

Лекция № 15.

Примеры SQL-запросов

Примеры SQL-запросов

Рассмотрим несколько групп примеров SQL-запросов.

В этих примерах мы будем использовать базу данных, состоящую из 3 таблиц: S (поставщики), P (детали) и SP (поставки деталей).

Замечание.

Во всех примерах сначала будем писать запрос на естественном (русском) языке, а затем – его формальное представление и, в случае необходимости, результат выполнения запроса.

Примеры SQL-запросов

Таблица *S*. *Поставщики*

Номер_Поставщика	Фамилия	Состояние	Город
S1	Смит	20	Лондон
S2	Джонс	10	Париж
S3	Блейк	30	Париж
S4	Кларк	20	Лондон
S5	Адамс	30	Афины

В таблице *S* каждый поставщик имеет уникальный номер, фамилию, значение рейтинга или состояние и местонахождение (город). Первичный ключ таблицы – номер поставщика.

Примеры SQL-запросов

Таблица *P. Детали*

Номер_Детали	Название	Цвет	Вес	Город
P1	гайка	красный	12	Лондон
P2	болт	зеленый	17	Париж
P3	винт	голубой	17	Рим
P4	винт	красный	14	Лондон
P5	кулачок	голубой	12	Париж
P6	блюдом	красный	19	Лондон

В таблице *P* каждый вид детали имеет уникальный номер, название, цвет, вес и местонахождение (город). Первичный ключ этой таблицы – номер детали.

Примеры SQL-запросов

Таблица *SP. Поставщики – детали*

Номер_Поставщика	Номер_Детали	Количество
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Примеры SQL-запросов

Таблица SP связывает детали из таблицы P с поставщиками из таблицы S . Для каждой поставки имеется номер поставщика, номер детали и количество деталей.

Первичный ключ образуют два атрибута – номер поставщика и номер детали.

1. Запросы к одной таблице

Рассмотрим сначала примеры, включающие запросы к одной таблице.

Запросы к одной таблице

1.1 Простая выборка

"Выдать номера всех поставляемых деталей".

SELECT Номер_Детали FROM SP

Результат: вернуть столбец из *SP* с именем *Номер_Детали* (с повторяющимися номерами).

Номер_Детали
P1
P2
P3
P4
P5
P6
P1
P2
P2
P2
P4
P5

Запросы к одной таблице

1.2 Выборка с исключением дубликатов

"Выдать номера всех поставляемых деталей, исключая дубликаты".

SELECT DISTINCT Номер_Детали FROM SP

Результат:

Номер_Детали
P1
P2
P3
P4
P5
P6

Запросы к одной таблице

1.3 Выборка вычисляемых значений

"Выдать номера и вес каждой детали в граммах, предполагая, что в таблице P веса деталей даны в фунтах".

*SELECT Номер_Детали, "Вес в граммах=", Вес*454 FROM P*

Результат:

Номер_Детали		
P1	Вес в граммах=	5448
P2	Вес в граммах=	7718
-- --	-- -- -- -- -- -- --	-- --
P6	Вес в граммах=	8626

Запросы к одной таблице

1.4 Простая выборка "SELECT *"

"Выдать полные характеристики всех поставщиков"

*SELECT * FROM S*

Результат: полная копия таблицы S.

1.5 Ограниченная выборка

"Выдать номера поставщиков, которые находятся в Париже и имеют состояние больше 20".

SELECT Номер_Поставщика FROM S

WHERE Город="Париж" AND Состояние>20

Результат:

Номер_Поставщика
S3

Запросы к одной таблице

1.6 Выборка с упорядочением

1.6.1 "Выдать номера и состояния поставщиков, находящихся в Париже, в порядке убывания их состояний"

```
SELECT Номер_Поставщика, Состояние FROM S  
WHERE Город = "Париж"  
ORDER BY Состояние DESC
```

либо

```
SELECT Номер_Поставщика, Состояние FROM S  
WHERE Город = "Париж"  
ORDER BY 2 DESC
```

Результат:

Номер_Поставщика	Состояние
S3	30
S2	10

Запросы к одной таблице

1.6.2 "Выдать номера и вес каждой детали в граммах, предполагая, что в таблице Р веса деталей даны в фунтах. Результат упорядочить по возрастанию номера детали в рамках возрастания веса в граммах"

```
SELECT Номер_Детали, "Вес в граммах=", Вес*454 FROM P  
ORDER BY 3, Номер_Детали
```

Предупреждение.

Поле в разделе *ORDER BY* должно включать столбцы результирующей таблицы, иначе будет выдана ошибка.

Нельзя, например, написать:

```
SELECT Номер_Поставщика FROM S  
ORDER BY Город
```

Запросы к одной таблице

1.7 Выборка с использованием BETWEEN

"Выдать сведения о деталях, вес которых находится в диапазоне от 16 до 19 включительно".

```
SELECT Номер_Детали, Название, Цвет, Вес, Город FROM P  
WHERE Вес BETWEEN 16 AND 19
```

Результат: очевиден.

1.8 Выборка с использованием IN

"Выдать сведения о деталях зеленого и красного цвета".

```
SELECT Номер_Детали, Название, Цвет, Вес, Город FROM P  
WHERE Цвет IN ("красный", "зеленый")
```

Результат: очевиден.

Запросы к одной таблице

1.9 Выборка с использованием предиката LIKE

"Выдать название и номера деталей, у которых название заканчивается на букву 'm' ".

```
SELECT Номер_Детали, Название FROM P  
WHERE Название LIKE "%m"
```

Результат:

Номер_Детали	Название
P2	болт
P3	винт
P4	винт

Запросы к нескольким таблицам

2. Запросы к нескольким таблицам

2.1 Простое эквисоединение

"Выдать сведения о таких поставщиках и деталях, которые размещены в одном и том же городе".

```
SELECT S.*, P.*  
FROM S, P  
WHERE S.Город = P.Город
```

Результат: таблица, полученная путем соединения таблиц *S* и *P* по значению атрибута *Город*.

Запросы к нескольким таблицам

2.2 Соединение двух таблиц с дополнительным условием

"Выдать сведения о таких поставщиках и деталях, которые размещены в одном и том же городе и их состояние больше 20".

```
SELECT S.*, P.*  
FROM S, P  
WHERE S.Город = P.Город AND S.Состояние > 20
```

Результат: ограничение результирующей таблицы из предыдущего примера.

Запросы к нескольким таблицам

2.3 Соединение трех таблиц

"Выдать информацию о поставщиках и деталях, размещенных в одном городе, и количество деталей больше 100".

```
SELECT S.Номер_Поставщика, P.Номер_Детали, SP.Количество  
FROM S, P, SP
```

```
WHERE S.Город = P.Город AND
```

```
P.Номер_Детали = SP.Номер_Детали AND
```

```
SP.Количество > 100
```

Результат:

Номер_Поставщика	Номер_Детали	Количество
S1	P1	300
S1	P4	200
S2	P2	400
S3	P2	200
S4	P4	300

Запросы к нескольким таблицам

2.4 Соединение таблицы с ней самой

"Выдать все пары поставщиков, находящихся в одном городе".

```
SELECT ПЕРВАЯ.Номер_Поставщика  
       ВТОРАЯ.Номер_Поставщика  
FROM S ПЕРВАЯ, S ВТОРАЯ  
WHERE ПЕРВАЯ.Город = ВТОРАЯ.Город AND  
       ПЕРВАЯ.Номер_Поставщика < ВТОРАЯ.Номер_Поставщика
```

Результат:

Номер_Поставщика	Номер_Поставщика
S1	S4
S2	S3

Использование подзапросов

3. Использование подзапросов

3.1 Простой подзапрос

"Выдать фамилии поставщиков, которые поставляют деталь P2".

```
SELECT Фамилия  
FROM S  
WHERE Номер_Поставщика IN  
      (SELECT Номер_Поставщика  
      FROM SP  
      WHERE Номер_Детали = "P2")
```

Результат:

Фамилия
Смит
Джонс
Блейк
Кларк

Замечание. Этот же результат можно получить путем соединения таблиц.

Использование подзапросов

3.2 Подзапрос с несколькими уровнями вложенности

"Выдать фамилии поставщиков, которые поставляют по крайней мере одну красную деталь".

```
SELECT Фамилия
FROM S
WHERE Номер_Поставщика IN
    (SELECT Номер_Поставщика
     FROM SP
     WHERE Номер_Детали IN
         (SELECT Номер_Детали
          FROM P
          WHERE Цвет = "красный" ) )
```

Результат:

Фамилия
Смит Джонс Кларк

Использование подзапросов

3.3 Использование одной и той же таблицы в запросе и подзапросе

"Выдать номера поставщиков, которые поставляют по крайней мере одну деталь, поставляемую поставщиком S2".

SELECT Номер_Поставщика

FROM SP

WHERE Номер_Детали IN

(SELECT Номер_Детали

FROM SP

WHERE Номер_Поставщика = "S2")

Результат:

Номер_Поставщика
S1
S2
S3
S4

4. Использование квантора существования EXISTS

"Выдать фамилии поставщиков детали P1".

SELECT Фамилия

FROM S

WHERE EXISTS

*(SELECT **

FROM SP

WHERE Номер_Поставщика = S.Номер_Поставщика

AND Номер_Детали = "P1")

Результат:

Фамилия
СМИТ ДЖОНС

5. Использование кванторов EXISTS и ALL

"Выдать фамилию поставщика с максимальным состоянием".

```
SELECT DISTINCT a.Фамилия, a.Состояние
FROM S a
WHERE NOT EXISTS
    (SELECT *
     FROM S b
     WHERE a.Состояние < b.Состояние)
```

либо

```
SELECT DISTINCT a.Фамилия, a.Состояние
FROM S a
WHERE a.Состояние >= ALL
    (SELECT b.Состояние
     FROM S b)
```

Результат:

Фамилия
Адамс

6. Использование функций в запросе

1. "Выдать фамилию поставщика с максимальным состоянием".

SELECT Фамилия

FROM S

*WHERE Состояние = (SELECT MAX(Состояние)
FROM S)*

Результат:

Фамилия
Адамс

6. Использование функций в запросе

2. "Выдать общее количество поставляемых деталей P2".

```
SELECT SUM(Количество)  
FROM SP  
WHERE Номер_Детали = "P2"
```

Результат:

1000

6. Использование функций в запросе

3. "Выдать общее количество поставщиков".

SELECT COUNT() FROM S*

Результат:

5

4. "Выдать общее количество поставщиков, поставляющих детали в настоящее время".

SELECT COUNT (DISTINCT Номер_Поставщика) FROM SP

Результат:

4

6. Запросы с группированием данных

1. "Вычислить общий объем поставок для каждой детали “.

SELECT Номер_Детали, SUM(Количество)

FROM SP

GROUP BY Номер_Детали

Результат:

Номер_Детали	
P1	600
P2	1000
P3	400
P4	500
P5	500
P6	100

6. Запросы с группированием данных

2. "Выдать номера всех деталей, поставляемых более чем одним поставщиком “.

```
SELECT Номер_Детали FROM SP  
GROUP BY Номер_Детали  
HAVING COUNT(*) > 1
```

Результат:

Номер_Детали
P1
P2
P4
P5