Experiment 6

BUILT-IN FUNCTIONS IN RDBMS

AIM :

To familiarize with numeric, date and string functions

DESCRIPTION

Date functions

To manipulate and extract values from the date column of a table

ADD_MONTHS: Returns date after adding the number of months specified in the function.

Syntax:

ADD_MONTHS (d, n)

LAST_DAY: Returns the last date of the month specified with the function

Syntax:

LAST_DAY (d)

MONTHS_BETWEEN: Returns number of months between d1 and d2

Syntax:

MONTHS_BETWEEN (d1, d2)

NEXT_DAY: Returns the date of the first weekday named by char that is after the date named by date.

Syntax:

NEXT_DAY (date, char)

ROUND: Returns a date rounded to a specific unit of measure.

Syntax:

ROUND (date, [format])

Numeric functions

ABS: Returns the absolute value of **n**.

Syntax:

ABS (n)

POWER: Returns the m raised to the n^{th} power.

Syntax:

POWER (m, n)

ROUND: Returns n, rounded to m places to the right of a decimal point.

Syntax:

ROUND (n, m)

SQRT: Returns the square root of n.

Syntax:

SQRT (n)

EXP: Returns e raised to the n^{th} power, where e=2.71828183.

Syntax:

EXP (n)

GREATEST: Returns the greatest value in a list of expressions.

Syntax:

GREATEST (expr1, expr2, … expr_n)

Where expr1, expr2… expr_n are expressions that are evaluated by the greatest function.

LEAST: Returns the least value in a list of expressions.

Syntax:

LEAST (expr1, expr2… expr_n)

Where expr1, expr2… expr_n are expressions that are evaluated by the least function.

MOD: Returns the remainder of a first number divided by second number passed a parameter.

Syntax:

MOD (m, n)

TRUNC: Returns a number truncated to a certain number of decimal places.

Syntax:

TRUNC (number, decimal_places)

FLOOR: Returns the largest integer value that is equal to or less than a number.

Syntax:

FLOOR (n)

CEIL: Returns the smallest integer value that is equal to or greater than a number.

Syntax:

CEIL (n)

String Functions:

LOWER: Returns char, with all letters in lower case.

Syntax:

LOWER (char)

Example:

SELECT LOWER (‘ ICET’) “ Lower” FROM DUAL;

INITCAP: Returns a string with the first letter of each word in upper case.

Syntax:

INITCAP (char)

UPPER: Returns char, with all letters forced to upper case.

Syntax:

UPPER (char)

SUBSTR: Returns a portion of characters, beginning at character m, and going up to character n.

Syntax:

SUBSTR (<string>, <start_position>, [<length>])

Where **string** is the source string

Start_position is the position for extraction

Length is the number of characters to extract.

INSTR: Returns the location of a sub string in a string.

Syntax:

SUBSTR (<string1>, <string2>,<start_position>, [<nth_appearance>])

Where **string1** is the string to search

String2 is the sub string to search for in string1.

Start_position is the position in string1 where the search will start

n^{th}_appearance is the n^{th} appearance of string2.

TRANSLATE: Replaces a sequence of characters in a string with another set of characters.

Syntax:

TRANSLATE (<string1>, <string_to_replace>, <replacement_string>)

Where **string1** is the string to replace a sequence of characters with another set of characters.

String_to_replace is the string that will be searched for in string1.

All characters in the **string_to_replace** will be replaced with the corresponding character in the replacement string.

LENGTH: Returns the length of a word.

Syntax:

LENGTH (word)

Example:

SELECT LENGTH(‘ ICET’) “ Length” FROM DUAL;

LTRIM: Removes characters from the left of char with initial characters removed up to the first character not in set.

Syntax:

LTRIM (char,set)

RTRIM: Returns char, with final characters removed after the last character not in the set

Syntax:

RTRIM (char,set)

TRIM: Removes all specified characters either from the beginning or the ending of a string.

Syntax:

LTRIM ([leading|trailing|both[<trim_character>FROM]]<string1>)

Where **leading**-remove **trim_string** from the front of **string1.**

trailing-remove **trim_string** from the end of **string1.**

both-remove **trim_string** from the front and end of **string1.**

Example:

SELECT LTRIM (‘ ICET ’) “ Trim Both sides” FROM DUAL;

SELECT TRIM (LEADING ‘ x’ FROM ‘ xxxICETxxx’)“ Remove Prefixes” FROM DUAL;

LPAD: Returns char1, left padded to length n with sequence of characters specified in char2

Syntax:

LPAD (char1, n, char2)

RPAD: Returns char1, right padded to length n with sequence of characters specified in char2

Syntax:

RPAD (char1, n, char2)

VSIZE: Returns then number of bytes in the internal representation of an expression.

Syntax:

VSIZE (<expression>)

Conversion Functions:

TO_NUMBER: Converts char, a CHARACTER value expressing a number, to a number data type

Syntax:

TO_NUMBER (char)

TO_CHAR(number conversion): Converts a value of a NUMBER data type to a character data type, using the optional format string.

Syntax:

TO_CHAR (n ,fmt)

TO_CHAR (date conversion): Converts a value of a DATE data type to a character data type, using the optional format string.

Syntax:

TO_CHAR (date, fmt)

TO_DATE (date conversion): Converts a character field to a date field.

Syntax:

TO_DATE (char, fmt)

QUESTIONS

1. Display today's date tomorrow's date.

2. Display the last date of this month.

3. Display the date of next monday.

4. Display the minimum length of employee name from employee table.

5. Display the details of employees whose name length is greater than 5.

6. Display the number of employees joined in each month.

7. Retrieve three characters from the word “ encyclopedia” with respect to 4th position.

8. Replace all a's in the word “ malayalam” .

9. Remove “ OR” from the word “ ORACLE” .

10.Write query to pad left of the word “ DATABASE” with “ .” to a column width of 12 and then pad right with “ _” upto a lenth of 15.

11. Write a query to to find daily salary of the employees in the table emp.Assume 30 days are in a month. Round daily salary into 2 positions. Then name the column as “ daily pay”