Universitat de Lleida
Departament d'Informàtica
i Enginyeria Industrial

Programming and Communications III: NoSQL

Jordi Ricard Onrubia Palacios

Departament d'Informàtica i Enginyeria Industrial Universitat de Lleida

Programming and Communications III: Introduction

- ➢ Introduction
- ❏ Types
- ❏ Playgrounds

Introduction

- NoSQL databases (AKA "not only SQL") store data in a more natural and flexible way than relational tables.
- As opposed to SQL, is a database management approach, whereas SQL is just a query language
- They provide flexible schemas and scale easily with large amounts of big data and high user loads.
- Come in a variety of types based on their data model. The main types are:
    - Document
    - Key-value
    - Wide-column
    - Graph

Introduction

- NoSQL databases are flexible, scalable, and distributed databases.
- BASE compliant, i.e., basic availability soft state eventual consistency.
    - Basic availability refers to the ability of the system to tolerate a partial failure (like a loss of a node).
    - Soft state means that the system allows temporary inconsistencies before eventually achieving consistency automatically over time.
- BASE compliance ensures high availability, faster data processing, scalability, and flexibility.

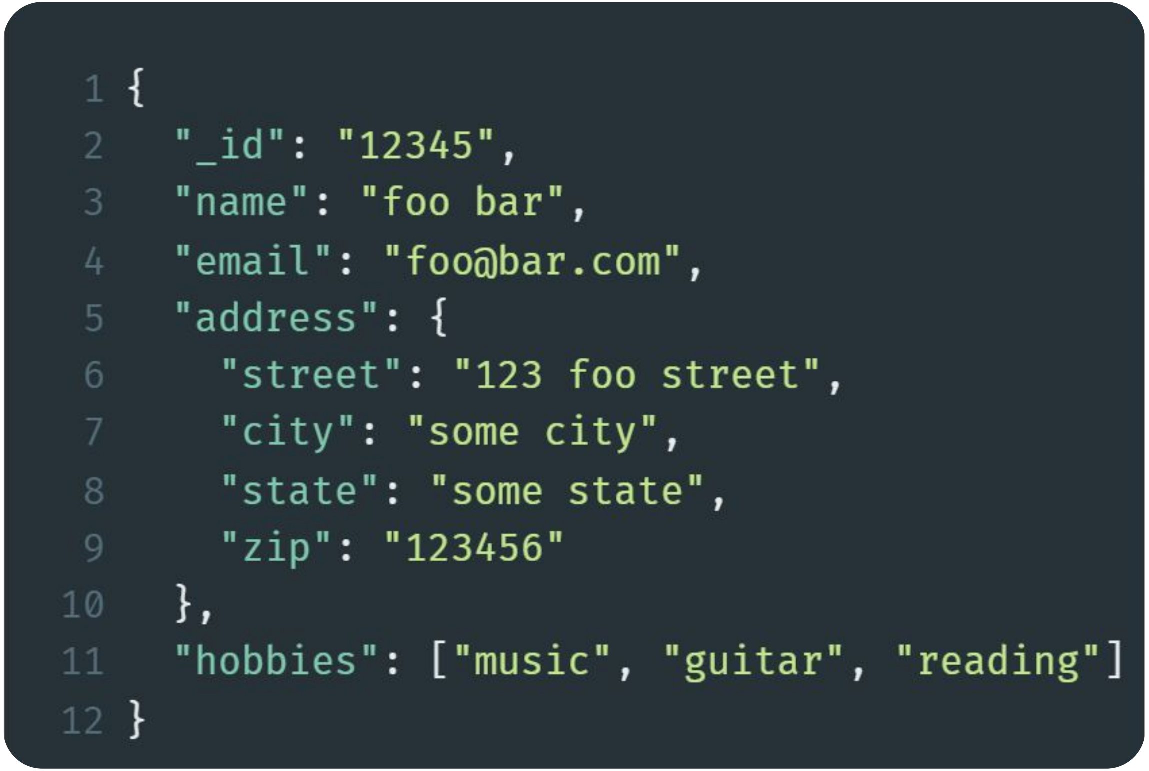
Programming and Communications III: Introduction

- ❏ Introduction
- ➢ Types
- ❏ Playgrounds

Types - Document

- A document-oriented database stores data in documents similar to JSON (JavaScript Object Notation) objects.
- Each document contains pairs of fields and values.
  - The values can typically be a variety of types, including things like strings, numbers, booleans, arrays, or even other objects.
  - They also support nested structures, making it easy to represent complex relationships or hierarchical data.
- Use Case:
  - A document database offers a flexible data model, much suited for semi-structured and typically unstructured data sets.

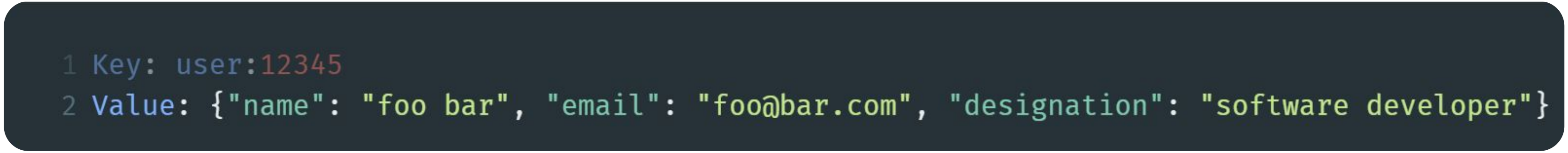
Types - Document



```
{
  "_id": "12345",
  "name": "foo bar",
  "email": "foo@bar.com",
  "address": {
    "street": "123 foo street",
    "city": "some city",
    "state": "some state",
    "zip": "123456"
  },
  "hobbies": ["music", "guitar", "reading"]
}
```

Types - Key-Value

- A key-value store is a simpler type of database where each item contains keys and values.
- Each key is unique and associated with a single value.
- They are used for caching and session management and provide high performance in reads and writes because they tend to store things in memory.

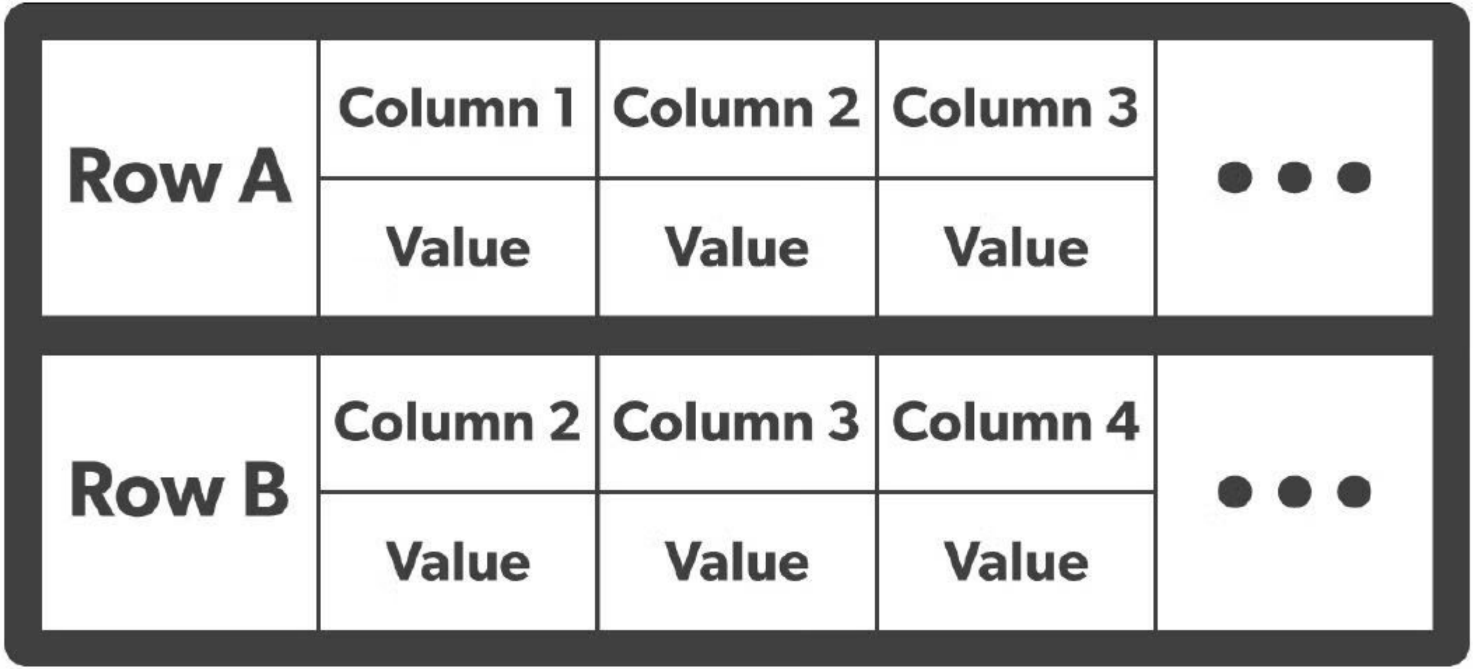


```
Key: user:12345
Value: {"name": "foo bar", "email": "foo@bar.com", "designation": "software developer"}
```

Types - Wide-Column

- Wide-column stores store data in tables, rows, and dynamic columns.
- The data is stored in tables. However, unlike traditional SQL databases, wide-column stores are flexible, where different rows can have different sets of columns.
- These databases can employ column compression techniques to reduce the storage space and enhance performance.
- Wide-column databases are ideal for use cases that require a large dataset that can be distributed across multiple database nodes, especially when the columns are not always the same for every row.
    - Log data
    - IoT (Internet of Things) sensor data
    - Time-series data, such as temperature monitoring or financial trading data
    - Attribute-based data, such as user preferences or equipment features
    - Real-time analytics

Types - Wide-Column

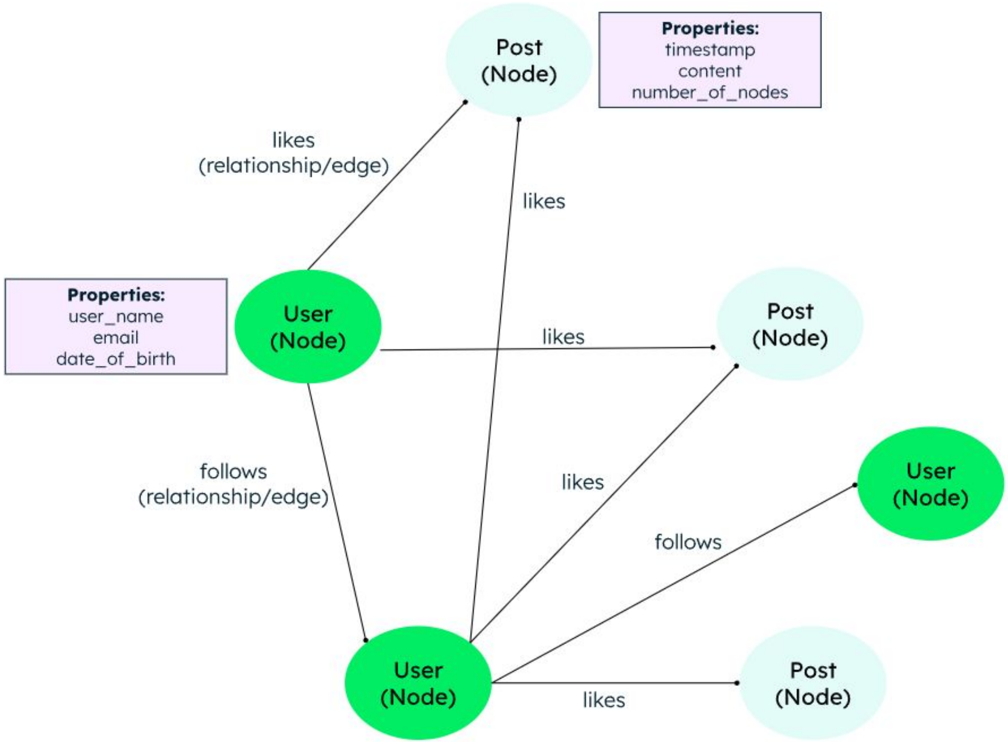


Row A	Column 1	Column 2	Column 3	● ● ●
	Value	Value	Value	
Row B	Column 2	Column 3	Column 4	● ● ●
	Value	Value	Value	

Types - Graph

- A graph database stores data in the form of nodes and edges.
- Nodes typically store information about people, places, and things (like nouns), while edges store information about the relationships between the nodes.
- They work well for highly connected data, where the relationships or patterns may not be very obvious initially.

Types - Graph


Post
(Node)
Properties:
timestamp
content
number_of_nodes
likes
(relationship/edge)
likes
Properties:
user_name
email
date_of_birth
User
(Node)
likes
Post
(Node)
follows
(relationship/edge)
likes
User
(Node)
follows
User
(Node)
likes
Post
(Node)

Programming and Communications III: Introduction

❏ Introduction

❏ Types

➢ Playgrounds

Playgrounds - Document

- MongoDB: https://mongoplayground.net/
- Redis: https://cloud.redis.io/#/databases
- Neo4J: https://sandbox.neo4j.com/