



ОДНОСТРОЧНЫЕ СИМВОЛЬНЫЕ ФУНКЦИИ SQL

Цели занятия

- Изучение понятий однострочных и многострочных функций для работы с числами, строками и датами.
- Изучение символьных однострочных функций.

ФУНКЦИИ ПОЗВОЛЯЮТ:

- ВЫПОЛНЯТЬ расчеты с данными,
- МЕНЯТЬ ОТДЕЛЬНЫЕ значения данных,
- УПРАВЛЯТЬ выводом групп строк,
- МЕНЯТЬ форматы вывода данных,
- ПРЕОБРАЗОВЫВАТЬ типы данных в столбцах.

Типы функций

- **Однострочные функции** – это функции, которые работают только с одной строкой и возвращают по одному результату на строку.
- Однострочные функции могут быть разных типов:
 1. Символьные.
 2. Числовые.
 3. Для работы с датами.
 4. Функции преобразования.
- **Многострочные функции** – это функции, которые работают с группами строк и выдают по одному результату на каждую группу строк.

Свойства однострочных функций

- Манипулируют элементами данных.
- Принимают аргументы и возвращают по одному результату на строку.
- Обработывают каждую строку, возвращенную запросом.
- Могут изменять тип данных (тип данных на выводе может отличаться от типа данных, к которым обращается пользователь).
- Могут быть вложенными.
- Могут использоваться в предложениях `SELECT`, `WHERE` и `ORDER BY`.

СИНТАКСИС

*Функция (столбец | выражение,
[аргумент1, аргумент2, ...])*

Где: функция – название функции.

Столбец – столбец из любой таблицы базы данных.

Выражение – любая строка символов или вычисляемое выражение.

Аргумент1, аргумент2 – аргумент, используемый конкретной функцией, который может быть константой, значением, названием столбца или выражением.

Символьные функции

Символьные функции – это однострочные функции, которые воспринимают на входе символные данные, а возвращают или символные, или числовые значения.

Функция	Назначение
LOWER (<i>столбец выражение</i>)	Преобразование символов в нижний регистр.
UPPER (<i>столбец выражение</i>)	Преобразование символов в верхний регистр.
INITCAP (<i>столбец выражение</i>)	Преобразование начальных букв в верхний регистр, а остальных букв в нижний регистр.
CONCAT (<i>столбец выражение, столбец выражение</i>)	Конкатенация (объединение) первого символьного значения со вторым. Эквивалентно оператору конкатенации ().
SUBSTR (<i>столбец выражение, m[,n]</i>)	Возвращает <i>n</i> символов из символьного значения, начиная с позиции <i>m</i> . Если число <i>m</i> отрицательно, отсчет начинается от конца символьного значения.
LENGTH (<i>столбец выражение</i>)	Возвращает количество символов в значении.

Символьные функции

Пример №1: вывод всех наименований товаров, три первых символа которых равны “Ace”, и длины этих наименований.

```
SELECT name, LENGTH(name) FROM s_product  
WHERE SUBSTR (name,1,3) = 'Ace';
```

Пример №2: вывод имени и должности всех служащих с фамилией “Patel”, независимо от того в каком регистре хранятся значения фамилий в таблице.

```
SELECT first_name, title FROM s_emp  
WHERE UPPER(last_name) = 'PATEL';
```

Оператор конкатенации

Оператор конкатенации (`||`) – это аналог функции `CONCAT`, который позволяет соединять столбцы с другими столбцами, арифметическими выражениями или постоянными значениями для создания символьных выражений.

Столбцы, указанные по обе стороны этого оператора, объединяются в один выходной столбец.

Конкатенация

Пример №1: вывод в одном столбце наименования и страны для клиентов с хорошим кредитным рейтингом.

```
SELECT CONCAT (name, country) AS CUSTOMER  
FROM s_customer  
WHERE LOWER (credit_rating) = 'good';
```

Пример №2: вывод имен и фамилий служащих под общим заголовком “Employees”.

```
SELECT first_name || last_name AS “Employees”  
FROM s_emp;
```

Строки символов (литералы)

- **Литерал** – это любой символ, выражение или число, включенные в список `SELECT` и не являющиеся ни именем, ни псевдонимом столбца. Они будут выведены сервером для каждой возвращаемой строки.
- Литералы в виде текста произвольного формата могут быть включены в результат запроса. В списке `SELECT` они рассматриваются как столбцы.
- Символьные литералы и литералы-даты должны быть заключены в апострофы (‘ ’), а числовые литералы – не обязательно.
- Строковые литеры – это один или несколько символов, заключенные в апострофы. Апостроф в самом литерале записывается в виде двух смежных апострофов. Например, ‘You’'re a happy person, aren’'t you?’.
- Числовые литеры могут быть целыми числами (типа 3, 1444, 23) или вещественными (типа 3.14159, .25, -12, 2.e, 7E). В списках числовые литералы отделяются друг от друга запятыми.

Литералы

Пример №1: вывод под псевдонимами имени и фамилии каждого вице-президента строчными буквами, идентификатора пользователя с заглавной буквы и должности заглавными буквами.

```
SELECT LOWER (first_name || ' ' || last_name) AS VP,  
        INITCAP(user_id) AS IDUSER, UPPER(title) AS POST  
FROM s_emp  
WHERE UPPER (title) LIKE 'VP%';
```

Пример №2: вывод фамилии и должности сотрудников через запятую.

```
SELECT last_name || ', ' || title AS STAFF  
FROM s_emp;
```

Функция	Назначение
LPAD (<i>‘строка’</i> , длина [, <i>‘набор символов’</i>])	Заменяет целевую строку в строку с удельной длиной, прибавляя точно установленный набор символов слева от строки.
LTRIM (<i>‘строка’</i> [, <i>‘набор символов’</i>])	Удаляет символы с левой стороны строки.
NVL (<i>пустая ячейка выражение,</i> <i>подставляемое значение</i>)	Возвращает новое значение, если значение столбца является неопределенным.
NVL2 (<i>столбец,</i> значение для <i>определенной ячейки,</i> значение для <i>неопределенной ячейки</i>)	Подставляет новое значение и для определенных (заполненных) ячеек и для неопределенных (пустых) ячеек столбца.
RPAD (<i>‘строка’</i> , длина [, <i>‘набор символов’</i>])	Заменяет целевую строку в строку с удельной длиной, прибавляя точно установленный набор символов справа от строки.
RTRIM (<i>‘строка’</i> [, <i>‘набор символов’</i>])	Удаляет символы с правой стороны строки.
TRIM (<i>‘символ’</i> из <i>‘строка’</i>)	Удаляет точно установленный символ с начала или конца строки, или с начала и конца одновременно.

Символьные функции

Пример: правка фамилий сотрудников с использованием функции TRIM.

- LEADING – усечение символов с начала строки,
- TRAILING – усечение символов с конца строки,
- BOTH – усечение символов с обеих сторон строки.

SELECT last_name,

TRIM (leading 'B' from UPPER (last_name)) AS “Правка с начала строки”,

TRIM (trailing 'i' from LOWER (last_name)) AS “Правка с конца строки”,

TRIM (both 'A' from UPPER (last_name)) “Правка с обеих сторон”

FROM s_emp;

Символьные функции

Пример №1: задание длины фамилии 15-тью символами и заполнение недостающих символов знаками *.

```
SELECT LPAD (last_name, 15, '*') AS  
LPAD_EXAMPLE  
FROM s_emp;
```

Пример №2: правка фамилии под первой букве N.

```
SELECT last_name, LTRIM (UPPER (last_name),  
'N') AS LTRIM_EXAMPLE  
FROM s_emp;
```


Обработка неопределенных значений

- **Неопределенным значением** называется недоступное, неприсвоенное или неизвестное значение. Пустое значение могут иметь столбцы с любым типом данных, если они не были определены как NOT NULL.
- Например, в таблице s_emp столбец COMMISSION_PCT показывает, что комиссионные получают только торговые представители, для других служащих это не предусмотрено.
- Если выражение содержит неопределенное значение в любом из столбцов, то и результатом вычисления выражения будет неопределенное значение. При попытке деления на ноль будет сообщение об ошибке, а результатом деления на неопределенное значение будет неопределенное значение.
- Пустое значение и значение ноль представляют собой совершенно разные вещи. Все выражения, содержащие хотя бы одно пустое значение, равны также пустому значению.

Преобразование NVL для различных ТИПОВ ДАННЫХ

Тип данных	Пример преобразования
NUMBER	NVL (<i>числовой столбец, 5</i>)
DATE	NVL (<i>столбец даты, '01-ЯНВ-95'</i>)
CHAR или VARCHAR2	NVL (<i>символьный столбец, 'Пример'</i>)

Обработка неопределенных значений

Пример №1: вывод фамилии, должности и процента комиссионных для всех сотрудников.

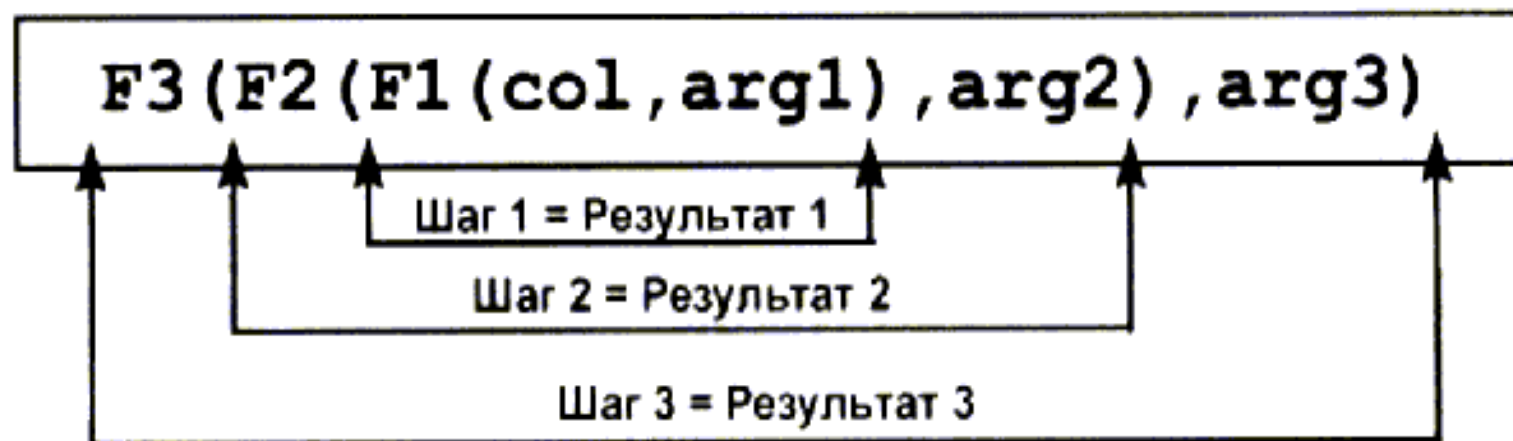
```
SELECT last_name, title, salary*NVL (commission_pct,  
0)/100 AS COMM1  
FROM s_emp;
```

Пример №2: вывод фамилии, процента комиссионных с добавлением 10% для тех кто их получает и установка 0% для сотрудников, которые их не получают.

```
SELECT last_name, commission_pct, NVL2 (commission_pct,  
commission_pct+10,0) AS COMM2  
FROM s_emp;
```

Вложенные функции

- Однострочные функции могут быть вложены на любую глубину.
- Вложенные функции вычисляются от самого глубокого уровня к верхнему.



Вложенные функции

Пример: вывод под заголовком “Vice Presidents” в одном столбце фамилии каждого вице-президента заглавными буквами и 3-ей части названия должности.

```
SELECT      CONCAT      (UPPER(last_name),  
      SUBSTR(title,3)) AS “Vice Presidents”  
  
FROM s_emp  
  
WHERE UPPER (title) LIKE ‘VP%’;
```