

3

Однострочные функции

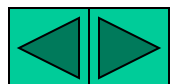


Темы

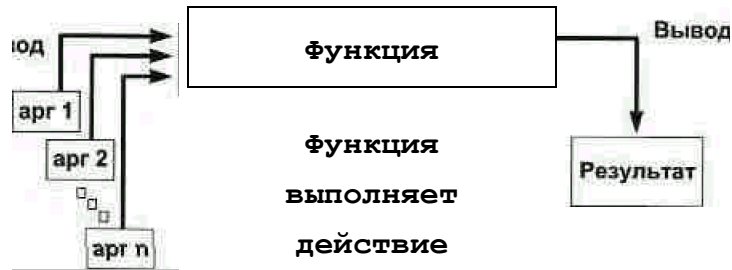
- **Различные типы функций в SQL.**
- **Включение в команды SELECT функций различных типов — символьных, числовых и типа "дата".**
- **Функции преобразования данных и их использование.**

Цели урока

Функции увеличивают мощность простого блока запроса и используются для манипулирования значениями данных. Это — первое из двух занятий, посвященные функциям. Вы узнаете об однострочных функциях для работы с числами, строками и датами, а также о функциях преобразования данных из одного типа в другой — например, из символьных в числовые



Функции SQL



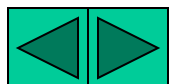
Функции SQL

Функции являются очень мощным средством SQL и используются в следующих целях:

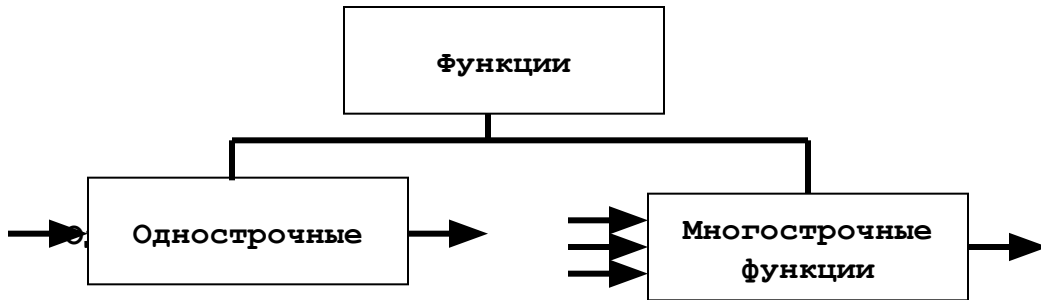
- Вычисления сданными.
- Изменение отдельных единиц данных.
- Управление выводом групп строк.
- Изменение формата выданных в столбцах.
- Преобразование типов данных в столбцах

Функции SQL принимают один или несколько аргументов и возвращают одно или несколько значений.

Примечание: Большинство описанных здесь функций взяты из версии SQL фирмы Oracle.



Два типа функций SQL



Функции SQL (продолжение)

Имеется два типа функций:

- Однострочные
- Многострочные

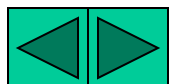
Однострочные строки

Эти функции работают только с одной строкой и возвращают по одному результату на строку. Однострочные функции могут быть разных типов. На этом занятии рассматриваются следующие типы функций:

- Символьные
- Числовые
- Для работы с датами
- Функции преобразования

Многострочные функции

Эти функции работают с группой строк и выдают по одному результату на каждую группу.



Oracle: SQL и PL/SQL 3-5

Однострочные функции

- Манипулируют элементами данных.
- Принимают аргументы и возвращают одно значение.
- Работают с каждой строкой, возвращаемой запросом.
- Возвращают один результат на строку.
- Изменяют тип данных.
- Могут быть вложенными.
- Синтаксис:

```
function name (column | expression, [arg2, arg2,...] )
```

Однострочные функции

Однострочные функции используются для работы с элементами данных. Они принимают один или несколько аргументов и возвращают по одному значению для каждой строки, выдаваемой запросом. Аргументом может быть:

- Константа , заданная пользователем
- Значение переменной
- Имя столбца
- Выражение

Особенности и однострочных функций

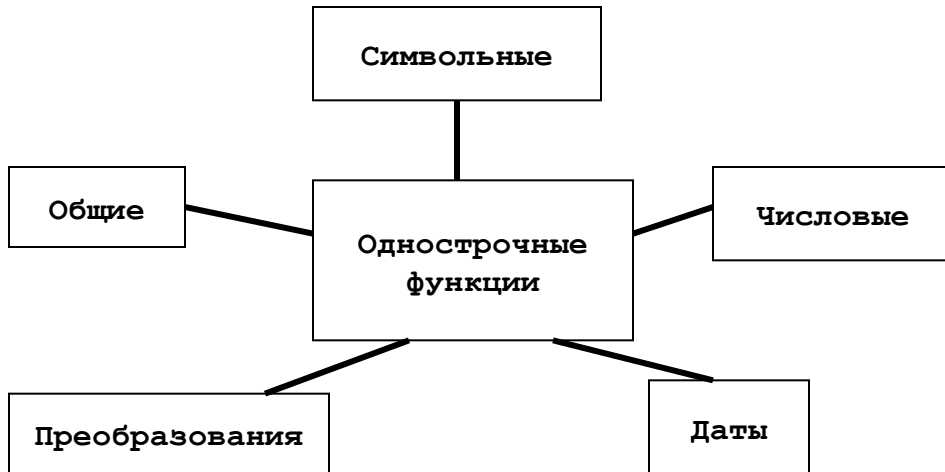
- Обработывают каждую строку, возвращенную запросом
- Возвращают по одному результату на строку.
- Могут изменять тип данных – тип данных на выводе может отличаться от типа данных, к которым обращается пользователь.
- Могут принимать один или несколько аргументов.
- Могут использоваться в предложениях **SELECT**, **WHERE** и **ORDER BY**.
- Могут быть вложенными,

Синтаксис:

<i>function_name</i>	имя функции.
<i>column</i>	любой именованный столбец базы данных.
<i>expression</i>	любая сорока символов или вычисляемое выражение.
<i>arg1, arg2</i>	любой аргумент, используемый функцией.



Однотрочные функции



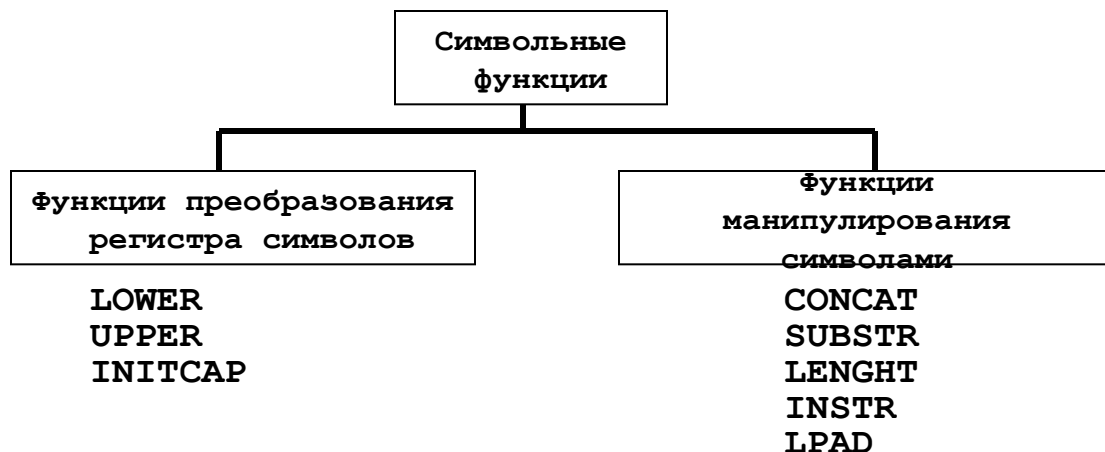
Однотрочные функции (продолжение)

Урок охватывает следующие однотрочные функции:

- Символьные функции; принимают на входе символьные данные, а возвращают как символьные, так и числовые значения
- Числовые функции: принимают на входе числовые данные и возвращают числовые значения.
- Функции даты: Работают с значениями типа DATE. Вес функции для работы с датами возвращают значение типа DATA за исключением функции MONTHS_BETWEEN , которая возвращает число.
- Функции преобразования: преобразуют значение из одного типа данных , в другой.
- Общие функции
 - Функция NVL
 - Функция DECODE



Символьные функции



Символьные функции

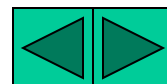
Однострочные символьные функции принимают на входе символьные данные, а возвращают символьное или числовое значение.

Символьные функции делятся на:

- Функции преобразования регистра символов
- Функции манипулирования символами

Функции	Назначение
LOWER(column expression)	Преобразует алфавитные символы в нижний регистр
UPPER (column expression)	Преобразует алфавитные символы в верхний регистр
INITCAP(column expression)	Преобразует символьные значения: первая буква каждого слова становится заглавной, а все остальные - строчными
CONCAT(column expression, column2 expression))	Присоединяет первое символьное значение ко второму. Эквивалентно оператору конкатенации (!!)
SUBSTR(column expression,m[,n])	Возвращает n символов символьного значения, начиная с символа m. Если m отрицательное, то отсчет начинается с конца символьного значения. Если не указано n, то возвращаются все символы до конца строки.
LENGTH (column expression)	Возвращает количество символов в значении
INSTR(column expression,m)	Возвращает номер позиции указанного символа
LPAD(column expression,n,'string')	Дополняет символьное значение, выровненное справа, заданными символами 'string' до длины n

Примечание : это только часть имеющихся символьных функций.



Функции преобразования регистра СИМВОЛОВ

Преобразование регистра для символьных строк:

Функция	Результат
LOWER(' SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP(' SQL Course')	SqL Course

Функции преобразования регистра символов

Три функции преобразования регистра символов - это **LOWER**, **UPPER** и **INITCAP**.

- **LOWER**: преобразует строку символов верхнего регистра или обоих регистров в символы нижней) регистра.
- **UPPER**: Преобразует строку символов нижнего регистра или обоих регистров у символы верхнего регистра.
- **INITCAP**: Преобразует первую букву каждого слова в заглавную, а опальные буквы - в строчные.

```
SELECT 'The job title for '||INITCAP (ename)|| ' is '  
      ||LOWER(job) AS "EMPLOYEE DETAILS"  
FROM emp;
```

EMPLOYER DETAILS

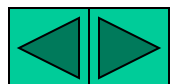
The job title for King is president

The job title for Blake is manager

The job title for dark is manager

. . .

14 rows selected.



Использование функций преобразования регистра

Вывод номера служащего, фамилии и номера отдела для служащего по фамилии Blake.

```
SELECT empno, ename, deptno
FROM emp
WHERE ename = 'blake'
```

no rows selected

```
SELECT empno, ename, deptno
FROM emp
WHERE LOWER (ename) = 'blake'
```

EMPNO	ENAME	DEPTNO
7698	BLAKE	30

Функции преобразования регистра символов

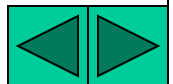
Пример показывает номер, фамилию и номер отдела служащего BLAKE. Предложение WHERE в первой команде SQL задает фамилию служащего - "blake". Т.к. все данные в таблице EMP хранятся в символах верхнего регистра, совпадения с фамилией "blake" не будет, вследствие чего ни одна строка не будет выбрана

Предложение WHERE во второй команде SQL указывает, что фамилия служащего из таблицы EMP должна быть преобразована в строчные буквы и только после этого сравниваться с "blake". Т.к. обе фамилии теперь представлены в символах нижнего регистра, совпадение будет обнаружено и будет выбрана одна строка. Следующее предложение WHERE даст такой же результат:

```
... WHERE ename = 'BLAKE'
```

В выходных данных фамилия выглядит так, как хранится в базе данных. Чтобы вывести фамилию с заглавной первой буквой, используйте функцию INITCAP в списке выбора команды SELECT.

```
SELECT empno, INITCAP(ename), deptno
FROM emp
WHERE LOWER(ename) = 'blake'
```



Oracle: SQL и PL/SQL 3-10
Функции манипулирования
символами

Манипулирование символьными строками

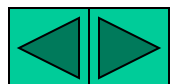
Функция	Результат
CONCAT (' Good ' , ' String ')	GoodString
SUBSTR (' String ' , 1,3)	Str
LENGTH (' String ')	6
INSTR (' String ' , ' r ')	3
LPAD (sal, ' * ')	*****5000

Функции манипулирования символами

CONCAT, SUBSTR, LENGTH, INSTR и LPAD – что пять функций манипулирования символами, которые обсуждаются в этом уроке.

- **CONCAT:** соединяет значения. Для функции CONCAT можно использовать не более двух параметров.
- **SUBSTR:** возвращает подстроку заданной длины.
- **LENGTH:** возвращает количество символов в виде числового значения.
- **INSTR:** возвращает номер позиции указанного символа.
- **LPAD:** дополняет символьное значение" выровненное справа, до заданной длины.

Примечание: функции манипулирования символами RPAD дополняет до нужной длины символьное значение, выровненное слева.



Использование функций манипулирования символами

```
SELECT  ename, CONCAT (ename, job), LENGTH (ename),
        INSTR (ename, 'A')
FROM    emp
WHERE   SUBSTR (job, 1,5) = 'SALES'
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
-----	-----	-----	-----
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALESMAN	5	1
TURNER	TURNERSALESMAN	6	0
WARD	WARDSALESMAN	4	2

Функции манипулирования символами (продолжение)

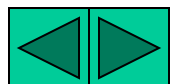
В примере сверху для всех служащих отдела сбыта выводятся следующие данные: фамилия и должность служащего, соединенные в одно целое, длина фамилии и номер позиции буквы А в фамилии.

Пример

Измените команду SQL так, чтобы получить данные по служащим, чьи фамилии заканчиваются буквой N.

```
SELECT  ename, CONCAT (ename, job), LENGTH (ename),
        INSTR (ename, 'A')
FROM    emp
WHERE   SUBSTR (ename, -1, 1) = 'N';
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
-----	-----	-----	-----
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALLESMAN	5	1



Числовые функции

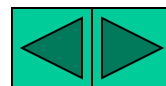
- ROUND:** Округляет значение до заданной точности
 $\text{ROUND}(45.926, 2) \longrightarrow 45.93$
- TRUNC:** Усекает значение до заданного количества десятичных знаков
 $\text{TRUNC}(45.926, 2) \longrightarrow 45.92$
- MOD:** Возвращает остаток от деления
 $\text{MOD}(1600, 300) \longrightarrow 100$

Числовые функции

Числовые функции принимают на входе числовые данные и возвращают числовые значения. Некоторые из числовых функций описаны в этом разделе.

Функция	Назначение
$\text{ROUND}(\text{column} \backslash \text{expression}, n)$	Округляет столбец, выражение или значение до n десятичных разрядов. Если n отрицательно, округляются разряды слева от десятичной точки.
$\text{TRUNC}(\text{column} \backslash \text{expression}, n)$	Усекает столбец, выражение или значение до n десятичных разрядов, а если n опущено, то до целого. Если n отрицательно, усекаются до нуля разряды слева от десятичной точки.
$\text{MOD}(m, n)$	Возвращает остаток от деления m на n .

Примечание: это только некоторые из имеющихся числовых функций.



Использование функции ROUND

```
SELECT    ROUND (45.923, 2),    ROUND (45.923, 0),
          ROUND (45.923, -1)
FROM      DUAL
```

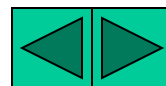
ROUND (45.923,2)	ROUND (45.923,0)	ROUND (45.923,-1)
-----	-----	-----
45.92	46	50

Функция ROUND

Функция ROUND округляет столбец, выражение или значение до n десятичных разрядов. Если второй аргумент равен нулю или отсутствует, значение округляется до нуля десятичных разрядов. Если второй аргумент равен 2, значение округляется до двух десятичных разрядов. Если второй аргумент равен -2, значение округляется вверх до сотен (до целого числа с двумя нулями).

Функция ROUND может использоваться и с функциями даты. Примеры приведены позже в этом уроке.

DUAL - что фиктивная таблица. Это еще будет обсуждаться.



Использование функции TRUNC

```
SELECT TRUNC (45.923, 2), TRUNC (45.923), TRUNC (45.923, -1)
FROM DUAL
```

TRUNC (45.923,2)	TRUNC (45.923)	TRUNC (45.923, -1)
-----	-----	-----
45.92	45	40

Функция TRUNC

Функция TRUNC усекает столбец, выражение или значение до n десятичных разрядов.

Аргументы функции TRUNC аналогичны аргументам функции ROUND. Если второй аргумент равен нулю или отсутствует, значение усекается до нуля десятичных разрядов. Если второй аргумент равен 2, значение усекается до двух десятичных разрядов. Если второй аргумент равен -2, значение усекается до сотен (до целого числа с двумя нулями).

Функция TRUNC, как и функция ROUND, может использоваться с функциями даты.



Использование функции MOD

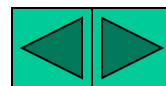
Вычисление остатка от деления оклада на комиссионные для всех служащих, работающих в должности продавца (salesman).

```
SELECT   ename, sal,  comm,  MOD(sal,comm)
FROM     emp
WHERE    job = 'SALESMAN'
```

ENAME	SAL	COMM	MOD (SAL, COMM)
-----	-----	-----	-----
MARTIN	1250	1400	1250
ALLEN	1600	300	100
TURNER	1500	0	1500
WARD	1250	500	250

Функция MOD

Функция MOD вычисляет остаток от деления *value 1* на *value 2*.
Пример сверху показывает остаток от деления оклада на комиссионные для всех служащих, состоящих в должности продавца (SALESMAN).



Работа с датами

- Oracle хранит данные во внутреннем цифровом формате.
 - Век, год, месяц, число, часы, минуты, секунды
- По умолчанию дата выдается в формате DD-MON-YY (число-месяц-год)
- функция SYSDATE возвращает текущие дату и время
- DUAL – это фиктивная таблица, используемая для просмотра SYSDATE.

Формат даты Oracle

Oracle хранит даты во внутреннем числовом формате, представляющем столетие, год, месяц, число, часы, минуты и секунды.

По умолчанию выходной и входной форматы любой даты – DD-MON-YY (число-месяц-год). Действительные даты Oracle лежат в диапазоне от 1 января 4712 до н.э. до 31 декабря 9999 н.э.

SYSDATE

SYSDATE – это функция, возвращающая текущие дату и время. SYSDATE можно использовать, как любое другое название столбца. Например, можно получить текущую дату путем выборки SYSDATE из таблицы. Обычно SYSDATE выбирается из фиктивной таблицы DUAL.

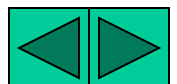
DUAL

Владельцем таблицы DUAL является пользователь SYS. Обращаться к ней могут все пользователи. Таблица содержит один столбец (DUMMY) и одну строку с значением X. Таблица DUAL полезна, если какое-то значение необходимо получить только один раз – например, константу, псевдостолбец или выражение, которые не вычисляются по таблице с пользовательскими данными.

Пример

Получите текущую дату с помощью таблицы DUAL.

```
SELECT    SYSDATE
FROM      DUAL
```



Арифметические операции с датами

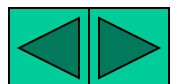
- Результатом прибавления числа к дате и вычитания числа из даты является дата.
- Результатом вычитания одной даты из другой является количество дней, разделяющих эти даты.
- Прибавление часов к дате производится путем деления количества часов на 24.

Арифметические операции с датами

Т.к. в базе данных даты хранятся в виде чисел, с ними можно выполнять такие арифметические операции, как сложение и вычитание. Прибавлять и вычитать можно как числовые константы, так и даты.

Возможны следующие операции:

Операции	Результат	Описание
Дата + число	Дата	Добавляется количество дней к дате
Дата - число	Дата	Вычитает количество дней из даты
Дата - дата	Количество дней	Вычитает одну дату из другой
Дата + число/24	Дата	Прибавляет часы к дате



Использование арифметических операторов с датами

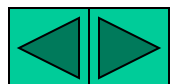
```
SELECT  ename, (SYSDATE-hiredate)/7 WEEKS
FROM    emp
WHERE   deptno = 10
```

ENAME	WEEKS
KING	830.93709
CIARK	853.93709
MILLER	821.36566

Арифметические операции с датами (продолжение)

Пример сверху показывает вывод фамилий и количества отработанных недель всех служащих отдела 10. Из текущей даты вычитается дата найма, а затем для вычисления количества недель результат делится на 7.

Примечание: SYSDATE – это функция SQL, возвращающая текущие дату и время, поэтому вы можете получить не такие результаты, как в примере.



Функции для работы с датами

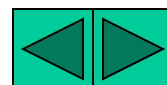
ФУНКЦИЯ	ОПИСАНИЕ
MONTHS BETWEEN	Число месяцев, разделяющих две даты
ADD_MONTHS	Добавление календарных месяцев к дате
NEXT_DAY	Ближайшая дата, когда наступит заданный день недели
LAST_DAY	Последняя дата текущего месяца
ROUND	Округление до целых суток
TRUNC	Отсечение части даты, обозначающей время

Функции для работы с датами

Эти функции работают с датами Oracle. Все функции для работы с датами возвращают значение типа DATE за исключением функции MONTHS_BETWEEN, возвращающей числовое значение.

- MONTHS_BETWEEN (date1, date2): вычисляет количество месяцев между date1 и date2. Результат может быть положительным или отрицательным. Если date1 позже date2, результат положителен; если date1 предшествует date2, результат отрицателен. Дробная часть результата представляет часть месяца.
- ADD_MONTHS (date, n): прибавляет n календарных месяцев к date. n должно быть целым и может быть отрицательным.
- NEXT_DAY (date, 'char'): возвращает дату после date, когда наступит заданный день недели ('char'). 'char' может быть числом, представляющим день недели, или строкой символов.
- LAST_DAY (date): возвращает последнюю дату месяца, содержащего date.
- ROUND (date[, 'fmt']): возвращает дату date, округленную до единицы, заданной моделью формата fmt. Если модель fmt отсутствует, date округляется до ближайшей даты.
- TRUNC (date[, 'fmt']): возвращает дату date, в которой время усечено до единицы, заданной моделью формата fmt. Если модель fmt отсутствует, date усекается до ближайшего дня.

Это только некоторые из имеющихся функций. Модели формата обсуждаются далее в этом уроке. Примерами моделей формата являются месяц (MONTH) или год (YEAR).



Функции для работы с датами

- `MONTHS_BETWEEN ('01 -SEP-95', '11 -JAN-94')` → 19.6774194
- `ADD_MONTHS ('11-JAN-94', 6)` → 11-JUL-94 '
- `NEXT_DAY ('01-SEPTS'/FRIDAY')` → 08-SEP-95 '
- `LAST_DAY ('01-SEP-95')` → '30-SEP-95 '

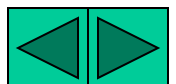
Функции для работы с датами (продолжение)

Для всех служащих, проработавших менее 200 месяцев, выводится номер служащего, дата найма, количество отработанных месяцев, дата аттестации после 6 месяцев работы, дата первой пятницы после даты найма и последний день месяца, когда служащий был нанят на работу.

```
SELECT empno, hiredate,
       MONTHS_BETWEEN (SYSDATE, hiredate) TENURE,
       ADD_MONTHS (hiredate, 6) REVIEW,
       NEXT_DAY(hiredate, 'FRIDAY'), LAST DAY(hiredate)
FROM emp
WHERE MONTHS_BETWEEN (SYSDATE, hiredate) < 200;
```

EMPNO	HIREDATE	TENURE	REVIEW	NEXT_DAY (LAST_DAY (
7839	17-NOV-81	192.24794	17-MAY-82	20-NOV-81	30-NOV-81
7698	01-MAY-81	198.76407	01-NOV-81	08-MAY-81	31-MAY-81

...
11 rows selected



Функции для работы с датами

- `ROUND('25-JUL-95','MONTH')` —→ `01-AUG-95`
- `ROUND('25-JUL-95','YEAR')` —→ `01-JAN-96`
- `TRUNC('25-JUL-95','MONTH')` —→ `01-JUL-95`
- `TRUNC('25-JUL-95','YEAR')` —→ `01-JAN-95`

Функции для работы с датами (продолжение)

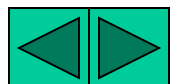
Функции `ROUND` и `TRUNC` могут использоваться для числовых значений и дат. Если они используются с датами, даты округляются или усекаются в соответствии с заданной моделью формата. Следовательно, можно округлять даты до ближайшего года или месяца.

Пример

Сравните даты найма всех служащих, нанятых в 1987 г. Выведите на экран номер каждого служащего, дату и месяц найма с помощью функции `ROUND` и `TRUNC`.

```
SELECT empno, hiredate,
       ROUND(hiredate,'MONTH'), TRUNC(hiredate,'MONTH')
FROM   emp
WHERE  hiredate like '%87'
```

EMPNO	HIREDATE	ROUND(HIR	TRUNC(HIR
7788	19-APR-87	01-MAY-87	01-APR-87
7876	23-MAY-87	01-JUN-87	01-MAY-87



Функции преобразование



Функции преобразования

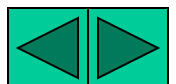
Помимо типов данных Oracle, столбцам таблиц в базе данных Oracle8 можно назначать чины данных ANSI, DB2 и SQL/DS. Но внутри системы сервер Oracle преобразует эти типы данных в типы данных Oracle8.

В некоторых случаях сервер Oracle допускает данные какого-то типа там, где он ожидает данные другого типа. Это допускается, если сервер Oracle может автоматически привести данные к определенному типу. Такое преобразование типов данных может производиться *неявно* сервером Oracle или *явно* пользователем.

Неявные преобразования типов данных производятся в соответствии с правилами, изложенными на следующих двух слайдах.

Явное преобразование типов данных производится с помощью функций преобразования. Функции преобразования преобразуют один тип данных в другой. Обычно эти функции следуют общепринятому правилу *datatype TO datatype*. Первый тип данных является входным, а второй - выходным.

Примечание: хотя неявное преобразование типов данных возможно, для упрощения чтения команд SQL рекомендуется делать это явно.



Неявное преобразование типов данных

Для операций присваивания Oracle может автоматически выполнять следующие преобразования:

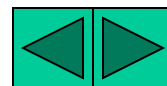
Исходный формат	Новый формат
VARCHAR2 или CHAR	NUMBER
VARCHAR2 или CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Неявное преобразование типов данных

Для операций присваивания сервер Oracle может автоматически выполнить следующие преобразования:

- VARCHAR2 или CHAR в NUMBER
- VARCHAR2 или CHAR в DATE
- NUMBER в VARCHAR2
- DATE в VARCHAR2

Присваивание значения будет успешным, если сервер Oracle сможет привести тип данных значения, расположенного справа от оператора присваивания, к типу переменной.



Неявное преобразование типов данных

При вычислении выражений Oracle может автоматически выполнять следующие преобразования:

Исходный формат	Новый формат
VARCHAR2 или CHAR	NUMBER
VARCHAR2 или CHAR	DATE

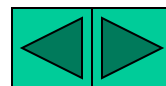
Неявное преобразование типов данных

При вычислении выражений сервер **Oracle** может автоматически выполнить следующие преобразования:

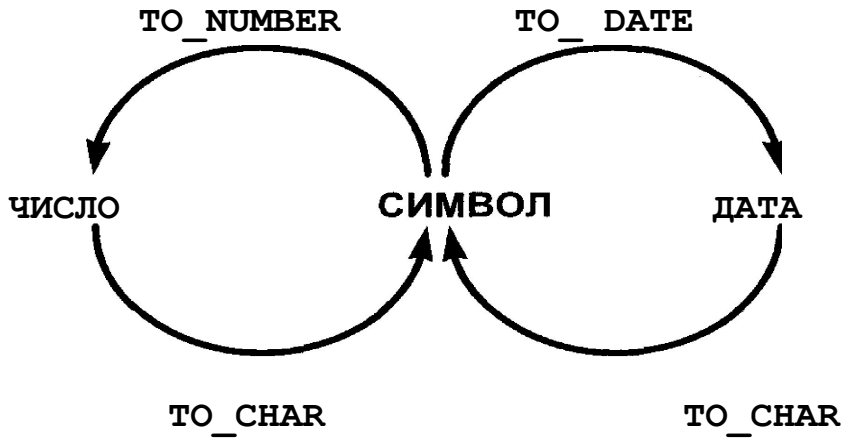
- VARCHAR2 или CHAR в NUMBER
- VARCHAR2 или CHAR в DATE

Обычно сервер Oracle автоматически выполняет преобразования в выражениях, если необходимое преобразование не охвачено правилом преобразования типов данных для операций присваивания.

Примечание: преобразования из CHAR в NUMBER успешны только в случае, если символьная строка представляет действительное число. Преобразования CHAR в DATE успешны только в случае, если символьная строка имеет формат даты по умолчанию DD-MON-YY (число-месяц-год).



Явное преобразование типов данных

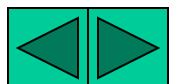


Явное преобразование типов данных

Для преобразования значения из одного типа данных в другой SQL предлагает три функции.

Функция	Назначение
TO_CHAR(<i>number\date</i> ,[' <i>fmt</i> '])	Преобразует число или дату в строку символов VARCHAR2 с моделью формата <i>fmt</i>
TO_NUMBER	Преобразует символьную строку, содержащую цифры, в число
TO_DATE	Преобразует символьную строку с датой в значение даты согласно указанному <i>fmt</i> . (элемент <i>fmt</i> опущен, используется формат DD-MON-YY.)

Примечание: это только часть имеющихся функций преобразования.



Функция TO_CHAR с датами

```
TO_CHAR(date, 'fmt')
```

Модель формата:

- Должна быть заключена в апострофы. Различает символы верхнего и нижнего регистров.
- Может включать любые разрешенные элементы формата даты.
- Использует элемент *fm* для удаления конечных пробелов и ведущих нулей.
- Отделяется от значения даты запятой.

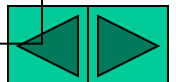
Вывод данных в заданном формате

До сих пор для вывода всех дат Oracle использовался стандартный формат DD-MON-YY (число-месяц-год). Функция TO_CHAR позволяет преобразовать дату из этого стандартного формата в формат, заданный пользователем.

Указания

- Модель формата различает символы верхнего и нижнего регистров и должна быть заключена в апострофы.
- Модель формата может включать любой действительный элемент формата даты. Дата обязательно отделяется от модели формата запятой.
- Названия дней и месяцев на выводе автоматически заполняются до нужной длины пробелами.
- Для удаления вставленных пробелов и ведущих нулей используйте элемент *fm* режима заполнения (fill mode).
- Изменить ширину выходного символьного столбца можно с помощью команды **COLUMN SQL*Plus**.
- Ширина столбца по умолчанию - 80 символов.

```
SELECT empno, TO_CHAR(hiredate, 'MM/YY') Month_Hired
FROM emp
WHERE ename = 'BLAKE'
```



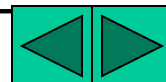
Oracle: SQL и PL/SQL 3-27

Элементы формата даты

YYYY	Полный год цифрами
YEAR	Год прописью
MM	Двухзначное цифровое обозначение месяца
MONTH	Полное название месяца
DY	Трехзначное алфавитное сокращенное название дня недели
DAY	Полное название дня недели

Примеры элементов формата даты

Элемент	Описание
SCC или CC	Век: если задано S, date до н.э. предшествует '' -''
Годы в датах YYYY или SYYYY	Год: если задано S, П ''_'' date до н.э. предшествует
YYY или YY или Y	3,2
Y, YY	или 1 цифра года Год с запятой в указанной позиции
IYYY, IYY, IY, I	4, 3, 2 или 1 цифра года по стандарту ISO
SYEAR или YEAR	Год прописью; если задано S, date до н.э. предшествует '' -''
BC или AD	Индикаторы даты до новой эры/ новой эры
B.C. или A.D.	Индикатор даты до новой эры/ новой эры с точками
Q	Квартал года
MM	Месяц двумя цифрами
MONTH	Название месяца, наполненное пробелами до 9 символов
MON	Трехбуквенное сокращенное название месяца
RM	Месяц римскими цифрами
WW или W	Неделя года или месяца
DDD или DD или D	День года, месяца или недели
DAY	Название дня, дополненное пробелами до 9 символов
DY	Сокращенное название дня из трех символов
J	Юлианская дата; количество дней с 31 декабря 4713 до н.э.



Элементы модели формата даты

- Элементы, которые задают формат части даты, обозначающей время.

HH24:MI:SSAM	15:45:32 PM
--------------	-------------

- Символьные строки добавляются в кавычках.

DD "of " MONTH	12 of OCTOBER
----------------	---------------

- Числовые суффиксы используются для вывода числительных прописью.

ddspth	fourteenth
--------	------------

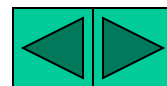
Форматы времени

Для вывода времени и литералов, а также для преобразования цифровых дат в даты прописью пользуйтесь следующими форматами.

Элемент	Описание
AM или PM	Индикатор ``до полудня/ после полудня``
A.M. или P.M.	Индикатор ``до полудня/ после полудня``- обозначение с точками
HH или HH12 или HH14	Время суток в 12- или 24-часовом диапазоне
MI	Минуты (0-59)
SS	Секунды (0-59)
SSSSS	Секунды после полуночи (0- 86399)

Другие форматы

Элемент	Описание
/ . ,	Знаки пунктуации воспроизводятся в результате
"of the"	Строка, указанная в кавычках, воспроизводится в результате



Oracle: SQL и PL/SQL 3-29

Функция TO_CHAR с датами

```
SELECT ename,  
       TO_CHAR(hiredate, 'fmDD Month YYYY') HIREDATE  
FROM    emp
```

```
ENAME      HIREDATE  
-----  
KING       17 November 1981  
BLAKE      1 May 1981  
CLARK      9 June 1981  
JONES      2 April 1981  
MARTIN     28 September 1981  
ALLEN      20 February 1981  
...  
14 rows selected.
```

Функция TO_CHAR с датами

Команда SQL на слайде выводит фамилии и даты найма всех служащих. Выходной формат даты найма - "17 November 1981".

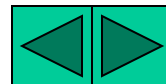
Пример

Измените пример для вывода дат в формате "Seventh of February 1981 08:00:00 AM".

```
SELECT  ename, TO_CHAR(hiredate,  
                       'fmDdspth "of"Month YYYY fmHH:MI:SS AM ') HIREDATE  
FROM    emp
```

```
ENAME      HIREDATE  
-----  
KING       Seventeenth of November 1981 12:00:00 AM  
BLAKE      First of May 1981 12:00:00 AM  
14 rows selected.
```

Обратите внимание на то, что название месяца соответствует заданной модели формата (IN ITCAP).



Oracle: SQL и PL/SQL 3-30

Функция TO_CHAR с числами

TO_CHAR (number, 'fmt')

Форматы, используемые с функцией TO_CHAR для вывода символьного значения в виде числа

9	Цифра
0	Вывод нуля
\$	Плавающий знак доллара
L	Плавающий символ местной валюты
.	Вывод десятичной точки
*	Вывод разделителя троек цифр

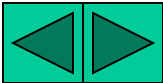
Функция TO_CHAR с числами

Функция TO_CHAR преобразует данные типа NUMBER в данные типа VARCHAR2. Это особенно полезно при конкатенации.

Элементы числового формата

Преобразуя число в данные типа VARCHAR2, можно пользоваться следующими элементами.

Элемент	Описание	Пример	Результат
9	Цифровой разряд (количество девяток определяет ширину поля вывода)	999999	1234
0	Вывод ведущих нулей	099999	001234
\$	Плавающий символ доллара	\$999999	\$1234
L	Плавающий символ местной валюты	L999999	FF1234
.	Десятичная точка в указанной позиции	999999.99	1234.00
,	Запятая в указанной позиции	999,999	1,234
MI	Знак "минус" справа (отрицательные значения)	999999MI	1234-
PR	Отрицательные символы в скобках	999999PR	<1234>
EEEE	Научное обозначение (обязательны четыре E)	99.999	1.234
V	Умножить на 10 ⁿ (n = кол-во девяток после запятой)	999999V99	123400
B	Вывод нулевых значений пробелами вместо нулей	B9999.99	1234.00



Oracle: SQL и PL/SQL 3-31

Функция TO_CHAR с числами

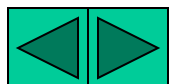
```
SELECT TO_CHAR(sal, '$99,999') SALARY  
FROM   emp  
WHERE  ename = 'SCOTT'
```

SALARY

\$3,000

Указания

- Если количество цифровых разрядов числа превышает количество разрядов, предусмотренное моделью формата, сервер Oracle выводит вместо всего числа строку символов фунта (#).
- Сервер Oracle округляет хранимое десятичное значение до количества десятичных разрядов, заданное в модели формата.



Функции TO_NUMBER и TO_DATE

- Преобразование строки символов в числовой формат с помощью функции TO_NUMBER

```
TO_NUMBER(char)
```

- Преобразование строки символов в формат даты с помощью функции TO_DATE

```
TO_DATE(char [ , 'fmt'])
```

Функции TO_NUMBER и TO_DATE

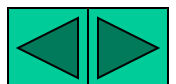
Иногда требуется преобразовать символьную строку в число или дату. Для этого используются функции TO_NUMBER и TO_DATE. Выбор модели формата будет основан на предыдущих демонстрациях использования элементов формата.

Пример

Вывод фамилий и даты найма всех служащих, принятых на работу 22 февраля 1981 г.

```
SELECT      ename, hiredate
FROM        emp
WHERE       hiredate = TO_DATE('February 22, 1981',
                               'Month dd, YYYY')
```

ENAME	HIREDATE
-----	-----
WARD	22-FEB-81



Формат даты RR

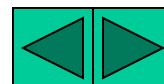
Текущий год	Заданная дата	Формат RR	Формат YY
1995	27-ОCT-95	1995	1995
1995	27-ОCT-17	2017	1917
2001	27-ОCT-17	2017	2017
2001	27-ОCT-95	1995	2095

		Год, заданный двузначным числом	
		0-49	50-99
Если две последних цифры текущего года равны:	0-49	Возвращаемая дата относится к текущему столетию.	Возвращаемая дата относится к столетию перед текущим.
	50-99	Возвращаемая дата относится к столетию после текущего.	Возвращаемая дата относится к текущему столетию.

Элемент RR в формате даты

Элемент RR аналогичен элементу YY, но позволяет задавать разные столетия. Элемент RR можно использовать вместо YY, чтобы столетие в возвращаемом значении варьировалось в зависимости от заданного двузначного года и двух последних цифр текущего года. Поведение элемента RR суммируется в таблице.

Текущий год	Заданная дата	Интерпретированная дата (RR)	Интерпретированная дата (YY)
1994	27-ОCT-95	1995	1995
1994	27-ОCT-17	2017	1917
2001	27-ОCT-17	2017	2017



Функция NVL

Преобразует неопределенное значение в действительное

- Используемые типы данных - DATE, символьные (CHARACTER) и числовые (NUMBER) .
- Типы данных должны совпадать
 - NVL(comm, 0)
 - NVL(hiredate, '01 -JAN- 97')
 - NVL(job, 'No Job Yet')

Функция NVL

Функция NVL используется для преобразования неопределенного значения (NULL) в действительное.

Синтаксис

NVL (*expr1*, *expr2*)

где: *expr1* исходное значение или выражение, которое может содержать неопределенное значение.

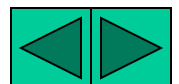
expr2 конечное значение для преобразования неопределенного значения.

Функцию NVL можно использовать для преобразования данных любого типа, но тип данных возвращаемого значения всегда такой, как у *expr1*.

Преобразования NVL для различных типов данных

Тип данных	Пример преобразования
NUMBER	NVL(<i>number column</i> , 9)
DATE	NVL (<i>date column</i> , ' 01-JAN-95')
CHAR VARCHAR2	NVL(<i>character column</i> , 'Unavailable')

или



Oracle: SQL и PL/SQL 3-35

Использование функции NVL

```
SELECT ename, sal, comm, (sal*12)+NVL(comm,0)
FROM emp
```

ENAME	SAL	COMM	(SAL*12)+NVL(COMM,0)
KING	5000		60000
BLAKE	2850		34200
CLARK	2450		29400
JONES	2975		35700
MARTIN	1250	1400	16400
ALLEN	1600	300	19500

14 rows selected.

Функция NVL

Для вычисления годового дохода служащих необходимо умножить их оклад на 12, а затем прибавить сумму комиссионных.

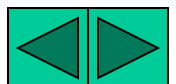
```
SELECT ename, sal, comm, (sal*12)+comm
FROM emp
```

ENAME	JOB	(SAL*12)+COMM
KING	PRESIDENT	
BLAKE	MANAGER	
CLARK	MANAGER	
JONES	MANAGER	
MARTIN	SALESMAN	16400

...

14 rows selected.

Следует отметить, что годовой доход вычисляется только для служащих, зарабатывающих комиссионные. Если какой-либо столбец в выражении содержит неопределенное значение, результатом также будет неопределенное значение. Чтобы вычислить результаты по всем служащим, необходимо преобразовать неопределенное значение в число прежде, чем применять арифметический оператор. В примере на слайде функция NVL используется для преобразования неопределенных значений в ноль.



Функция DECODE

Упрощает условные запросы, выполняя работу команды CASE или IF-THEN-ELSE

```
DECODE (col/ expression, search1, result1  
        [,search2, result2, ...,]  
        [,default])
```

Функция DECODE

Функция DECODE действует подобно IF-THEN-ELSE в различных языках. Функция DECODE расшифровывает выражение (*expression*) после сравнения его с каждым искомым значением (*search*). Если выражение равно искомому значению, функция возвращает результат (*result*).

Если выражение не совпадает ни с одним из искомых значений, а значение по умолчанию не задано, функция возвращает неопределенное значение.



Oracle: SQL и PL/SQL 3-37

Использование функции DECODE

```
SELECT job, sal ,
       DECODE(job, 'ANALYST', SAL*1.1, 'CLERK', SAL*1.15,
              'MANAGER', SAL*1.20, SAL)
REVISSED_SALARY
FROM emp
```

JOB	SAL	REVISSED_SALARY
-----	-----	-----
PRESIDENT	5000	5000
MANAGER	2850	3420
MANAGER	2450	2940
...		

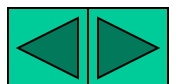
14 rows selected.

Использование функции DECODE

В вышеуказанной команде SQL расшифровывается значение JOB. Если значение JOB равно ANALYST, прибавка к окладу составляет 10%; если значение JOB равно CLERK, прибавка к окладу составляет 15%; если значение JOB равно MANAGER, прибавка к окладу составляет 20%. Для остальных должностей оклады не увеличиваются.

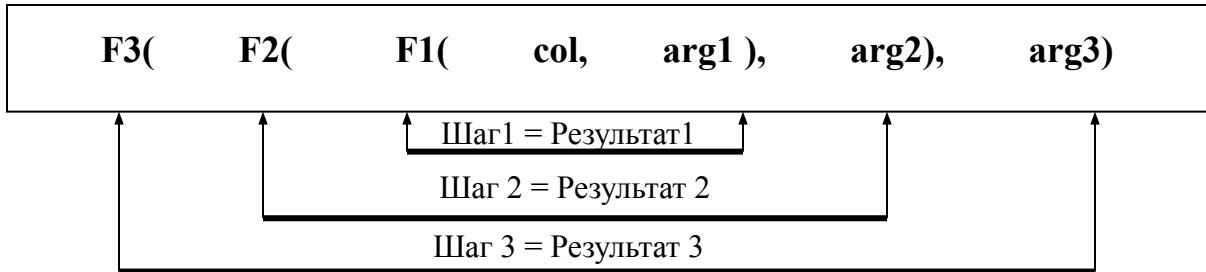
То же самое можно сделать с помощью команды IF-THEN-ELSE:

```
IF job = 'ANALYST'           THEN sal = sal*1.1
IF job = 'CLERK'             THEN sal = sal*1.15
IF job = 'MANAGER'           THEN sal = sal*1.20
ELSE sal = sal
```



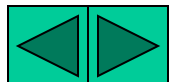
Вложенные функции

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



Вложенные функции

Однострочные функции могут быть вложены на любую глубину. Вложенные функции вычисляются от самой внутренней к самой внешней. Примеры демонстрируют гибкость этих функций.



Вложенные функции

```

SELECT  ename,NVL (TO_CHAR(mgr),  'No Manager')
FROM    emp
WHERE   mgr IS NULL

```

```

ENAME          NVL (TO_CHAR (MGR) ,  'NOMANAGER')
-----
KING           No Manager

```

Вложенные функции (продолжение)

В примере на слайде на экран выводится фамилия главы компании, который не имеет менеджера. Оценка команды SQL включает два шага:

1. Вычисление внутренней функции для преобразования числа в символьную строку.

- Result1 = TO_CHAR (mgr)

2. Вычисление внешней функции для замены неопределенного значения текстовой строкой.

- NVL (Result1, 'No Manager')

Т.к. псевдоним не задан, все выражение становится заголовком столбца.

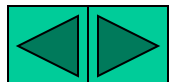
Пример

Вывод даты первой пятницы через 6 месяцев с даты найма. Формат даты - ''Friday, March 12th, 1982''. Результат упорядочен по датам найма.

```

SELECT  TO_CHAR (NEXT_DAY
              (ADD_MONTHS(hiredate, 6), 'FRIDAY'),
              'fmDay, Month ddth, YYYY') "Next 6 Month Review"
FROM    emp
ORDER BY hiredate

```



Заключение

С помощью функций осуществляются:

- Вычисления с данными
- Изменение отдельных элементов данных
- Манипулирование выводом групп строк
- Изменение форматов дат для вывода
- Преобразование формата данных столбцов

Однострочные функции

Однострочные функции могут быть вложены на любую глубину. С их помощью можно манипулировать следующими данными:

- Символьные данные
 - LOWER, UPPER, INITCAP, CONCAT, SUBSTR, INSTR, LENGTH
- Числовые данные
 - ROUND, TRUNC, MOD
- Даты
 - MONTHS_BETWEEN, ADDJHONTHS, NEXT_DAY, LAST_DAY, ROUND, TRUNC
 - Выражения с датами могут также использовать арифметические операторы.
- Функции преобразования могут использоваться для преобразования символьных данных, дат и чисел.
 - TO_CHAR, TO_DATE, TO_NUMBER

SYSDATE и DUAL

SYSDATE – это функция, возвращающая текущую дату и время. Обычно SYSDATE выбирается из фиктивной таблицы DUAL.



Вопросы к практическому занятию 3

1. Напишите запрос для вывода текущей даты. Назовите столбец **Date** .
2. Выведите номер служащего , его фамилию, оклад и новый оклад, повышенный на 15% и округленный до целого. Назовите столбец **New Salary** .
3. Измените предыдущий запрос . Добавьте еще один столбец , который будет содержать результат, вычитания старого оклада из нового. Назовите столбец **Increase** .
4. Для каждого служащего выведите фамилию, дату найма и дату пересмотра зарплаты, которая приходится на первый понедельник после шести месяцев работы. Назовите столбец **REVIEW**. Формат даты на выходе - как “Sunday, the Seventh of September, 1981“.
5. Для каждого служащего выведите фамилию и вычислите количество месяцев со дня найма до настоящего времени, округленное до ближайшего целого . Назовите столбец **MONTHS_WORKED** . Результаты отсортируйте по количеству отработанных месяцев .
6. Получите по каждому служащему отчет в следующем виде: <фамилия> зарабатывает <оклад> в месяц , но желает <утроенный оклад>. Назовите столбец **Dream Salaries** .
7. Напишите запрос для вывода фамилий и окладов всех служащих . Назовите выходной столбец **SALARY** . Длина столбца **SALARY** - 15 символов с заполнением символом \$ слева .
8. Выведите фамилии служащих (первая буква каждой фамилии должна быть заглавной, а остальные - строчными) и длину каждой фамилии для тех служащих, фамилии которых начинается с символа *J, A* или *M* . Присвойте соответствующие заголовки столбцам.
9. По каждому служащему выведите фамилию, дату найма и день недели, когда он был нанят на работу. Назовите последний столбец **DAY** . Отсортируйте результаты по дням недели, начиная с понедельника.
10. Напишите запрос для вывода фамилии и суммы комиссионных каждого служащего. Если служащий не зарабатывает комиссионных , укажите в столбце “No Commission” . Назовите столбец **COMM** .