

Ира https://docs.google.com/document/d/1m1kSZckCkdzBroYgbjkX0KShICw0DU_qAEwsp9tIutQ/edit#

Все

<https://docs.google.com/document/d/1TWgGDnVutqhC9tBcDqaePj49vDydggiGMT1d6bcRUFk/edit#>

WHERE Description LIKE '%pasta%'

% - Знак процента представляет нулевой, один или несколько символов

_ - Подчеркнутый символ представляет собой один символ

РА

8 операторов и 2 группы

1. Традиционные операции над множествами
 - 1.1. Объединение **UNION** (БК)
 - 1.2. Пересечение **INTERSECT** (БК)
 - 1.3. Разность **MINUS** (EXCEPT) (Б)
(Для первых 3 – принцип одинаковых атрибутов)
 - 1.4. Декартово произведение **TIMES** (Б)
2. Специальные операции
 - 2.1. Проекция **PROJECT** (уменьшенный select *) (У)
 - 2.2. Фильтрация **WHERE** (У)
 - 2.3. Соединение **JOIN** расширение вширь (по сути-декартово произведение с уменьшением) (inner, а outer-декартово произведение с условием) (БК)
 - 2.4. Деление **DIVIDE BY** (Б)

RENAME x **AS** y

Join строго по совпадающим

переименования таков: (терм RENAME список- переименований). Здесь каждый из элементов списка переименований представляет собой выражение имя_атрибута AS имя_атрибута.

Count()

=, <>, IS_EMPTY (реляционное выражение)

S(Sno: integer, Sname: string, Status: integer, City: string)

P(Pno: integer, Pname: string, Color: string, Weight: real, City: string)

SP(Sno: integer, Pno: integer, Qty: integer)

Объединение 2 и выполнение условия хоть раз

Получить имена поставщиков, которые поставляют деталь под номером 2

((SP JOIN S) WHERE Pno = 2) [Sname]

Объединение 3 и выполнение условия хоть раз

Получить имена поставщиков, которые поставляют по крайней мере одну красную деталь

(((P WHERE Color = 'Красный') JOIN SP) [Sno] JOIN S) [Sname]

Другая формулировка того же запроса:

```
(( ( P WHERE Color = 'Красный' ) [ Pno ] JOIN SP ) JOIN S ) [ Sname ]
```

Деление

Получить имена поставщиков, которые поставляют все детали

```
(( SP [ Sno, Pno] DIVIDE BY P [ Pno ] JOIN S ) [ Sname ]
```

Условие Деление по другому участнику

Получить номера поставщиков, поставляющих по крайней мере все те детали, которые поставяет поставщик под номером 2

```
SP [ Sno, Pno ] DIVIDE BY ( SP WHEPE Sno = 2 ) [ Pno ]
```

Двойное по 1 таблице

Получить все пары номеров поставщиков, размещенных в одном городе

```
(( ( S RENAME Sno AS FirstSno ) [ FirstSno, City ] JOIN ( S RENAME Sno AS SecondSno ) [ SecondSno , City ] ) WHEPE FirstSno < SecondSno ) [ FirstSno, SecondSno ]
```

Отрицание условия

Получить имена поставщиков, которые не поставяют деталь под номером 2

```
((S[ Sno ] MINUS (SP WHEPE Pno = 2 ) [ Sno ] ) JOIN S ) [Sname]
```

Операция расширения

```
EXTEND S ADD 'Supplier' AS TAG
EXTEND P ADD (Weight * 454 ) AS GMWT
( EXTEND P ADD (Weight * 454 ) AS GMWT ) WHERE GMWT > Weight ( 10000.0 ) ) { ALL BUT GMWT }
EXTEND ( P JOIN SP ) ADD (Weight * Qty ) AS SHIPWT
( EXTEND S ADD City AS SCity ) { ALL BUT City }
EXTEND P ADD Weight * 454 AS GMW, Weight * 16 AS OZWT )
EXTEND S ADD COUNT ( ( SP RENAME SNo AS X ) WHERE X = SNo ) AS NP
```

Обобщение

<имя функции> (<реляционное выражение> [, <имя атрибута>])

Если параметр имеет значение COUNT, то параметр недопустим и должен быть опущен. В остальных случаях параметр может быть опущен тогда и только тогда, когда параметр задает отношение со степенью, равной единице.

Summarize

SUMMARIZE SP PER SP { PNo } ADD SUM (Qty) AS TOTQTY

В результате его вычисления создается отношение с заголовком {P#, TOTQTY}, содержащее один кортеж для каждого значения атрибута P# в проекции SP{P#}. Каждый из этих кортежей содержит значение атрибута P# и соответствующее общее количество деталей. Другими словами, концептуально исходное отношение P «перегруппировано» в множество групп кортежей (по одной группе для каждого уникального значения атрибута P#), после чего для каждой полученной группы сгенерирован один кортеж, помещаемый в окончательный результат.

В общем случае выражение выглядит: SUMMARIZE A PER B ADD <обобщение> AS

Вот еще один пример:

SUMMARIZE (P JOIN SP) PER P { City } ADD COUNT AS NSP

SUMMARIZE SP PER S { SNo } ADD COUNT AS NP

По сути, это сокращенная запись представленного ниже более сложного выражения
{ EXTEND S { SNo } ADD ((SP RENAME SNo AS X) WHERE X=SNo) AS Y, COUNT (Y)
AS NP } { SNo, NP }

1. Найти все тройки вида водительские права, тип нарушения, дата нарушения

((Drivers join DF)[FineID] join Fines)[DriverLicense, FineType, FineDate]

DriverID подразумевается

2.

- Найти все штрафы водителей, автомобили которых были зарегистрированы в 2020 году

((Cars where RegistrationDate >='01.01.2020' and RegistrationDate < '01.01.2021') join DC)[DriverID]
join Fines

Пример с лекции:

Получить имена поставщиков, которые не поставляют деталь под номером 2

((S[SNo] MINUS (SP WHERE Pno = 2) [SNo]) JOIN S) [Sname]

Наш пример

Найти автомобили, водители которых не получили ни одного штрафа

((Drivers[DriverID] minus Fines[DriverID]) join DC)[CarId] join Cars

ИК

1. Общее

RANGE OF переменная IS список-областей

переменная | переменная.атрибут [AS атрибут]

(PX.Pno, PX.Weight AS GMWT) или (SX, 'Поставщик' AS TAG)

(список-целевых-элементов)[WHERE wff]

wff ::= условие | NOT wff | условие AND wff | условие OR wff | IF условие THEN wff | **EXISTS**
переменная (wff) | **FORALL** переменная (wff) | (wff)

DISTINCT для исключения дублирующих кортежей – не надо. Вычисление аргумента expression дает отношение, из которого повторяющиеся кортежи всегда исключаются по определению. Аргумент attribute обозначает атрибут такого отношения, по которому выполняются итоговые вычисления, а дублирующие значения перед подсчетом итога из такого атрибута не удаляются.

Осторожно с агрегатными

(PX.Pno, SUM (SPX WHERE SPX.Pno = PX.Pno, Qty) AS TOTQTY)

2. Объявление

SY может принимать значения из множества кортежей S для поставщиков, которые или размещены в Смоленске, или поставляют деталь под номером 1, или и то и другое.

RANGE OF SX IS S

RANGE OF SPX IS SP

RANGE OF SY IS

(SX) WHERE SX.City = 'Смоленск'

(SX) WHERE EXISTS SPX (SPX.Sno = SX.Sno AND SPX.Pno = 1)

3. Агрегатные функции

aggregate_function (expression [, attribute]), где

- aggregate_function - это COUNT, SUM, AVG, MAX или MIN (возможны, конечно, и некоторые другие функции),
- expression - это выражение исчисления кортежей (вычисляющее отношение),
- attribute - это такой атрибут результирующего отношения, по которому подсчитывается итог.

Для функции COUNT аргумент attribute не нужен и опускается; для других итоговых функций его можно опустить. Обратите внимание, что ссылка на итоговую функцию возвращает скалярное значение и поэтому допустима в качестве операнда скалярного выражения.

aggregate_function ((tic) WHERE f [, attribute])

tic- целевой элемент списка (target_item_commalist), а f - это формула WFF,

4. Запросы

а. Общее

Просто выполнение условий - WHERE

Получить номера поставщиков из Смоленска со статусом больше 20

(SX.Sno) WHERE SX.City = 'Смоленск' AND SX.Status > 20

Дважды по одной таблице - WHERE

Получить все такие пары номеров поставщиков, что два поставщика размещаются в одном городе
(SX.Sno AS FirstSno, SY.Sno AS SecondSno) WHERE SX.City = SY.City AND SX.Sno < SY.Sno

Спецификации «AS FirstSno» и «AS SecondSno» дают имена атрибутам результата; следовательно, такие имена недоступны для использования во фразе WHERE и потому второе сравнение во фразе WHERE «SX.Sno < SY.Sno», а не «FirstSno < SecondSno».

Объединение – WHERE EXISTS A.s1=B.s1 - 2 и выполнение условия

Получить имена поставщиков, которые поставляют деталь с номером 2

SX.Sname WHERE EXISTS SPX (SPX.Sno = SX.Sno AND SPX.Pno = 2)

Объединение 3 - WHERE EXISTS A.s1=B.s1 AND EXISTS - и выполнение условия

Получить имена поставщиков, которые поставляют по крайней мере одну красную деталь

SX.Sname WHERE EXISTS SPX (SX.Sno = SPX.Sno AND EXISTS PX (PX.Pno = SPX.Pno AND PX.Color = 'Красный'))

Получить имена поставщиков, которые поставляют по крайней мере одну деталь, поставляемую поставщиком под номером 2 - значит надо дважды по объединяющей

SX.Sname WHERE EXISTS SPX (EXISTS SPY (SX.Sno = SPX.Sno AND SPX.Pno = SPY.Pno AND SPY.sno = 2))

Получить номера поставщиков, которые поставляют по крайней мере все детали, поставляемые поставщиком с номером 2

SX.Sno WHERE FORALL SPY (SPY.Sno <> 2 OR EXISTS SPZ (SPZ.Sno = SX.Sno AND SPZ.Pno = SPY.Pno))

Деление – значит FORALL или отрицание условия

Получить имена поставщиков, которые поставляют все детали

SX.SNAME WHERE FORALL PX (EXISTS SPX (SPX.Sno = SX.Sno AND SPX.Pno = PX.Pno))

SX. SNAME WHERE NOT EXISTS PX (NOT EXISTS SPX (SPX.Sno = SX.Sno AND SPX.Pno = PX.Pno))

Отрицание условия – NOT EXISTS

Получить имена поставщиков, которые не поставляют деталь под номером 2

SX.Sname WHERE NOT EXISTS SPX (SPX.Sno = SX.Sno AND SPX.Pno = 2)

b. Агрегатные

Получить номера деталей и их вес всех типов деталей, вес которых превышает 10 ед.

(PX.Pno, PX.Weight AS GMWT) WHERE PX.Weight > 10

Добавление столбца

Получить всех поставщиков, добавив для каждого литеральное значение "Поставщик"
(SX, 'Поставщик' AS TAG)

Умножение столбцов между таблицами

Получить каждую поставку с полными данными о входящих в нее деталях и общим весом поставки

(SPX.Sno, SPX.Qty, PX, PX.Weight * SPX.Qty AS SHIPWT) WHERE PX.Pno = SPX.Pno

Сумма по группе

Для каждой детали получить ее номер и общее поставляемое количество

(PX.Pno, SUM (SPX WHERE SPX.Pno = PX.Pno, Qty) AS TOTQTY)

Получить города, в которых хранится а) более пяти красных деталей;

PX.City WHERE COUNT (PY WHERE PY.City = PX.City AND PY.Color = 'Красный') > 5

Получить общее количество поставляемых деталей

(SUM (SPX, Qty) AS GRANDTOTAL)

Для каждого поставщика получить его номер и общее количество поставляемых им деталей

(SX.Sno, COUNT (SPX WHERE SPX.Sno = SX.Sno) AS NUMBER_OF_PARTS)

с. СРК

1. Найти все тройки вида водительские права, тип нарушения, дата нарушения

(DX.driverlicense, FX.type, FX.date) where FX.driverID=DX.DriverID

Права, цвет авто, модель авто

(Dx.DriverLicense, Cx.color, Cx.model) where exists DCx(DCx.Did = Dx.Did and exists Cx(DCx.Cid = Cx.Cid))

2. Найти все штрафы водителей, автомобили которых были зарегистрированы в 2020 году

RANGE OF CX IS C where C.RD >= '2020-01-01' and < '2021-01-01'

RANGE OF DX IS D

RANGE OF DCS IS DC

RANGE OF FX IS F

Fx.type where exists DCx (exists Cx (Fx.DID = DCx.DID and DC.CID = Cx.CID)

Fx.Fid where exists DX(Dx.Did = Fx.Did and exists DCx(Dcx.Did = Dx.Did and exists Cx(Cx.Cid = Dcx.Cid)))-
без промежутка

3. Найти общую сумму выписанных штрафов в 2018 году

RANGE OF FX IS F where date like '%2018%'

Sum(FX where where date like '%2018%', amount)

Найти водителей, получивших более 3х штрафов

Dx.FIO where count(Fx where Fx.Did = D.Did) > 3

SQL

Select	5	обязательно
From	1	обычно необходимо
Where	2	
Group by	3	
Having	4	
Order by	6	отступление от реляционной модели

- 1.** Имена поставщиков, которые поставляют деталь номер 3
(первый вариант лучше)

```
Select S.Snam
From S join SP on
      S.Sno=SP.Sno
Where SP.Pno=2
```

Или

```
Selet S.Sname
From S join (selet *
             From SP
             Where Pno=2) t
On S.Sno=t.Sno
```

- 2.** Имена поставщиков, поставляющих хотя бы 1 красную деталь

```
Select Sname
From S
where S.Sno in (select S.Sno
                from SP inner join P
                on P.Pno=SP.Pno
                where P.color='K') t
```

или

```
Select S.Sname
From S join SP on S.Sno=SP.Sno
      Join P on SP.Pno=P.Pno
Where P.Color='K'
```

- 3.** Пары поставщиков из одного города

```
Select fs.Sno
From S fs join S ss
```

```
On fs.City=ss.City
Where fs.Sno<ss.Sno
```

- 4.** Имена поставщиков, не поставляющих деталь 2
Из s вычесть S с Pno=2

```
Select S.Sno, S.Sname
From S except
      Select S.Sno, S.Sname
      From S join SP on S.Sno=SP.Sno
      Where SP.Pno=2
```

- 5.** Все поставщики, поставляющие все детали

```
Select S.Sno, S.Sname
From SP join S on SP.Sno=S.Sno
Group by S.Sno, S.Sname
Having Count(distinct (Pno))=(select count(*)
                               From P)
```

- 6.** Поставщик, поставляющий больше всех деталей

```
Select S.Sno, S.Sname
From SP join S on SP.Sno=S.Sno
Group by S.Sno, S.Sname
Having count(distinct(Pno))=(select max(cnt)
                              From (
                                    Select Sno, count(distinct(Pno) as cnt
                                    From SP
                                    Group by Sno) t
                              )
```

Или используя ОТВ-обобщенное табличное выражение

```
With groupSP(Sno, cnt) as
      (select Sno, Count(distinct(Pno))
      From SP
      Group by Sno)
Select S.Sno, S.Sname
From groupSP join S on groupSP.Sno=S.Sno
Where cnt=(select max(cnt)
            From groupSP)
```

Ф3

<https://github.com/Panda-Lewandowski/DataBase/wiki/Теория-проектирования-реляционных-баз-данных>

https://github.com/Sunshine-ki/BMSTU5_DB/wiki

Правила Армстронга

1. Правило рефлексивности: (В принадлежит А) следовательно $(A \rightarrow B)$
2. Правило дополнения: $(A \rightarrow B)$ следовательно $AC \rightarrow BC$
3. Правило транзитивности: $(A \rightarrow B)$ и $(B \rightarrow C)$ следовательно $(A \rightarrow C)$
4. Правило самоопределения: $A \rightarrow A$
5. Правило декомпозиции: $(A \rightarrow BC)$ следовательно $(A \rightarrow B)$ и $(A \rightarrow C)$
6. Правило объединения: $(A \rightarrow B)$ и $(A \rightarrow C)$ следовательно $(A \rightarrow BC)$
7. Правило композиции: $(A \rightarrow B)$ и $(C \rightarrow D)$ следовательно $(AC \rightarrow BD)$
8. теоремой объединения: $(A \rightarrow B)$ и $(C \rightarrow D)$ следовательно $(A(C \rightarrow B) \rightarrow BD)$.

Суперключ – покрывает все, но избыточен

Потенциальный- супер - избыточные

Первичный – потенциальный выбранный

Множество ФЗ является **неприводимым** тогда и только тогда, когда оно обладает всеми перечисленными ниже свойствами.

1. Каждая ФЗ этого множества имеет одноэлементную правую часть.
2. Ни одна ФЗ множества не может быть устранена без изменения замыкания этого множества.
3. Ни один атрибут не может быть устранен из левой части любой ФЗ данного множества без изменения замыкания множества.

1. Найти минимальное покрытие

Упражнение. Пусть дана переменная-отношение R с атрибутами A, B, C, D и следующими функциональными зависимостями. $A \rightarrow BC$ $B \rightarrow C$ $A \rightarrow B$ $AB \rightarrow C$ $AC \rightarrow D$ Найти неприводимое множество функциональных зависимостей эквивалентное данному множеству.

1. Прежде всего, следует переписать заданные ФЗ таким образом, чтобы каждая из них имела одноэлементную правую часть. $A \rightarrow B$ $A \rightarrow C$ $B \rightarrow C$ $A \rightarrow B$ $AB \rightarrow C$ $AC \rightarrow D$

Нетрудно заметить, что зависимость $A \rightarrow B$ записана дважды, так что одну из них можно удалить.

2. Затем в левой части зависимости $AC \rightarrow D$ может быть опущен атрибут C, поскольку дана зависимость $A \rightarrow C$, из которой по правилу дополнения можно получить зависимость $A \rightarrow AC$. Кроме того, дана зависимость $AC \rightarrow D$, из которой по правилу транзитивности можно получить зависимость $A \rightarrow D$. Таким образом, атрибут C в левой части исходной зависимости $AC \rightarrow D$ является избыточным.

3. Далее можно заметить, что зависимость $AB \rightarrow C$ может быть исключена, поскольку дана зависимость $A \rightarrow C$, из которой по правилу дополнения можно получить зависимость $AB \rightarrow CB$, а затем по правилу декомпозиции - зависимость $AB \rightarrow C$.

4. Наконец зависимость $A \rightarrow C$ подразумевается зависимостями $A \rightarrow B$ и $B \rightarrow C$, так что она также может быть отброшена. В результате получается неприводимое множество зависимостей.

$A \rightarrow B$ $B \rightarrow C$ $A \rightarrow D$

2. Дана переменная-отношение $R(A, B, C, D, E)$, для которой выполняется множество функциональных зависимостей $S=\{AB \rightarrow DE, C \rightarrow E, D \rightarrow C, E \rightarrow A\}$. В результате декомпозиции получена переменная-отношение $R_1(A, B, C)$. **Какие функциональные зависимости из S будут выполняться для R_1 ?**

Необходимо вычислить замыкания всех подмножеств множества $\{A, B, C\}$, кроме пустого множества и ABC . Затем, не учитывая функциональные зависимости, которые являются тривиальными и те, которые имеют D или E в правой части, получим искомое множество.

$$S=\{AB \rightarrow DE, C \rightarrow E, D \rightarrow C, E \rightarrow A\}$$

$$A^+ = A$$

$$B^+ = B$$

$$C^+ = CEA \{C \rightarrow E, E \rightarrow A\}$$

$$AB^+ = ABDEC \{AB \rightarrow DE, D \rightarrow C\}$$

$$AC^+ = ACE \{C \rightarrow E\}$$

$$BC^+ = BCEAD \{C \rightarrow E, E \rightarrow A, AB \rightarrow DE\}$$

Не учитываем D и E .

Искомое множество функциональных зависимостей: $\{C \rightarrow A, AB \rightarrow C, BC \rightarrow A\}$

(замечание: $BC \rightarrow A$ можно не учитывать, так как эта функциональная зависимость логически следует из $C \rightarrow A$)