

©Мендкович Н. А., 2016

DOI: 10.18255/1818-1015-2016-2-153-163

УДК 517.9

Об эффективности минимизирующего подхода к оптимизации запросов

Мендкович Н. А.

получена 4 апреля 2016

Аннотация. Стандартной проблемой использования СУБД является недостаток эффективности и высокая стоимость доступа к хранимым данным. Допустимый уровень работы системы может достигаться с помощью технологий оптимизации запросов, определяющих наиболее эффективный способ выполнения конкретного запроса с помощью его модификации и определения возможных планов выполнения.

Целью данной работы является доказательство эффективности алгоритмов минимизации запроса, основанных на минимизации ограничения запроса и удаления избыточных условий.

Статья представляет алгоритмы минимизации, основанные на математических преобразованиях, определяющих и удаляющих избыточные условия из ограничения запроса, чтобы упростить его. Она включает алгоритмы, основанные на технологиях «поглощения условий», первичных импликант и минимизации множеств линейных неравенств.

Работа также включает теоретическое доказательство эффективности минимизирующего подхода, основанного на упрощении ограничения. Мы также рассматриваем экспериментальные результаты применения этих технологий оптимизации и их влияния на скорость обработки запроса. В конце мы представляем обзор влияния минимизации запроса на весь процесс оптимизации запроса.

Ключевые слова: оптимизация запросов, лексическая оптимизация запросов

Для цитирования: Мендкович Н. А., "Об эффективности минимизирующего подхода к оптимизации запросов", *Моделирование и анализ информационных систем*, **23:2** (2016), 153–163.

Об авторах:

Мендкович Никита Андреевич, orcid.org/0000-0003-1694-7947, ООО «Фринет Групп», инженер, Ленинский проспект, 47, Москва, 119991 Россия, e-mail: mend@f-group.ru

Введение

Решение задачи оптимизации запроса к СУБД путем минимизации его лексического представления относится к числу классических идей в теории баз данных. Применение минимизирующих преобразований запроса направлено на минимизацию числа содержащихся в нем условий путем исключения из его состава плеоназмов. Указанное преобразование оказывает влияние на ход и выполнения, и оптимизации запроса.

Указанный подход был представлен в классической работе П. Холла, в которой рассматривается применение приемов реляционной алгебры по преобразованию логических выражений в целях оптимизации запросов. Основные сформулированные им идеи лексической оптимизации путем упрощения запроса сводятся к следующему: объединение последовательности проекций в одну; исключение избыточных операций; упрощение выражений, использующих пустые отношения и тривиальные условия; вынос общих выражений «за скобки» [1].

Подходы, основанные на минимизации запроса, применяются также в современных СУБД. Например, в системе Oracle, подобная процедура производится над подзапросами на основании «свойства включения» («блок запроса X включает другой блок запроса Y, если результат Y является подмножеством (не обязательно собственным) результата X») [2]. Судя по корпоративным отчетам о модернизации оптимизирующих программ СУБД, минимизирующие процедуры также используются в MySQL [3], PostgreSQL [4] и Oracle [5].

Однако в литературе долгое время не были представлены исследования, теоретически обосновывающие эффективность минимизирующих преобразований, а также ее экспериментальные подтверждения. Кроме того, отсутствует общая характеристика места минимизирующих алгоритмов в процессе оптимизации запросов, особенно в контексте наличия оптимизационных процедур, ведущих к увеличению длины запроса и созданию дополнительных условий. В частности, это характерно для «улучшающих» преобразований запросов на основе так называемых «магических множеств» [6, 7].

Решению указанной проблемы посвящена настоящая статья. В разделе 1 описаны минимизирующие алгоритмы, избранные в качестве объекта изучения. В разделе 2 приводится теоретическое обоснование эффективности описанных минимизирующих преобразований. В разделе 3 описаны результаты экспериментального изучения результатов минимизации запросов. В разделе 4 описаны примеры запросов, содержащих избыточности описанных ниже типов, а также сферы применения СУБД, для которых они наиболее характерны. В разделе 5 анализируется прямое и косвенное влияние предложенных алгоритмов на процесс оптимизации запроса с помощью иных алгоритмов.

1. Описание минимизирующих алгоритмов оптимизации

В настоящей главе дано краткое описание трех алгоритмов оптимизации, описанных и опубликованных автором в более ранних работах [8–10]. Указанные алгоритмы позволяют преобразовать условия SQL-запросов, чьи ограничения представимы в виде ДНФ (дизъюнктивная нормальная форма), к наиболее краткой форме путем устранения избыточных условий, что может способствовать решению задачи определения вхождения для некоторых классов запросов.

Опишем формы плеоназмов, которые могут встречаться в составе ограничений запросов.

1. Плеоназм одного конъюнкта, т. е. включение в один и тот же конъюнкт запрос с одинаковым множеством переменных, характеризующих одно и то же

множество атрибутов, из которых некоторые являются логическим подмножеством других (лексическая тавтология) или образующих вместе тождественно истинное или ложное выражение.

2. Плеоназм множества конъюнктов, где однокоренные условия находятся в различных конъюнктах, однако в результате их присутствия запрос содержит смысловую избыточность.
3. Плеоназм общих атрибутов, при котором условия, входящие в состав запросов, не имеют общего корня, но корни двух ограничений имеют общие атрибуты, что приводит к возникновению плеоназма.

Особенностью предлагаемых алгоритмов являются способы минимизации ограничений, представимых в виде логической функции ДНФ:

- минимизация конъюнктов путем определения плеоназмов;
- минимизация ограничения в ДНФ путем поиска общих импликант;
- минимизация конъюнктов и ограничений, включающих условия, содержащие 2 и более атрибутов, путем определения условий — «следствий» из представленных условий.

Программная реализация представленного комплекса алгоритмов рассматривается в качестве оптимизатора, основанного на правилах (rule based optimization), т. е. преобразует запрос в более оптимальное, чем исходное, представление, что будет доказано в тексте ниже.

Границы применимости указанных алгоритмов определяются грамматикой вводимых выражений. Алгоритмы предназначены только для обработки выражений, содержащих ограничения, представимые в виде булевых функций в ДНФ.

Формальное описание указанных ограничений запросов представлено с помощью формальной грамматики на основе подмножества правил формы Бакуса–Наура (так называемые BNF-грамматики) для языка SQL-2003 [11] (рис. 1).

```
<boolean value expression> ::= <boolean term>|<boolean value expression>  
OR <boolean term>  
<boolean term> ::= <boolean factor>|<boolean term> AND <boolean factor>  
<boolean factor> ::= [ NOT ] <boolean test>  
<boolean test> ::= <boolean primary> [ IS [ NOT ] <truth value> ]  
<truth value> ::= TRUE | FALSE | UNKNOWN  
<boolean primary> ::= <predicate>  
<predicate> ::= <left paren> <boolean value expression> <right paren>  
<left paren> ::= (  
<right paren> ::= )  
<boolean value expression> ::= <variable><operator><number>  
<variable> ::= <sumstring>|<sumstring><sign><sumstring>  
<sumstring> ::= <string>[<degree>]|  
<string> ::= <letter>|<letter><string>
```

```

<letter> ::= A|...|Z|*|/
<degree> ::= ^<number>
<number> ::= <figure> | <figure><number>
<figure> ::= 0|...|9
<operator> ::= >|<|=|>|<=|
<sign> ::= +|-

```

Рисунок 1

Fig. 1

Представленное множество правил отличается от канонического отсутствием возможности создания вложенных ограничений в составе запросов. Однако данное ограничение само по себе не является препятствием для оптимизации ограничений, содержащих вложения, так как существует ряд алгоритмов, позволяющих преобразовывать ограничения, содержащие вложенные подзапросы, исключая вложения [12, 13].

Кроме этого, необходимо отметить следующие обстоятельства. Для некоторых запросов оптимизация путем исключения плеоназмов может сократить число задействованных в нем отношений, обращение к которым присутствует в избыточном конъюнкте, а также присутствие атрибутов, которые в конечном представлении не предусмотрены запросом.

Примером таких запросов может служить поиск данных сотрудников по нескольким суммам критериев с учетом атрибутов подразделений, в которых они работают. Пример ограничения такого запроса представлен на рис. 2.

```

...WHERE (status.person = 'phd' AND salary.person > 500) OR
(salary.person > 1000 AND person.dep = name.department AND
loc.department = 'Moscow' AND status.person = 'phd')

```

Рисунок 2

Fig. 2

В приведенном примере обращение к отношению department является избыточным, так как задействовано исключительно в рамках логически избыточного конъюкта, реализация которого не влияет на окончательную выборку.

2. Теоретическая оценка эффективности минимизации запроса

Эффективность указанных алгоритмов определяется влиянием минимизации ограничения на стоимость его обработки. Опубликованные работы предлагают следующую формулу, описывающую затраты на обработку ограничения запроса, представимого в виде ДНФ, с учетом числа и структуры условий (Наиболее ранняя публикация, видимо, [14]):

$$T = \sum_{i=1}^N \left(\sum_{j=1}^{M_j} t(C_j^i) \cdot \prod_{l=0}^{l=j-1} s \left(C_l^j \cdot \prod_{l=0}^{l=j-1} (1 - s(C_l^j)) \right) \right). \quad (1)$$

В формуле (1): t — стоимость обработки условия, N — число конъюнктов в функции, M_j — число условий в конъюнкте j , C_j^i — обозначает условие j конъюнкта i , $s(C_j^i)$ — селективность условия (при $s(C_0^i) = 1$).

В рамках предложенной модели стоимость обработки ограничения (T) может быть уменьшена в результате исключения условия C_j или группы условий $\bigcap_{j=1}^X C_j^a$ как плеоназма. Или же в результате его идентификации как условия-дубликата и использования вместо обработки данного условия ранее полученных результатов. Т. е. необходимо идентифицировать пары условий, где $C_j^i \in C_q^i$, или пары конъюнктов a и b , где $\bigcap_{j=1}^X C_j^a \in \bigcap_{i=1}^Y C_i^b$.

Представленные выше алгоритмы позволяют идентифицировать ряд указанных пар, которые не поддаются идентификации с помощью иных опубликованных на сегодняшний день алгоритмов оптимизации запросов.

Пусть N — множество кортежей в исходной таблице, которые необходимо проверить на соответствие условию C_j^i при обработке запроса, а $s(C_j^i)$ — селективность условия, являющегося частью ограничения, представимого в ДНФ. При этом подразумевается, что в рамках конъюнкта обращение к исходным данным производится один раз при обработке первого конъюнкта, а затем алгоритм обрабатывает промежуточные результаты, полученные при обработке предыдущего условия.

Пусть в конъюнкт i входит M и более условий, часть которых являются избыточными некоему условию C_q^i . Обработка данного конъюнкта в неоптимизированном ограничении потребует проверки следующего числа кортежей в основной таблице и промежуточных результатов (формула (2)):

$$T = s(C_1^i) + \sum_{j=2}^M t(C_j^i) \cdot \prod_{l=0}^{l=j-1} s(C_l^i) \quad (2)$$

В реальном выражении избыточные C_q^j условия будут иметь селективность равную 1 при применении после него. (Существующие алгоритмы оптимизации предполагают продвижение условий с наименьшей селективностью и стоимостью к началу конъюнкта, чтобы удешевить обработку, а селективность C_q^i всегда меньше или равна избыточным выражениям, см., например, [15, 16].)

Таким образом, избыточные затраты при обработке плеоназмов в составе конъюнкта могут быть определены формулой (3):

$$T_{\text{избыточное}} = \sum_p \left(t(C_p^i) \cdot \prod_{l=0}^q s(C_l^i) \right), \quad (3)$$

где p — число, принадлежащее множеству Q , а Q — множество чисел-номеров условий конъюнкта, избыточных относительно C_q^i .

Эффективность оптимизации ограничения, завершающейся исключением избыточного конъюнкта, отличается с учетом вида содержащейся избыточности.

В случае конъюнкта, удаляемого простым поглощением, избыточные затраты на обработку относительно невелики, так как требуется обработка лишь одного

условия, чтобы установить, что исходное множество данных больше не содержит кортежей, отвечающих данному конъюнкту, избыточному относительно обработанного ранее (если СУБД не допускает повторный выбор одних и тех же кортежей в рамках обработки одного ограничения).

Таким образом, избыточной является обработка одного условия избыточного конъюнкта, если изначально обработка происходит оптимальным образом, т.е. начинается с проверки соответствия условий, однокоренных обработанным ранее. В этом случае избыточность определяется формулой (4):

$$T_{\text{избыточное}} = t(C_j^i) \cdot \prod_{l=0}^{l=j-1} (1 - s(C^j)). \quad (4)$$

Избыточность, устраняемая путем «склейки» конъюнктов на основе обнаружения общих импликант, равна общей стоимости обработки избыточного конъюнкта (формула (5)):

$$T_{\text{избыточное}} = \sum_{j=1}^{M_i} t(C_j^i) \cdot \prod_{l=0}^{l=j-1} s(C_l^i). \quad (5)$$

Третий вид оптимизации, разработанный нами на основе алгоритма минимизации систем линейных неравенств, позволяет удалять оба описанных выше вида избыточности, а также путем анализа ограничения идентифицировать случаи, когда они являются тождественно ложными или тождественно истинными. Таким образом, в части возможных запросов оптимизирующее воздействие минимизации ограничения — несомненно. Формулы (3)—(5) позволяют оценить затраты на обработку избыточных условий, устранимых при использовании представленных алгоритмов.

3. Экспериментальная оценка эффективности минимизации

В данном разделе мы описываем проведение тестирования и его результаты. Оно осуществляется с помощью реализации ряда запросов к описанной выше базе данных в среде Oracle Database 10g.

Все запросы были направлены на выборку равного числа атрибутов из таблиц одинаковой размерности и отличались исключительно составом ограничения. Каждый из них содержал те или иные виды плеоназмов. В запросах не используются вложенные подзапросы. Ограничение состояло из множества условий, представленных в дизъюнктивной нормальной форме.

Порядок эксперимента имел следующий вид:

- реализация запроса в первичной форме (время выполнения T_1);
- выполнение программы оптимизации запроса (T_{opt});
- реализация оптимизированного запроса (T_2).

Эффективность алгоритма определялась соотношением

$$I_{eff} = (T_2 + T_{opt})/T_1, \quad (6)$$

где I_{eff} — коэффициент эффективности оптимизации.

Кроме того, целью эксперимента является оценка взаимосвязи числа условий в составе запроса со скоростью его обработки, для чего также проводится учет двух других показателей:

- число условий в составе запроса до оптимизации (A_1);
- число условий в составе запроса после оптимизации (A_2).

Результаты тестов представлены в таблице 1

Таблица 1. Результаты тестов (время указывается в секундах)

	T_1	T_{opt}	T_2	I_{eff}	A_1	A_2
1	0,3	0,00008	0,1	0,3336	3	1
2	0,5	0,00011	0,1	0,2002	6	2
3	0,8	0,00008	0,15	0,1876	5	2
4	0,4	0,00010	0,2	0,5003	7	4
5	0,8	0,00008	0,2	0,2501	7	3
6	0,14	0,00007	0,03	0,2148	5	3
7	0,06	0,00008	0,01	0,1680	4	3
8	0,02	0,00009	0,00(9)*	0,4545	9	6
9	0,23	0,00008	0,02	0,0873	3	1
10	0,15	0,00008	0,08	0,5339	4	2

Примечание: (*) Менее 0,01 секунды, приводится максимально возможное значение из-за погрешности измерения.

Коэффициент эффективности оптимизации демонстрирует возможность сокращения стоимости обработки запроса на 47%—84%. Причем временные затраты на собственно оптимизацию запроса пренебрежимо малы в сравнении с затратами на его обработку. Проведенные тесты показывают, что они в 10^3 — 10^4 раз меньше, чем временные затраты системы на выполнение запроса.

Таким образом, экспериментально подтвержден оптимизирующий эффект минимизации запроса, что определяется:

- пренебрежимо малыми затратами на выполнение минимизирующих алгоритмов относительно общей стоимости обработки запроса;
- обнаруженным оптимизирующим эффектом минимизации запросов, содержащих избыточные условия.

4. Примеры возникновения избыточности, устраняемой путем минимизации

Вопрос о реальном распространении неоптимальных сложных запросов, включающих множество конъюнктов условий, в реальной работе современных БД и, следовательно, практическом значении представленных алгоритмов является дискуссионным. Зачастую они предполагают логическую ошибку пользователя при составлении сложных запросов, включающих множество конъюнктов условий, и указание избыточных или логически противоречивых критериев.

Для многих реальных СУБД такого рода запросы нетипичны, но существует ряд исключений. Для работы многих систем все же характерны задачи выделения множеств сущностей, отвечающих одной из сумм представленных критериев.

Подобные запросы на практике встречаются в работе систем, предназначенных для хранения и обработки персональных данных большого числа индивидов. В частности, внутренние запросы в системах онлайн рекламы представимы в виде сложных булевых функций, содержащих множество условий (характеристик пользователей). Также наличие подобных запросов характерно для систем «издатель–подписчик» (publish&subscribe systems), где при формировании перечня адресатов выполняются запросы, содержащие многосоставные ограничения, представимые в виде булевых функций в ДНФ [17].

В существующей литературе описаны примеры реальных запросов к СУБД, содержащих большое число плеоназмов, оказывающих значительное влияние на размеры запроса. В частности, в одной из работ в качестве примера упомянут неискusstвенный запрос, в исходном виде включавший 98 условий, а в результате преобразования выражения в конъюнктивную нормальную форму и тривиальных упрощающих преобразований сократившийся до 12 [18].

Наконец, следует оговориться, что не все ограничения, поддающиеся минимизации с использованием представленных алгоритмов, могут быть определены пользователем как избыточные без проведения математического анализа. Это в особенности касается ограничений, представимых в виде конъюнктов линейных неравенств с использованием 2 и более атрибутов.

5. Влияние минимизации на общий ход оптимизации запроса

Разработка алгоритмов минимизации ограничений запросов, представимых в виде ДНФ, содержащих 2 и более конъюнктов, представляет и общетеоретический интерес. Это связано в первую очередь с задачами оптимизации серий запросов путем идентификации в них общих подвыражений (см., например, [19, 20]).

В контексте данной задачи группа запросов может рассматриваться как подзапросы, объединенные дизъюнкцией. В этом случае для запросов, имеющих общие подмножества элементов в части select и from, актуальна задача поиска общих групп условий, входящих в ограничение. В данном случае ограничения оптимизируемых запросов могут быть рассмотрены как единое ограничение, составные части которого объединены дизъюнкцией.

В множестве независимых запросов, отобранных по признаку наличия общих отношений, вероятно наличие повторяющихся и избыточных условий, устранение которых путем сокращения позволяет ускорить обработку. При решении этой задачи возникает необходимость идентификации общих подвыражений в составе ограничения, решению такой задачи может способствовать приведение запроса к единому наиболее лаконичному представлению с использованием всех доступных алгоритмов минимизации.

Без учета фактической оптимальности полученного минимального представления оно способствует решению двух задач:

- упрощение процедуры автоматической идентификации тождества двух выражений;
- сокращение числа возможных планов представления сложных запросов, что упрощает их оптимизацию.

Запрос к СУБД может быть представлен в виде *B*-дерева, где листья — отношения БД, а узловые элементы содержат операции и условия. При этом выполнение запроса характеризуется движением от листьев вверх по графу с целью последовательной реализации запроса. Задача оптимизации характеризуется выбором из множества существующих представлений запроса лучшего с точки зрения стоимости обработки представления. Учитывая значительное число возможных конечных представлений дерева, актуальной является задача сокращения числа представлений путем исключения однозначно неоптимальных.

В ряде работ в качестве параметра, определяющего число вариантов представления графов, рассматривалось число отношений без учета числа операций. Это определялось наличием «тривиального» правила оптимальности представления запроса, сформулированного, в частности, у Ионнидиса, согласно которому только один операнд отношения соединения может являться промежуточным результатом. Выполнение этого требования приводит к тому, что число вариантов представления дерева запроса определяется функцией $O(2^N)$, где N — число отношений [21].

Однако на текущий момент существуют работы, доказывающие, что реальные запросы могут иметь более сложную структуру, включающую множественные соединения одних и тех же отношений, что ведет к влиянию числа операций на число представлений графа. Кроме того, доказывается, что в случае, если запрос представим в виде гиперграфа, возникает задача его упрощения путем сокращения числа соединений, чтобы сократить затраты на поиск оптимального представления. Причем в ряде случаев без предварительного упрощения эта задача рассматривается как не поддающаяся решению [22, 23].

Таким образом, можно выделить следующие способы применения представленных алгоритмов:

- оптимизирующий эффект минимизации ограничения запроса;
- минимизация ограничения как одно из средств приведения ограничений к универсальному виду в целях определения их тождества;
- сокращение числа представлений запроса путем минимизации числа содержащихся в нем условий, что сокращает расходы на оптимизацию запроса.

Заключение

В данной статье проблема влияния минимизации запроса путем исключения из ограничения избыточных условий изучена на примере ранее разработанных алгоритмов лексической оптимизации.

Теоретически доказано, что минимизирующие преобразования ведут к сокращению времени выполнения запроса, и приведены теоретически обоснованные оценки экономии (формулы (3)–(5)).

Также приведены экспериментальные результаты сравнения стоимости обработки исходных и минимизированных запросов, содержащих избыточные условия. Доказано, что реализация алгоритмов поиска избыточности имеет стоимость, в 10^3 – 10^4 раз уступающую обработке запроса, а ликвидация избыточности может приводить к существенному сокращению времени обработки запроса.

Наконец, в статье рассмотрен контекст минимизации запросов с учетом практики работ в области оптимизации. Показана реальная встречаемость запросов, содержащих избыточности, отвечающие критериям представленных алгоритмов, а также показано косвенное влияние минимизирующих преобразований на процесс обработки запроса.

Список литературы / References

- [1] Hall P. A. V., “Optimization of single expressions in a relational data base system”, *IBM Journal of Research and Development*, **20**:3 (1976), 244–257.
- [2] Bellamkonda S. at al., “Enhanced Subquery Optimizations in Oracle”, *Proceedings of the 35th international conference on Very large data base*, August, 2009, **2**, 2009, 1368.
- [3] “Chapter 7. Optimization”, *MySQL 5.5 Reference Manual*.
<http://dev.mysql.com/doc/refman/5.5/en/optimization.html>.
- [4] “PostgreSQL 8.3.3”, postgresql-8.3.3/src/backend/optimizer/util/predtest.c.
- [5] “Query Optimization in Oracle Database 10g Release 2. P. 9”.
- [6] Seshadri P. at al., “Cost-Based Optimization for Magic: Algebra and Implementation”, *ACM SIGMOD Record*, **25**:2 (1996).
- [7] Faber W., Greco G., Leone N., “Sets and their application to data integration”, *Journal of Computer and System Sciences*, **73**:4 (2007).
- [8] Mendkovich N., Kuznetsov S., “New Algorithms for Lexical Query Optimization”, *Proceedings of the 31st International Conference on Information Technology Interfaces*, Cavtat/Dubrovnik, June 22–25, 2009 (Zagreb, University of Zagreb), 2009, 187–192.
- [9] Кузнецов С.Д., Мендкович Н.А., “Новые алгоритмы лексической оптимизации запросов”, *Моделирование и анализ информационных систем*, **16**:4 (2009), 22–33; [Kuznetsov S.D., Mendkovich N.A., “New algorithms for query modifications”, *Modeling and Analysis of Information Systems*, **16**:4 (2009), 22–33, (in Russian).]
- [10] Кузнецов С.Д., Мендкович Н.А., “Оптимизация конъюнктов условий в составе запросов”, *Модел. и анал. информ. систем*, **18**:3 (2011), 144–154; [Kuznetsov S.D., Mendkovich N.A., “Optimization of queries containing conjunctions of conditions”, *Modeling and Analysis of Information Systems*, **18**:3 (2011), 144–154, (in Russian).]
- [11] “BNF Grammar for ISO/IEC 9075-2:2003”, <http://savage.net.au/SQL/sql-2003-2.bnf.html>.
- [12] Khaitan P. at al., “Improved query plans for unnesting nested SQL queries”, *Proceedings of 2nd International Conference on Computer Science and its Applications*, December 10–12, South Korea, 2009 (Jeju Island, IEEE), 2009, 147–152.

- [13] Muralikrishna M., “Improved unnesting algorithms for join aggregate SQL queries”, *Proceedings of the 18th International Conference on Very Large Data Bases*, August 23–27, Vancouver, Canada, 1992 (San Francisco: Morgan Kaufmann Publishers Inc.), 1992, 91–102.
- [14] Satoh K. at al., “Local and Global Query Optimization Mechanisms”, *Proceedings of 11th International Conference on Very Large Data Bases*, August 21–23, 1985, Stockholm, Sweden (Berlin: Morgan Kaufmann), 1985, 408–409.
- [15] Hellerstein J. M., Stonebraker M., “Predicate Migration: Optimizing Queries with Expensive Predicates”, *ACM SIGMOD Record*, **22:2** (1993), 267–276.
- [16] Chaudhuri S., “Optimization of queries with user-defined predicates”, *Journal ACM Transactions on Database Systems (TODS)*, **24:2** (1999), 177–228.
- [17] Fontoura M. at al., “Efficiently Evaluating Complex Boolean Expressions”, *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (New York: ACM), 2010, 3–4.
- [18] Vorwerk K., Paulley G. N., “On Implicate Discovery and Query Optimization”, *Proceedings of the International Database Engineering and Applications Symposium, IDEAS 2002*. July 17–19, 2002, Edmonton, Canada (Los Alamitos: Computer Society), 2002, 2–12.
- [19] Roy P. at al., “Efficient and extensible algorithms for multi-query optimization”, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of data* (ACM New York, NY, USA), 2000, 249–260.
- [20] Dalvia N. N. at al., “Pipelining in multi-query optimization”, *Journal of Computer and System Science*, **66:4** (2003).
- [21] Ioannidis Y. E., “Query Optimization”, *ACM Computing Surveys (CSUR)*, **28:1** (1996), 121–123.
- [22] Neumann T., “Query Simplification: Graceful Degradation for Join-Order Optimization”, *SIGMOD’09*, June 29–July 2, 2009, Providence, Rhode Island, USA, 2009, 405–406.
- [23] Moerkotte G., Neumann T., “Dynamic programming strikes back”, *Proceedings of SIGMOD Conference 2008*, June 9–12, 2008, Vancouver, BC, Canada, 2009.

Mendkovich N., "On the Effectiveness of the Minimization Approach to the Query Optimization", *Modeling and Analysis of Information Systems*, **23:2** (2016), 153–163.

DOI: 10.18255/1818-1015-2016-2-153-163

Abstract. A standard problem of DBMSs usage is a lack of efficiency and high cost of the access to the stored data. The acceptable level of system performance may be achieved by query optimization technics that determine the most efficient way to execute a given query by its modification and considering possible query execution plans. The goal of this paper is to prove the efficiency of the query minimization algorithms based on minimization of the query restriction by elimination of the redundant conditions. The paper represents minimization algorithms based on the mathematical transformations, which detect and remove redundant conditions from query restriction to simplify it. It includes minimization algorithms based on “condition absorption”, prime implicants, and a set of linear inequalities minimization technics. The paper also includes theoretical justification of the efficiency of minimization approach to the query optimization based on restriction simplification. We also observe experimental results of the implementation of these optimization techniques and their influence on the query processing speed. In the end, we represent an observation of the query minimization impact on the whole optimization process

Keywords: query optimization, lexical optimization

On the authors: Mendkovich Nikita, orcid.org/0000-0003-1694-7947, FREENet Group Ltd., engineer, Leninsky Ave. 47, Moscow, 119991 Russia, e-mail: mend@f-group.ru