

# ***Разработка информационных систем с использованием СПО NoSQL СУБД MongoDB***

С.А.Мартишин\*, В.Л. Симонов\*\*, М.В. Храпченко\*

\* Институт системного программирования  
Российской академии наук

\*\* Российский государственный социальный  
университет

# **Специальности и направления подготовки студентов**

- «Информационные системы и технологии»
  - «Программная инженерия»
  - «Информатика и вычислительная техника»
  - «Автоматизированные системы обработки информации и управления»
- и ряд других.

Подготовка будущих специалистов ориентирована на использование технологий, наилучшим образом подходящих для выполнения стоящих перед ними задач. Необходимо, чтобы студенты были знакомы со всем многообразием средств разработки и проектирования ИС, поэтому важным является изучение студентами помимо реляционных СУБД также СУБД NoSQL («не только SQL»).

За последние 15-20 лет значительно изменился характер информации, подлежащий обработке и, соответственно, изменились интерактивные приложения, а именно:

- 1 - значительно увеличился объем данных, подлежащих хранению и обработке (проблема Big Data);
- 2 - в значительной степени возросло количество пользователей, использующих приложения, причем часто одновременно, в том числе и мультимедийную информацию на мобильных устройствах;
- 3 - значительное количество хранимой информации сложно структурировать, используя технологию систем управления реляционными базами данных (реляционных СУБД).

Проблема «Big Data», характеризуется «тремя V»:

- объём (Volume) - величина физического объёма;
- скорость (Velocity) – как скорость прироста, так и необходимость высокоскоростной обработки и получения результатов;
- многообразиие (Variety) - возможность одновременной обработки различных типов структурированных и полуструктурированных данных.

Примеры - хранение данных картографирования и обзора земной поверхности со спутников; хранение записей видеокамер в службах ГИБДД, охранных и иных систем; хранение с.м.с.-сообщений, писем электронной почты, данных пользователей в социальных сетях, а также фотографий, видео и пр., хранение записей телефонных переговоров и др. Миллионы приложений и миллиарды пользователей.

Традиционные реляционные СУБД перестали справляться с указанными задачами «Big Data».

Причины:

- Объем данных превышает границы вертикального масштабирования;
- Время принятие решения по обработке данных неудовлетворительно по сравнению со скоростью изменения данных;
- Многообразие форматов делает интеграцию дорогой;
- Многовариантность интерпретации данных затрудняет анализ.

При существенном увеличении нагрузок традиционно применение масштабирования – разделение данных на группы и выделение их на отдельные серверы. Здесь возможны две основные стратегии – репликация и шардинг.

Репликация - создание дубликата базы данных.

Шардинг - разделение базы данных на отдельные части так, чтобы каждую из них можно было вынести на отдельный сервер.

Шардинг подразделяется на вертикальный и горизонтальный



Разнообразие структуры различной информации привело к организации иной структуры данных – NoSQL, которая характеризуется:

- гибкостью модели данных (здесь более слабые ограничения на модель данных);
- уменьшением объема администрирования (более простая модель данных снижает потребности в администрировании и настройке);
- удобное масштабирование (разрабатывалась с целью добавления любого количества новых узлов);
- улучшенные технические и экономические показатели (более быстрые выборки данных из-за отсутствия операции join), размещение на недорогих массовых серверах

## Недостатки NoSQL:

- Ограниченная или отсутствующая поддержка транзакций – препятствие для использования в ответственных решениях;
- Несколько ограниченные средства запросов и оперативного анализа;
- Являются проектами с открытым кодом и не имеют такого высокого уровня поддержки, как реляционные СУБД;
- NoSQL-подход находится в состоянии развития. Каждый разработчик NoSQL-системы находится «в режиме обучения»

Об ACID (atomicity, consistency, isolation, durability - атомарность, согласованность, изоляция и надежность)

Реляционные системы БД традиционно поддерживают такую стратегию, которая обеспечивает набор гарантий ACID.

В NoSQL же базах ослаблен или имеет место отказ от поддержки ACID.

Для NoSQL выполняется теорема CAP: для любой реализации распределенных вычислений обеспечивается не более 2-х из трех свойств:

- *Consistency (согласованность данных)* — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;
- *Availability (доступность)* — любой запрос к распределённой системе завершается корректным откликом;
- *Partition tolerance (устойчивость к разделению)* - расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

Однако для систем со значительным количеством клиентов, в которых приоритетными свойствами являются доступность и устойчивость к разделению (теорема CAP), использование NoSQL имеет значительные преимущества как для пользователей, так и для разработчиков, а именно: снижение времени отклика, использование различных типов хранилищ (в том числе облачных), возможность обойтись без создания схемы данных, за счет чего сокращается требуемое время разработки.

## Примеры некоторых NoSQL баз данных:

- Redis
- MongoDB
- Riak
- Neo4j
- Memcached

Одной из популярных NoSQL СУБД является MongoDB, которая строится на JSON (JavaScript Object Notation)

- подобных документах, хранящими данные в виде «ключ - значение» (документо-ориентированное хранение), а также поддерживает возможности:

- Гибкая поддержка индекса;
- Автоматический шардинг;
- Большое количество запросов;
- Встроенные репликации;
- Поддержка спецификации хранения GridFS для хранения и извлечения файлов, которые превышают предельный размер BSON-документ 16 Мб, и др.

# Основные различия MongoDB и MySQL

Параметр	<u>MongoDB</u>	<u>MySQL</u>
Модель данных	<u>Документо</u> – ориентированная	Реляционная
Типы данных	String, 32-bit и 64-bit <u>int</u> , Double, Boolean, Date, Regular Expression, Object, JavaScript	Типы данных <u>MySQL</u>
Организация хранения наборов данных	Коллекции	Таблицы
Способ организации запросов	Объектный язык запросов	SQL



Терминология реляционных баз данных и MongoDB имеют различия. Например, термину **таблица** в реляционной базе данных соответствует термин **коллекция** в NoSQL т.п.

Термины и понятия <u>MongoDB</u>	Термины и понятия <u>SQLs</u>
Коллекция	Таблица
Документ или BSON документ	Строка
Встроенные документы и ссылки	Объединение таблиц
Первичный ключ В <u>MongoDB</u> , первичный ключ автоматически устанавливается в поле <u>_id</u>	Первичный ключ В качестве первичного ключа выступает одно или несколько полей (столбцов), <u>комбинация</u> значений которых однозначно определяет каждую запись в таблице

Знакомство студентов с MongoDB начинается с её установки, которая осуществляется в терминале, начиная с команд:

```
# yum install mongodb mongodb-server mongodb-devel php-  
pecl-mongo
```

Далее студенты отвечают на вопрос:

```
this is ok [y/d/N] y
```

... осуществляют выбор директории и так далее (подробно представлено в сданном в печать учебном пособии авторов)

После установки, в качестве первого знакомства с применением MongoDB студентам предлагается создать БД путем создания первой коллекции, затем осуществлять добавление, редактирование, удаление записей. Также студенты практикуются в осуществлении выборки данных из коллекции

Если не использовать оболочку UMongo, то работа производится из командной строки.

Коллекция создается командой **insert()**:

```
> db.collection1.insert({ name: "Соколов"})
```

```
> show dbs;
```

```
local  0.078125GB
```

```
staff  0.203125GB
```

```
test   0.203125GB
```

```
tutorial 0.203125GB
```

```
users  0.203125GB
```

```
>
```

Для проверки содержимого коллекции (создана ли она), в командной строке набираем:

```
> db.personnel.find()
```

В ответ получаем содержимое коллекции:

```
{ "_id" : ObjectId("530db671aeef2e9d5ba4188c"),  
  "id_worker" : "w1", "name" : "Иванов", "phone" : "916 111  
111 111", "age" : 55 }
```

Работа с большим числом записей приводит к выводу на экран:

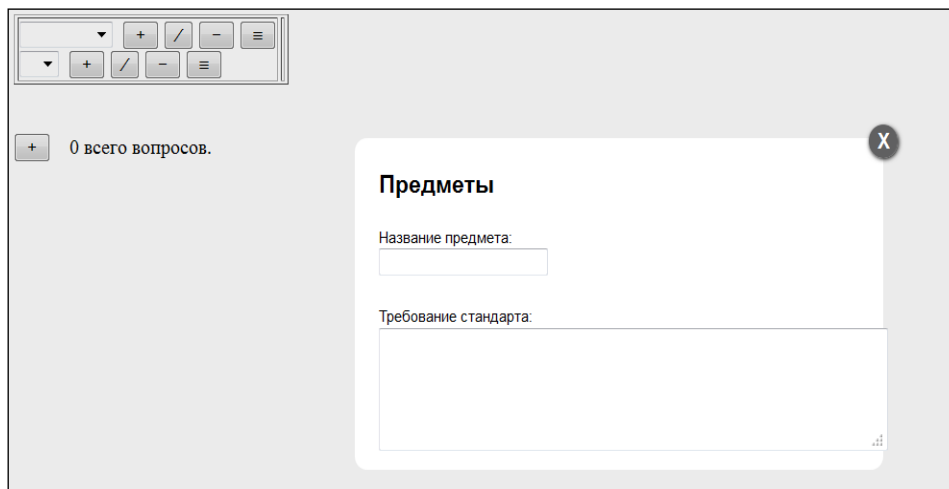
```
> db.personnel.find()
```

```
{ "_id" : ObjectId("533401994025dab37604c4aa"),  
  "address" : "", "age" : 55, "id_worker" : "w1", "name" :  
  "Иванов", "phone" : "916 111 111 111" }  
  
{ "_id" : 123, "address" : "", "age" : 65, "id_worker" : "w2",  
  "name" : "Петров", "phone" : "916 222 222 22" }  
  
{ "_id" : 11, "address" : "", "age" : 25, "id_worker" : "w3",  
  "name" : "Сидоров", "phone" : "916 333 33 33" }  
  
{ "_id" : ObjectId("533404d0c4982060e00156c1"), "address"  
  : "", "age" : 35, "id_worker" : "w4", "name" : "Смирнов",  
  "phone" : "916 444 44 4" }
```

В качестве более сложного примера  
приведём проект информационной  
системы, использующийся в учебном  
процессе

В рассматриваемой информационной системе необходимо хранить данные о преподаваемых предметах, темах, которые необходимо изучить и контрольные вопросы по темам. Интерфейс:

Окно для ввода информации о дисциплине / предмете



The screenshot shows a software window with a title bar at the top containing standard window controls (minimize, maximize, close) and a toolbar with icons for adding, deleting, and editing. Below the title bar, there is a status bar on the left that says "+ 0 всего вопросов." To the right of the status bar is a white panel titled "Предметы" (Subjects) with a close button (X) in the top right corner. Inside this panel, there are two input fields: "Название предмета:" (Subject name) with a single-line text input box, and "Требование стандарта:" (Standard requirement) with a multi-line text input box. A small icon is visible in the bottom right corner of the multi-line input box.



# Заполнение данных о предмете

**Предметы**

Название предмета:

Требование стандарта:  
Федеральный компонент государственного стандарта (общего образования, основного общего образования, среднего (полного) общего образования) по математике, утвержден приказом Минобразования России от 5.03.2004 г. № 1089

# Данные о предмете внесены

Математика ▾ + / - ≡

▾ + / - ≡

+ 0 всего вопросов.

# Отображение служебной информации

# Окно для ввода вопросов

+ 0 всего вопросов.

**Предметы**

Название предмета:

Требование стандарта:  
Федеральный компонент государственного стандарта (общего образования, основного общего образования, среднего (полного) общего образования) по математике, утвержден приказом Минобрнауки России от 5.03.2004 г. № 1089

sNameDB=school  
sNameCollection=predmet  
s\_id=53675b3d9d368e4c0ef89c02  
sNameField=tребстандпредм  
sValueField=Федеральный компонент государственного стандарта (общего образования, основного общего образования, среднего (полного) общего образования) по математике, утвержден приказом Минобрнауки России от 5.03.2004 г. № 1089  
nSignText=1  
Array ( [treбстандпредм] => Федеральный компонент государственного стандарта (общего образования, основного общего образования, среднего (полного) общего образования) по математике, утвержден приказом Минобрнауки России от 5.03.2004 г. № 1089 ) Update document with ID: 53675b3d9d368e4c0ef89c02

Математика + / - ≡  
Тригонометрические функции + / - ≡

+ 0 всего вопросов.

**Вопрос**

Номер вопроса:

Вопрос:

# Окончательный вид внесенных вопросов по теме

Математика > Производная

+ 7 всего вопросов.

1. Понятие о пределе последовательности. / -
2. Существование предела монотонной ограниченной последовательности. / -
3. Длина окружности и площадь круга как пределы последовательностей. / -
4. Бесконечная геометрическая прогрессия и ее сумма. / -
5. Понятие о непрерывности функции. / -
6. Понятие о производной. / -
7. Производная суммы, разности, произведения, частного. / -

# Файлы, необходимы для функционирования системы

Имя	↑Тип	Размер	Дата
⋈ [.]	<DIR>		06.05.2014
index	htm	10 664	05.05.2014
dobavpredm	php	595	21.04.2014
dobavtema	php	656	21.04.2014
dobavvopros	php	648	24.04.2014
getfield	php	944	17.04.2014
lo	php	850	22.04.2014
predm	php	902	17.04.2014
remove	php	714	15.04.2014
setfield	php	1 113	21.04.2014
tema	php	1 010	22.04.2014
vopros	php	1 802	05.05.2014

# Коды некоторых файлов

айла index.htm

```
<?PE html>
charset="UTF-8">
<title>Учебный план</title>
<style>
деление стилей, в том числе модального окна*/
modalDialog {
    position: fixed;
    font-family: Arial, Helvetica, sans-serif;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    background: rgba(0,0,0,0.08);
    z-index: 99999;
    opacity: 0;
    -webkit-transition: opacity 4ms ease-in;
    -moz-transition: opacity 4ms ease-in;
    transition: opacity 4ms ease-in;
    pointer-events: none;
}
modalDialog:target {
    opacity: 1;
    pointer-events: auto;
}
modalDialog > div {
    width: 400px;
    position: relative;
    margin: 10% auto;
    padding: 5px 20px 13px 20px;
    border-radius: 10px;
    background: #fff;
}
close {
    background: #606061;
    color: #FFFFFF;
    line-height: 25px;
```

Код файла dobavt

```
<?php
$SNameDB = "school";
$SNameCollection = "predmet";
try {
    // open connection to MongoDB
    $conn = new MongoClient('localhost');
    // access database
    $db = $conn->$SNameDB;
    // access collection
    $collection = $db->$SNameColl
    //
    $item['nazvpredm'] = "";
    $item['trebstandpredm'] = "";
    $collection->insert($item);
    echo $item['_id'];
    // disconnect from server
    $conn->close();
}
//обработка исключений
catch (MongoConnectionExcept
die('Error connecting to Mongo
} catch (MongoException $e) {
die('Error: ' . $e->getMessage()
}
?>
```

Код файла dobavtema.php

```
$t = $_POST['idpredmet'];
$db = "school";
$collection = "tema";

connection to MongoDB server
new MongoClient('localhost');
$db = database
$conn->$SNameDB;
$collection = $db->$SNameCollection;

$idpredmet'] = $idpredmet;
$zvtema'] = "";
$hasitema'] = "";
$conn->insert($item);
echo $item['_id'];
disconnect from server
close();

отка исключений
MongoConnectionException $e) {
    //connecting to MongoDB server'
    MongoException $e) {
        : ' . $e->getMessage());
```

Код файла dobavvopros.php

```
$t = $_POST['idtema'];
$db = "school";
$collection = "vopros";

connection to MongoDB server
new MongoClient('localhost');
$db = database
$conn->$SNameDB;
$collection = $db->$SNameCollection;
```

Опыт использования СУБД NoSQL-типа пока небольшой, хотя есть достаточно крупные организации, с успехом работающие на MongoDB. Однако от частных разработчиков данные поступают достаточно противоречивые. Нет твердо устоявшихся предпочтений, и различные разработчики сообщают свои частные мнения типа «я вот не использовал, и никому другому не советую».

Так, описаны попытки использовать MongoDB для хранения данных о сериалах, а также для создания проекта распределенной социальной сети. В одном случае был успех, во втором – провал, что впоследствии объяснилось непродуманностью выбора инструментария перед началом работ.

## Выводы

Перед студентами ставится задача ознакомления с достаточно непривычным материалом – БД типа NoSQL, находящим широкое применение на практике.

Студенты должны овладеть основными приемами работы с коллекциями (создание, обновление, выборка, удаление и т.д.).

Поэтому на данном этапе не ставится глобальных задач по работе с «большими проектами»

## Литература

- [1] Фаулер М., Садаладж П. Дж. NoSQL: новая методология разработки нереляционных баз данных - NoSQL Distilled. - М.: «Вильямс», 2013. – 192 с. - ISBN 978-5-8459-1829-1
- [2] MongoDB. [Электронный ресурс]. Режим доступа: URL: <http://www.mongodb.org> – Яз. англ. Дата обращения: 31.10.2014.

*Благодарим за внимание*