Компьютерные инструменты в образовании, 2019
№ 3: 15–28
УДК: 004.652
http://cte.eltech.ru
doi:10.32603/2071-2340-2019-3-15-28

ОПРЕДЕЛЕНИЕ СОВОКУПНОСТИ КОЛЛЕКЦИЙ ДЛЯ БАЗ ДАННЫХ ТИПА КЛЮЧ-ДОКУМЕНТ ПО ЗАДАННОМУ НАБОРУ СВОЙСТВ ОБЪЕКТОВ И ЗАПРОСОВ К БАЗЕ ДАННЫХ *

Ха В. М.[1], аспирант, muon.ha@mail.ru
Шичкина Ю. А.[1], доктор физико-математических наук, ✉ strange.y@mail.ru
Костичев С. В.[1], кандидат технических наук, snenv@mail.ru

[1]Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), ул. Профессора Попова, д. 5, корп. 2, 197376, Санкт-Петербург, Россия

Аннотация

Проблема трансляции базы данных из одного формата в другой периодически появляется в разных организациях по разным причинам. Сегодня хорошо отработан механизм смены формата реляционных баз данных. Но с появлением новых типов баз данных, таких как NoSql, проблема трансляции вновь стала очень актуальной в связи с радикальным отличием способов организации данных в различных базах данных. В статье рассматривается формализованный метод, основанный на теории множеств, по выбору числа и состава коллекций для базы данных типа ключ-документ. Исходными данными являются свойства объектов, информация о которых хранится в базе данных, и совокупность запросов, которые наиболее часто выполняются. Рассмотренный метод можно применять не только при создании новой базы данных типа ключ-документ, но также при трансформации существующей, при переходе от реляционных баз данных к NoSql, при консолидации баз данных.

Ключевые слова: *NoSql, запрос к базе данных, коллекция, ключ-значение, трансляция данных, формат данных, оптимизация структуры базы данных.*

Цитирование: Ха В. М., Шичкина Ю. А., Костичев С. В. Определение совокупности коллекций для баз данных типа ключ-документ по заданному набору свойств объектов и запросов к базе данных // Компьютерные инструменты в образовании. 2019. № 3. С. 15–28. doi:10.32603/2071-2340- 2019-3-15-28

1. ВВЕДЕНИЕ

Для многих информационных систем необходимо надежное хранение информации с возможностью обновления и поиска данных. Эту задачу решают базы данных. Реляционные базы данных являются самыми распространенными формами хранения данных. Но кроме них существуют NoSQL и NewSQL.

* Исследование выполнено при финансовой поддержке РФФИ и СИТМА в рамках научного проекта № 18-57-34001.

В случае, когда изменяются требования к базе данных, становится целесообразным переход на другую систему управления базами данных, например перевод БД из реляционной в NoSQL. В этом случае необходимо произвести трансляцию данных. Другим примером является консолидация баз данных различного типа. В этом случае также перед процессом слияния данных необходимо привести данные к единому формату, то есть опять же необходима трансляция данных из формата одного типа СУБД в формат СУБД другого типа. Для автоматизации этого процесса служат трансляторы баз данных, однако в случае перехода, например, с MySQL на MongoDB существующих решений мало.

Консолидация и трансляция баз данных являются проблемой во многих областях применения баз данных, таких как гетерогенная интеграция баз данных, электронная коммерция, семантическая обработка запросов. В мире было проведено и проводится сегодня много исследований как отдельными учеными, так и исследовательскими организациями, и, как следствие, предлагаются различные решения для этой проблемы. Среди известных результатов в области консолидации и трансляции реляционных баз данных можно назвать системы SemInt [1], LSD (Learning Source Descriptions) в [2], SKAT (Semantic Knowledge Articulation Tool) [3], TranScm [4], Palopoli в [5], ARTEMIS [6], MOMIS (Mediator enviroment for Multiple Information Sources) [7].

Вопросы трансляции реляционных баз данных в NoSQL — это большая проблема, актуальная по срочности решения и мало изученная сегодня. Тем не менее, можно выделить следующие решения.

В [8] предложен способ трансляции реляционных баз данных в MongoDB путем преобразования таблиц в файлы CSV и импорта преобразованных файлов с помощью встроенной команды MongoDB. Однако этот метод только непосредственно переводит таблицы в коллекции без учета связей между ними. Hanine и соавторы в [9] разработали также подход к трансляции данных из реляционных баз данных в MongoDB, состоящий из трех этапов: извлечение данных из исходной базы данных, преобразование данных и перенос преобразованных данных в новую базу данных. Схожим технологиям посвящены исследования [10]. К сожалению, эти подходы не затрагивают вопросов зависимости производительности запросов от схем баз данных и связей между объектами.

Авторы статьи [11] предложили модель преобразования реляционной схемы в схему базы данных NoSQL на основе структуры данных и запросов к данным. Но в этой модели новая структура NoSql базы данных основывается только на метаданных об объектах и запросах и не включает информацию о связях объектов. А это неизбежно приводит к потере универсальности запросов.

В других случаях авторами описывается программная надстройка NoSQLayer [12, 13]. Эта программа не предназначена для миграции данных, и многие запросы работают медленнее, чем при непосредственном обращении к базе данных NoSQL на встроенном языке запросов [14].

Согласно [15], отношения в документно-оринтированной модели базы данных могут быть представлены в форме встроенных документов и связей между этими документами. Однако, во-первых, встроенные документы могут использоваться для ограниченного объема данных, а во-вторых, остается проблемой определение формы встраивания.

В статье [16] авторы применили традиционные правила теории нормализации реляционной схемы базы данных (теоремы 2NF, 3NF и BCNF) для разработки схемы MongoDB. Подход не учитывает наличие связности отношений типа «многие ко многим», первичные ключи и внешние ключи.

Трансляция данных между различными источниками данных является необходимым шагом для многих задач интеллектуального анализа данных. Однако различия между методами хранения данных в этих двух формах SQL и NoSQL ставят много проблем в области трансляции, трансформации и консолидации данных. Одна из них — это соответствие коллекций NoSQL таблицам реляционных баз данных. Для трансляции, трансформации и консолидации баз данных различного типа необходимо учитывать также отсутствие структуры и особенности языка запросов. Метод, представленный в данной статье, позволяет создавать совокупность коллекций MongoDB с учетом связей между реляционными таблицами и связей между таблицами и запросами.

2. ПОСТАНОВКА ЗАДАЧИ

Если для реляционных баз данных существуют методы формализованного построения отношений на основе заданных свойств объекта и функциональных зависимостей между ними, то для NoSQL и NewSQL такого формализованного аппарата не существует. И, например, при трансляции реляционной базы данных к формату MongoDB всегда встает вопрос, как этот перевод осуществлять.

При этом возможны следующие варианты изменения формата (перечень не полный):

— каждой таблице в реляционной базе данных поставить в соответствие отдельную коллекцию документов в MongoDB;
— из всех таблиц реляционной базы данных сделать одну коллекцию документов в MongoDB;
— создать такой набор коллекций документов в MongoDB, чтобы они наиболее полно подходили под выполняемые запросы.

Если с первыми двумя вариантами изменения формата данных все просто и понятно, то третий вариант приводит к проблеме: как выбрать этот набор коллекций и состав каждой коллекции.

В данной статье представлено одно из решений этой задачи — выбора числа и состава коллекций MongoDB по известному набору свойств объекта (или атрибутов реляционного отношения) и запросов к базе данных.

3. МЕТОД ОПРЕДЕЛЕНИЯ СОВОКУПНОСТИ КОЛЛЕКЦИЙ ДЛЯ БАЗ ДАННЫХ ТИПА КЛЮЧ-ДОКУМЕНТ ПО ЗАДАННОМУ НАБОРУ СВОЙСТВ ОБЪЕКТОВ И СТРУКТУРЕ ЗАПРОСОВ

3.1. Входные данные для метода определения структуры коллекций базы данных

В качестве входных данных выступают:

- Множество свойств объектов, хранимых в базе данных. При трансляции реляционной базы данных в MongoDB в качестве набора свойств объектов выступает множество полей всех таблиц, при создании базы данных не из реляционной базы данных в качестве набора свойств объектов может быть любая совокупность атрибутов. По определению «множества», среди его элементов не может быть дубликатов. Пусть T_r — это таблица реляционной базы данных, где r — номер таблицы, $r = 1..k$, k — число таблиц.

Пусть $T_{r,j}$ — это поле в таблице T_r, где j — номер поля, $j = 1..r_n$, r_n — число полей в r-й таблице.
Тогда множество полей одной таблицы — это множество вида:

$$T_r = \{T_{r,j},\ j = 1,2,..,r_n\}. \tag{1}$$

А множество всех полей реляционной базы данных будет определяться по формуле:

$$M = \{T_{r,j},\ r = 1,2,..,k \mid T_{r,j} \neq T_{q,i},\ \forall\, r, q \leq k,\ j \leq r_n,\ i \leq q_n\}, \tag{2}$$

где r — номер таблицы, $r = 1..k$, k — число таблиц, j — номер поля в таблице, $j = 1..r_n$, r_n — число полей в r-й таблице.
Длина множества — это количество его элементов. Обозначение: $|M|$ — длина множества M.

- Совокупность множеств полей, входящих в запрос:

$$S_i = \{T_{r,j}, r \leq k, j \leq r_n\}, \tag{3}$$

где i — номер запроса ($i = 1,2,..m$), m — число запросов к базе данных.

3.2. Выходные данные для метода определения структуры коллекций базы данных

Выходными данными является совокупность коллекций документов, выражаемая через множества атрибутов объектов:

$$V_i = \{T_{r,j}, r \leq k, j \leq r_n\} \tag{4}$$

удовлетворяющих условиям:

$$V_1 \cap V_2 \cap V_3 = \varnothing, \tag{5}$$

$$V_1 \cup V_2 \cup V_3 = M = \bigcap_{r=1}^{k} T_r, \tag{6}$$

$$(\forall S_i)(\exists V_j)(S_i \in V_j, S_i \notin V_i, i \neq j), \tag{7}$$

где i — номер коллекции ($i = 1,2,..l$), l — число коллекций.

3.3. Метод формирования коллекций для базы данных в формате ключ-документ

Шаг 1. Создать множества полей таблиц и запросов по формулам (2–3).

Шаг 2. Выбрать поля, которые не участвуют в запросах. Для этого:

2.1. Для каждого поля составить множества запросов, в которых это поле участвует в любой части конструкции:

$$T'_{r,j} = \{S_i, i = 1..p, p \leq m \mid T_{r,j} \in S_i\},$$

где m — число запросов, r — номер таблицы, $r = 1..k$, k — число таблиц, j — номер поля в таблице, $j = 1..r_n$, r_n — число полей в r-й таблице.

2.2. Все поля $T_{r,j}$, для которых длина множества $|T'_{r,j}| = 0$, могут быть включены в единую коллекцию MongoDB:

$$V_1 = \{T_{r,j}, r \le k, j \le r_n \mid |T'_{r,j}| = 0\},$$

где m — число запросов, r — номер таблицы, $r = 1..k$, k — число таблиц, j — номер поля в таблице, $j = 1..r_n$, r_n — число полей в r-й таблице.

Примечание: коллекции должны формироваться с учетом связей между таблицами. В терминах реляционной алгебры составление коллекции V_1 из пяти таблиц на шаге 2.2 означает:

$$V_1 = \pi(T1 \bowtie T2 \bowtie T3 \bowtie T4)_{T1_4, T2_2, T2_5, T3_5, T3_6, T4_2, T4_4, T4_5, T4_6},$$

где операции: $\bowtie$ — естественное соединение таблиц, $\pi(X)_{y,z}$ — проекция или вертикальный выбор в таблице X полей y, z.

Шаг 3. Выбрать поля, которые участвуют только в одном запросе, то есть $|T'_{r,j}| = 1$.

Если для любого $T_{r,j} \in S_i$ выполняется условие $|T'_{r,j}| = 1$, то все поля $T_{r,j}$ множества S_i должны войти в новую коллекцию:

$$V_p = \{T_{r,j}, r \le k, j \le r_n \mid T_{r,j} \in S_i \ \& \ |\{T_{r,j}\}| = 1\}, p > 1. \tag{8}$$

Шаг 4. Убрать из рассмотрения все поля, которые вошли в коллекции V_1 и V_p на шагах 1–3.

Шаг 5. Составить коллекции из полей, к которым обращается несколько запросов.

5.1. Составить множество рассматриваемых запросов: $I = \{S_i\}, i = 1..m$, где m — число рассматриваемых запросов.

5.2. Составить попарные пересечения множеств $S_i \in I$.

$$S'_k = S_i \cap S_j, \forall\, i \ne j;\ i, j = 1, 2, .., |I|;\ k = 1, 2, .., C^2_{|I|},$$

где $C^2_{|I|} = \frac{|I|!}{|(|I|-2)|!2!} = \frac{|I|(|I|-1)}{2}$.

5.3. Для полученных не пустых пересечений составить множество P из запросов, входящих в эти пересечения:

$$P = \{S'_i \mid |S'_i| \ne 0, \forall\, i \le k, k = 1, 2, .., C^2_{|I|}\}.$$

5.4. Найти разность множеств: $I - P$.

Если $I - P \ne \oslash$, то новая коллекция будет состоять из всех полей, входящих в запросы разности множеств $I - P$.

$$V_p = \bigcup_{i=1}^{|P|} S'_i, \forall\, S'_i \in P.$$

5.5. Положить $I = P$.

5.6. Из полученных пересечений найти новые не пустые пересечения по правилу:

$$if(\exists(S_i \cap S_j) \& \exists(S_j \cap S_k)), find(S_i \cap S_j \cap S_k)$$

или в общем виде:

$$S''_k = S'_i \cap S'_j,\ if\ \exists S_m \mid (S_m \in S'_i \& S_m \in S'_j, i = j); i, j = 1, 2, .., |I|; k = 1, 2, .., C^2_{|I|}.$$

5.7. Повторить шаги 5.3–5.7 до тех пор, пока не останется одно пересечение, то есть длина множества I не станет равной 1: $|I| = 1$.

Шаг 6. В новое отношение включить все поля, вошедшие в последнее единственное пересечение и не вошедшие в другие пересечения:

$$V_{(p+1)} = \grave{\mathrm{I}} - \bigcup_{i=1}^{p} V_i.$$

Шаг 7. Конец метода.

4. ПРИМЕР ПРИМЕНЕНИЯ МЕТОДА СОЗДАНИЯ СОВОКУПНОСТИ КОЛЛЕКЦИЙ БАЗЫ ДАННЫХ

Пусть дана база данных, состоящая из 5 отношений:

$T1(f1, f2, f3, f4, f5)$,
$T2(f1, f2, f3, f4, f5, f6)$,
$T3(f1, f2, f3, f4, f5, f6)$,
$T4(f1, f2, f3, f4, f5, f6)$,
$T5(f1, f2, f3)$.

Пусть даны запросы к базе данных:

1. Select $sum(f4 - f2), f1$ From $T1$ Where $f5 > a$ and $f5 < b$ Group by $f1$;
2. Select $count(*), f1$ From $T2$ Where $f6 > a$ and $f6 < b$ Group by $f1$;
3. Select $T3.f1, T3.f2$ From $T3, T4$ Where $T3.f3 = T4.f1$ and $T4.f3 = a$ and $t3.f6 = null$;
4. Select $T1.f2, T1.f3$ From $T1$ Where $T1.f1 = a$ and $f5 > a$ and $f5 < b$ Union Select $T2.f3, T2.f4$ From $T2$ Where $T2.f1 = a$ and $f6 > a$ and $f6 < b$;
5. Select $T1.f2, T1.f1$ From $T1$ Where $T1.f3 = a$ and $f5 > a$ and $f5 < b$ Union Select $T2.f1, T2.f4$ From $T2$ Where $T2.f3 = a$ and $f6 > a$ and $f6 < b$;
6. Select $T3.f1, T3.f2$ From $T3, T4$ Where $T3.f3 = T4.f1$ and $T4.f3 = a$ and $t3.f4 \geq a$ and $t3.f4 \leq b$ Order by $t3.f4$;
7. Select distinct $T3.f1, T4.f6$ From $T3, T4$ Where $T3.f1 = a$ and $T3.f3 = T4.f1$;
8. Select $T5.f2, T5.f3$ From $T3, T5$ Where $T3.f1 = a$ and $T5.f3 > a$ and $T5.f3 < b$ and $T3.f4 = T5.f1$;
9. Select $T5.f2, T5.f3$ From $T5$ Where $T5.f3$ in Select $max(T5.f3)$ From $T3, T5$ Where $T3.f1 = a$ and $T3.f4 = T5.f1$.

Задача: применяя метод из раздела 3, построить коллекции документов для заданных отношений.

Решение:

Шаг 1. Составим множество полей базы данных и множество запросов:

$$M = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9\}$$
$$S_1 = \{T1_1, T1_2, T1_4, T1_5\}$$
$$S_2 = \{T2_1, T2_6\}$$
$$S_3 = \{T3_1, T3_2, T3_3, T3_6, T4_1, T4_3\}$$
$$S_4 = \{T1_1, T1_2, T1_3, T1_5, T2_1, T2_3, T2_4, T2_6\}$$
$$S_5 = \{T1_1, T1_2, T1_3, T1_5, T2_1, T2_3, T2_4, T2_6\}$$
$$S_6 = \{T3_1, T3_2, T3_3, T3_4, T4_1, T4_3\}$$
$$S_7 = \{T3_1, T3_3, T4_1, T4_6\}$$
$$S_8 = \{T3_1, T3_4, T5_1, T5_2, T5_3\}$$
$$S_9 = \{T3_1, T3_4, T5_1, T5_2, T5_3\}$$

Шаг 2.

2.1. Для каждого поля составим множества запросов, в которых это поле участвует в любой части конструкции:

$$T1_1 = \{S_1, S_4, S_5\}$$
$$T1_2 = \{S_1, S_4, S_5\}$$
$$T1_3 = \{S_4, S_5\}$$

$$T1_4 = \{S_1\}$$
$$T1_5 = \{S_4, S_5\}$$
$$T2_1 = \{S_2, S_4, S_5\}$$
$$T2_2 = \varnothing$$
$$T2_3 = \{S_4, S_5\}$$
$$T2_4 = \{S_4, S_5\}$$
$$T2_5 = \varnothing$$
$$T2_6 = \{S_2, S_4, S_5\}$$
$$T3_1 = \{S_3, S_6, S_7, S_8, S_9\}$$
$$T3_2 = \{S_3, S_6\}$$
$$T3_3 = \{S_3, S_6, S_7\}$$
$$T3_4 = \{S_6, S_8, S_9\}$$
$$T3_5 = \varnothing$$
$$T3_6 = \{S_3\}$$
$$T4_1 = \{S_3, S_6, S_7\}$$
$$T4_2 = \varnothing$$
$$T4_3 = \{S_3, S_6\}$$
$$T4_4 = \varnothing$$
$$T4_5 = \varnothing$$
$$T4_6 = \{S_7\}$$
$$T5_1 = \{S_8, S_9\}$$
$$T5_2 = \{S_8, S_9\}$$
$$T5_3 = \{S_8, S_9\}$$

Шаг 5.

5.1. Составим множество рассматриваемых запросов: $I = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9\}$.

5.2. Составим попарные не пустые пересечения множеств S_i:

$$S_1 \cap S_4 = \{T1_1, T1_2, T1_5\}$$
$$S_1 \cap S_5 = \{T1_1, T1_2, T1_5\}$$
$$S_2 \cap S_4 = \{T2_1, T2_3, T2_4, T2_6\}$$
$$S_2 \cap S_5 = \{T2_1, T2_3, T2_4, T2_6\}$$
$$S_3 \cap S_6 = \{T3_1, T3_2, T3_3, T4_1, T4_3\}$$
$$S_3 \cap S_7 = \{T3_1, T3_3, T4_1\}$$
$$S_3 \cap S_8 = \{T3_1\}$$
$$S_3 \cap S_9 = \{T3_1\}$$
$$S_4 \cap S_5 = \{T1_1, T1_2, T1_3, T1_5, T2_1, T2_3, T2_4, T2_6\}$$
$$S_6 \cap S_7 = \{T3_1, T3_3, T4_1\}$$
$$S_6 \cap S_8 = \{T3_1, T3_4\}$$
$$S_6 \cap S_9 = \{T3_1, T3_4\}$$
$$S_7 \cap S_8 = \{T3_1\}$$
$$S_7 \cap S_9 = \{T3_1\}$$
$$S_8 \cap S_9 = \{T3_1, T3_4, T5_1, T5_2, T5_3\}.$$

5.3. Для полученных не пустых пересечений составим множество P из запросов, входящих в эти пересечения:

$$P = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9\}.$$

5.4. Найдем разность множеств: $I - P = \varnothing$. Следовательно, новая коллекция на этом шаге получена не будет.

5.5. $I = P$.

5.6. Из полученных пересечений найдем новые не пустые пересечения по правилу:

$$S_1 \cap S_4 \cap S_5 = \{T1_1, T1_2, T1_5\}$$
$$S_2 \cap S_4 \cap S_5 = \{T2_1, T2_3, T2_4, T2_6\}$$
$$S_3 \cap S_6 \cap S_7 = \{T3_1, T3_3, T4_1\}$$
$$S_3 \cap S_6 \cap S_8 = \{T3_1, T3_4\}$$
$$S_3 \cap S_6 \cap S_9 = \{T3_1, T3_4\}$$
$$S_3 \cap S_7 \cap S_8 = \{T3_1\}$$
$$S_3 \cap S_7 \cap S_9 = \{T3_1\}$$
$$S_6 \cap S_7 \cap S_8 = \{T3_1\}$$
$$S_6 \cap S_7 \cap S_9 = \{T3_1\}$$
$$S_7 \cap S_8 \cap S_9 = \{T3_1\}.$$

Продолжая итерационно шаги 5.3–5.7, получим пересечение, состоящее из одного элемента:

$$S_3 \cap S_6 \cap S_7 \cap S_8 \cap S_9 = \{T3_1\}.$$

Поэтому следующей итерации не будет.

Шаг 6. В новое отношение включим все поля, вошедшие в последнее единственное пересечение и не вошедшие в другие пересечения:

$$V_3 = \{S_3 \cup S_6 \cup S_7 \cup S_8 \cup S_9\} - \{V_1 \cup V_2\}$$
$$V_3 = \{T3_1, T3_2, T3_3, T3_6, T4_1, T4_3, T3_4, T4_6, T5_1, T5_2, T5_3\} -$$
$$\{T1_1, T1_2, T1_4, T1_5, T2_1, T2_6, T1_3, T2_3, T2_4\}$$
$$V_3 = \{T3_1, T3_2, T3_3, T3_6, T4_1, T4_3, T3_4, T4_6, T5_1, T5_2, T5_3\}.$$

Шаг 7. Конец метода.

В результате выполнения этого метода будут получены три коллекции:

$$V_1 = \{T2_2, T2_5, T3_5, T4_2, T4_4, T4_5\}$$
$$V_2 = \{T1_1, T1_2, T1_4, T1_5, T2_1, T2_6, T1_3, T2_3, T2_4\}$$
$$V_3 = \{T3_1, T3_2, T3_3, T3_6, T4_1, T4_3, T3_4, T4_6, T5_1, T5_2, T5_3\}.$$

При этом будут выполняться все свойства (5–7).

5. ТЕСТИРОВАНИЕ МЕТОДА СОЗДАНИЯ СОВОКУПНОСТИ КОЛЛЕКЦИЙ БАЗЫ ДАННЫХ

Одним из интересных и показательных результатов тестирования была реализация методов трансляции в MongoDB реляционной базы данных, состоящей из 5 отношений, аналогов отношений $T1 - T5$, описанных выше, и запросов 1–8, описанных также выше (рис. 1).

При тестировании применялись следующие четыре подхода к трансляции реляционной базы данных в MongoDB:

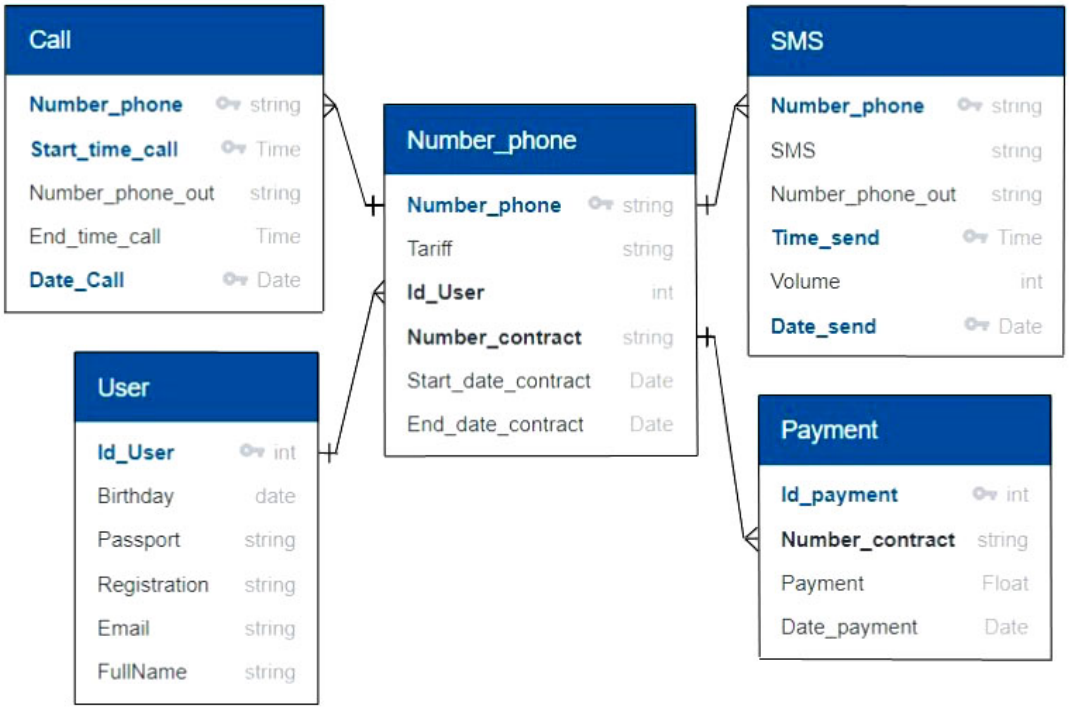

Рис. 1. Исходная схема базы данных на MySQL

1) «прямая» трансляция: каждой таблице в реляционной базе данных была поставлена в соответствие отдельная коллекция документов в MongoDB;
2) «обобщенная» трансляция: из всех таблиц реляционной базы данных была создана одна коллекция документов в MongoDB;
3) «пользовательская» трансляция: была выбрана совокупность коллекций MongoDB по опыту авторов;
4) «формализованная» трансляция: создан такой набор коллекций документов в MongoDB, чтобы они наиболее полно подходили под выполняемые запросы (по методу, описанному в данной статье).

Для оценки эффективности каждого из четырех подходов к трансляции реляционной базы данных в MongoDB было проведено тестирование на различных объемах реляционной базы данных. Тестовые образцы имеют следующие характеристики:

- Объём 1: User(10.000), Number_phone(10.000), Payment(25.000), SMS(43.000), Call(32.000)
- Объём 2: User(50.000), Number_phone(50.000), Payment(65.000), SMS(81.000), Call(32.000)
- Объём 3: User(100.000),Number_phone(100.000),Payment(123.000), SMS(230.000), Call(320.000)

Для объективности результатов тестирование каждого запроса проводилось 25 раз. На диаграммах ниже (рис. 2) представлено среднее время выполнения запросов за 25 раз.

Из диаграмм с рис. 2 видно, что метод, представленный в данной статье, не является лучшим для всех случаев. Но, по сравнению с остальными подходами, этот метод обладает стабильностью. Так, если, «обобщенная» трансляция в отдельных случаях, например,

для запросв 4, 6, 7 или 8 дает лучшее время, то для запросов 1, 2 и 5 «обобщенная» трансляция работает с большим отставанием от остальных трех подходов. То же самое можно сказать и для «непосредственной» трансляции: для каких-то запросов она быстрее остальных подходов, для других запросов — это самая медленная трансляция. На этом фоне «формализованная» трансляция занимает по скорости стабильное второе место.

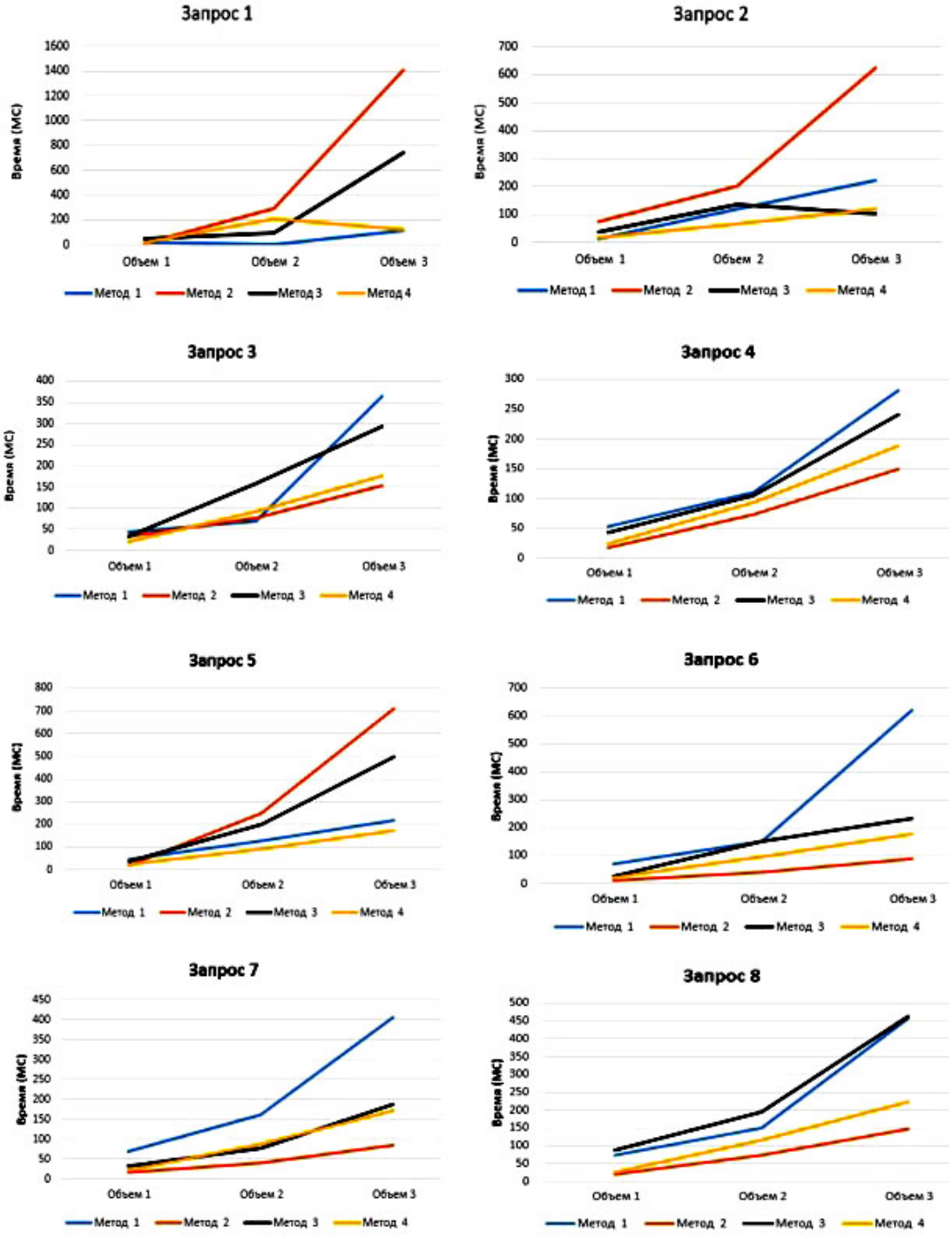

Рис. 2. Время выполнения запросов в MongoDB

6. ЗАКЛЮЧЕНИЕ

В результате проведенных исследований был разработан формализованный метод, основанный на теории множеств и позволяющий автоматизировать процесс построения коллекций для СУБД MongoDB по заданной совокупности свойств объектов и с учетом их вхождений в запросы к базе данных, а также определить число и состав коллекций для СУБД MongoDB. Этот метод может быть использован как при трансляции базы данных из формата SQL в формат MongoDB, так и для определения коллекций в новой базе данных MongoDB. В настоящий момент проводится тестирование данного метода на базах данных различного объема и сложности и исследования по части оптимизации коллекций MongoDB с учетом надичия функциональных зависимостей между свойствами объектов и возможности построения вложенных документов.

Список литературы

1. *Li W., Clifton C.* Semantic Integration in Heterogeneous Databases Using Neural Networks // Proc. 20th Int. Conf. Very Large Data Bases, 1994. P. 1–12.
2. *Doan A., Domingos P., Levy A.* Learning source description for data integration // Proc. Int'l Workshop on The Web and Databases (WebDB-2000). 2000. P. 81–86.
3. *Miller R. J., Haas L. M., Hernandez M.* Schema Mapping as Query Discovery // Proc. 26th Int. Conf. Very Large Data Bases. 2000. Cairo. P. 77–88.
4. *Milo T., Zohar S.* Using schema matching to simplify heterogeneous data translation // VLDB, 1998. P. 1–21.
5. *Palopoli L. ,Sacca D, Ursino D.* An automatic technique for detecting type conflits in database shemes. Proceedings of the seventh international conference on Information and knowledge management, 1998. P 306–313. doi: 10.1145/288627.288671
6. *Castano S., De Antonellis V.* A schema analysis and reconciliation tool environment for heterogeneous databases. Proceedings. IDEAS'99 // Int. Database Eng. Appl. Symp. (Cat. No.PR00265). 1999. doi: 10.1109/IDEAS.1999.787251
7. *Bergamaschi S., Castano S., Vincini M., Beneventano D.* Semantic integration of heterogeneous information sources // Data Knowl. Eng., 2001. Vol. 36. № 3. P. 215–249. doi:10.1016/S0169-023X(00)00047-1
8. *Chickerur S.* Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications // Int. Conf. on Advanced Software Engineering & Its Applications (ASEA–2015). Jeju, 2015. P. 41–47. doi: 10.1109/ASEA.2015.19
9. *Hanine M., Bendarag A., Boutkhoum O.* Data Migration Methodology from Relational to NoSQL Databases. World Academy of Science, Engineering and Technology // International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2015. Vol. 9. № 12. P. 2566–2570.
10. *Karnitis G., Arnicans G.* Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation // 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN–2015). Riga, 2015. P. 113–118. doi: 10.1109/CICSyN.2015.30
11. *Li X., Ma Z., Chen H.* QODM: A Query-Oriented Data Modeling Approach for NoSQL Databases // IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA–2014). Ottawa, 2014. P. 338–345. doi: 10.1109/WARTIA.2014.6976265
12. *Rocha L., Vale F., Cirilo E.* A Framework for Migrating Relational Datasets to NoSQL // Procedia Computer Science, 2015. Vol. 51. P. 2593–2602. doi: 10.1016/j.procs.2015.05.367
13. *Liang D., Lin Y., Ding G.* Mid-model Design Used in Model Transition and Data Migration between Relational Databases and NoSQL Databases // IEEE Int. Conf. on Smart City/SocialCom/SustainCom (SmartCity–2015). Chengdu, 2015. P. 866–869. doi: 10.1109/SmartCity.2015.177SmartCity

14. *Hamid S., Rezapour M., Moradi M., Ghadiri N.* Performance evaluation of SQL and MongoDB databases for big ecommerce data // 2015 Int. Symp. on Computer Science and Software Engineering (CSSE), Tabriz, Iran, August 2015. P. 259–268. doi: 10.1109/CSICSSE.2015.7369245
15. *Mason R. T.* NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database // Proceedings of Informing Science & IT Education Conference (InSITE). 2015. P. 259–268.
16. *Gu Y., Shen S., Wang J., Kim J.-U.* Application of NoSQL Database MongoDB // IEEE International Conference on Consumer Electronics (2015), Taipei, Taiwan, 2015. P. 158–159. doi: 10.1109/ICCE-TW.2015.7216831

Поступила в редакцию 15.08.2019, окончательный вариант — 07.09.2019.

Ха Ван Муон, аспирант кафедры вычислительной техники СПбГЭТУ «ЛЭТИ», muon.ha@mail.ru

Шичкина Юлия Александровна, доктор физико-математических наук, профессор кафедры вычислительной техники СПбГЭТУ «ЛЭТИ», ✉ strange.y@mail.ru

Костичев Сергей Валентинович, кандидат технических наук, доцент кафедры вычислительной техники СПбГЭТУ «ЛЭТИ», snenv@mail.ru

Computer tools in education, 2019
№ 3: 15–28
http://cte.eltech.ru
doi:10.32603/2071-2340-2019-3-15-28

Determining the Composition of Collections for Key-Document Databases Based on a Given Set of Object Properties and Database Queries

Ha V. M.[1], postgraduate student, muon.ha@mail.ru
Shichkina Yu. A.[1], PhD, professor, ✉ strange.y@mail.ru
Kostichev S. V.[1], PhD, snenv@mail.ru

[1]Saint Petersburg Electrotechnical University,
5, building 2, st. Professora Popova, 197376, Saint Petersburg, Russia

Abstract

The work of transforming a database from one format periodically appears in different organizations for various reasons. Today, the mechanism for changing the format of relational databases is well developed. However, with the advent of new types of databases, such as NoSQL, this problem is prevalent due to the radically different ways of data organization at the various databases. This article discusses a formalized method based on set theory, at the choice of the number and composition of collections for a key-value type database. The initial data are the properties of objects, about which information is stored in the database, and the set of queries that are most frequently executed. The considered method can be applied not only when creating a new keyvalue database, but also when transforming an existing one, when moving from relational databases to NoSQL, when consolidating databases.

Keywords: *NoSql, database query, collection, key-value, data translation, data format, database structure optimization.*

Citation: V. M. Ha, Yu. A. Shichkina, and S. V. Kostichev, "Determining the Composition of Collections for Key-Document Databases Based on a Given Set of Object Properties and Database Queries," *Computer tools in education*, no. 3, pp. 15–28, 2019 (in Russian); doi:10.32603/2071-2340-2019-3-15-28

Acknowledgements: *The reported study was funded by RFBR and CITMA according to the research project No. 18-57-34001.*

References

1. W. Li and C. Clifton, "Semantic Integration in Heterogeneous Databases Using Neural Networks," in *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 1–12.
2. A. Doan, P. Domingos, and A. Levy, "Learning source description for data integration," in *Proc. Int'l Workshop on The Web and Databases* (WebDB-2000), 2000, pp. 81–86.
3. R. J. Miller, L. M. Haas, and M. Hernandez, "Schema Mapping as Query Discovery," in *Proc. 26th Int. Conf. Very Large Data Bases*, Cairo, Egypt, 2000, pp. 77–88.
4. T. Milo and S. Zohar, "Using schema matching to simplify heterogeneous data translation," in *VLDB*, 1998, pp. 1–21.
5. L. Palopoli, D. Sacca, and D. Ursino, "An automatic technique for detecting type conflits in database shemes," in *Proc. of the 7th int. conf. on Information and knowledge management*, Bethesda, MD, USA, 1998, pp. 306–313; doi: 10.1145/288627.288671
6. S. Castano and V. De Antonellis, "A schema analysis and reconciliation tool environment for heterogeneous databases," in *Proc. IDEAS'99. Int. Database Eng. Appl. Symp. (Cat. No.PR00265)*, 1999; doi: 10.1109/IDEAS.1999.787251
7. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano, "Semantic integration of heterogeneous information sources," *Data Knowl. Eng.*, vol. 36, no. 3, pp. 215–249, 2001; doi: 10.1016/S0169-023X(00)00047-1
8. S. Chickerur, "Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications," in *Int. Conf. on Advanced Software Engineering & Its Applications* (ASEA), Jeju, South Korea, 2015, pp. 41–47; doi: 10.1109/ASEA.2015.19
9. M. Hanine, A. Bendarag, and O. Boutkhoum, "Data Migration Methodology from Relational to NoSQL Databases," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 12, pp. 2566–2570, 2015.
10. G. Karnitis and G. Arnicans, "Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation," in *7th International Conference on Computational Intelligence, Communication Systems and Networks* (CICSyN), Riga, Latvia, 2015, pp. 113–118; doi: 10.1109/CICSyN.2015.30
11. X. Li, Z. Ma, and H. Chen, "QODM: A Query-Oriented Data Modeling Approach for NoSQL Databases," in *IEEE Workshop on Advanced Research and Technology in Industry Applications* (WARTIA), Ottawa, ON, Canada, 2014, pp. 338–345; doi: 10.1109/WARTIA.2014.6976265
12. L. Rocha, F. Vale, and E. Cirilo, "A Framework for Migrating Relational Datasets to NoSQL," *Procedia Computer Science*, vol. 51, pp. 2593–2602, 2015; doi: 10.1016/j.procs.2015.05.367
13. D. Liang, Y. Lin, and G. Ding, "Mid-model Design Used in Model Transition and Data Migration between Relational Databases and NoSQL Databases," in *IEEE Int. Conf. on Smart City/SocialCom/SustainCom* (SmartCity), Chengdu, China, 2015, pp. 866–869; doi: 10.1109/SmartCity.2015.177
14. S. Hamid, M. Rezapour, M. Moradi, and N. Ghadiri, "Performance evaluation of SQL and MongoDB databases for big ecommerce data," in *2015 Int. Symp. on Computer Science and Software Engineering* (CSSE), Tabriz, Iran, August 2015; doi: 10.1109/CSICSSE.2015.7369245
15. R. T. Mason, "NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database," in *Proc. Inf. Sci. IT Educ. Conf.* (InSITE), 2015, pp. 259–268.

16. Y. Gu, S Shen, J. Wang, and J.-U. Kim, “Application of NoSQL Database MongoDB,” in *IEEE International Conference on Consumer Electronics*, Taipei, Taiwan, 2015, pp. 158–159; doi: 10.1109/ICCE-TW.2015.7216831

Received 15.08.2019, the final version — 07.09.2019.

Ha Van Muon, postgraduate student, Department of Computer Engineering, Saint Petersburg Electrotechnical University, muon.ha@mail.ru

Yulia A. Shichkina, professor, Department of Computer Engineering, Saint Petersburg Electrotechnical University, ✉ strange.y@mail.ru

Sergey V. Kostichev, PhD, associate professor, Department of Computer Engineering, Saint Petersburg Electrotechnical University, snenv@mail.ru