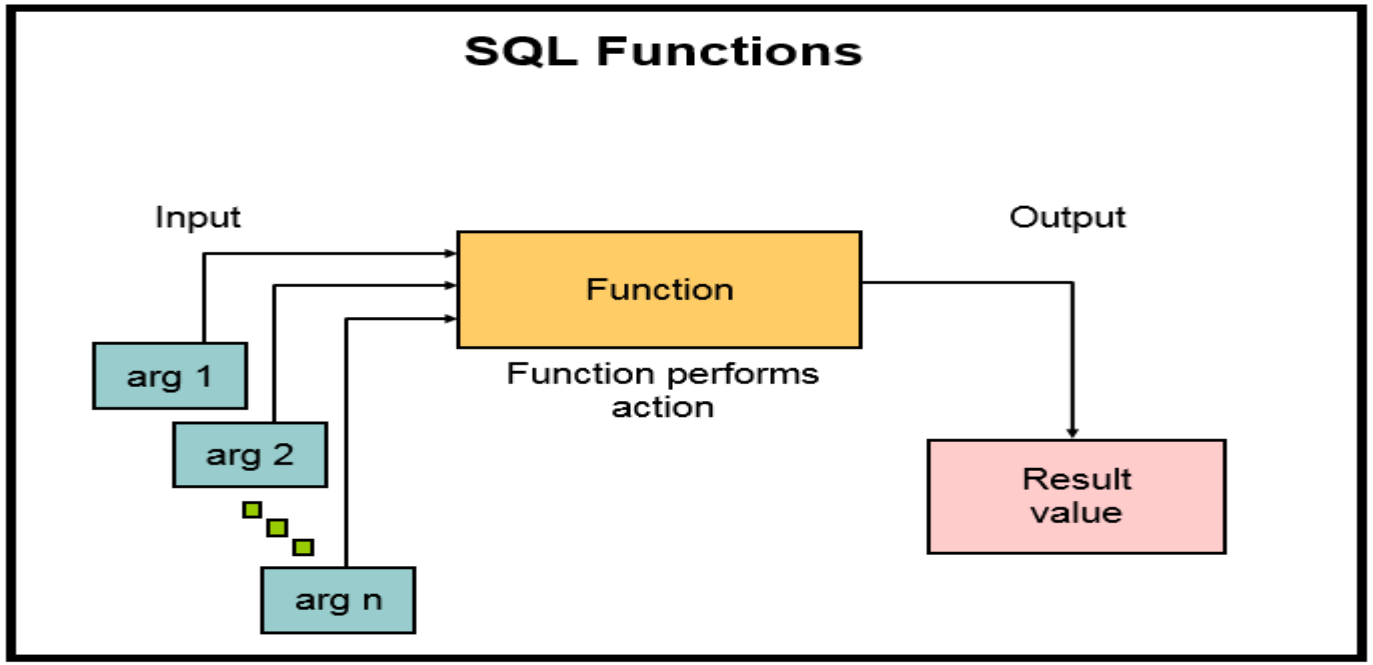
Using Single-Row Functions to Customize Output

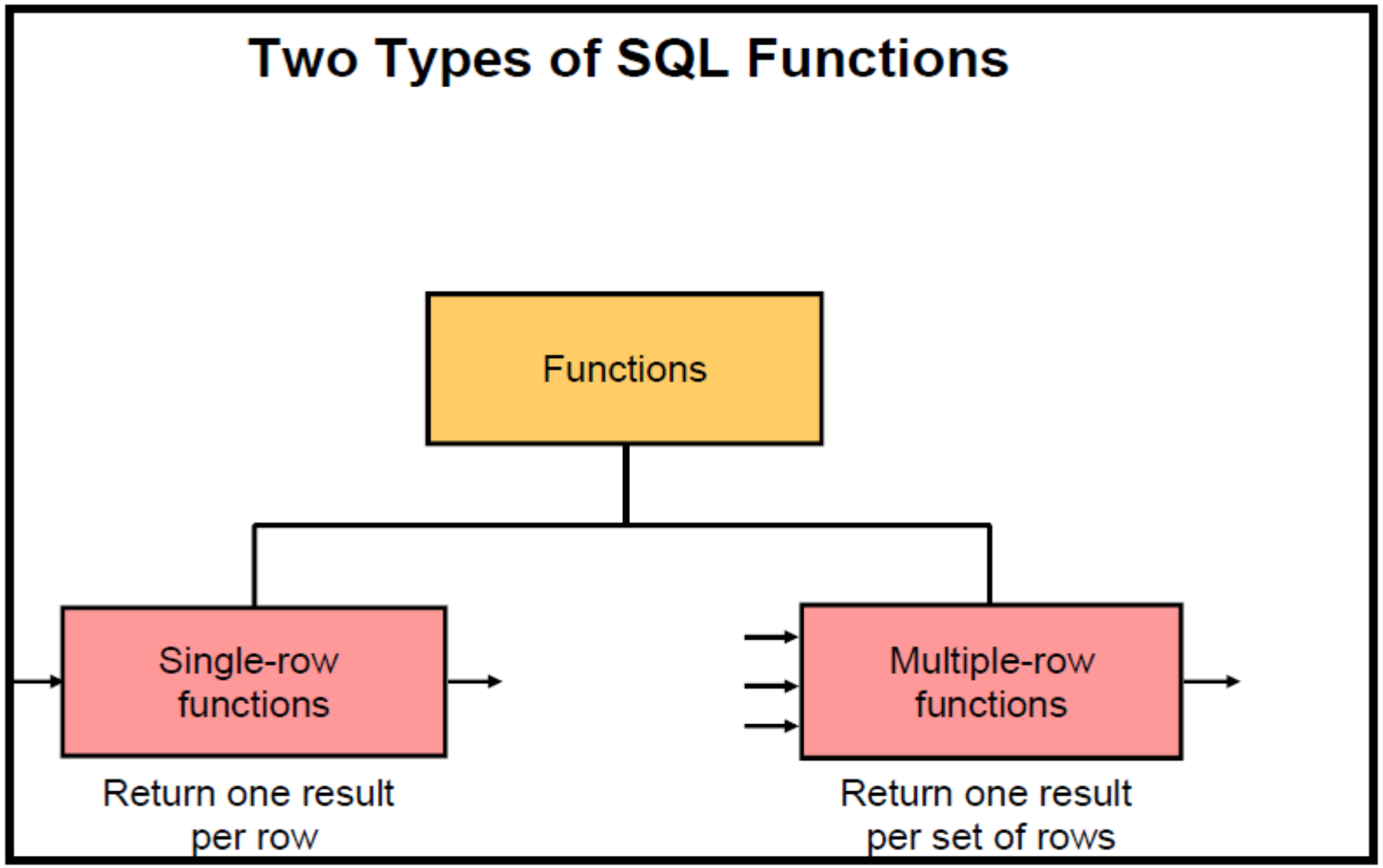
A function is a subprogram that return a Value



Functions are a very powerful feature of SQL. They can be used to do the following:

- Perform calculations on data
- Modify individual data items
- Manipulate output for groups of rows
- Format dates and numbers for display
- Convert column data types

SQL functions sometimes take arguments and always return a value.


Two Types of SQL Functions
Functions
Single-row functions
Multiple-row functions
Return one result per row
Return one result per set of rows

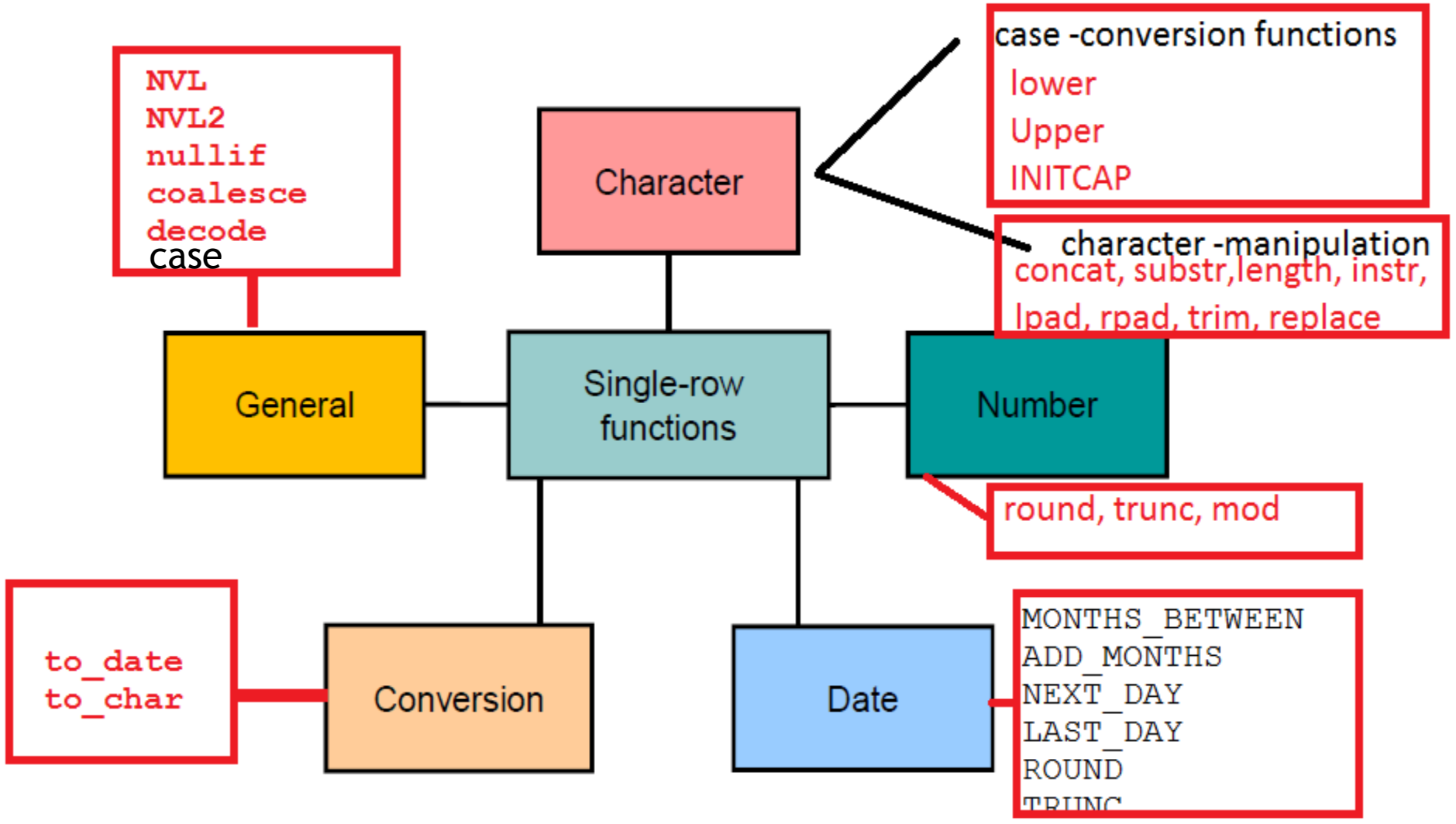
Single-Row Functions

Single-row functions:

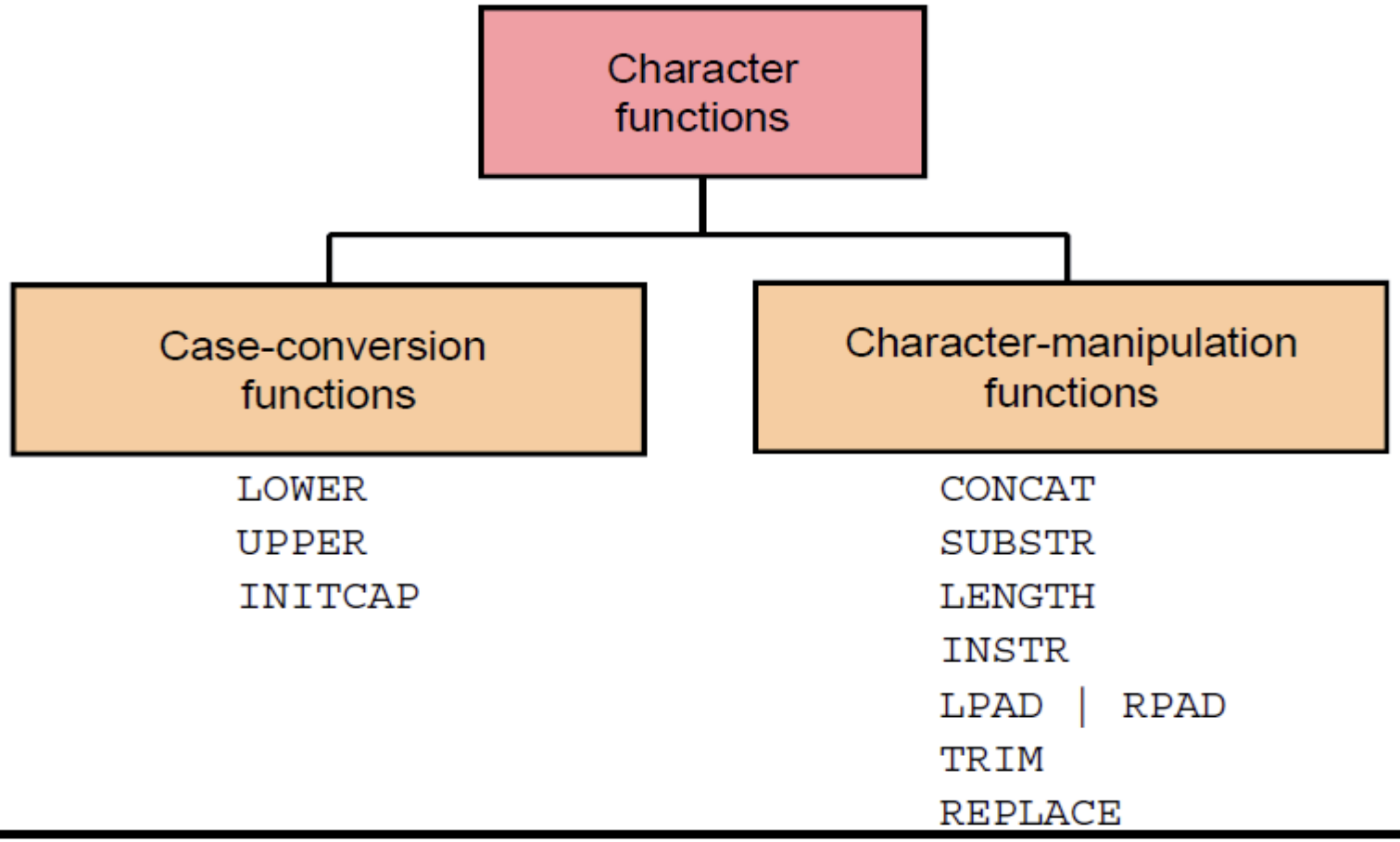
- Manipulate data items
- Accept arguments and return one value
- Act on each row that is returned
- Return one result per row
- May modify the data type
- Can be nested
- Accept arguments that can be a column or an expression

```
function_name [(arg1, arg2,...)]
```

Single-Row Functions



Character Functions



Function	Purpose
LOWER(*column|expression*)	Converts alpha character values to lowercase
UPPER(*column|expression*)	Converts alpha character values to uppercase
INITCAP(*column|expression*)	Converts alpha character values to uppercase for the first letter of each word; all other letters in lowercase
CONCAT(*column1|expression1, column2|expression2*)	Concatenates the first character value to the second character value; equivalent to concatenation operator (||)
SUBSTR(*column|expression,m[,n]*)	Returns specified characters from character value starting at character position *m*, *n* characters long (If *m* is negative, the count starts from the end of the character value. If *n* is omitted, all characters to the end of the string are returned.)

Function	Purpose
LENGTH(*column|expression*)	Returns the number of characters in the expression
INSTR(*column|expression, 'string', [,m], [n]*)	Returns the numeric position of a named string. Optionally, you can provide a position *m* to start searching, and the occurrence *n* of the string. *m* and *n* default to 1, meaning start the search at the beginning of the string and report the first occurrence.
LPAD(*column|expression, n,* '*string*') RPAD(*column|expression, n,* '*string*')	Returns an expression left-padded to length of *n* characters with a character expression. Returns an expression right-padded to length of *n* characters with a character expression.
TRIM(*leading|trailing|both, trim_character FROM trim_source*)	Enables you to trim leading or trailing characters (or both) from a character string. If *trim_character* or *trim_source* is a character literal, you must enclose it in single quotation marks. This is a feature that is available in Oracle8*i* and later versions.
REPLACE(*text, search_string, replacement_string*)	Searches a text expression for a character string and, if found, replaces it with a specified replacement string

Numeric Functions

- ROUND: Rounds value to a specified decimal
- TRUNC: Truncates value to a specified decimal
- MOD: Returns remainder of division

Function	Result
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92
MOD(1600, 300)	100

Function	Purpose
ROUND(*column*|*expression*, *n*)	Rounds the column, expression, or value to *n* decimal places or, if *n* is omitted, no decimal places (If *n* is negative, numbers to the left of decimal point are rounded.
TRUNC(*column*|*expression*, *n*)	Truncates the column, expression, or value to *n* decimal places or, if *n* is omitted, *n* defaults to zero
MOD(*m*,*n*)	Returns the remainder of *m* divided by *n*

Working with Dates

- The Oracle Database stores dates in an internal numeric format: century, year, month, day, hours, minutes, and seconds.
- The default date display format is DD-MON-RR.
  - Enables you to store 21st-century dates in the 20th century by specifying only the last two digits of the year
  - Enables you to store 20th-century dates in the 21st century in the same way

```
SELECT last_name, hire_date
FROM   employees
WHERE  hire_date < '01-FEB-08';
```

Arithmetic with Dates

- Add to or subtract a number from a date for a resultant date value.
- Subtract two dates to find the number of days between those dates.
- Add hours to a date by dividing the number of hours by 24.

Operation	Result	Description
date + number	Date	Adds a number of days to a date
date – number	Date	Subtracts a number of days from a date
date – date	Number of days	Subtracts one date from another
date + number/24	Date	Adds a number of hours to a date

Date-Manipulation Functions

Function	Result
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Week day of the date specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

Date functions operate on Oracle dates. All date functions return a value of the DATE data type except MONTHS_BETWEEN, which returns a numeric value.

- **MONTHS_BETWEEN(*date1, date2*)**: Finds the number of months between *date1* and *date2*. The result can be positive or negative. If *date1* is later than *date2*, the result is positive; if *date1* is earlier than *date2*, the result is negative. The noninteger part of the result represents a portion of the month.
- ADD_MONTHS(*date, n*): Adds *n* number of calendar months to *date*. The value of *n* must be an integer and can be negative.
- **NEXT_DAY(*date, 'char'*)**: Finds the date of the next specified day of the week (*'char'*) following *date*. The value of *char* may be a number representing a day or a character string.
- **LAST_DAY(*date*)**: Finds the date of the last day of the month that contains *date*

The above list is a subset of the available date functions. ROUND and TRUNC number functions can also be used to manipulate the date values as shown below:

- **ROUND(*date*[,*'fmt'*])**: Returns *date* rounded to the unit that is specified by the format model *fmt*. If the format model *fmt* is omitted, *date* is rounded to the nearest day.
- **TRUNC(*date*[, *'fmt'*])**: Returns *date* with the time portion of the day truncated to the unit that is specified by the format model *fmt*. If the format model *fmt* is omitted, *date* is truncated to the nearest day.

Using ROUND and TRUNC Functions with Dates

Assume SYSDATE = '25-JUL-03':

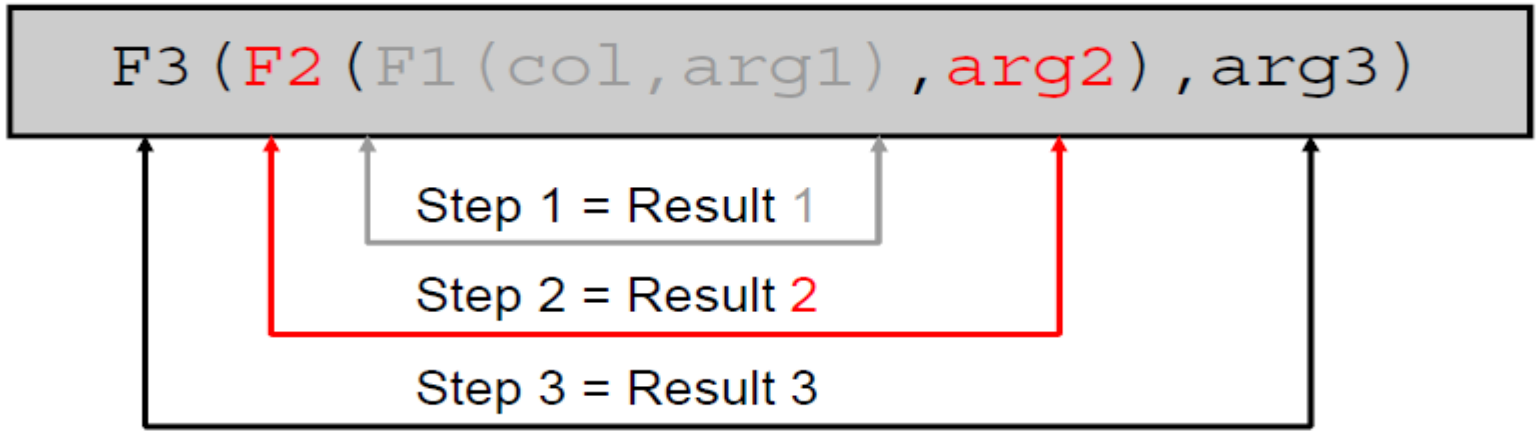
Function	Result
ROUND(SYSDATE,'MONTH')	01-AUG-03
ROUND(SYSDATE ,'YEAR')	01-JAN-04
TRUNC(SYSDATE ,'MONTH')	01-JUL-03
TRUNC(SYSDATE ,'YEAR')	01-JAN-03

The ROUND and TRUNC functions can be used for number and date values. When used with dates, these functions round or truncate to the specified format model. Therefore, you can round dates to the nearest year or month. If the format model is month, dates 1-15 result in the first day of the current month. Dates 16-31 result in the first day of the next month. If the format model is year, months 1-6 result in January 1 of the current year. Months 7-12 result in January 1 of the next year.

maxvlearn

Nesting Functions

- Single-row functions can be nested to any level.
- Nested functions are evaluated from the deepest level to the least deep level.



SUBSTR(UPPER(first_name),1,3)

Nested function example: split string into 3 segments

- Segment 1= xxxxxx

```
SUBSTR('String',1,INSTR('String',' ' ,1,1)-1 ) FIRST_NAME
```

- Segment 2= xxxxxxxxxxxxx

```
SUBSTR('string', INSTR('string ',' ',1,1)+1,
INSTR('string',' ',1,2)-INSTR('string',' ',1,1) -1) MIDDLE_NAME
```

- Segment 3= xxxxxxxxxxxxxxx

```
SUBSTR('string', INSTR('string',' ',1,2)+1) last_name
```

Thank You