

# Оптимизация SELECT-запросов с агрегатной функцией в коррелированном подзапросе в СУБД ЛИНТЕР

Д. В. Борисенков, email: xuser@relex.ru

Воронежский государственный университет

***Аннотация.** В данной работе рассмотрены SELECT-запросы специального типа и оптимизационные преобразования для них, которые были реализованы в компиляторе языка SQL СУБД ЛИНТЕР.*

***Ключевые слова:** СУБД ЛИНТЕР, оптимизация SQL-запросов, коррелированный подзапрос, агрегатная функция, группировка.*

## Введение

Необходимость реализации описанного в данной статье оптимизационного преобразования SELECT-запросов была выявлена при анализе проблемы с производительностью работы приложения, созданного некоторой организацией – пользователем СУБД ЛИНТЕР [1]. Данное приложение было портировано с СУБД Oracle на СУБД ЛИНТЕР в рамках программы импортозамещения, при этом первоначально скорость работы приложения с СУБД ЛИНТЕР оказалась примерно 3 раза ниже, чем с СУБД Oracle. В процессе анализа проблемы было обнаружено, что указанное приложение выполняет большое количество однотипных SELECT-запросов с агрегатными функциями в коррелированных подзапросах, причем планы выполнения этих запросов в СУБД ЛИНТЕР являются явно не оптимальными. Дальнейший анализ показал возможность оптимизационного преобразования таких запросов компилятором языка SQL, при котором коррелированный подзапрос с агрегатной функцией становится группированным, перемещается из конструкции WHERE в конструкцию FROM и перестает быть коррелированным. Реализация автоматического выполнения этого преобразования обеспечила примерно такую же производительность работы приложения с СУБД ЛИНТЕР, как и с СУБД Oracle.

Далее описаны: структура оптимизируемых SELECT-запросов, ограничения на выполнение данной оптимизации, сущность производимого преобразования запроса.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00037.

## 1. Структура оптимизируемых SELECT-запросов

В упомянутом приложении встречалось несколько десятков различных SELECT-запросов, имеющих следующую однотипную структуру:

```
select ...
from TABLE1, ...
where TABLE1.COLUMN1 vs
      (select aggr(TABLE2.COLUMN2)
       from TABLE2
       where TABLE1.COLUMN3 = TABLE2.COLUMN4 and ... )
and ... ;
```

где

vs – операция сравнения (=, <>, >=, <=, >, <),  
aggr – агрегатная функция (MIN, MAX, AVG, SUM),

Запрос такого типа содержит коррелированный подзапрос (внешняя ссылка – TABLE1.COLUMN3) с агрегатной функцией, и результат этого подзапроса вычисляется для каждой записи таблицы TABLE1. При большом количестве записей в таблице TABLE1 накладные на циклическое выполнение одного и того же подзапроса с разными значениями TABLE1.COLUMN3 приводят к многократному замедлению выполнения основного запроса.

Направивается следующее преобразование исходного запроса, смысл которого в том, чтобы запрос с агрегатной функцией выполнялся только один раз – с агрегацией по группам:

```
select ...
from TABLE1, ...,
      (select aggr(TABLE2.COLUMN2) AGGR_V,
       TABLE2.COLUMN4
       from TABLE2
       where ...
       group by TABLE2.COLUMN4) as QUERY
where TABLE1.COLUMN3 = QUERY.COLUMN4 and
      TABLE1.COLUMN1 vs QUERY.AGGR_V and ...;
```

Выигрыш от использования этой оптимизации тем больше, чем меньше селективность столбца COLUMN4 таблицы TABLE2.

## 2. Требования к запросу для возможности выполнения оптимизации

Для выполнения оптимизационного преобразования к запросу предъявляются перечисленные ниже требования. Этот набор требований

должен обеспечивать эквивалентность выполняемых преобразований и упрощение выполняемых действий.

1. Основной запрос должен быть SELECT-запросом.
2. В состав основного запроса должен входить коррелированный подзапрос (т.е. содержащий внешнюю ссылку).
3. Коррелированный подзапрос должен иметь только один источник данных в конструкции FROM
4. Коррелированный подзапрос должен иметь только один столбец в конструкции SELECT и этот столбец должен быть результатом агрегатной функции, отличной от функции COUNT.
5. Условие WHERE коррелированного подзапроса должно представлять собой одиночный предикат либо группу предикатов, соединенных операциями AND. В случае наличия логических операций OR они должны быть перенесены на нижний уровень, т.е. конструкция WHERE должна быть представлена в конъюнктивной нормальной форме.
6. Подзапрос, непосредственно включающий в себя коррелированный подзапрос, должен представлять собой внутреннее соединение.

### **3. Оптимизационное преобразование запроса**

Коррелированный подзапрос (внутренний) и подзапрос, непосредственно включающий его (внешний), преобразуются следующим образом:

1. Для всех столбцов таблицы-источника внутреннего подзапроса, к которым есть обращения в его WHERE-предикатах с внешними ссылками, добавляется группировка по этим столбцам.
2. Все эти группированные столбцы добавляются в список столбцов результата внутреннего подзапроса.
3. Все предикаты, которые содержат внешние ссылки, из условия WHERE внутреннего подзапроса переносятся в условие WHERE внешнего подзапроса (при этом по возможности создается группа предикатов, соединенных по AND, или эти предикаты добавляются в группу на верхнем уровне). При этом обращения к столбцам таблицы-источника исходного коррелированного подзапроса заменяются на обращения к столбцам преобразованного группированного подзапроса.
4. Полученный группированный подзапрос добавляется в конструкцию FROM внешнего подзапроса, а ссылка на него из конструкции WHERE внешнего подзапроса заменяется на

ссылку на столбец группированного подзапроса – результат агрегатной функции.

5. При этом учитывается, что внешние ссылки в перенесенных предикатах могут перестать быть внешними ссылками (а могут и остаться ими, если они ссылаются на еще более внешние подзапросы).

Дополнительно могут быть выполнены и другие сопутствующие изменения запроса. Например, если предикат с внешней ссылкой представляет собой равенство столбцов внешнего и внутреннего подзапроса, и есть еще предикат равенства того же столбца внешнего подзапроса константе, то можно заменить последний на предикат равенства соответствующего столбца внутреннего подзапроса константе и перенести его во внутренний подзапрос.

#### 4. Тестирование

При тестировании сделанных правок были проверены, в частности, следующие ситуации:

1. в потенциально оптимизируемом подзапросе есть внешние ссылки к нескольким разным таблицам;
2. в потенциально оптимизируемом подзапросе есть предикат с подзапросом (с внешней ссылкой или без нее);
3. внешняя ссылка в потенциально оптимизируемом подзапросе идет более чем через один уровень;
4. внешняя ссылка стоит не в предикате равенства, а в предикате сравнения другого типа.

#### 5. Примеры оптимизационного преобразования

Запросы содержат обращение к одной таблице из книги К.Дж.Дейта [2], используется одна таблица – таблица поставщиков S.

```
create table S (  
    SNUM char(5) primary key,  
    SNAME char(20),  
    STATUS integer,  
    CITY char(15));
```

Задание: найти всех поставщиков, статус которых является максимальным из всех поставщиков того же города.

```
select *  
from S S1  
where STATUS =  
    (select max(STATUS)  
     from S S2  
     where S1.CITY = S2.CITY);
```

Преобразованный запрос имеет такой вид:

```
select S1.*
from S S1,
      (select max(STATUS) MAXST, CITY
       from S S2
       group by CITY) S3
where S1.CITY = S3.CITY and S1.status = S3.MAXST;
```

Задание: найти всех поставщиков, статус которых является максимальным из всех поставщиков того же города со статусом меньше 30.

```
select * from S S1
where STATUS =
      (select max(STATUS)
       from S S2
       where S1.CITY = S2.CITY and S2.STATUS < 30);
```

Преобразованный запрос имеет такой вид:

```
select S1.*
from S S1,
      (select max(STATUS) MAXST, CITY
       from S S2 where S2.STATUS < 30
       group by CITY) S3
where S1.CITY = S3.CITY and S1.status = S3.MAXST;
```

### **Заключение**

Выполненная в компиляторе SQL описанная оптимизация SELECT-запросов указанного типа помогла обеспечить нужную производительность выполнения запросов в СУБД ЛИНТЕР и таким образом удовлетворить требованиям пользователя.

### **Список литературы**

1. Рекомендации по настройке СУБД. ЛИНТЕР. [Электронный ресурс] : электронная документация по СУБД ЛИНТЕР. – Режим доступа: <http://www.linter.ru/ru/documentation/information/pdf/rec.pdf>
2. Дейт К.Дж. Введение в системы баз данных. / К. Дж. Дейт. – 8-е изд. – М.: «Вильямс», 2006. – 1328 с.