**ЛАБОРАТОРНАЯ РАБОТА №2.3**

**Преобразование вывода и встроенные функции**

В SQL реализованы операторы преобразования данных и встроенные функции, предназначенные для работы со значениями столбцов и/или константами в выражениях. Использование этих операторов допустимо в запросах везде, где можно использовать выражения.

**Числовые, символьные и строковые константы.**

Несмотря на то, что SQL работает с данными в понятиях строк и столбцов таблиц, имеется возможность применения значений выражений, построенных с использованием встроенных функций, констант, имен столбцов, которые определяются как своего рода виртуальные столбцы. Они помещаются в списке столбцов и могут сопровождаться псевдонимами,

Если в запросе вместо спецификации столбца SQL обнаруживает числа, то оно интерпретируется как числовая константа.

Символьные константы должны указываться в одинарных кавычках. Если одинарная кавычка должна выводиться как часть строковой константы, то ее нужно предварить другой одинарной кавычкой.

Например, результатом выполнения запроса

**SELECT** 'Фамилия', SURNAME, 'Имя', NAME, 100

**FROM** STUDENT;

является таблица следующего вида:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SURNAME |  | NAME |  |
| Фамилия | Иванов | ИМЯ | Иван | 100 |
| Фамилия | Петров | ИМЯ | Пётр | 100 |
| Фамилия | Сидоров | ИМЯ | Вадим | 100 |
| Фамилия | Кузнецов | ИМЯ | Артём | 100 |

**Арифметические операции для преобразования числовых данных.**

• Унарный (одиночный) оператор - (знак минус) изменяет знак числового значения, перед которым он стоит, на противоположный.

• Бинарные операторы +, -, \* и / предоставляют возможность выполнения арифметических операций сложения, вычитания, умножения и деления.

Например, результат запроса

**SELECT** SURNAME, NAME, STIPEND, - ( STIPEND\*KURS)/2

**FROM** STUDENT

**WHERE** KURS = 4 **AND** STIPEND >0;

Будет выглядеть следующим образом.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SURNAME | NAME | STIPEND | KURS |  |
| Сидоров | Вадим | 150 | 4 | -300 |
| Петров | Антон | 200 | 4 | -400 |

**Символьная операция конкатенации строк**.

Операция конкатенации, обозначаемая символом "||", позволяет соединять ("склеивать") значения двух или более столбцов символьного типа или символьных констант в одну строку.

Эта операция имеет синтаксис

<значимое символьное выражение> || <значимое символьное выражение>.

Например:

**SELECT** SURNAME || '\_' || NAME, STIPEND

**FROM** STUDENT

WHERE KURS = 4 AND STIPEND > О;

Результат запроса будет выглядеть следующим образом:

|  |  |
| --- | --- |
|  | STIPEND |
| Сидоров\_Вадим  Петров\_Антон | 150  200 |

**Символьные функции преобразования букв различных слов в строке.**

* • LOWER — перевод в строчные символы (нижний регистр)

LOWER (<строка>) •

* • UPPER — перевод в прописные символы (верхний. Регистр)

UPPER (<строка >)

* • INITCAP — перевод первой буквы каждого слова строки в заглавную (прописную)

INITCAP (<строка>

Например:

**SELECT LOWER**(SURNAME), **UPPER**(NAME)

**FROM** STUDENT

**WHERE** KURS = 4 **AND** STIPEND > 0;

Результат запроса будет выглядеть следующим образом

|  |  |
| --- | --- |
| SURNAME | NAME |
| Сидоров  Петров | Вадим  Антон |

**Символьные. строковые функции**.

* • LPAD —. дополнение строки слева.

LPAD (<строка>, <длина> [,<подстрока.>]

<строка> дополняется слева указанной в <подстроке> последовательностью символов до указанной <длины>.(возможно, с повторением последовательности);

если <подстрока> не указана, то по умолчанию <строка> дополняется пробелами;

если <длина> меньше длины <строки>, то исходная <строка> усекается слева до заданной <длины>.

* • RPAD — дополнение строки cnpава

RPAD(<строка>,<длина>[, <подстрока>])

<строка> дополняется справа указанной в <подстроке>

последовательностью символов до указанной <длины> (возможно, с повторением последовательности);

— если <подстрока> не указана, то по умолчанию <строка> дополняется пробелами;

— если <длина> меньше длины <строки>, то исходная <строка> усекается справа до заданной <длины>.

* • LTRIM — удаление левых граничных символов

LTRIM (<строка> [, <подстрока>])

— из <строки> удаляются слева символы, указанные в <подстроке>;

— если <подстрока> не указана; то по умолчанию удаляются пробелы;

— в <строку> справа добавляется столько пробелов, сколько символов слева было удалено, т.е. длина <строки> остается неизменной.

* • RTRIM — удаление правых граничных символов

RTRIN(<строка>[,<подстрока>])

—из <строки> удаляются справа символы, указанные в <подстроке>;

- если <подстрока> не указана, то по умолчанию удаляются пробелы;

- в <строку> слева добавляется, столько пробелов, сколько символов справа было удалено, т.е. длина <строки> остается неизменной.

Функции LTRIM и RTRIM рекомендуется использовать при написании условных выражений, в которых сравниваются текстовые строки. Дело в том, что наличие начальных или конечных пробелов в сравниваемых операндах может исказить результат сравнения.

Например, константы ' ААА' и ' ААА ' не равны друг другу.

• SUBSTR — выделение подстроки

SUBSTR(<строка>, <начало>[, <количесиио>])

— из <строки> выбирается заданное <количество> символов,

начиная с указанной позиции в строке <начало>;

— если <количество> не задано, символы выбираются с <начала> и до конца <строки>;

— возвращается подстрока, содержащая число символов, заданное параметром <количество>, либо число символов от позиции, заданной параметром <начало>, до конца <строки>;

— если указанное <начало> превосходит длину <строки>, то возвращается строка, состоящая из пробелов. Длина этой строки будет равна заданному <количеству> или исходной длине <строки> (при не заданном <количестве>),

INSTR — поиск подстроки

INSTR(<строка>, <подстрока> [, <начало поиска>'[,<номер вхождения>]])

— <начало поиска> задает начальную позицию в строке для поиска <подстроки>; если не задано, то по умолчанию принимается значение 1;

— <номер вхождения> задает порядковый номер искомой подстроки; если не задан, то по умолчанию принимается значение 1;

—. значимые выражения в, <начале поиска> или в <номере вхождения> должны иметь беззнаковый целый тип или приводиться к этому типу.

* • LENGTH — определение длины строки

LENGTH(<cmpoxa >)

— длина <строки>, тип возвращаемого значения — INT;

— функция возвращает NULL, если <строка> имеет NULL-значение.

Пример 1

Результат запроса

**SELECT** **LPAD**(SURNAME, 10, '@' ), **RPAD**(NAME,10, '$'.)

**FROM** STUDENT

**WHERE** KURS = 3 **AND** STIPEND > 0;

будет выглядеть следующим образом

|  |  |
| --- | --- |
| @@@@Петров  @@@@Павлов  @@@@Лукин | Пётр$$$$$$  Андрей$$$$$$  Артём$$$$$$ |

Пример 2. Запрос

**SELECT** **SUBSTR**(NAME, 1,1) ||'.' || SURNAME, CITY,

LENGTH(CITY)

FROM STUDENT

WHERE KURS IN(2, З, 4) AND STIPEND > 0;

выдаст результат

|  |  |  |
| --- | --- | --- |
|  | CITY |  |
| П.Петров  С.Сидоров  0.Зайцева  А. Лукин  А. Петров | Курск  Москва  Липецк  Воронеж  NULL | 5  6  6  7  NULL |

**Функции работы с числами.**

• ABS — абсолютное значение

ABS (<значимое числовое выражение>)

• FLOOR — наибольшее целое, не превосходящее заданное число с плавающей точкой

FLOOR(<значимое числовое выражение>)

• CEIL — наименьшее целое, которое равно или больше заданного числа

CEIL(<значимое числовое выражениё>)

— Функция округления — ROUND

ROUND (<значимое числовое выражение>, <точность>)

аргумент <точность> задает точность округления (см. пример ниже).

• Функция усечения — TRUNC

TRUNC (<значимое числовое выражение>, <точность>)

• Тригонометрические функции — СОS, SIN, TAN

СОS(<значимое числовое выражение>)

SIN(<значимое числовое выражение>)

TAN(значимое числовое выражение>)

• Гиперболические функции — COSH, SINH, TANH

СОSH(<значимое числовое выражение>)

SINH(<значимое числовое выражение>)

TANH(<значимое числовое выражение>)

**•** Экспоненциальная функция — EXP

***ЕХР****(<значимое числовое выражение>)*

**•** Логарифмические функции — ***LN****,* LOG

***LN****(<значимое числовое выражение>)*

***LOG****(<значимое числовое выражение>)*

**•** Функция возведения в степень — POWER

POWER *(<значимое числовое выражение>,<показатель степени>)*

**•** Определение знака числа — SIGN

SIGN*(<значимое числовое выражение>)*

**•** Вычисление квадратного корня — SQRT

SQRT *(<значимое числовое выражение>)*

Пример. Запрос

**SELECT** UNIV\_NAME, RATING,

**ROUND**(RATING, -1), **TRUNC**(RATING, -1)

**FROM** UNIVERSITY;

вернет результат

|  |  |  |  |
| --- | --- | --- | --- |
| UNIV\_NAME | RATING |  |  |
| МГУ  ВГУ  НГУ  РГУ | 610  296  345  421 | 610  300  350  420 | 600  290  340  410 |

**Функции преобразования значений.**

**•** Преобразование в символьную строку — **TO\_CHAR**

**TO\_CHAR** *(<значимое выражение> [, <символьный формат>])*

*— <значимое выражение>* должно представлять числовое значение или значение типа дата-время;

— для числовых значений <символьный формат> должен иметь синтаксис [S]9[9…][.9[9…]], где S – представление знака числа (при отсутствии предполагается без отображения знака), 9 — представление цифр-знаков числового значения (для каждого знакоместа). Символьный формат определяет вид отображения чисел. По умолчанию для числовых значений используется формат '999999.99';

для значений типа ДАТА-ВРЕМЯ <символьный формат> имеет вид (т. е. вид отображения значений даты и времени)

— в части даты:

'DD-Mon-YY'

'DD-Mon-YYYY'

'MM/DD/YY'

'MM/DD/YYYY'

'MM.DD.YY'

'MM.DD.YYYY'

В части времени:

'НН24'

'НН24:Ml'

'НН24:MI.SS’

'НН24:Ml:SS.FF’

где: - НН24 — часы в диапазоне от 0 до 24;

Ml — минуты;

SS — секунды;

FF — тики (сотые доли секунды)

При выводе времени в качестве разделителя по умолчанию используется двоеточие (":"), но при желании можно использовать любой другой символ.

Возвращаемое значение — символьное представление <значимого выражения> в соответствии с заданным <символьным форматом> преобразования.

• Преобразование из символьного значения в числовое – TO\_NUMBER

TO\_NUMBER(<значимое символьное выражение>)

При этом <значимое символьное выражение> должно задавать символьное значение числового типа.

• Преобразование символьной строки в дату — ТО\_DATE

ТО\_DATE (<значимое символьное выражение> [,

<символьный формат>])

- <значимое символьное выражение> должно задавать символьное значение типа ДАТА-ВРЕМЯ;

— <символьный формат> должен описывать представление значения типа ДАТА-ВРЕМЯ в <значимом символьном выражении>. Допустимые форматы (в том числе и формат по умолчанию) приведены выше.

Возвращаемое значение — <значимое символьное выражение> во внутреннем представлении. Тип возвращаемого значения — DАТЕ. Над значениями типа DATE разрешены следующие операции:

• к значению типа DATE можно прибавлять значения типа INTERVAL, в результате чего получается значение типа DATE;

• при вычитании двух значений типа DATE получается значение типа INTERVAL;

• при вычитании из значения типа DATE значения типа INTERVAL получается значение типа DATE.

В бинарных операциях один из операндов должен иметь значение отдельного элемента даты: только год, или только месяц, или только день.

Пример. Запрос

**SELECT** SURNAME, МАМЕ, BIRTHDAY,

TO\_CHAR(BIRTHDAY, 'DD-Mon-YYYY' ),

TO\_CHAR(BIRTHDAY, 'DD.ММ.YY' )

FROM STUDENT;

вернет результат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SURNAME | NAME | BIRTHDAY |  |  |
| Иванов  Петров  Сидоров Кузнецов Зайцева  Павлов  Котов  Лукин  Петров  Белкин | Иван  Петр  Вадим  Борис  Ольга  Андрей  Павел  Артем  Антон  Вадим | 3/12/1988 11/12/1986 7/06/1985 8/12/1987 21/05/1987 5/11/1985  NULL  11/12/1987 5/08/1987 20/01/1986 | 3-де к-1988  11-дек-1986  7-июн-1985  8-дек-1987  21-май-1987  5-ноя-1985 NULL  11-дек1987  5-авг-1987  20-янв-1986 | 3.12.88 11.12.86  7.06.85 8.12.87  21.05.87 5.11.85  NULL  11.12.87 5.08.87  20.01.86 |

Функция CAST является средством явного преобразования данных из одного типа в другой. Синтаксис этой команды имеет вид

CAST <значимое выражение> AS <тип данных>

• <значимое выражение> должно иметь числовой или символьный тип языка SQL (возможно, с указанием длины, точности и масштаба) или быть NULL-значением.

• Любое числовое выражение может быть явно преобразовано в любой другой числовой тип.

• Символьное выражение может быть преобразовано в любой числовой тип. При этом в результате такого преобразования отсекаются начальные и конечные пробелы, а остальные символы преобразуются в числовое значение по правилам языка SQL.

• Если заданная явным образом длина символьного типа недостаточна и преобразованное значение не размещается в нем, то результативное значение усекается справа.

• Возможно явное преобразование символьного типа в символьный с другой длиной. Если длина результата больше длины аргумента, то значение дополняется пробелами; если меньше, то усекается.

• NULL преобразуется в NULL.

• Числовое выражение может быть преобразовано в символьный тип.

Пример.

SELECT CAST STUDENT\_ID AS CHAR(10)

FROM STUDENT;

**УПРАЖНЕНИЯ**

1. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала один столбец, содержащий последовательность разделенных символом ";" (точка с запятой) значений всех столбцов этой таблицы; при этом текстовые значения должны отображаться прописными символами (верхний регистр), т. е. быть представленными в следующем виде:

10; КУЗНЕЦОВ; БОРИС; 0; БРЯНСК; 8.12.1987; 10.

2. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Б. КУЗНЕЦОВ; место жительства — БРЯНСК; родился — 8.12.87.

3. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: б. кузнецов; место жительства — брянск; родился: 8-дек-1987.

4. Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Борис Кузнецов родился в 1987 году.

5. Составьте запрос, выводящий фамилии, имена студентов и величину получаемых ими стипендий, при этом значения стипендий должны быть увеличены в 100 раз.

6. То же, что и в упр. 4, но только для студентов 1, 2 и 4 курсов и таким образом, чтобы фамилии и имена были выведены прописными буквами.

7 Составьте запрос для таблицы UNIVERSITY таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде: Код-10; ВГУ-r. ВОРОНЕЖ; Рейтинг=296.

8. То же; что и в упр. 7, но значения рейтинга требуется округлить до первого знака (например, значение -382 округляется-до 400).