

SQL Functions – Conditional Statements

Version 1.0

4 - CONDITIONAL STATEMENTS

4.1 Case When

The CASE statement chooses from a sequence of conditions, and executes a corresponding statement. The CASE statement evaluates a single expression and compares it against several potential values, or evaluates multiple Boolean expressions and chooses the first one that is TRUE.

Syntax :

```
CASE [ expression ]
WHEN condition_1 THEN result_1
WHEN condition_2 THEN result_2
...
...
WHEN condition_n THEN result_n
ELSE result
END
```

If expression = condition_1 then result_1 is displayed/returned else
 if expression = condition_2 then result_2 is displayed/returned else
 ...
 ...
 if expression = condition_n then result_n is displayed/returned else
 result is displayed

Example:

Consider the table Date_Table

```
SQL> select * from Date_Table;
```

DAY	MONTH	YEAR
2	1	2013
10	3	2013
20	6	2013
28	8	2013

Display the Date in Date_table. Display the month in words.

```

select t.day,
       t.month,
       t.year,
       case t.month
         when 1 then 'January'
         when 2 then 'February'
         when 3 then 'March'
         when 4 then 'April'
         when 5 then 'May'
         when 6 then 'June'
         when 7 then 'July'
         when 8 then 'August'
         when 9 then 'September'
         when 10 then 'October'
         when 11 then 'November'
         when 12 then 'December'
         else 'Invalid Month'
       End As Month_Name
from date_table t;

```

```

SQL> select
2      t.day,
3      t.month,
4      t.year,
5      case t.month
6        when 1 then 'January'
7        when 2 then 'February'
8        when 3 then 'March'
9        when 4 then 'April'
10       when 5 then 'May'
11       when 6 then 'June'
12       when 7 then 'July'
13       when 8 then 'August'
14       when 9 then 'September'
15       when 10 then 'October'
16       when 11 then 'November'
17       when 12 then 'December'
18       else 'Invalid Month'
19     End As Month_Name
20 from date_table t;

```

DAY	MONTH	YEAR	MONTH_NAME
2	1	2013	January
10	3	2013	March
20	6	2013	June
28	8	2013	August

4.2 Decode

The Decode function compares expression against each search value in order. If a match (equality) is found between the expression and the search argument then it returns the corresponding result. If there is no match, the default value is returned (if defined), else it returns NULL. This function has the functionality of an IF-THEN-ELSE statement.

Syntax :

```
decode( expression , search_1 , result_1 [, search_2 , result_2] ...  
          [, search_n , result_n] [, default] )
```

```
If expression = search_1 then result_1 is displayed/returned else  
  if expression = search_2 then result_2 is displayed/returned else  
    ...  
    ...  
    if expression = search_n then result_n is displayed/returned else  
      default is displayed
```

Example :

Display the Date in Date_table. Display the month in words.

```
select t.day, t.month, t.year,  
Decode( t.month,  
        '1' , 'January',  
        '2' , 'February',  
        '3' , 'March',  
        '4' , 'April',  
        '5' , 'May',  
        '6' , 'June',  
        '7' , 'July',  
        '8' , 'August',  
        '9' , 'September',  
        '10' , 'October',  
        '11' , 'November',  
        '12' , 'December',  
        'Invalid Month') Month_Name  
from date_table t;
```

```
SQL> select t.day, t.month, t.year,  
2      Decode(t.month,  
3          '1' , 'January',  
4          '2' , 'February',  
5          '3' , 'March',  
6          '4' , 'April',  
7          '5' , 'May',  
8          '6' , 'June',  
9          '7' , 'July',  
10         '8' , 'August',  
11         '9' , 'September',  
12         '10' , 'October',  
13         '11' , 'November',  
14         '12' , 'December',  
15         'Invalid Month') Month_Name  
16 from date_table t;
```

DAY	MONTH	YEAR	MONTH_NAME
2	1	2013	January
10	3	2013	March
20	6	2013	June
28	8	2013	August