Математические структуры и моделирование
2007, вып. 17, с. 100–107

УДК 004.657

ОПТИМИЗАЦИЯ ЗАПРОСОВ К БАЗАМ ДАННЫХ

О.В. Пашинин

Рассматривается возможность вывода текущих запросов к базе данных из множества уже обработанных запросов

1. Введение

В настоящее время бурно развивается Интернет и Интранет, а также технологии с ними связанные. Корпоративные сети охватывают даже самые отдаленные производственные участки предприятия. Пропускная способность каналов связи до некоторых структурных подразделений недостаточна для корректной и адекватной работы части программных продуктов, построенных на технологии «клиент-сервер». В большинстве своем эти программные продукты так или иначе касаются баз данных. В данной ситуации чаще всего проблема решается либо за счет установки дополнительного сервера в удаленном структурном подразделении и пользователь работает именно с этим сервером, либо за счет увеличения пропускной способности канала передачи данных, что не всегда экономически и технологически оправдано.

Проблему можно решить и другим способом. Во-первых, вместо дополнительного сервера в удаленном структурном подразделении ставится рабочая станция (будем называть ее – буфер), которая хранит результаты последних запросов. Во-вторых, оптимизируется запрос к базе данных.

Вопросу оптимизации запросов посвящено множество статей и обзоров [1,2,6]. Можно выделить два основных метода оптимизации – статистический и алгебраический. Статистический метод базируется на системе оценок, статистике базы данных и допущения модели. Применение различных эвристик сужает пространство поиска, и выбирается оптимальный план выполнения запроса [7]. Алгебраический метод основан на применении к запросу операций реляционной алгебры и математической логики, благодаря чему на выходе получается эквивалентный канонический запрос [4, 8].

Используя алгоритм, предложенный в этой статье, сначала проверяется возможность вывода текущего запроса из предыдущих запросов, хранящихся в буфере и обработанных за данное время, и только в случае невозможности вывода запрос полностью или частично передается на сервер, находящийся на головном предприятии.

Copyright © 2007 **О.В. Пашинин.**
Омский государственный университет.
E-mail: olegpashinin@mail.ru

2. Определения и обозначения

Рассматриваемый в данной статье подход предполагает использование стандартного запроса $S = \pi_X(\sigma_F(R_1 \bowtie R_2 \bowtie ... \bowtie R_n))$. Заполнение памяти буфера происходит следующим образом. Пользователь посылает запрос к головному серверу. Этот запрос идет к серверу не напрямую, а через буфер. Будем считать, что на буфере установлен программный модуль (ПМ), который делит поступающий запрос на две части: $S_1 = \sigma_F(R_1 \bowtie R_2 \bowtie ... \bowtie R_n)$ (редуцированный запрос) и $S_2 = \pi_X(S_1)$. Запрос S_1 отправляется на головной сервер, а запрос S_2 выполняется средствами ПМ. В результате для каждого запроса S в буфере помимо результата выполнения самого запроса S хранится таблица, соответствующая запросу S_1.

Также положим, что логическое выражение в запросе приведено к ДНФ. Для каждого запроса S алгоритм применяется последовательно к каждому дизъюнкту в логическом выражении F.

Например, $F = \{A_1 = x \vee (A_2 = y \wedge A_3 < z)\}$. Тогда алгоритм применяется сначала к $S_1^1 = \sigma_{A_1=x}(R)$, а затем к $S_1^2 = \sigma_{A_2=y \wedge A_3<z}(R)$.

Если результатом работы алгоритма является возможность вывода текущего запроса из предыдущих полностью или частично, то к множеству таблиц на выходе алгоритма необходимо применить аппарат математической логики и реляционной алгебры.

Введем следующие обозначения:

S – запрос на момент времени t к базе данных (DB).

$F(S)$ – логическое выражение в запросе S.

$r = S(DB)$ – таблица (результат выполнения запроса).

$S' = \{S_1, S_2, ..., S_k\}$ – множество запросов, находящихся в системном буфере на момент времени t.

$r' = \{r_1, r_2, ..., r_k\}$ – множество таблиц, соответствующих запросам из множества S'.

Сначала рассмотрим случай, когда база данных содержит только одну таблицу $R(A_1, A_2, ..., A_n)$. Далее обобщим рассуждения на случай нескольких таблиц.

Рассмотрим множество запросов S' и ограничим его редуцированными запросами. Т. е. вместо множества $\{\pi_X(\sigma_{F1}(R)), \pi_Y(\sigma_{F2}(R)), ..., \pi_Z(\sigma_{Fk}(R))\}$, где X, Y, Z – подмножества множества атрибутов $\{A_1, A_2, ..., A_n\}$, к которым применяется операция проекции, имеем дело с множеством $s' = \{\sigma_{F1}(R), \sigma_{F2}(R), ..., \sigma_{Fk}(R)\}$.

Введем определения:

Определение 1. Пусть g – булев вектор, соответствующий запросу. Он формируется следующим образом: $g(j) = 1$, если в запросе S при выполнении операции селекции накладывались любые ограничения на j-ый атрибут (отрицание, сравнение, равенство с элементом из множества значений данного атрибута). Вектор g будем называть **характеристическим вектором** для запроса S.

Например, R=(№ чит. билета, назв. книги, назв. изд-ва, дата заказа, № заказа). Запросу $\pi_{A1}(\sigma_{A_2=\text{'Незнайка'}\&A_4=\text{'01.01.2000'}}(R))$ будет соответствовать характеристический вектор $g = (01010)$; запросу $\pi_{A1}(\sigma_{A1<\text{'100'}\&A3=\text{'Наука'}}(R))$ будет соответствовать вектор $g = (10100)$.

Определение 2. Прямоугольную матрицу, составленную из коэффициентов характеристических векторов, назовем **характеристической матрицей**.

Например, $g_1 = (1010)$, $g_2 = (1100)$, $g_3 = (1010)$, $g_4 = (1000)$, $g_5 = (1000)$, тогда

$$G = \begin{matrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix}$$

3. Алгоритм проверки выводимости запроса S из множества S' в случае одной таблицы

Входные данные:

S – текущий запрос;

$F = F(S)$ – логическое выражение в запросе S;

$g = g(S)$ – характеристический вектор запроса S;

$S' = \{S_1, ...S_k\}$ – множество запросов в системном буфере;

$F' = \{F_1, ...F_k\}$ – множество логических выражений, соответствующих запросам в системном буфере, приведенные к ДНФ;

$g' = \{g_1, ..., g_k\}$ – множество характеристических векторов, соответствующих запросам в системном буфере;

$r' = \{r_1, r_2, ..., r_n\}$ – множество таблиц, находящихся в системном буфере на момент времени t;

G' – характеристическая матрица множества характеристических векторов g';

G – вспомогательная матрица такой же размерности, как и G', на начальном этапе она составлена из одних нулей. Область значений элементов $(0, 1, *)$.

Выходные данные:

S'' – множество запросов, из которых возможно вывести S.

Алгоритм:

Ш.1. Если существует F_i такое, что $F_i \equiv F$, то $S'' = \{S_i\}$. *Конец.*

Иначе, разбиваем F на дизъюнкты F^k и переходим на шаг 2 с очередным дизъюнктом.

Ш.2. Рассматриваем вектор g. Смотрим на каком месте стоит очередная единица. Пусть на месте p имеет место $g(p) = 1$ (т.е. на атрибут A_p накладывается условие в логическом выражении $F^k(S)$).

Если непройденных единиц нет, то переходим на шаг 5.

Иначе ищем среди векторов множества g' векторы $g_{t1}, \ldots, g_{tm}$, такие, что $g_{ti}(p) = g(p) = 1$, $i = 1...m$.

Если хотя бы один такой вектор есть, то переходим на шаг 3, иначе переходим к следующей единице.

Ш.3. Если строка матрицы G, соответствующая вектору g_{ti} не помечена, то помечаем ее и копируем в неё соответствующую строку из матрицы G'.

Ш.4. Проверяем справедливость $F^k_{ti}(p) \supseteq F(p)$, $i = 1...m$.

Если условие выполняется, то для данного номера t_i в матрицу G в строку t_i, столбец p ставим звездочку и переходим на шаг 2.

Ш.5. Вычеркиваем нулевые строчки в матрице G, не помеченные на шаге 3, – тем самым мы ограничиваемся запросами, в которых при выполнении селекции не накладывались ограничения на атрибуты $A_{t1}, A_{t2}, ..., A_{tm}$.

Вычеркиваем строчки, в которых стоят «1» там, где у вектора g стоят нули, – тем самым мы исключаем те запросы, в которых были ограничения на другие атрибуты, нежели в S.

Переходим на шаг 6.

Ш.6. Просматриваем полученную матрицу по строкам.

Проверяем для каждой i-ой строки, есть ли строка, состоящая только из $\{0, *\} : \forall p\ G[i, p] \neq 1$.

Если такие строки есть, то запрос S выводим и $S'' = \{S_i\}$. *Конец.*

4. Обобщение алгоритма проверки выводимости запроса S из множества S' в случае n таблиц

Пусть в нашей базе данных n таблиц $R_1, R_2, ..., R_n$.

Тогда запрос S можно представить следующим образом:

$S = \pi_X(\sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk}))$, где $1 \leq k \leq n$.

Опять обозначим через s редуцированный запрос, соответствующий запросу S: $s = \sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk})$.

Определение 3. Пусть дана реляционная база данных DB;

$R_1, R_2, \ldots, R_n$ – таблицы DB;

$S = \pi_X(\sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk}))$– некоторый запрос к DB;

$s = \sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk})$ – редуцированный запрос.

Тогда **расширенным характеристическим вектором** запроса S в DB будем называть булев вектор eg (enhanced – расширенный), формирующийся следующим образом: $eg[(n_1+\ldots+n_{ti-1})+j] = 1$, если в запросе s накладывались ограничения на j-ый атрибут, $2 \leq t_i \leq k$, n_i – размерность i-ой таблицы.

Например, дана DB «библиотека» (R_1, R_2, R_3, R_4).

$R_1 =$(**№ издательства**, наименование изд-ва, адрес изд-ва);

$R_2 =$(**№ книги**, название книги, наименование изд-ва, ФИО автора);

$R_3 =$(ФИО читателя, адрес читателя, **№ чит. билета**);

$R_4 =$(№ чит. билета, № книги, дата, **№ заказа**).

Пронумеровав атрибуты, формально получим:

$R_1 = (A_1, A_2, A_3)$;

$R_2 = (A_4, A_5, A_2, A_6)$;

$R_3 = (A_7, A_8, A_9)$;

$R_4 = (A_9, A_4, A_{10}, A_{11})$.

Запросу $s = \sigma_{A_2=\text{'Мир'}\&A_6=\text{'Иванов'}}(R_1 \bowtie R_2)$ будет соответствовать вектор $eg = (01000100000)$.

Запросу $s = \sigma_{A_2=\text{'Питер'}\&A_5=\text{'Война и мир'}\&A_6=\text{'Л.Толстой'}\&A_9=\text{'27'}}(R_1 \bowtie R_2 \bowtie R_4)$ будет соответствовать вектор $eg = (01001100100)$.

Определение 4. Пусть дана реляционная база данных DB;

$R_1, R_2, \ldots, R_n$ – таблицы DB;

$S = \pi_X(\sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk}))$ – некоторый запрос к DB;

$s = \sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tk})$ – редуцированный запрос.

Тогда **маской** запроса S в DB будем называть булев вектор mg размерности n, построенный следующим образом: $mg[i] = 1$, если в запросе s таблица $R_i \in (R_1, R_2, R_3, R_4)$, и $mg[i] = 0$ в противном случае.

Например, для DB из предыдущего примера:

запросу $s = \sigma_{A2=\text{'Мир'}\&A6=\text{'Иванов'}}(R_1 \bowtie R_2)$ будет соответствовать маска $mg = (1100)$.

Запросу $s = \sigma_{A2=\text{'Питер'}\&A5=\text{'Война и мир'}\&A6=\text{'Л.Толстой'}\&A9=\text{'27'}}(R_1 \bowtie R_2 \bowtie R_4)$ будет соответствовать маска $mg = (1101)$.

Добавим к алгоритму для одной таблицы 0-шаг:

$$s = \sigma_F(R_{t1} \bowtie R_{t2} \bowtie ... \bowtie R_{tm}).$$

Ищем среди запросов множества

$$s' = \{s_1, s_2, \ldots, s_k\} = \{\sigma_{F1}(R_{11} \bowtie \ldots \bowtie R_{1p}), \ldots, \sigma_{Fk}(R_{k1} \bowtie \ldots \bowtie R_{kq})\}$$

те запросы, для которых выполняется следующее: маска этих запросов должна быть идентична маске запроса s.

Обозначим $R = R_{t1} \bowtie R_{t2} \bowtie \ldots \bowtie R_{tm}$ и в алгоритм входим с этой таблицей R. Вместо вектора g в алгоритме используем расширенный вектор eg.

5. Пример работы алгоритма

Рассмотрим пример работы алгоритма с одной таблицей.

На входе имеем:

- Таблицу $R(A_1, A_2, A_3, A_4, A_5)$.
- R =(№ чит. билета, назв. книги, назв. изд-ва, дата заказа, № заказа).
- Текущий запрос S:

$$S = \pi_{A1}(\sigma_{A2=\text{'Незнайка'}\&A4=\text{'01.03.2000'}}(R)).$$

- Характеристический вектор g запроса S: $g = (01010)$.

- Логическое выражение $F = F(S)$:

  $F = (A_2 =$'Незнайка'$\& A_4 =$'01.01.2000').

- Множество запросов в системном буфере $S' = \{$

  $S_1 = \pi_{A1}(\sigma_{A2=\text{'Незнайка'}\& A4>\text{'02.02.2000'}}(R))$;

  $S_2 = \pi_{A2}(\sigma_{A1=\text{'1710'}}(R))$;

  $S_3 = \pi_{A5}(\sigma_{A2=\text{'Мишкина каша'}\& A4=\text{'01.03.2000'}}(R))$;

  $S_4 = \pi_{A4}(\sigma_{A2=\text{'О, женщины...'}\& A3=\text{'Новый мир'}}(R))$;

  $S_5 = \pi_{A2}(\sigma_{A3=\text{'Просвещение'}\& A4=\text{'01.01.2000'}}(R))$;

  $S_6 = \pi_{A2}(\sigma_{A5=\text{'211'}}(R))\}$.

- Множество характеристических векторов $g' = \{g_1 = (01010), g_2 = (10000), g_3 = (01010), g_4 = (01100), g_5 = (00110), g_6 = (00001)\}$.

- Матрица G':

$$G' = \begin{matrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

- Множество логических выражений $F' = \{F_1, \ldots F_6\}$:

  $F' = \{$

  $(A_2 =$'Незнайка'$\& A_4 >$'02.02.2000');

  $(A_1 =$'1710');

  $(A_2 =$'Мишкина каша'$\& A_4 =$'01.03.2000');

  $(A_2 =$'О, женщины...'$\& A_3 =$'Новый мир');

  $(A_3 =$'Просвещение'$\& A_4 =$'01.01.2000');

  $(A_5 =$'211'$)\}$.

Ш.1. Нет таких F_i, что $F_i \equiv F$, $i = 1 \ldots 6$. Переходим на ш. 2.

Ш.2. Очередная единица стоит на месте № 2. Находим векторы g_1, g_3, g_4, у которых вторая координата равна «1». Переходим на ш. 3.

Ш.3. Имеем матрицу G:

$$G = \begin{matrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Переходим на ш. 4.

Ш.4.

$F_1(2) = \{A_2 =\text{'Незнайка'}\} \supseteq F(2)$;

$F_3(2) = \{A_2 =\text{'Мишкина каша'}\} \not\supseteq F(2)$;

$F_4(2) = \{A_2 =\text{'О, женщины'}\} \not\supseteq F(2)$.

Имеем матрицу G:

$$G = \begin{matrix} \mathbf{0} & * & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Переходим на шаг 2.

Ш.2. Очередная единица стоит на месте № 4 Находим векторы g_1, g_3, g_5, у которых четвертая координата равна «1». Переходим на ш. 3.

Ш.3. Имеем матрицу G:

$$G = \begin{matrix} \mathbf{0} & * & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Переходим на ш. 4.

Ш.4.

$F_1(4) = \{u[4] >\text{'02.02.2000'}\} \supseteq F(4)$;

$F_3(4) = \{u[4] =\text{'01.03.2000'}\} \supseteq F(4)$;

$F_5(4) = \{u[4] =\text{'01.01.2000'}\} \not\supseteq F(4)$.

Имеем матрицу G:

$$G = \begin{matrix} \mathbf{0} & * & \mathbf{0} & * & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & * & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Переходим на ш. 2.

Ш.2. Больше непройденных единиц нет. Переходим на ш. 5.

Ш.5. Вычеркиваем нулевые строчки и строчки, в которых стоят единицы в столбцах, отличных от № 2 и № 4.

Имеем матрицу G:

$$G = \begin{matrix} \mathbf{0} & * & \mathbf{0} & * & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & * & \mathbf{0} \end{matrix}$$

Переходим на ш. 6.

Ш.6. Получаем набор строк $\{1, 3\}$. Первая строка состоит только из $\{0, *\}$. Значит, из запроса № 1, находящегося в буфере, можно вывести текущий запрос.

Конец.

Литература

1. Кузнецов С.Д. Методы оптимизации выполнения запросов в реляционных СУБД / Тем. изд. «Итоги науки и техники. Вычислительные науки». Т.1. Москва, ВИНИТИ, 1989, С.76-153
2. Чаудхари С. Методы оптимизации запросов в реляционных системах // СУБД. 1998. N.3. C.22-36.
3. Чери С., Готлоб Г. Логическое программирование и базы данных. Пер с англ. М.: Мир, 1992.
4. Latha S.Colby. A recursive algebra and query optimization for nested relations // ACM SIGMOD Record archive. 1989. V.18, N.2. P.273-283.
5. Yannis E. Ioannidis and Raghu Ramakrishnan. Containment of Conjunctive Queries: Beyond Relations as Sets // ACM Trans. Database Syst. 1995. V.20, N.3. P.288-324.
6. Zykin S.V. Relation queries execution under the estimators control // International conference ADBIS-95. Moscow. 1995. V.2 P.52-55.
7. Sun W., Yu C. Semantic Query Optimization for Tree and Chain Queries // IEEE Transactions on Knowledge and Data Engineering. 1994 V.6, N.1 P.136-151.
8. Maher M., Wang J. Optimizing Queries in Extended Relational Databases // Proceedings of the 11th International Conference on Database and Expert Systems Applications. 2000. September 04-08. P.386-396.