**8. Ограничение и сортировка выходных данных.**

WHERE оператор

**Описание**

Оператор SQL WHERE используется для фильтрации результатов и применения условий в предложениях SELECT, INSERT, UPDATE или DELETE.

**Синтаксис**

WHERE conditions;

**Параметры или аргументы**

conditions — условия, которые должны быть выполнены для выбранных записей.

Текстовые значения заключаются в одинарные кавычки и зависят от регистра:

**SELECT** last\_name, job\_id

**FROM** employees

**WHERE** last\_name = **‘**Whalen**’**;

Для дат используется формат DD-MON-RR, так же записывается в кавычках

**SELECT** last\_name

**FROM** employees

**WHERE** hire\_date = **’**17-FEB-96**’**;

Пример:

**SELECT** \*

**FROM** employees

**WHERE** employee\_id = 101;

**SELECT** \*

**FROM** employees

**WHERE** employee\_id = 110

**AND** salary > 1000;

Условия сравнения с операторами =, <=, BETWEEN, IN, LIKE и NULL

Изображение выглядит как снимок экрана

Автоматически созданное описание

Условия диапазона:



Оператор **BETWEEN** используется для вывода строк на основе диапазона значений:

**SELECT** last\_name, salary

**FROM** employees

**WHERE** salary **BETWEEN** 2500 **AND** 3500

Условие принадлежности:



Оператор **IN** используется для проверки совпадения со значениями списка:

**SELECT** employee\_id, last\_name, salary, manager\_id

**FROM** employees

**WHERE** manager\_id **IN** (100, 101, 201)

В данном запросе будут выведены только те записи, manager\_id которых будет равен 100, 101 или 201

Сопоставление с шаблоном:



Оператор **LIKE** применяется при поиске допустимых строковых значений с использованием подстановочных знаков:

- **%** обозначает произвольное число символов (в том числе, возможно, нулевое)  
 - **\_** обозначает строго один символ

**SELECT** first\_name

**FROM** employees

**WHERE** first\_name **LIKE** ‘S%’

В данном запросе будут выведены только те записи, где first\_name начинается на букву S

**SELECT** last\_name

**FROM** employees

**WHERE** last\_name **LIKE** ‘\_o%’

В данном запросе будут выведены только те записи, где перед буквой ‘o’ есть только одна любая буква, а после нее любое количество символов

Если необходимо выбрать записи, в которых содержаться символы % или \_, можно использовать идентификатор **ESCAPE**

Использование условий NULL



Проверка на наличие пустых значение(null) производится с помощью операторы **IS NULL**

**SELECT** last\_name, manager\_id

**FROM** employees

**WHERE** manager\_id **IS NULL**

В данном запросе будут выведены только те фамилии, у которых нету начальника

Использование оператора **AND**



В случае оператора **AND** необходимо, чтобы оба составляющих условия были истинными:

**SELECT** employee\_id, last\_name, job\_id, salary

**FROM** employees

**WHERE** salary >= 10000

**AND** job\_id **LIKE** ‘%MAN%’

Использование оператора **OR**



В случае оператора **OR** необходимо, чтобы любое из составляющих условий было истинным:

**SELECT** employee\_id, last\_name, job\_id, salary

**FROM** employees

**WHERE** salary >= 10000

**OR** job\_id **LIKE** ‘%MAN%’

Использование оператора **NOT**



Используется для исключения

**SELECT** last\_name, job\_id

**FROM** employees

**WHERE** job\_id

**NOT IN** (‘IT\_PROG’, ‘ST\_CLERK’, ‘SA\_REP’)

В данном запросе будут выведены только те фамилии, профессия которых не находится в этом списке

Правила приоритета операторов

Изображение выглядит как снимок экрана

Автоматически созданное описание

Сортировка строк с использованием предложения ORDER BY

Предложение **ORDER BY** должно быть последним в инструкции SELECT



Сортировка извлеченных строк с помощью предложения ORDER BY:



* **ASC**: в порядке возрастания (по умолчанию)
* **DESC**: в порядке убывания

Можно сортировать по:

- Псевдониму столбца

- Позиции столбца

- Нескольким столбцам

**SELECT** last\_name, job\_id, department\_id, hire\_date

**FROM** employees

**ORDER BY** hire\_date

Переменные подстановки

Переменная с одним амперсандом(&) позволяет запрашивать ввод значения пользователем (не равно объявлению переменной через ‘:=’!)



**SELECT** employee\_id, last\_name, salary, department\_id

**FROM** employees

**WHERE** employee\_id = &employee\_num

Для дат и символьных значений необходимо использовать кавычки:

**SELECT** last\_name, salary, department\_id, salary\*12

**FROM** employees

**WHERE** job\_id = ‘&job\_title’

Переменные подстановки можно использовать для задания имен столбцов, условий в WHERE

Переменная с двумя амперсандами (&&) позволяет повторно использовать значение переменной, не запрашивая каждый раз ввод пользователя:



**SELECT** employee\_id, last\_name, job\_id, &&column\_name(ввод столбца)

**FROM** employees

**ORDER BY** &column\_name

Команда DEFINE

Использование команды **DEFINE**



* Команда DEFINE используется для создания переменной и присваивания ей значения
* Команда UNDEFINE используется для удаления переменной

**DEFINE** employee\_num = 200

**SELECT** employee\_id, last\_name, salary, department\_id

**FROM** employees

**WHERE** employee\_id = &employee\_num

**UNDEFINE** employee\_num