УДК 004.622:004.654

МОДИФИКАЦИЯ АЛГОРИТМОВ НОРМАЛИЗАЦИИ ТАБЛИЦ БАЗ ДАННЫХ ДЛЯ ПРИМЕНЕНИЯ В СИСТЕМАХ BIG DATA

А.А. Карпук
Профессор кафедры программного обеспечения сетей телекоммуникаций Белорусской государственной академии связи, кандидат технических наук, доцент
a_karpuk@mail.ru

Л.С. Лазута
Аспирант кафедры телекоммуникационных систем Белорусской государственной академии связи, магистр
lenya.lazuta@mail.ru

А.А. Карпук

Окончил Белорусский государственный университет, автор более 260 опубликованных научных трудов, включая 3 монографии. Область научных интересов связана с моделированием и оптимизацией сложных систем, проектированием баз данных и хранилищ данных, оценкой качества радиосвязи и оптимизацией присвоения радиочастот в радиосетях и радиолиниях.

Л.С. Лазута

Окончил Белорусскую государственную академию связи, имеет 8 опубликованных научных трудов. Область научных интересов связана с моделированием и оптимизацией сложных систем, проектированием баз данных и хранилищ данных.

Аннотация. Рассмотрены классические алгоритмы нормализации таблиц в реляционных базах данных. Показаны недостатки алгоритмов, ограничивающие их применение для нормализации таблиц в системах Big Data. Предложены модификации алгоритмов Делобеля-Кейси, Бернштейна и Ислура, которые имеют полиномиальную сложность и могут применяться в системах Big Data для автоматического приведения таблиц данных к третьей нормальной форме.

Ключевые слова: большие данные, функциональная зависимость, третья нормальная форма, алгоритмы нормализации таблиц, минимальное покрытие, замыкание атрибутов, ключ таблицы.

Введение. Современные системы Big Data могут иметь архитектуру реляционного хранилища данных (Relational Data Warehouse, RDW), озера данных (Lake Data), современного хранилища данных (Modern Data Warehouse, MDW), фабрики данных (Data Fabric), озерного хранилища данных (Data Lakehouse), сетки данных (Data Mesh) [1]. Во всех перечисленных архитектурах, кроме архитектуры Lake Data, на одном или нескольких этапах обработки данных решаются задачи очистки данных, предварительной обработки данных и нормализации данных в виде приведения данных к третьей нормальной форме (3НФ). Возможные подходы к решению задач очистки и предварительной обработки данных рассмотрены в работах [2, 3]. Для решения задачи нормализации данных требуется знание функциональных зависимостей (ФЗ), существующих между данными. В работе [4] рассматривались методы выделения и алгоритмы поиска ФЗ между данными в системах Big Data. В настоящей работе рассмотрены классические алгоритмы приведения таблиц реляционной базы данных к 3НФ Делобеля-Кейси, Бернштейна, Ислура и Неклюдовой-Цаленко, показаны недостатки алгоритмов, ограничивающие их применение для нормализации таблиц данных в системах Big Data, и предложены модификации алгоритмов Делобеля-Кейси, Бернштейна и Ислура, которые могут применяться в системах Big Data для автоматического приведения таблиц данных к 3НФ.

Классические алгоритмы нормализации таблиц баз данных. Пусть $X \subset A$ – подмножество атрибутов (признаков) системы Big Data, ZA – некоторый атрибут. Говорят, что существует ФЗ XZ, если любой комбинации значений атрибутов из X всегда соответствует единственное значение атрибута Z. Очевидно, что из $Z \in X$ следует, что XZ. Такая ФЗ, в которой зависимый атрибут входит в состав левой части ФЗ, называется тривиальной. В дальнейшем будем рассматривать только нетривиальные ФЗ между атрибутами. Структура ФЗ на множестве атрибутов удовлетворяет аксиомам Армстронга [5]. Для задания структуры ФЗ, отличающейся от тривиальной, требуется постулировать конечное множество ФЗ $F=\{Fj=XjYj|Xj,YjA,j=1,m\}$, которое называется системой образующих структуры ФЗ на A. Структуру ФЗ, заданную системой образующих F, будем обозначать S(F). Замыканием множества XA относительно структуры ФЗ S(F) называется множество $X+(F)A$, такое, что для любого YA из XY следует $YX+(F)$.

Структуры ФЗ S(F1) и S(F2) на множестве A с системами образующих $F1=\{Xi1Yi1|Xi1 \subset A,Yi1 \subseteq A,i=1,m1\}$ и $F2=\{Xj2Yj2|Xj2 \subset A,Yj2 \subseteq A,j=1,m2\}$ соответственно называются эквивалентными, если для любого $X \subset A$ имеет место равенство $X+(F1)=X+(F2)$. В работе [6] было доказано, что структуры ФЗ S(F1) и S(F2) на множестве A эквивалентны тогда и только тогда, когда для всех $i=1,m1$ выполняются условия $Xi1+(F1)=Xi1+(F2)$ и для всех $j=1,m2$ выполняются условия $Xj2+(F1)=Xj2+(F2)$.

Система образующих $E=\{HjTj|Hj,TjA,j=1,m\}$ структуры ФЗ S(E) на множестве A называется минимальным покрытием (minimal coverage), если выполняются следующие условия:

для любого j1,m и любого XTj система образующих E', полученная из E путем замены ФЗ HjTj на ФЗ $HjTj\backslash X$, задает структуру ФЗ S(E'), не эквивалентную структуре ФЗ S(E);

для любого j1,m и любого YHj система образующих E', полученная из E путем замены ФЗ HjTj на ФЗ $Hj\backslash YTj$, задает структуру ФЗ S(E'), не эквивалентную структуре ФЗ S(E).

В русскоязычной литературе минимальное покрытие иногда называют элементарным базисом. Классический алгоритм Делобеля-Кейси [7] состоит из следующих шагов.

1. Разложить ФЗ так, чтобы в правой части каждой ФЗ был только один атрибут. Каждая ФЗ XjYj из F, где Yj={Aj1, Aj2, ...}, заменяется на несколько ФЗ с одним атрибутом в правой части XjAj1, XjAj2, ….

2. Удалить избыточные атрибуты в левых частях полученных ФЗ. Атрибут BXj является избыточным в левой части ФЗ XjAj1, если Aj1(Xj\B)+. При построении замыкания рассматриваемая ФЗ не учитывается.

3. Удалить избыточные ФЗ из полученного множества ФЗ. ФЗ XjAj1 является избыточной, если после ее удаления выполняется условие Aj1(Xj)+.

4. Все полученные ФЗ с одинаковыми левыми частями собираются в одну ФЗ, в правой части которой находится объединение атрибутов правых частей собранных ФЗ.

5. На основе каждой полученной ФЗ формируется таблица, содержащая все атрибуты, находящиеся в левой и правой части этой ФЗ. Первичным ключом таблицы объявляется множество атрибутов левой части ФЗ.

6. Если среди полученных таблиц найдутся две таблицы, такие что все атрибуты первой таблицы входят в множество атрибутов второй таблицы, то первая таблица удаляется. При этом проверка ФЗ всех неключевых атрибутов первой таблицы от первичного ключа этой таблицы должна выполняться при обработке строк второй таблицы.

7. Построить замыкания первичных ключей полученных таблиц относительно полученной структуры ФЗ. Если ни одно из замыканий не совпадает с множеством атрибутов A, то найти любой ключ этого множества атрибутов и добавить к полученным таблицам в 3НФ таблицу, состоящую из атрибутов найденного ключа. Для поиска ключа применяется алгоритм последовательного удаления атрибутов из списка всех атрибутов и построения замыкания оставшихся атрибутов. Если удаленный атрибут не входит в полученное замыкание, то он входит в состав ключа и возвращается в список атрибутов.

Делобель и Кейси доказали, что полученные таблицы находятся в 3НФ, и исходная таблица восстанавливается из полученных таблиц с помощью операций естественного соединения. Легко убедиться, что шаги 1-4 алгоритма Делобеля-Кейси строят минимальное покрытие исходной структуры ФЗ на множестве атрибутов A.

Классический алгоритм Бернштейна [8] состоит из следующих шагов.

1-4. Построить минимальное покрытие структуры ФЗ аналогично шагам 1-4 алгоритма Делобеля-Кейси.

5. Построить замыкание левой части каждой из полученных ФЗ и разбить ФЗ на группы таким образом, чтобы в одной группе были все ФЗ с совпадающими замыканиями левых частей.

6. На основе каждой полученной группы ФЗ формируется таблица, содержащая все атрибуты, находящиеся в левых и правых частях всех ФЗ группы. Левая часть одной из ФЗ группы (любой) объявляется первичным ключом таблицы, левые части остальных ФЗ объявляются уникальными потенциальными ключами таблицы.

7. При формировании таблиц возможен случай, когда в таблице, содержащей более одного ключа, неключевой атрибут, попавший в таблицу из правой части одной ФЗ (ключа), окажется в частичной ФЗ от правой части другой ФЗ, попавшей в таблицу.(другого ключа). Такие атрибуты удаляются из таблицы. Для поиска таких атрибутов последовательно рассматриваются неключевые атрибуты таблицы, каждый атрибут временно удаляется из таблицы и из той ФЗ, из которой он попал в таблицу, и строится замыкание оставшихся атрибутов таблицы. Если рассматриваемый атрибут принадлежит этому замыканию, то его действительно надо удалить из таблицы, в противном случае он возвращается в таблицу.

8. Этот шаг аналогичен шагу 6 алгоритма Делобеля-Кейси.

9. Этот шаг аналогичен шагу 7 алгоритма Делобеля-Кейси.

Бернштейн доказал, что полученные таблицы находятся в 3НФ, и исходная таблица восстанавливается из полученных таблиц в 3НФ с помощью операций естественного

соединения. Классический алгоритм Ислура [9] совпадает с алгоритмом Бернштейна, за исключением того, что на шаге 9 для поиска ключа всего множества атрибутов используется усовершенствованный алгоритм. Из исходного списка всех атрибутов сразу удаляются все атрибуты, не входящие в левые части исходных ФЗ, а затем оставшиеся атрибуты проверяются на вхождение в ключ таким же образом, как в алгоритмах Делобеля-Кейси и Бернштейна.

На первых шагах классического алгоритма Неклюдовой–Цаленко [10] также строится минимальное покрытие, затем находятся все ключи таблицы, и множество всех ключевых атрибутов таблицы рассматривается как полная подструктура структуры ФЗ. Биекции всех ключей включаются в систему образующих структуры ФЗ, в результате чего в минимальном покрытии могут появиться транзитивные ФЗ, которые удаляются. Производится расширение подструктуры путем добавления неключевых атрибутов, функционально полно зависящих от ключей. Если все атрибуты таблицы войдут в формируемую подструктуру, то она находится в 3НФ. В противном случае анализируются ФЗ, оставшиеся в минимальном покрытии. Если в них входят не все атрибуты таблицы, то множество атрибутов этих ФЗ рассматривается как полная подструктура структуры ФЗ, для которой все шаги алгоритма повторяются, иначе применяется алгоритм Бернштейна.

Модификация алгоритмов для применения в системах Big Data. Сразу отметим, что поиск всей ключей таблицы, используемый в алгоритме Неклюдовой-Цаленко, является NP-трудной задачей, поэтому применение алгоритма Неклюдовой-Цаленко для автоматической нормализации таблиц систем Big Data, содержащих сотни атрибутов и сотни исходных ФЗ между атрибутами, невозможно.

В алгоритмах Делобеля-Кейси, Бернштейна и Ислура имеется шаг поглощения меньшей таблицы большей таблицей, если все атрибуты меньшей таблицы входят в состав большей таблицы (шаг 6 в алгоритме Делобеля-Кейси). Это фактически ведет к потере ФЗ минимального покрытия, на которой построена меньшая таблица. В базах данных поддержка таких ФЗ производится средствами СУБД с помощью специально написанных триггеров. Большинство проектировщиков баз данных по этой причине не выполняют шаг 6 алгоритма Делобеля-Кейси и шаг 8 алгоритмов Бернштейна и Ислура. Лучше иметь в базе данных лишнюю таблицу, чем решать проблему с поддержкой ФЗ между атрибутами. Мы считаем, что при использовании алгоритмов Делобеля-Кейси, Бернштейна и Ислура для автоматической нормализации таблиц в системах Big Data этот шаг следует исключить из алгоритмов.

Во всех рассмотренных алгоритмах при построении минимального покрытия исходной структуры ФЗ сначала все ФЗ раскладываются на ФЗ, содержащие один атрибут в правой части, а после удаления избыточных атрибутов в левых частях и избыточных ФЗ снова собираются в ФЗ с одинаковыми левыми частями. Применяемый алгоритм удаления избыточных атрибутов в левых частях и избыточных ФЗ основан на том, что в правой части ФЗ находится один атрибут. В работе [11] Карпук предложил алгоритм построения минимального покрытия структуры ФЗ, в котором не требуется раскладывать ФЗ на ФЗ, содержащие один атрибут в правой части, а затем собирать ФЗ с одинаковыми левыми частями. Этот алгоритм состоит из следующих шагов.

1. Удалить избыточные атрибуты из левых частей ФЗ из F. Атрибут BXj называется избыточным в Xj, если B(Xj\B)+(F′), где через F′ обозначена система образующих структуры ФЗ, полученная из F путем удаления ФЗ XjYj.

2. Если в результате получим в F две или более ФЗ с одинаковыми левыми частями, то объединить их в одну ФЗ.

3. Удалить избыточные атрибуты из правых частей ФЗ из F. Атрибут BYj называется избыточным в Yj, если B(Xj)+(F′), где через F′ обозначена система образующих структуры ФЗ, полученная из F путем замены ФЗ XjYj на XjY\Bj.

4. Удалить из F ФЗ с пустыми правыми частями.

Карпук доказал, что описанный алгоритм строит минимальное покрытие структуры ФЗ, совпадающее с минимальным покрытием, построенным по шагам 1-4 алгоритма Делобеля-Кейси. Отказ от разложения ФЗ по одному атрибуту в правой части позволяет избежать увеличения мощности множества ФЗ и приводит к снижению вычислительной сложности построения минимального покрытия с O(m2k2ln) до O(m2(l+k)n), где n – количество атрибутов, m – количество ФЗ, k – среднее количество атрибутов в правой части ФЗ, l – среднее количество атрибутов в левой части ФЗ. Предложенный подход обеспечивает асимптотическое ускорение, особенно заметное при наличии ФЗ с большой мощностью правой части, что характерно для систем Big Data. Еще более заметно преимущество алгоритма Карпука для построения минимального покрытия структуры ФЗ при оценке объема оперативной памяти, необходимой для реализации алгоритма. Для реализации алгоритма Делобеля-Кейси объем требуемой оперативной памяти оценивается величиной O(mkl), а для реализации алгоритма Карпука величиной Om(k+l). Таким образом, при использовании алгоритмов Делобеля-Кейси, Бернштейна и Ислура для автоматической нормализации таблиц в системах Big Data вместо шагов 1-4 для построения минимального покрытия структуры ФЗ следует использовать алгоритм Карпука.

В таблицах систем Big Data количество атрибутов, не входящих в левые части ФЗ, может быть очень большим, поэтому при использовании алгоритмов Делобеля-Кейси, Бернштейна и Ислура для автоматической нормализации таблиц в системах Big Data для поиска ключа всего множества атрибутов следует использовать алгоритм Ислура.

Список литературы

[1] Serra James. Deciphering Data Architectures. Choosing Between a Modern Data Warehouse,Data Fabric, Data Lakehouse, and Data Mesh. O'Reilly 2024. – 252 p.

[2] Лазута Л.С. Методы и алгоритмы очистки и предварительной обработка данных в хранилищах данных // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы XXV Междунар. науч.-техн. конф., 13-14 мая 2025 г. / редкол.: А.О. Зеневич [и др.]. – Минск: Белорусская государственная академия связи, 2025. – С. 153–154.

[3] Lazuta L.S., Karpuk A.A. Cleaning and Preprocessing of Data in Data Warehouses // International Journal of Engineering Research and Development. – 2025. – Vol. 21, Issue 9. – P. 263–269.

[4] Карпук А.А., Лазута Л.С. Выделение и поиск функциональных зависимостей между атрибутами в системах Big Data // 11-ая Международная научно-технической конференция «BIG DATA and Advanced Analytics», 23-24 апреля 2025 г. БГУИР, Минск, Беларусь. – С. 397–405.

[5] Armstrong W.W. Dependency Structure of Data Base Relationships // Proc. IFIP Congress. – Geneva, Switzerland, 1974. – P. 580–583.

[6] Карпук А.А. Выбор элементарного базиса структуры функциональных зависимостей при проектировании базы данных // Вопросы радиоэлектроники. Сер. ОВР. – 1983. – Вып. 6. – С. 38–41.

[7] Delobel C., Casey R.G. Decomposition of a data base and the theory Boolean switching functions // IBM J. Res. And Dev. – 1973. – Vol. 17, No 5. – P. 374–386.

[8] Bernstein P.A. Synthesizing third normal form relations from functional dependencies // ASM Transactions on Database Systems. – 1976. – Vol. 1, No 4. – P. 277–298.

[9] Isloor S.S. An algorithm with logical simplicity for designing third normal form relational database schema from functional dependencies // Proc. of Int. Conf. on DBMSs (ICMOD 78). – Fast Milan, Italy, 1978. – P. 31–50.

[10] Неклюдова Е.А., Цаленко М.Ш. Синтез логической схемы реляционной базы данных // Программирование. – 1979. – № 6. – С. 58–68.

[11] Карпук А.А. О построении элементарного базиса системы функциональных зависимостей в базе данных // Информационные технологии и программные средства: проектирование, разработка и применение: Сб. научн. ст. – Гродно: ГрГУ, 2011. – С. 185–190.

Авторский вклад

Карпук Анатолий Алексеевич – руководство исследованием алгоритмов автоматической нормализации таблиц в таблицах систем Big Data, постановка задач исследования, оценка результатов исследования и определение направлений дальнейшей работы по модификации алгоритмов.

Лазута Леонид Сергеевич – анализ алгоритмов автоматической нормализации таблиц в таблицах систем Big Data, определение области применения алгоритмов, разработка предложений по модификации алгоритмов.

MODIFICATION OF DATABASE TABLE NORMALIZATION ALGORITHMS FOR USE IN BIG DATA SYSTEMS

A.A. Karpuk
Professor, Department of Telecommunication Network Software, Belarusian State Academy of Communications, PhD of Technical sciences, Associate Professor

L.S. Lazuta
Postgraduate student, Department of Telecommunication Systems, Belarusian State Academy of Communications, Master's Degree

Abstract. Classic algorithms for normalizing tables in relational databases are examined. The algorithms' shortcomings that limit their use for normalizing tables in Big Data systems are highlighted. Modifications to the Delobel-Casey, Bernstein, and Isloor algorithms are proposed. These algorithms have polynomial complexity and can be used in Big Data systems to automatically convert data tables to third normal form.

Keywords: big data, functional dependency, third normal form, table normalization algorithms, minimum coverage, attribute closure, table key.