**Build-in Functions in RDBMS**

**AIM :**To familiarize with numeric,date and string functions

**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)**

Example**:**

SELECT ADD\_MONTHS (SYSDATE,4)“Add Months” FROM DUAL;

* + **LAST\_DAY:** Returns the last date of the month specified with the function

Syntax**:**

**LAST\_DAY (d)**

Example**:**

SELECT SYSDATE, LAST\_DAY (SYSDATE)“LastDay” FROM DUAL;

* + **MONTHS\_BETWEEN:** Returns number of months between d1 and d2

Syntax**:**

**MONTHS\_BETWEEN (d1, d2)**

Example**:**

SELECT MONTHS\_BETWEEN (’02-FEB-92’, ’02-JAN-92’)“Months” FROM DUAL;

* + **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)**

Example**:**

SELECT NEXT\_DAY (’06-JULY-02’, Saturday)“Next Day” FROM DUAL;

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

Syntax**:**

**ROUND (date, [format])**

Example**:**

SELECT ROUND (TO\_DATE (‘01-JUL-04’), ‘YYYY’)“Year” FROM DUAL;

**Numeric functions**

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

Syntax**:**

**ABS (n)**

Example**:**

SELECT ABS (-15)“Absolute” FROM DUAL;

* + **POWER:** Returns the m raised to the nth power.

Syntax**:**

**POWER (m, n)**

Example**:**

SELECT POWER (3,2) “Raised” FROM DUAL;

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

Syntax**:**

**ROUND (n, m)**

Example**:**

SELECT ROUND (15.19,1) “Round” FROM DUAL;

* + **SQRT:** Returns the square root of n.

Syntax**:**

**SQRT (n)**

Example**:**

SELECT SQRT (25) “Square Root” FROM DUAL;

* + **EXP:** Returns e raised to the nth power, where e=2.71828183.

Syntax**:**

**EXP (n)**

Example**:**

SELECT EXP (5) “Exponent” FROM DUAL;

* + **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.

Example**:**

SELECT GREATEST (4,5,17)“Num”, GREATEST (‘4’,‘5’,‘17’)“Text” FROM DUAL;

* + **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.

Example**:**

SELECT LEAST (4,5,17) “Num”, LEAST (‘4’,‘5’,‘17’)“Text” FROM DUAL;

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

Syntax**:**

**MOD (m, n)**

Example**:**

SELECT MOD (15,7) “Mod1”, MOD (’15.7’,‘ 7’)“Mod2” FROM DUAL;

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

Syntax**:**

**TRUNC (number, decimal\_places)**

Example**:**

SELECT TRUNC (125.815) “Trunc1”, TRUNC (125.815, -2)“Trunc2” FROM DUAL;

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

Syntax**:**

**FLOOR (n)**

Example**:**

SELECT FLOOR (24.8) “Flr1”, FLOOR (13.15)“Flr2” FROM DUAL;

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

Syntax**:**

**CEIL (n)**

Example**:**

SELECT CEIL (24.8) “Ceil1”, CEIL (13.15)“Ceil2” FROM DUAL;

* **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)**

Example**:**

SELECT INITCAP (‘ICET’) “Title Case” FROM DUAL;

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

Syntax**:**

**UPPER (char)**

Example**:**

SELECT UPPER (‘icet’) “Capitalised” FROM DUAL;

* + **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.

Example**:**

SELECT SUBSTR (‘SECURE’, 3, 4) “Substring” FROM DUAL;

* + **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

**nth\_appearance**is the nth appearance of string2.

Example**:**

SELECT INSTR (‘SCT on the net’,‘t’) “Instr1” FROM DUAL;

* + **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.

Example**:**

SELECT TRANSLATE (‘1sct523’, ‘123’, ‘7a9’) “Change” FROM DUAL;

* + **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)**

Example**:**

SELECT LTRIM (‘ICET’, ‘I’) “Ltrim” FROM DUAL;

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

Syntax**:**

**RTRIM (char,set)**

Example**:**

SELECT RTRIM (‘ICET’, ‘T’) “Rtrim” FROM DUAL;

* + **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)**

Example**:**

SELECT LPAD (‘ICET’,10, ‘\*’) “LPAD” FROM DUAL;

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

Syntax**:**

**RPAD (char1, n, char2)**

Example**:**

SELECT RPAD (‘ICET’,10, ‘\*’) “RPAD” FROM DUAL;

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

Syntax**:**

**VSIZE (<expression>)**

Example**:**

SELECT VSIZE (‘SCT on the net’) “Size” FROM DUAL;

* **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)**

Example**:**

SELECT TO\_CHAR (17145,‘$099,999’)“Char” FROM DUAL;

* + **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)**

Example**:**

SELECT TO\_CHAR (DT, ‘Month, DD, YYYY’)“New Date Format” FROM DUAL;

* + **TO\_DATE (date conversion):** Converts a character field to a date field.

Syntax**:**

**TO\_DATE (char, fmt)**

Example**:**

SELECT TO\_DATE (DT, ‘Month, DD, YYYY’)“New Date Format” FROM DUAL;