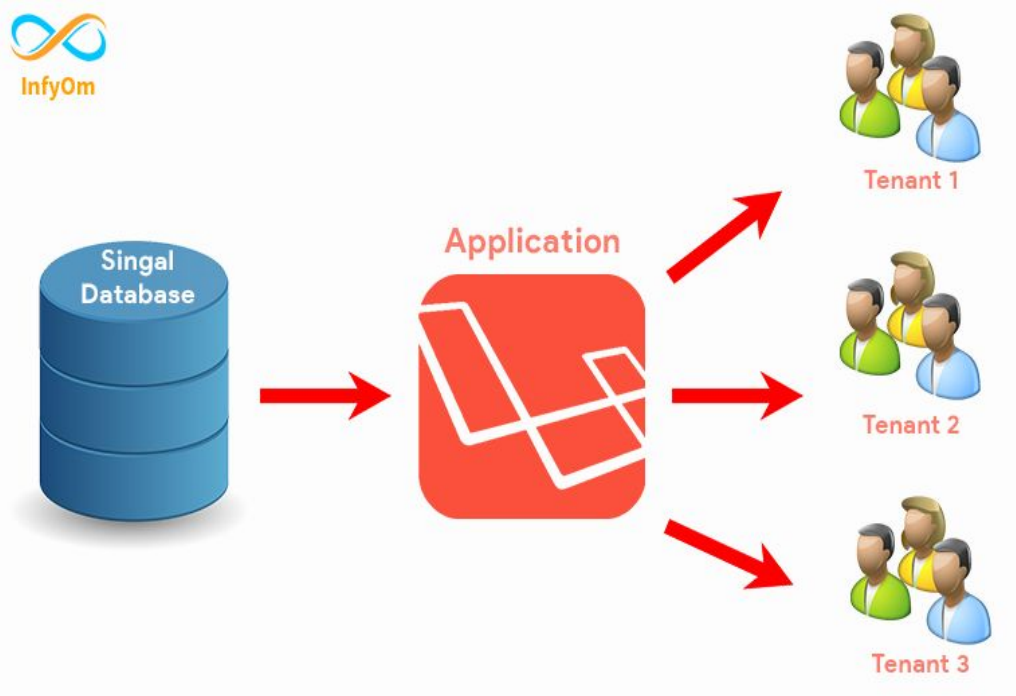
The practical relational model has 3 components these are:

- 1) Attributes and their data types
- 2) Tuples
- 3) Tables

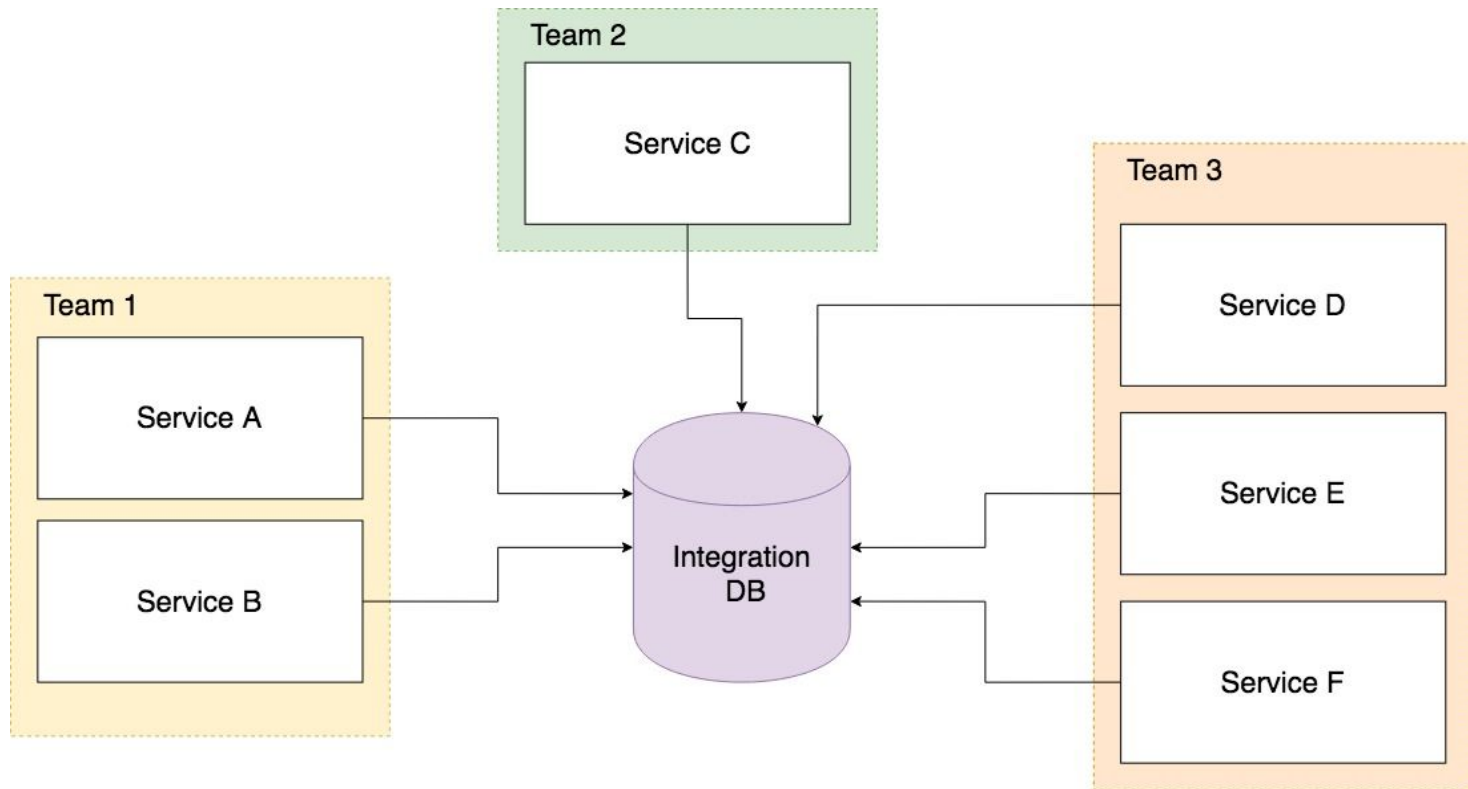
Data type mismatch

Results of most queries are sets or multisets of tuples and each tuple is formed of a sequence of attribute values

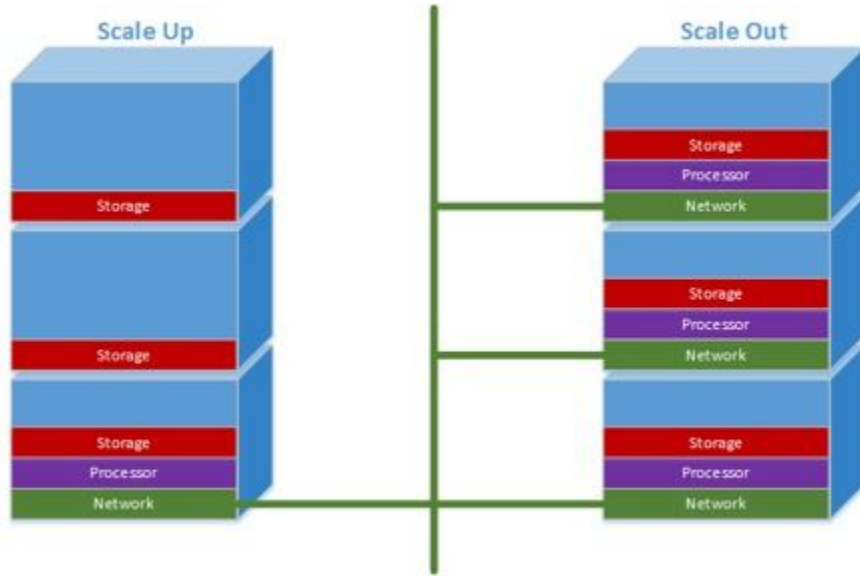
Application Database



Integration Database



Attack of the clusters



The Emergence of NoSQL

- The word first came in 90's as name of the open-source relational database Strozzi NoSQL developed by Carlo Strozzi, it doesn't use SQL for querying.
- meetup on June 11, 2009 by Johan Oskarsson for the projects inspired from BigTable and Dynamo.
- The tem “NoSQL” was suggested by Eric Evans
- There are two primary reasons for considering NoSQL. One is to handle data access with sizes and performance that demand a cluster; the other is to improve the productivity of application development by using a more convenient data interaction style.

- ❑ Google developed a NOSQL system known as **BigTable**, used in many of Google's application that requires vast amounts of data storage, such as Gmail, Google Maps, and Web site indexing. This system uses concepts from **column-based** or wide column stores.
- ❑ Amazon developed a NOSQL system called DynamoDB that is available through Amazon's cloud services. This system uses concepts from **key-value stores**.
- ❑ Facebook developed a NOSQL systems called Cassandra, which is now open source and known as Apache Cassandra. This system uses concepts from both **key-value stores** and **column-based systems**.
- ❑ Other software companies started developing their own solutions. For example, MongoDB and CouchDB, which are classified as document-based NOSQL systems or **document stores**.
- ❑ Another category of NOSQL systems is the graph-based NOSQL Systems, or **graph databases**; these includes Neo4J and GraphBase.

Key points

- ❖ Relational databases have been a successful technology for twenty years, providing persistence, concurrency control, and an integration mechanism.
- ❖ Application developers have been frustrated with the impedance mismatch between the relational model and the in-memory data structures.
- ❖ There is a movement away from using databases as integration points towards encapsulating databases within applications and integrating through services.
- ❖ The vital factor for a change in data storage was the need to support large volumes of data by running on clusters. Relational databases are not designed to run efficiently on clusters.
- ❖ NoSQL is an accidental neologism. There is no prescriptive definition—all you can make is an observation of common characteristics.
- ❖ The common characteristics of NoSQL databases are
 - ❖ Not using the relational model
 - ❖ Running well on clusters
 - ❖ Open-source
 - ❖ Built for the 21st century web estates
 - ❖ Schemaless
- ❖ The most important result of the rise of NoSQL is Polyglot Persistence.