

ОПТИМИЗАЦИЯ ЗАПРОСОВ В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

OPTIMIZATION OF QUERY
IN RELATIONAL DATABASES

I. Kutsenko

Summary. This article considers statistical method for optimizing the query of relational databases. Methods for converting the current query into more efficient form using heuristic rules.

Keywords: database, query optimization, heuristic rules, query tree, relational algebra operations, canonical query tree.

Куценко Ирина Львовна

К.ф.-м.н., доцент, РУДН

i.kutsenko@mail.ru

Аннотация. В данной статье рассматривается статистический метод оптимизации запросов реляционных баз данных. Приведены способы преобразования текущего запроса в заведомо более эффективную форму с помощью эвристических правил.

Ключевые слова: база данных, оптимизация запросов, эвристические правила, дерево запроса, операции реляционной алгебры, каноническое дерево запроса.

Существуют два основных подхода к оптимизации запросов — статистический и алгебраический. Статистический метод основан на применении различных эвристик, с помощью которых можно сузить пространство поиска и выбрать оптимальный план выполнения запроса. Алгебраический метод основан на применении к запросу операций реляционной алгебры и математической логики, благодаря чему на выходе получается эквивалентный канонический запрос.

Эвристические методы
оптимизации запросов

Рассмотрим задачу минимизации количества операций, необходимых для обработки запроса z при заданных размерах таблиц базы данных:

$$N_z(n, a_1, a_2, \dots, a_m) \rightarrow \min, \quad (1)$$

где N_z — количество операций в дереве реляционной алгебры, $a_1, a_2, \dots, a_m > 0$ — заданные параметры, характеризующие информационные стоимости операций.

Для решения задачи (1) можно использовать необходимые и достаточные условия экстремума функции нескольких переменных [1]. Однако, стоит отметить, что в данном случае целесообразнее применять эвристические методы, которые заключаются в использовании правил преобразования выражений, написанных на языке реляционной алгебры в некоторую эквивалентную форму, обработка которой будет более понятной и эффективной. К примеру, эффективнее произвести операции выборки данных, удовлетворяющих заданному условию (условиям) из отношений прежде, чем проводить связи между отношениями. Данные

правила будут использоваться для реструктуризации канонического дерева реляционной алгебры. Рассмотрим правила преобразования операций реляционной алгебры более подробно.

- a) Операция выборки с конъюнктивным предикатом можно преобразовать в последовательность операций выборки по членам конъюнкции (и обратно).
- b) Правило коммутативности операций выборки.
- c) В последовательности операций проекции необходима только последняя из операций.
- d) Правило коммутативности операций выборки и проекции.
- e) Правило коммутативности операции тета-соединения и прямого произведения.
- f) Правило коммутативности операций выборки и тета-соединения иди прямого произведения.
- g) Правило коммутативности операций проекции и тета-соединения.
- h) Правило коммутативности операций объединения и пересечения (но не разности множеств).
- i) Правило коммутативности операции выборки и операций над множествами (\cap, \cup, \setminus).
- j) Правило коммутативности операций проекции и объединения.
- k) Правило ассоциативности операции тета-соединения.
- l) Правило ассоциативности операций объединения и пересечения [2].

Пусть имеются три отношения R_1 — Гости, R_2 — Бронирование, R_3 — Отели, которые соответственно содержат информацию о гостях, операциях бронирования и информацию об отелях. Сформулируем запрос, который подыскивает для гостя определенного отеля с рейтингом ≥ 4 и с максимальной ценой 150000.

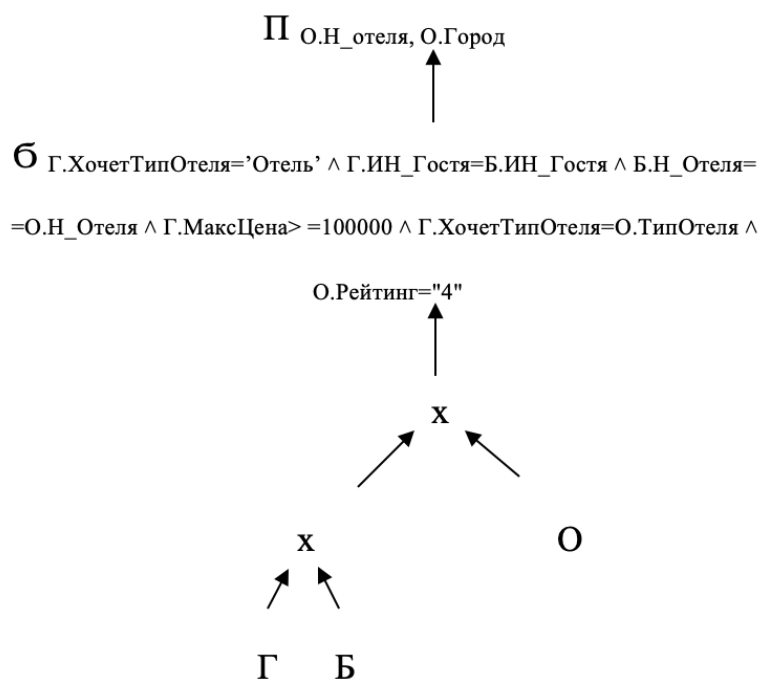


Рис. 1. Изначальное дерево реляционной алгебры.

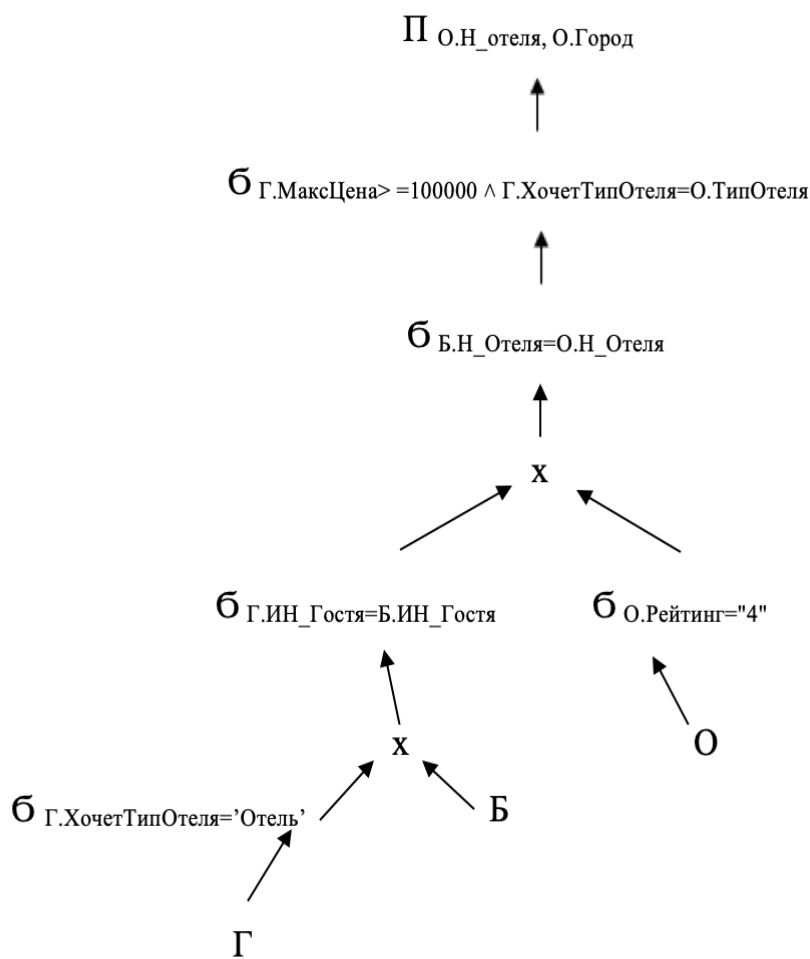


Рис. 2. Упрощённое дерево запроса.

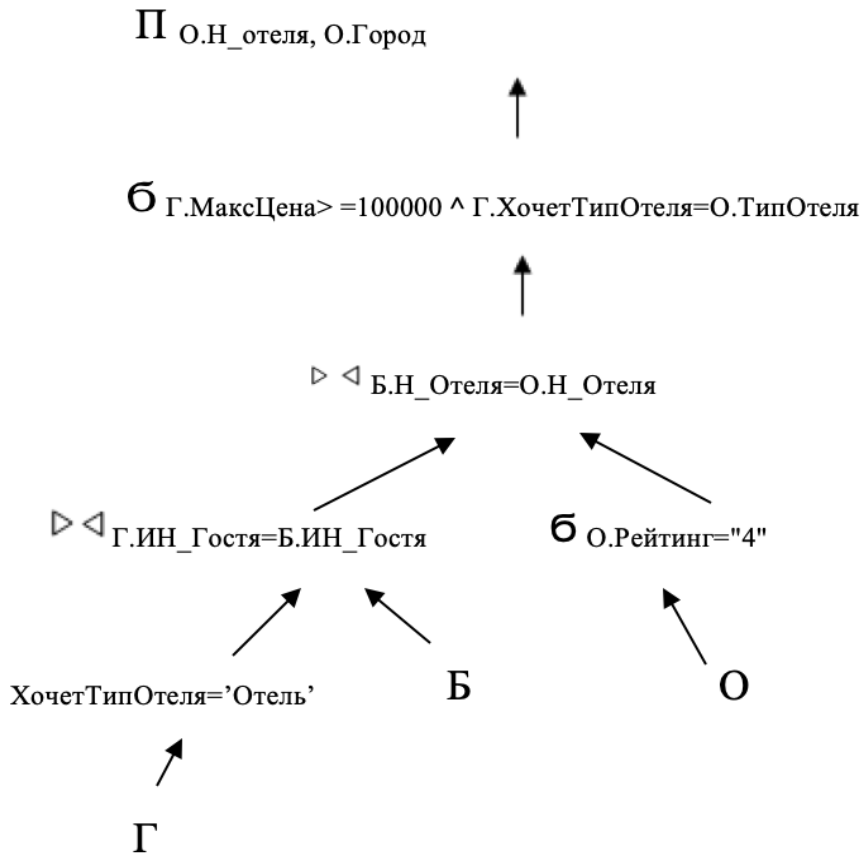


Рис. 3. Дальнейшее преобразование дерева запроса.

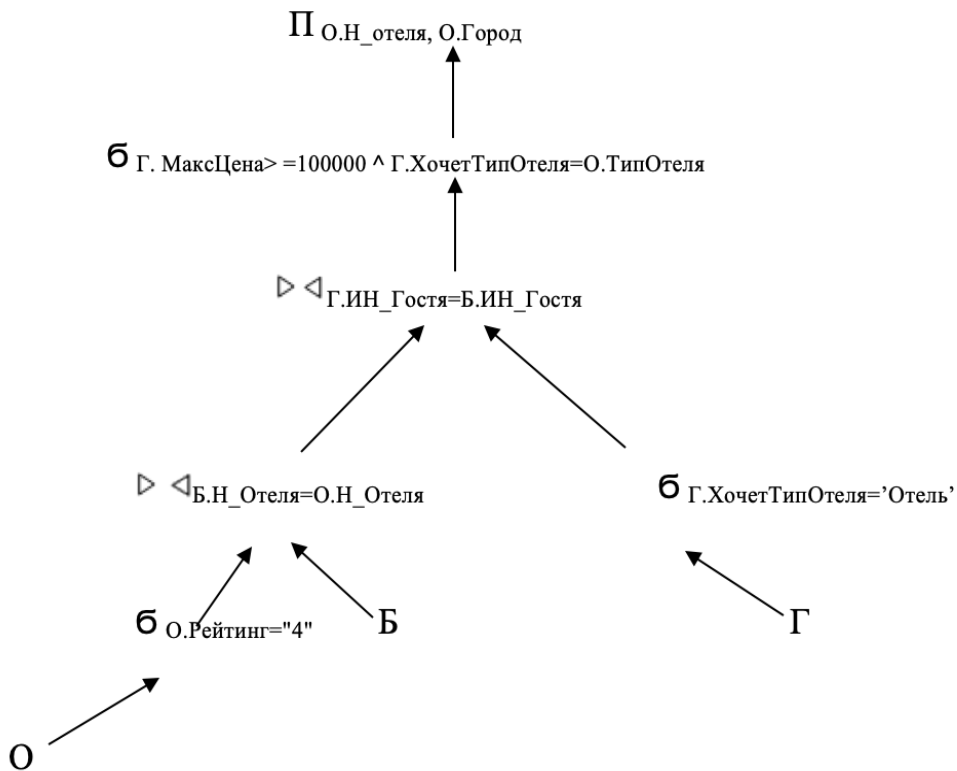


Рис. 4. Более эффективный план выполнения запроса.

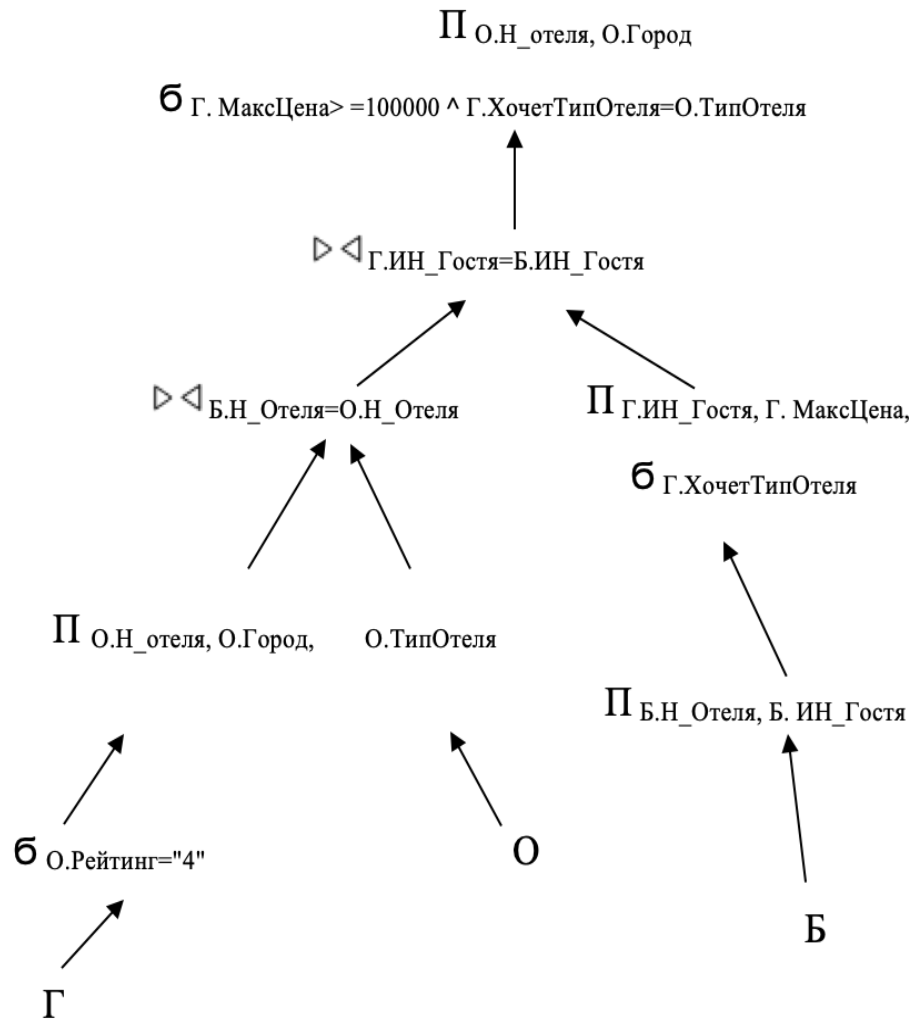


Рис. 5. Измененный план выполнения запроса.

Руководствуясь вышеперечисленными правилами трансформации выражений реляционной алгебры, рассмотрим запрос:

```
SELECT Отели.Н_отеля, Отели.Город
FROM Гости, Бронирование, Отели
WHERE Гости.ХочетТипОтеля='Отель' AND Гости.
ИН_Гостя=Бронирование.ИН_Гостя AND Бронирова-
ние.Н_Отеля=Отели.Н_Отеля AND Гости.ХочетТипОте-
ля=Отели.ТипОтеля AND Отели.Рейтинг=>4» AND Гости.
МаксЦена=150000
```

Перепишем данный SQL-запрос в терминах реляционной алгебры, выражение примет вид:

```
Π О.Н_отеля, О. Город ( ⋈ Г. ХочетТипОтеля='Отель' ∧ Г.ИН_Гостя=
Б.ИН_Гостя ∧ Б.Н_Отеля=О.Н_Отеля ∧ Г. МаксЦена> =100000 ∧
Г. ХочетТипОтеля=О.ТипОтеля ∧ О. Рейтинг=>4» ((ГхБ)хО)),
```

где в дальнейшем будем обозначать в качестве Г отношение R1, Б отношение R2, О — отношение R3.

Дерево выполнения данного запроса имеет вид рис. 1.

Теперь, применяя правила преобразования выражений реляционной алгебры, покажем, как можно добиться повышения эффективности обработки запроса (его оптимизации).

Для начала воспользуемся правилом *a)* — преобразовывающее операцию выборки по конъюнкции условий в последовательность операций выборки по каждому отдельному условию, входящему в конъюнкцию. Далее применим *b)* и *f)*, т.е. изменим порядок последовательности операций выборки, а затем применим правило коммутативности к операциям выборки и декартова произведения.

Дерево примет вид рис. 2.

Воспользуемся правилом, по которому операции выборки по предикату, включающему соединение по эквивалентности и декартово произведение, преобразуются в операцию соединения по эквивалентности:

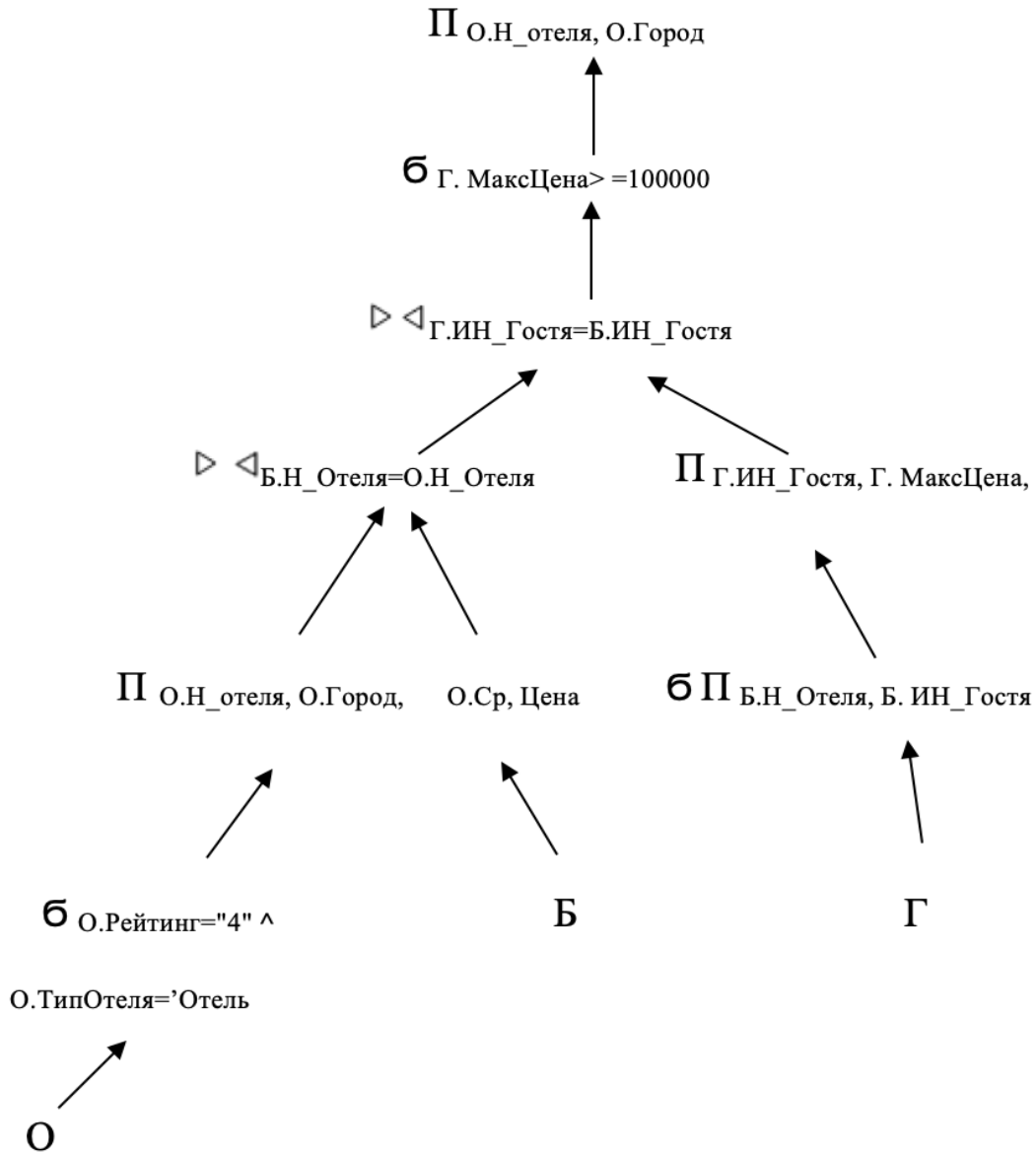


Рис. 6. Итоговое дерево данного запроса.

$$\mathbb{B}_{K,a=L,b} (K \times L) = K \bowtie_{K,a=L,b} L$$

В результате получается рис. 3.

С помощью *l)* преобразуем операцию соединения по эквивалентности так, чтобы выборка по условию *О. Рейтинг=»4»* запускалась вначале, получится рис. 4.

Далее воспользуемся пунктами *d)* и *g)* чтобы операции проекции производились после выполнения операций соединения по эквивалентности. Затем можно ввести новые проекции (рис. 5).

На последнем шаге можно так же заменить условие выборки *Г. ХочетТипОтеля= О. ТипОтеля* на *О. ТипОтеля = 'Отель'*.

Теперь, после того как были применены все возможные методы для оптимизации запроса, получаем следующий вид дерева оптимизированного запроса (рис. 6).

Выбирая стратегии эвристической обработки запросов в разных СУБД, применяются разные эвристические правила для выбора алгоритма обработки запросов. Но все же можно выделить несколько правил, к которым позволяют упростить дерево запроса для оптимизации процесса обработки запросов.

1. Выборка должна выполняться на первом этапе обработки. Так как эта операция позволяет уменьшить мощность таблицы и сокращает время выполнения последующих операций обработки. То есть из наших 12 правил, сначала следует применить, *a)* далее *b), d), f), j)*. Говоря

на языке деревьев, операция выборки нужно поместить как можно ниже в дереве реляционной алгебры.

2. Следует также объединять в одну операцию соединения операции прямого (декартова) произведения и операции выборки, предикат которой представляет условие соединения.
3. Следует применять правила ассоциативности бинарных операций с целью изменения порядка лист-узлов, чтобы лист-узлы с самыми ограничительными условиями выборки выполнялись в самом начале.
4. Необходимо также выполнить операции проекции на ранних этапах обработки. Эта операция, как и выборка уменьшает объем данных за счет снижения мощности отношения. Наиболее це-

лесообразна такая комбинация применения эвристических правил: *c)*, затем *d)*, *g)*, *k)*. В итоге операции проекции переместятся, на сколько это возможно, вниз по дереву реляционной алгебры.

5. Если одно выражение встречается в дереве реляционной алгебры несколько раз и результат его вычисления невелик, то эффективнее сохранить полученный результат после первого вычисления.

Обещающие направления включают разработку запросов к реляционной базе данных с использованием эвристических методов, которые демонстрируют эффективное применение для процесса оптимизации запросов в реляционных моделях.

ЛИТЕРАТУРА

1. О.Г. Алексеев. Комплексное применение методов дискретной оптимизации. — М. Наука, 2005.
2. Т. Коннолли, К. Бегг — Базы данных. Проектирование, реализация и сопровождение. Теория и практика. (3-е издание) — М.: Издательский дом «Вильямс», 2004.
3. О.Г. Алексеев. Комплексное применение методов дискретной оптимизации. — М. Наука, 2005.

© Куценко Ирина Львовна (i.kutsenko@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»



Российский университет дружбы народов