

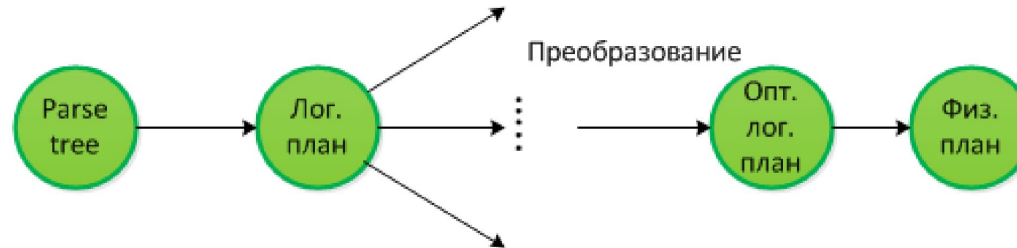
Дисциплина «Проектирование баз данных»



Маркова Ирина Васильевна,
начальник управления
информатизации
markova@mit.ru



Построение логического плана



Процесс состоит из двух этапов:

- замена вершин и структур дерева разбора соответствующими операторами реляционной алгебры;
- перезапись с помощью алгебраических законов (генерация эквивалентных логических планов).



Замена вершин и структур дерева разбора

Если синтаксическая категория `<Query>` представляет собой конструкцию `<SFW>` и подчинённая категория `<Condition>` не содержит подзапросов, её целиком можно заменить выражением реляционной алгебры, состоящем из следующих компонент:

- ✓ декартова произведения всех отношений из списка `<Fromlist>`, которое является аргументом;
- ✓ оператора выбора σ_c ;
- ✓ оператора проекции π_L (L соответствует списку атрибутов `<SelList>`).

Изъятие подзапросов из условий:

самостоятельно стр. 775-780 (Гарсия-Молина, Гектор, Ульман, Джеффри, Д., Уидом, Дженнифер. Системы баз данных. Полный курс.: Пер. с англ.- М.: Издательский дом «Вильямс», 2004.-1088 с.: ил.).



Изображение дерева запроса

Дерево запроса, соответствующее дереву разбора для оператора::

```
SELECT К.Наименование  
FROM Клиенты К, Поставки П  
WHERE К.Код_К = П.Код_К AND П.Код_Г='Г2'
```





Алгебраические законы

Алгебраические законы:

- коммутативный и ассоциативный законы;
- законы выбора;
- законы проекции;
- законы соединения и декартового произведения;
- законы удаления кортежей-дубликатов;
- законы группировки и агрегирования.



Коммутативные и ассоциативные законы

Коммутативный закон: порядок аргументов оператора не влияет на результат.

Ассоциативный закон: разрешает группирование аргументов нескольких одноимённых операторов.

$$\begin{array}{ll}
 R_{s,b} \times S_{s,b} = S_{s,b} \times R_{s,b}; & (R_{s,b} \times S_{s,b}) \times T_{s,b} = R_{s,b} \times (S_{s,b} \times T_{s,b}); \\
 R_{s,b} \boxtimes S_{s,b} = S_{s,b} \boxtimes R_{s,b}; & (R_{s,b} \boxtimes S_{s,b}) \boxtimes T_{s,b} = R_{s,b} \boxtimes (S_{s,b} \boxtimes T_{s,b}); \\
 R_{s,b} \boxdot S_{s,b} = S_{s,b} \boxdot R_{s,b}; & (R_{s,b} \boxdot S_{s,b}) \boxdot T_{s,b} = R_{s,b} \boxdot (S_{s,b} \boxdot T_{s,b}); \\
 R_{s,b} \boxplus S_{s,b} = S_{s,b} \boxplus R_{s,b}; & (R_{s,b} \boxplus S_{s,b}) \boxplus T_{s,b} = R_{s,b} \boxplus (S_{s,b} \boxplus T_{s,b});
 \end{array}$$

где

s и b – подстрочный индекс для операций над множествами и мультимножествами соответственно.

Мультимножество – множество, допускающее наличие дубликатов элементов.



Коммутативные и ассоциативные законы (продолжение)

Операции разности $(R \setminus S)$ и деления $(R \div S)$ не коммутативны и не ассоциативны.

Θ -соединение не причисляется к категории коммутативно-ассоциативных.

В общем виде

$$R \underset{c}{\Theta} S = S \underset{c}{\Theta} R \text{ – коммутативен.}$$

Если условие c сохраняет смысл при различном группировании аргументов, операция Θ -соединения является и ассоциативной.

Но существуют примеры, где ассоциативный закон утрачивает силу, т.к. условия операторов не допускают применения к атрибутам отношения, подвергающихся Θ -соединению.



Коммутативные и ассоциативные законы (продолжение)

Пример (исключения):

Пусть даны отношения $R(a,b)$, $S(b,c)$ и $T(c,d)$ и выражение $(R \underset{R.b > S.b}{\Theta} S) \underset{a < d}{\Theta} T$.

Применим гипотетический ассоциативный закон $R \underset{R.b > S.b}{\Theta} (S \underset{a < d}{\Theta} T)$, но эти отношения нельзя соединить на основе условия $a < d$, т.к. $a \notin S$, $a \notin T$, а $d \in T$, но $d \notin S$.

Отсюда следует, что ассоциативный закон не может быть применен к произвольным выражением с оператором Θ -соединения.



Коммутативные и ассоциативные законы (продолжение)

Не все законы, справедливые для множеств, справедливы и для мультимножеств.

Например, рассмотрим дистрибутивный закон для операции пересечения. Он справедлив для множеств:

$$A \cap_s (B \cap_s C) = (A \cap_s B) \cap_s (A \cap_s C).$$

Однако, этот закон не работает для мультимножеств.

Пусть, например, имеются мультимножества $A = \{x\}$, $B = \{x\}$, $C = \{x\}$.

Тогда левая часть равенства $A \cap_b (B \cap_b C) = \{x\} \cap_b \{x, x\} = \{x\}$, а правая

$(A \cap_b B) \cap_b (A \cap_b C) = \{x\} \cap_b \{x\} = \{x, x\}$, откуда видно, что для мультимножеств

$$A \cap_b (B \cap_b C) \neq (A \cap_b B) \cap_b (A \cap_b C).$$



Законы выбора

Порядок выполнения операций выборки является значимым, так как при осуществлении выбора кортежей размеры отношений способны существенным образом уменьшаться, одно из важнейших правил повышения эффективности обработки запросов состоит в смещении операторов вниз по дереву выражений настолько «глубоко», насколько это возможно без изменения семантики выражения в целом.

- законы расщепления;
- законы распределения выборки по бинарным операторам.



Закон расщепления

1. $\sigma_{c_1 \text{ AND } c_2}(R) = \sigma_{c_1}(\sigma_{c_2}(R))$
2. $\sigma_{c_1 \text{ OR } c_2}(R) = (\sigma_{c_1}(R)) \bowtie_s (\sigma_{c_2}(R))$

Пример:

Пусть дано отношение $R(a, b, c)$ и выражение $\sigma_{(a=1 \text{ OR } a=3) \text{ AND } b < c}(R)$.

- а) применив сначала первый закон расщепления, а затем второй, исходное выражение можно преобразовать сначала к виду $\sigma_{a=1 \text{ OR } a=3}(\sigma_{b < c}(R))$, а затем к $\sigma_{a=1}(\sigma_{b < c}(R)) \bowtie \sigma_{a=3}(\sigma_{b < c}(R))$;
- б) можно начать преобразование, сделав условие $\sigma_{b < c}$ внешним оператором $\sigma_{b < c}(\sigma_{a=1 \text{ OR } a=3}(R))$, а затем применив второй закон $\sigma_{b < c}(\sigma_{a=1}(R) \bowtie \sigma_{a=3}(R))$



Законы распределения

1. для операции объединения выборка применяется к обоим аргументам:

$$\sigma_c(R \bowtie S) = \sigma_c(R) \bowtie \sigma_c(S).$$

В данном случае оператор продвигается «вниз» по обеим ветвям дерева запроса.

2. для разности выборку следует применять к первому и (необязательно) ко второму аргументу:

$$\sigma_c(R \setminus S) = \sigma_c(R) \setminus S,$$

$$\sigma_c(R \setminus S) = \sigma_c(R) \setminus \sigma_c(S).$$



Законы распределения (продолжение)

3. для остальных бинарных операторов выборка применяется к одному или обоим аргументам (аргументом оператора выборки σ_c может быть только такое отношение, которое содержит все атрибуты, упомянутые в условии c):

$$\begin{aligned}\sigma_c(R \times S) &= \sigma_c(R) \times S, \\ \sigma_c(R \bowtie S) &= \sigma_c(R) \bowtie S, \\ \sigma_c(R \oplus_d S) &= \sigma_c(R) \oplus_d S, \\ \sigma_c(R \Join S) &= \sigma_c(R) \Join S.\end{aligned}$$

Если всеми атрибутами из условия σ_c обладают оба отношения, тогда будет справедливо:

$$\begin{aligned}\sigma_c(R \bowtie S) &= \sigma_c(R) \bowtie S, \\ \sigma_c(R \Join S) &= \sigma_c(R) \Join \sigma_c(S).\end{aligned}$$

Такое распределение невыполнимо для операций декартова произведения $\times$ и тета-соединения $\oplus_d$, т.к. схема итогового отношения $R \times S$ (или $R \oplus_d S$) представляет собой объединение схем исходных отношений и все их атрибуты строго различаются.



Законы распределения (продолжение)

Пример:

Пусть даны отношения $R(a,b)$, $S(b,c)$ и выражение $\sigma_{(a=1 \vee a=3) \wedge b < c}(R \bowtie S)$.

Проанализировав условия и наличие соответствующих атрибутов в отношениях, можно применить закон расщепления $\sigma_{a=1 \vee a=3}(\sigma_{b < c}(R \bowtie S))$, затем распределяя условия $\sigma_{b < c}$ по бинарной операции, имеем $\sigma_{a=1 \vee a=3}(R \bowtie \sigma_{b < c}(S))$. Теперь применяем внешний оператор выборки к отношению R : $\sigma_{a=1 \vee a=3}(R) \bowtie \sigma_{b < c}(S)$

Вывод: оператор выбора с более простым условием, охватывающим меньшее количество атрибутов, может быть перемещен в такое место, куда исходный оператор продвинуть просто нельзя.



Некоторые важные тривиальные законы

$$\sigma_c(R) = \emptyset, \text{ если } R = \emptyset,$$

$$\sigma_c(R) = R, \text{ если условие } c \text{ – всегда TRUE}$$

$$R \bowtie S = S, \text{ если } R = \emptyset$$

и др.



Продвижение оператора выбора «вверх»

Перемещение оператора выбора «вниз» по дереву выражений – один из наиболее мощных инструментов преобразования и оптимизации запросов. В некоторых случаях целесообразно переместить оператор выбора сначала «вверх», чтобы потом «охватить» им наибольшее число ветвей.

```
CREATE VIEW К.Москва AS
SELECT К.Код, К.Наименование
FROM Клиенты К
WHERE К.Город = 'Москва'
```

```
SELECT Г.Код, Г.Наименование, КМ.
Наименование
FROM К.Москва КМ, Грузы Г
WHERE КМ.Город = Г.Город
```



Изображение дерева запроса

$$\sigma_c(R \bowtie S) = \sigma_c(R) \bowtie S$$



$$\sigma_c(R \bowtie S) = \sigma_c(R) \bowtie \sigma_c(S)$$





Законы проекции (обозначения)

Оператор можно размещать в любом месте дерева выражений, если он удаляет только такие атрибуты, которые не используются им ни в одном из операторов, расположенных выше по дереву, и не присутствуют в отношении, являющимся итогом вычисления выражения в целом.

Введём дополнительные обозначения для расширенной проекции:

$E \rightarrow x$ – элемент списка проекции, где

E – входной атрибут или выражение, состоящее из атрибутов и констант,

x – выходной атрибут.

Если элемент списка является одиночным атрибутом, то он является входным и выходным.

Простая проекция – проекция, в списке которой присутствуют только одиночные атрибуты.

- все операторы проекции, рассматриваемые в рамках классической реляционной алгебры, являются простыми;
- в большинстве основных форм законов операции проекции трактуются как простые.



Законы проекции

1. В последовательности проекции можно игнорировать все проекции, кроме последней:

$$\pi_B(\pi_C(R)) = \pi_B(R), \text{ если } B \subset C.$$

2. Выборка проекции может быть трансформирована в проекцию выборки (т.е. σ и π – коммутативны):

$$\sigma_F(\pi_C(R)) = \pi_C(\sigma_F(R)),$$

если предикат F включает только те атрибуты, которые входят в список проекции C .

В большинстве случаев считают, что σ должна предшествовать π , т.к это приводит к уменьшению размера входных данных для операции π , что, в свою очередь, снижает объём данных, который надо сортировать для исключения дублирующих записей в процессе вычисления π .



Законы проекции (продолжение)

3. Распределение π по бинарным операторам (для расширенной π в версиях для множеств и мультимножеств):

$$\pi_L(R \bowtie S) = \pi_L(\pi_M(R) \bowtie \pi_N(S)),$$

$$\pi_L(R \ominus_d S) = \pi_L(\pi_M(R) \ominus_d \pi_N(S)),$$

$$\pi_L(R \times S) = \pi_L(\pi_M(R) \times \pi_N(S)),$$

где

M – список всех атрибутов отношения R , которые являются либо общими для R и S , либо входят в список L ,

N – список всех атрибутов отношения S , которые являются либо общими для R и S , либо входят в список L .

Применение этих законов приводит к выполнению «ранних» проекций.



Законы проекции (пример)

Пусть даны отношения $R(a, b, c)$, $S(c, d, e)$ и выражение $\pi_{a+e \rightarrow x, b \rightarrow y}(R \bowtie S)$.

Применим закон,

$$\pi_{a+e \rightarrow x, b \rightarrow y}(R \bowtie S) = \pi_{a+e \rightarrow x, b \rightarrow y}(\pi_{a, b, c}(R) \bowtie \pi_{c, e}(S)) = \pi_{a+e \rightarrow x, b \rightarrow y}(R \bowtie \pi_{c, e}(S)).$$

Полученное выражение отличается от исходного тем, что предполагает удаление атрибута d отношения S непосредственно перед выполнением соединения.



Законы проекции (продолжение)

4. Замена проекции «мультимножественной» версии операции объединения отношений объединением отношений-проекций:

$$\pi_L(R \bowtie_B S) = \pi_L(R) \bowtie_B \pi_L(S).$$

5. Преобразования, связанные с операциями $\cup$ множеств, $\cap$ и $\setminus$ (для множеств и мультимножеств), применять нельзя.

Пример:

Допустим, что отношения имеют по одному кортежу $R(a, b) = \{(1, 2)\}$ и $S(a, b) = \{(1, 3)\}$, тогда е

$$\begin{aligned}\pi_a(R \bowtie S) &= \pi_a(\emptyset) = \emptyset, \\ \pi_a(R) \bowtie \pi_a(S) &= \{(1)\} \bowtie \{(1)\} = \{(1)\}.\end{aligned}$$



Законы проекции (продолжение)

6. Применение нового оператора проекции к аргументу «внутреннего» оператора выбора:

$$\pi_L(\sigma_C(R)) = \pi_L(\sigma_C(\pi_M(R))) ,$$

где

– список всех атрибутов, которые является либо входными атрибутами из списка , либо присутствуют в условии .

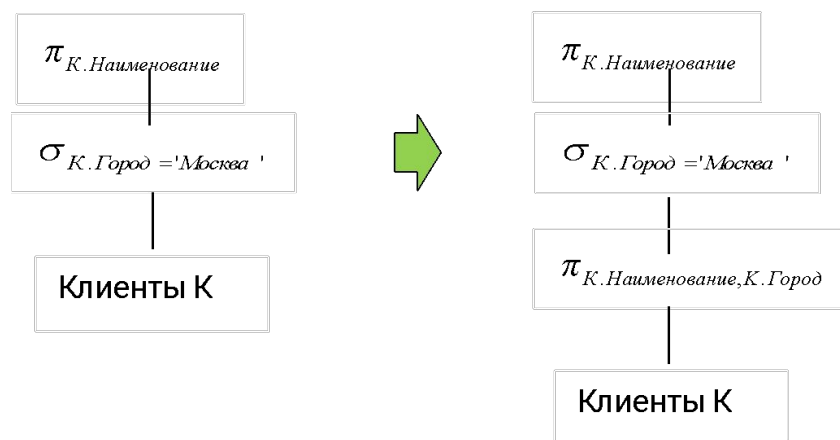
Зачастую целесообразно продвигать операторы проекции «вниз» по дереву выражений, даже если при этом исходный оператор проекции остается на месте, т.к. при выполнении проекции уменьшается размер кортежей, что снижает количество дисковых блоков, необходимых при чтении/записи промежуточных выражений.



Законы проекции (изображение дерева запроса)

Применим рассматриваемые законы к гипотетическому дереву запроса (перед выполнением операции выбора осуществим проекцию на атрибуты Город и Наименование):

```
SELECT К.Наименование  
FROM Клиенты К  
WHERE К.Город = 'Москва'
```



Если бы отношение *Клиенты* было не хранимой таблицей, а результатом промежуточной операции (например, соединения), такое преобразование имело бы смысл, т.к. сокращался бы объем входных данных для последующей операции (удалось бы опустить лишний атрибут).



Законы соединения и декартового произведения

$$R \oplus_C S = \sigma_C(R \times S),$$
$$R \bowtie_L S = \pi_L(\sigma_C(R \times S)),$$

где

C – условие сравнения значения одноименных атрибутов R и S ,

L – список атрибутов R , за которыми следуют атрибуты S , отсутствующие в R .

На практике правила применяются справа налево. Т.е. декартово произведение, к результату которого применяется оператор выбора, трактуется как определенная разновидность операции соединения, поскольку алгоритмы вычисления соединения выполняются гораздо быстрее, чем алгоритмы, реализующие декартово произведение с последующей выборкой.



Законы удаления кортежей-дубликатов

Оператор δ , осуществляющий удаление кортежей-дубликатов, можно продвинуть внутрь многих (но не всех) операторов реляционной алгебры. Перемещение δ вниз по дереву выражений позволяет сократить размеры промежуточных отношений и поэтому дает определённые выгоды.

Иногда имеет смысл разместить δ на таких позициях, где он, строго говоря, не нужен (заведомо известно, что отношение-операнд не содержит кортежей-дубликатов):

1. $\delta(R) = R$,
если R не имеет в своём составе повторяющихся кортежей.

Закон справедлив для:

- ✓ хранимых отношений, объявленных с первичным ключом;
- ✓ отношений, служащих итогом выполнения операции группировки (агрегирования), т.к. при группировке создаётся отношение, содержащее уникальные кортежи.

2. Законы, связывающие оператор δ с другими операторами:

$$\begin{aligned}\delta(R \times S) &= \delta(R) \times \delta(S), \\ \delta(R \bowtie S) &= \delta(R) \bowtie \delta(S), \\ \delta(R \underset{d}{\oplus} S) &= \delta(R) \underset{d}{\oplus} \delta(S), \\ \delta(\sigma_C(R)) &= \sigma_C(\delta(R)).\end{aligned}$$



Законы удаления кортежей-дубликатов (продолжение)

3. Законы, связывающие оператор δ с аргументами «вложенного» оператора $\bowtie$ (в версии с «мультимножествами»):

$$\delta(R \bowtie_B S) = \delta(R) \bowtie_B \delta(S) = R \bowtie_B \delta(S) = \delta(R) \bowtie_B S.$$

4. Сочетания δ с операциями $\bowtie_B$, $\setminus_B$ и π , в общем случае, запрещены.

Пример:

Пусть отношения R содержит 2 копии кортежа t , а S – одну, тогда в отношении $\delta(R \bowtie_B S)$ будет один экземпляр кортежа t , а в $\delta(R) \bowtie_B \delta(S)$ – два.

Аналогично, $\delta(R \setminus_B S)$ будет содержать один экземпляр кортежа t , а $\delta(R) \setminus_B \delta(S) = 0$.

Рассмотрим отношение $R(a, b) = \{(1, 2), (1, 3)\}$. Отношение $\delta(\pi_a(R))$ будет обладать одним экземпляром кортежа (1), а отношение $\pi_a(\delta(R))$ будет обладать двумя такими кортежами.

5. Сочетание δ с «множественными» версиями операторов $\bowtie_S$, $\bowtie_S$ и $\setminus_S$ разности лишено смысла, т.к. применение «множественной» версии этих операторов само по себе служит гарантией отсутствия кортежей-дубликатов.



Законы группировки и агрегирования

В отличие от рассмотренных ранее операторов для оператора группировки γ невозможно сформулировать некие универсальные законы. Однако некоторые общие закономерности всё-таки существуют.

1. Правило поглощения оператором γ оператора δ :

$$\delta(\gamma_L(R)) = \gamma_L(R).$$

2. Удаление ненужных атрибутов посредством π :

$$\gamma_L(R) = \gamma_L(\pi_M(R)),$$

где

M – список, содержащий, как минимум, все атрибуты R , упомянутые в списке L .

Оператор γ_L называется невосприимчивым к присутствию дубликатов, если в списке L нет других операторов агрегирования, кроме функций MIN и/или MAX .

В этом случае имеет место соотношение:

$$\gamma_L(R) = \gamma_L(\delta(R)).$$



Законы группировки и агрегирования (пример)

```
SELECT К.Наименование, П.КодГ, Max(П.Вес)  
FROM Клиенты К, Перевозки П  
WHERE К.Код_К = П.Код_К  
GROUP BY К.Наименование, П.КодГ
```

Первый логический план запроса:

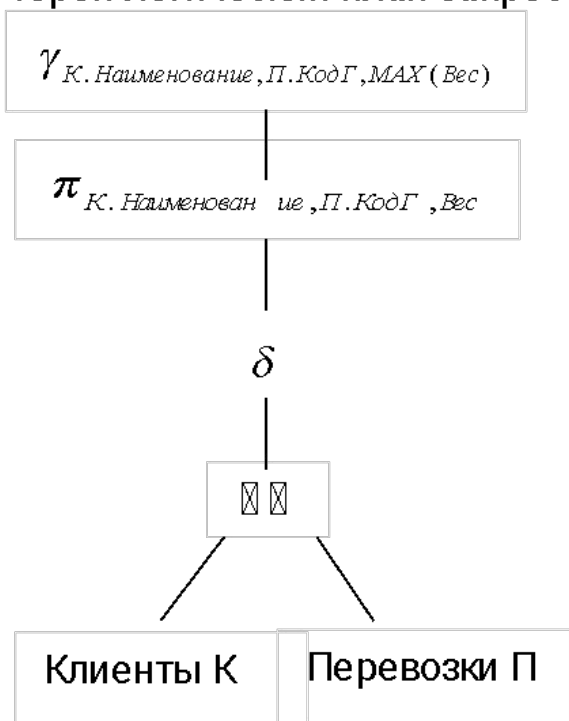




Законы группировки и агрегирования (второй логический план запроса)

1. Заменить операции выбора σ и $\times$ на $\bowtie$.
2. Вставить δ перед группировкой (т.к. в данном случае он невосприимчив к наличию дубликатов).
3. Разместить дополнительную π на атрибуты, которые присутствуют в итоговом отношении.

Второй логический план запроса:





Законы группировки и агрегирования (третий логический план запроса)

После оператора $\bowtie$ можно включить пары вершин σ и π .

Третий логический план запроса:

