Database Systems LAB # 07

Built-In functions in RDBMS

Functions

Function and Operators accept zero or more arguments and both return one or more results. Both are used to manipulate individual data items. Operators differ from functional in that they don't follow the format of function_name (arg...). An argument is a user defined variables or constants. Most operators accept at most 2 arguments while the structure of functions permit to accept 3 or more arguments. Function can be classifies into **single row function and group functions**.

Single Row functions

A single row function or scalar function returns only one value for every row queries in table. Single row function can appear in a select command and can also be included in a where clause. The single row function can be broadly classified as,

- Date Function
- Numeric Function
- Character Function
- Conversion Function
- Miscellaneous Function

The example that follows mostly uses the symbol table "**dual**". It is a table, which is automatically created by oracle along with the data dictionary.

1. Date Function

They operate on date values and produce outputs, which also belong to date data type except for months, between, date function returns a number.

i. **Add months**

This function returns a date after adding a specified date with specified number of months.
Syntax: Add_months (d,n); where d-date n-number of months
Example: *Select add_months (sysdate,2) from dual;*

ii. **last_day**

It displays the last date of that month.
Syntax: last_day (d); where d-date
Example: *Select last_day ('1-jun-2009') from dual;*

iii. **Months_between**

It gives the difference in number of months between d1 & d2.
Syntax: months_between (d1, d2); where d1 & d2 –dates.

Example: *Select months_between ('1-aug-2009','1-jun-2009') from dual;*

iv. **next_day**

It returns a day followed the specified date.
Syntax: next_day (d, day);
Example: *Select next_day (sysdate,'wednesday') from dual*

v. **round**

This function returns the date, which is rounded to the unit specified by the format model.
Syntax: round (d, [fmt]);
Where d- date, [fmt] – optional. By default date will be rounded to the nearest day
Example: *Select round (to_date ('1-jun-2009','dd-mm-yy'),'year') from dual;*

2. Numerical Functions

Command	Query	Output
Abs(n)	Select abs(-15) from dual;	15
Ceil(n)	Select ceil(55.67) from dual;	56
Exp(n)	Select exp(4) from dual;	54.59
Floor(n)	Select floor(100.2) from dual;	100
Power(m,n)	Select power(4,2) from dual;	16
Mod(m,n)	Select mod(10,3) from dual;	1
Round(m,n)	Select round(100.256,2) from dual;	100.26
Trunc(m,n)	Select trunc(100.256,2) from dual;	100.23
Sqrt(m,n)	Select sqrt(16) from dual;	4

3. Character Functions

Command	Query	Output
initcap(char);	*select initcap("hello") from dual;*	Hello

lower (char); upper (char);	*select lower ('HELLO') from dual;* *select upper ('hello') from dual;*	hello HELLO
ltrim (char,[set]);	*select ltrim ('cseit', 'cse') from dual;*	it
rtrim (char,[set]);	*select rtrim ('cseit', 'it') from dual;*	cse
replace (char,search string, replace string);	*select replace ('jack and jue', 'j', 'bl') from dual;*	black and blue
substr (char,m,n);	*select substr ('information', 3, 4) from dual;*	form

4. Conversion Function

i. to_date()

Syntax: to_date (d, [format]);
This function converts character to date data format specified in the form character.
Example: *select to_date ('aug 15 2009','mm-dd-yy') from dual;*

ii. to_char()

Syntax: to_char (d, [format]);
This function converts date to a value of varchar type in a form specified by date format.
Example: *select to_char (sysdate, 'dd-mm-yy') from dual;*

5. Miscellaneous Functions

i. **uid** – This function returns the integer value (id) corresponding to the user currently logged in.
Example: *select uid from dual;*

ii. **user** – This function returns the logins user name.
Example: *select user from dual;*

iii. **nvl** – The null value function is mainly used in the case where we want to consider null values as zero.
Syntax; nvl(exp1, exp2)
If exp1 is null, return exp2. If exp1 is not null, return exp1.
Example: *select custid, shipdate, nvl(total,0) from order;*

iv. **vsize:** It returns the number of bytes in expression.
Example: *select vsize('tech') from dual;*

Group Functions

A group function returns a result based on group of rows.

i. **avg**
Example: *select avg (total) from student;*

ii. **max**
Example: *select max (percentagel) from student;*

iii. **min**
Example: *select min (marksl) from student;*

iv. **sum**
Example: *select sum(price) from product;*

Count Function

In order to count the number of rows, count function is used.

i. **count(*)** – It counts all, inclusive of duplicates and nulls.
Example: *select count(*) from student;*

ii. **count(col_name)–** It avoids null value.
Example: *select count(total) from order;*

iii. **count(distinct col_name)** – It avoids the repeated and null values.
Example: *select count(distinct ordid) from order;*

Group by clause
This allows us to use simultaneous column name and group functions.
Example: *Select max(percentage) from student group by deptname;*

Having clause
This is used to specify conditions on rows retrieved by using group by clause.
Example: *Select max(percentage) from student group by deptname having count(*)>=50;*

Special Operators:

In / not in – used to select an equi from a specific set of values.
Syntax: Select *XYZ from avc where WXY IN (10,8) ;*

Any - used to compare with a specific set of values
Syntax: Select XYZ from avc where WXY =any(10,8) ;

Between / not between – used to find between the ranges
Syntax: Select XYZ from avc where WXY between 8 and 10 ;

Like / not like – used to do the pattern matching

The LIKE Special Operator
The LIKE special operator is used in conjunction with wildcards to find patterns within string attributes. Standard SQL allows you to use the per-cent sign (%) and underscore (_) wildcard characters to make matches when the entire string is not known:
1. % means any and all following or preceding characters are eligible.

For example:
'J%' includes Johnson, Jones, Jernigan, July, and J-231Q. 'Jo%' includes Johnson and Jones. '%n' includes Johnson and Jernigan.
2. _ means any one character may be substituted for the underscore.

For example:
_234566789' includes 1234566789, 2234566789, and 3234566789.
_23_56678_' includes 1231566781, 1232566782, and 8239566788.

'_o_es' includes Jones, Cones, Cokes, totes, and roles.

CASE/DECODE

Compares an attribute or expression with a series of values and returns an associated value or default value if no match is found.
Syntax DECODE (e, x, y,
d) Where
e = attribute or expression
x = value with which to compare e
y = value to return in e = x
d = default value to return if e is not equal to x