Введение в NoSQL и работа с MongoDB

Основные отличия между реляционными и NoSQL базами данных.

Реляционные базы данных (SQL):

- Структура: Данные хранятся в виде таблиц, состоящих из строк и столбцов. Каждая таблица имеет схему, определяющую структуру данных (например, типы данных и ограничения).
- Связи: Таблицы могут быть связаны между собой с помощью внешних ключей, что позволяет создавать сложные запросы и обеспечивать целостность данных.
- Транзакции: SQL базы данных поддерживают транзакции, что позволяет выполнять несколько операций как одно целое (ACID-свойства: атомарность, согласованность, изолированность и долговечность).
- Примеры: MySQL, PostgreSQL, MS SQL Server, Oracle.

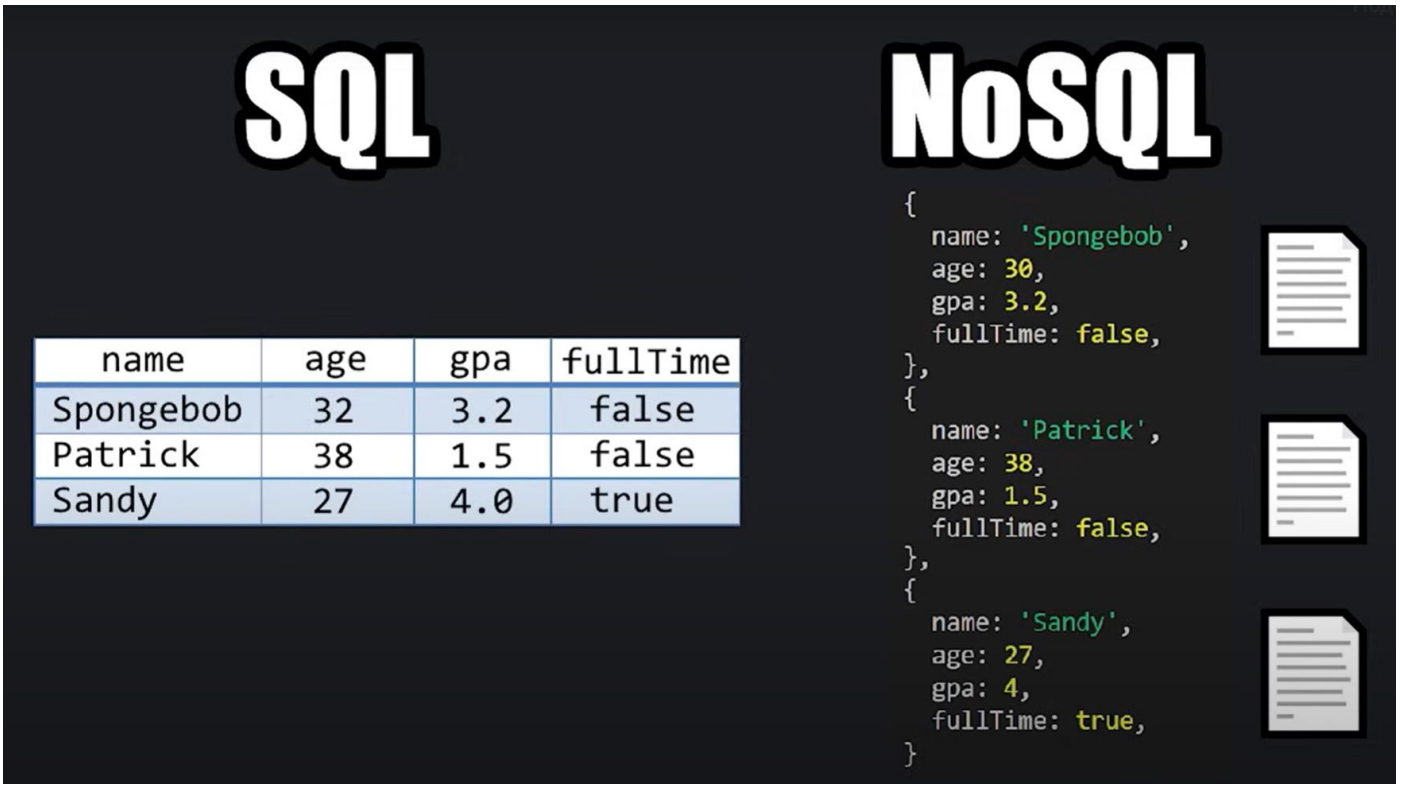
NoSQL базы данных:

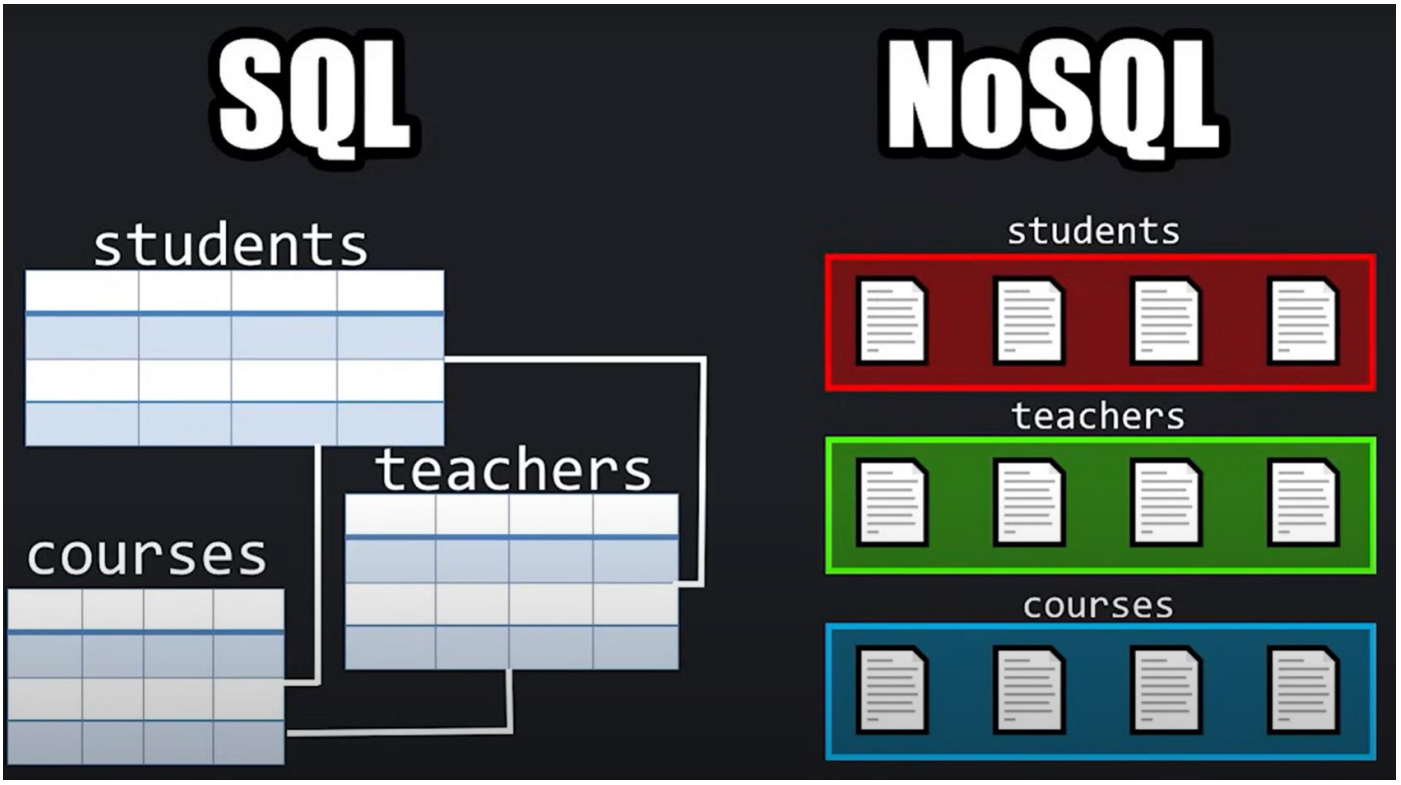
- Гибкость: Нет фиксированной схемы, данные могут быть организованы по-разному (документы, графы, пары ключ-значение). Это позволяет легко адаптировать структуру данных к изменениям в требованиях приложения.
- Масштабируемость: NoSQL базы данных обычно проектируются с учетом горизонтальной масштабируемости, что позволяет легко распределять данные между несколькими серверами.
- Быстрота: Оптимизированы для быстрого доступа и обработки больших объемов данных, что делает их подходящими для работы в реальном времени.

- Примеры: MongoDB (документоориентированная), Redis (ключ-значение), Cassandra (колоночная), Neo4j (графовая).

Ключевые различия:

- Схема: В SQL — жесткая схема, в NoSQL — схема отсутствует или гибкая.
- Типы данных: SQL ориентирован на строгие типы данных, NoSQL может работать с более свободными и изменяемыми типами данных.
- Масштабируемость: SQL базы данных обычно масштабируются вертикально (увеличение ресурсов на одном сервере), NoSQL — горизонтально (добавление новых серверов).





Основные типы NoSQL баз данных

1. Документоориентированные базы данных:

Пример: MongoDB.

Описание: Данные хранятся в виде документов, обычно в формате JSON или BSON. Каждый документ может иметь различную структуру, что делает эту модель гибкой.

Использование: Приложения, работающие с неструктурированными данными, например, профили пользователей, каталоги товаров.

2. Key-Value хранилища:

Пример: Redis.

Описание: Данные хранятся как пары "ключ-значение". Очень простая и быстрая структура, используемая для кэширования, сессий и других задач, где важна скорость.

Использование: Кэширование, хранение сессий, обработка больших объемов временных данных.

3. Колоночные базы данных:

Пример: Apache Cassandra.

Описание: Данные хранятся в столбцах, что позволяет эффективно обрабатывать большие объемы данных и выполнять аналитические запросы.

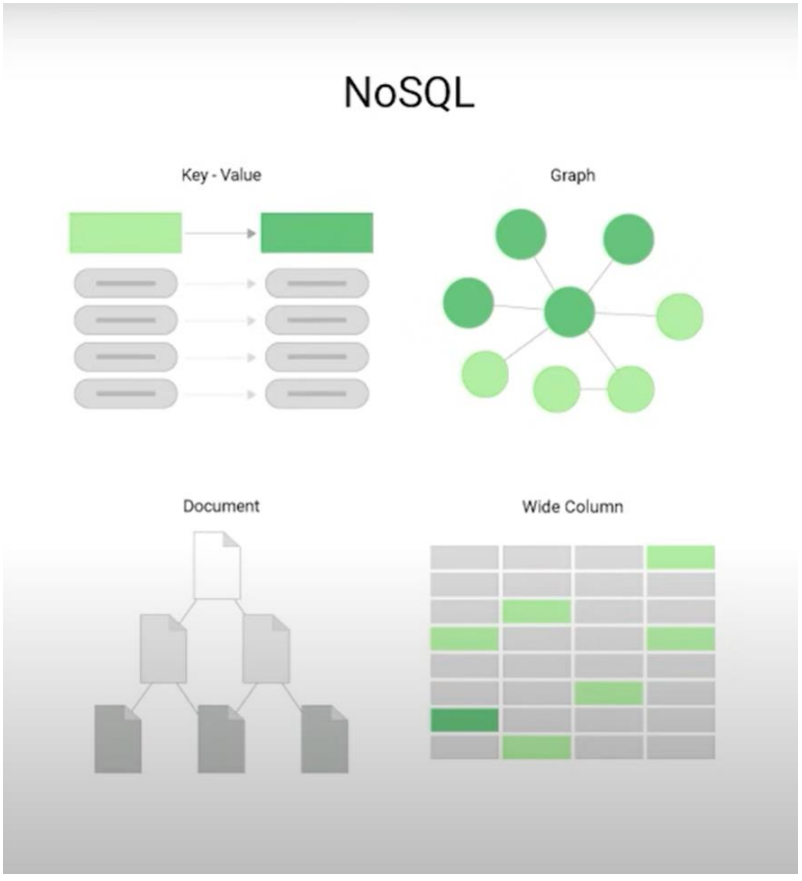
Использование: Аналитика, работа с большими наборами данных, распределенные системы.

4. Графовые базы данных:

Пример: Neo4j.

Описание: Данные хранятся в виде узлов и ребер, что позволяет моделировать сложные связи между данными, такие как социальные сети, рекомендательные системы.

Использование: Социальные сети, графовые поисковые системы, рекомендательные системы.



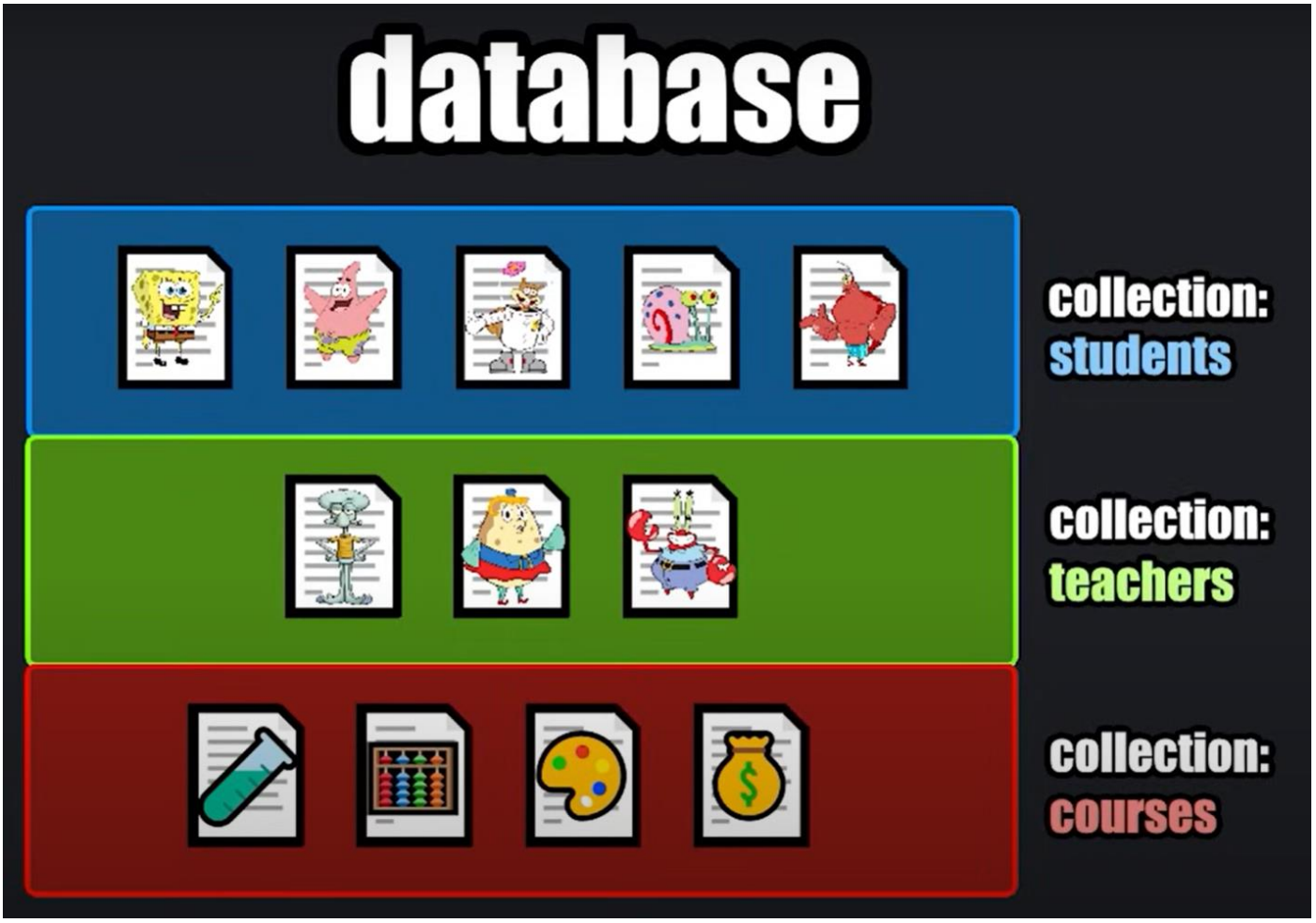
Основы работы с MongoDB

MongoDB — это документоориентированная база данных, где данные хранятся в виде документов в формате BSON (бинарный аналог JSON). Это позволяет гибко управлять данными, без строгой схемы, как в реляционных базах данных.

Основные концепции MongoDB

1. База данных (Database):

База данных в MongoDB — это контейнер для коллекций. В одной базе данных можно хранить различные коллекции, каждая из которых предназначена для определенных типов данных. В отличие от SQL, базы данных в MongoDB не создаются явно. Они создаются автоматически при вставке данных в коллекцию.

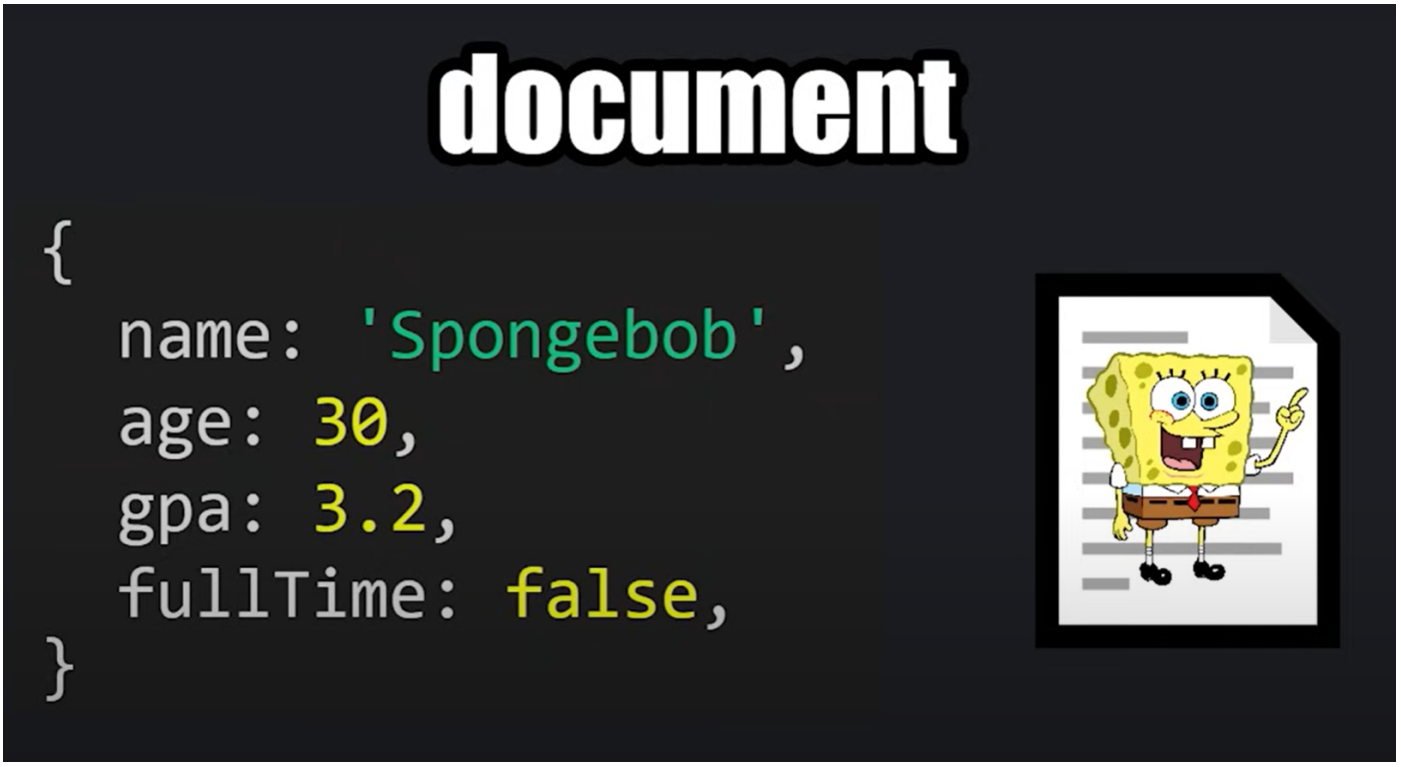


2. Коллекция (Collection):

Коллекция в MongoDB аналогична таблице в реляционной базе данных, но в отличие от таблицы, она не требует заранее определенной схемы. Документы в одной коллекции могут иметь разные поля. Это дает большую гибкость при работе с данными.

3. Документ (Document):

Документ — это основной объект хранения в MongoDB. Он представляет собой запись, хранящуюся в коллекции, и состоит из пар "ключ-значение". Документы записываются в формате BSON, который похож на JSON, но имеет некоторые дополнительные возможности.



4. Ключевые особенности:

Гибкость схемы: Вы можете добавлять новые поля в документы без необходимости изменения всей коллекции.

Индексация: MongoDB поддерживает индексацию, что позволяет ускорять поиск по определенным полям документов.

Основные CRUD операции в MongoDB

CRUD — это акроним, который обозначает основные операции с данными: **Create (создание), Read (чтение), Update (обновление), Delete (удаление)**.

1. **Create (Создание):**

Создание документа осуществляется путем его вставки в коллекцию. Если коллекция не существует, она создается автоматически.

Пример команды:

```
db.myCollection.insertOne({ name: "Alice", age: 25, profession: "Engineer" })
```

2. **Read (Чтение):**

Для чтения данных в MongoDB используются команды find и findOne. Команда find возвращает курсор на множество документов, соответствующих критерию запроса, а findOne возвращает только первый найденный документ.

Пример команды:

```
db.myCollection.find({ age: { $gt: 20 } })
db.myCollection.findOne({ name: "Alice" })
```

3. **Update (Обновление):**

MongoDB позволяет обновлять документы с помощью команды updateOne, updateMany или replaceOne. Вы можете изменить значения конкретных полей или полностью заменить документ.

Пример команды:

```
db.myCollection.updateOne({ name: "Alice" }, { $set: { age: 26 } })
```

4. Delete (Удаление):

Удаление документов выполняется с помощью команд deleteOne и deleteMany. Команда deleteOne удаляет первый найденный документ, соответствующий критерию, а deleteMany — все подходящие документы.

Пример команды:

```
db.myCollection.deleteOne({ name: "Alice" })
```