Lecture 3:
Big Data Storage

Big Data Systems Design

In Last Lecture

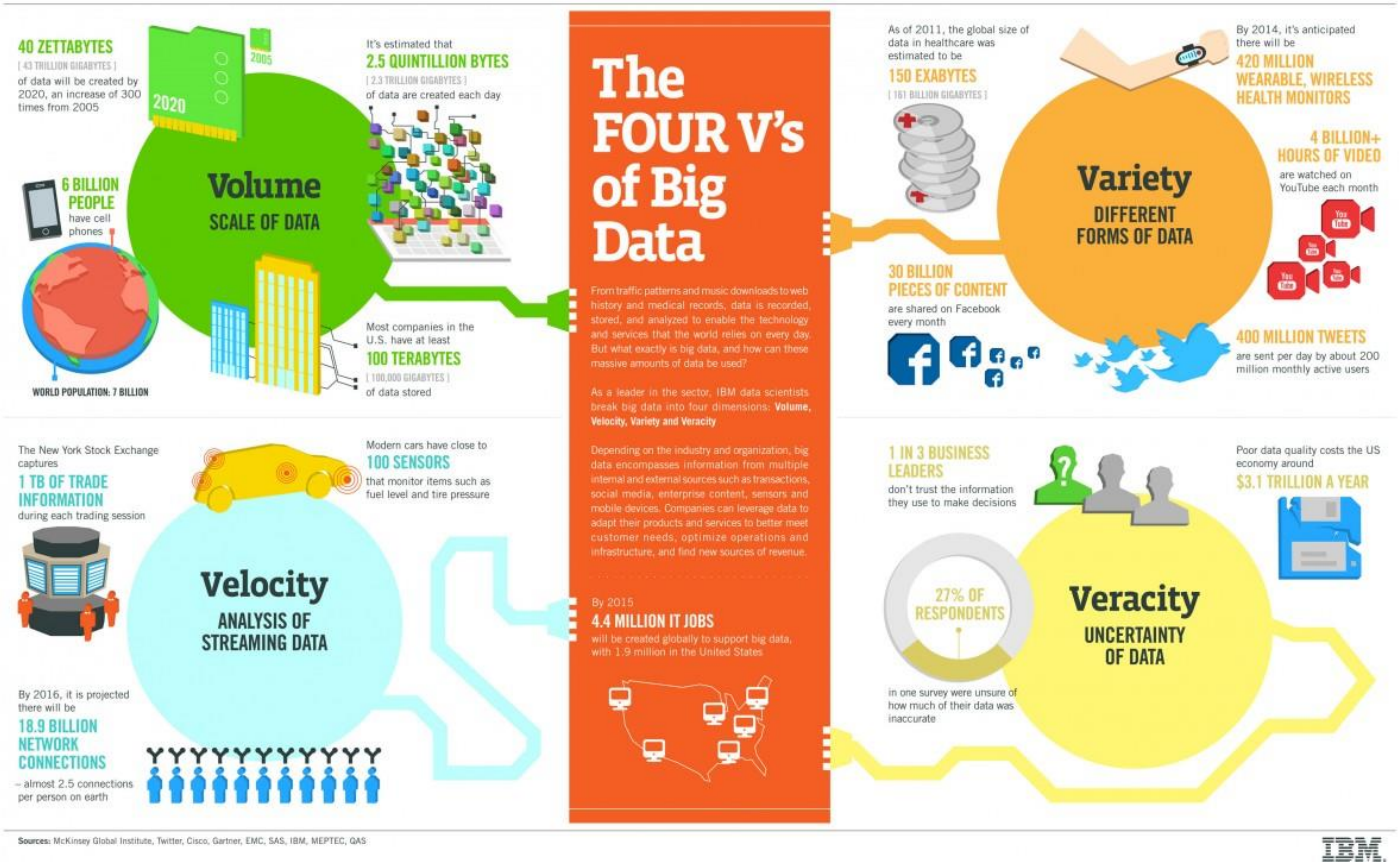
- Intro to Big Data
  - Data most commonly refers to information that is transmitted or stored electronically
  - 4Vs of Big Data

- What is Big Data Analytics
  - Big Data analytics is a process used to extract meaningful insights, such as hidden patterns, unknown correlations, market trends, customer preferences

- Preparing working environment
  - MongoDB Installation

In Last Lecture

❖ What is Big Data?



The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015 **4.4 MILLION IT JOBS** will be created globally to support big data, with 1.9 million in the United States

Volume
SCALE OF DATA

40 ZETTABYTES [43 TRILLION GIGABYTES] of data will be created by 2020, an increase of 300 times from 2005

6 BILLION PEOPLE have cell phones

WORLD POPULATION: 7 BILLION

It's estimated that **2.5 QUINTILLION BYTES** [2.3 TRILLION GIGABYTES] of data are created each day

Most companies in the U.S. have at least **100 TERABYTES** [100,000 GIGABYTES] of data stored

Velocity
ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures **1 TB OF TRADE INFORMATION** during each trading session

Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be **18.9 BILLION NETWORK CONNECTIONS** – almost 2.5 connections per person on earth

Variety
DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be **150 EXABYTES** [161 BILLION GIGABYTES]

By 2014, it's anticipated there will be **420 MILLION WEARABLE, WIRELESS HEALTH MONITORS**

4 BILLION+ HOURS OF VIDEO are watched on YouTube each month

30 BILLION PIECES OF CONTENT are shared on Facebook every month

400 MILLION TWEETS are sent per day by about 200 million monthly active users

Veracity
UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS don't trust the information they use to make decisions

27% OF RESPONDENTS in one survey were unsure of how much of their data was inaccurate

Poor data quality costs the US economy around **$3.1 TRILLION A YEAR**

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

Table of Contents

❖ **Part 1.**

- Centralized storage

❖ **Part 2.**

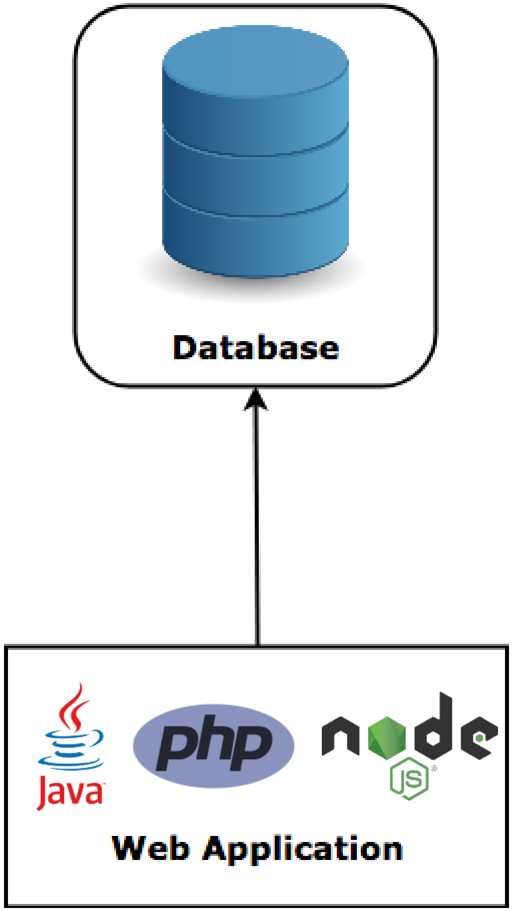
- Decentralized storage

❖ **Part 3.**

- NoSQL databases

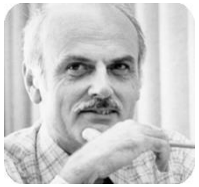
Centralized Storage

- Centralized storage
  - Data is stored on the database of one single machine
  - Whenever you want to read/insert information in it you communicate with that machine directly
  - Relational databases



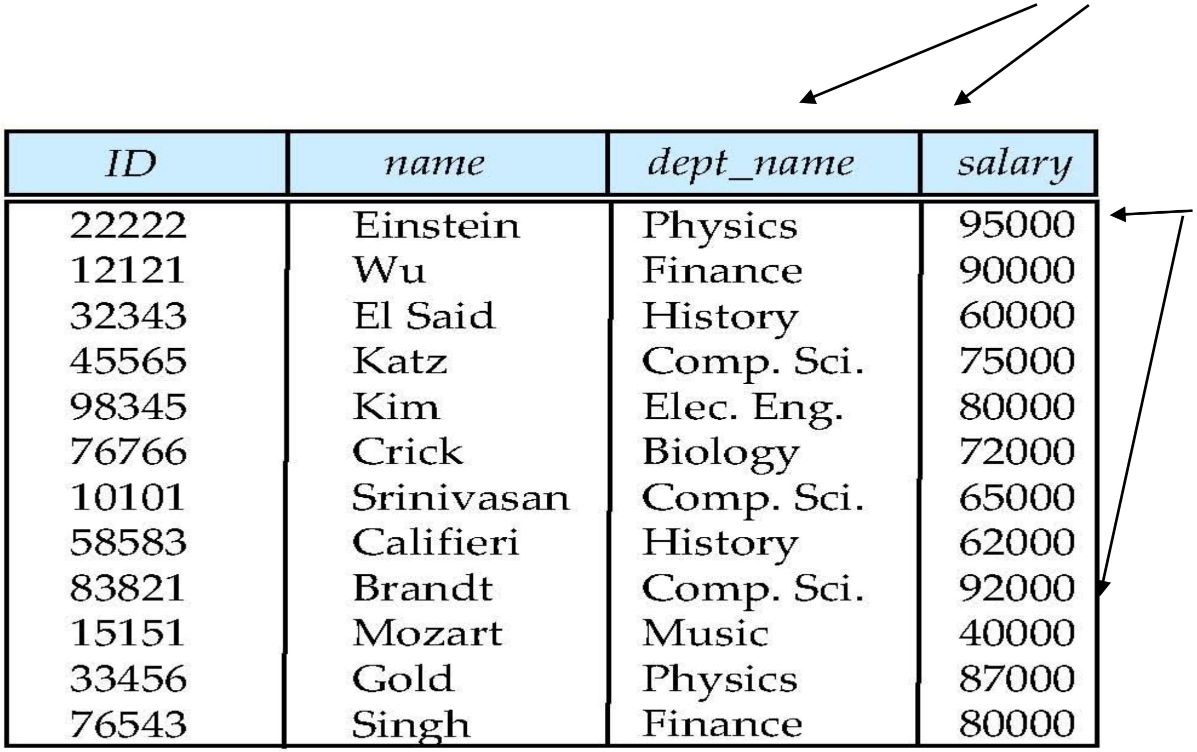
Relational model

- ❖ What is relational model?
  - Relation = table = schema



Ted Codd
Turing Award 1981

Columns



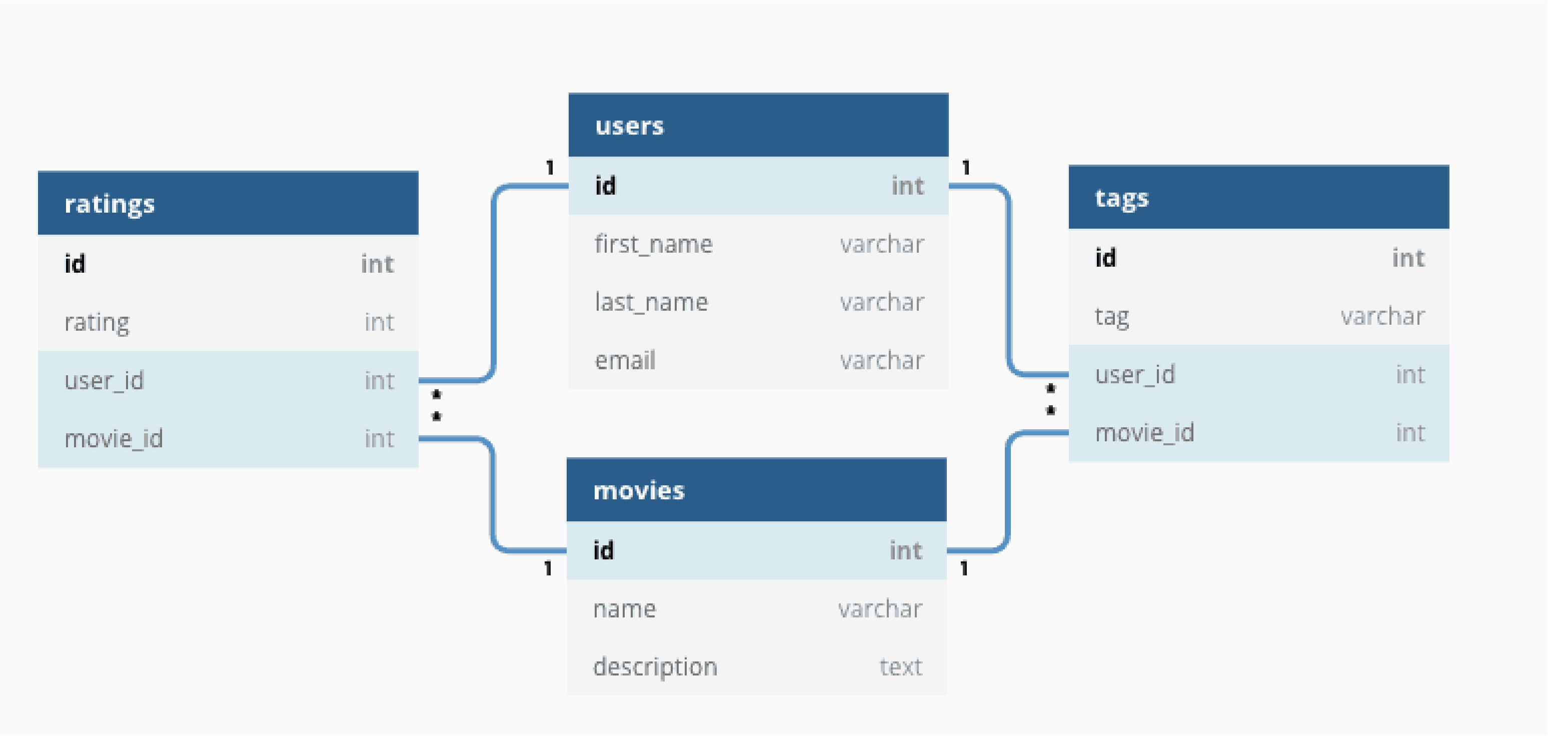
Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Relational model

- What is relational model?
  - Allows related data to be stored across multiple **tables**, and linked by establishing a **relationship** between the tables



ratings	
id	int
rating	int
user_id	int
movie_id	int

users	
id	int
first_name	varchar
last_name	varchar
email	varchar

tags	
id	int
tag	varchar
user_id	int
movie_id	int

movies	
id	int
name	varchar
description	text

Relational Database

- ❖ What is DBMS?
  - ▪ Software for creating and managing databases

ORACLE
foundation
firebird
PostgreSQL
Microsoft
SQL Server 2008
IBM
DB2
SQLite
MySQL
1-2-3
TERADATA
CSV

Relational Database

- Why not relational database?

MySQL

PostgreSQL

Microsoft SQL Server

ORACLE

Inflexible

Primarily suitable for structured data and not flexible for other types

Velocity

Designed for steady data retention, rather than for rapid growth

No scalability

Don’t scale well to very large size

Weak SQL

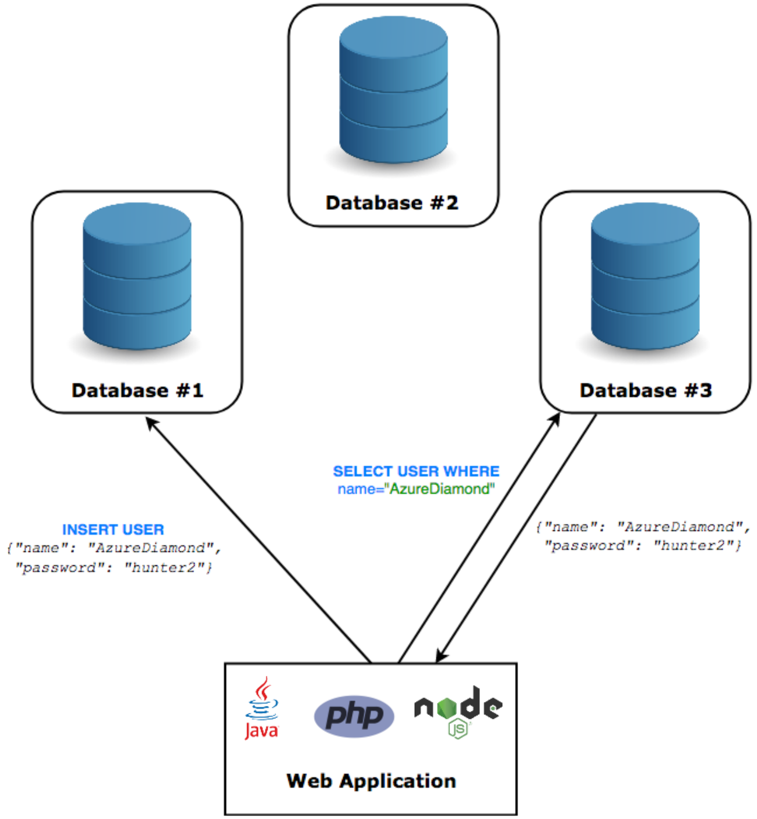
Difficult to implement certain kinds of basic queries using SQL

Part 2

DECENTRALIZED STORAGE

Decentralized Storage

- Decentralized storage
  - Database run on multiple machines at the same time
  - User does not know if he is talking to single machine or multiple machines
  - NoSQL databases



What is NoSQL?

- What is NoSQL?
  - Some claim for it to mean "**No SQL**"
    - Meaning that the system doesn't use SQL – it uses an alternative query language)
  - The definition is sometimes expanded to mean **"Not only SQL"**
    - Meaning that the system uses SQL along with other technologies/query languages
  - Many argue that the one thing all NoSQL databases have in common is that they're **non-relational**
    - **"NoREL" is a more suitable name**

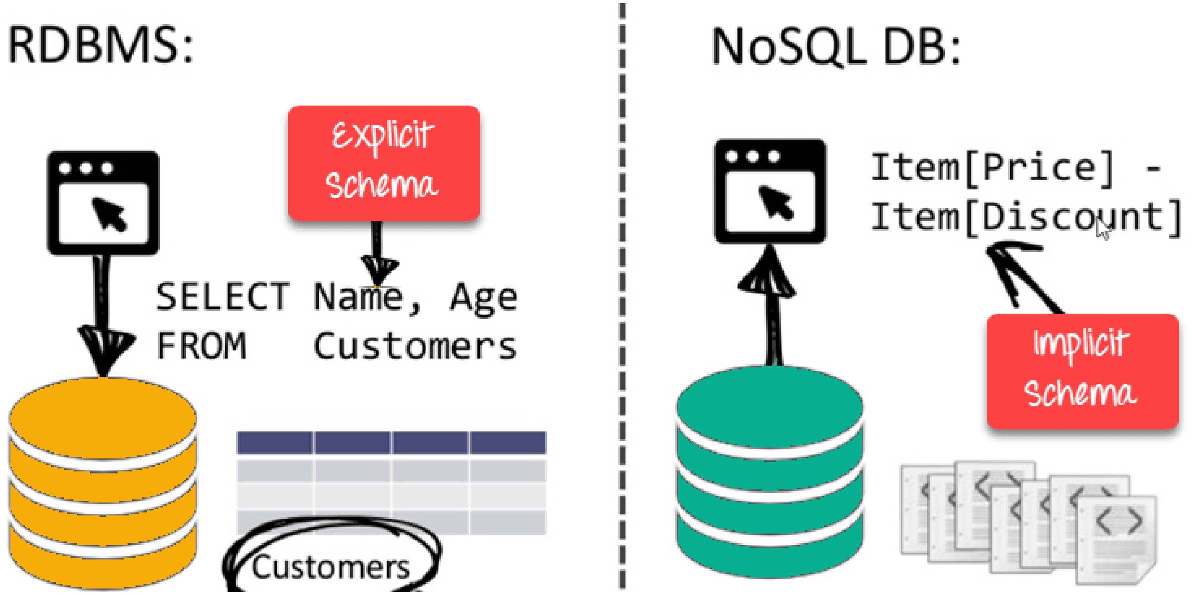
History of NoSQL

- History of NoSQL
  - 1998- Carlo Strozzi use the term NoSQL for his lightweight, open-source relational database
  - 2000- Graph database Neo4j is launched
  - 2004- Google BigTable is launched
  - 2005- CouchDB is launched
  - 2007- The research paper on Amazon Dynamo is released
  - 2008- Facebooks open sources the Cassandra project
  - 2009- The term NoSQL was reintroduced

Why NoSQL?

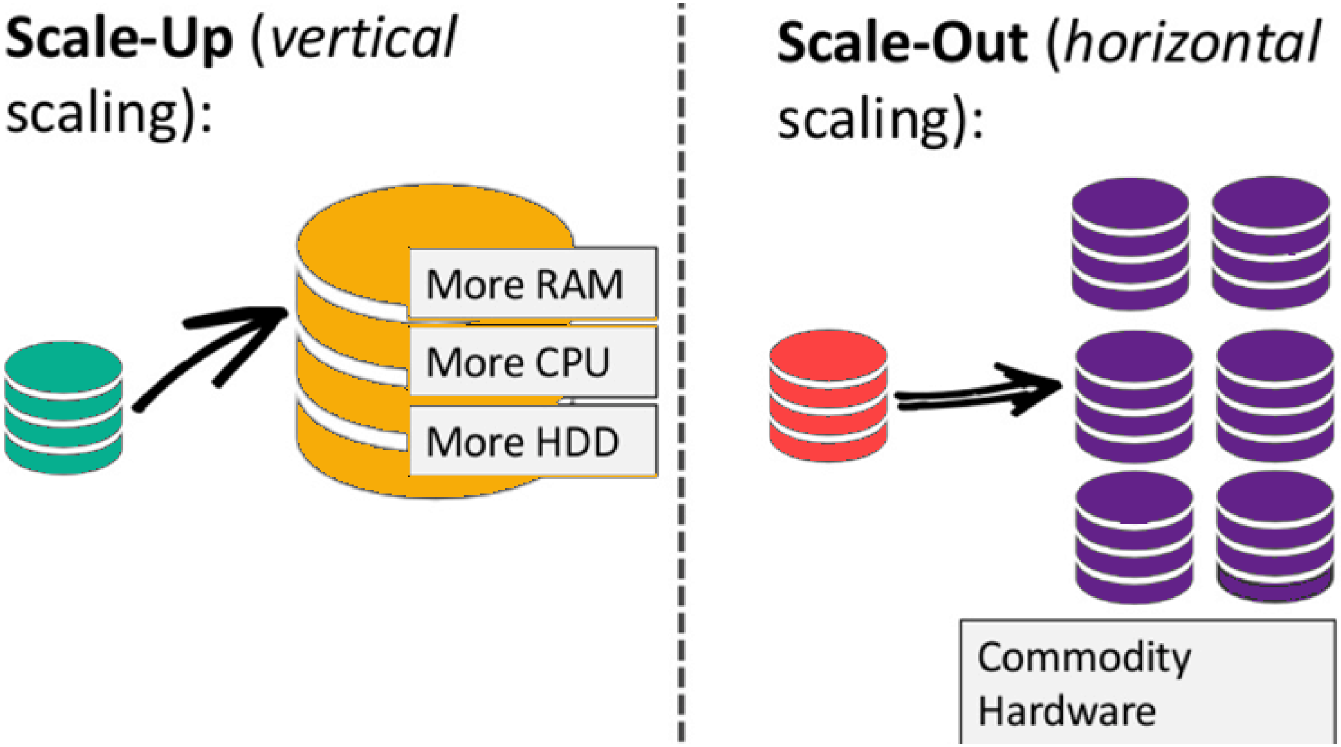
❖ Features of a NoSQL

- Flexible
  - NoSQL databases are either schema-free or have relaxed schemas
    - Do not require any sort of definition of the schema of the data



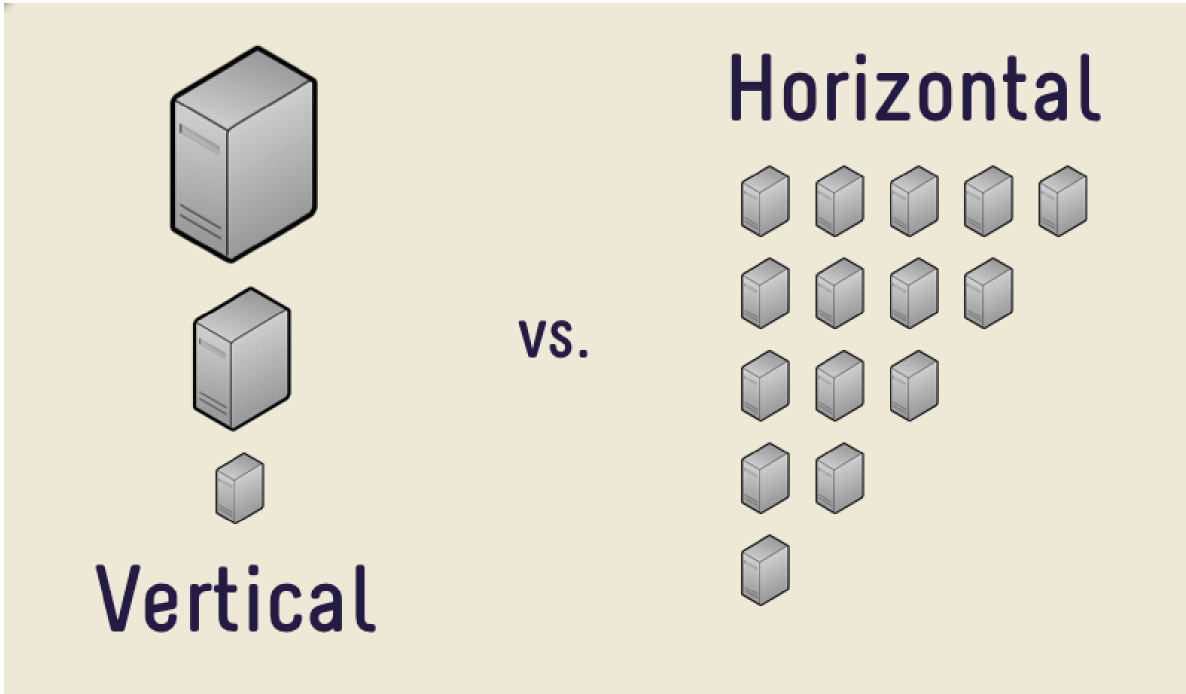
Why NoSQL?

- Features of a NoSQL
  - Scalability


Scale-Up (*vertical* scaling):
More RAM
More CPU
More HDD
Scale-Out (*horizontal* scaling):
Commodity Hardware

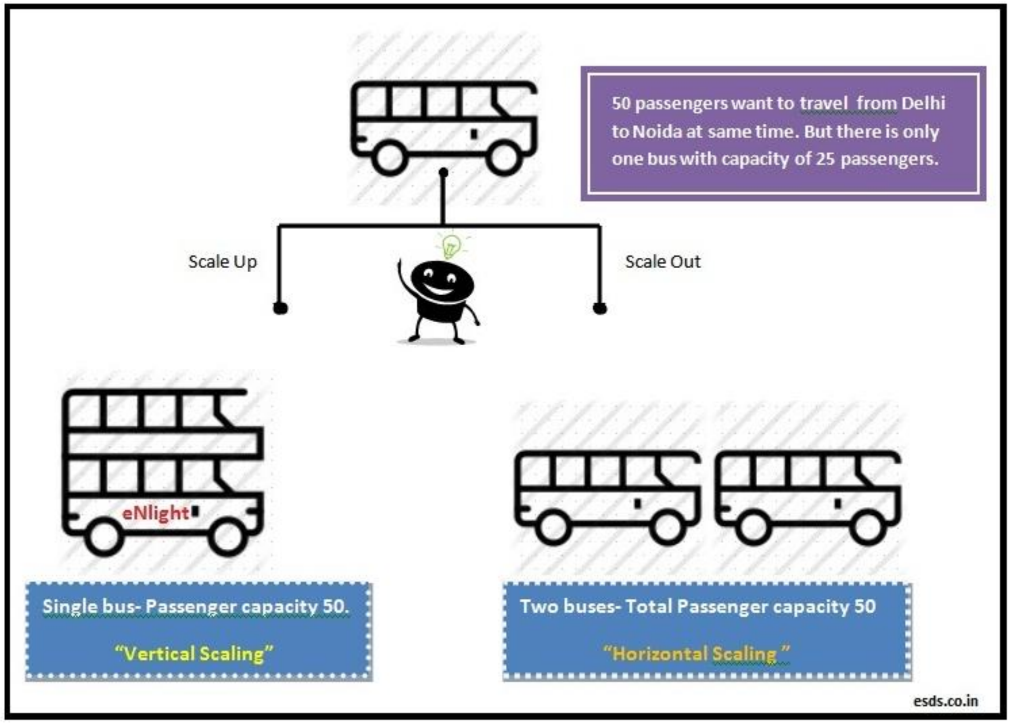
Why NoSQL?

- Features of a NoSQL
  - Scalability
    - Horizontal scaling vs. vertical scaling



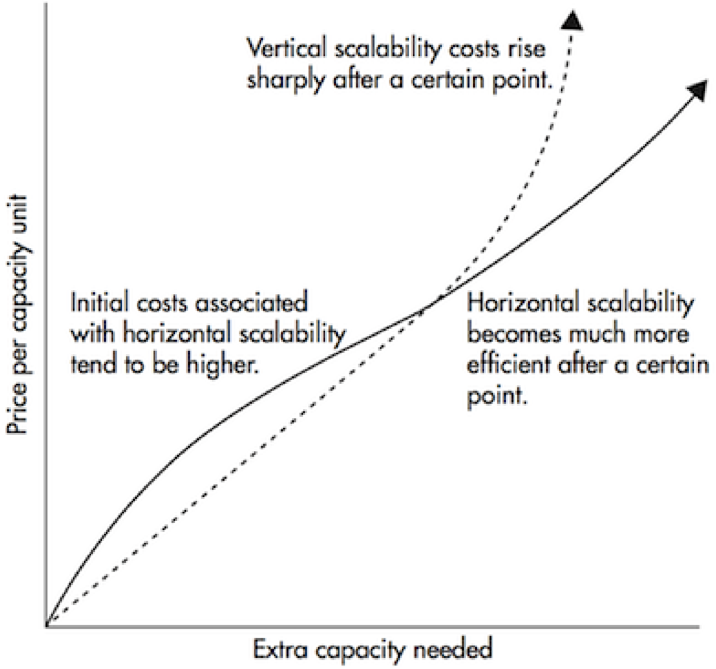
Why NoSQL?

- Features of a NoSQL
  - Scalability
    - Horizontal scaling vs. vertical scaling


50 passengers want to travel from Delhi to Noida at same time. But there is only one bus with capacity of 25 passengers.
Scale Up
Scale Out
eNlight
Single bus- Passenger capacity 50.
"Vertical Scaling"
Two buses- Total Passenger capacity 50
"Horizontal Scaling"
esds.co.in

Why NoSQL?

- Features of a NoSQL
  - Scalability
    - Horizontal scaling becomes much cheaper after a certain threshold



Relational Database

- Features of a NoSQL
  - Scalability

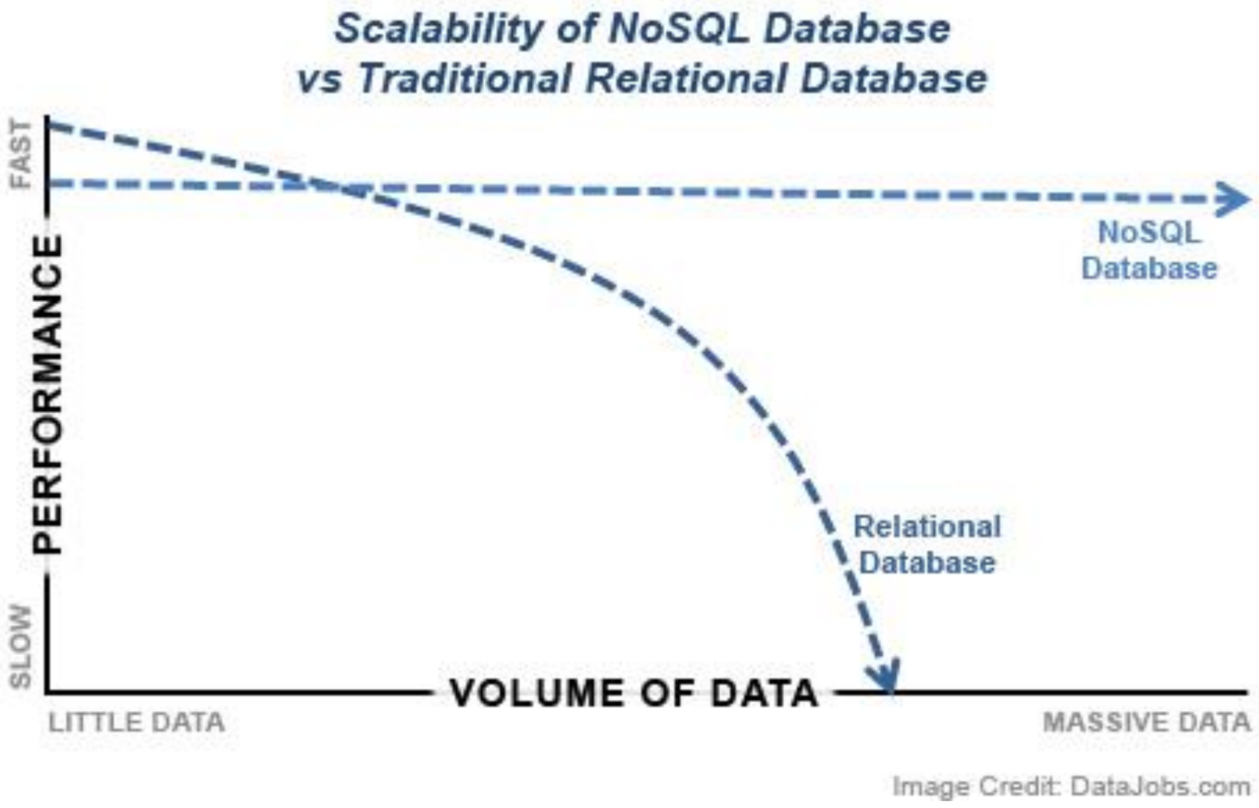
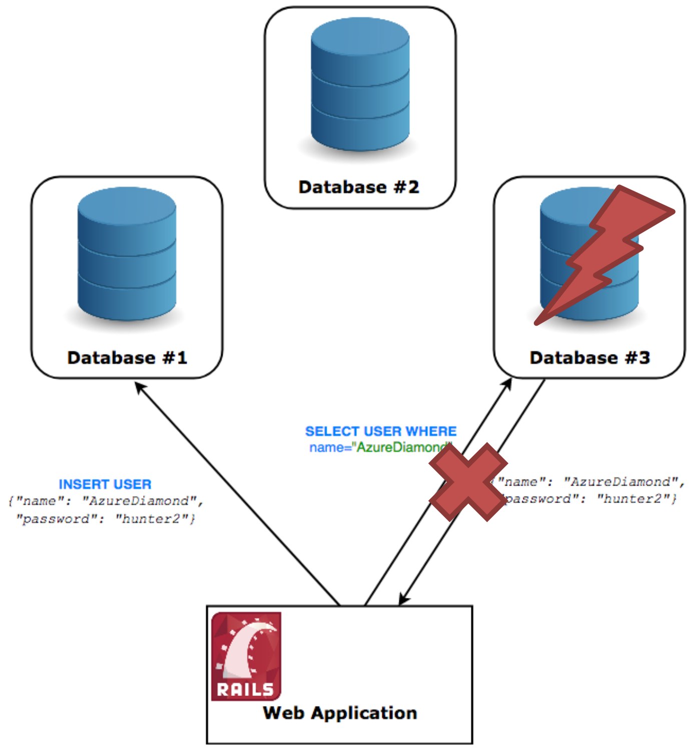


Image Credit: DataJobs.com

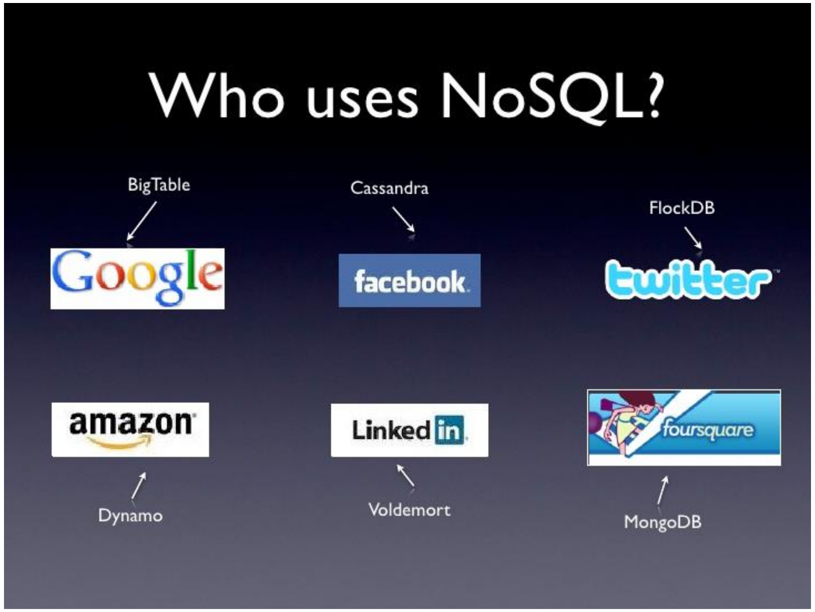
Why NoSQL?

- Fault tolerance
  - A cluster of several machines is inherently more fault-tolerant than a single machine
  - It improves availability of your system



Why NoSQL?

- Features of NoSQL
  - Large companies change from traditional schema-based DBMS to NoSQL database
    - Apple is known to use 75,000 Apache Cassandra nodes storing over 10 petabytes of data



Why NoSQL?

- ❖ Features of a NoSQL
  - ▪ Open Source
    - Most of NoSQL databases are open source

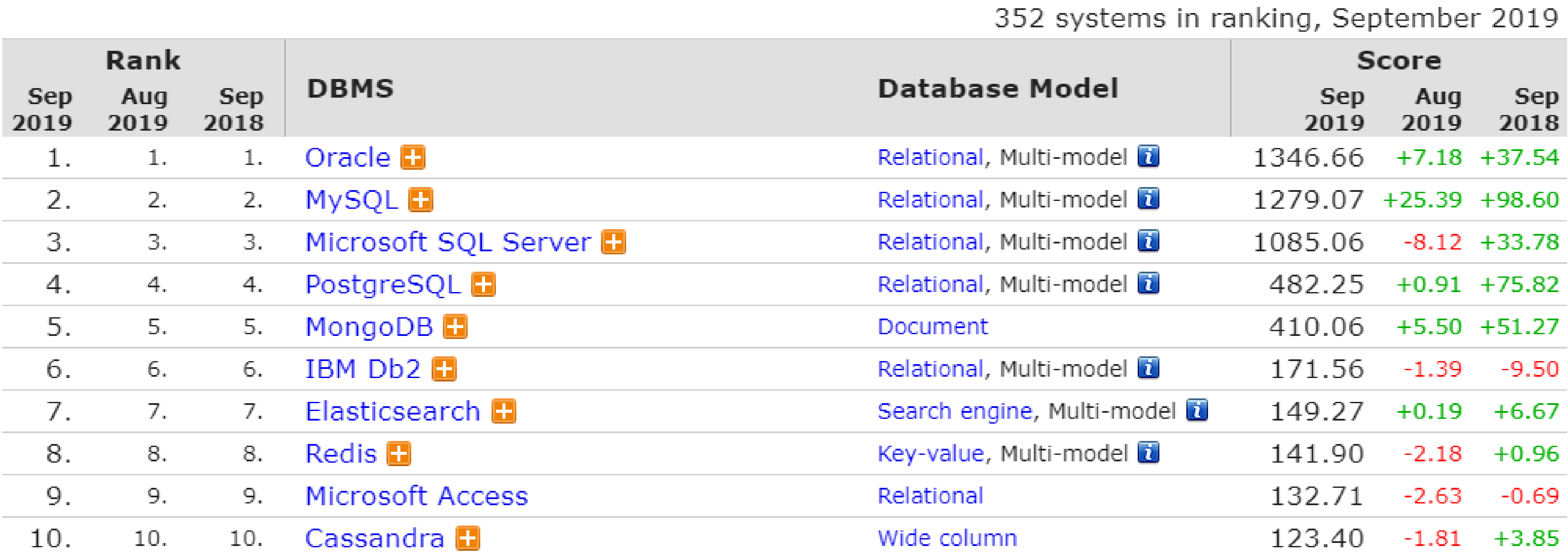
APACHE HBASE
Cassandra
riak
CouchDB relax
mongoDB
HYPERTABLE INC
Neo4j
redis

Part 3

NOSQL DATABASES

NoSQL Databases

- Popularity rise of NoSQL



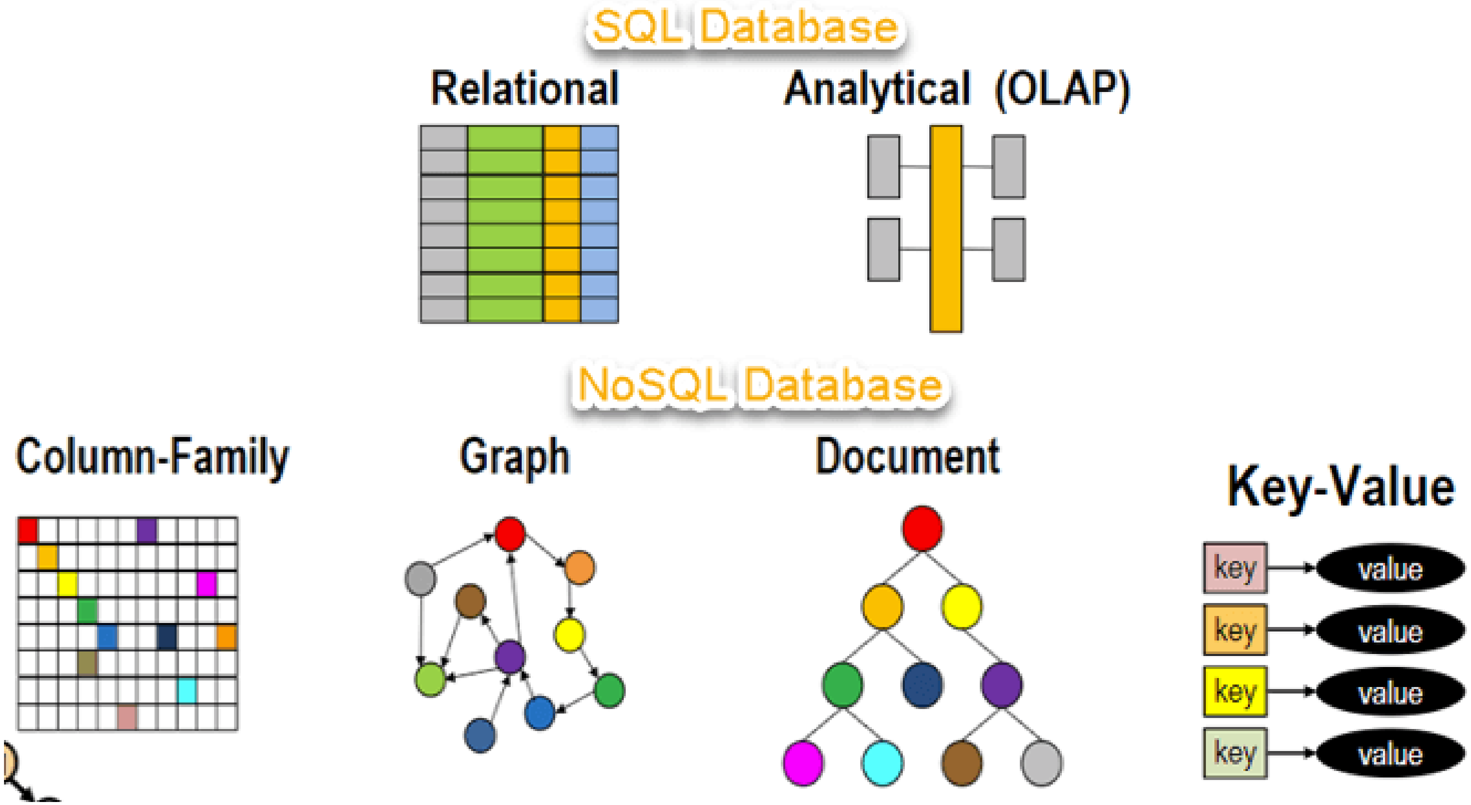
352 systems in ranking, September 2019

Rank			DBMS	Database Model	Score		
Sep 2019	Aug 2019	Sep 2018			Sep 2019	Aug 2019	Sep 2018
1.	1.	1.	Oracle	Relational, Multi-model	1346.66	+7.18	+37.54
2.	2.	2.	MySQL	Relational, Multi-model	1279.07	+25.39	+98.60
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1085.06	-8.12	+33.78
4.	4.	4.	PostgreSQL	Relational, Multi-model	482.25	+0.91	+75.82
5.	5.	5.	MongoDB	Document	410.06	+5.50	+51.27
6.	6.	6.	IBM Db2	Relational, Multi-model	171.56	-1.39	-9.50
7.	7.	7.	Elasticsearch	Search engine, Multi-model	149.27	+0.19	+6.67
8.	8.	8.	Redis	Key-value, Multi-model	141.90	-2.18	+0.96
9.	9.	9.	Microsoft Access	Relational	132.71	-2.63	-0.69
10.	10.	10.	Cassandra	Wide column	123.40	-1.81	+3.85

- See the trend here

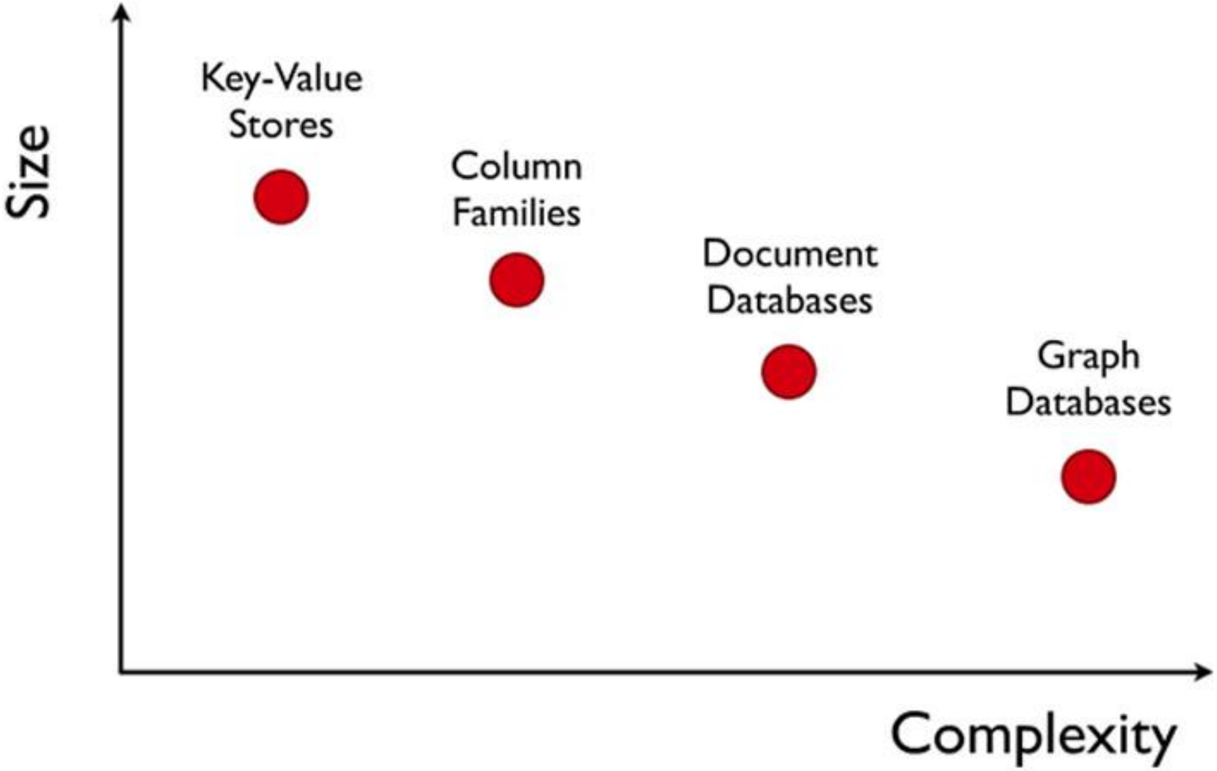
NoSQL Databases

❖ Representative NoSQL Databases



NoSQL Databases

- Representative NoSQL Databases

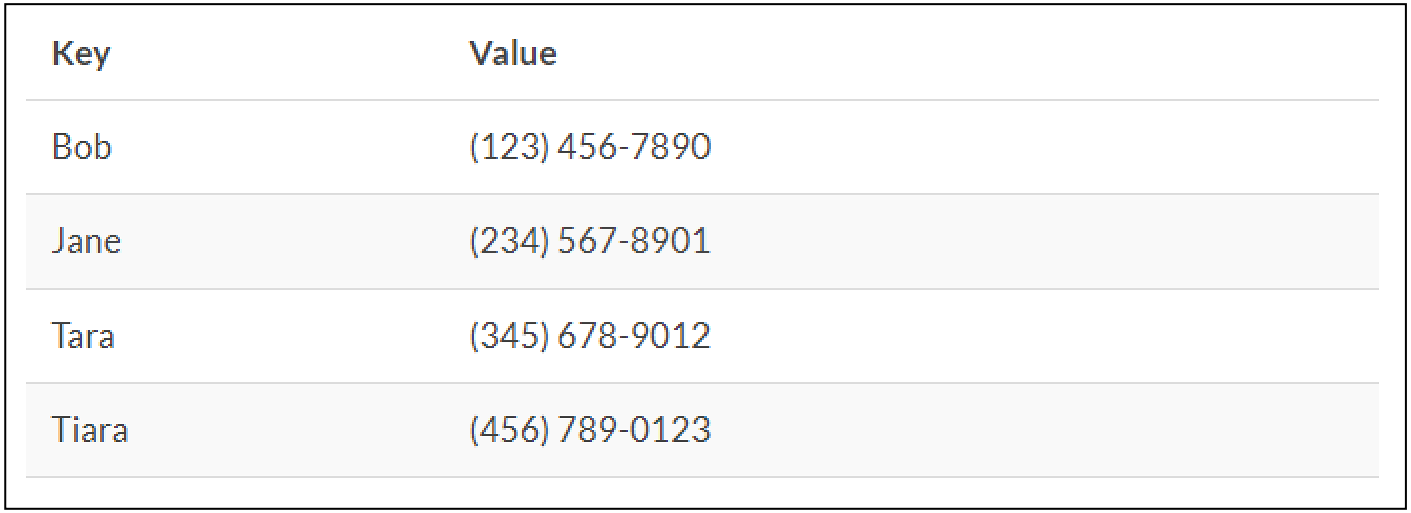


Key-Value Store Database

- What is a Key-Value Store Database?
  - A type of NoSQL database that uses a simple key/value method to store data
    - Also known as a key-value store and key-value store database
  - The key-value part refers to the fact that the database stores data as a collection of key/value pairs
    - Simple method of storing data
    - Scale well
  - The key-value pair is a well established concept in many programming languages
    - Dictionary, hash, associative array, etc

Key-Value Store Database

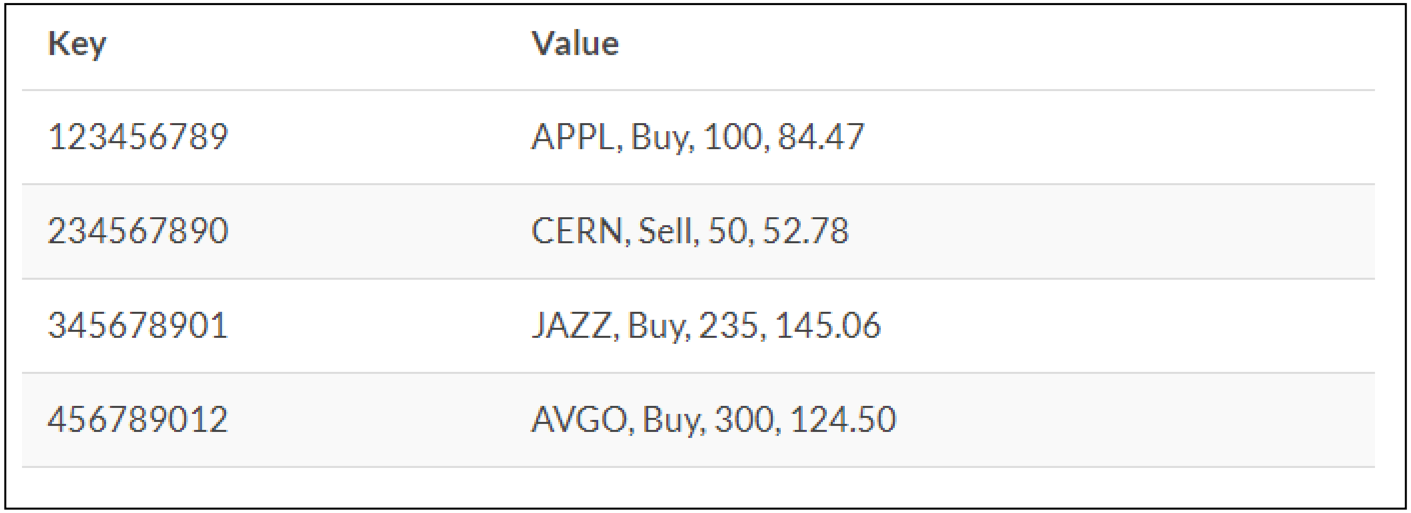
- Examples of Key-Value Stores
  - Phone Directory
    - Observe how a key-value database works



Key	Value
Bob	(123) 456-7890
Jane	(234) 567-8901
Tara	(345) 678-9012
Tiara	(456) 789-0123

Key-Value Store Database

- Examples of Key-Value Stores
  - Stock Trading
    - A list as the value



Key	Value
123456789	APPL, Buy, 100, 84.47
234567890	CERN, Sell, 50, 52.78
345678901	JAZZ, Buy, 235, 145.06
456789012	AVGO, Buy, 300, 124.50

Key-Value Store Database

- What can a key-value store database be used for?
  - User profiles and session info on a website
  - Article/Blog comments
  - Telecom directories
  - IP forwarding tables
  - Shopping cart contents on e-commerce sites
  - Product categories, details, reviews

Key-Value Database

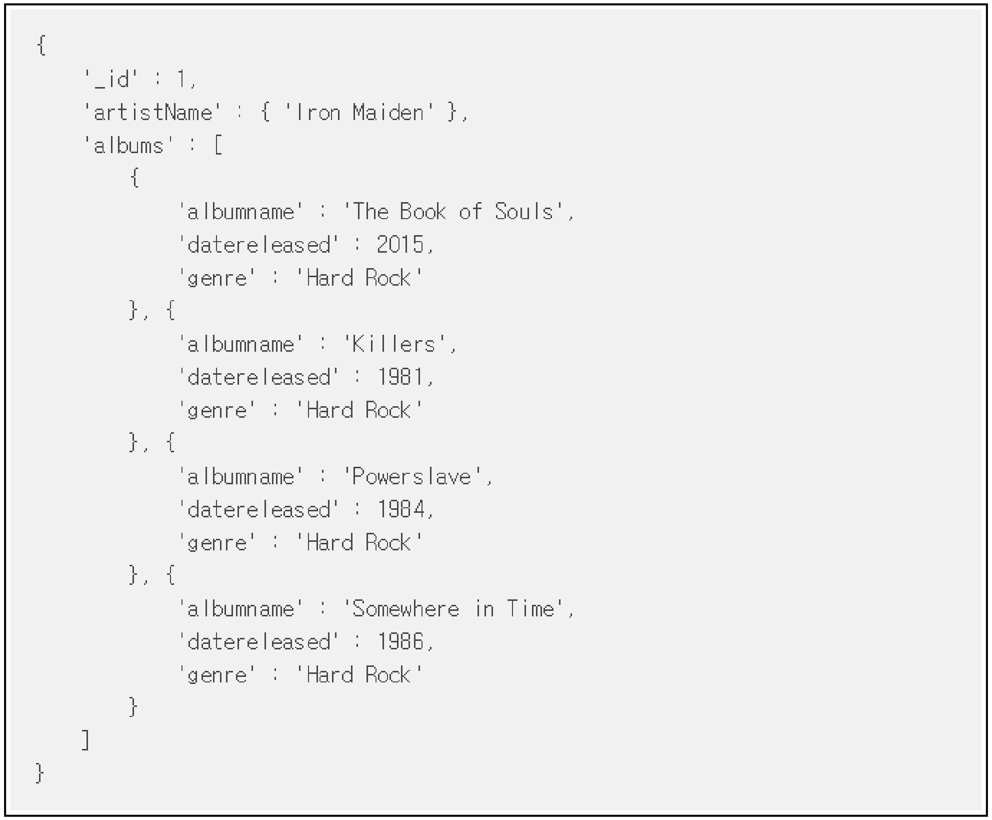
- Examples of Key-Value Database Management Systems
  - Redis
  - Oracle NoSQL Database
  - Project Voldemort
  - Aerospike
  - Oracle Berkeley DB

Document Store Database

- What is a Document Store Database?
  - Uses a document-oriented model to store data
    - Similar to a key-value database in that it uses a key-value approach
      - The difference is that, the value in a document store database consists of semi-structured data
  - Stores each record and its associated data within a single *document*
  - Each document contains semi-structured data that can be queried against using various query
    - Usually XML or JSON

Document Store Database

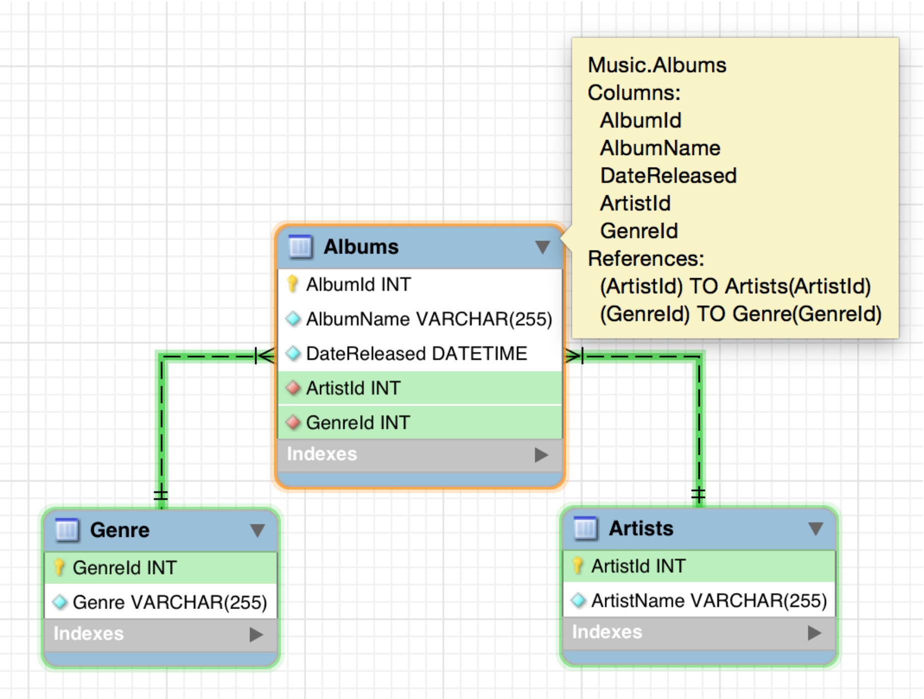
- Document Example
  - Written in JSON



```
{
    '_id' : 1,
    'artistName' : { 'Iron Maiden' },
    'albums' : [
        {
            'albumname' : 'The Book of Souls',
            'datereleased' : 2015,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Killers',
            'datereleased' : 1981,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Powerslave',
            'datereleased' : 1984,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Somewhere in Time',
            'datereleased' : 1986,
            'genre' : 'Hard Rock'
        }
    ]
}
```

Document Store Database

- Document Store vs Relational Databases
  - In relational databases, we need three different tables linking them together via their primary key and foreign key fields



Music.Albums
Columns:
  AlbumId
  AlbumName
  DateReleased
  ArtistId
  GenreId
References:
  (ArtistId) TO Artists(ArtistId)
  (GenreId) TO Genre(GenreId)

Albums
AlbumId INT
AlbumName VARCHAR(255)
DateReleased DATETIME
ArtistId INT
GenreId INT
Indexes

Genre
GenreId INT
Genre VARCHAR(255)
Indexes

Artists
ArtistId INT
ArtistName VARCHAR(255)
Indexes

Document Store Database

- Document Store vs Relational Databases
  - Tables
    - Store all data on a given entity within a single document
  - Schemas
    - Any two documents can contain a different structure and data type
  - Scalability
    - Can scale horizontally very well
  - Relationships
    - Any data associated with a record is stored within the same document

Document Store Database

- What can a Document Database be used for?
  - Web Applications
    - Content management systems, blogging platforms, eCommerce applications, web analytics, user preferences data
  - User Generated Content
    - Chat sessions, tweets, blog posts, ratings, comments
  - Catalog Data
    - User accounts, product catalogs, device registries for Internet of Things, bill of materials systems
  - Networking/computing
    - Sensor data from mobile devices, log files, realtime analytics, various other data from Internet of Things

Document Store Database

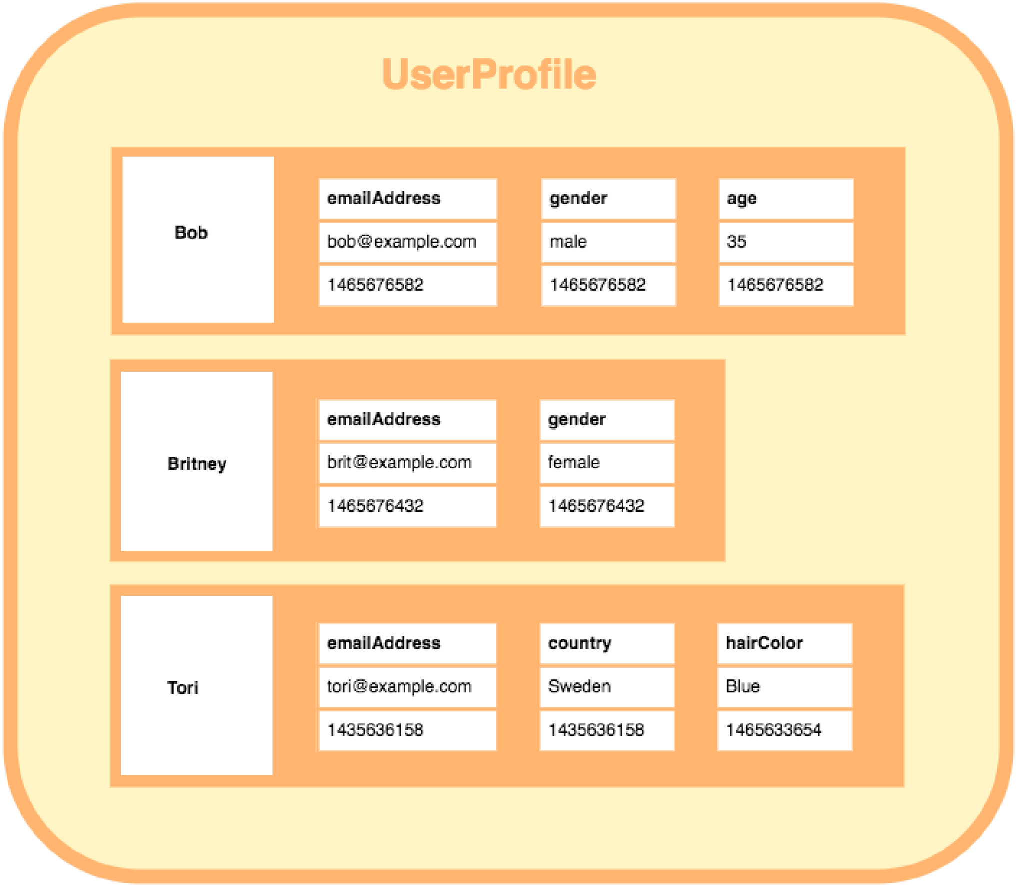
- Examples of Document Store DBMSs
  - MongoDB
  - DocumentDB
  - CouchDB
  - MarkLogic
  - OrientDB

Column Store Database

- The Structure of a Column Store Database
  - A type of database that stores data using a column oriented model.
  - A column family consists of multiple rows
  - Each row can contain a different number of columns to the other rows
    - They can have different column names, data types, etc
  - Each column is contained to its row
    - It doesn’t span all rows like in a relational database
    - Each column contains a name/value pair, along with a timestamp

Column Store Database

- The Structure of a Column Store Database



UserProfile

Bob

emailAddress	gender	age
bob@example.com	male	35
1465676582	1465676582	1465676582

Britney

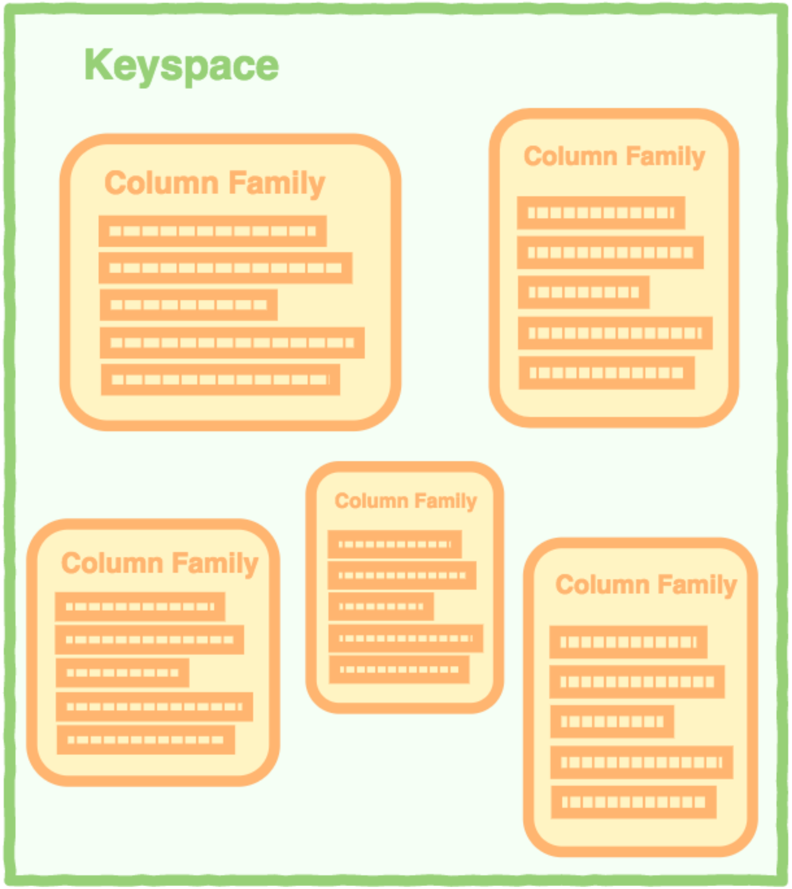
emailAddress	gender
brit@example.com	female
1465676432	1465676432

Tori

emailAddress	country	hairColor
tori@example.com	Sweden	Blue
1435636158	1435636158	1465633654

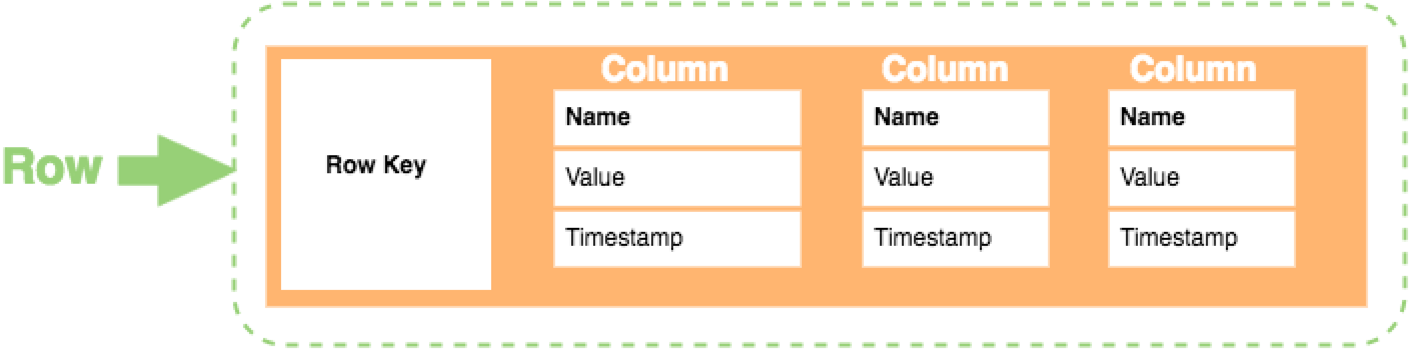
Column Store Database

- The Structure of a Column Store Database



Column Store Database

- The Structure of a Column Store Database
  - Here’s a breakdown of each element in the row:
    - **Row Key:** Each row has a unique key, which is a unique identifier for that row
    - **Column:** Each column contains a name, a value, and timestamp
    - **Name:** This is the name of the name/value pair
    - **Value:** This is the value of the name/value pair
    - **Timestamp:** This provides the date and time that the data was



Column Store Database

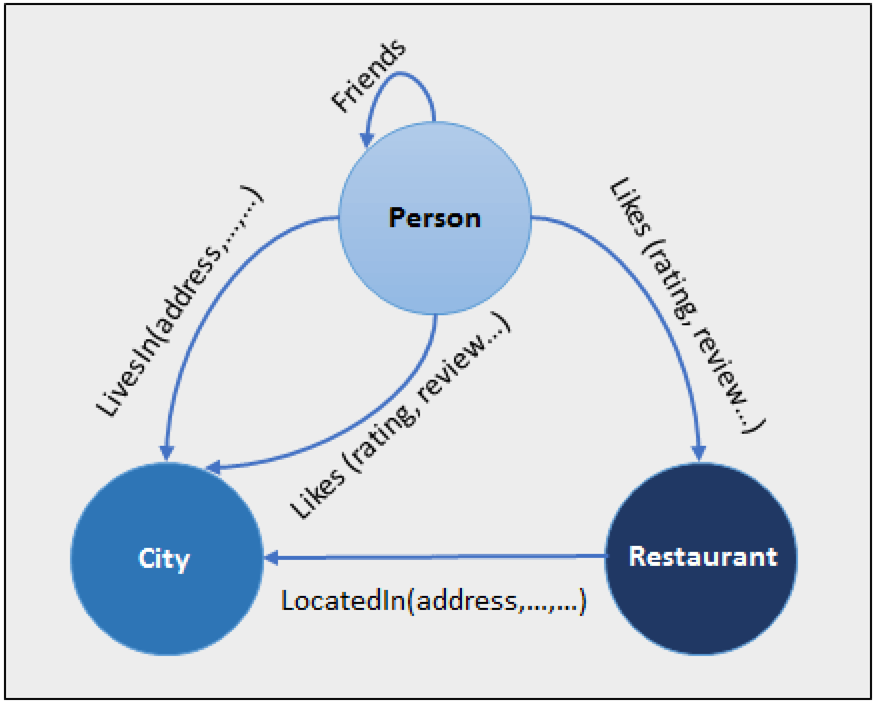
- Examples of column store databases include
  - Bigtable
  - Cassandra
  - HBase
  - Vertica
  - Druid

Graph Database

- What is graph database?
  - A database that uses a graphical model to represent and store the data
  - The graph database model is an alternative to the relational model
    - In a relational database, data is stored in tables using a rigid structure with a predefined schema
    - In a graph database, there is no predefined schema as such
      - Rather, any schema is simply a reflection of the data that has been entered
  - Graph databases are an excellent choice for working with connected data

Graph Database

- Example of how graph databases store and present data
  - The circles are *nodes* – they contain the data
  - The arrows represent the relationships between nodes

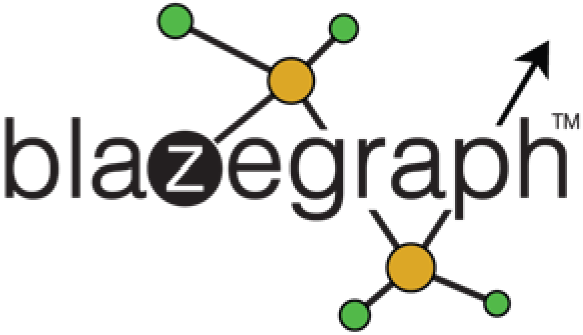

Friends
Person
LivesIn(address,.......)
Likes (rating, review...)
Likes (rating, review...)
City
Restaurant
LocatedIn(address,...,...)

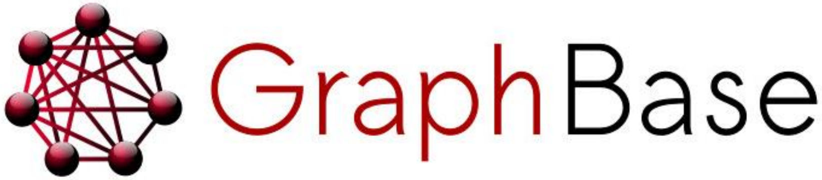
Graph Database

- What can a Graph Database be used for?
  - Social networks
  - Real-time product recommendations
  - Network diagrams
  - Fraud detection
  - Access management
  - Graph based search of digital assets
  - Master data management

Graph Database

- Examples of Graph Databases
  - Neo4j
  - Blazegraph
  - GraphBase





Summary

- Centralized storage
- Decentralized storage
  - NoSQL
    - What is NoSQL?
    - Why NoSQL?
    - NoSQL databases
- NoSQL Databases
  - Key-Value Store
  - Column
  - Document
  - Graph