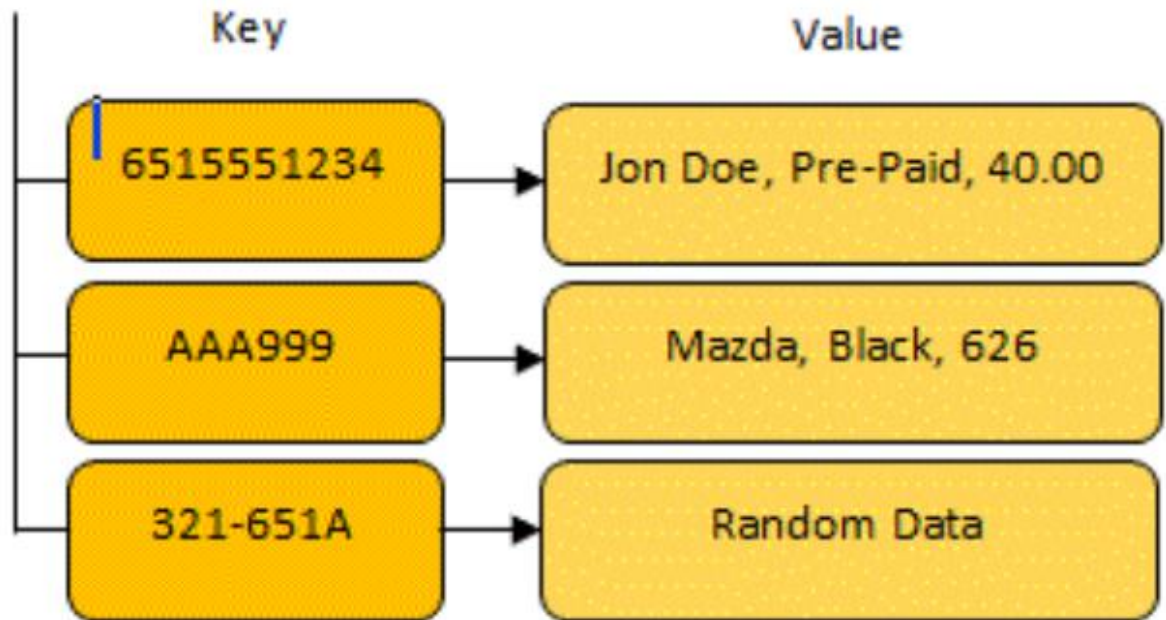# NOSQL Types

1- Key-Value Stores

- Key-value stores are most basic types of NoSQL databases.

- Designed to handle huge amounts of data.

- Based on Amazon's Dynamo paper.

- Key value stores allow developer to store schema-less data.

- In the key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBjec) etc.

- A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.

- For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin".

- Key-Value stores can be used as collections, dictionaries, associative arrays etc.

- Key-Value stores follow the 'Availability' and 'Partition' aspects of CAP theorem.

- Key-Values stores would work well for shopping cart contents, or individual values like color schemes, a landing page URI, or a default account number.

   **Database Engine Examples**: Redis, Dynamo, Riak. etc.

Key - Value Store

Database

Table : supplier

Row
ID : 1
Name : Bob
City : New York
Country : USA
Order_no : ORD-0056

Row
ID : 2
Name : Jack
City : Paris
Country : France
Order_no : ORD-0057

Table : order

Row
Order_ID : ORD-0056
Cost : 250 USD
Item_Qty1 : 2450
Item_Qty2 : 2560

Row
Order_ID : ORD-0057
Cost : 400 USD
Item_Qty1 : 3000
Item_Qty2 : 3530

2- Document

```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
            phone: "123-456-7890",        Embedded sub-
            email: "xyz@example.com"      document
           },
  access: {
            level: 5,                     Embedded sub-
            group: "dev"                  document
          }
}
```

- A collection of documents

- Data in this model is stored inside documents.

- A document is a key value collection where the key allows access to its value.

- Documents are not typically forced to have a schema and therefore are flexible and easy to change.

- Documents are stored into collections in order to group different kinds of data.

- Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Database Engine Examples: MongoDB, CouchDB etc.
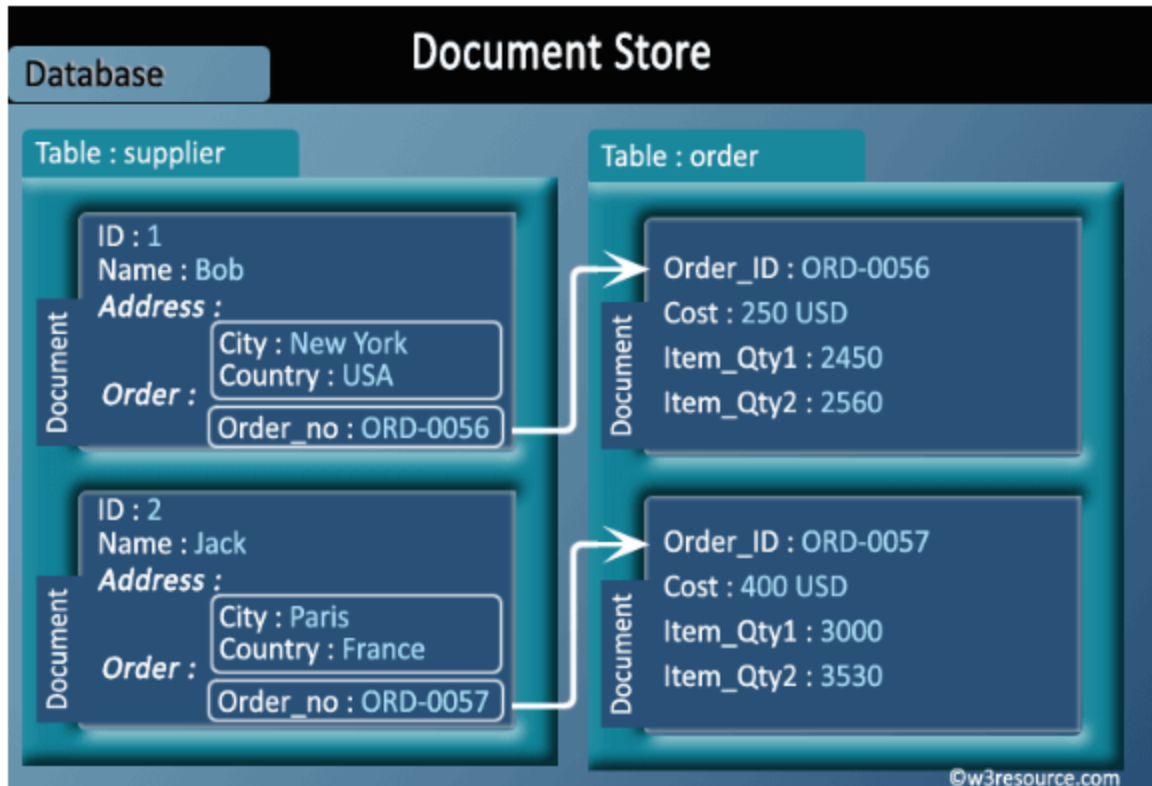
3- Column-oriented database

- Column-oriented databases primarily work on columns and every column is treated individually.

- Values of a single column are stored contiguously.

- Column stores data in column specific files.

- In Column stores, query processors work on columns too.

- All data within each column datafile have the same type which makes it ideal for compression.

- Column stores can improve the performance of queries as it can access specific column data.

- High performance on aggregation queries (e.g. COUNT, SUM, AVG, MIN, MAX).

- Works on data warehouses and business intelligence, customer relationship management (CRM), Library card catalogs etc.

Database Engine Examples: BigTable, Cassandra, SimpleDB etc.
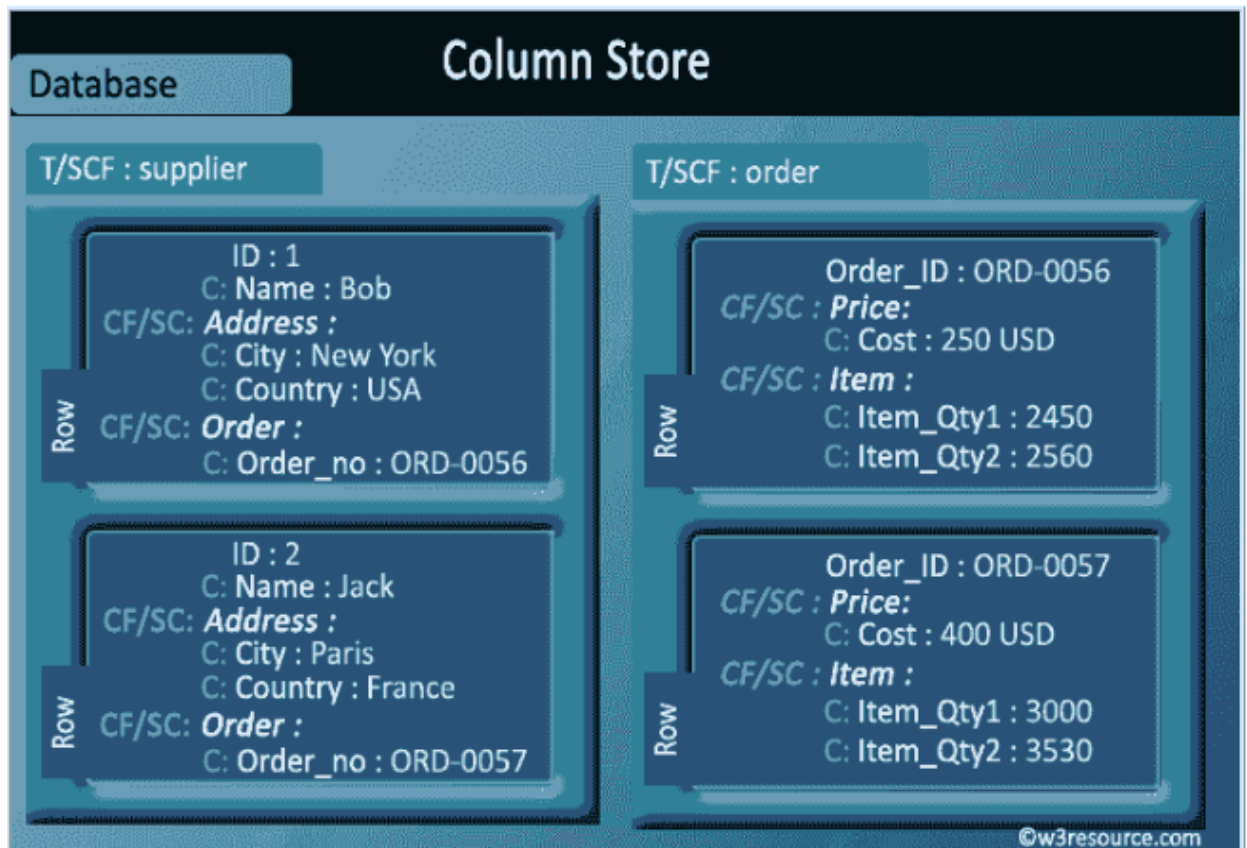
4- Graph

- A graph database stores data in a graph.

- It is capable of elegantly representing any kind of data in a highly accessible way.

- A graph database is a collection of nodes and edges

- Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes.

- Every node and edge are defined by a unique identifier.

- Each node knows its adjacent nodes.

- As the number of nodes increases, the cost of a local step (or hop) remains the same.

- Index for lookups.
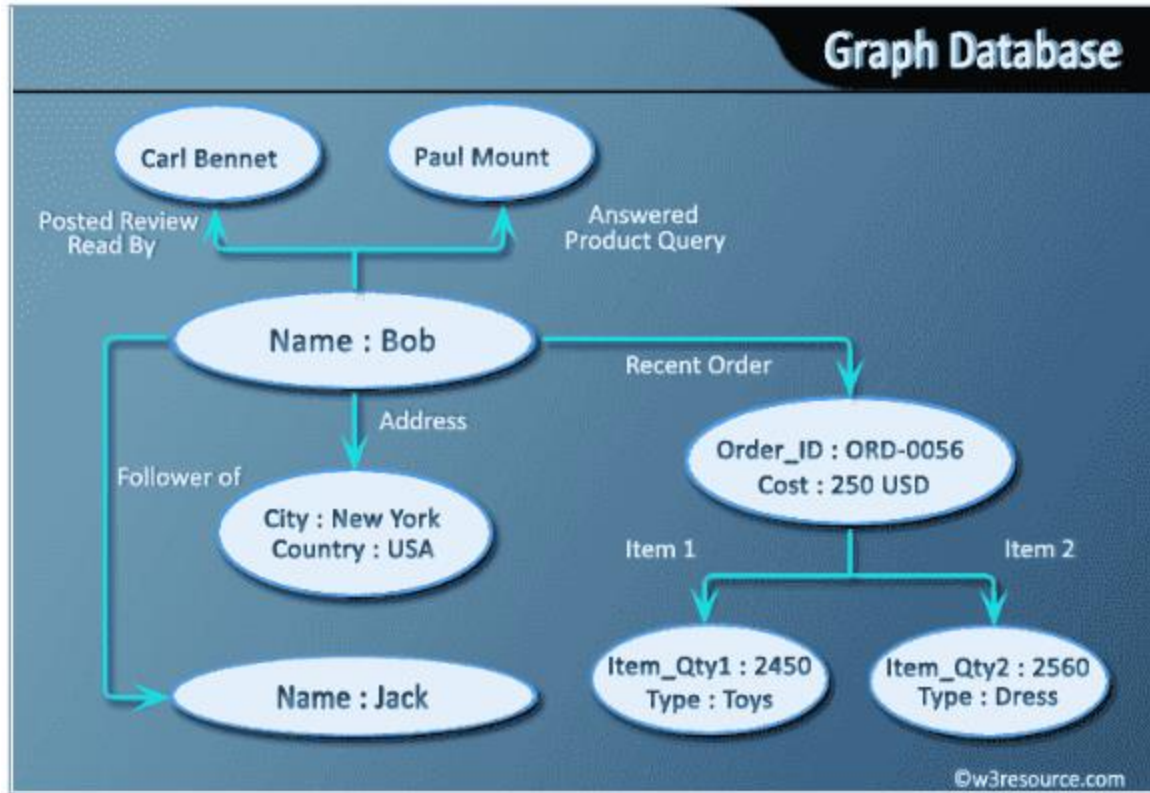
Database Engine Examples: OrientDB, Neo4J, Titan.etc.

Reference:

- https://www.w3resource.com/mongodb/nosql.php
- https://1drv.ms/b/s!ApnqAz-mA33khFV-Kqsf_oFe0y52