

## ЛАБОРАТОРНАЯ РАБОТА №9

*Тема:* Раздел WHERE предложения SELECT. Формирование условий включения строк в результат запроса.

*Цель:* Пробрести навыки составления логических условий включения строк в результирующий набор данных.

В данной лабораторной работе используются база данных HR, созданная в рамках первой лабораторной работы. Также используется таблица TB\_ELEKTROSTAL\_2018, которая создана и наполнена данными в процессе выполнения лаб. работы №1.

Дополнительно используется удаленная база данных проекта stackoverflow.com. Структура этой базы данных описана в методических указаниях к выполнению лабораторной работы № 3. Напомним, адрес ресурса <http://data.stackexchange.com/stackoverflow/query/new>.

В разделе WHERE задаются условия отбора записей в результат запроса, т.е. условие которому должна удовлетворять запись, чтобы попасть в результат запроса, аналог операции селекции в реляционной алгебре.

В выражении условий раздела WHERE могут быть использованы следующие предикаты:

1) Предикаты сравнения: =,<>,<,>,>=,<=;

**SELECT \* FROM authors WHERE 1=1**

2) **AND**– соединение нескольких логических выражений;

**SELECT title FROM titles WHERE yearpub>=1995 AND yearpub<=1997**

3) **OR**– если одно из двух условий истинно, то результат True;

**SELECT title FROM titles WHERE yearpub<1995 OR yearpub>1997**

4) **NOT** – отрицание, может ставиться непосредственно перед нижеследующими предикатами;

5) Предикат диапазона: **between A and B** – принимает значение True, если сравниваемое значение лежит между A и B;

```
SELECT title FROM titles WHERE yearpub NOT BETWEEN 1995 AND 1997
```

6) Вхождение во множество: **IN** (<список значений>)- принимает True, если сравниваемое значение входит во множество заданных значений;

```
SELECT title FROM titles WHERE yearpub IN (1995, 1996, 1997)
```

7) Сравнение с образцом: **LIKE**. В шаблон могут входить специальные символы «\_» – для обозначения любого одиночного символа, и «%» – для обозначения произвольной последовательности символов;

```
SELECT publisher, url FROM publishers WHERE publisher LIKE '%Wiley%'
```

8) Предикат сравнения с неопределённым значением: **IS NULL**.

```
SELECT publisher, "url not defined !" FROM publishers WHERE url IS NULL
```

Предикаты представляют собой выражения, принимающие истинностное значение. Они могут представлять собой как одно выражение, так и любую комбинацию из неограниченного количества выражений, построенную с помощью булевых операторов **AND**, **OR** или **NOT**. Кроме того, в этих комбинациях может использоваться SQL-оператор **IS**, а также круглые скобки для конкретизации порядка выполнения операций.

Предикат в языке SQL может принимать одно из трех значений **TRUE** (истина), **FALSE** (ложь) или **UNKNOWN** (неизвестно). Исключение составляют следующие предикаты: **IS NULL** (отсутствие значения), **EXISTS** (существование), **UNIQUE** (уникальность) и **MATCH** (совпадение), которые не могут принимать значение **UNKNOWN**.

Правила комбинирования всех трех истинностных значений легче запомнить, обозначив **TRUE** как **1**, **FALSE** как **0** и **UNKNOWN** как **1/2** (где-то между истинным и ложным значениями).

**AND** с двумя истинностными значениями дает минимум этих значений. Например, **TRUE AND UNKNOWN** будет равно **UNKNOWN**.

**OR** с двумя истинностными значениями дает максимум этих значений. Например, **FALSE OR UNKNOWN** будет равно **UNKNOWN**.

Отрицание истинностного значения равно 1 минус данное истинностное значение. Например, **NOT UNKNOWN** будет равно **UNKNOWN**.

Предикаты сравнения { =, <>, >, <, >=, <= } имеют традиционный смысл. В качестве элементов условия могут использоваться любые корректные SQL выражения, построенные на основе полей из источника данных, который сконструирован в разделе **FROM**. Например, следующий код языка SQL использованный для БД HR вернет записи сотрудников, у которых размер заработной платы меньше 3000:

```
SELECT * FROM employees  
WHERE salary<3000
```

Предикаты могут объединяться в связки с помощью логических операторов **AND** и **OR**. Например, следующий запрос вернет записи сотрудников с заработной платой больше 1000 но меньше 3000.

```
SELECT * FROM employees  
WHERE salary<3000  
AND salary>1000
```

Для иллюстрации операции **OR** рассмотрим запрос, который возвращает записи, соответствующие сотрудникам, у которых заработная плата меньше 3000, или они получают премию (**COMMISSION\_PCT**)

```
SELECT * FROM employees  
WHERE salary<3000  
OR coalesce(commission_pct, 0)>0
```

Напоминаем, что функция **coalesce()** используется для проверки значений **null**.

Предикат **between A and B** - принимает значения между A и B. Предикат истинен, когда сравниваемое значение попадает в заданный диапазон, включая границы диапазона. Одновременно в стандарте задан и противоположный предикат **not between A and B**, который истинен тогда, когда сравниваемое значение не попадает в заданный интервал, включая его границы. Например:

```
SELECT * FROM employees
```

**WHERE salary between 2000 and 3000**

В результате выполнения этого запроса будут возвращены все записи, у которых значения поля «Зарплата» находятся в интервале от 2000 до 3000, включительно.

**SELECT \* FROM employees  
WHERE salary no between 2000 and 3000**

Этот запрос вернет множество строк, которые соответствуют сотрудникам, получающим зарплаты в любом размере кроме той, которая попадает в диапазон от 2000 до 3000

Предикат вхождения во множество - IN (множество) истинен тогда, когда сравниваемое значение входит во множество заданных значений. При этом множество значений может быть задано простым перечислением или встроенным подзапросом. Одновременно существует противоположный предикат NOT IN (множество), который истинен тогда, когда сравниваемое значение не входит в заданное множество.

**SELECT \* FROM tb\_elektrostat\_2018  
WHERE typstr IN('переулок', 'тупик')**

В результирующий набор данных будут включены те записи, для которых значения поля «typstr» совпадут с элементами множества определенными предикатом сравнения IN, т.е. записи об абонентах, проживающих в переулках и тупиках.

Предикаты сравнения с образцом **LIKE** и **NOT LIKE**. Предикат **LIKE** требует задания шаблона, с которым сравнивается заданное значение, предикат истинен, если сравниваемое значение соответствует шаблону, и ложен в противном случае. Предикат **NOT LIKE** имеет противоположный смысл. По стандарту в шаблон могут быть включены специальные символы:

- символ подчеркивания ( ) - для обозначения любого одиночного символа;
- символ процента (%) - для обозначения любой произвольной последовательности символов;
- остальные символы, заданные в шаблоне, обозначают самих себя.

```
SELECT * FROM tb_elektrosta1_2018  
WHERE fio Like 'ЗАЙ%'
```

В результате выполнения запроса будут получены все записи, значения поля «FIO» для которых совпадет с шаблоном 'ЗАЙ%', т.е. Все фамилии сотрудников начинающиеся с символов «ЗАЙ».

Предикат сравнения с неопределенным значением **IS NULL**. Понятие неопределенного значения было внесено в концепции БД позднее. Неопределенное значение интерпретируется как значение, неизвестное на данный момент времени. При сравнении неопределенных значений не действуют стандартные правила сравнения: одно неопределенное значение никогда не считается равным другому неопределенному значению. Для выявления равенства значения некоторого поля неопределенному применяют специальные стандартные предикаты: **<имя поля>IS NULL** и **<имя поля > IS NOT NULL**.

### **Порядок выполнения работы**

Порядок выполнения работы.

Составить следующие запросы

- Составить запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка - имя и фамилия работника (first\_name, last\_name). Вторая колонка – размер заработной платы (salary). В результат нужно включить только данные о работниках с размером заработной платы более 10000.
- Составить запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка - имя и фамилия работника (first\_name, last\_name). Вторая колонка – размер заработной платы (salary). В результат нужно включить только данные о работниках с размером заработной платы более 10000. Также из выборки следует исключить работников, которые поступили на работу в период между 2003 и 2005

годами (границы диапазона входят в условие). Дата поступления – поле HIRE\_DATE. Для выделения года из даты воспользуйтесь функцией extract. Составьте запрос в двух вариантах: с использованием предиката between и без его использования.

- Составить запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка - имя и фамилия работника (first\_name, last\_name). Вторая колонка – размер заработной платы (salary). В результат нужно включить только данные о работниках с размером заработной платы более 10000. Размер заработной платы рассчитывается как сумма полей salary и выражения salary\*commission\_pct.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список имен пользователей, которые указали в адресе один из трех вариантов названия города Луганск, Lugansk или, Luhask.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список имен пользователей, которые указали в адресе город название которого начинается на Lug или Luh, но это не Lugansk или Luhansk.
- Составить запрос с использованием таблицы tb\_elektrostal\_2018, который выводит слово 'да', если количество жителей проспектов, больше чем количество жителей бульваров.
- Составить запрос с использованием таблицы tb\_elektrostal\_2018, который выводит одним предложением select два числа. Первое число – количество жителей с фамилией Волков или Волкова. Второе число – количество жителей с фамилией Зайцев или Зайцева.
- Составить запрос с использованием таблицы tb\_elektrostal\_2018, который выводит одним предложением select список фамилий жителей, фамилия которых начинается с последовательности символов «Иван», но фамилия не Иванов. (Например, Иваненко)
- Вывести фамилии абонентов, фамилия которых заканчивается на буквы “ов”, “ин”, “ий”, “ев”.

### Содержание отчета:

1. Тема, цель лабораторной работы.
2. Примеры выполнения запросов к базе данных.
3. Составленные согласно заданию запросы и скриншоты полученных результатов.
5. Выводы.

### Контрольные вопросы:

1. Опишите назначение раздела **WHERE** предложения **SELECT**.
2. Приведите примеры использования предикатов сравнения: **=**, **<>**, **<**, **>**, **>=**, **<=** для формулирования логических выражений в разделе **where**.
3. Приведите примеры использования операции логического отрицания (**NOT**) в построении логических выражений раздела **where**.
4. Опишите особенности использования логических операций **OR** и **AND** для построения логических выражений в разделе **where**.
5. Дайте описание особенностей использования предиката **between** в построении логических выражений раздела **where**.
6. Дайте характеристику использования предиката **like** для построения логических выражений раздела **where**. Какие метасимволы можно использовать при построении шаблонов для использования в предикате **like**?
7. Какие предикаты предусмотрены для обработки **null** значений при построении логических выражений раздела **where**?
8. Можно ли построить предложение **SELECT** без использования раздела **where**?
9. Каков будет результат вычисления следующих логических выражений **TRUE AND NULL**? **FALSE AND NULL**? **TRUE OR NULL**? **FALSE OR NULL**? **NOT NULL**?

Файл: Лабораторная работа 9  
Каталог: E:\Projects\documents\md\predmety\BD\_2018  
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm  
Заголовок:  
Содержание:  
Автор: sss  
Ключевые слова:  
Заметки:  
Дата создания: 15.11.2018 12:04:00  
Число сохранений: 50  
Дата сохранения: 17.11.2018 16:58:00  
Сохранил: sss  
Полное время правки: 280 мин.  
Дата печати: 17.11.2018 16:58:00  
При последней печати  
    страниц: 7  
    слов: 1 658 (прибл.)  
    знаков: 9 452 (прибл.)