Реляционная алгебра (продолжение)

Технологии баз данных. Лекция 8

Традиционные операции над множествами

Операция	Обозначение матем.	Обозначение лат. алф.
Объединение	$R1 \cup R2$	R1 **UNION** R2
Пересечение	$R1 \cap R2$	R1 **INTERSECT** R2
Разность	R1 **-** R2	R1 **MINUS** R2
Декартово произведение	R1 **×** R2	R1 **TIMES** R2

Специальные реляционные операции

Операция	Обозначение греч. алф.	Обозначение лат. алф.
Выборка (ограничение)	$\sigma_{condition}(R)$	R **WHERE** condition
Проекция	$\pi_{Attr1, \ldots, AttrN}$ (R)	R**[**Attr1, …, AttrN**]**
Естественное соединение	R1 $\bowtie$ R2	R1 **JOIN** R2
Θ-соединение	R1 $\underset{\mathbf{R1.Attr1\ \Theta\ R2.Attr2}}{\bowtie}$ R2	-
Деление	R1 ÷ R2	R1 **DIVIDE** R2

Пример 1

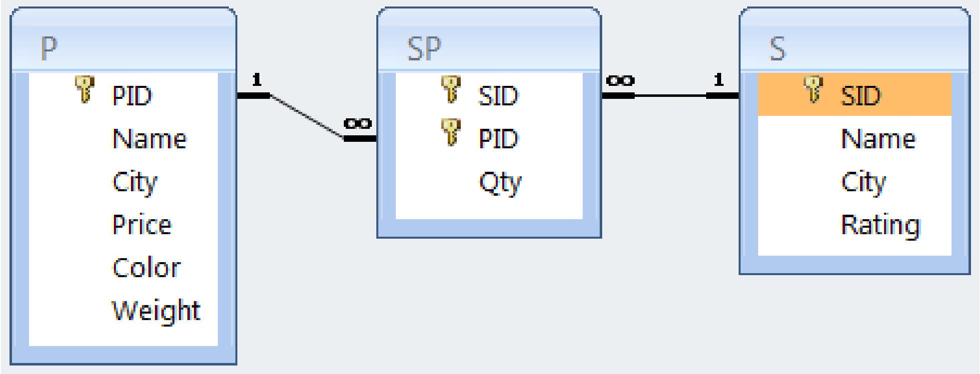
- Получить имена поставщиков, которые поставляют деталь с кодом P2.
- **SQL:**

```
select Name
from SP, S
where SP.SID=S.SID and
    PID='P2';
```

- **Операции реляционной алгебры:**

1) $R1 := SP \bowtie S$
2) $R2 := \sigma_{PID='P2'}(R1)$
3) $Result := \pi_{Name}(R2)$

P (Детали) **SP** (Поставки) **S** (Поставщики)



Результат:

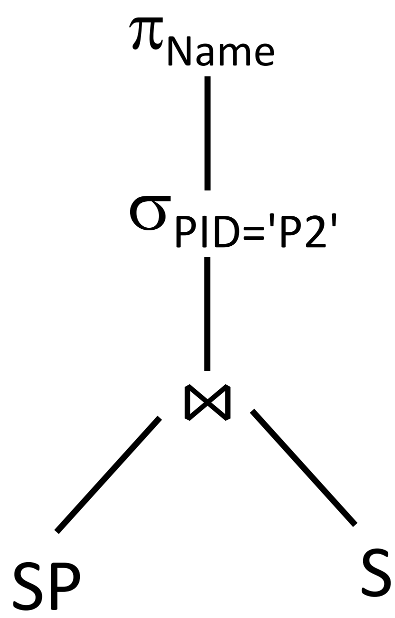
$$\pi_{Name}(\sigma_{PID='P2'}(SP \bowtie S))$$

Способы представления реляционных выражений

Линейная формула:

$\pi_{\text{Name}}(\sigma_{\text{PID='P2'}}(\text{SP} \bowtie \text{S}))$

Дерево частных выражений реляционной алгебры:



Пример 2

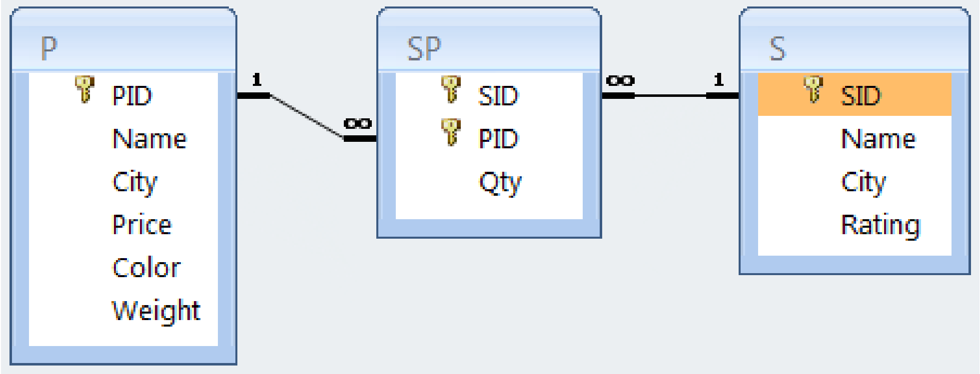
- Получить имена поставщиков, которые поставляют детали красного цвета.
- **SQL:**

```
select S.Name
from SP, S, P
where SP.SID=S.SID and
    SP.PID=P.PID and
    Color='красный';
```

- **Операции реляционной алгебры:**

1) R1 := SP ⋈ S
2) R2 := R1 ⋈ P
3) R3 := $\sigma_{Color='красный'}$(R2)
4) Result := $\pi_{S.Name}$(R3)

P (Детали) **SP** (Поставки) **S** (Поставщики)



Результат:

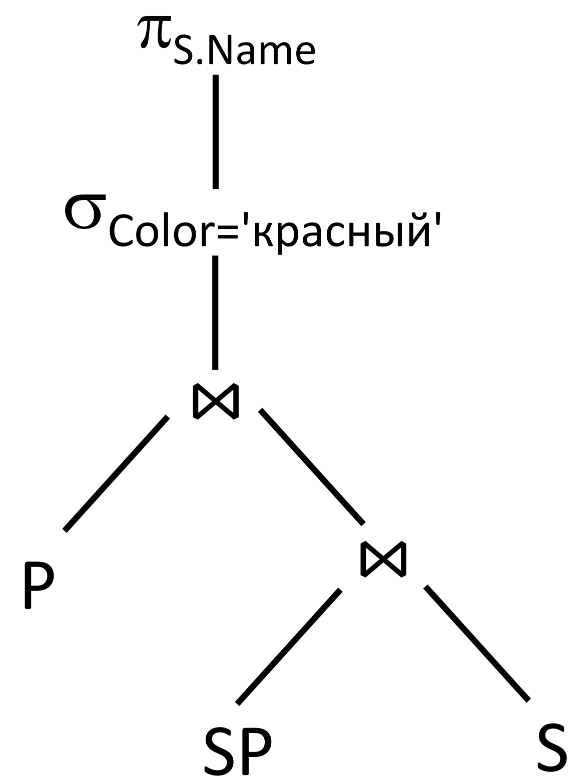
$$\pi_{S.Name}(\sigma_{Color='красный'}(P \bowtie (SP \bowtie S)))$$

Пример 2. Дерево частных выражений реляционной алгебры

Линейная формула:

$\pi_{S.Name}\,(\sigma_{Color='красный'}(P \bowtie (SP \bowtie S)))$

Дерево частных выражений реляционной алгебры:



Дополнительные реляционные операции

Операция	Обозначение греч. алф.	Обозначение лат. алф.
Удаление дубликатов	δ (R)	
Агрегирование		SUM, AVG, MIN, MAX, COUNT
Группировка	$\gamma_{\text{Attr1, func(Attr2)} \rightarrow \text{NewAttr}}$ (R)	
Сортировка	$\tau_{\text{Attr1, …, AttrN}}$ (R)	

Удаление дубликатов

- Результатом ***удаления дубликатов отношения R*** является отношение

$$\delta(R),$$

имеющее заголовок, состоящий из атрибутов отношения *R*,

и тело, в котором каждый различный кортеж *R* представлен не более чем одной копией.

R

Name	Price
Болт	40
Гайка	24
Болт	40

δ(*R*)

Name	Price
Болт	40
Гайка	24

Агрегирование

- **SUM**(Attr) – возвращает сумму значений по атрибуту Attr
- **AVG**(Attr) – вычисляет среднее арифметическое по атрибуту Attr
- **MIN**(Attr) – вычисляет минимальное значение по атрибуту Attr
- **MAX**(Attr) – вычисляет максимальное значение по атрибуту Attr
- **COUNT**(Attr) – возвращает количество значений (не обязательно различных) в определенном столбце, или же число кортежей в отношении, включая дубликаты

Строго говоря, операции агрегирования, не относятся к компетенции реляционной алгебры, но используются в контексте операции группировки.

Группировка

- Результатом ***группировки по атрибуту Attr1 отношения R с агрегацией по атрибуту Attr2*** является отношение

$$\gamma_{\text{Attr1, func(Attr2)}\rightarrow\text{NewAttr}}\ (R),$$

имеющее заголовок, состоящий из атрибутов Attr1 и NewAttr,

и тело, которое строится следующим образом:

1) все кортежи R разбиваются на группы, каждая группа включает кортежи с одинаковым значением атрибута Attr1
2) для каждой группы создается один кортеж, состоящий из значения Attr1 и результата агрегирования func(Attr2) по группе

- Attr1 – ***группирующий атрибут***
- Attr2 – ***агрегируемый атрибут***
- func – агрегирующая функция (SUM, AVG, MIN, MAX, COUNT)
- NewAttr – название атрибута для результата func(Attr2)

R

Name	Price
Болт	40
Гайка	24
Болт	33

$$\boldsymbol{\gamma_{\text{Name, SUM(Price)}\rightarrow\text{'Цена'}}\ (R)}$$

Name	Цена
Болт	73
Гайка	24

Группировка

- Группировка по атрибуту Attr1 отношения R ***с агрегацией по нескольким атрибутам***:

$$\gamma_{\text{Attr1, func(Attr2)}\rightarrow\text{NewAttr,…, func(AttrN)}\rightarrow\text{NewAttrN}}\ (R)$$

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Гайка	Челябинск	20	24
P3	Болт	Одесса	15	33

$\gamma_{\text{Name, SUM(Price)}\rightarrow\text{'Цена', COUNT(*)}\rightarrow\text{'Кол-во'}}\ (R)$

Name	Цена	Кол-во
Болт	73	2
Гайка	24	1

Группировка

- В общем виде операция группировки имеет ***более одного группирующего атрибута***:

$$\gamma_{\underbrace{\text{Attr\_G1,…, Attr\_GN}}_{\text{N группирующих атрибутов}},\ \text{func(Attr1)}\rightarrow\text{NewAttr1,…, func(AttrK)}\rightarrow\text{NewAttrK}}\ (R)$$

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Болт	Челябинск	3	20
P3	Болт	Париж	11	32
P4	Гайка	Одесса	5	25
P5	Гайка	Одесса	20	24
P6	Болт	Париж	15	33

$\gamma_{\text{Name, City, SUM(Price)}\rightarrow\text{'Цена', COUNT(*)}\rightarrow\text{'Кол-во'}}\ (R)$

Name	City	Цена	Кол-во
Болт	Париж	41	3
Болт	Челябинск	20	1
Гайка	Одесса	49	2

Группировка

- В общем виде операция группировки имеет ***более одного группирующего атрибута***:

$$\gamma_{\underbrace{\text{Attr\_G1,…, Attr\_GN,}}_{\text{N группирующих атрибутов}}\ \text{func(Attr1)}\rightarrow\text{NewAttr1,…, func(AttrK)}\rightarrow\text{NewAttrK}}(R)$$

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Болт	Челябинск	3	20
P3	Болт	Париж	11	32
P4	Гайка	Одесса	5	25
P5	Гайка	Одесса	20	24
P6	Болт	Париж	15	33

$\gamma_{\text{Name, City, SUM(Price)}\rightarrow\text{'Цена', COUNT(*)}\rightarrow\text{'Кол-во'}}(R)$

Name	City	Цена	Кол-во
Болт	Париж	41	3
Болт	Челябинск	20	1
Гайка	Одесса	49	2

группа {Болт, Париж}

Группировка

- В общем виде операция группировки имеет ***более одного группирующего атрибута***:

$$\gamma_{\underbrace{\text{Attr\_G1,…, Attr\_GN}}_{\text{N группирующих атрибутов}}\text{, func(Attr1)}\rightarrow\text{NewAttr1,…, func(AttrK)}\rightarrow\text{NewAttrK}}\ (R)$$

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Болт	Челябинск	3	20
P3	Болт	Париж	11	32
P4	Гайка	Одесса	5	25
P5	Гайка	Одесса	20	24
P6	Болт	Париж	15	33

$\gamma_{\text{Name, City, SUM(Price)}\rightarrow\text{'Цена', COUNT(*)}\rightarrow\text{'Кол-во'}}\ (R)$

Name	City	Цена	Кол-во
Болт	Париж	41	3
Болт	Челябинск	20	1
Гайка	Одесса	49	2

группа {Болт, Париж}

группа {Болт, Челябинск}

группа {Гайка, Одесса}

Группировка

$$\gamma_{\text{SUM(Price)}\rightarrow\text{'Цена'}}\,(R)$$

Если атрибут группировки не задан, то отношение целиком рассматривается как одна группа.

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Гайка	Челябинск	20	24
P3	Болт	Одесса	15	33

$\gamma_{\text{SUM(Price)}\rightarrow\text{Цена}}\,(R)$

Сколько столбцов
Сколько строк

?

Группировка

$$\gamma_{\text{SUM(Price)}\rightarrow\text{'Цена'}}\,(R)$$

Если атрибут группировки не задан, то отношение целиком рассматривается как одна группа.

R

PID	Name	City	Weight	Price
P1	Болт	Париж	15	40
P2	Гайка	Челябинск	20	24
P3	Болт	Одесса	15	33

$\boldsymbol{\gamma_{\text{SUM(Price)}\rightarrow\text{Цена}}\,(R)}$

Цена
97

Сортировка

- Результатом ***сортировки R*** является отношение

$$\tau_{\text{Attr1, …, AttrN}} (R),$$

имеющее заголовок, состоящий из атрибутов отношения *R*,

и тело, в котором представлены все кортежи отношения R, вначале упорядоченные по значениям Attr1, затем, если существуют кортежи, равные по Attr1, то они сортируются по Attr2 и т.д.

R

Name	Price
Болт	40
Гайка	24
Болт	33

$\tau_{\text{Name, Price}} (R)$

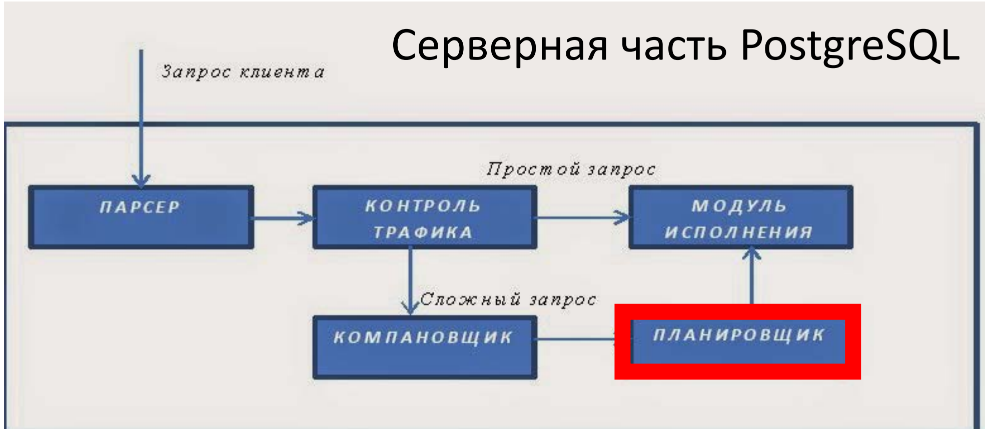
Name	Price
Болт	33
Болт	40
Гайка	24

$\tau_{\text{Price, Name}} (R)$

Name	Price
Гайка	24
Болт	33
Болт	40

Реляционная алгебра, как базис оптимизации запросов

- Зачастую ***один и тот же алгоритм вычислений допускает представление посредством нескольких различных выражений реляционной алгебры***. Одни выражения поддаются машинной обработке более эффективно, чем другие.
- Одна из основных функций оптимизатора запросов (в PostgreSQL называется «планировщик»), состоит в замене одного выражения реляционной алгебры другим, допускающим более эффективную реализацию.



PostgreSQL

Реляционная алгебра, как базис оптимизации запросов

- ***Возможность оптимизации запросов является сильной стороной реляционного подхода***.
- Благодаря тому что реляционные выражения оптимизируются на достаточно высоком семантическом уровне (на уровне операций реляционной алгебры), появляется возможность осуществления автоматической оптимизации. При этом пользователь может не задумываться над наилучшим способом выражения своих запросов. Начиная с исходного выражения, оптимизатор будет шаг за шагом применять различные правила преобразования до тех пор, пока будет получено выражение, которое, согласно встроенным в оптимизатор эвристическим правилам, будет считаться оптимальным для рассматриваемого запроса.

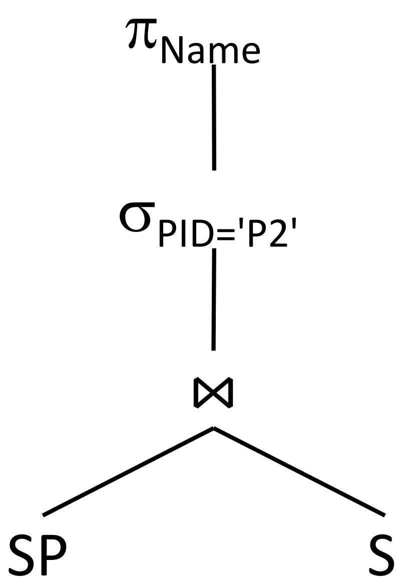
Реляционная алгебра, как базис оптимизации запросов

- Мера оценки производительности – ***количество операций ввода-вывода кортежей отношений***.
- Т.о. основная задача процесса оптимизации:
  - минимизировать количество сканируемых кортежей на каждом этапе вычисления,
  - по возможности избежать выполнения дорогостоящих операций произведений и соединения.

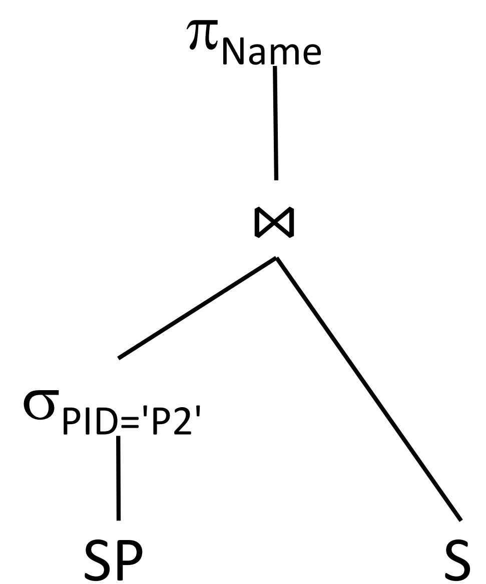
Реляционная алгебра, как базис оптимизации запросов

- Получить имена поставщиков, которые поставляют деталь с кодом P2.

1) $\pi_{Name}(\sigma_{PID='P2'}(SP \bowtie S))$ **2)** $\pi_{Name}(\sigma_{PID='P2'}(SP) \bowtie S)$



Оптимизация за счет уменьшения промежуточных отношений



Преобразование запроса в эквивалентную форму

- Операции выборки и проекции
- Распределительный закон
- Коммутативность и ассоциативность
- Идемпотентность и поглощение

Операции выборки и проекции

- Последовательность операций выборки одного и того же отношения может быть заменена единственной операцией выборки для этого отношения (причем условные выражения всех исходных операций с помощью операций AND объединяются в одно условное выражение):

$$\sigma_{CN}(\ldots(\sigma_{C2}(\sigma_{C1}(R)))\ldots) \equiv \sigma_{C1\ \mathbf{AND}\ C2\ \mathbf{AND}\ \ldots\ \mathbf{AND}\ CN}$$

- Операцию выборки для результата операции проекции можно преобразовать в операцию проекции для результата операции выборки:

$$\sigma_C(\pi_A(R)) \equiv \pi_A(\sigma_C(R))$$

Распределительный закон для унарного оператора

- ***Унарный оператор f распределяется по бинарной операции*** $\circ$ тогда и только тогда, когда для всех A и B выполняется следующее тождество:

$$f(A \circ B) \equiv f(A) \circ f(B)$$

Распределительный закон для выборки

- В реляционной алгебре операция выборки распределяется по операциям *объединения*, *пересечения* и *разности*.

$$\sigma_C (A \cup B) \equiv \sigma_C (A) \cup \sigma_C (B)$$

$$\sigma_C (A \cap B) \equiv \sigma_C (A) \cap \sigma_C (B)$$

$$\sigma_C (A - B) \equiv \sigma_C (A) - \sigma_C (B)$$

- Операция выборки распределяется по операции *естественного соединения*, но лишь тогда и только тогда, когда условие операции выборки состоит (в самом сложном случае) из двух простых условий (проверка такого условия может выполняться для определенного кортежа t без исследования любых других кортежей, кроме t) выборки, соединенных операцией AND — по одному условию выборки для каждого операнда операции естественного соединения:

$$\sigma_{C1\ \mathrm{AND}\ C2} (A \bowtie B) \equiv \sigma_{C1} (A) \bowtie \sigma_{C2} (B)$$

Т.о. операция выборки может быть выполнена раньше остальных операций, что приводит к уменьшению количества кортежей, обрабатываемых следующей операцией.

Распределительный закон для проекции

- В реляционной алгебре операция проекции распределяется по операциям *объединения* и *пересечения*.

$$\pi_C (A \cup B) \equiv \pi_C (A) \cup \pi_C (B)$$

$$\pi_C (A \cap B) \equiv \pi_C (A) \cap \pi_C (B)$$

- Операция проекции распределяется по операции *естественного соединения* при условии, что в результате операции проекции сохраняются все атрибуты соединения, поэтому справедливо следующее тождество:

$$\pi_C (A \bowtie B) \equiv \pi_C (A) \bowtie \pi_C (B)$$

Закон можно использовать для организации предварительного выполнения операций проекции, что приводит к уменьшению количества кортежей, обрабатываемых следующей операцией.

Распределительный закон для бинарной операции

- ***Бинарная операция Δ распределяется по бинарной операции ∘*** тогда и только тогда, когда для всех A , B и C истинно следующее тождество:

$$A \,\Delta\, (B \circ C) \equiv (A \,\Delta\, B) \circ (A \,\Delta\, C)$$

- В реляционной алгебре операция *объединения* распределяется по операции пересечения, а операция *пересечения* — по операции объединения:

$$A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$$

В силу этого закона операция пересечения может быть выполнена раньше операции объединения, что может привести к существенному уменьшению количества кортежей, обрабатываемых следующей операцией.

Коммутативность и ассоциативность

- Бинарная операция $\circ$ является ***коммутативной*** тогда и только тогда, когда для всех A и B истинно следующее тождество:

$$A \circ B \equiv B \circ A$$

- В реляционной алгебре коммутативными являются операции объединения, пересечения, произведения и соединения, а операции разности и деления таковыми не являются.

$$A \cup B \equiv B \cup A$$

$$A \cap B \equiv B \cap A$$

$$A \bowtie B \equiv B \bowtie A$$

$$A \times B \equiv B \times A$$

Коммутативность и ассоциативность

- Бинарная операция $\circ$ является ***ассоциативной*** тогда и только тогда, когда для всех A , B и C истинно следующее тождество:

$$A \circ (B \circ C) \equiv (A \circ B) \circ C$$

- В реляционной алгебре ассоциативными являются операции объединения, пересечения, произведения и соединения, а операции разности и деления таковыми не являются:

$$A \cup (B \cup C) \equiv (A \cup B) \cup C$$

$$A \cap (B \cap C) \equiv (A \cap B) \cap C$$

$$A \bowtie (B \bowtie C) \equiv (A \bowtie B) \bowtie C$$

$$A \times (B \times C) \equiv (A \times B) \times C$$

Идемпотентность и поглощение

- Бинарную операцию $\circ$ называют ***идемпотентной*** тогда и только тогда, когда для любого А выполняется следующее тождество:

  $$A \circ A \equiv A$$

- В реляционной алгебре операции объединения, пересечения, произведения и соединения являются идемпотентными, а операции деления и разности — нет

  $$A \cup A \equiv A$$

  $$A \cap A \equiv A$$

  $$A \bowtie A \equiv A$$

- Операции объединения и пересечения подчиняются законам поглощения:

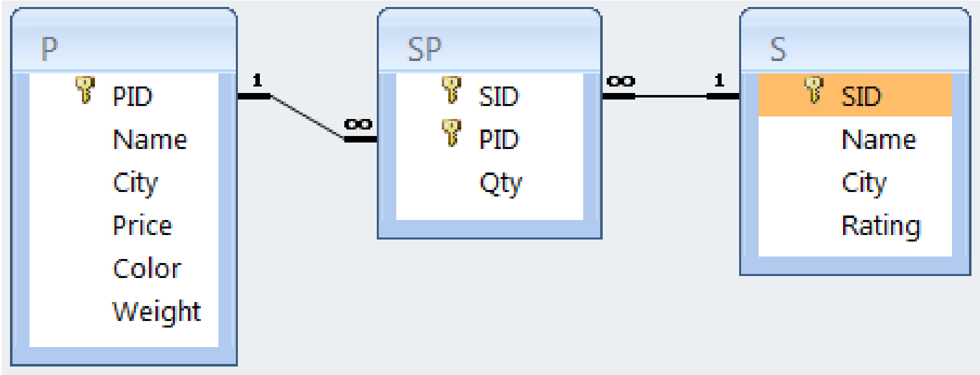
  $$A \cup (A \cap B) \equiv A$$

  $$A \cap (A \cup B) \equiv A$$

Задачи

1. Выдать полную информацию о деталях.
2. Выдать список названий и цен деталей не из Парижа с весом более 10.
3. Выдать список названий без повторений и цен деталей не из Парижа с весом более 10.
4. Выдать список названий и цен деталей (без повторений) не из Парижа с весом более 10, упорядоченный по названию детали.

P (Детали) **SP** (Поставки) **S** (Поставщики)



Задачи

5. Выдать список всех пар поставщиков и деталей, размещенных в одном городе.
6. Получить общее число поставщиков.
7. Получить код и общее количество поставки каждой детали
8. Получить коды деталей, поставляемых более чем одним поставщиком.
9. Получить имена поставщиков, которые не поставляют деталей.

P (Детали) **SP** (Поставки) **S** (Поставщики)