

# Обзор развития методов лексической оптимизации запросов

*Мендкович Н. А., Кузнецов С. Д.,  
mend@rambler.ru, kuzloc@ispras.ru*

**Аннотация.** Статья посвящена лексической оптимизации запросов и описывает работы, опубликованные в течение последних четырех десятилетий. Особое внимание уделяется таким подходам к оптимизации как модификация, украшение и сокращение запросов. Обсуждаются алгоритмы оптимизации для реляционных и нереляционных СУБД.

**Ключевые слова:** оптимизация запросов; упрощение запросов; лексическая оптимизация запросов; магические множества

## 1. Введение

За 40 лет, прошедших с момента выхода первой работы, в которой была сформулирована проблема оптимизации запросов к базам данных [1], вышло множество публикаций, посвященных этой тематике. Полный их обзор, актуальный на настоящий момент, отсутствует, и некоторое представление об их численности может дать анализ библиографий вышедших на настоящий момент обзорных работ.

Так, исследование М. Ярке (M. Jarke) и Ю. Коха (J. Koch) включает в себя перечень 253 работ [2], М. Маннино (M. V. Mannino), П. Чу (P. Chu) и Т. Сейджера (T. Sager) – 64 работы [3], Я. Ионнидиса (Y. E. Ionnidis) – 50 работ [4], С. Чаудхари (S. Chaudhari) – 61 работа [5], С. Д. Кузнецова – 128 работ [6]. Обзоры методов оптимизации запросов также содержатся в тематических главах работ Г. Грейфа (G. Graefe) [7, разделы 6, 11, 14], К. Дейта (C. Date) [8, стр. 639-690], М. Т. Оцсу (M. Ozsu) и П. Валдуреза (P. Valduriez) [9, pp. 228-273], Р. Рамакришнана (R. Ramakrishnan) и Дж. Герке (J. Gehrcke) [10, pp. 412-413]. Кроме этого, следует учесть краткую библиографию работ, посвященных оптимизации запросов, в которой содержится классификация этих работ по темам [11]. Множества рассмотренных исследований в перечисленных выше публикациях частично пересекаются, однако приведенные данные дают общее представление о публикациях, посвященных проблемам оптимизации запросов.

Наравне с академическими исследованиями, посвященными оптимизации запросов, параллельно выходят публикации компаний, посвященные решению

этой задачи в рамках выпускаемых ими СУБД (например, MySQL [12], [13, глава 19], Oracle [14-15, 92-93], DB2 [16], PostgreSQL [17]). Подробный обзор оптимизаторов СУБД различных производителей был представлен в [9], но позже подобные исследования не предпринимались, во многом по причине неготовности компаний подробно описать собственные разработки в области оптимизации запросов. Исключениями могли бы служить СУБД с открытым кодом, в частности, PostgreSQL, но публикации с описанием использованных в них алгоритмов нам неизвестны.

Как показывает представленная выше статистика публикаций в области оптимизации запросов, их полный обзор является нетривиальной задачей, и с высокой вероятностью в итоге такой обзор может быть неполным. Поэтому более разумным представляется сосредоточиться на одном из классов оптимизационных алгоритмов. Ранее публиковались подобные обзоры, посвященные применению гистограмм, в том числе, для целей оптимизации запросов [18], применению материализованных представлений в процессе обработки запросов [19], решению выбора порядка соединений [20].

Данная работа посвящена методам так называемой лексической оптимизации запросов, но чтобы определить, что собой представляют эти методы, необходимо представить краткий очерк основных этапов оптимизации и признаков, позволяющих отнести конкретный метод к одному из этих этапов.

Вслед за Я. Ионнидисом мы определим оптимизацию запросов как «изучение алгебраических выражений, эквивалентных заданному запросу, и выбор из них одного, оцениваемого как наиболее дешевое» [4, p. 1041].

На основе классификации, принятой в [2], в рамках задачи оптимизации запроса выделим четыре основных этапа, или класса оптимизирующих операций:

- i преобразование запроса в стандартную внутреннюю форму;
- ii модификация запроса, т.е. приведение его к форме, при которой выполнение наиболее эффективно;
- iii анализ запроса, выбор потенциальных низкоуровневых процедур, используемых при его выполнении;
- iv генерация и сравнительный анализ различных планов доступа – возможных способов выполнения запроса.

На практике четвертый класс операций воспринимается как основа оптимизационных алгоритмов; первые три класса используются для подготовки генерации планов доступа, в то время как на четвертом этапе оптимизатор использует эти планы, оценивает и выбирает лучший из них для выполнения запроса. Именно четвертый класс операций сегодня привлекает наибольшее внимание исследователей, и его описанию уделяется наибольшее внимание в корпоративных обзорах оптимизаторов выпускаемых СУБД. Так, например, оптимизатор СУБД компании Oracle, судя по доступным нам описаниям [21], ориентирован в первую очередь на сравнительный

«оценочный» анализ существующих путей доступа, и во многих обзорах, посвященных оптимизатору данной СУБД, методы лексической оптимизации не упоминаются вовсе [например, 22, с. 417-445]. В оптимизаторах Starburst и DB2 основное внимание уделяется семантической оптимизации запросов и совершенствованию системы выбора планов выполнения запросов [16].

Однако, несмотря на подчиненную роль, которую играют модификация и первичные преобразования запросов, рассмотрение наиболее эффективных подходов к решению данной задачи представляется вполне актуальным, так как оптимизатор по-прежнему нуждается в генерации достаточного числа субоптимальных, в рамках того или иного критерия, планов доступа.

При анализе подобных подходов к оптимизации запросов можно выделить два вида модификаций:

- *лексическая оптимизация*<sup>1</sup> запроса, при которой единственным источником информации является сам текст запроса как лексическая конструкция, и иные сведения о базе данных и ее структуре в анализе не используются. Процесс лексической оптимизации включает в себя анализ ограничения запроса, сравнение и анализ содержащихся в нем условий с целью выявления избыточности;
- *семантическая оптимизация*, включающая в себя использование ограничений целостности БД и сами хранимые данные с целью генерации оптимальных планов доступа к данным (обзор методов семантической оптимизации представлен в [2, глава 3], а ряд алгоритмов рассматривается в некоторых более поздних работах [23-24]).

Методы лексической оптимизации представляются особо интересными в связи с тем, что, во-первых, их можно легче адаптировать для нужд оптимизации запросов в базах данных с нереляционной структурой, во-вторых, в ряде существующих разработок предусматривается вынесение некоторых оптимизирующих операций из сервера базы данных в удаленное клиентское приложение [25]. Кроме того, поскольку наибольший интерес к исследованиям в данной области проявлялся в 1990-е и даже 1980-е гг., многие современные исследователи могут быть недостаточно знакомы с более ранними работами, алгоритмы и методы, представленные в которых, могут быть вновь востребованы в настоящее время.

---

<sup>1</sup> В зарубежной литературе для обозначения подобных преобразований часто используется термин «перезапись» (rewriting) [4, р. 1044]. Однако нам представляется, что этот термин в недостаточной степени выражает суть данного вида оптимизации запросов. Кроме того, в ряде работ он используется по отношению к преобразованиям запросов с использованием материализованных представлений запросов [19], т.е. выходит за рамки используемого определения.

С точки зрения логики, лексическая оптимизация может устранять три различных вида неоптимальности запроса тремя различными способами.

- Сокращение запроса – удаление избыточных условий, например, дублирующих одно другое или тождественно истинных, обработка которых является излишней с точки зрения выполнения запроса, а также условий, делающих запрос невыполнимым, противоречащих друг другу или тождественно ложных. При обнаружении таковых условий все ограничение или его часть признается ложной и исключается из выполнения запроса.
- Усовершенствование (украшение) – усложнение структуры запроса, которое, однако, ведет к оптимизации его выполнения.
- Преобразование запроса – изменение запроса, не относящееся ни к сокращениям, ни к усовершенствованиям запроса, и играющее роль, служебную по отношению к иным формам оптимизации запроса, например, стандартизация формы представления запроса.

Далее в статье приводятся классификация алгоритмов лексической оптимизации, обзор работ в данной области, а также отмечаются возможные направления дальнейших исследований.

Мы стремились свести к минимуму возможные повторы сведений, изложенных в предыдущих обзорных работах, поэтому в ряде случаев отказываемся от подробного анализа тех или иных вопросов, отсылая читателей к более ранним публикациям. Допускается краткое повторение некоторых сведений в ряде случаев с целью показать развитие исследований по совершенствованию методов оптимизации запросов.

## 2. Преобразования запросов

Критерием классификации алгоритма как относящегося к числу алгоритмов простого преобразования является его функциональное предназначение. Если итогом работы некоторого алгоритма должно стать обретение запросом более оптимальной формы с точки зрения того или иного критерия (например, числа условий в ограничении), то алгоритм может быть отнесен к числу алгоритмов сокращения или усовершенствования. Если же алгоритм, как упоминалось выше, играет подчиненную роль, преобразуя запрос к иному виду, в частности, меняя порядок операций, то речь идет именно о простом преобразовании запроса.

Прежде всего, к технологиям преобразования нужно отнести алгоритмы стандартизации запросов, приведения их к универсальной стандартной форме в целях дальнейшей обработки. Такие преобразования требуются не только для нужд дальнейшего улучшения или упрощения запроса, но также для сопоставления обрабатываемых запросов с ограничениями целостности базы данных и сохраняемыми материализованными представлениями запросов.

Возможные подходы к представлению запросов к реляционным базам данных в виде графов, таблиц, реляционных выражений и т.п. рассмотрены в [2, раздел 3.1], каковой источник, несмотря на давность, вполне актуален.

У представлений запросов в нереляционных СУБД имеется своя специфика. Основы представления запросов на языке XQuery уже описаны в рекомендациях W3C [26], однако ряд особенностей внутреннего представления запросов заслуживает отдельного рассмотрения. В интересах оптимизации запросов, формулируемых на новых языках, требуется создание новых алгебр с учетом особенностей моделей данных, лексик, синтаксиса и семантики этих языков.

Можно выделить два подхода к решению этой задачи: реляционно-центричный и алгебраический. Первый подход основывается на преобразовании запросов на нереляционных языках, а при необходимости и самих данных к реляционному виду и их представлению в терминах реляционных СУБД [27-30], чтобы иметь возможность полностью использовать для оптимизации новых видов запросов арсенал накопленных ранее средств. Обзор подобных работ представлен в [31].

Во втором подходе предлагается не приближать новые языки и модели данных к старым алгоритмам, а создавать новые алгебры для представления запросов с учетом особенностей новых языков. Проблема представления запросов к нереляционным базам данных решается с помощью расширения реляционной алгебры для соответствующих языков запросов: OQL [32-33] и XQuery [34-37]. Проблема алгебраического представления запросов к базам XML-данных также рассматривается в [38-39].

Примером преобразования запроса может также служить перемещение логических условий вниз в дереве запроса. Целью ряда улучшающих преобразований является минимизация размера конструируемых, сохраняемых и считываемых промежуточных результатов. Важная эвристика перемещает селективные операции, такие как ограничение и проекция, ниже конструктивных операций, таких как соединение и декартово произведение, чтобы выполнять селективные операции как можно раньше [2, 40]. В пользу как можно более раннего исполнения селективных операций высказывались и другие авторы [41]. Данный подход считается сейчас если и не оптимизирующим запрос, то, по крайней мере, не повышающим стоимость его обработки, и он автоматически применяется ко всем запросам. Также имеются работы, посвященные перемещению по дереву запроса операций сортировки [42] и предикатов (в тех случаях, когда запросы не имеют иерархической структуры) [43].

В ряде работ анализируются способы упорядочения операций проекции и соединения с целью упрощения обработки запросов (например, [44-47]). Однако наряду с некоторыми алгоритмами, которые можно отнести к лексическим, в этих подходах применяются методы, которые основываются на использовании описаний данных, содержащихся в базе, и поэтому мы не

имеем достаточных оснований останавливаться на таких подходах к оптимизации в настоящем обзоре. Более подробно существующие варианты решения задачи определения порядка соединений анализируются в [2, раздел 4], [5, раздел 4.1] и [20].

Наравне с подходом к оптимизации, основанным на упорядочении операций, имеется ряд работ, ориентированных на сокращение их числа; эти работы подробно рассматриваются в разделе 4 данной статьи.

Еще одну группу алгоритмов преобразования составляют алгоритмы устранения вложенных подзапросов, что для стандартных SQL-запросов означает исключение из блоков SELECT-FROM-WHERE вложенных в них аналогичных блоков путем их удаления или слияния с телом основного запроса. Приведем пример запроса о поиске всех поставщиков детали P2 из [8, с. 683]:

```
SELECT S.Name
FROM S
WHERE S.S# IN (SEL query optimization; query simplification; lexical query
optimization; magic sets
ECT SP.S#
FROM SP
WHERE SP.P# = 'P2');
```

Его семантически эквивалентной, но более лаконичной и быстрой в обработке формулировкой является следующий запрос:

```
SELECT S.Name
FROM S, SP
WHERE S.S# =SP.S#
AND SP.P# = 'P2';
```

Впервые решение проблемы вложенных запросов было предложено в работе [48]. Автор выявил 5 типов вложенных запросов и описал алгоритмы устранения этих вложений. Позже работа [48] подвергалась критике с указанием на то, что в ряде случаев предложенные алгоритмы работают некорректно [49-51]. В этих работах предлагались свои варианты алгоритмов, в которых исправлялись недочеты, допущенные в [48], и обеспечивалась большая эффективность. Альтернативное решение задачи было недавно предложено в [52].

Поскольку проблема вложенных подзапросов существует и в нереляционных СУБД, в нескольких работах представляются алгоритмы исключения вложений из запросов, сформулированных на OQL [32-33] и XQuery [34-35, 53-54]. Важно отметить, что изучение этой проблемы исторически совпало с разработкой новых нереляционных алгебраических представлений запросов для этих языков.

Более подробно работы по проблеме вложенных подзапросов рассмотрены в [8, с. 682-684], [5, раздел 4.2.2] и [35, раздел 2]. Примеры применения алгоритмов устранения вложенности в различных системах описаны в [55-56].

### 3. Лексическая оптимизация путем улучшения

Оптимизация запроса путем улучшения основывается на усложнении структуры запроса, включении в него новых табличных выражений, использование которых позволяет сократить расходы на обработку исходного запроса. Большинство алгоритмов, решающих эту задачу, относятся к технологии «магических множеств».

*Алгоритм магических множеств* относится к числу алгоритмов лексической оптимизации запроса. Он означает перезапись запроса, исключающую генерацию нерелевантных кортежей путем вычисления дополнительных таблиц, которые содержат связывания, используемые для ограничения таблицы, и действуют как фильтры.

Перезапись исходного запроса с использованием этих фильтров включает следующие основные этапы:

- анализ связей в рамках реализации запроса, их отражение путем аннотирования предикатов;
- создание на основе полученных описаний дополнительных (магических) таблиц;
- интеграцию магических таблиц в уже существующее ограничение и изменение описания существующих условий.

Изначально алгоритм магических множеств был создан для нужд дедуктивных баз данных, поддерживающих язык запросов «Datalog»[57], где запрос на выборку данных описывается с помощью системы правил (rules) и фактов (facts).

Алгоритм магических множеств предлагает создать дополнительные правила и факты, которые усложняют запись правил системы, но позволят исключить из обработки индивидов, заведомо нерелевантных запросу. Для этого в систему правил включается отношение  $magic(x)$ , обозначающее релевантность объекта по отношению к обрабатываемому запросу.

Предложенная перезапись увеличивает число правил и удлиняет их запись, однако очевидно, что скорость обработки запроса при этом возрастает, так как добавленные магические отношения исключают из анализа нерелевантные объекты и ускоряют работу.

Для создания магического варианта запроса все предикаты правил снабжаются «украшениями», своеобразными аннотациями, показывающими, какие аргументы связываются с константами или ограничиваются правилами, а какие являются свободными. «Украшение» для некоторого  $n$ -арного предиката, где  $n$  – число элементов, – это строка из символов  $b$  (bonded) и  $f$  (free). В [58] была введена третья литера украшения  $c$  (condition), используемая только для независимых условий. Условие на атрибуте  $X$  называется независимым, если оно может быть выражено без ссылки на

какой-либо свободный атрибут. Например, условие  $X > 10$  является независимым. Условие  $X > Y$  независимо, если атрибут  $Y$  является связанным, в противном случае это условие зависимо.

Из «украшенных» предикатов создается граф правил и целей, представляющий структуру правил и фактов и их взаимодействие между собой. На графе должны быть представлены связи между предикатами и правилами, с помощью пунктирных линий – маршрут передачи данных при выполнении запроса. На основе анализа этого графа создается стратегия SIPS (Sideways Information Passing Strategy), определяющая, как передавать информацию сторонним образом в теле табличного выражения при вычислении табличного выражения.

Позже авторы предложили усовершенствованную версию алгоритма «магических множеств» для реляционных баз данных. Формальное определение алгоритма оставалось прежним, исключая замену терминов: «предикат» на «таблицу», а «магическое правило» - на «магическое выражение».

На основе полученных данных производится перезапись запроса, включающая в себя следующие элементы [59]: созданию запроса и пустого множества таблиц, создание для каждой таблицы из начального запроса новой магической версии и модифицированного табличного выражения, создание начальной таблицы из предикатов сравнения по равенству в наиболее внешнем блоке запроса. После этой перезаписи запрос сокращается путем объединения табличных выражений с одним и тем же заголовком в одно выражение, в теле которого содержится объединение соответствующих тел.

Представим работу алгоритма на следующем примере. Исходный запрос, выбирающий фамилии старших программистов, которые получают зарплату, большую, чем средняя зарплата их отдела, выглядит следующим образом:

```
SELECT Ename FROM emp e1
WHERE Job = "Sr Programmer" AND
Sal > (SELECT AVG(e2.Sal)
      FROM emp e2
      WHERE e2.Dno =e1.Dno)
```

Исходный запрос обладает набором недостатков, снижающих эффективность его обработки: средняя зарплата отдела может вычисляться несколько раз, доступ к  $e1$  и  $e2$  должен производиться в строгом порядке, обработка  $e2$  является покортежной, а не ориентированной для множества, что делает недоступными в данном случае преимущества реляционной модели, ориентированной на обработку множеств.

Преобразованный с помощью алгоритма магических множеств запрос выглядит следующим образом:

```

SELECT Ename FROM s_mag, mag_avgsal
WHERE Sal > Asal AND s_mag.Dno = mag_avgsal.Dno
mag_avgsal(Dno, Asal) AS
(SELECT Dno, AVG(Sal)
FROM mag, emp
WHERE mag.Dno = emp.Dno GROUPBY Dno)
mag(Dno) AS
(SELECT DISTINCT Dno
FROM s_mag)
s_mag(Ename, Dno, Sal) AS
(SELECT Ename, Dno, Sal FROM emp
WHERE Job = "Sr Programmer")

```

Данная формулировка запроса позволяет избежать повторного обращения к одним и тем же данным и максимально использовать выгоды реляционной модели. Подзапрос s\_mag выбирает служащих, являющихся старшими программистами, mag определяет, в каких отделах имеется хотя бы один из таких служащих, mag\_avgsal вычисляет среднюю зарплату только для этих отделов. Несмотря на то, что данный алгоритм требует вычисления дополнительных таблиц, очевидно, что его выполнение более эффективно по сравнению с изначально представленным запросом. На примере, представленном в [59], показано, что преобразование методом декорреляции не дает аналогичного эффекта.

Наравне с преобразованием «магических множеств» для работы в рамках реляционной СУБД в алгоритм внесен ряд усовершенствований: в частности, после введения украшения с стала возможной обработка условий, не связанных с равенством, предлагается способ работы алгоритма для нерекурсивных запросов. Аналогичная проблема для работы магических множеств в среде Datalog рассматривалась в [60].

Значительной проблемой оставалась интеграция оптимизации методом «магических множеств» с оценочным оптимизатором, т.е. разработка алгоритмов выбора этого оптимизационного алгоритма из числа иных при обработке различных запросов оптимизатором. Тем более, что результат работы алгоритма очень сильно зависит от начального представления запроса: если перед «магической» оптимизацией он уже подвергся перезаписи, его преобразованный вид будет отличаться от вида, полученного при обработке его начального представления.

Решение этой проблемы было представлено в [61]. Авторы предложили метод включения алгоритма «магических множеств» в схему оценочного оптимизатора в качестве одного из методов соединений, схожего с операцией полусоединения. Авторы отмечают: «Оценочная оптимизация предотвращает использование перезаписи на основе магических множеств, когда этого делать не следует, и позволяет выбрать один из лучших вариантов, когда перезапись на основе магических множеств целесообразно применять» [61]. В статье

также предлагается расширение алгебры запросов путем введения мультимножественной алгебраической операции  $\Theta$ -полусоединения, которое позволяло бы моделировать магическую перезапись запросов.

Позже возобновился интерес к применению магических множеств для оптимизации запросов в среде Datalog, в частности – возможной избыточности магических фактов и правил. В ряде работ были предложены способы усовершенствования алгоритма магических множеств и проверки итогового представления запроса с целью устранения избыточностей [62-64].

Судя по доступным нам публикациям, дальнейшие попытки усовершенствовать метод «магических множеств» для реляционных баз данных не предпринимались. Более поздние работы касаются применения магических множеств для задач, отличных от оптимизации запросов, в том числе, интеграции данных [65], применения к запросам на языках, отличных от «классических» Datalog и SQL [66, 67] и т.п.. Отдельно следовало бы выделить работы о применении «магических множеств» для запросов к XML-данным [68, 69].

#### **4. Лексическая оптимизация путем сокращения**

Сокращение запроса - это повышение лаконичности и уменьшение числа условий, которое может производиться только при сохранении семантики запроса. Конечно, как показывает ряд примеров (см. предыдущий раздел), оно не может являться самоцелью, однако удаление семантически избыточных условий, дублирующих информацию, уже содержащуюся в запросе, будет способствовать ускорению его обработки.

Каждое условие в составе ограничения требует определенных системных ресурсов на свою реализацию. Следовательно, любые семантические повторы в составе ограничения приводят к росту издержек на обработку запроса. Аналогичным образом, выполнение тождественно ложных или тождественно истинных условий повышает расходы на исполнении запроса. Устранение подобных проявлений семантической избыточности запроса на стадии до его выполнения оптимизирует запрос.

Подобное преобразование условия ставит задачу распознавания эквивалентности запросов и их ограничений. Мы не будем подробно останавливаться на теоретических работах в этой области, и отметим лишь, что они актуальны и сегодня (см., например, [70-71]). Интерес к этой области также возрос в связи с тем, что было доказано сходство данной задачи с проблемой поиска допустимого решения в области искусственного интеллекта и теории операций [72-73].

Распознавание эквивалентности актуально не только при лексической оптимизации запроса, но и при использовании материализованных представлений, которые позволяют оптимизировать запрос путем поиска общих подвыражений, с запросами, ранее обработанными системой, чьи

представления хранятся в памяти и могут быть использованы при реализации нового запроса. Однако последний вид оптимизации находится вне темы нашего обзора, и мы ограничимся ссылкой на работы, посвященные непосредственно ему [см., например, 19, 74-75], а также краткий обзор [5, раздел 7.3].

Исследованиям в области упрощающей лексической оптимизации запроса способствовала, в том числе, работа [76], где было доказано, что любое конъюнктивное ограничение запроса вида

$$X_1 \text{ AND } X_2 \dots \text{ AND } X_n$$

имеет минимальное эквивалентное представление, которое может быть определено с помощью последовательности операций. Эта задача является NP-сложной. Тем не менее, любое конъюнктивное ограничение может быть преобразовано к более краткой форме путем исключения избыточных условий.

Теоретическую основу упрощающей лексической оптимизации, алгоритмам которой посвящена настоящая часть статьи, по всей видимости, составила работа П. А. В. Холла (P. A. V. Hall) [41], 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».

Ряд общих соображений по использованию реляционной алгебры при оптимизации запросов рассматривается также в [77]. При решении задач оптимизации запросов в нереляционных базах данных ряд авторов также применяли подход, представленный, в том числе, в [77]. В его рамках сокращаются избыточные условия, объединенные логическим конъюнктом с представлением ограничений в виде табло, т.е. табличных нотаций [78, 79]. Позже этот алгоритм был усовершенствован для ограничений, содержащих операции дизъюнкции и отрицания [80].

Следующий шаг в решении задачи сокращения запроса сделан в [43, раздел 4.5], где процедура перемещения предиката по дереву запроса используется среди прочего для сопоставления его с предикатами в других узлах на разных уровнях вложенности с целью исключения избыточности. Фактически речь идет о способах сопоставления ограничений запросов, основанных на [73].

Обзор некоторых родственных методов оптимизации запроса, основанных на передаче предикатов между блоками, представлен в [5, раздел 4.3].

Судя по опубликованным данным о работе оптимизаторов широко распространенных СУБД, с целью удаления избыточных условий в них реализуются только алгоритмы поиска общих подвыражений в конъюнктах условий. В частности это касается MySQL [81], PostgreSQL [82] и Oracle [21, р. 9]. Причем в случае PostgreSQL анализ открытого кода позволяет установить, что поиск общих подвыражений реализовывался без использования представления табло на основе сформулированных для различных пар условий правил. Несмотря на то, что этот алгоритм, в сущности, достаточно тривиален, долгое время не было сообщений о попытках его усовершенствования.

Отказ от обработки дизъюнктов ограничений можно рассмотреть на примере работы оптимизатора из описания MySQL 5.5 [12], где при рассмотрении приемов лексической оптимизации приводится преобразование:

$$((a \text{ AND } b) \text{ AND } c \text{ OR } (((a \text{ AND } b) \text{ AND } (c \text{ AND } d)))) = > \\ (a \text{ AND } b \text{ AND } c) \text{ OR } (a \text{ AND } b \text{ AND } c \text{ AND } d).$$

Однако очевидно, что данное преобразование не завершено и итоговое кратчайшее представление этой функции

$$a \text{ AND } b \text{ AND } c,$$

так как

$$X \text{ OR } (X \text{ AND } a) = > X.$$

Применимые для решения этой задачи способы минимизации логических функций были разработаны еще в начале 1950-х гг. [83, 84] (т.е. еще до начала специальных исследований в области оптимизации запросов [1]), позже они были применены для программной реализации [85-86]. Возможность использования алгоритма Квайна-Маккласки (Quin-McCluskey) [85-86] при оптимизации запросов допускалась в [87, р. 234], [88], [89, с. 171-172.].

Авторы данного обзора в своих недавних работах [90-92] предложили ряд алгоритмов, позволяющих оптимизировать запрос, используя алгоритмы алгебры логики и линейной алгебры, в том числе упомянутый выше алгоритм Квайна. При этом ограничения запросов рассматривались бы как логические функции или системы линейных неравенств.

Нами были предложены:

- алгоритм поглощения условий-дубликатов и их конъюнктов, основанный на распознавании дубликатов не только в рамках одного конъюнкта, как в ранее упомянутых работах, но и разных конъюнктах одного ограничения, представленного в дизъюнктивной нормальной форме;
- алгоритм минимизации запроса на основе дополненного алгоритма

Квайна (дополнения Маккласки нереализуемы для данного вида задач), при котором ограничение в ДНФ рассматривается как множество утверждений;

- алгоритм минимизации ограничения запроса как системы линейных неравенств.

Кроме алгоритмов, удаляющих избыточные условия, используется также методы упрощения запроса путем объединения и сокращения не отдельных предикатов и их групп, а вложенных подзапросов. Например, в системе Oracle [93, раздел 2] 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» [93, p. 1368]).

Важным направлением в области оптимизации запросов путем сокращения является адаптация существующих и разработок новых алгоритмов для обработки запросов к нереляционным СУБД, в особенности, на языке XQuery, предназначенном для взаимодействия с XML-данными.

Однако в существующих работах к оптимизации XQuery-запросов в значительной мере применяются уже хорошо известные приемы. Работы [34, 35] содержат преобразования для алгебры XQuery, тяготеющие, однако, к приемам из [41]. Авторы [54] особо подчеркивают важность развития алгебраического представления запросов с целью применения для оптимизации XQuery-запросов приемов, доказавших свою эффективность для реляционных баз данных. В [30] даже описываются оптимизирующие преобразования над XQuery-запросом, преобразованным в реляционную форму.

## 5. Заключение

Несмотря на большое число разработанных методов оптимизации запросов, эта задача все еще остается актуальной для существующих СУБД. Как отмечает известный специалист в данной области С. Чаудхари: «Оптимизация запросов столь же, если не более, актуальна, чем в прошлом. Современные база данных включают в себя системы обработки транзакций в реальном времени, ERP-системы, системы управления взаимодействием с клиентами, аналитическую обработку в реальном времени, анализ данных с использованием хранилищ данных (Data-warehouses). Запросы, генерируемые этими приложениями, все сложнее, а базы данных – больше, чем когда бы то ни было. Таким образом, центральная роль оптимизации запросов, которая ищет различные стратегии обработки и выбирает лучший план исполнения, остается несомненной» [94, p. 961]. В том числе, определенного внимания заслуживают алгоритмы лексической оптимизации запросов.

В данном обзоре мы постарались рассмотреть вышедшие за прошедшие годы работы в области лексического подхода к оптимизации запросов и выделить

основные предметы исследований в этой области. Мы предлагаем следующую классификацию.

Перезапись запросов:

- Создание алгебраических представлений для управления запросами
- Перестановка и упорядочение элементов запроса с целью повышения скорости обработки
- Интеграция вложенных подзапросов

Улучшение подзапроса:

- Магические множества

Сокращение подзапросов:

- Удаление избыточных условий и групп условий в рамках одного ограничения
- Сопоставление и сокращений ограничений подзапросов на различных уровнях вложенности

Кроме того, общим для всех указанных направлений является адаптация существующих в них разработок к нереляционным базам данных, в том числе, к обработке запросов на XQuery, предназначенном для обращения к XML документам. Ряд существующих алгоритмов легко применимы к новым запросам, другие нуждаются в существенной доработке для новых задач.

Как показывает пример [90], более внимательного изучения заслуживают существующие математические алгоритмы, применимые для решения различных задач, связанных с оптимизацией запросов.

## Список литературы

- [1] Palermo F. A data base search problem // Proceedings of the 4th Symposium on Computer and Information Sciences, Virginia, USA, 1972. Reston: AFIPS Press, 1972. Pp. 67-101.
- [2] Jarke M., Koch J. Query Optimization in Database Systems // ACM Computing Surveys (CSUR), 1984. March, Volume 16, Issue 2. Pp. 111-152.
- [3] Mannino M. V., Chu P., Sager T. Statistical profile estimation in database systems // ACM Computing Surveys, 1988. September, Volume 20, Issue 3. Pp. 191-221.
- [4] Ionnidis Y. E. Query Optimization // The Computer Science and Engineering Handbook. Boca Raton: CRC Press, 1996. Pp. 1038-1054.
- [5] Chaudhari S. An Overview of Query Optimization in Relational Systems // Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. New York: SIGMOD, 1998. Pp. 34-43.
- [6] Кузнецов С. Д. Методы оптимизации выполнения запросов в реляционных СУБД // [http://www.citforum.ru/database/articles/art\\_26.shtml](http://www.citforum.ru/database/articles/art_26.shtml) [Обращение 20 сентября 2012].

- [7] Graefe G. Query Evaluation Techniques for Large Databases // ACM Computing Surveys, 1993. Volume 25, Issue 2. P. 73-169.
- [8] Дейт К. Дж. Введение в системы баз данных. Москва – Санкт-Петербург – Киев: Издательский дом «Вильямс», 2001. 1072 с.
- [9] Ozsu M. N., Valduriez P. Principles of Distributed Database Systems. Second Edition. New Jersey: Prentice Hall International, 1999. 666 pp.
- [10] Ramakrishnan R., Gehrcke J. Database System Management. 2nd Edition. Singapore: The McGraw-Hill Book Co, 2000. 906 pp.
- [11] Karayannidis N. Query Optimization Bibliography, [http://www.dbnet.ece.ntua.gr/~nikos/edith/qopt\\_bibl/](http://www.dbnet.ece.ntua.gr/~nikos/edith/qopt_bibl/), [Обращение 20 сентября 2012].
- [12] MySQL 5.5 Reference Manual. Chapter 7. Optimization. <http://dev.mysql.com/doc/refman/5.5/en/optimization.html>, [Опубликовано в 2010 году, обращение 20 сентября 2012].
- [13] Веллин Л., Томсон Дж. MySQL. Учебное пособие. Перевод с английского. М.: Издательский дом «Вильямс», 2005. 292 с.
- [14] Upgrading from Oracle Database 10 g to 11 g: What to expect from the Optimizer. An Oracle White Paper, November 2009. Redwood Shores: Oracle Corporation, 2009. 36 pp.
- [15] Comparison of Materialized Views & Analytic Workspaces in Oracle Database 11 g. An Oracle White Paper, March 2008. Redwood Shores: Oracle Corporation, 2008. 27 pp.
- [16] Markl V., Lohman G. M., Raman V. LEO: An autonomic query optimizer for DB2 // IBM System Journal, 2003. V. 42, № 1. Pp. 98-106.
- [17] PostgreSQL 8.1.19 Documentation. Chapter VII: Internals, <http://www.postgresql.org/docs/8.1/interactive/internals.html>, [Обращение 20 сентября 2012].
- [18] Ioannidis Y. The History of Histograms // Proceedings of 29th International Conference on Very Large Data Bases, September 9-12, 2003, Berlin, Germany. Berlin: Morgan Kaufmann, 2003. Pp. 19-30.
- [19] Halevy Y. Answering queries using views: A survey // The International Journal on Very Large Data Bases, December 2001. Volume 10 Issue 4. Pp. 270-294.
- [20] Braga D., Ceri S., Grossniklaus M. Join Methods and Query Optimization // Lecture Notes on Computer Science, 2010. Issue 5950. Pp. 188-210.
- [21] Query Optimization in Oracle Database 10g Release 2. An Oracle White Paper, June 2005. Redwood Shores: Oracle Corporation, 2005. 31 pp.
- [22] Смирнов С. Н., Задворьев И. С. Работаем с Oracle: Учебное пособие. М: Гелиос АРВ, 2002. 496 сс.
- [23] Shenoy S. T., Ozsoyoglu Z. M. A System for Semantic Query Optimization // SIGMOD Record, 1987. Volume 16, Number 3. Pp. 181-195
- [24] Geng K., Dobbie G., Meng Y. Survey of XML Semantic Query Optimization // Proceedings 4th International Conference on Internet Computing for Science and Engineering (ICICSE), Harbin, 2009. Washington D. C.: IEEE Computer Society, 2009. Pp. 297-300
- [25] Зверев Д. Л. Оптимизация потоков простых SQL-запросов: Диссертация кандидата технических наук: 05.13.11. Санкт-Петербург: Санкт-Петербургский Государственный Университет Аэрокосмического Приборостроения, 2005. 169 сс.
- [26] XQuery 1.0 and XPath 2.0 Formal Semantics. World Wide Web Consortium (W3C), W3C Recommendation, 2007. <http://www.w3.org/TR/xquery-semantics/>, [Обращение 20 сентября 2012].
- [27] Deutsch A., V. Tannen XML queries and constraints, containment and reformulation // Theoretical Computer Science, 2005. № 336. Pp. 57-87.
- [28] Fan W., Xu J. Y., Ding B., Qin L., Rastogi R. Query Translation from XPath to SQL in the Presence of Recursive DTDs // Very Large Data Bases Journal, 2009. Issue 18. Pp. 857-883.
- [29] Барашев Д. В., Горшкова Е. А., Новиков Б. А. Оптимизация представления XML документов в реляционной базе данных // Вторая Всероссийская научная конференция. Электронные библиотеки. Перспективные методы и технологии, электронные коллекции. 26-28 сентября, Протвино, 2000. С. 224-229.
- [30] Manolescu I., Florescu D., Kossmann D. Answering xml queries on heterogeneous data sources // Proceedings of the 27th International Conference on Very Large Data Bases, San Francisco, 2001. San Francisco: Morgan Kaufmann Publishers Inc., 2001. Pp.241-250.
- [31] Krishnamurthy R., Kaushik R., Naughton J. XML-SQL query translation literature: The state of the art and open problems // XML Database Symposium (XSym 2003) at VLDB 2003. Berlin, September 2003. Lecture Notes in Computer Science, Volume 2824, 2003. Pp. 1-18.
- [32] Cluet S., Moerkotte G. Nested queries in object bases // Proceedings of the 4th International Workshop on Database Programming Languages: Object Models and Language. Manhattan, New York City, USA, 30 August-1 September, 1993. London: Springer-Verlag, 2004. Pp. 226-242.
- [33] Steenhagen H. J., Apers P. M. G., Blanken H.M., de By R. A. From nested-loop to join queries in OODB // Proceedings of the 20th International Conference on Very Large Data Bases, 1994. Santiago: Morgan Kaufman, 1994 P. 618-629.
- [34] Frasinca F., Houben G.-J., Pau C. XAL: An algebra for XML query optimization // Proceedings of the 13th Australasian Database Conference, Australian Computer Society, Inc. Darlinghurst, Australia, 2002. Volume 5. Melbourne: Australian Computer Society, 2002. Pp. 49-56.
- [35] May N., Helmer S., Moerkotte G. Strategies for Query Unnesting in XML Databases // ACM Transactions on Database Systems (TODS) 2006. Volume 31, Issue 3. Pp. 968-1013.
- [36] Re C., Simeon J., Fernandez M. F. A Complete and Efficient Algebraic Compiler for XQuery // Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April, 2006, Atlanta, GA, USA: IEEE Computer Society, 2006. Lecture Notes on Computer Society, 2007. Volume 4797. Pp. 81-96.
- [37] Ghelli G., Onose N., Rose K., Siméon J. XML Query Optimization in the Presence of Side Effects // Proceedings of the 2008 ACM SIGMOD international conference on Management of data. New York, USA, 2008. New York: ACM, 2008. Pp. 339-352.
- [38] Lukichev M., Barashev D. XML Query Algebra for Cost-based Optimization // SYRCODIS\*07 The Fourth Spring Young Researchers Colloquium on Databases and Information Systems, Moscow, May 31 - June 1, 2007. <http://ceur-ws.org/Vol-256/>, [Обращение 20 сентября 2012].
- [39] Лукичев М. С. Оптимизация запросов в слабоструктурированной модели данных. Диссертация кандидата физико-математических наук: 05.13.11. Санкт-Петербург: Санкт-Петербургский Государственный Университет, 2009. 120 с.
- [40] Smith M., Chang P. Y. W. Optimizing the performance of a relational algebra database interface // Communications for the ACM, October, 1975. Volume 18, Issue 10. Pp. 568-579.



- [41] Hall P. A. V. Optimization of single expressions in a relational data base system // IBM Journal of Research and Development. Volume 20, Number 3, 1976. Pp. 244-257.
- [42] Chaudhuri S., Shim K. Including Group-By in Query Optimization // Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann, San Mateo, USA, 1994. San Francisco: Morgan Kaufmann Publishers Inc., 1994. Pp. 354-366.
- [43] Levy Y., Mumick I. S., Sagiv Y. Query Optimization by Predicate Move Around // Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann, San Mateo, USA, 1994. Morgan Kaufmann Publishers Inc., 1994. Pp. 96-107.
- [44] Wong E., Youssefi K. Decomposition - a strategy for query processing // ACM Transactions On Database Systems, September 1976. Volume 1, Number 3. Pp. 223-241.
- [45] Yannakakis M. Algorithms for acyclic database schemes // Proceedings of Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings. IEEE Computer Society 1981. New York: IEEE Press, 1981. Pp. 82-94.
- [46] McMahan B., Porter P., Pan G., M. Y. Vardi Projection Pushing Revisited // Proceedings of the 9th International Conference on Extending Database Technology, Heraklion, Crete, Greece, March 14-18, 2004. Berlin: Springer, 2004. Pp. 441-458.
- [47] McMahan B. J. Structural Heuristics for Query Optimization. Master of Science Degree Thesis. Houston: Rice University, 2004. 64 pp.
- [48] Kim W. On optimizing an SQL-Like Nested Query // ACM Transactions on Database Systems (TODS), September, 1982. Volume 7 Issue 3. Pp. 443-469.
- [49] Kisessing W. On Semantic Reefs and Efficient Processing of Correlation Queries Revisited // Proceedings of 11th International Conference for Very Large Data Bases. August 21-23, Stockholm, Sweden, 1985. New York: Morgan Kaufmann, 1985. Pp. 241-250.
- [50] Ganski R. A., Wong H. K. T. Optimization of Nested SQL Queries Revisited // Proceedings of the ACM SIGMOD international conference on Management of data. San Francisco, May 1987. New York: ACM, 1987. Pp. 23-33.
- [51] 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. Pp. 91-102.
- [52] Khaitan P., Satish K. M., S. B. Korra, S. K. Jena 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. Pp. 147-152.
- [53] Fegaras L., D. Levine, S. Bose, V. Chaluvadi Query processing of streamed XML data // Proceedings of the eleventh international conference on Information and knowledge management. ACM Press, New York, USA, 2002. Berlin: Springer, 2004. Pp. 195-215.
- [54] May N., Moerkotte G. Normalization and Translation of XQuery // Advanced Applications and Structures in XML Processing: Label Streams, Semantics Utilization and Data Query Technologies. Hershey: Igi Global Publishing, 2010. Pp. 283-307.
- [55] Pirahesh H., Hellerstein J., Hasan W. Extensible/rule based query rewrite optimization in Starburst // ACM SIGMOD Record, June 1, 1992. Volume 21, Issue 2. Pp. 39-48.
- [56] MySQL Nested-Loop Join Algorithms, <http://dev.mysql.com/doc/refman/5.5/en/nested-loop-joins.html>, [Обращение 20 сентября 2012].
- [57] Bancilhon F., Maierl D., Sagiv Y., Ullman J. D. Magic Sets and Other Strange Ways to Implement Logic Programs // Proceedings of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, Cambridge, Massachusetts, March 24-26, 1986. New York: ACM, 1986. Pp. 1-15.
- [58] Mumick S., Finkelsteint S. J., Pirahesh H., Ramakrishnan R. Magic Conditions // ACM Transactions on Database Systems (TODS), March 1996. Volume 21 Issue 1. Pp. 107-155.
- [59] Mumick I. S., Finkelstein S. J., Pirahesh H., Ramakrishnan R. Magic is Relevant // Proceedings of the 1990 ACM SIGMOD international conference on Management of data. New York: ACM, 1990. Pp. 247-258.
- [60] Jezek K., Zima M. Query optimization in deductive programs with aggregates // Proceeding of the 4th International Conference Information Systems Modelling, Ostrava: MARQ, 2001. Pp. 85-92.
- [61] Seshadri P., Hellerstein J. M., Pirahesh H., Cliff Leung T. Y., Ramakrishnan R., Srivastava D., Stuckey P. J., Sudarshan S. Cost-Based Optimization for Magic: Algebra and Implementation // ACM SIGMOD Record, June 1996. Volume 25, Issue 2, Pp. 28-33.
- [62] Sagiv Y. Is there anything better than magic? // Logic Programming, Proceedings of the 1990 North American Conference, Austin, Texas, October 29 - November 1, 1990. Austin: MIT Press 1990. Pp. 235-254.
- [63] Sippu S., Soisalon-Soininen E. An Analysis of Magic Sets and Related Optimization Strategies for Logic Queries // Journal of the ACM, November 1996. Volume 43, № 6. Pp. 1046-1088.
- [64] Azevedo P. J. Magic sets with full sharing // The Journal of Logic Programming, 1997. Volume 30, № 3. Pp. 223-237.
- [65] Faber W., Greco G., Leone N. Magic Sets and their application to data integration // Journal of Computer and System Sciences, June 2007. Volume 73, Issue 4. Pp. 584-609.
- [66] Ruckhaus E., Ruiz E., Vidal M. E. OnEQL: An Ontology Efficient Query Language Engine for the Semantic Web // Proceedings of the Workshop on Applications of Logic Programming to the Web, Semantic Web and Semantic Web Services (ALPSWS), Porto, Portugal, September 13th, 2007. Porto: CEUR Workshop Proceedings, 2007. Pp. 65-88.
- [67] Alviano M., Faber W., Greco G., Leone N. Magic Sets for Disjunctive Datalog Programs // Artificial Intelligence, 2012. Volume 187. Pp. 156-192.
- [68] Almendros-Jiménez M., Becerra-Terón A., Enciso-Banos F. J. Magic Sets for the XPath Language // Journal of Universal Computer Science, 2006. Volume 12, № 11. Pp. 1651-1678.
- [69] Ozcan F., Seemann N., Wang L. XQuery Rewrite Optimization in IBM DB2 pureXML // IEEE, Data Engineering Bulletin, December 2008. Volume 34, Number 4. Pp. 25-32.
- [70] Kolaitis P. G., Martin D.L., Thakur M.N. On the complexity of the containment problem for conjunctive queries with built-in predicates // Proceeding of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Seattle, Washington, USA, June 1-3, 1998. New York: ACM, 1998. Pp. 197-204.
- [71] Benedikt M., Gottlob G. The Impact of Virtual Views on Containment // Proceedings of the The Very Large Data Bases, September 2010. Volume 3 Issue 1-2. Pp. 297-308.
- [72] Kolaitis P. G., Vardi M. Conjunctive-query containment and constraint satisfaction // Proceeding of the 17th ACM SIGACT-SIGMOD-SIGART SIGART Symposium on Principles of Database Systems (PODS), Seattle, Washington, USA, June 1-3, 1998. New York: ACM, 1998. Pp. 205-213.

- [73] Kolaitis P. G., Vardi M. Conjunctive-Query Containment and Constraint Satisfaction // *Journal of Computer and System Sciences*, 2000. № 61. Pp. 302-332.
- [74] Goldstein J., Larson P.-A. Optimization Queries Using Materialized Views: A Practical Scalable Solution // *ACM SIGMOD Record*, June 2001. Volume 30 Issue 2. Pp.331-342.
- [75] Пашинин О. В. Оптимизация запросов к базам данных // *Математические структуры и моделирование*. Выпуск 17, 2007. С. 100-107ю
- [76] Chandra K., Merlin P. M. Optimal Implementation of Conjunctive Queries in Relational Databases // *Proceedings of the 9th annual ACM symposium on Theory of computing*, May, 1977. New York: ACM, 1977. Pp. 77-90.
- [77] Stroet J. W. M., Engmann R. Manipulation of expressions in a relational algebra // *Information Systems*, 1979. Volume 4, Issue 4. Pp. 195-203.
- [78] Aho V., Sagiv Y., Ullman J. D. Equivalences among relational expressions // *Society for Industrial and Applied Mathematics Journal on Computing*, 1979. Volume 8, Issue 2. Pp. 218-246.
- [79] Aho V., Sagiv Y., Ullman J. D. Efficient optimization of a class of relational expressions // *Journal ACM Transactions on Database Systems (TODS)*, December 1979. Volume 4, Issue 4. Pp.435-454.
- [80] Sagiv Y., Yannakakis M. Equivalences among relational expressions with the union and difference operators // *Journal of the ACM*, October 1980. Volume 27, Issue 4. Pp. 633-655.
- [81] 7.2.1.2 How MySQL Optimizes WHERE Clauses  
<http://dev.mysql.com/doc/refman/5.5/en/where-optimizations.html>, [Обращение 20 сентября 2012].
- [82] Приложение PostgreSQL 8.3.3., <http://www.postgresql.org/download/>, [Обращение 20 сентября 2012]. Адрес файла, содержащего обсуждаемый код, в архиве `postgresql-8.3.3\src\backend\optimizer\util\pretest.c`.
- [83] Veitch E. W. A Chart Method for Simplifying Truth Functions // *ACM Annual Conference/Annual Meeting: Proceedings of the 1952 ACM Annual Meeting*. Pittsburg: ACM, NY, 1952. Pp. 127-133.
- [84] Karnaugh M. The Map Method for Synthesis of Combinational Logic Circuits // *Transactions of the American Institute of Electrical Engineers*, November 1953. Part I, № 72 (9). Pp. 593-599.
- [85] Quin W. V. On Cores and Prime Implicants of Truth Functions // *American Mathematics Monthly*, 1959. V. 66, № 9. P. 755-760.
- [86] McCluskey E. J. Minimization of Boolean Functions // *The Bell System Technical Journal*, November 1956. V. 35, Issue 5. Pp. 1236-1249.
- [87] Wu M.-C. Query Optimization for Selecting Using Bitmaps // *ACM SIGMOD Record*, June 1999. Volume 28 Issue 2. Pp. 227-238.
- [88] Das Sarma A., Theobald M., Widom J. Exploiting Lineage for Confidence Computation in Uncertain and Probabilistic Databases. Technical Report. Stanford, 2007. <http://ilpubs.stanford.edu:8090/800/>, [Обращение 20 сентября 2012].
- [89] Тарасенко П. Ф., Бухарова М. Ф. Технология «The Reporter» для построения отчетов по базам данных // *Вестник Томского Государственного Университета*, № 275, апрель 2002. С. 167-176.
- [90] Mendkovich N., Kuznetsov S. New Algorithms for Lexical Query Optimization // *Proceedings of the 31st International Conference on Information Technology Interfaces*. Cavtat/Dubrovnik, Croatia, June 22-25, 2009. Zagreb: University of Zagreb, 2009. Pp. 187-192.
- [91] Кузнецов С. Д., Мендкович Н. А. Новые алгоритмы лексической оптимизации запросов // *Модели и анализ информационных систем*, 2009. Т. 16, № 4. С. 22-33.
- [92] Мендкович Н. А., Кузнецов С. Д. Оптимизация конъюнктов условий в составе запросов // *Модели и анализ информационных систем*, 2011. Т. 18, № 3. С. 144-154.
- [93] Bellamkonda S., Ahmed R., Witkowski A., Amor A., Zait M., Lin Ch.-Ch. Enhanced Subquery Optimizations in Oracle // *Proceedings of the 35th international conference on Very large data base*, August 2009. Volume 2 Issue 2. Pp. 1366-1377.
- [94] Chaudhuri S. Query Optimizers: Time to Rethink the Contract? // *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Providence, Rhode Island, USA, June 29 - July 2, 2009. New York: ACM, 2009. Pp. 961-968.