УДК 004.622: 004.658.2

С.Л. ЗИНОВАТНАЯ, А.Ю. ЛЕВЧЕНКО, К.А. ГАЛАНЮК

Одесский национальный политехнический университет, Украина

ИМИТАЦИОННАЯ МОДЕЛЬ ДЛЯ ПРОЕКТИРОВАНИЯ ОПТИМАЛЬНОЙ ИНДЕКСНОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ

Предложен метод формирования оптимальной индексной структуры базы данных для повышения производительности информационной системы, использующей эту базу данных. Представлено формальное описание индексной структуры таблиц с учетом поведения системы, выраженного множеством запросов. Разработан алгоритм формирования набора индексов-кандидатов для получения наименьшего времени выполнения запросов в информационной системе. Поиск оптимальной структуры индексов выполняется с использованием имитационной модели поведения информационной системы в условиях применения различных наборов индексов.

Ключевые слова: *имитационная модель, индекс, оптимизация индексной системы, производительность информационной системы, база данных, SQL.*

Введение

Очевидно, что в настоящее время информационные системы (ИС) используются во всех областях жизни человека. С ростом объема данных, изменением числа пользователей, возникновением новых задач в ИС могут появиться проблемы недостаточной производительности. Улучшить производительность ИС, в основе которой находится реляционная база данных (БД), можно различными способами, в том числе не связанными с заменой аппаратуры. Вариантами изменений, приводящих к уменьшению времени выполнения запросов к БД, являются реструктуризация таблиц БД, использование материализованных представлений, репликация данных, настройка индексной структуры и т.п. [1, 2].

Применение всех указанных методов предусматривает предварительное исследование не только структуры РБД, но и её работы в течение определенного периода времени.

В различных системах управления базами данных (СУБД) существуют различные инструменты для оптимизации запросов. Например, в «Руководстве по проектированию и настройке производительности» Oracle описаны основные средства, которыми может пользоваться разработчик: Explain Plan, SQL_TRACE, TKPROF, Autotrace и Statspack (пакет сбора статистики). В частности, команда Explain plan служит для получения плана выполнения запроса – последовательности операций, необходимых для получения результата SQL-запроса [3]. Однако для настройки производительности от администратора БД требуются очень большие знания и умения не только подключать названные средства, но и разбирать полученные с их помощью отчеты, а также выполнять анализ полученных данных.

Один из наиболее распространенных, естественных и доступных способов повышения производительности ИС – индексирование таблиц БД. Индекс – это структура данных, которая помогает СУБД быстрее обнаруживать отдельные записи в файле, а потому позволяет сократить время выполнения запросов пользователей [4]. Для оптимальной производительности ИС индексы обычно создаются на тех столбцах таблицы, которые часто используются в запросах. Для одной таблицы могут быть созданы несколько индексов. Однако увеличение числа индексов замедляет операции добавления, обновления, удаления строк таблицы, поскольку при этом приходится обновлять сами индексы. Кроме того, индексы занимают дополнительный объем памяти, поэтому перед созданием индекса следует убедиться, что планируемый выигрыш в производительности запросов превысит дополнительную затрату ресурсов компьютера на сопровождение индекса [5, 6]

Существует множество автоматизированных средств проектирования и документирования баз данных. Например, система ERwin позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных, визуализировать структуру данных, обеспечивая эффективный процесс организации и управления базами данных и средой развертывания [7]. Одним из инструментов систем автоматизированного проектирования БД является возможность работы с индексами. В частности, ERwin при генерации схемы на основе модели данных автоматически создает индекс для пер-

вичного ключа и отдельный индекс для каждого альтернативного ключа и внешнего ключа. ERwin автоматически присваивает все атрибуты ключа индексу. После того как индекс создан, можно изменить его характеристики в редакторе Index – имя, определение (уникальные или дублирующиеся значения), изменить порядок сортировки данных. Однако ERwin не позволяет создать новый индекс на основе первичного ключа или внешнего ключа. В редакторе Index нельзя поменять местами колонки индекса по первичному ключу или внешнему ключу. Потребуется изменить порядок расположения атрибутов первичного ключа в соответствующем редакторе [8].

Как правило, утилиты и средства для работы с запросами и индексами не дают единой картины для выбора лучшего варианта из различных индексов таблиц БД относительно различных запросов к ИС. Кроме того, зачастую утилиты позволяют работать только с конкретной СУБД.

Очевидно, что с практической точки зрения удобно иметь универсальное программное средство, с помощью которого можно выполнить сравнение скорости выполнения запросов к БД в случае применения различных вариантов индексации.

1. Постановка задачи исследования

Индексы удобны для выполнения следующих трех действий: сортировка, поиск, поддержка уникальности атрибута (группы атрибутов). Однако при обновлении данных в таблице индексы также требуют обновления, что может привести к дополнительным затратам времени [9].

СУБД поддерживают различные виды индексов: кластеризованные и некластеризованные, простые и составные, уникальные и неуникальные и т.д.

Задача разработчика выбрать для каждой таблицы индексы, наиболее эффективные с точки зрения производительности ИС в целом. В литературе можно встретить советы, для каких атрибутов стоит создавать индексы [5]. При этом следует анализировать структуру таблицы, возможное содержимое атрибутов, объем таблиц и запросы, поступающие к БД.

В данной работе предложена имитационная модель, которая позволяет провести исследование поведения ИС. Входными данными являются структура БД и множество запросов. Выходными данными служат рекомендации по созданию индексов для формирования оптимальной индексной структуры БД. Подключение к БД выполняется с использованием универсальных средств, что позволяет применить разработанный продукт к различным СУБД.

2. Представление входных и выходных данных имитационной модели

Представим исследуемую БД в виде множества кортежей

$$D =< R, I >,$$

где R – таблица БД; I – множество индексов таблицы R.

В свою очередь каждый индекс множества I также можно представить в виде кортежа

$$I =<< p, s >, VI >,$$

где p – атрибут таблицы R; s – порядок сортировки для атрибута p в текущем индексе; VI – тип индекса.

Множество запросов Q представлено с точки зрения времени их выполнения при наличии определенных индексов

$$Q =< qt, < I_i, \tau >>$$

где qt – текст запроса на языке SQL; I_i – i-й индекс множества I; τ – время выполнения запроса qt при условии существования индекса I_i.

Пусть D^A – БД со всеми возможными вариантами индексации атрибутов ее таблиц, D^E – БД с наиболее эффективными вариантами индексов для текущего набора запросов Q.

В результате имитационного моделирования требуется найти такое множество $I^E \subseteq I^A$, при котором достигается максимальная производительность ИС для исследуемого множества запросов.

3. Методика решения задачи поиска оптимальной индексной структуры

Для получения эффективного индекса может потребоваться просмотр различных комбинаций элементов множеств: запросов, таблиц, набора индексов, порядка атрибутов в каждом индексе, порядка сортировки в атрибутах индекса, типа индекса. Такой набор циклов включен в алгоритм имитационной модели (рис. 1).

После формирования статистических данных о времени выполнения запросов из множества Q для каждой таблицы $R_k \subset R$ требуется выбрать такой набор индексов $I^E(R_k)$, для которого выполняется следующее условие

$$\sum_{j=1}^{|Q|} \tau_{ij} = \min(\sum_{j=1}^{|Q|} \tau_{ij}). \qquad (1)$$

Если условию (1) удовлетворяют несколько наборов индексов, то при выборе оптимальной индексной структуры используются следующие правила:

– кластеризованный индекс имеет приоритет по сравнению с некластеризованным;

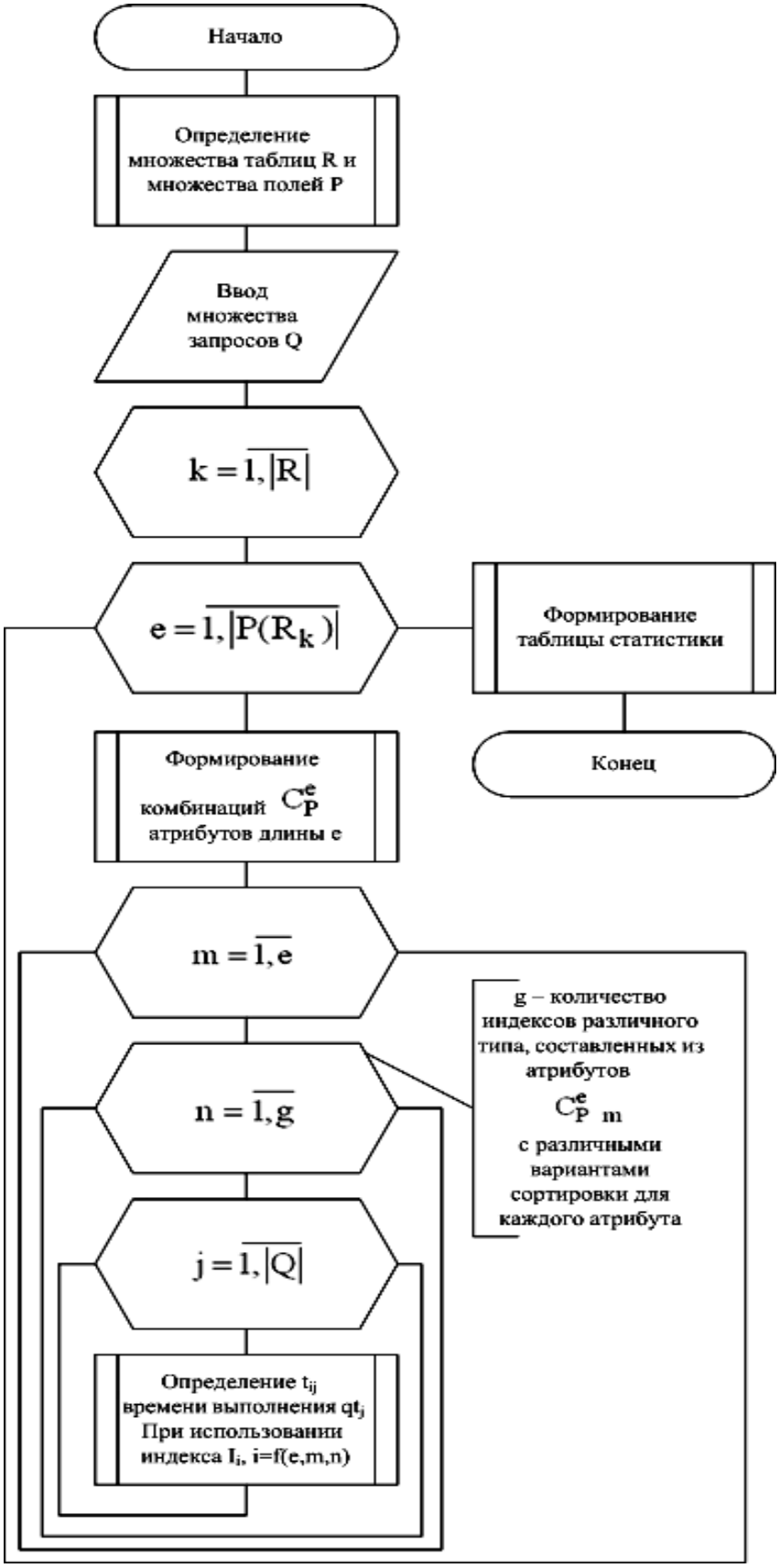

Рис.1. Алгоритм формирования статистических данных

– набор простых индексов для множества атрибутов из C_P^e имеет приоритет по сравнению с составным индексом для этого же набора атрибутов.

Работа алгоритма может быть усовершенствована заменой множества R на множество $R^Q \subseteq R$, в R^Q включаются только те таблицы, которые используются в запросах из множества Q.

В целом имитационное моделирование включает следующие этапы (рис. 2):

Рис. 2. Общая схема работы имитационной модели

– задание параметров для подключения к БД;
– доступ к метаданным;
– генерация различных наборов индексов;
– сохранение статистической информации в сервисной БД;
– формирование рекомендаций по получению оптимальной индексной структуры;
– согласование рекомендаций с пользователем;
– модификация индексной структуры БД.

Каждая СУБД имеет собственный встроенный оптимизатор запросов, который может принимать решение об использовании или игнорировании отдельных индексов. При этом постоянно должно поддерживаться актуальное состояние индекса, при любом изменении данных в таблице БД. Очевидно, что в процессе эксплуатации БД количественный и качественный состав данных может изменяться. Следовательно, может измениться план выполнения запросов и решение СУБД о применении индексов. Требуется периодическая настройка индексной структуры. Предложенная имитационная модель позволяет быстро получать рекомендации по созданию наиболее эффективных индексов в соответствии с текущим состоянием БД.

4. Реализация имитационной модели

Программная система обладает следующими характеристиками:

– определение метаданных с использованием подключения к БД средствами универсального доступа к данным;
– направление запросов к БД с определением продолжительности их выполнения;
– формирование зависимостей между использованным индексом и временем выполнения запроса и их представление в табличном и графическом виде;
– эргономичный интерфейс пользователя;
– квалификация пользователя: пользователь должен иметь минимальный опыт работы с системами, которые имеют графический интерфейс, схожий с интерфейсом Winows XP и выше, и минимальные знания в администрировании СУБД (рис. 3).

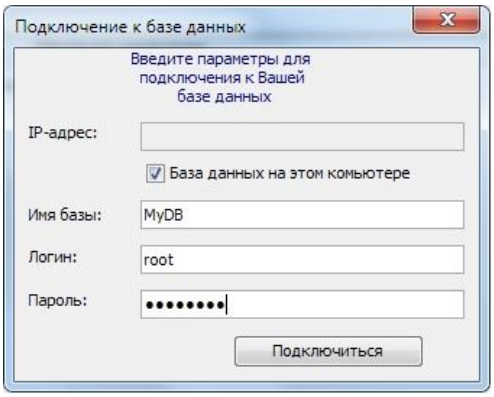

Рис. 3. Окно подключения к БД

Результатом работы программы является таблица, содержащая время выполнения каждого запроса из множества входящих запросов для каждой таблицы, участвующей в запросе (рис. 4).

Полученная информация сохраняется в специально разработанной сервисной БД и может быть просмотрена пользователем в любой момент времени.

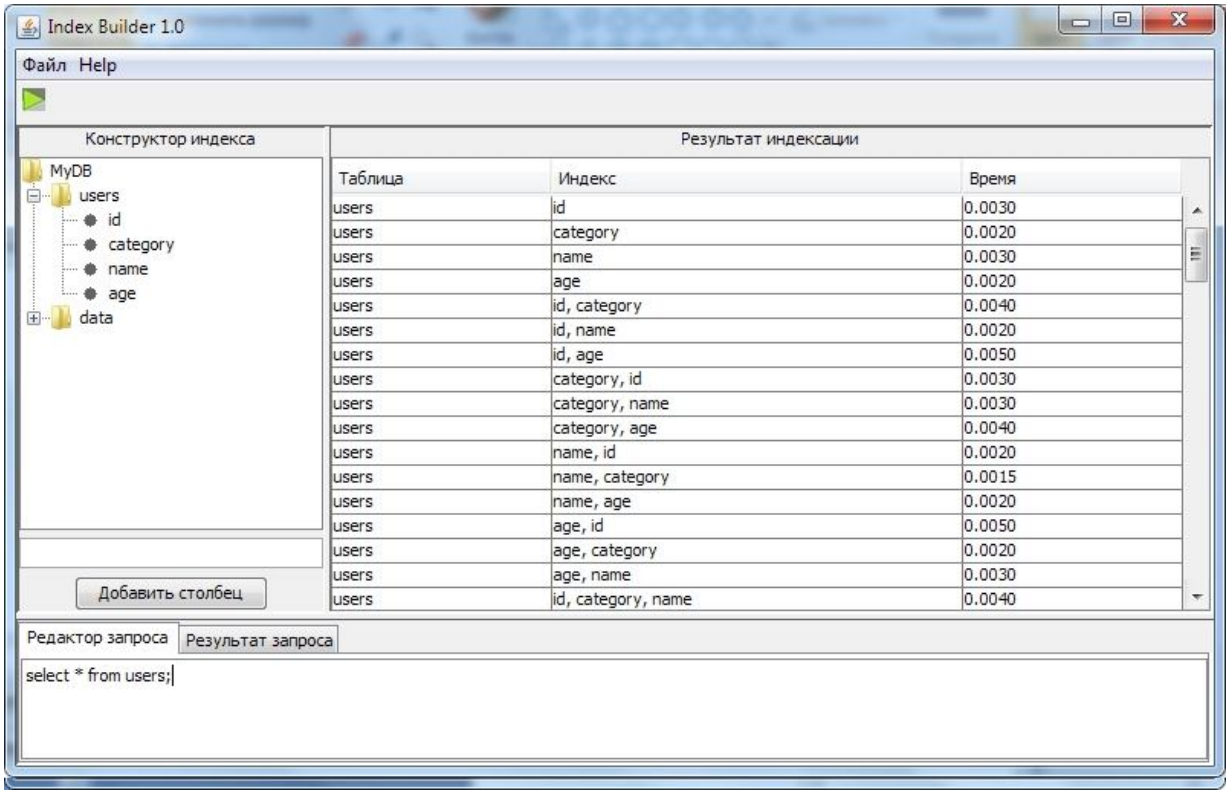

Таблица	Индекс	Время
users	id	0.0030
users	category	0.0020
users	name	0.0030
users	age	0.0020
users	id, category	0.0040
users	id, name	0.0020
users	id, age	0.0050
users	category, id	0.0030
users	category, name	0.0030
users	category, age	0.0040
users	name, id	0.0020
users	name, category	0.0015
users	name, age	0.0020
users	age, id	0.0050
users	age, category	0.0020
users	age, name	0.0030
users	id, category, name	0.0040

Рис. 4. Числовые результаты работы программы

Заключение

Рассмотрена задача выбора оптимального индекса для таблицы БД с точки зрения скорости выполнения конкретных запросов.

Предложена имитационная модель для автоматизированного поиска наилучших индексов с точки зрения производительности ИС.

Разработана модель для формализованного представления индексов БД.

Применение предлагаемого подхода обеспечивает решение следующих задач: применение подхода для большого множества различных СУБД; сокращение времени, затраченного разработчиком БД на выбор оптимального индекса; наглядное представление результатов применения различных индексов по отношению к множеству запросов; прозрачный механизм определения оптимального набора индексов.

Выполнена программная реализация построения эффективной индексной структуры на основе сбора и анализа статистики, полученной в результате имитационного моделирования ИС.

Литература

1. Зиноватная, С.Л. Анализ методов повышения производительности информационных систем [Текст] / С.Л. Зиноватная // Моделирование в прикладных научных исследованиях : XII семинар, 19 – 20 янв. 2005 г. – Одесса, 2005. – С. 41 – 43.

2. Левченко, О. Засоби автоматизації процесу поліпшення характеристик продуктивності СКБД [Текст] / О. Левченко, О. Блажко, Алхашмі Яссер Маруан // Комп'ютерні науки та інформаційні технології: міжн. наук.-техн.конф., 15-17 жов., 2009 р. : тези допов. – Львів, 2009. – С. 305 – 307.

3. МакДональд, К. Oracle PL/SQL для профессионалов: практические решения [Текст] / К. МакДональд, Х. Кац, К. Бек, Дж. Кальман, Д. Нокс. – К. : ДиаСофт, 2005. – 560 с.

4. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика[Текст] / Т. Коннолли, К. Бегг, А. Страчан. – М.: Изд. дом "Вильямс", 2002. – 1120 с.

5. Индекс (базы данных). Материал из Википедии – свободной энциклопедии [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Индекс_(базы данных) - 14.02.2013 г.

6. Database Systems: The Complete Book [Текст]; 2-е изд. / Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom. – Prentice Hall, 2008. — 1248 p.

7. Официальный сайт CA ERwin® Process Modeler [Электронный ресурс]. – Режим доступа: http://erwin.com/products/data-modeler- 04.03.2013 г.

8. Руководство по программному пакету ERwin [Электронный ресурс]. – Режим доступа: http://www.xserver.ru/computer/database/erwin/2/15.shtml - 04.03.2013 г.

9. Кренке, Д. Теория и практика построения баз данных. 8-е изд. [Текст] / Д. Кренке. – СПб.: Питер, 2003. – 800 с.

Поступила в редакцию 14.02.2013, рассмотрена на редколлегии 13.03.2013

Рецензент: канд. техн. наук, проф., проф. каф. системного программного обеспечения А.Б. Кунгурцев, Одесский национальный политехнический университет, Одесса.

ІМІТАЦІЙНА МОДЕЛЬ ДЛЯ ПРОЕКТУВАННЯ ОПТИМАЛЬНОЇ ІНДЕКСНОЇ СТРУКТУРИ БАЗИ ДАНИХ

С.Л. Зіноватна, О.Ю. Левченко, К.О. Галанюк

Запропоновано метод формування оптимальної індексної структури бази даних з метою підвищення продуктивності інформаційної системи, що використовує цю базу даних. Представлено формальний опис індексної структури таблиць із урахуванням поводження системи, вираженого множиною запитів. Розроблено алгоритм формування набору індексів-кандидатів для одержання найменшого часу виконання запитів в інформаційній системі. Пошук оптимальної структури індексів виконується з використанням імітаційної моделі поводження інформаційної системи в умовах застосування різних наборів індексів.

Ключові слова: імітаційна модель, індекс, оптимізація індексної системи, продуктивність інформаційної системи, база даних, SQL.

THE SIMULATION MODEL FOR DESIGNING THE OPTIMAL INDEX STRUCTURE OF THE DATABASE

S.L. Zinovatna, O.Yu. Levchenko, K.O. Halaniuk

It is proposed a method to design the optimal index structure of the database to improve the performance of the information system that uses this database. The formal description of the index structure of the tables with system behavior is represented by query set. It is developed the algorithm to form a set of the candidate indexes to minimize the query execution in the information system. The optimal index structure search is performed by simulating the information system behavior using the different sets of index

Key words: simulation model, index, optimization of index system, performance of information system, database, SQL.

Зиноватная Светлана Леонидовна – канд. техн. наук, доцент каф. системного программного обеспечения, Одесский национальный политехнический университет, Одесса, Украина, e-mail: svzino@rambler.ru.

Левченко Александра Юрьевна – канд. техн. наук, ассистент каф. системного программного обеспечения, Одесский национальный политехнический университет, Одесса, Украина.

Галанюк Константин Александрович – бакалавр каф. системного программного обеспечения, Одесский национальный политехнический университет, Одесса, Украина.