

*Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
Московский государственный технический университет им.
Н. Э. Баумана*

23 0102

*Домашнее задание №3 по дисциплине
«Теоретические основы реляционной алгебры»*

Студент группы ИУ5-92

_____ Гуца А. В

“ _____ ” _____

Содержание

1	Постановка задачи	3
2	Выполнение	5
2.1	Оператор SELECT	5
2.2	Форма реляционной алгебры	6
2.3	Логический план	8
2.4	Выбор порядка соединений	9
2.4.1	Вариант 4	11
2.4.2	Вариант 5	14
3	Физический план	17

1 Постановка задачи

Выдать имена поставщиков из Лондона, которые поставляют для изделия с номером «J1», по крайней мере, одну красную деталь.

Задание:

1. Написать соответствующий оператор SELECT и построить логический план выполнения этого оператора
2. Определить оптимальный физический план выполнения оператора SELECT при следующих исходных данных:

- 1) Количество записей в таблице:

$$T(S) = 10000$$

$$T(P) = 100000$$

$$T(SPJ) = 1000000$$

- 2) Количество записей в одном блоке таблицы:

$$L_S = 500$$

$$L_P = 500$$

$$L_{SPJ} = 1000$$

$$L_{JOIN} = 2000$$

- 3) Индексы атрибутов и число записей таблицы в одном блоке индекса (L):

— таблица S:

- 1) индекс по атрибуту « $N_{\text{пост}}$ », $L = 200$

— таблица P:

- 1) индекс по атрибуту « $N_{\text{дет}}$ », $L = 200$

– таблица SPJ:

- 1) индекс по атрибуту « $N_{\text{пост}}$ », $L = 200$
- 2) индекс по атрибуту « $N_{\text{дет}}$ », $L = 200$
- 3) индекс по атрибуту « $N_{\text{изд}}$ », $L = 200$

Примечения:

- записи таблиц могут читаться в отсортированном виде по своим индексированным атрибутам
- записи во всех таблицах не сгруппированы (нет кластеризации)

3. Мощности атрибутов:

$$I(S, \text{город}) = 50$$

$$I(S, N_{\text{пост}}) = 10000$$

$$I(P, \text{цвет}) = 20$$

$$I(P, N_{\text{дет}}) = 100000$$

$$I(SPJ, N_{\text{пост}}) = 5000$$

$$I(SPJ, N_{\text{дет}}) = 100000$$

$$I(SPJ, N_{\text{изд}}) = 10000$$

4. Число блоков $B = 10$, значения:

$$C_{\text{comp}} = C_{\text{move}} = C_{\text{filter}} = 0.01 \text{ мс}$$

$$C_B = 10 \text{ мс}$$

5. Предполагается, что используются левосторонние деревья для поиска оптимального плана и применяются каналы. Примечание: рассматривается метод NLJ (Nested Left Joins).

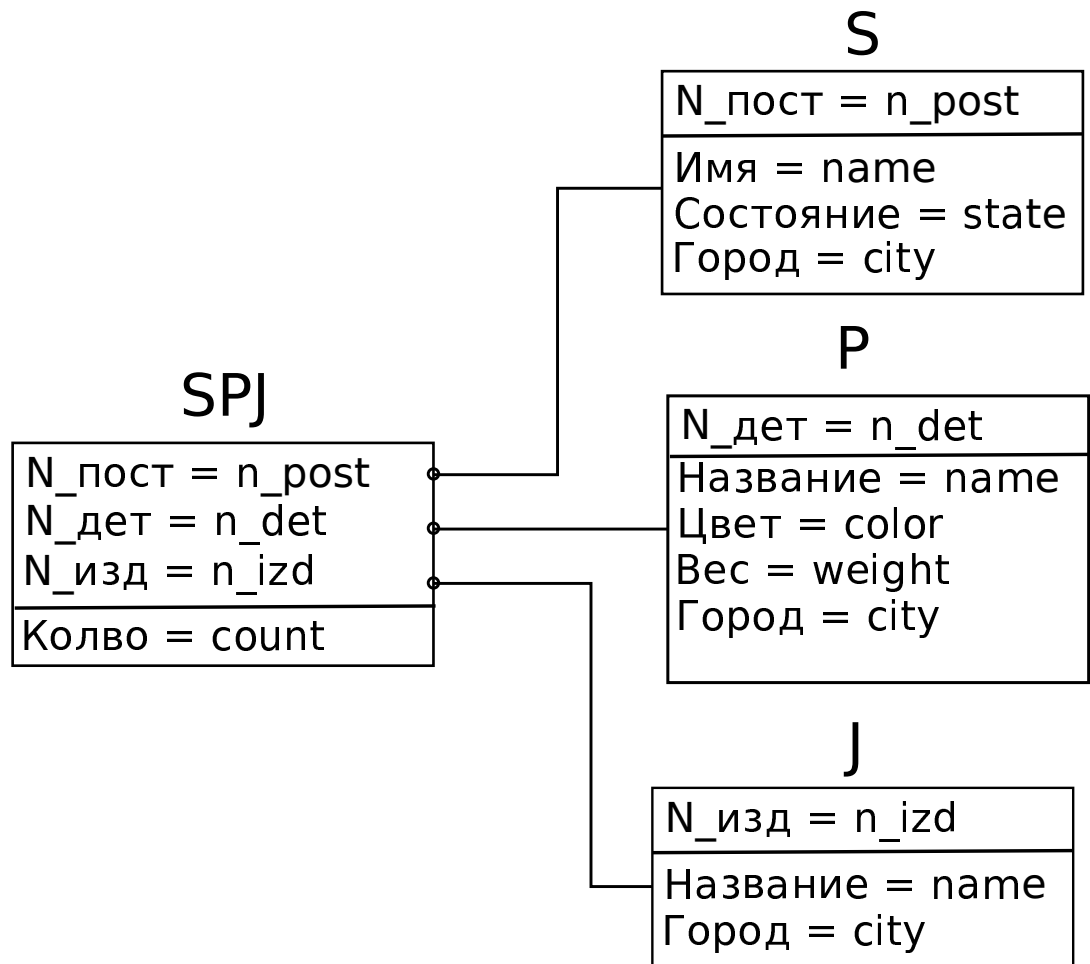


Рисунок 1 – Схема базы данных с переименованными атрибутами

2 Выполнение

2.1 Оператор SELECT

```
select S.name from SPJ, P, S
where SPJ.n_det = P.n_det
and SPJ.n_post = S.n_post
and SPJ.n_изд = 'J1'
and P.color = 'red'
and SPJ.count >= 1
and S.city = 'London'
```

2.2 Форма реляционной алгебры

Используемые законы:

1. Закон каскада селекций:

$$\sigma_{f_1 \wedge f_2}(R) = \sigma_{f_1}(\sigma_{f_2}(R))$$

2. Закон перестановки селекции и декартового произведения:

$$\sigma_{f_1}(R_1 \times R_2) = \sigma_{f_1}(R_1) \times R_2$$

$$attr(f_1) \in R_1 \wedge attr(f_1) \notin R_2$$

3. Закон перестановки проекции и селекции:

$$\Pi_{a_1..a_n}(\sigma_f(R)) = \Pi_{a_1..a_n}(\sigma_f(\Pi_{a_1..a_n b_1..b_k}(R)))$$

4. Закон перестановки проекции и декартового произведения:

$$b_1..b_n \in R_1$$

$$c_1..c_n \in R_2$$

$$\Pi_{b_1..b_n c_1..c_k}(R_1 \times R_2) = \Pi_{b_1..b_n}(R_1) \times \Pi_{c_1..c_k}(R_2)$$

Преобразуем оператор SELECT в форму реляционной алгебры:

$$Q = \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost \wedge spj.nizd='J1' \wedge p.color='red' \wedge \\ \wedge spj.count \geq 1 \wedge s.city='London'}(SPJ \times P \times S))$$

Применяем закон каскада селекций:

$$Q = \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost}(\sigma_{spj.nizd='J1' \wedge p.color='red' \wedge spj.count \geq 1 \wedge s.city='London'}(SPJ \times P \times S)))$$

Применяем закон перестановки селекции и декартового произведения:

$$Q = \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost}(\sigma_{spj.nizd='J1' \wedge spj.count \geq 1}(SPJ) \times \sigma_{p.color='red'}(P) \times \sigma_{s.city='London'}(S)))$$

Применяем закон перестановки проекции и селекции:

$$Q = \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost}(\Pi_{s.name, spj.ndet, p.ndet, spj.npost, s.npost}(\sigma_{spj.nizd='J1' \wedge spj.count \geq 1}(SPJ) \times \sigma_{p.color='red'}(P) \times \sigma_{s.city='London'}(S))))$$

Применяем закон перестановки проекции и декартового произведения:

$$Q = \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost}(\Pi_{spj.ndet, spj.npost}(\sigma_{spj.nizd='J1' \wedge spj.count \geq 1}(SPJ)) \times \Pi_{p.ndet}(\sigma_{p.color='red'}(P)) \times \Pi_{s.name, s.npost}(\sigma_{s.city='London'}(S))))$$

Выделяем подзапросы:

$$\begin{aligned} Q &= \Pi_{s.name}(\sigma_{spj.ndet=p.ndet \wedge spj.npost=s.npost}(Q_1 \times Q_2 \times Q_3)) \\ Q_1 &= \Pi_{spj.ndet, spj.npost}(\sigma_{spj.nizd='J1' \wedge spj.count \geq 1}(SPJ)) \\ Q_2 &= \Pi_{p.ndet}(\sigma_{p.color='red'}(P)) \\ Q_3 &= \Pi_{s.name, s.npost}(\sigma_{s.city='London'}(S)) \end{aligned}$$

2.3 Логический план

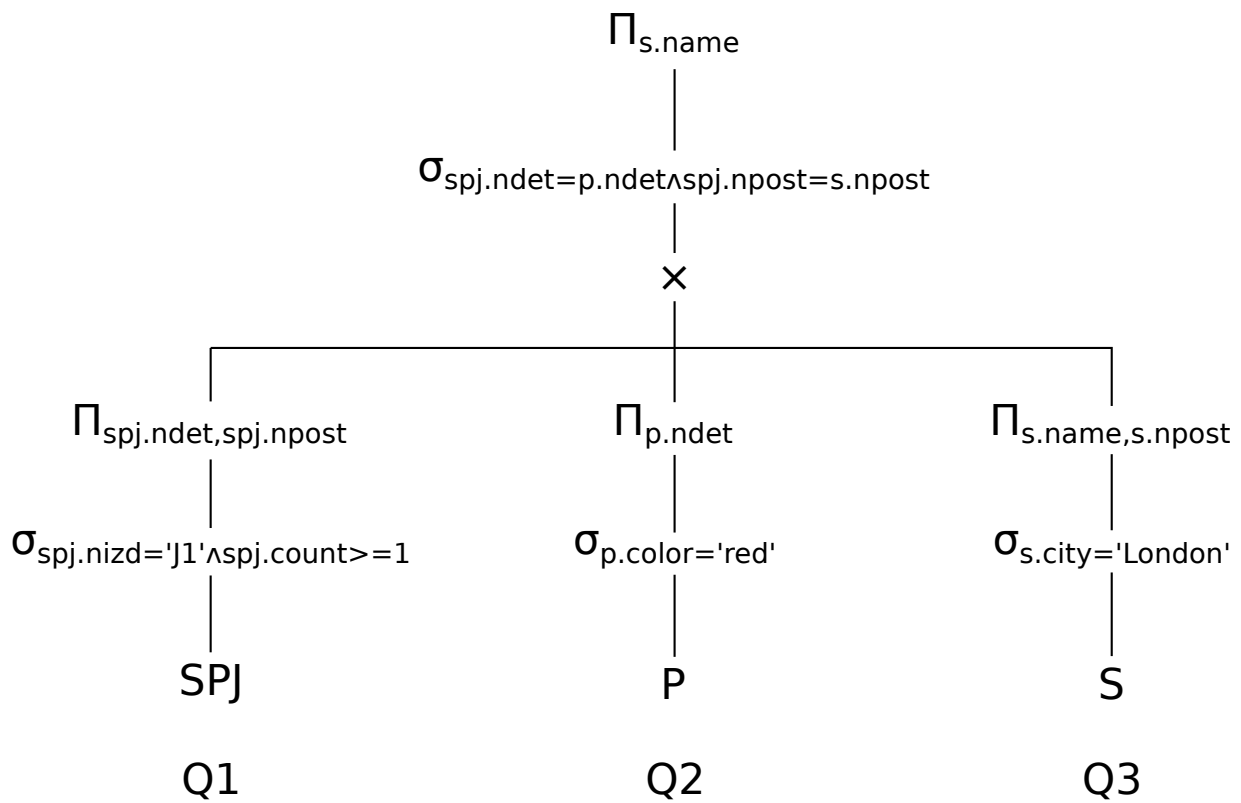


Рисунок 2 – Логический план запроса

2.4 Выбор порядка соединений

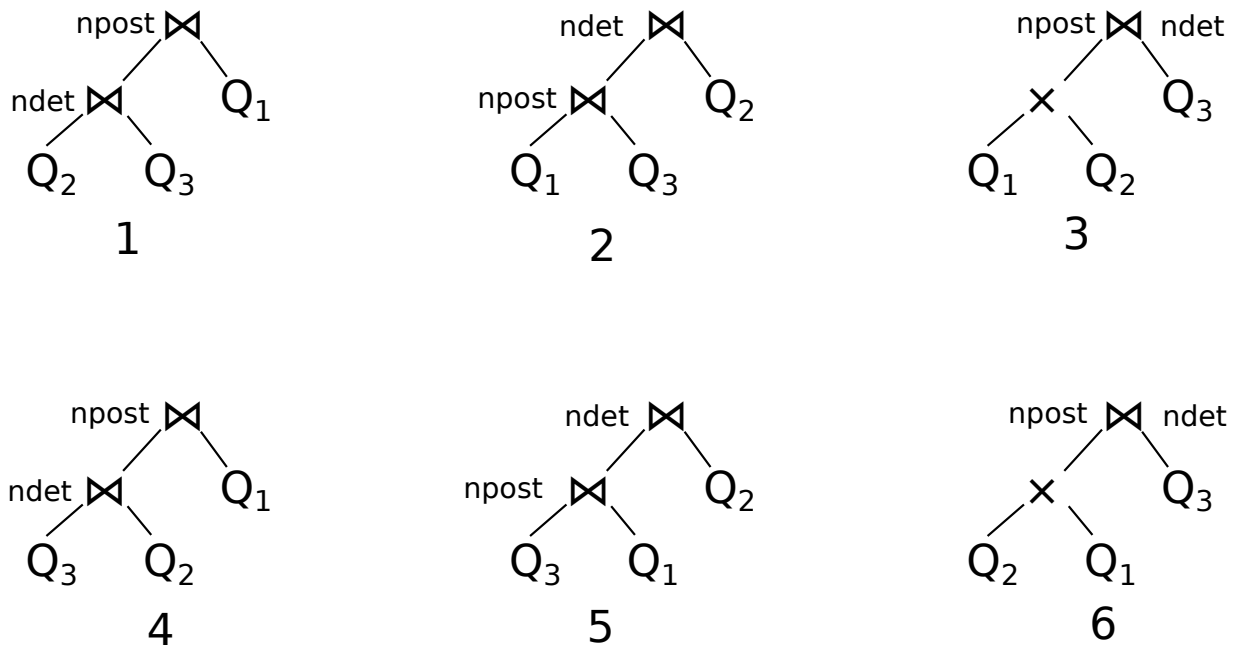


Рисунок 3 – Варианты соединения таблиц

Варианты 3 и 6 не рассматриваются в алгоритме, так как там используется декартовое произведение, что заведомо хуже других вариантов. Переномеруем варианты соединения, исключая 3 и 6.

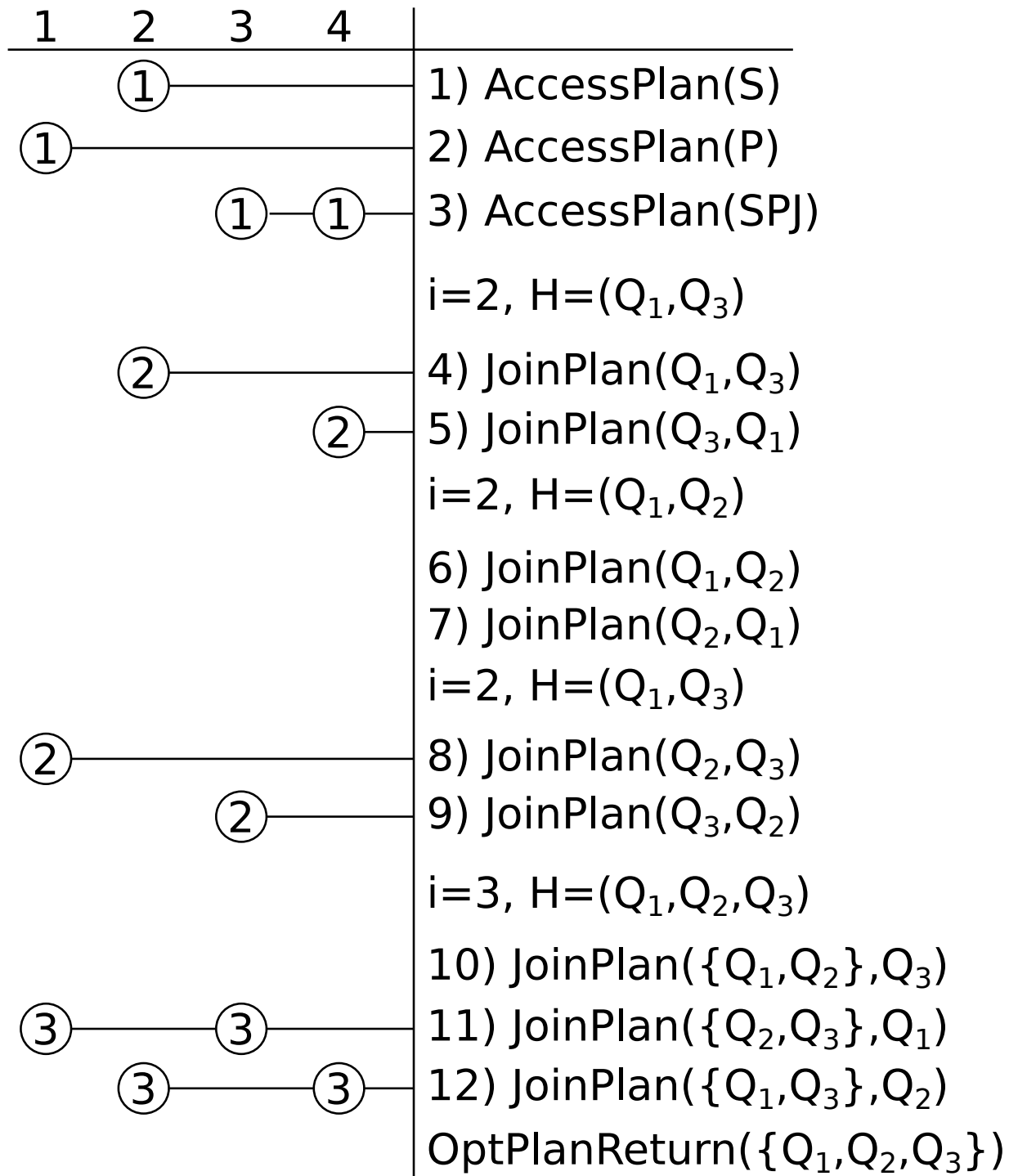


Рисунок 4 – Схема алгоритма на основе динамического программирования

Варианты 1 и 2 не рассматриваются, так как в них потребуется полное сканирование таблиц из-за соединения по атрибутам, для которых не построены индексы.

Итого из рисунка 3 рассматриваются варианты 4 и 5. Посчитаем условную стоимость (единицу измерения примем в миллисекундах) выполнения каждого из вариантов соединения таблиц.

2.4.1 Вариант 4

AccessPlan(SPJ) :

1) Чтение всех SPJ, TableScan(SPJ):

$$\begin{aligned} C_1 &= C_{cpu1} + C_{io1} = T(SPJ)C_{filter} + \frac{T(SPJ)}{L_{SPJ}}C_B = \\ &= 20000 \text{ мс} \end{aligned}$$

Где:

- C_{cpu1} - оценка стоимости обработки на ЦПУ в миллисекундах
- C_{io1} - оценка стоимости операций ввода-вывода в миллисекундах
- $T(SPJ)$ - количество строк в таблице SPJ
- L_{SPJ} - количество записей в одном блоке таблицы SPJ
- C_{filter} - стоимость фильтрации одной строки в миллисекундах
- C_B - стоимость чтения одной строки в миллисекундах

2) Чтение по индексу, IndexScan(SPJ, nizzd):

$$\begin{aligned} C_2 &= C_{cpu2} + C_{io2} = \frac{T(SPJ)k}{I(SPJ, nizzd)}C_{filter} + \\ &+ \left(\left\lceil \frac{\frac{T(SPJ)}{L}k}{I(SPJ, nizzd)} \right\rceil + \left\lceil \frac{T(SPJ)k}{I(SPJ, nizzd)} \right\rceil \right) C_B = \\ &= 1011 \text{ мс} \end{aligned}$$

Где:

- L - число записей в одном блоке индекса
- L_{SPJ} - количество записей в одном блоке таблицы SPJ
- k - мощность атрибута в запросе
- $I(SPJ, nizzd)$ - мощность атрибута nizzd в таблице SPJ

Выбираем способ с минимальной оценкой:

$$C = \min(C_1, C_2) = C_2 = 1011$$

Оценка стоимости операций ввода-вывода:

$$C_{io} = 1010$$

Считаем количество строк в промежуточной таблице:

$$T(Q_3) = T(SPJ)p_{nizd} = T(SPJ)\frac{k}{I(SPJ, nizd)} = 100$$

Где p_{nizd} - вероятность того, что строка в SPJ удовлетворит условию фильтрации.

Количество блоков в промежуточной таблице:

$$B(Q_3) = \left\lceil \frac{T(Q_3)}{L_{SPJ}B} \right\rceil = 1$$

Заполняем промежуточную структуру:

$$str[1] = \{Q_3, \emptyset, \emptyset, 1011, 1010, \{100, 1, \{ndet, 100\}, nizd\}\}$$

Данная структура имеет следующие поля:

- W - кортеж из таблиц или таблица, которая получается в результате операции (соединения)
- X - левый аргумент операции
- Y - правый аргумент операции
- Z - оценка времени совокупная
- ZIO - оценка времени ввода-вывода
- V - подструктура:
 - T - оценка количества строк в новой таблице
 - I - оценка количества блоков в новой таблице
 - подструктура, содержащая индекс, по которому будет проводится следующее соединение и мощность этого атрибута, ко-

торое считается как $\min(T(Q), I(R, atr))$, где R - исходная таблица, Q - промежуточная таблица

– K - атрибут-индекс, по которому происходило объединение

JoinPlan(Q_3, Q_2)

1. Для Q_3 :

$$N = T(Q_3) = str[1].V.T = 100$$

Для каждой полученной записи в Q_3 выполняется поиск в индексе по номеру детали в R.

$$C = N \left(\frac{T(P)k}{I(P, ndet)} C_{filter} + \left(\left\lceil \frac{\frac{T(P)}{L_P} k}{I(P, ndet)} \right\rceil + \left\lceil \frac{T(P)k}{I(P, ndet)} \right\rceil \right) C_B \right)$$

$$= 2000 \text{ мс}$$

$$T(Q_2) = T(P) \frac{str[1].V.T}{I(P, ndet)} p_{color='red'} = 5$$

$$B(Q_2) = \left\lceil \frac{T(Q_2)}{L_P B} \right\rceil = 1$$

$$str[2] = \{Q_2, \emptyset, \emptyset, 2000, 2000, \{5, 1, \{ndet, 5\}, ndet\}\}$$

2. Для $Q_3 \bowtie Q_2$:

$$m_1 = 1, m_2 = 2$$

$$C = str[1].Z + str[2] = 1011 + 2000 = 3011$$

$$C_{io} = str[1].ZIO + str[2] = 1010 + 2000 = 3010$$

$$T(Q_3 \bowtie Q_2) = T(Q_2) = str[2].V.T = 5$$

$$B(Q_3 \bowtie Q_2) = \left\lceil \frac{T(Q_3 \bowtie Q_2)}{L_{JOIN}} \right\rceil = 1$$

$$str[3] = \{(Q_3, Q_2), Q_3, Q_2, 3011, 3010, \{5, 1, \{\{nizd, 5\}\}, ndet\}\}$$

JoinPlan($\{Q_3, Q_2\}, Q_1$)

1. $\{Q_3, Q_2\}$

$$V = T(Q_3 \bowtie Q_2) = str[3].V.T = 5$$

$$C = N \left(\frac{T(S)k}{I(S, nized)} C_{filter} + \left(\left\lceil \frac{\frac{T(S)}{L_S} k}{I(s, nized)} \right\rceil + \left\lceil \frac{T(S)k}{I(S, nized)} \right\rceil \right) C_B \right) = 100$$

$$T(Q_1) = T(S) \frac{str[3].V.T}{I(S, nized)} p_{city='London'} = 1$$

$$B(Q_1) = \left\lceil \frac{T(Q_1)}{L_S B} \right\rceil = 1$$

$$str[4] = \{Q_1, \emptyset, \emptyset, 100, 100, \{1, 1, \{nized, 1\}, nized\}\}$$

2. $(Q_3 \bowtie Q_2) \bowtie Q_1$

$$C = str[3].Z + str[4].Z = 3011 + 100 = 3111$$

$$C_{io} = str[3].ZIO + str[4].ZIO = 3010 + 100 = 3110$$

$$T((Q_3 \bowtie Q_2) \bowtie Q_1) = T(Q_2) = str[4].V.T = 1$$

$$B((Q_3 \bowtie Q_2) \bowtie Q_1) = \left\lceil \frac{(Q_3 \bowtie Q_2) \bowtie Q_1}{L_{JOIN} B} \right\rceil = 1$$

$$str[5] = \{\{Q_3, Q_2, Q_1\}, \{Q_3, Q_2\}, Q_1, 3111, 3110, \{1, 1, \{\}, nized\}\}$$

2.4.2 Вариант 5

AccessPlan(SPJ)

1) Чтение всей таблицы SPJ, TableScan(SPJ)

$$C_1 = C_{cpu1} + C_{io1} = T(SPJ)C_{filter} + \frac{T(SPJ)}{L_{SPJ}} C_B = 10000 \text{ мс}$$

2) Чтение записи с помощью индекса:

$$\begin{aligned}
C_2 &= C_{cpu2} + C_{io2} = \frac{T(SPJ)k}{I(SPJ, nizzd)} C_{filter} + \\
&+ \left(\left\lceil \frac{\frac{T(SPJ)k}{L_{SPJ}}}{I(SPJ, nizzd)} \right\rceil + \left\lceil \frac{T(SPJ)k}{I(SPJ, nizzd)} \right\rceil \right) C_B = \\
&= 1011 \text{ мс}
\end{aligned}$$

$$C' = \min(C_1, C_2) = 1011 \text{ мс}$$

$$C_{io} = 1010 \text{ мс}$$

$$T(Q_3) = T(SPJ)p_{nizzd=J1'} = T(SPJ)\frac{k}{I(SPJ, nizzd)} = 100$$

$$B(Q_3) = \left\lceil \frac{T(Q_3)}{L_{SPJ}B} \right\rceil = 1$$

$$str[1] = \{Q_3, \emptyset, \emptyset, 1011, 1010, \{100, 1, \{nizzd, 100\}, nizzd\}\}$$

JoinPlan(Q_3, Q_1)

1. Q_3

$$N = T(Q_3) = str[1].V.T = 100$$

$$C = N \left(\frac{T(S)k}{I(S, npost)} C_{filter} + \left(\left\lceil \frac{\frac{T(S)k}{L_S}}{I(S, npost)} \right\rceil + \left\lceil \frac{T(S)k}{I(S, npost)} \right\rceil \right) C_B \right) = 2$$

$$B(Q_1) = \left\lceil \frac{T(Q_2)}{L_S B} \right\rceil = 1$$

$$str[2] = \{Q_1, \emptyset, \emptyset, 2000, 2000, \{2, 1, \{npost, 2\}, npost\}\}$$

2. $Q_3 \bowtie Q_1$

$$C = str[1].Z + str[2].Z = 1011 + 2000 = 3011$$

$$C_{io} = str[1].ZIO + str[2].ZIO = 1010 + 2000 = 3010$$

$$T(Q_3 \bowtie Q_1) = T(Q_1) = 2$$

$$B(Q_3 \bowtie Q_1) = \left\lceil \frac{T(Q_3 \bowtie Q_1)}{L_{JOIN}} \right\rceil = 1$$

$$str[3] = \{Q_3 \bowtie Q_1, Q_3, Q_1, 3011, 3010, \{2, 1, \{ndet, 2\}, npost\}\}$$

JoinPlan($\{Q_3, Q_1\}, Q_2$)

1. $\{Q_3, Q_1\}$

$$N = T(Q_3 \bowtie Q_1) = str[3].V.T = 2$$

$$C = N \left(\frac{T(P)k}{I(P, ndet)} C_{filter} + \left(\left\lceil \frac{\frac{T(P)}{L_P} k}{I(P, ndet)} \right\rceil + \left\lceil \frac{T(P)k}{I(P, ndet)} \right\rceil \right) C_B \right) = 40$$

$$C_{io} = 40$$

$$T(Q_2) = \frac{T(P)str[2].V.T}{I(P, ndet)} P_{color='red'} = 1$$

$$str[4] = \{Q_2, \emptyset, \emptyset, 40, 40, \{1, 1, \{ndet, 1\}, ndet\}\}$$

2. $(Q_3 \bowtie Q_1) \bowtie Q_2$

$$C = str[3].Z + str[4].Z = 3051$$

$$C_{io} = str[3].ZIO + str[4].ZIO = 3050$$

$$T((Q_3 \bowtie Q_1) \bowtie Q_2) = T(Q_2) = str[4].V.T = 2$$

$$B((Q_3 \bowtie Q_1) \bowtie Q_2) = \left\lceil \frac{T((Q_3 \bowtie Q_1) \bowtie Q_2)}{L_{JOIN}} \right\rceil = 1$$

$$str[5] = \{\{Q_3, Q_1, Q_2\}, \{Q_3, Q_1\}, Q_2, 3051, 3050, \{2, 1, \{\}, ndet\}\}$$

OptPlanReturn Сравниваем $str[5].Z$:

$$3111 > 3051$$

Значит берем вариант 4.

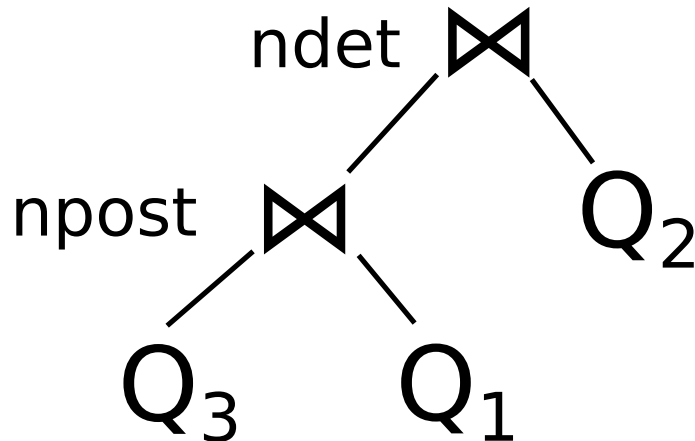


Рисунок 5 – Лучший порядок соединения таблиц

3 Физический план

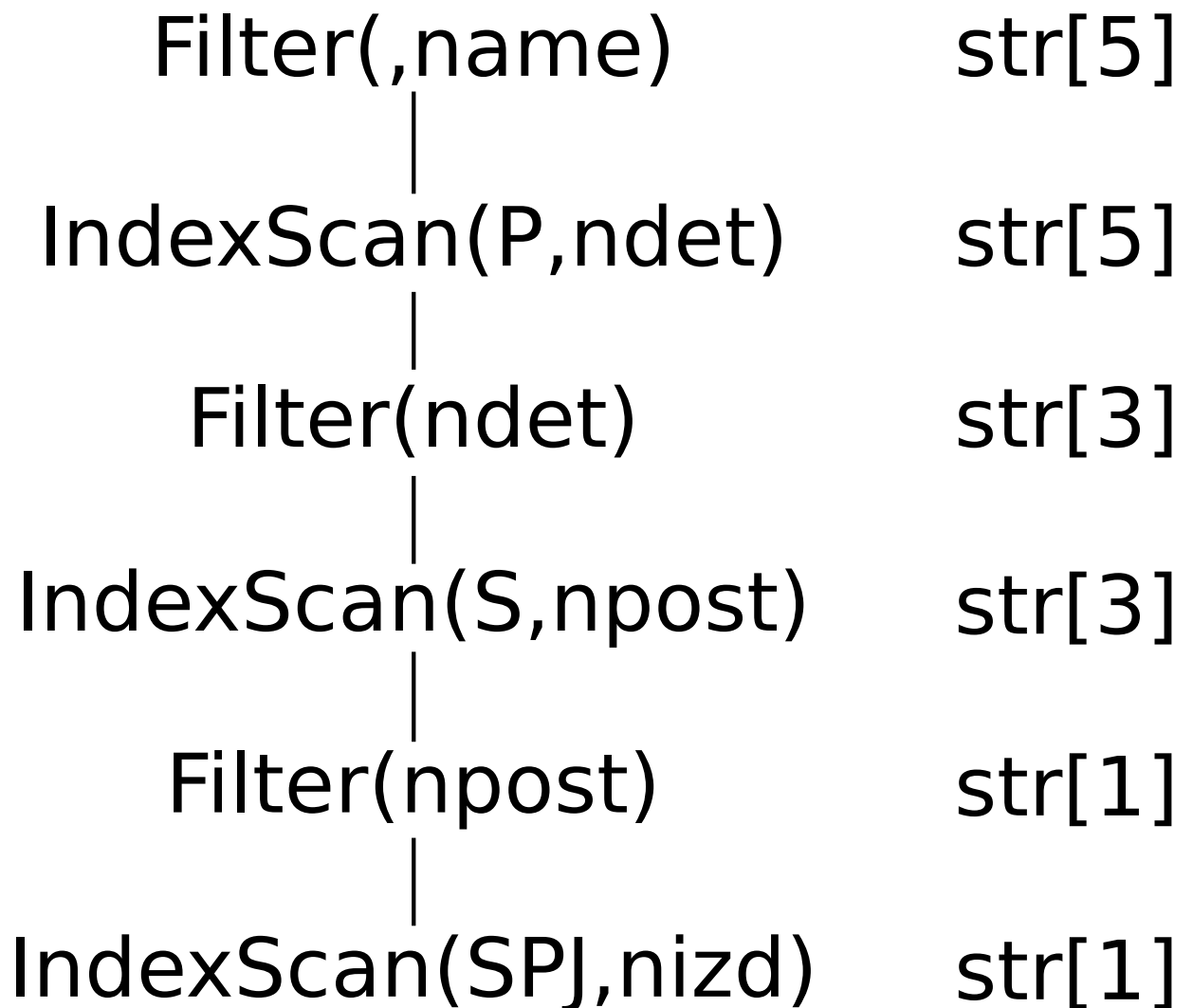


Рисунок 6 – Физический план выполнения запроса