



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО

ОБРАЗОВАНИЯ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ
ТЕХНОЛОГИЙ»**

Факультет Управление и информатика в технологических системах

Кафедра Информационная безопасность

**Специальность 10.05.03 «Информационная безопасность автоматизированных
систем»**

Язык SQL

Выполнил студент гр. УБ-01
Зенищева Дарья Леонидовна

Воронеж – 2023

Чтение заданных столбцов из одиночной таблицы

Для того, чтобы вывести выполнить запрос для выборки используем оператор SELECT. Выведем id руководителя, его ФИО и стаж работы из таблицы rykovoditel.

```
domcult=# SELECT id,fio,stajwork
domcult=# FROM rykovoditel;
```

id	fio	stajwork
1	Melnikova Ksenia Vitalievna	1
2	Ivanova Sofia Ivanovna	2
3	Sapsay Ivan Alexeyevich	4

(3 строки)

Чтение заданных строк из одиночной таблицы

Ранее рассмотренные SQL-запросы выбирали определенные столбцы всех строк таблицы. Теперь рассмотрим запросы, позволяющие выбирать столбцы определенных строк. Следующий запрос

получает все столбцы из тех строк таблицы deti, которые содержат сведения о кружке по рисованию:

```
SELECT id_deti, fio
```

```
FROM deti
```

```
WHERE id_kryg = '1';
```

```
domcult=# SELECT id_deti,fio
domcult=# FROM deti
domcult=# WHERE id_kryg='1';
```

id_deti	fio
3	Grishina Marya Romanovna

(1 строка)

Чтение заданных строк и столбцов из одиночной таблицы

Можно выбирать из таблицы определенные столбцы и определенные строки.

```
SELECT kolvo_mest, stoimost
```

```
FROM krygi
```

```
WHERE stoimost = '1700 p';
```

```
domcult=# SELECT kolvo_mest, stoimost
domcult=# FROM krygi
domcult=# WHERE stoimost = '1700 p';
 kolvo_mest | stoimost
-----+-----
          10 | 1700 p
(1 строка)
```

Еще одна форма предложения WHERE предполагает задания списка значений, которые может иметь столбец. Это делается с помощью ключевого слова IN:

```
SELECT kolvo_mest, stoimost
```

```
FROM krygi
```

```
WHERE stoimost IN ('1700 p', '2000 p');
```

```
domcult=# SELECT kolvo_mest, stoimost
domcult=# FROM krygi
domcult=# WHERE stoimost IN ('1700 p', '2000 p');
 kolvo_mest | stoimost
-----+-----
          10 | 1700 p
          16 | 2000 p
(2 строки)
```

Диапазоны, специальные символы и пустые значения в предложениях WHERE

В предложениях WHERE могут также указываться диапазоны и шаблоны поиска. Для задания диапазонов используется ключевое слово BETWEEN.

```
SELECT naimenovaniokr, kolvo_mest
```

```
FROM krygi
```

```
WHERE kolvo_mest BETWEEN 10 and 16;
```

```
domcult=# SELECT naimenovaniokr, kolvo_mest
domcult=# FROM krygi
domcult=# WHERE kolvo_mest BETWEEN 10 and 16;
 naimenovaniokr | kolvo_mest
-----+-----
      izo       |          14
      music     |          10
      dance     |          16
(3 строки)
```

Для задания шаблонов поиска в SQL используется ключевое слово LIKE. Символ подчеркивания ('_') представляет любой одиночный символ. С его помощью можно находить значения, удовлетворяющие заданному шаблону:

```
SELECT *
FROM krygi
WHERE naimenovaniekr LIKE 'i_o';
```

```
domcult=# SELECT *
domcult=# FROM krygi
domcult=# WHERE naimenovaniekr LIKE 'i_o';
 id_kryg | naimenovaniekr | id_rykovod | kabinet | kolvo_mest | stoimost | time
-----+-----+-----+-----+-----+-----+-----
      1 | izo             |          2 |      31  |          14 | 1500p    | 15:00:00
(1 строка)
```

Символ процента ('%') представляет последовательность из одного или нескольких произвольных символов.

```
SELECT *
FROM rykovoditel
WHERE fio LIKE '%a';
```

```
domcult=# SELECT *
domcult=# FROM rykovoditel
domcult=# WHERE fio LIKE '%a';
 id | fio | birthdate | adress | phone | datawork | stajwork
-----+-----+-----+-----+-----+-----+-----
  1 | Melnikova Ksenia Vitalievna | 1990-02-01 | yl.Chekiniva dom 1 | 8911234567 | 2022-03-01 | 1
  2 | Ivanova Sofia Ivanovna | 1993-12-11 | yl.Vekiniva dom 12 | 8912434567 | 2021-02-13 | 2
(2 строки)
```

Сортировка результатов

Порядок строк в таблице, возвращаемой оператором SELECT, является произвольным. Если нужно отсортировать строки результата, это можно сделать с помощью конструкции ORDER BY.

```
SELECT fio, phone
FROM rykovoditel
ORDER BY fio;
```

```
domcult=# SELECT fio,phone
domcult=# FROM rykovoditel
domcult=# ORDER BY fio;
      fio | phone
-----+-----
 Ivanova Sofia Ivanovna | 8912434567
 Melnikova Ksenia Vitalievna | 8911234567
 Sapsay Ivan Alexeyevich | 8912432367
(3 строки)
```

По умолчанию сортировка в SQL производится в порядке возрастания. Для явного указания порядка сортировки можно использовать ключевые слова ASC (по возрастанию) и DESC (по убыванию).

```
SELECT fio, phone
FROM rykovoditel
ORDER BY fio DESC;
```

```
domcult=# SELECT fio, phone
domcult=# FROM rykovoditel
domcult=# ORDER BY fio DESC;
      fio      | phone
-----+-----
Sapsay Ivan Alexeyevich | 8912432367
Melnikova Ksenia Vitalievna | 8911234567
Ivanova Sofia Ivanovna | 8912434567
```

Агрегатные функции SQL

В SQL имеется пять агрегатных функций: COUNT, SUM, AVG, MAX и MIN. Они выполняют различные действия над результатами оператора SELECT. Функция COUNT работает вне зависимости от типа данных столбца, а функции SUM, AVG, MAX и MIN оперируют только числовыми столбцами (integer, numeric и т. д.).

Функция COUNT подсчитывает количество строк в результате, а функция SUM вычисляет сумму значений числового столбца.

```
SELECT COUNT(*)
FROM krygi;
```

```
count
-----
      3
(1 строка)
```

```
SELECT MIN(kolvo_mest), MAX(kolvo_mest), SUM(kolvo_mest)
FROM krygi;
```

```
min | max | sum
-----+-----
  10 |  16 |  40
(1 строка)
```

Чтение данных из нескольких таблиц с применением вложенных запросов

Все запросы, рассмотренные ранее, считывали данные из одиночной таблицы. Бывает, однако, что для получения требуемой информации необходимо обработать более одной таблицы.

```
SELECT naimenovaniokr
FROM krygi
WHERE id_kryg IN(
SELECT DISTINCT id_kryg
FROM krygi
WHERE kolvo_mest > 10);
```

```
naimenovaniokr
-----
izo
dance
(2 строки)
```

Чтение данных из нескольких таблиц с помощью операции соединения

Вложенные запросы подходят для обработки нескольких таблиц до тех пор, пока результаты (столбцы в предложении SELECT) относятся к одной и той же таблице. Если же нам нужно извлечь данные из двух или более таблиц, при помощи вложенного запроса это сделать не удастся. Вместо этого необходимо использовать операцию соединения (JOIN).

Основная идея здесь – создать новое отношение, связав между собой содержимое двух или более исходных отношений.

```
SELECT fio, kolvo_mest
FROM rykovoditel, krygi
WHERE rykovoditel.id=krygi.id_rykovod;
```

```
      fio      | kolvo_mest
-----+-----
Ivanova Sofia Ivanovna |      14
Melnikova Ksenia Vitalievna |      10
Sapsay Ivan Alexeyevich |      16
(3 строки)
```

В стандарте SQL-92 введен альтернативный синтаксис соединения, ставший весьма популярным, поскольку его легче интерпретировать. Здесь вместо WHERE используются ключевые слова JOIN и ON:

```
SELECT fio, kolvo_mest
FROM rykovoditel JOIN krygi
ON rykovoditel.id=krygi.id_rykovod;
```

fio	kolvo_mest
Ivanova Sofia Ivanovna	14
Melnikova Ksenia Vitalievna	10
Sapsay Ivan Alexeyevich	16
(3 строки)	

Объединения

Иногда возникает ситуация, когда необходимо объединить результаты двух и более запросов в единую таблицу, состоящую из схожих строк. При этом, в отличие от операции соединения, операция объединения связывает схожие по структуре отношения, присоединяя одну таблицу к другой снизу. Для выполнения объединения все отношения, участвующие в данном процессе, должны удовлетворять двум ограничениям:

1 Количество и порядок столбцов в объединяемых запросах должны совпадать.

2 Типы данных в соответствующих столбцах должны быть совместимы.

Одним из возможных вариантов использования операции объединения является ситуация, когда для увеличения быстродействия в базе данных существует несколько идентичных по структуре таблиц, содержащих данные за различный период времени.

Для объединения отношений используется оператор UNION.

```
SELECT naimenovaniekr, stoimost
FROM krygi
WHERE id_kryg=1
UNION
SELECT naimenovaniekr, stoimost
FROM krygi
WHERE id_rykovod=2;
```

naimenovaniekr	stoimost
izo	1500p

(1 строка)

FROM krygi

WHERE id_kryg=1

UNION

SELECT naimenovaniekr, stoimost

FROM krygi

WHERE id_rykovod=3;

naimenovaniekr	stoimost
dance	2000 p
izo	1500p

(2 строки)

Средства модификации данных языка SQL

Модификация данных подразумевает три возможных операции: вставка, изменение и удаление. В этом разделе показывается, как выполняются эти операции средствами языка SQL.

Вставка данных, оператор INSERT

Для вставки данных в таблицы служит SQL-оператор INSERT. Этот оператор имеет две формы, в зависимости от того, всем ли столбцам таблицы присваиваются значения. Если указываются значения всех столбцов, то оператор имеет следующий вид:

INSERT INTO группа VALUES

(4,20);

INSERT 0 1

id_gryp	kolvodet
1	8
2	10
3	14
4	20

(4 строки)

Изменение данных, оператор UPDATE

Значение существующих данных могут быть изменены с помощью SQL-оператора UPDATE.

UPDATE grypa

SET kolvodet=15

WHERE id_gryp =4;

UPDATE 1

```
domcult=# SELECT * FROM grypa;
 id_gryp | kolvodet 
-----+-----
      1 |         8
      2 |        10
      3 |        14
      4 |        15
(4 строки)
```

Удаление данных, оператор DELETE

Для удаления строк служит SQL-оператор DELETE.

DELETE

FROM grypa

WHERE id_gryp=4;

DELETE 1

```
domcult=# SELECT * FROM grypa;
 id_gryp | kolvodet 
-----+-----
      1 |         8
      2 |        10
      3 |        14
(3 строки)
```