Общая характеристика работы

Актуальность проблемы

В настоящее время СУБД является одним из основных элементов современных ИС. СУБД – программный продукт, с помощью которого реализуется централизованное управление информацией хранимой в БД, а также доступ к ней и поддержание ее в состоянии, соответствующем предметной области. Эти функции СУБД должна выполнять с максимальной эффективностью, обеспечивая заданную производительность и заданное время отклика на запросы пользователя

В работах Джонатана Льюиса, Джефа Хольта, Керри Миллсап отмечено, что 80% проблем производительности СУБД связано с реализованными операторами. Решением данной проблемы занимались и занимаются многие российские и зарубежные ученые. В работах Григорьева Ю.А., Острикова С.П. рассматривается вопрос о нерациональном способе выполнения запросов с коррелированными подзапросами и операциями агрегирования, предлагаются новые способы решения. В свою очередь, в работах Григорьева Ю.А., Гребенникова Н.А. рассмотрены существующие методы поиска способов, выполнения запросов, предложены более качественные способы, на основе нисходящей стратегии анализа альтернативных планов. В работе Самарева Р.С. выполнен анализ производительности объектных СУБД и предложен метод организации параллельного выполнения запросов. Кузнецов Л.А., Погодаев А.К, Овчинников В.В предложили собственный алгоритм оптимизации запросов в СУБД. В работах Григорьева Ю.А, Гребенникова Н.А. и Плутенко А.Д. проводится детальный анализ методов оценки времени выполнения запросов. Из выполненного анализа проводимых исследований в области повышения быстродействия СУБД можно сказать, что в настоящее время существует несколько направлений исследований в данной области науки: разработка новых оптимизаторов запросов; повышение скорости обработки запросов за счет их параллельного выполнения; повышения скорости выполнения некоторых классов запросов для современных СУБД.

С целью уменьшения времени выполнения запросов в состав СУБД включен оптимизатор, задача которого – анализ множества альтернативных планов выполнения запросов и выбор рационального плана. План запроса - это форма запроса написанного на языке высокого уровня, преобразованная в корректную и эффективную последовательность операций, записанных на языке низкого уровня, реализующем операции реляционной алгебры. С каждой такой операцией связана параметризированная формула, позволяющая оценить стоимость операции. Стоимость плана - это сумма стоимостей отдельных операций, входящих в его состав. Задача СУБД сводится к вычислению формул стоимости для каждой отдельной операции. Основная проблема состоит в том, что формулы стоимости выполнения зависят от размера отношений (или отношений), которые обрабатывает эта процедура. Поскольку все запросы, за исключением самых простых, требуют создания некоторых промежуточных результатов выполнения (по меньшей мере концептуально), СУБД должна быть

способна оценить размер этих промежуточных результатов (оценить кардинальность) и использовать полученные значения при вычислении формул стоимости. Кардинальность – прогнозируемое количество записей, генерируемых операцией, сильно зависит от конкретных значений хранимых данных, поэтому точная оценка стоимости является достаточно сложной задачей. Как было отмечено выше, данное понятие играет критическую роль в выборе первоначального порядка соединения отношений и выбора методов доступа, поэтому ее неверная оценка негативно сказывается на производительности.

Анализ литературных источников позволяет сказать, что современные способы повышения скорости выполнения запросов поиска данных в БД предполагают использование вейвлетов или сэмплинга. Но эти методы не эффективны в таком классе запросов, как запросы с операторами поиска подстрок, реализованных при помощи оператора Like. В случае использования оператора Like количество возвращаемых записей определяется как строго заданный процент от размера отношения, на котором оператор был выполнен. Например, в СУБД Oracle версий 9, 10 и 11 это 5%. Проведенные исследования показали, что данная ошибка при оценке кардинальности запросов с оператором Like снижает производительность БД и количество подобных запросов достигает 15% от всех запросов, выполняемых в БД.

Выше сказанное позволяет говорить об **актуальности** и необходимости решения проблемы повышения скорости выполнения запросов с операторами поиска подстрок.

Цель и основные задачи работы

Цель – исследование и разработка методов и алгоритмов преобразования информации в БД для повышения скорости выполнения запросов поиска данных. Для достижения указанной цели в диссертации поставлены и решены следующие задачи:

1. Анализ процессов выполнения запросов поиска данных и выделение класса запросов требующих повышения скорости выполнения;
2. Разработка методов повышения скорости выполнения запросов выделенного класса;
3. Разработка формализованных процедур и алгоритмов позволяющих реализовать предложенные методы;
4. Экспериментальная оценка уменьшения времени выполнения запросов и снижения загрузки ЦП при использовании предложенных методов.

Методы исследования. Для решения поставленных задач в данной работе использовались методы теории генетических алгоритмов, объектно-ориентированного программирования, теории баз данных и систем управления базами данных, методы модульного и структурного программирования.

Научная новизна заключается в следующем:

1. Выполнен анализ выполнения запросов с операторами поиска подстрок, и выявлена ошибка определения кардинальности, в данном классе запросов снижающая скорость выполнения запроса;

2. Разработаны новые способы оценки кардинальности исследуемого класса запросов;
3. Разработаны новые методы повышения скорости выполнения запросов выделенного класса, на основании предложенных способов определения кардинальности;
4. Разработан способ экспериментальной проверки разработанных методов при помощи ранее выполненных в БД запросов и трассировке сессии для получения результатов эксперимента;

Практическая новизна результатов работы состоит в:

1. Использовании предложенных методов для решения практических задач повышения производительности БД;
2. Разработанном программном обеспечении, реализующем предложенные способы определения кардинальности;
3. Разработанном способе получения ранее выполненных в БД запросов для проведения экспериментов, реализации предложенных методов повышения эффективности работы БД и анализа выполняемых в БД запросов;
4. Разработанном программном обеспечении определения эффективности предложенных методов повышения скорости выполнения запросов;
5. Полученных практических результатах внедрения предложенных методов.

На защиту выносятся следующие положения:

1. Метод повышения скорости выполнения исследуемого класса запросов на основании разработанного способа с повышенной точностью определения кардинальности;
2. Метод повышения скорости выполнения исследуемого класса запросов на основании разработанного способа с низкой вычислительной стоимостью определения кардинальности;
3. Способ оценки рациональности использования предложенных методов;
4. Результаты экспериментов по оценке точности определения кардинальности предложенными способами и повышению скорости выполнения запросов исследуемого класса.

Апробация работы. Содержание отдельных разделов диссертации было доложено:

1) на всероссийской научно-технической конференции «Наукоемкие технологии в прибора- и машиностроении и развитие инновационной деятельности в вузе» (Москва 2009);

2) на всероссийской научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» (Томск 2010);

3) на всероссийской научной школе-семинаре студентов, аспирантов и молодых ученых «Интеллектуализация информационного поиска, скантехнологии и электронные библиотеки» (Таганрог 2010);

4) на всероссийской научной конференции с международным участием «Научное творчество XXI века» (Красноярск 2010);

5) на II научно технической конференции студентов аспирантов и молодых специалистов «Информационные системы и технологии 2009» (Обнинск 2009);

6) на II всероссийской конференции «НГАСУ (Сибстрин)» (Новосибирск 2009);

7) на семинарах и заседаниях кафедры «Компьютерные системы и сети» КФ МГТУ им. Н.Э. Баумана.

Публикации. По материалам диссертации опубликовано 16 работ, отражающих основные результаты работы, в том числе 3 работы в журналах, рекомендованных перечнем ВАК.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, библиографического списка из 116 наименований. Работа содержит 150 страниц машинописного текста содержательной части, 21 рисунок, 13 таблиц и 12 страниц библиографии.

Содержание работы

Во введении обосновывается актуальность научной задачи, сформулированы цель и задачи диссертационного исследования, основные положения, выносимые на защиту. Показана новизна исследований, проведенных автором, их практическая ценность, приведена структура диссертации.

В первой главе выполнен анализ предметной области, дан краткий обзор технологий выбора способа выполнения запросов в БД, рассмотрены основные этапы обработки запросов в СУБД, в частности шаг выбора и формирования плана выполнения.

Выполнен обзор текущего состояния решения задачи выбора рационального плана выполнения запроса в БД. Перечислена основная терминология, решаемые задачи, а также показана специфика данной области и существующие в ней проблемы. Выполнен анализ современного состояния исследований в области повышения производительности СУБД за счет улучшения работы оптимизатора запросов и разработки новых методов выполнения классов запросов. Показана важность точного определения кардинальности (прогнозируемого количества записей, генерируемых операцией) на выбор рационального способа выполнения запроса и как следствие, на скорость его выполнения.

Выявлена проблема неверной оценки кардинальности в запросах с операторами поиска подстрок, реализованных при помощи оператора Like, и влияние выявленной ошибки на выбор в СУБД плана выполнения запросов данного типа. Пример исследуемого класса запросов приведен ниже: SELECT * FROM t1 WHERE t1.id IN (SELECT t2.id_t1 FROM t2 WHERE a LIKE '%111%').

Разработан способ выборки ранее выполненных в БД операторов для определения частоты встречаемости исследуемого класса запросов в БД. При помощи данного способа на примере производственных БД была определена

частота встречаемости исследуемого класса запросов в конструкторско-технологических БД, которая составила 15%.

Показано влияние определенной ошибки оценки кардинальности подзапроса с оператором Like на скорость выполнения запросов, в связи с выбором не рационального плана выполнения запроса и как следствие и снижение скорости выполнения данного класса запросов.

Выполнен анализ существующих исследований, показавший незначительное число готовых и апробированных решений в данной области. В настоящее время наиболее распространенный метод - это использование строго заданного значения кардинальности при выполнении запросов с оператором Like. Например, в СУБД Oracle версий 9, 10 и 11 это 5% от размера отношения, на котором оператор был выполнен. Проведенные предварительные исследования показали существенное повышение скорости выполнения запросов анализируемого класса при повышении точности оценки кардинальности на промежуточных этапах выполнения. Это позволило сформировать задачу работы, а именно разработка способов определения кардинальности запросов с операторами поиска подстрок для повышения скорости выполнения запросов поиска данных.

Были описаны критерии, предъявляемые к разрабатываемым способам. В частности предложены два выражения, первое

$$C_T = \frac{S_{CT} * N}{B_T * M},$$

для определения стоимости оценки кардинальности запросов с операторами поиска подстрок, если доступ к данным осуществляется путем полного просмотра таблицы (FTS), где N – количество кортежей, хранимых в отношении; B_T – объем пространства доступного в блоке данных для хранения кортежей отношения; S_{CT} – средний размер одного кортежа, хранимого в отношении; M - количество блоков, считываемых при операции многоблочного чтения. Второе выражение:

$$C_И = H + \frac{S_{ЗП} * N}{B_И * M},$$

для определения стоимости оценки кардинальности запросов с операторами поиска подстрок, если доступ к данным осуществляется путем быстрого полного просмотра индекса, где $B_И$ – объем пространства, доступный в блоке данных для хранения записей индекса; $S_{ЗП}$ – средний размер одной записи, хранимой в индексе, построенном по требуемому атрибуту отношения; H – высота индекса.

Полученные выражения используются как критерий вычислительной сложности выполнения операций по оценке кардинальности в запросах с операторами поиска подстрок.

Были сформулированы ограничения на класс исследуемых запросов и представление в СУБД условий поиска из запросов.

Во второй главе описаны разработанные методы повышения скорости выполнения запросов с операторами поиска подстрок. Предложенные методы позволяют повысить скорость выполнения исследуемого класса запросов за счет более точного определения кардинальности подзапроса с оператором Like. Это позволяет выбрать СУБД рациональный план выполнения запроса.

Пусть в БД есть множество отношений $T = (t_1, t_2, t_3, ..., t_k)$. Выделим некоторое отношение t. Тогда $A = (a_1, a_2, a_3, ..., a_i)$ - атрибуты, определенные на данном отношении. По некоторому атрибуту a, определенному на отношении t, выполняются запросы с операторами поиска подстрок, задаваемому при помощи оператора Like. $S = (s_1, s_2, s_3, ...s_i)$ - множество условий поиска, по которым ранее осуществлялась выборка данных с использованием оператора Like по атрибуту a отношения t. По множеству S определим классы запрашиваемых условий поиска. На основании полученных классов осуществим прогнозирование значений возможных переменных, которые в будущем пользователь может указать в запросе, и определим их кардинальность.

В дальнейшем, при выполнении запросов с оператором Like по атрибуту a отношения t, будем извлекать из запроса значения переменных, искать их в прогнозируемых переменных и указывать найденную кардинальность.

Для реализации данного метода повышения скорости выполнения запросов необходим критерий, позволяющий оценить сходство группы строк – условий поиска. Для решения данной задачи используются средства регулярных выражений. Если некоторая строка удовлетворяет регулярному выражению, то она входит в класс, описанный этим выражением Таким образом, регулярное выражение есть признак класса.

Для каждой строки s из множества S существует множество регулярных выражений $R = (r_1, r_2, r_3, ...r_i)$, удовлетворяющих s.

Для каждого регулярного выражения r, входящего во множество регулярных выражений R, удовлетворяющих некоторому условию поиска s, определим функцию w. Данная функция возвращает набор подстрок из строки d, удовлетворяющих регулярному выражению r, на котором определена данная функция.

Из множества регулярных выражений R, удовлетворяющих строке s, необходимо выбрать некоторое регулярное выражение r’, по которому будет осуществляться прогнозирование. На данной r’ определим функцию w’. Путем применения функции w’ ко множеству значений $D = (d_1, d_2, d_3, ...d_i)$, которые принимает атрибут a отношения t, по которому осуществлялся поиск с использованием строки s, получим множество неповторяющихся подстрок $D' = (d'_1, d'_2, d'_3, ...d'_j)$.

Определим h=f(d’,a), как количество строк из множества D, определенного на атрибуте a, содержащих подстроку d’. Для каждой d’ из множества D’ определим h=f(d’,a) и получим множество $H = (f(d'_1, a), f(d'_2, a), f(d'_3, a), ...f(d'_j, a))$ или $H = (h_1, h_2, h_3, ...h_j)$.

Определив r’ для каждой s из множества S, получим множество $R' = (r'_1, r'_2, r'_3, ...r'_i)$. Для каждой r’ из множества R’ получим множество D’ и соответствующее ему множество H. В дальнейшем, при выполнении запросов будем из множества R’ выбирать регулярное выражение, удовлетворяющее s, и в соответствующей r’ паре D’-H искать s=d’. В случае успеха, при помощи подсказки оптимизатору программным путем будем передавать кардинальность запроса в СУБД, равную h=f(d’,a), где d’=s, или использовать среднее значение кардинальности для класса.

Для предложенного алгоритма необходимо определить критерий выбора регулярного выражения r’ из множества регулярных выражений $R = (r_1, r_2, r_3, ...r_i)$, удовлетворяющих s.

Определим f(s,a), как количество строк из множества D, определенного на атрибуте a отношения t, содержащих строку s.

Определим f’(r,S), как количество строк из множества S, определенного на атрибуте a отношения t, удовлетворяющих регулярному выражению r.

Для классификации и прогнозирования условий поиска необходимо выбрать регулярное выражение r, которое удовлетворяет следующим критериям:

- $w(s) = s$ - регулярное выражение r удовлетворяет значению условия s. При применении функции w, определенной на регулярном выражении r, на строку s, будет возвращен результат равный s;
- $f(s,a) > k$ - если данная s содержится хотя бы в k элементах из множества D, определенного как значения атрибута a отношения t;
- $f'(r,S) = max(f'(r_i,S)) > l$ - если для данного r, f’(r,S) имеет максимальное значение из всех r во множестве R, построенных по s и f’(r,S) больше некоторой l. Т.е. r удовлетворяет наибольшему количеству подстрок поиска s из множества S по сравнению с другими r, построенным по тому же s;
- $\exists d' = s; d' \in D' = w(D)$ - если во множестве D’, полученном путем применения функции w, определенной на регулярном выражении r на множество D, существует d’=s.
- $|D'| = \max(|w(D)|)$ - если регулярное выражение имеет наибольшее результирующее множество, полученное путем применения функции w, определенной на данном регулярном выражении к множеству значений, которые принимает атрибут отношения, по которому выполняется выборка.

Для получения множества R из s используется генетический алгоритм. Разработанный ГА представлен в виде черного ящика, на его вход подается строка – условие поиска из ранее выполненного запроса. На выходе получают множество регулярных выражений, удовлетворяющих данной строке. Особи популяции – регулярные выражения представляются на естественном языке. Применяемый алфавит состоит из букв языка, используемого для хранения

информации. Также алфавит включает арабские цифры, знаки пунктуации и закодированные метасимволы регулярных выражений.

В качестве операций получения новых особей используются операции скрещивания и мутации. Особи для скрещивания выбираются случайным образом из всех особей, входящих в популяцию. Точки скрещивания и мутирующий ген также выбираются случайно. Особь включается в популяцию, если она удовлетворяет входной строке. ГА прекращает свою работу, если в течение заданного, достаточно большого количества циклов, не было получено ни одного нового регулярного выражения (особи), удовлетворяющей входной строке, или получено прогнозируемое количество регулярных выражений, удовлетворяющих входной строке.

Прогнозируемое количество особей, входящих в популяцию, рассчитывается на основании значения и длины входной строки. В общем случае размер популяции регулярных выражений P, удовлетворяющих условию поиска, будет зависеть от ее размера. Пусть имеем некоторую подстроку поиска $s = s_1 s_2 s_3 ... s_n$, где n – количество символов, входящих в нее. Тогда P будет определяться выражением:

$$P = \sum_{k=0}^{n} C_n^k,$$

где C_n^k - сочетание из n элементов по k и определяется выражением:

$$C_n^k = \frac{n!}{k!(n-k)!};$$

Таким образом, P можно представить выражением:

$$P = 1 + \frac{n!}{(n-1)!} + ... + \frac{n!}{k!(n-k)!};$$

Действительно, любая строка s удовлетворяет регулярному выражению r=s, что и описывает первое слагаемое выражения. Если заменить любой символ в строке s, на некоторый метасимвол m, который удовлетворяет любому символу, то строка s может быть представлена $n!/(n-1)!$ количеством регулярных выражений, что удовлетворяет второму слагаемому выражения и т.д.

Для каждого s из множества R полученного при помощи ГА выберем r’ – регулярное выражение для классификации и прогнозирования. Из всех r’ получим множество R’.

При помощи функций w', определенных на регулярных выражениях из множества R’, выполним прогнозирование. Подав на вход функции значения атрибута отношения, анализ которого был выполнен, получим множество строк. Для каждой полученной строки определим значение h=f(s,a) и будем его использовать, как значение кардинальности данной подстроки. Также для каждого класса условий поиска определим ее среднюю кардинальность $h_{CP} = \sum_{j=1}^{n} h_j \Big/ n$, где n – мощность множества H для r’.

Первый способ определения кардинальности по значениям прогнозируемых условий поиска обеспечивает повышенную точность и, как следствие, большее уменьшение времени выполнения запроса и загрузки ЦП.

Второй способ определения кардинальности использует среднее значение кардинальности для класса - h_{CP}, которому удовлетворяет условие поиска из запроса. В этом случае не требуется хранить прогнозируемые условия поиска и выполнять выборку по ним, что снижает вычислительную стоимость, но падает точность определения кардинальности и, как следствие, время выполнения запроса и загрузка ЦП становится больше.

На основании двух предложенных способов определения кардинальности разработаны два метода повышения скорости выполнения запросов.

В первом методе при помощи первого разработанного способа определяется кардинальность запроса, если указанная пользователем условие поиска найдено среди прогнозируемых. Если такого условия поиска нет, то используется второй способ определения кардинальности. Если для данного условия поиска нет удовлетворяющего ему регулярного выражения, то используется классический способ определения кардинальности. Вычислена кардинальность при помощи подсказки (ключевого слова) передается в СУБД.

Во втором методе используется второй способ определения кардинальности. Если его применение невозможно из-за отсутствия класса которому удовлетворяет условие поиска указанное пользователем в запросе, то используется классический способ определения кардинальности. Аналогично первому методу вычислена кардинальность при помощи передается в СУБД.

Описание классов условий поиска (регулярных выражений) и значений прогнозируемых условий поиска с кардинальностью хранятся непосредственно в БД. Для повышения скорости работы, для хранения данных о прогнозируемых строках используется древообразная структура – индекс. Для хранения классов - регулярных выражений используется отношение. При обработке запросов с оператором Like выполняется поиск регулярного выражения, удовлетворяющего искомой подстроке среди ранее полученных. Затем выполняется поиск искомой подстроки из данного класса.

Пусть R - количество полученных регулярных выражений; S_{PB} - средний размер записи о регулярном выражении. Стоимость чтения информации о всех классах определяется выражением:

$$C_{PB} = \frac{S_{PB} * R}{B_T * M},$$

где H - высота индекса, построенного по прогнозируемым строкам.

Стоимость выполнения первого предложенного способа определения кардинальности описывается выражением:

$$C = \frac{S_{PB} * R}{B_T * M} + H + 1;$$

Стоимость определяется необходимостью чтения всех регулярных выражений (первое слагаемое выражения) для выбора, удовлетворяющего искомой подстроке, и последующего поиска данной подстроки с

использованием индекса (H+1). Сложение с единицей вызвано необходимостью доступа к значению искомого условия по идентификатору строки, полученному из индекса. Напомним, что для сравнения эффективности ранее были предложены выражения:

$$C_И = H + \frac{S_{ЗП} * N}{B_И * M},\ \text{и}\ C_Т = \frac{S_{СТ} * N}{B_Т * M};$$

Предлагаемый способ будет эффективнее, если выполняется условие $C_И > C$ или $C_Т > C$, в зависимости от типа доступа.

$$H + \frac{S_{ЗП} * N}{B_И * M} > \frac{S_{РВ} * R}{B_Т * M} + H + 1;$$

По причине того, что стоимость выполнения выражается в операциях одноблочного чтения и данное значение для исследуемого класса запросов велико, единицей можно пренебречь, таким образом, получаем следующее неравенство:

$$\frac{S_{ЗП} * N}{B_И * M} > \frac{S_{РВ} * R}{B_Т * M},\ \text{или}\ \frac{S_{ЗП} * N}{B_И} > \frac{S_{РВ} * R}{B_Т};$$

В связи с выбранным способом кодирования регулярных выражений, их длина не будет превышать длину данных, хранимых в поле, по которому производится выборка, т.е. в худшем случае $S_{ЗП} = S_{РВ}$. Также можно пренебречь разницей между $B_И$ и $B_Т$ и тогда получим:

$$N > R;$$

Следовательно, первый разработанный способ целесообразно использовать в случае, если количество строк, хранимых в отношении, по которому выполняется поиск с использованием оператора Like больше, чем количество регулярных выражений, полученных после анализа ранее искомых подстрок поиска.

Для второго способа стоимость определяется чтением только регулярных выражений, т.к. прогнозируемые строки не хранятся, следовательно, отсутствует составляющая (H+1). При этом высота индекса в большинстве случаев равна 2-3 узлам. Если объем оперируемых данных велик, H также можно пренебречь. Критерий эффективности использования второго разработанного способа аналогичен первому.

В третьей главе приведены результаты экспериментальных исследований разработанных методов повышения скорости выполнения запросов исследуемого класса. В рамках эксперимента был проведен анализ работы предложенного ГА; выполнена оценка точности определения кардинальности по предлагаемым способам; выполнена сравнительная оценка уменьшения времени выполнения запросов и загрузки ЦП при использовании предложенных методов повышения скорости выполнения запросов по сравнению с запросами, в которых кардинальность определена традиционными способами.

Эксперименты проводились на примере конструкторско-технологических БД по запросам выборки данных из отношений содержащих, обозначения изделий (узлов, деталей, стандартных изделий, материалов и т.д.).

Для фиксации результатов эксперимента, а именно времени выполнения запросов и загрузки ЦП использовалась трассировка сессии. Для исключения грубых погрешностей при обработке результатов применялся аппарат проверки статистических гипотез, а именно критерий Романовского. В соответствии с ним конкурирующая гипотеза о наличии грубых погрешностей в подозрительных результатах подтверждается, если выполняется неравенство: $\left|x_{iпод} - \overline{X}_{Ц.Р.}\right| \geq t_P S$, где t_P – квантиль распределения Стьюдента при заданной доверительной вероятности с числом степеней свободы $k = n - k_n$ (k_n - число подозрительных результатов наблюдений).

Точечные оценки распределения $\overline{X}_{Ц.Р.}$ и средне квадратичного отклонения S результатов наблюдений вычисляется без учета k_n подозрительных результатов наблюдений. За оценку центра $\overline{X}_{Ц.Р.}$ принималась медиана вариационного ряда оценок: среднее арифметическое; медиана наблюдений; среднее арифметическое 90%-ной выборки; срединный размах вариационного ряда; центр размаха. Значение доверительной вероятности выбрано как 0.95. Параметры выполнения эксперимента и характеристики экспериментальных БД представлены в таблице 1.

Таблица 1.

Параметр эксперимента	Значение параметра
Вид экспериментальных БД	Конструкторско-технологические
Экспериментальные СУБД	Oracle, MSSql
Кол-во пользователей работающих в БД	350-1700
Объем тестовых БД	200 - 500Гб
Кол-во записей хранимых в отношениях БД, по которым выполнялись запросы	100 тыс. – 70 мил.
Кол-во тестовых запросов	Порядка 30000
Количество экспериментов	20
Средства фиксации результатов эксперимента	Трассировка сессии БД
Тестовые запросы	Ранее выполненные в БД
Кол-во выполняемых транзакций в БД	20 в секунду
Среднее кол-во логических чтений в БД	900 тыс. операций/сек
Среднее кол-во физических чтений в БД	160 тыс. операций/сек
Среднее кол-во физических записей в БД	100 тыс. операций/сек
Среднее кол-во сортировок в БД	600 операций/сек
Среднее кол-во новых подключений пользователей к БД	0.15 в сек

Результаты экспериментов по оценке времени выполнения запросов представлены в таблице 2.

Таблица 2.

1	2	3	4	5	6	7	8
1	18,06	17,08	12,04	0,83	0,56	1,25	0,38
2	17,58	17,80	11,99	1,73	1,22	0,97	0,52
3	18,42	17,87	12,92	2,00	0,56	0,80	0,98
4	18,05	17,58	12,38	0,74	1,05	1,32	0,37
5	17,99	17,46	12,17	1,21	0,66	0,17	0,78
6	18,39	18,15	13,34	1,16	0,80	1,48	0,63
7	18,41	17,88	12,12	0,69	1,42	0,81	0,78
8	18,68	18,59	12,80	0,50	0,44	0,25	0,95
9	17,85	17,27	12,18	0,40	0,98	0,31	0,43
10	18,00	17,02	12,40	0,97	0,41	1,34	1,28
11	18,52	17,82	11,83	0,79	0,47	0,57	0,04
12	17,38	17,17	12,34	0,59	0,97	0,69	0,44
13	18,38	16,86	12,17	0,66	0,47	1,01	0,22
14	17,63	16,84	12,34	0,57	0,14	1,66	1,30
15	17,89	17,47	12,42	0,46	1,24	0,67	0,85
16	17,37	17,55	12,78	1,21	1,15	0,13	0,29
17	18,45	17,43	12,17	0,58	1,10	0,63	0,83
18	17,84	17,32	12,04	1,09	0,83	1,03	0,85
19	18,39	17,36	12,90	1,05	0,69	1,15	0,36
20	18,03	17,49	12,48	0,92	0,45	0,70	0,76
Среднее	**18,06**	**17,50**	**12,39**	**0,91**	**0,78**	**0,85**	**0,65**

Столбцы таблицы содержат следующую информацию:

1 – номер эксперимента;

2 – выигрыш во времени выполнения запросов с точно определенной кардинальностью по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

3 – выигрыш во времени выполнения запросов с кардинальностью, определенной при помощи первого разработанного способа, по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

4 – выигрыш во времени выполнения запросов с кардинальностью, определенной при помощи второго разработанного способа, по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

5 – значение среднеквадратичного отклонения результатов эксперимента для набора запросов с классическим способом определения кардинальности;

6 – значение среднеквадратичного отклонения результатов эксперимента для набора запросов с точно определенной кардинальностью;

7 – значение среднеквадратичного отклонения результатов эксперимента для набора запросов с кардинальностью, вычисленной первым разработанным способом;

8 – значение среднеквадратичного отклонения результатов эксперимента для набора запросов с кардинальностью, вычисленной вторым разработанным способом;

Результаты экспериментов по оценке загрузки ЦП после удаления данных с большой погрешностью представлены в таблице 3.

Таблица 3.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	21,28	20,73	17,29	0,42	0,40	0,17	0,17
2	21,11	20,83	17,63	0,41	0,31	0,20	0,34
3	21,90	21,04	17,93	0,31	0,25	0,64	0,39
4	21,38	21,01	18,01	0,37	0,31	0,36	0,69
5	21,78	20,94	17,89	1,00	0,50	0,69	0,28
6	21,67	20,94	17,78	0,34	0,13	0,19	0,34
7	21,72	21,41	17,68	0,66	0,29	0,51	0,13
8	21,57	21,27	17,81	0,86	0,18	0,24	0,53
9	21,80	21,05	17,73	0,71	0,44	0,28	0,41
10	21,56	21,06	17,79	0,91	0,33	0,24	0,39
11	21,49	20,82	17,47	0,63	0,33	0,43	0,29
12	21,58	21,11	17,71	1,42	0,30	0,03	0,24
13	21,91	20,99	17,87	0,82	0,39	0,34	0,66
14	21,46	20,81	17,40	0,91	0,35	1,07	0,03
15	21,15	20,45	17,19	0,20	0,32	0,73	0,39
16	21,33	20,70	17,47	0,65	0,30	0,49	0,52
17	21,71	20,94	17,74	0,84	0,66	0,25	0,33
18	21,54	20,96	17,67	0,26	0,46	0,12	0,12
19	21,67	21,33	17,88	1,28	0,77	0,72	0,81
20	20,96	20,53	17,66	0,70	0,79	1,20	0,91
Среднее	**21,53**	**20,95**	**17,68**	**0,68**	**0,39**	**0,45**	**0,40**

Столбцы таблицы содержат следующую информацию:

1, 5-8 аналогично таблице 1.

2 – уменьшение загрузки ЦП при выполнении запросов с точно определенной кардинальностью по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

3 – уменьшение загрузки ЦП при выполнении запросов с кардинальностью, определенной при помощи первого способа, по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

4 – уменьшение загрузки ЦП при выполнении запросов с кардинальностью, определенной при помощи второго способа, по сравнению с аналогичными запросами, в которых кардинальность определена классическим способом;

На графиках (рис. 3, 4) приведено уменьшение времени выполнения запросов, и загрузки ЦП при использовании предложенных способов определения кардинальности в разработанных методах повышения скорости выполнения запросов.

Рис. 3. Выйгрыш при использовании первого метода

Рис. 4. Выйгрыш при использовании второго метода

Эксперимент показал, что использование первого метода в среднем позволяет уменьшить время выполнения запросов на 17%, загрузка ЦП в среднем снижается на 20%. В случае использования второго метода время выполнения запросов в среднем уменьшается на 12%, загрузка ЦП на 17%.

В четвертой главе описана разработанная библиотека классов на языке C#, позволяющая выполнить классификацию условий поиска, прогнозирование возможных условий поиска и определение их кардинальности на основании входного набора запросов, ранее выполненных в БД. Также предложена схема организации БД для хранения служебной информации, используемой для реализации разработанных методов.

В заключении сформулированы основные выводы и результаты диссертационной работы.

1. Разработан метод повышения скорости выполнения выделенного класса запросов на основании разработанного способа определения

кардинальности запросов с операторами поиска подстрок с повышенной точностью;
2. Разработан метод повышения скорости выполнения выделенного класса запросов на основании разработанного способа определения кардинальности запросов с операторами поиска подстрок с низкой вычислительной стоимостью;
3. Выполнена экспериментальная оценка корректности и эффективности выносимых на защиту результатов, показана возможность использования предложенных методов;
4. Использование метода с повышенной точностью оценки кардинальности позволяет уменьшить время выполнения запросов в среднем на 17%, при этом в среднем на 20% снижается загрузка ЦП;
5. Использование метода оценки кардинальности с низкой вычислительной стоимостью позволяет уменьшить время выполнения запросов в среднем на 12%, загрузку ЦП в среднем на 17%;
6. Результаты диссертации внедрены на промышленных предприятиях ОАО НПП Калужский приборостроительный завод «Тайфун», ОАО «Русполимет», в учебный процесс КФ МГТУ им. Н.Э.Баумана, что подтверждается актами внедрения.

Список работ по теме диссертации

1. Либман М.С. Влияние размера блока базы данных на работу индексов // Наукоемкие технологии в приборо- и машиностроении и развитие инновационной деятельности в вузе: Тез. докл. региональной н-т. конф. Москва, 2009. Т. 2. С. 77-81.
2. Либман М.С. Рекомендации веб-програмисту для минимизации ввода/вывода в базе данных взаимодействующей с приложением // Информационные системы и технологии: Тез. докл. региональной н-т. конф. Обнинск, 2009. С. 17-18.
3. Мазин А.В., Либман М.С. Влияние размера блока табличного пространства на скорость выполнения операций чтения/записи в БД // Актуальные проблемы строительной отрасли: Тез. докл. Всерос. конф. Новосибирск, 2009. С. 184-185..
4. Либман М.С. Телерман Н.Э. Ошибка оптимизатора при оценке кардинальности запросов с оператором Like // Наукоемкие технологии в прибора- и машиностроении и развитие инновационной деятельности в вузе: Материалы Всерос. н-т конф. Москва, 2009. С. 93-97.
5. Либман М.С. Алгоритм оценки кардинальности запросов с оператором Like // Наукоемкие технологии в прибора- и машиностроении и развитие инновационной деятельности в вузе: Тез. докл. региональной н-т. конф. Москва, 2009. Т. 3. С. 98-100.
6. Либман М.С. Определение оптимального регулярного выражения для описания подстроки поиска // Наукоемкие технологии в прибора- и машиностроении и развитие инновационной деятельности в вузе: Тез. докл. региональной н-т. конф. Москва, 2009. Т. 3. С. 101-103.

7. Либман М.С. Использование генетического алгоритма для решения задачи поиска оптимального регулярного выражения // Молодежь и современные информационные технологии: Сборник трудов Всерос. н-пр. конф. Томск, 2010. С 111-112.
8. Либман М.С. Исследование вопроса об изменении классов данных, запрашиваемых пользователем // Применение кибернетических методов в решении проблем общества XXI века: Тез. докл. межрегиональная н-т. конф. Обнинск, 2010. С. 16-17.
9. Либман. М.С. Расчет стоимости выполнения SQL запросов с операцией Like // Наукоемкие технологии в приборо- и машиностроении и развитие инновационной деятельности в вузе: Материалы региональной н-т. конф. Москва, 2010. Т. 2. С. 109-111.
10. Либман М.С., Мазин А.В. Влияние ошибки оценки кардинальности оптимизатором на производительность баз данных // Интеллектуализация информационного поиска, скантехнологии и электронные библиотеки: Сб. тр. всерос. науч. школы-семинара. – Таганрог, 2010. С. 138-140.
11. Либман М.С., Мазин А.В., Телерман Н.Э. Получение регулярных выражений удовлетворяющих строке при помощи генетического алгоритма // Научное творчество XXI века: Материалы Всерос. н. конф. Красноярск, 2010. С. 42.
12. Либман М.С. Мазин А.В. Повышение точности оценки кардинальности SQL запросов с оператором Like // Вопросы радиоэлектроники. 2010. Выпуск 4. С. 155-167.
13. Либман М.С. Анализ выполнения запросов с оператором Like в базах данных // Сборник науч. работ лауреатов областных премий и стипендий. 2010. Выпуск 6. С. 75-79.
14. Либман М.С. Мазин А.В. Тестирование производительности баз данных // RSDN Magazine. 2010. Выпуск 2. С. 3-5.
15. Либман М.С. Мазин А.В. Анализ эффективности предлагаемых методов повышения точности оценки кардинальности запросов с подзапросами поиска данных, удовлетворяющих шаблону // T-com. Телекоммуникации и транспорт. 2010. Выпуск 5. С. 20-25.
16. Либман М.С., Мазин А.В., Телерман Н.Э. Использование генетического алгоритма для описания классов условий поиска // Радиопромышленность. 2011. Выпуск 1. С. 127-141.