

УДК 003(075)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОСОБЕННОСТЕЙ ПРОЕКТИРОВАНИЯ И РЕАЛИЗАЦИИ ЗАПРОСОВ В РЕЛЯЦИОННЫХ И ДОКУМЕНТНО-ОРИЕНТИРОВАННЫХ БАЗАХ ДАННЫХ НА ПРИ- МЕРЕ СУБД SQL SERVER И MONGODB

В. В. Филатов, Е. И. Чигарина

Все современные автоматизированные системы используют для хранения данных либо хранилища, либо базы данных. Существует множество видов систем управления базами данных, поддерживающих реляционные, объектные, документно-ориентированные и другие модели данных. Задача данной работы – определение особенностей проектирования, а также достоинств и недостатков хранения, манипулирования данными в реляционных и документно-ориентированных базах данных, используя при этом системы управления базами данных MS SQL Server и MongoDB.

Ключевые слова: проектирование баз данных, концептуальная модель баз данных, логическая модель баз данных, реляционная модель баз данных, документно-ориентированная модель баз данных.

1. Анализ особенностей проектирования систем баз данных SQL и NoSQL

При проектировании баз данных выделяют такие основные этапы как концептуальное, логическое и физическое проектирование. На концептуальном уровне вне зависимости от структуры данных делается описание модели хранения в терминах модели «сущность-связь». Для перехода к логической модели для реляционных баз данных используются правила Джексона, а для перехода к документно-ориентированной модели таких правил нет.

С учетом особенностей структуры данных, реализуемой в MongoDB, в таблице 1 приведено сравнение правил перехода от концептуальной модели к логической модели реляционной базы данных и документно-ориентированной базы данных с учетом всех существующих видов связей, особенностей

классов принадлежностей сущностей и мощностей связей (табл. 1).

Базовыми понятиями в MongoDB являются коллекции и документы [2]. Коллекция является эквивалентом таблицы в реляционных системах управления базами данных. Коллекция существует в одной базе данных [2].

Документ – запись в коллекции и базовая единица данных в MongoDB [2]. Документы в коллекции могут иметь разные поля. Как правило, все документы в коллекции имеют аналогичное или связанное назначение. Документы аналогичны объектам JSON, но существуют в базе данных в формате с большим количеством типов, известным как BSON [2].

С учетом анализа особенностей структуры данных в рассматриваемой документно-ориентированной СУБД сформулированы следующие правила.

Таблица 1

Правила перехода от концептуальной модели к логической

Концептуальное проектирование S - сущность	Реляционное проектирование (методология IDEF1X) R - отношение	Документно-ориентированное проектирование C - коллекция

Правила перехода от концептуальной модели к документно-ориентированной модели базы данных:

1. Если степень связи 1:1 и класс принадлежности обоих сущностей является обязательным, то в документно-ориентированной модели создается один документ, одним из полей которого является вложенный документ, содержащий данные о второй сущности.

2. Если степень связи 1:1 и класс принадлежности одной сущности является обязательным, а другой – необязательным, то в документно-ориентированной модели создается один документ, одним из полей которого может быть вложенный документ, содержащий данные о второй сущности.

3. Если степень связи 1:1 и класс принадлежности ни одной сущности не является обязательным, то в документно-ориентированной модели создается два документа, при этом в каждом из них может содержаться поле, хранящее ссылку(идентификатор) на другой документ.

4. Если степень связи 1:n и класс принадлежности n-связной сущности является обязательным, то в документно-ориентированной модели возможны 2 варианта:

- один документ, одним из полей которого является вложенный документ, содержащий данные о второй сущности

- два документа, при этом второй документ содержит поле, хранящее ссылку(идентификатор) на первый документ.

5. Если степень связи 1:n, класс принадлежности n-связной сущности является необязательным и нет дополнительных полей, то в документно-ориентированной модели возможны 2 варианта:

- один документ, одним из полей которого может быть вложенный документ, содержащий данные о второй сущности

- два документа, при этом второй документ может содержать поле, хранящее ссылку(идентификатор) на первый документ.

Если степень связи 1:n, класс принадлежности n-связной сущности является необязательным и есть дополнительные поля, то создается два документа, при этом второй

документ содержит массив объектов следующего типа:

```
arr1:{{
    K1id: Objectld,
    data: String
}, ...
```

], где KNid – ссылка на другой документ

data – дополнительное поле

6. Если степень связи m:n и нет дополнительных полей, создается 2 документа, каждый из которых содержит массив ссылок на другой документ. Если степень связи m:n и есть дополнительные поля, создается 2 документа, каждый из которых содержит массив объектов следующего типа:

```
arrN:{{
    KNid: Objectld,
    data: String
}, ...
```

], где KNid – ссылка на другой документ

data – дополнительное поле

2. Сравнительный анализ применения правил перехода от концептуальной модели к логической для реляционных и документно-ориентированных баз данных на примере описания предметной области «Студент вуза»

Описанные правила использованы на примере концептуальной модели базы данных “Студент вуза”. Модель сущность-связь базы данных “Студент вуза” приведена ниже (рис. 1).

Согласно правилам Джексона, осуществлен переход к логической модели реляционной базы данных по методологии IDEF1X (рис. 2).

Согласно разработанным правилам перехода от концептуальной модели к логической, осуществлен переход к документно-ориентированной модели (рис. 3).

Видно, что по количеству объектов более предпочтительна документно-ориентированная модель (4 коллекции против 7 таблиц).

3. Разработка алгоритмов и программная реализация решения задачи сравнительного анализа выполнения

запросов в системах баз данных SQL и NoSQL на примере СУБД MS SQL Server и MongoDB

Для реализации спроектированных моделей реляционной и объектно-ориентированной базы даны были выбраны СУБД MS SQL Server и MongoDB. MS SQL Server – это мощная и надежная система управления данными, обеспечивающая

множество функций, защиту данных и высокую производительность для внедренных приложений-клиентов, «легких» веб-приложений и локальных хранилищ данных. SQL Server предназначен для развертывания и создания прототипов; его можно получить бесплатно и свободно распространять вместе с приложениями.

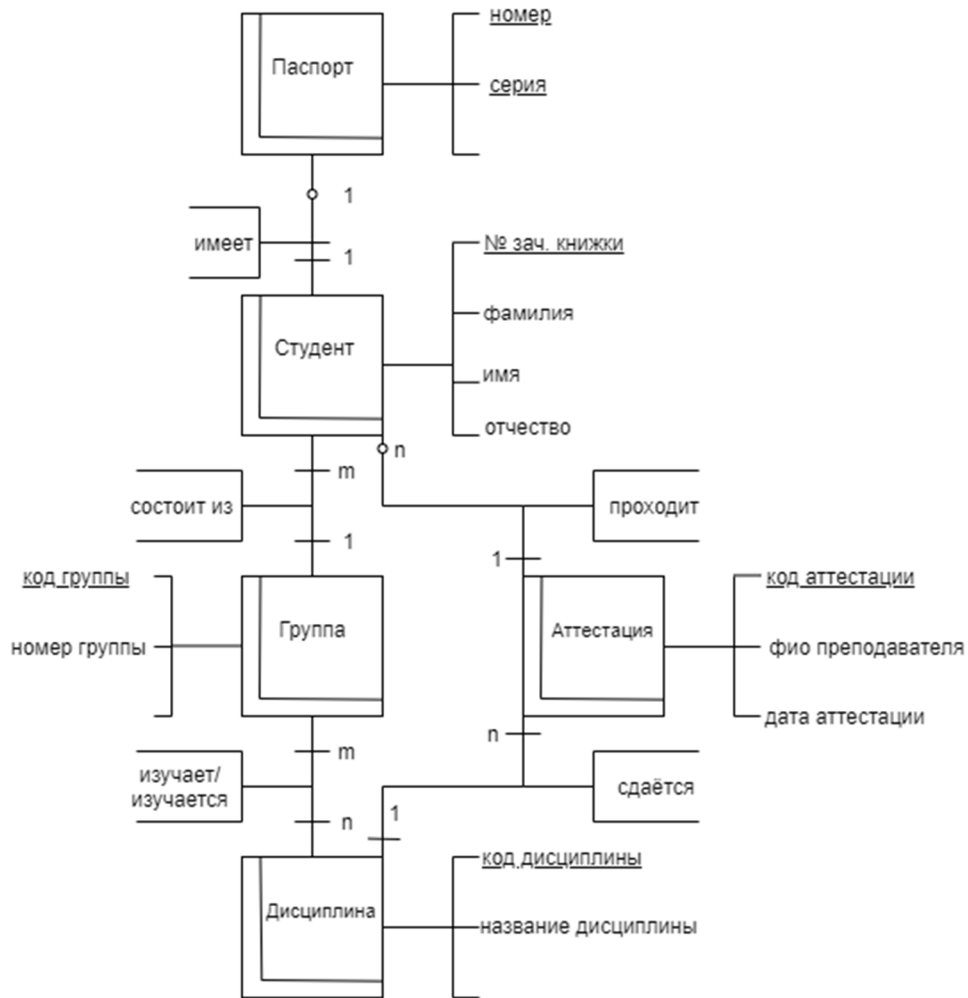


Рис. 1. Концептуальная модель базы данных «Студент вуза»

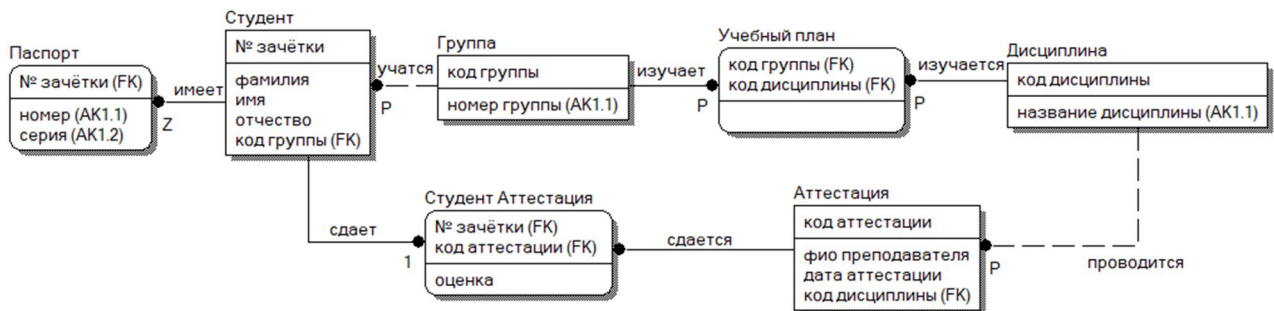


Рис. 2. Логическая модель базы данных «Студент вуза» по методологии IDEF1X

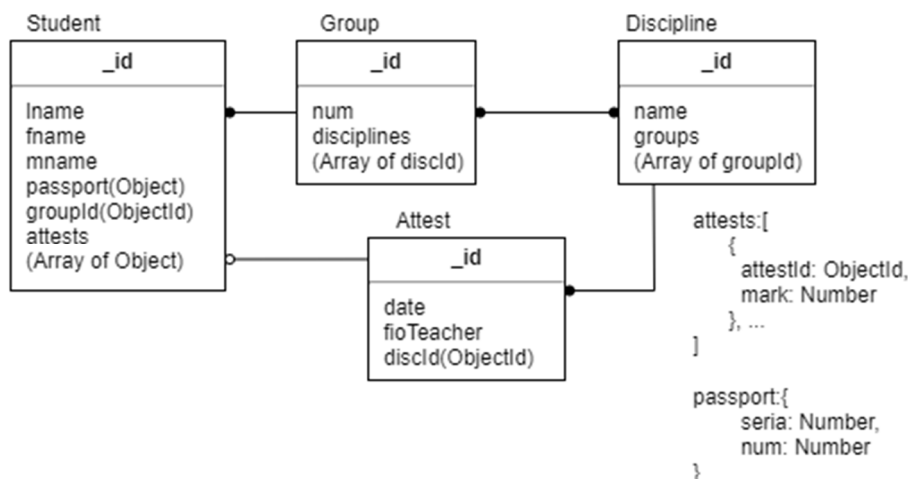


Рис. 3. Документно-ориентированная модель базы данных “Студент вуза”

MongoDB (от англ. humongous – огромный) – документно-ориентированная СУБД с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++. MongoDB реализует новый подход к построению баз данных, где нет таблиц, схем, запросов SQL, внешних ключей и многих других вещей, которые присущи реляционным базам данных.

Для подсчета времени выполнения запросов в MS SQL Server в работе использована утилита SQL Server Profiler – это интерфейс для создания трассировок и управления ими, а также для анализа и воспроизведения полученных результатов[3].

Для подсчета времени выполнения запросов в MongoDB – системная коллекция system.profile. Профилировщик базы данных собирает подробную информацию о командах базы данных, выполняемых для работающего экземпляра mongod. Профилировщик записывает все данные, которые он собирает, в коллекцию system.profile, ограниченную коллекцию в базе данных администратора[2].

Для заполнения реляционной и объектно-ориентированной базы данных тестовыми записями были написаны соответственно хранимые процедуры в SQL Server и программный код на языке JavaScript с использованием NodeJS.

Для проведения сравнительного анали-

за были написаны следующие группы запросов: запрос на выборку данных из одной таблицы, запрос на выборку данных из двух таблиц, запрос на выборку данных из двух таблиц с условием, запрос на выборку данных из двух таблиц с условием и сортировкой, запрос на выборку данных с группировкой и агрегированием данных, запрос на выборку данных с соединением всех таблиц базы данных, запрос на добавление данных, запрос на обновление данных, запрос на удаление данных.

Результаты измерения времени выполнения перечисленных запросов в разных системах для разного объема данных приведены в таблицах 2 и 3.

На рисунках 4, 5 и 6 представлены графики зависимости среднего времени выполнения запросов на выборку и удаление от количества записей в базе данных.

Таким образом можно сделать вывод, что большинство запросов выполняются быстрее в MongoDB, за исключением запросов, связанных с группировкой данных.

Заключение

По количеству объектов более предпочтительна документно-ориентированная модель (4 коллекции против 7 таблиц). По объему памяти, занимаемому данными для рассматриваемого примера при одинаковом количестве записей в SQL и NoSQL базах данных, выигрывает также документно-ориентированный подход.

Таблица 2

Запросы на выборку данных

100000 записей	MS SQL Server	MongoDB
1.1. выборка из одной таблицы	1957ms	310ms
1.2. выборка из двух таблиц	2251ms	417ms
1.3. выборка из двух таблиц с условием	475ms	181ms
1.4. выборка из двух таблиц с условием и сортировкой	555ms	272ms
1.5. выборка с группировкой	1270ms	17922ms
1.6. выборка с соединением всех таблиц	4980ms	53010ms
1000 записей	SQL Server	MongoDB
1.1. выборка из одной таблицы	121ms	4ms
1.2. выборка из двух таблиц	165.2ms	7ms
1.3. выборка из двух таблиц с условием	3.4ms	3ms
1.4. выборка из двух таблиц с условием и сортировкой	10ms	3ms
1.5. выборка с группировкой	15.6ms	327ms
1.6. выборка с соединением всех таблиц	622ms	551ms
10 записей	SQL Server	MongoDB
1.1. выборка из одной таблицы	1.8ms	0ms
1.2. выборка из двух таблиц	1.16ms	0ms
1.3. выборка из двух таблиц с условием	3.4ms	1ms
1.4. выборка из двух таблиц с условием и сортировкой	2.4ms	1ms
1.5. выборка с группировкой	1.7ms	4ms
1.6. выборка с соединением всех таблиц	343ms	29ms

Таблица 3

Запросы на изменение данных

100000 записей	SQL Server	MongoDB
2.1. вставка записи	3.2ms	0ms
2.2. обновление записи	1.2ms	0ms
2.3. удаление записи	112ms	64ms
2.4. вставка записи	7.8ms	55ms
2.5. обновление записи	9.8ms	0ms
2.6. удаление записи	1.8ms	0ms

Продолжение табл. 3

1000 записей	SQL Server	MongoDB
2.1. вставка записи	1.2ms	0ms
2.2. обновление записи	1.1ms	0ms
2.3. удаление записи	20.4ms	0ms
2.4. вставка записи	8.6ms	0ms
2.5. обновление записи	2ms	0ms
2.6. удаление записи	1.7ms	0ms
10 записей	SQL Server	MongoDB
2.1. вставка записи	2.6ms	0ms
2.2. обновление записи	1ms	0ms
2.3. удаление записи	11.8ms	0ms
2.4. вставка записи	2.6ms	0ms
2.5. обновление записи	1.8ms	0ms
2.6. удаление записи	1.5ms	0ms

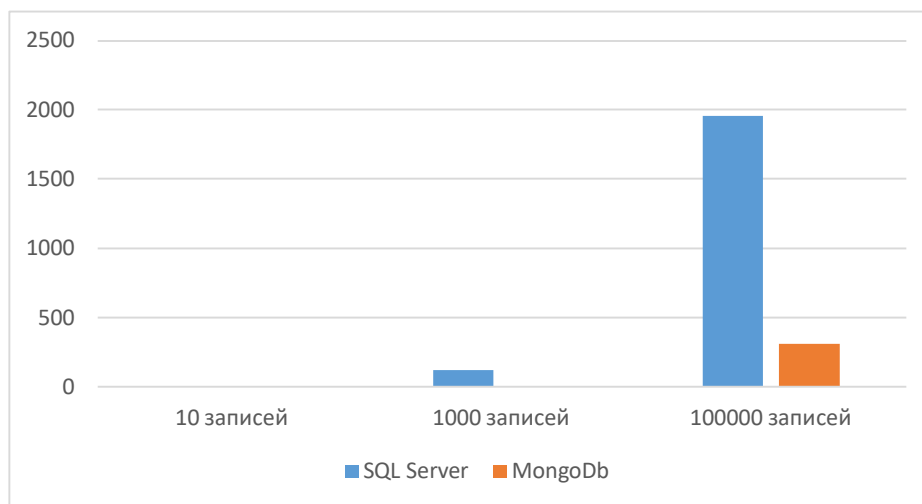


Рис. 4. График зависимости среднего времени выполнения запроса на выборку из одной таблицы от количества записей

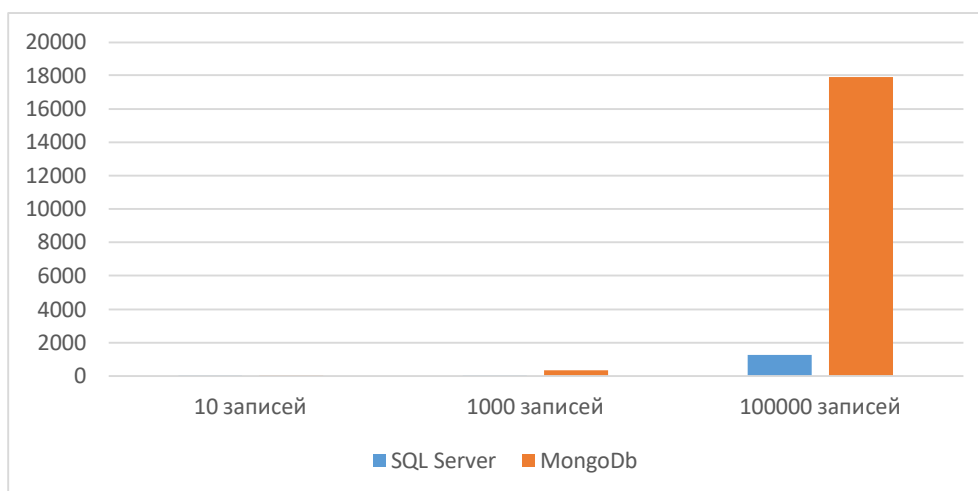


Рис. 5. График зависимости среднего времени выполнения запроса на выборку с группировкой от количества записей

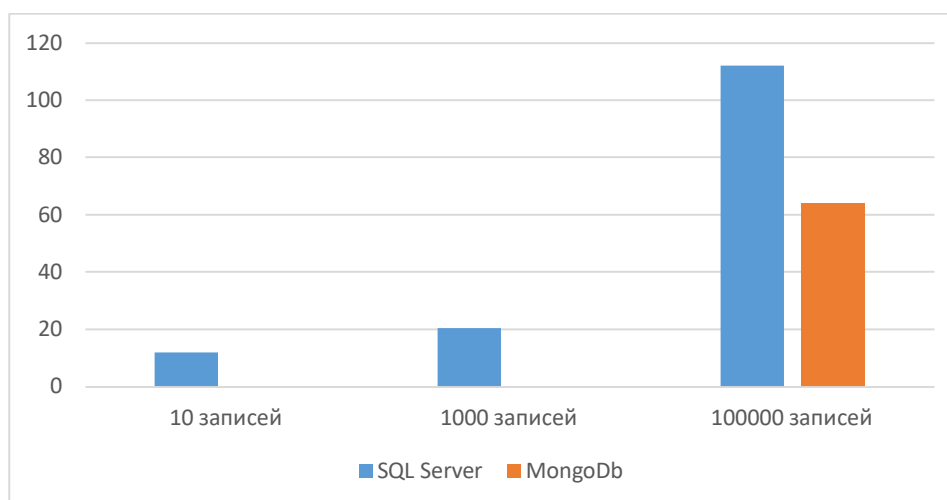


Рис. 6. График зависимости среднего времени выполнения запроса на удаление от количества записей

Литература

1. Правила порождения реляционных отношений из модели «сущность-связь». URL: http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_5.html (дата обращения: 04.02.2020).

2. Документация по MongoDB. URL:

<https://docs.mongodb.com/manual/reference/glossary/> (дата обращения: 10.02.2020).

3. Приложение SQL Server Profiler. URL: <https://docs.microsoft.com/ru-ru/sql/tools/sql-server-profiler/sql-server-profiler?view=sql-server-ver15> (дата обращения: 11.02.2020).

A COMPARATIVE ANALYSIS OF THE FEATURES OF DESIGNING AND REALIZATION OF REQUESTS IN RELATIONAL AND DOCUMENT-ORIENTED DATABASE USING DBMS SQL SERVER AND MONGODB AS AN EXAMPLE

V. V. Filatov, E. I. Chigarina

All modern automated systems use storages or database for data storage. There are many types of database management system, that assist relational, object, document-oriented and other data models. The goal of this work is determining of advantages and disadvantages, manipulation of data in relational and document-oriented database using the database management system MS SQL Server and MongoDB. The first stage of this task is comparing of rules of moving from conceptual model to logical database model. When designing any databases the first step is a conceptual designing. The most popular model of this stage is an entity-relationship model. For moving from conceptual model to logical model of relational database is used the rule of Jackson. The rules of moving to document-oriented database weren't described before.

Key words: database designing, conceptual model of database, logical model of database, relational model of database, document-oriented model of database.

Статья поступила в редакцию 08.07.2020 г.

© Filatov V. V., Chigarina E. I., 2020.

Filatov Vladislav Vyacheslavovich (phil182@mail.ru), undergraduate student of the Faculty of Informatics;

Chigarina Elena Ivanovna (chigarinaei@gmail.com),

associate professor of the Department of Information Systems and Technologies of the Samara University, 443086, Russia, Samara, Moskovskoye Shosse, 34.