

Таблица поставщиков (s):

n_post	name	rating	town
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Таблица поставок (sp):

n_post	n_det	date_post	kol
S1	P1	2021-02-01	300
S1	P2	2021-04-05	200
S1	P3	2021-05-12	400
S1	P4	2021-06-15	200
S1	P5	2021-07-22	100
S1	P6	2021-08-13	100
S2	P1	2021-03-03	300
S2	P2	2021-06-12	400
S3	P2	2021-04-04	200
S4	P2	2021-03-23	200
S4	P4	2021-06-17	300
S4	P5	2021-08-22	400

Таблица товаров (p):

n_det	name	cvet	ves	town
P1	Monitor screen	Red	12000	London
P2	Computer mouse	Green	150	Paris
P3	Keyboard	Blue	400	Rome
P4	Keyboard	Red	350	London
P5	Power unit	Blue	1200	Paris
P6	Computer case	Red	7500	London

Пример. Внешнее соединение внутреннего соединения с третьей таблицей.

Для получения внешнего соединения будем использовать представление (*View*). Представление можно рассматривать как хранимый запрос, на основании которого создается объект базы данных. Этот объект схож с таблицей, но в его содержимом динамически отражаются только те записи, которые были заданы при создании.

Представление создается командой *Create view*:

Create view имя_представления *as* запрос

Представление удаляется командой *Drop view*.

Create view z1 as

select sp.n_post, sp.n_det, p.name, p.cvet, p.ves

from SP

join P on SP.n_det = P.n_det

where cvet in ('Red', 'Green');

Первый оператор выполняет
внутреннее соединение таблиц SP и P

n_post	n_det	name	cvet	ves
S1	P1	Monitor screen	Red	12000
S1	P2	Computer mouse	Green	150
S1	P4	Keyboard	Red	350
S1	P6	Computer case	Red	7500
S2	P1	Monitor screen	Red	12000
S2	P2	Computer mouse	Green	150
S3	P2	Computer mouse	Green	150
S4	P2	Computer mouse	Green	150
S4	P4	Keyboard	Red	350

Select S.n_post, S.name, z1.n_det, z1.name,

z1.cvet, z1.ves from s

left join z1 on (s.n_post = z1.n_post)

Затем оператором Select выполняется
внешнее соединение как
комбинирование информации,
полученной в первом запросе, с
данными из главной таблицы S

n_post	name	n_det	name	cvet	ves
S1	Smith	P1	Monitor screen	Red	12000
S1	Smith	P2	Computer mouse	Green	150
S1	Smith	P4	Keyboard	Red	350
S1	Smith	P6	Computer case	Red	7500
S2	Jones	P1	Monitor screen	Red	12000
S2	Jones	P2	Computer mouse	Green	150
S3	Blake	P2	Computer mouse	Green	150
S4	Clark	P2	Computer mouse	Green	150
S4	Clark	P4	Keyboard	Red	350
S5	Adams	NULL	NULL	NULL	NULL

Использование значений oid-столбца

Для нахождения в таблице дубликата значения можно использовать в самосоединении скрытый oid-столбец.

Условие $x.oid \neq y.oid$, заданное в следующем примере эквивалентно следующему утверждению: строка x не является той же самой, что и строка y .

Пример.

В таблице описания деталей найти детали с одинаковым весом.

```
Select x.номер_детали, x.название, x.вес
```

```
from P x
```

```
cross join P y
```

```
where x.вес = y.вес and x.oid != y.oid
```

Подзапросы

Подзапрос – оператор *select*, вложенный в спецификатор *where* другого оператора *select* (или одного из операторов *insert*, *delete*, *update*).

- В состав каждого подзапроса должны входить спецификаторы *select* и *from*.
- Каждый подзапрос должен быть заключен в круглые скобки, чтобы указать серверу баз данных на то, что эту операцию следует выполнить первой.

Подзапросы бывают **коррелированными** и **некоррелированными**. Подзапрос является коррелированным, если его значение зависит от значения, производимого внешним оператором *select*, который содержит этот подзапрос. Любой другой вид запроса называется некоррелированным.

Важное свойство коррелированного подзапроса состоит в следующем: так как он зависит от значения результата внешнего оператора *select*, то должен выполняться повторно по одному разу для каждого значения, производимого внешним оператором *select*. Некоррелированный подзапрос выполняется только один раз.

Подзапрос включается в спецификатор where оператора select с помощью следующих ключевых слов:

- ALL
- ANY
- IN
- EXISTS

Некоррелированные подзапросы

Фраза ALL. Ключевое слово ALL, указываемое перед подзапросом используется для определения того, выполняется ли условие сравнения для каждого возвращаемого подзапросом значения. Если подзапрос не возвращает ни одного значения, то условие поиска считается выполненным.

Пример.

Получить перечень поставщиков, рейтинг которых выше рейтинга любого лондонского поставщика.

```
Select x.n_post, x.name, x.rating  
from S x where x.rating > all  
(select y.rating  
from S y  
where y.town='London')
```

n_post	name	rating
S3	Blake	30
S5	Adams	30

Сначала выполняется независимый внутренний подзапрос, его результатом является выборка (20, 20), затем - внешний запрос.

Фраза ANY. Ключевое слово ANY, указываемое перед запросом, используется для определения того, выполняется ли сравнение по крайней мере для одного значения, возвращаемого подзапросом. Если подзапрос не возвращает ни одного значения, то условие поиска считается не выполненным.

Пример.

Получить перечень поставщиков, рейтинг которых выше рейтинга хотя бы одного парижского поставщика.

```
Select x.n_post, x.name, x.rating  
from S x  
where x.rating > any  
(select y.rating  
from S y  
where y.town = 'Paris')
```

n_post	name	rating
S1	Smith	20
S3	Blake	30
S4	Clark	20
S5	Adams	30

Сначала выполняется независимый внутренний подзапрос, его результатом является выборка (10, 30), затем - внешний запрос.

Фраза IN. Простой подзапрос.

Пример.

Выдать фамилии поставщиков, поставляющих деталь под номером P2.

Select name

from S

where n_post in

(Select n_post from SP where n_det ='P2')

name

Smith

Jones

Blake

Clark

Сначала выполняется внутренний подзапрос, его результатом является выборка (S1, S2, S3, S4), затем - внешний запрос, который после подстановки результатов внутреннего подзапроса имеет вид:

Select name

from S

where n_post in ('S1', 'S2', 'S3', 'S4')

Подзапрос с несколькими уровнями вложенности

Пример.

Выдать фамилии поставщиков, поставляющих по крайней мере одну красную деталь.

```
Select name from S
where n_post in (Select n_post from SP
where n_det in (select n_det from P
where cvet='Red'))
```

name
Smith
Jones
Clark

Сначала осуществляется самый внутренний подзапрос, дающий выборку (P1, P4, P6). После подстановки его результатов выполняется второй по вложенности подзапрос, дающий выборку (S1, S2, S4). Подстановка результатов второго выполненного подзапроса во внешний запрос приводит к окончательному результату.

Использование одной и той же таблицы в подзапросе внешнем запросе

Пример.

Выдать номера поставщиков, поставляющих, по крайней мере, одну деталь, поставляемую поставщиком S2.

```
Select distinct n_post from SP spx
where spx.n_det in
(Select spy.n_det from SP spy
where spy.n_post='S2')
```

n_post
S1
S2
S3
S4

Сначала выполняется внутренний подзапрос, дающий выборку (P1, P2). Подстановка его результатов во внешний запрос приводит к окончательному результату.

Подзапрос с оператором сравнения отличным от IN

Пример.

Выдать номера поставщиков, находящихся в том же городе, что и поставщик S1

```
Select n_post  
from S  
where town = (Select town  
from S  
where n_post = 'S1')
```

n_post
S1
S4

Сначала выполняется внутренний подзапрос, дающий единственное значение "Лондон". Подстановка его результатов во внешний запрос приводит к окончательному результату.

Коррелированные подзапросы. Коррелированный подзапрос с использованием в коррелированном и внешнем запросе одной и той же таблицы.

Пример. Выдать номера деталей, поставляемых более чем одним поставщиком.

```
Select distinct spx.n_det  
from SP spx  
where spx.n_det in  
(Select spy.n_det  
from SP spy  
where spy.n_post<>spx.n_post)
```

n_det
P1
P2
P4
P5

Квантор существования EXISTS.

В языке SQL предикат с квантором существования представляется выражением вида:

EXISTS (select * from...)

Данное выражение истинно тогда и только тогда, когда результат вычисления подзапроса, представленного с помощью **select * from** является непустым множеством, т.е. когда существует какая-либо запись в таблице, указанной во фразе **from** подзапроса, который удовлетворяет условию **where** этого подзапроса.

Пример. Выдать фамилии поставщиков, поставляющих деталь P2.

Select name from S where exists

(Select * from SP where n_post = S.n_post and n_det = 'P2')

Последовательность обработки запроса:

- выбирается первая строка из S (номер_поставщика='S1');
- поскольку условие

номер_поставщика = 'S1' и номер_детали = 'P2' - истина, результат обработки запроса для первой строки - фамилия Смит.

name
Smith
Jones
Blake
Clark

Запрос, реализующий квантор общности

Квантор общности FORALL в SQL не поддерживается, однако он может быть выражен через квантор существования при помощи тождества

$$\text{FORALL } x(p) = \text{NOT}(\text{EXISTS } x(\text{NOT}(p))).$$

Пример

Выдать фамилии поставщиков, которые поставляют все детали.

Эквивалентная формулировка задачи может звучать так:

Выдать фамилии поставщиков таких, что для всех деталей существует запись в таблице SP, указывающая, что данный поставщик поставляет эту деталь.

Последнее утверждение, в свою очередь, эквивалентно следующему: выдать фамилии поставщиков таких, что не существует детали такой, что не существует записи в таблице SP, указывающей, что данный поставщик поставляет эту деталь.

Select name from S where not exists

(Select * from P where not exists

(Select * from SP where n_post=S.n_post

and n_det=P.n_det))

name
Smith

Использование функций в подзапросе.

Пример

Выдать номера поставщиков со значением поля рейтинг меньшим, чем максимальный рейтинг в таблице S.

**Select номер_поставщика from S
where рейтинг < (Select max(рейтинг) from S)**

Результат: номер_поставщика
 S1
 S2
 S4

Пример

Выдать номер_поставщика, рейтинг и город всех поставщиков, у которых рейтинг больше либо равен среднему для их конкретного города (использование функций в коррелированном подзапросе).

**Select номер_поставщика, рейтинг, город
from S sx
where рейтинг >= (Select avg(рейтинг)
from S sy
where sy.город=sx.город)**

Результат: номер_поставщика Рейтинг Город
 S1 20 Лондон
 S3 30 Париж
 S4 20 Лондон
 S5 30 Атенс