

SQL. Вводный курс

Лекция №4: LIKE, CASE, Функции

Оператор LIKE

<поле> LIKE <шаблон> возвращает логическую истину при нахождении первого подходящего строкового шаблона. В шаблоне используются спец.символы:

- **< % >** : знак процента - заменяет любую последовательность символов
- **< _ >** : знак подчеркивание - заменяет один любой символ

Все преподаватели фамилии которых начинаются на "К"

SELECT * FROM LECTURERS WHERE SURNAME LIKE 'К%'

ID	SURNAME	NAME	CITY	UNIV_ID
1	Кравчук	ЛИ	Днепропетровск	7
5	Коцюба	ВС	Киев	6

Все преподаватели отчество которых начинаются на "И" (вторая буква поля Name = "И")

SELECT * FROM LECTURERS WHERE NAME LIKE '_И'

ID	SURNAME	NAME	CITY	UNIV_ID
1	Кравчук	ЛИ	Днепропетровск	7
3	Телегин	ВИ	Харьков	2
14	Сулимова	ИИ	Харьков	2

Оператор IS NULL

<поле> **IS NULL** используется для проверки содержимого поля на наличие в нем пустого значения NULL, а также для проверки **НЕ** пустого значения:

<поле> **IS NOT NULL**

SELECT * FROM STUDENTS WHERE BIRTHDAY IS NULL

ID	SURNAME	NAME	STIPEND	COURSE	CITY	BIRTHDAY	UNIV_ID
21	Запорожец	Владимир	750	5	Киев	(NULL)	5

Оператор CASE

Многовариантная проверка значения выражения (или выражений).
Используется в перечне полей выборки оператора SELECT.

для одного проверочного выражения:

```
CASE <выражение>  
    WHEN <значение проверки> THEN <итоговое выражение>  
    [WHEN <значение проверки> THEN <итоговое выражение>]  
    [ELSE <итоговое значение>]  
END
```

для нескольких выражений проверки:

```
CASE  
    WHEN <условие проверки> THEN <итоговое выражение>  
    [WHEN <условие проверки> THEN <итоговое выражение>]  
    [ELSE <итоговое значение>]  
END
```

Примечание: часто используется для подмены NULL значений

Оператор CASE (пример)

Вывести список студентов и их дифференцированный уровень стипендий:
низкая, средняя, высокая

```
SELECT NAME, SURNAME,  
       CASE  
         WHEN STIPEND BETWEEN 300 AND 500 THEN 'Низкая'  
         WHEN STIPEND BETWEEN 501 AND 650 THEN 'Средняя'  
         WHEN STIPEND BETWEEN 651 AND 800 THEN 'Высокая'  
         ELSE 'Другая'  
       END DIF_STIPEND  
FROM STUDENTS
```

NAME	SURNAME	DIF_STIPEND
Тимофей	Цилюрик	Средняя
Игнат	Козьменко	Низкая
Роман	Березовский	Высокая
Леонид	Тарара	Другая

Использование константных выражений

Стандарт предусматривает использование константных выражений любых типов в любом количестве и последовательности в качестве полей выборки

-- MSSQL

```
SELECT 'Фамилия' Surname, 'ИМЯ' Name, 'Вес' Weight, 74.6 Kg, '2002-02-15' Birthday
```

Surname	Name	Weight	Kg	Birthday
Фамилия	ИМЯ	Вес	74.6	2002-02-15

-- Oracle

```
SELECT 'Фамилия' Surname, 'ИМЯ' Name, 'Вес' Weight, 74.6 Kg, '2002-02-15' Birthday FROM dual;
```

Арифметические операции

Перечень выборки также может содержать вычисляемые поля с использованием операций:

- [+] : сложение
- [-] : вычитание
- [*] : умножение
- [/] : деление
- [%] : деление по модулю (остаток от деления)

Вывести список студентов и их стипендий в долларовом эквиваленте

```
SELECT *, (STIPEND / 7.98) AS STIPEND_USD FROM STUDENTS
```

ID	SURNAME	NAME	STIPEND	COURSE	CITY	BIRTHDAY	UNIV_ID	STIPEND_USD
4	Козьменко	Игнат	500	1	Киев	1994-04-26	6	62.65
5	Ориненко	Анатолий	450	4	Львов	1990-09-08	4	56.39
6	Березовская	Роман	750	2	Киев	1992-03-09	1	93.98
7	Пименчук	Дмитрий	800	1	Харьков	1992-11-20	2	100.25

Соединение строк

Разные реализации SQL по-разному реализуют "конкатенацию" строк

- в Microsoft SQL используется `<+>`

```
SELECT 'Студент ' + NAME + ' ' + SURNAME + ' учится на ' + CAST(COURSE as VARCHAR) + ' курсе'  
FROM STUDENTS
```

- в Oracle используется `<||>`

```
SELECT 'Студент ' || NAME || ' ' || SURNAME || ' учится на ' || COURSE || ' курсе'  
FROM STUDENTS
```

- в MySQL используется спец. функция `CONCAT(<стр1>,<стр2>[,...])`

```
SELECT CONCAT('Студент ', NAME, ' ', SURNAME, ' учится на ', COURSE, ' курсе') FROM STUDENTS
```

CONCAT_RESULT
Студент Виталий Кабанов учится на 4 курсе
Студент Игорь Павленко учится на 1 курсе

Строковые функции

Строковые преобразования:

- **LOWER** (<строка>) - перевод в строчные символы (нижний регистр)
- **UPPER** (<строка>) - перевод в прописные символы (верхний регистр)
- **INITCAP** (<строка>) - перевод первой буквы каждого слова строки в прописную (заглавную); *отсутствует реализация для MSSQL*

```
SELECT LOWER (SURNAME) L_FLD, UPPER (NAME) U_FLD  
FROM STUDENTS WHERE COURSE = 4 AND STIPEND > 600
```

L_FLD	U_FLD
тарара	ЛЕОНИД

PS: Полный перечень строковых функций для MS Sql по ссылке [String Built-in functions](#)

Строковые функции

- LEFT(<строка>,<длина>) - взять из строки слева первые символы
- RIGHT(<строка>,<длина>) - взять из строки справа последние символы
- LTRIM (<строка>) - удаление левых граничных символов пробела
- RTRIM (<строка>) - удаление правых граничных символов пробела

Примечание: функции LTRIM и RTRIM часто используются при написании условных выражений. Это связано с тем что константы 'ABC' и 'ABC ' не равны друг другу

- SUBSTRING(<строка>,<начало>,<количество>) - выделение подстроки
- CHARINDEX(<подстрока>,<строка>) - поиск подстроки, тип возвращаемого значения INT. Аналог функции стандарта INSTR()
- LEN(<строка>) - определение длины строки, тип возвращаемого значения INT. Функция возвращает NULL, если <строка> имеет NULL-значение. Аналог функции стандарта LENGTH()

Строковые функции

```
SELECT LEFT ('TEST1234567890',7) L_PAD,  
       RIGHT ('TEST1234567890',7) R_PAD,  
       ('['+ LTRIM ('      TEST')+ ']') L_TRIM,  
       ('['+ RTRIM ('TEST    ')+ ']') R_TRIM,  
       CASE  
         WHEN 'TEST'='TEST ' THEN 'YES' ELSE 'NO'  
       END BOOL1,  
       CASE  
         WHEN LTRIM (' TEST')=RTRIM ('TEST ') THEN 'YES' ELSE 'NO'  
       END BOOL2,  
       CHARINDEX ('TEST', '_EST1-TEST2-TEST3') IN_STR,  
       SUBSTRING ('TEST1-TEST2-TEST3', 6, 7) SUB_STR,  
       PATINDEX ('%T_ST3', 'TEST1-TEST2-TEST3') INS2,  
       LEN ('1234567890TEST') LEN
```

L_PAD	R_PAD	L_TRIM	R_TRIM	BOOL 1	BOOL2	IN_STR	SUB_STR	INS2	LEN
TEST123	4567890	[TEST]	[TEST]	NO	YES	7	-TEST2-	13	14

Функции работы с числами

- **ABS (X)** - абсолютное значение
- **FLOOR (X)** - урезает значение числа с плавающей точкой до наибольшего целого, не превосходящего заданное число
- **CEIL (X)** - самое малое целое, равное или большее заданного числа
- **ROUND (X,<точность>)** - функция округления, аргумент <точность> задает точность округления
- **TRUNC (X,<точность>)** - функция усечения
- **COS (X), SIN (X), TAN (X)** - тригонометрические функции
- **EXP (X)** - экспоненциальная функция
- **LN (X), LOG (X)** - логарифмические функции
- **POWER (X,<экспонента>)** - функция возведения в степень
- **SIGN (X)** - определение знака числа
- **SQRT (X)** - вычисление квадратного корня

где *X* - значимое числовое выражение

Функции преобразования

CAST(<преобразовываемое выражение> AS <результатирующий тип>) - универсальная функция преобразования

CONVERT(<результатирующий тип>, <преобразовываемое выражение>[, <стиль>]) - универсальная функция преобразования (MS SQL)

- <стиль> - необязательный параметр, задающий формат, применяемый для преобразования типов DATETIME, FLOAT, MONEY в строку
например, 1 (101) при преобразовании даты в строку выдаст результат в US-формате mm/dd/yy (mm/dd/yyyy)

ORACLE ONLY:

TO_CHAR (<значимое выражение>[, <символьный формат>]) - преобразование в символьную строку, где

- <значимое выражение> - числовое значение или значение типа дата-время
- <символьный формат> - описывает вид выражения, так для числовых значений - '999999.99' и для значений типа дата-время имеет вид 'YYYY/MM/DD HH24:MI:SS:FF'

TO_NUMBER (<значимое символьное выражение>) - преобразование из символьного значения в числовое

TO_DATE (<значимое символьное выражение >[, <символьный формат>]) - преобразование символьной строки в дату

Функции преобразования

SELECT

```
'[' + CAST(123.456 AS CHAR(12)) + ']' TO_CHAR,  
CAST(' 123.456 ' AS DECIMAL(12,5)) TO_NUMBER,  
CAST('2012-02-15 12:23:34:456' AS DATETIME) TO_DATETIME,  
CONVERT(VARCHAR,  
        CAST('2012-02-15 12:23:34:456' AS DATETIME), 101) CONVERT_DATE
```

TO_CHAR	TO_NUMBER	TO_DATETIME	CONVERT_DATE
[123.456]	123.45600	2012-02-15 12:23:34	15/02/2012

PS: Более подробно о функциях преобразования для MS Sql: [Conversion Built-in functions](#)