

Урок 1.2. Условный оператор WHERE

Введение WHERE	2
Задание для закрепления	4
Оператор поиска по маске	6
Задание для закрепления	7
Операторы множественного выбора	8
Задание для закрепления	9
Значение NULL	10
Задание для закрепления	11

Введение WHERE

SQL-запросы, извлекающие из таблицы все строки, полезны при просмотре базы данных и создании отчетов, однако редко применяются для чего-нибудь еще.

Обычно требуется выбрать из таблицы и включить в результаты запроса только несколько строк.



Оператор WHERE - это оператор, который используется, чтобы указать, какие строки нужно отобрать.

Оператор WHERE состоит из ключевого слова WHERE, за которым следует условие отбора, определяющее, какие именно строки требуется извлечь.

Шаблон:

```
None
SELECT column1, column2
FROM table
WHERE column3 УСЛОВИЕ
```

Важные правила использования WHERE:

- В условиях фильтрации могут применяться любые комбинации констант, литералов, функций и имен полей, а также операторов, возвращающих логическое значение:
- Предложение WHERE не является обязательным. Однако если оно присутствует, то должно следовать после предложения FROM.
- Предложение WHERE может включать до 40 выражений, соединенных логическими операторами, такими как AND и OR. При формировании условий фильтрации могут быть задействованы арифметические, логические и операторы сравнения. Если в выражении присутствует несколько операторов, значения его компонентов вычисляются в установленном порядке.

- Если выражение содержит операторы разных типов, то сперва выполняются арифметические операции, затем операции сравнения, а в последнюю очередь — логические операции. Все операторы сравнения имеют одинаковый приоритет и выполняются в порядке их расположения в выражении слева направо.

Операторы сравнения

Равенство	=	если оба значения равны, то результат будет равен 1, иначе 0
Неравенство	<> или !=	если оба значения не равны, то результат будет равен 1, иначе 0
Меньше	<	если одно значение меньше другого, то результат будет равен 1, иначе 0
Меньше или равно	<=	если одно значение меньше или равно другому, то результат будет равен 1, иначе 0
Больше	>	если одно значение больше другого, то результат будет равен 1, иначе 0
Больше или равно	>=	если одно значение больше или равно другому, то результат будет равен 1, иначе 0

Логические операторы

NOT	меняет значение оператора сравнения на противоположный
AND	возвращает общее значение выражения истинно, если они оба истинны
OR	возвращает общее значение выражения истинно, если хотя бы одно из них истинно

☆ Задание для закрепления

**Важно!**

На уроке будет использоваться база данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password

Таблица: order_details

База данных: northwind

1. Выбрать все строки с ценой единицы товара unit price превышающей 20

None

```
USE northwind;
SELECT *
FROM order_details
WHERE unit_price>20
```

2. Выбрать идентификаторы заказов order_id, если в заказе был продукт product_id равный 43 и статус заказа status_id равен 2

None

```
SELECT *
FROM order_details
WHERE product_id =43 and status_id = 2
```

3. Выбрать все строки там где количество определенного продукта в заказе quantity не превышает либо равно 15 или цена товара равна 15

None

```
SELECT *
```

```
FROM order_details
WHERE quantity<=15 or unit_price =15
```

Обратить внимание, что если нет ни одной строки отвечающей условию, то выводится пустая выборка.

4. Выбрать Inventory_id для всех заказов кроме заказов с id не равными 79 и 46

None

```
SELECT inventory_id
FROM order_details
WHERE order_id != 46 and order_id != 79
```

5. Выбрать только те строки в которых либо unit_price больше 9 и order_id равен 30 либо unit_price больше 9 и status_id равен 1

Задача решается двумя способами:

В данном случае мы используем (), чтобы объединять условия. Скобки предполагают, что выражения в них выполняются в первую очередь.

Аналогия с математическими операциями $(2+2)*2 = 2+2*2$

None

```
SELECT *
from order_details
where (order_id = 30 and unit_price>9) or (unit_price>9 and status_id = 1)
```

None

```
SELECT *
from order_details
where unit_price>9 and (order_id = 30 or status_id = 1)
```

Оператор поиска по маске



Оператор поиска по маске - это оператор, который позволяет находить строки, соответствующие определенному шаблону.

Для этой цели используется оператор **LIKE** в сочетании с символами подстановки (wildcards).



Оператор **LIKE** - это оператор, который позволяет сравнивать строковые значения с шаблоном и возвращает строки, которые соответствуют этому шаблону.

%	соответствует любому количеству символов (включая ноль).
-	(нижнее подчеркивание) соответствует <u>одному</u> любому символу.

Как вы думаете, какие слова подходят к следующим шаблонам?

- '%a%p_'
- '_'
- '_%п_ж%'
- 'ac'

☆ Задание для закрепления

**Важно!**

На уроке будет использоваться база данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password

Таблица: employees

База данных: northwind

1. Выбрать всех сотрудников с именами на букву A

None

```
SELECT * FROM northwind.employees
where first_name LIKE 'A%'
```

2. Выбрать сотрудников, живущих в городах, названия которых заканчиваются на 'mond'

None

```
SELECT * FROM northwind.employees
where city like '%mond'
```

3. Выбрать домашние номера сотрудников, если в заметках которых (notes) есть упоминание французского языка (French), и в середине фамилии этого сотрудника (last_name) присутствует английская буква k

None

```
SELECT * FROM northwind.employees
where notes LIKE '%French%' or last_name LIKE '%k%'
```

Операторы множественного выбора

Операторы множественного выбора в SQL позволяют работать с несколькими вариантами значений в запросах. Наиболее распространенные из них:

- Оператор **IN**
- Оператор **BETWEEN**



Оператор IN — это оператор, который используется для проверки того, входит ли значение в указанный список возможных значений.

Он эквивалентен набору OR для нескольких значений.



Оператор BETWEEN — это оператор, который включает в себя граничные значения.

Если вы хотите заменить его, сохраняя аналогичное поведение, можно использовать сочетание операторов `>=` и `<=`. Таким образом, вы явно указываете пределы.

BETWEEN...AND	оператор, который выбирает все элементы внутри заданного диапазона.
IN()	оператор позволяет узнать входит ли проверяемое значение столбца в список определённых значений.

☆ Задание для закрепления

**Важно!**

На уроке будет использоваться база данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password

Таблица: orders

База данных: northwind

1. Выбрать все строки, в которых shipping_fee больше либо равен 10 и меньше либо равен 15.

None

```
SELECT * FROM northwind.orders
WHERE shipping_fee BETWEEN 10 AND 15
```

2. Выбрать ship_name только для городов отгрузки ship_city Chicago и Miami.

None

```
SELECT ship_name FROM northwind.orders
WHERE ship_city IN ('Chicago', 'Miami')
```

Значение NULL



NULL — это отсутствие данных и указывает на неизвестность значения.

- По умолчанию столбцы и переменные могут принимать значение NULL, если не установлено ограничение NOT NULL.
- Ключевая особенность NULL заключается в том, что он не равен ничему, включая другой NULL.
- Нельзя сравнивать NULL с любым значением, используя операторы =, <, >, LIKE и др.

Выражение **NULL != NULL** не является истинным, так как невозможно достоверно сравнить одну неизвестность с другой.

Поиск пропусков - важная задача аналитика, ведь именно полнота данных влияет на корректность результата. Не всегда пропуски говорят о некачественных данных.

☆ Задание для закрепления

**Важно!**

На уроке будет использоваться база данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password

Таблица: orders

База данных: northwind

1. Найдите все строки в которых не указана shipped_date

None

```
SELECT * FROM northwind.orders
where shipped_date IS NULL
```

2. Выберите Id всех заказов где отсутствует shipped_date и shipper_id

None

```
SELECT * FROM northwind.orders
where shipped_date IS NULL and shipper_id is NULL
```