SQL Day: 11 SQL Functions

2) Numeric Functions:

Numeric function accept numeric input and return numeric value.
Numeric functions are as below

1) Round()
2) trunc()
3) mod()

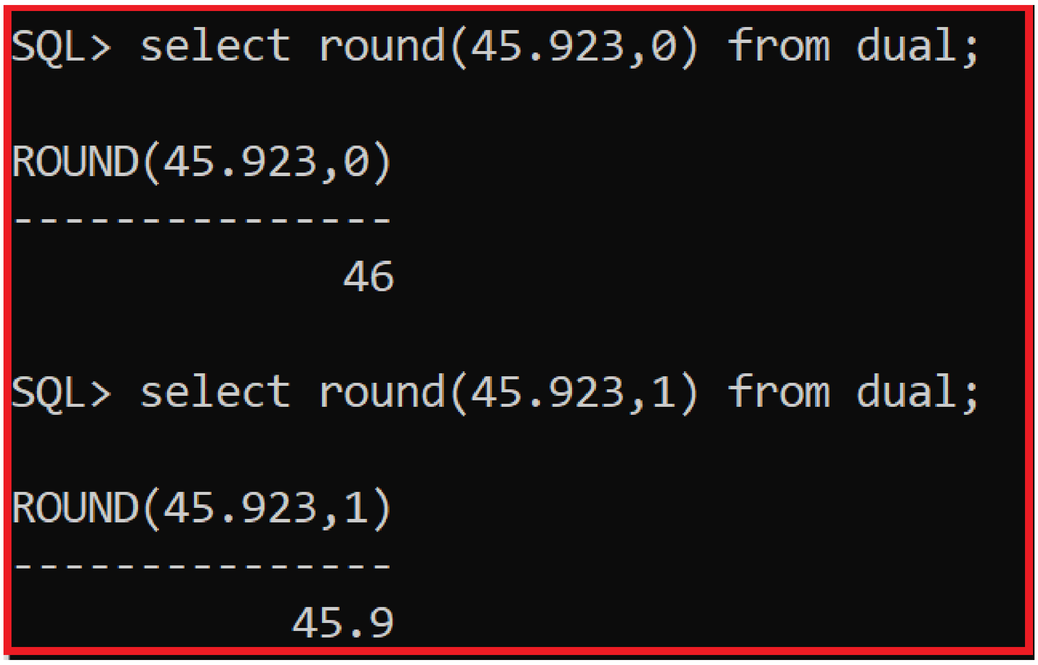
1) Roud()

=> This function round the value to specified decimal.
=> This function takes two input arguments, so syntax is as below

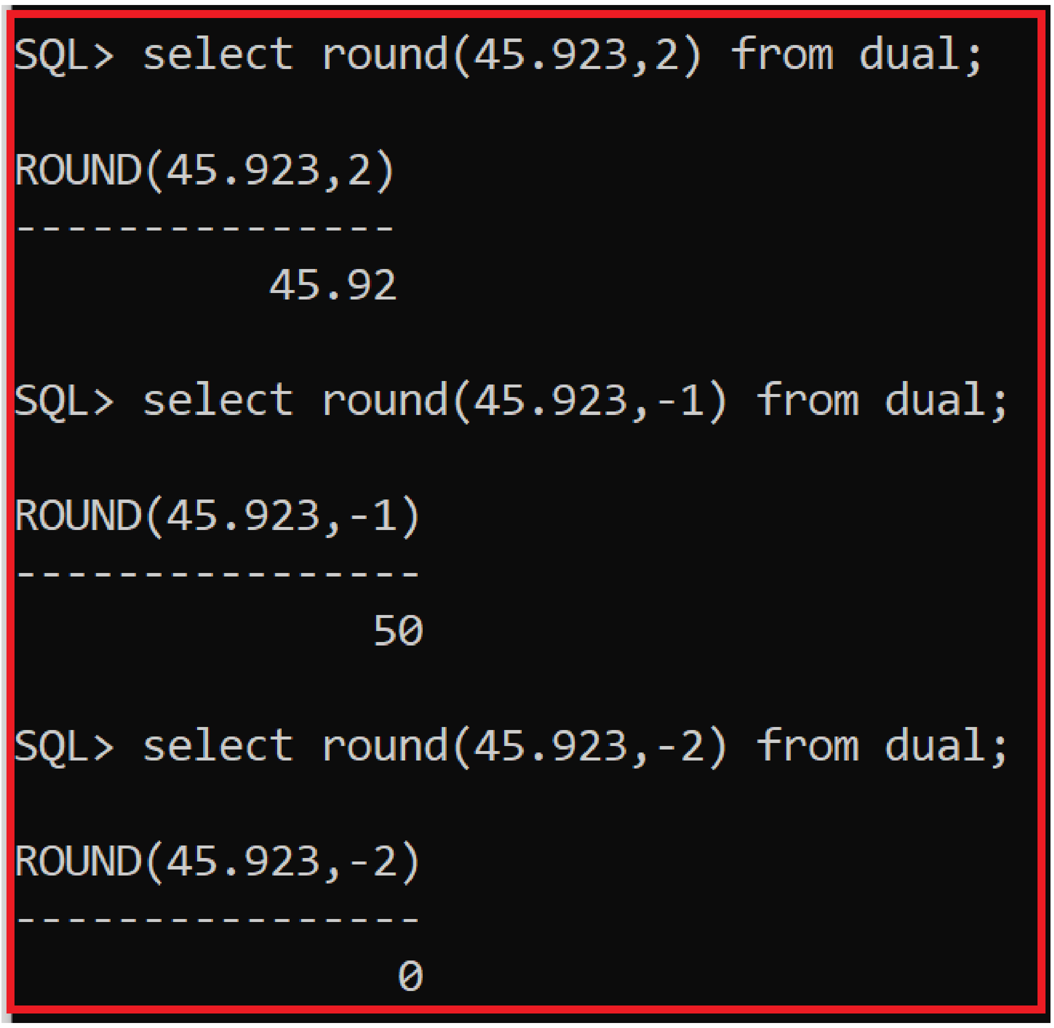
round(source_value,s)

Here, s stands for scale.Scale value is saying that after decimal point how many digits you want.

Examples



```
SQL> select round(45.923,0) from dual;

ROUND(45.923,0)
---------------
             46

SQL> select round(45.923,1) from dual;

ROUND(45.923,1)
---------------
           45.9
```



```
SQL> select round(45.923,2) from dual;

ROUND(45.923,2)
---------------
          45.92

SQL> select round(45.923,-1) from dual;

ROUND(45.923,-1)
----------------
              50

SQL> select round(45.923,-2) from dual;

ROUND(45.923,-2)
----------------
               0
```

If scale value less than -1 then you always get out put as 0

Here less than -1 means value is like -2, -3

2) trunc()

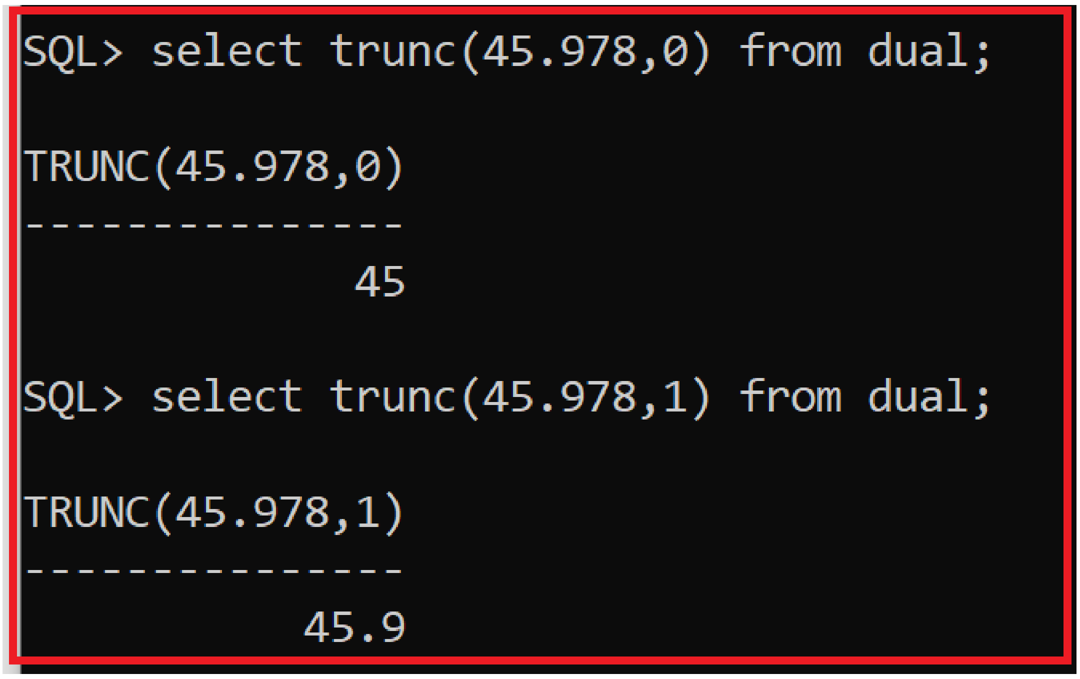
=>This function truncate(cut) the value to specified digit.

=> This function takes 2 input arguments, so syntax is as below

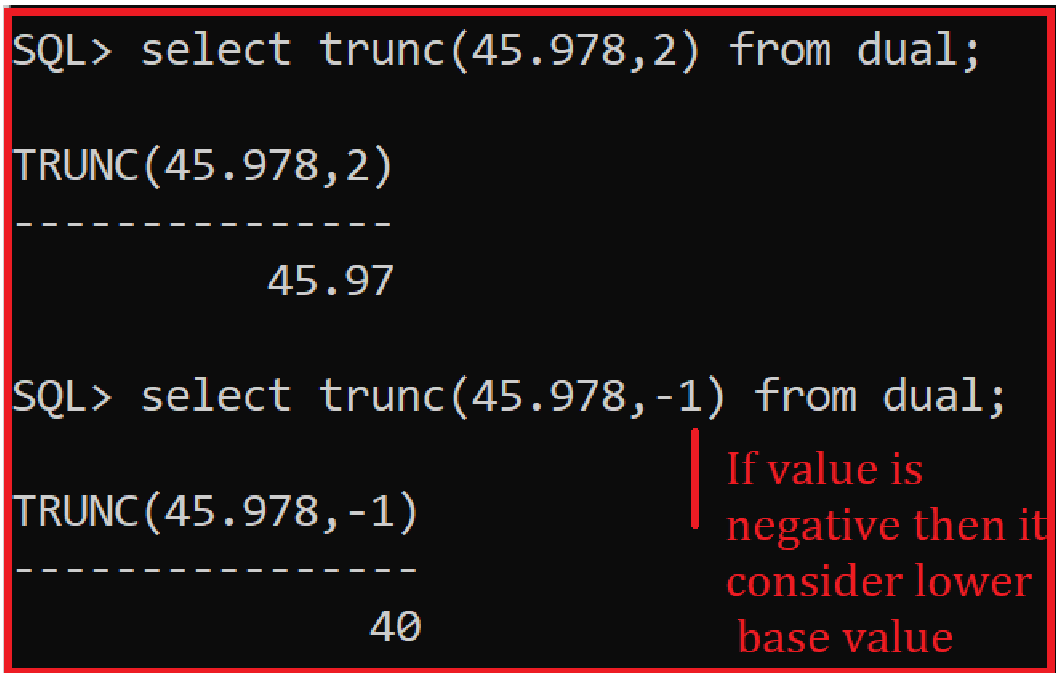
trunc(source value,s)

Here s stands for scale. Meaning of scale is -> after the decimal point how many digits you want.

Examples

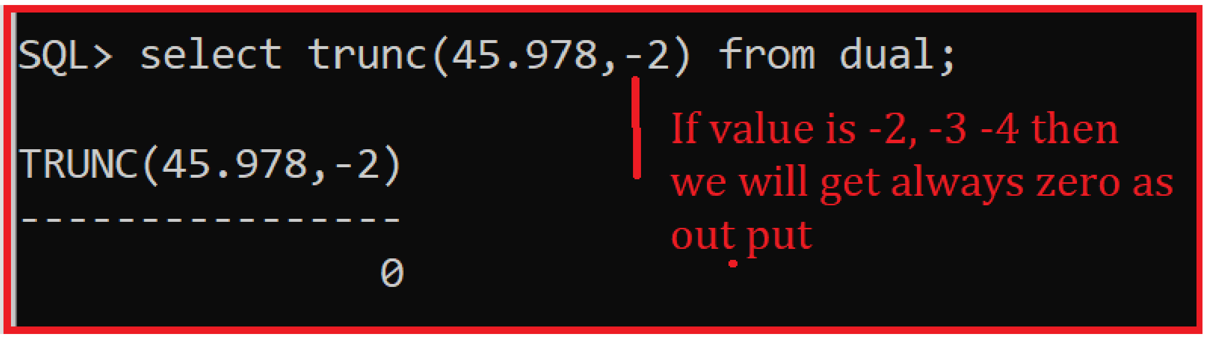


```
SQL> select trunc(45.978,0) from dual;

TRUNC(45.978,0)
---------------
             45

SQL> select trunc(45.978,1) from dual;

TRUNC(45.978,1)
---------------
           45.9
```



```
SQL> select trunc(45.978,2) from dual;

TRUNC(45.978,2)
---------------
          45.97

SQL> select trunc(45.978,-1) from dual;

TRUNC(45.978,-1)
----------------
              40
```

If value is negative then it consider lower base value



```
SQL> select trunc(45.978,-2) from dual;

TRUNC(45.978,-2)
----------------
               0
```

If value is -2, -3 -4 then we will get always zero as out put

Q. What is difference between round () function and trunc () function?

Ans:

round()	trunc()
round(45.978,0) => 46	trunc(45.978,0) => 45
round(45.978,1) => 46	trunc(45.978,1) =>45.9
round(45.978,-1) =>50	round(45,978,-1) => 40
round(43.978,-1) => 40	round (43.978,-1) =>40

3) Mod()

=> We know the Arithmetic Operators as below

1) Addition +
2) Substraction -
3) Multiplication *
4) Division /
5) Modulo %

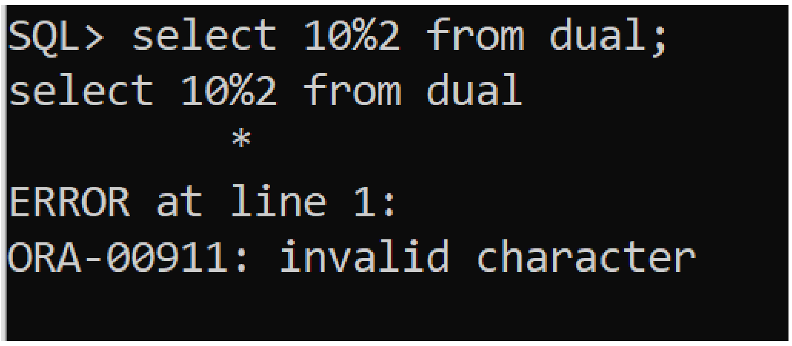
So If we perform query like

1) select 10+20 from dual;
   o/p=> 30

2) select 20-10 from dual;
   o/p=> 10

3) select 20*10 from dual;
   o/p=> 200
4) select 20/5 from dual;
   o/p=> 4

But if we perform operation like



```
SQL> select 10%2 from dual;
select 10%2 from dual
         *
ERROR at line 1:
ORA-00911: invalid character
```

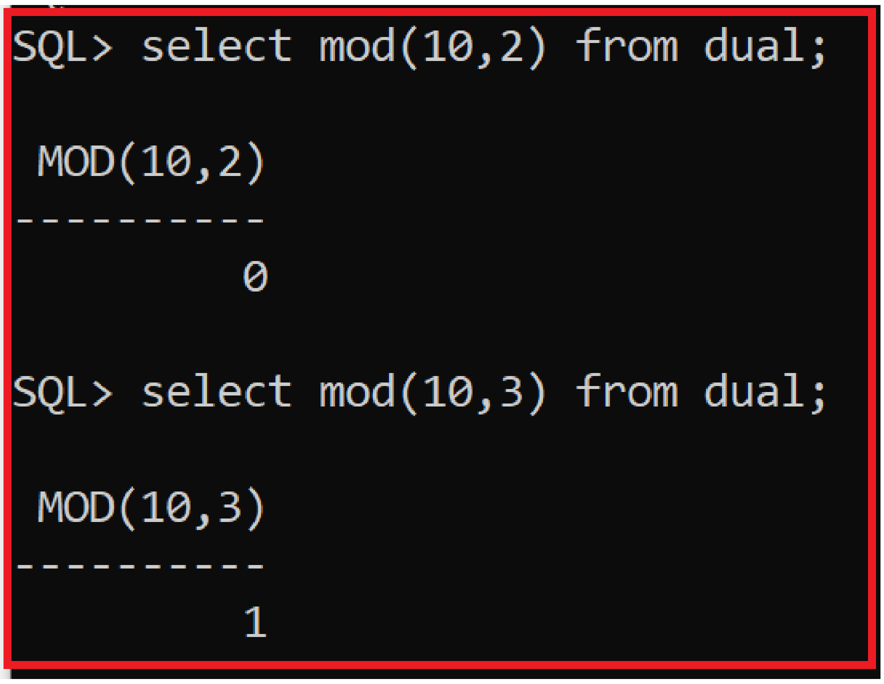
Then we will get an error saying "invalid character"
so Modulo operator % is not applicable in oracle.
So overcome this problem Oracle has provided mod()
function.

=> Mod() function is used to find the remainder

=> mod() function take two input arguments

mod(source value, divisor)

Examples:



```
SQL> select mod(10,2) from dual;

 MOD(10,2)
----------
         0

SQL> select mod(10,3) from dual;

 MOD(10,3)
----------
         1
```

3) Date functions
=> We have 3 data types - String,Numeber and Date

=> Date functions operate on Date data type.

=> All Date functions returns a value of Date data type value
except MONTHS_BETWEEN() function, which return numeric value.

=> Default date format in oracle is dd-MON-RR
Example:10-SEP-21
(Q. What is default date format in oracle?)

Date functions in Oracle are as below

1) sysdate
2) Months_Between()
3) Add_Months
4) Next_Day()
5) Last_Day()
6) Round()
7) Trunc()

1) sysdate

=> This function display or return system date.
=> Here system date means todays date.
=> This function does not take input arguments

Example : select sysdate from dual;
O/p => 10-SEP-21

2) MONTHS_BETWEEN()

=> This function returns number of months between two dates.
=> This function takes two input arguments.
so syntax is

MONTHS_BETWEEN(first_date_value, second_date_value)

Example: select months_between(sysdate, '10-SEP-20') from dual;

O/p: 12

Here sysdate is considered as 10-SEP-21

Example 2: select months_between('10-sep-20', '10-sep-21') from dual;
o/p=> -12

Here first date value is smaller than seond date value, so out put is negative.
so Pass the first date value higher than second date value.

3) ADD_MOTHS()
=> This function add the month in the given date and return date value.
=> This function takes two input arguments. so syntax is

```
add_months(source_date_value, no_of_month)
```

Here no_of_month value can positive or negative.
=>If value is positive then month is added in the given date

Example: select add_months('10-SEP-21',3) from dual;
=> 10-DEC-21)

=>If value is negative then month is substracted from given date

Example: select add_months('10-SEP-21',-3) from dual;
=> 10-JUN-21

4) NEXT_DAY():
=> This function returns next week day after the given date.

=> This function takes two input arguments,so syntax is

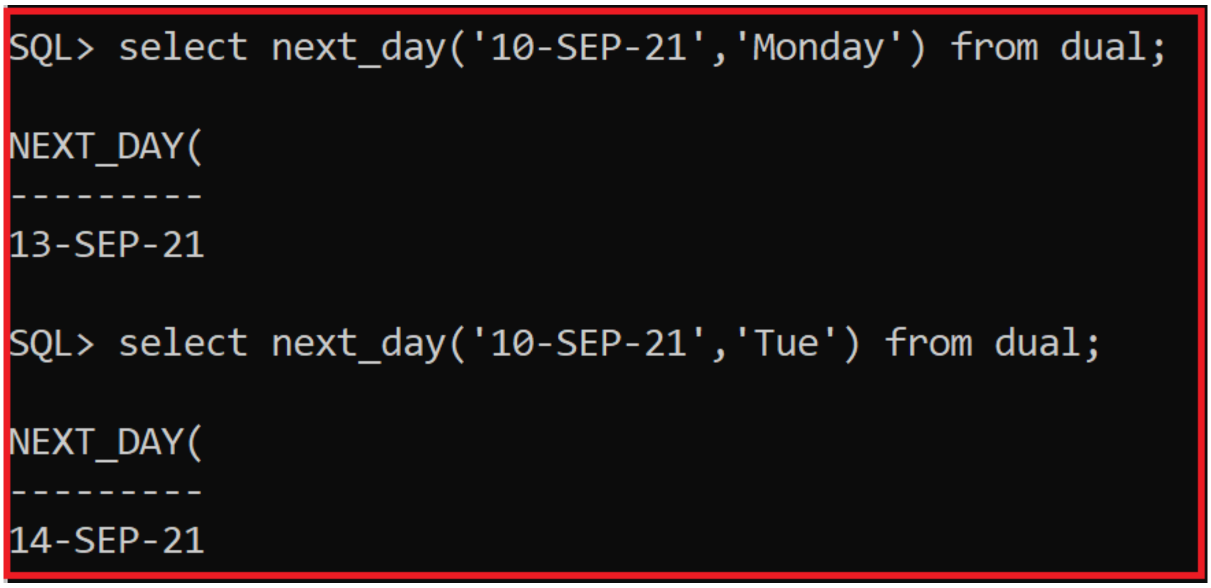
```
NEXT_DAY(source_date_value,weekday value)
```

here,For week day value we can pass week day full name or short name or numeric value as below

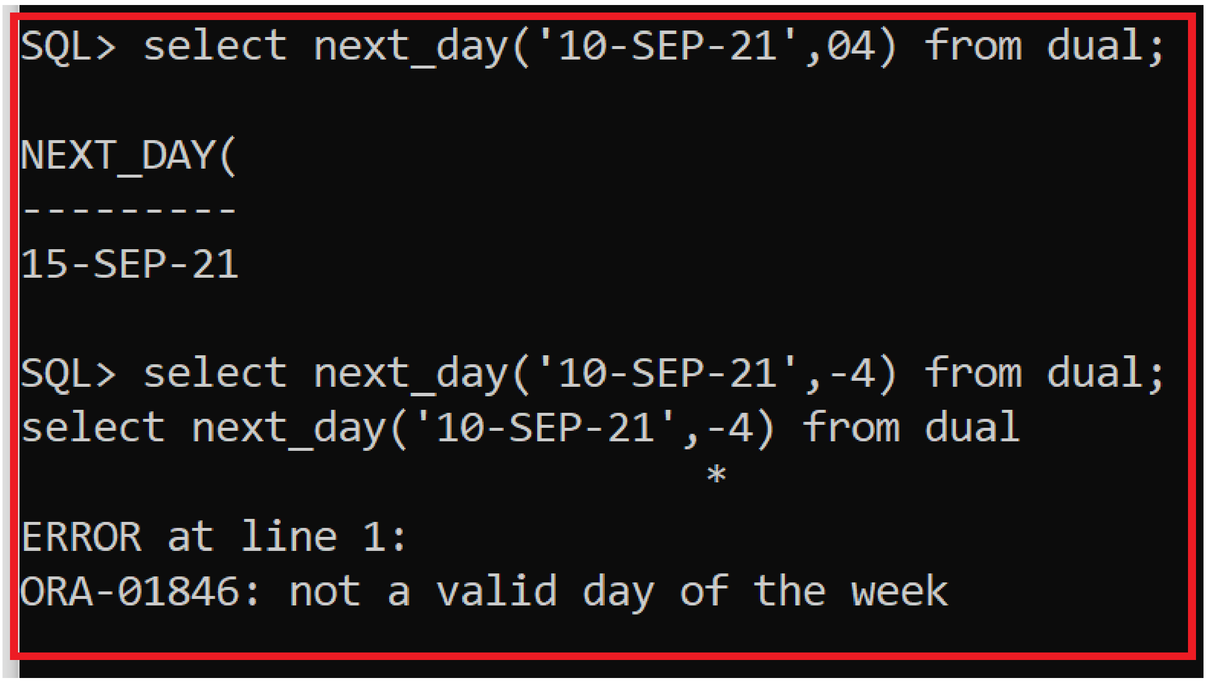
Full Name	Short Name	Numeric Value
Sunday	Sun	01
Monday	MON	02
Tuesday	Tue	03
Wednesday	Wed	04
Thursday	Thur	05
Friday	Fri	06
Saturday	Sat	07

Note: If we are passing week day value as numeric then it should be between 1 to 7, by mistake if we are passing other values then we will get an error saying : "not valid day of week"

Examples



```
SQL> select next_day('10-SEP-21','Monday') from dual;

NEXT_DAY(
---------
13-SEP-21

SQL> select next_day('10-SEP-21','Tue') from dual;

NEXT_DAY(
---------
14-SEP-21
```



```
SQL> select next_day('10-SEP-21',04) from dual;

NEXT_DAY(
---------
15-SEP-21

SQL> select next_day('10-SEP-21',-4) from dual;
select next_day('10-SEP-21',-4) from dual
                            *
ERROR at line 1:
ORA-01846: not a valid day of the week
```

5) Last_day()

=> This function returns last day of given date.

=> This function takes one input arguments
so Syntax is as below

```
last_day(date_value)
```

Example: select last_day('10-SEP-21') from dual;

o/p=> 30-SEP-21

6) round()

This function round the date value.

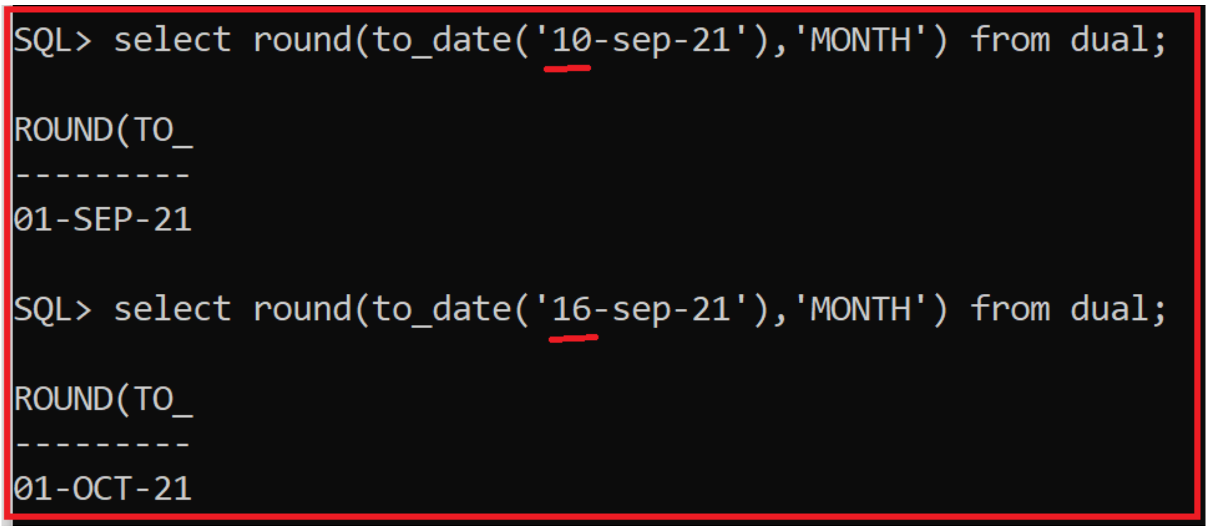
We can round month value and we can round year value.

Round-Month

If day of month value is between 1 to 15 then it will return 1 st day of same month

If day of month value is greater than 15 then it will retrun 1 st day of next month

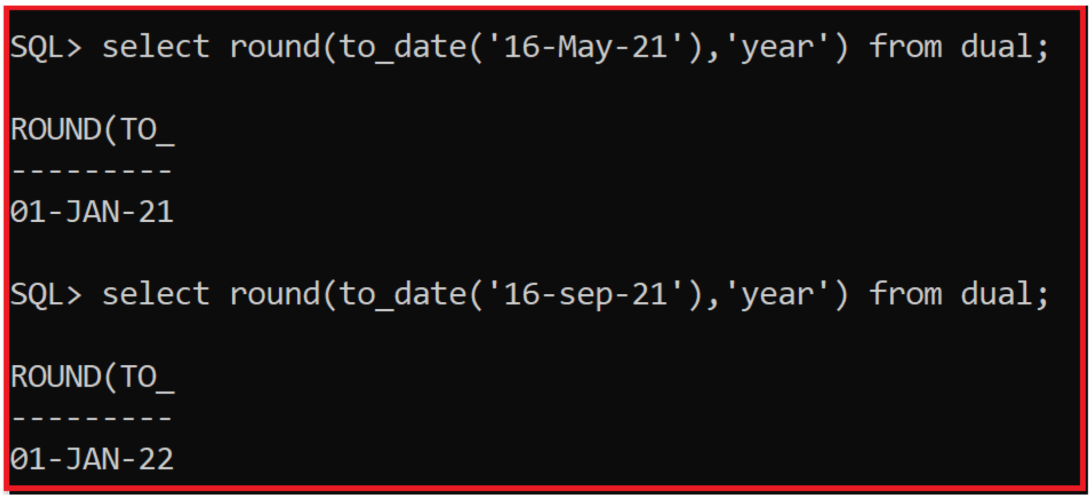
Examples- Round month


```
SQL> select round(to_date('10-sep-21'),'MONTH') from dual;

ROUND(TO_
---------
01-SEP-21

SQL> select round(to_date('16-sep-21'),'MONTH') from dual;

ROUND(TO_
---------
01-OCT-21
```

Round-year
If month value is between January to june then it will consider first January of same year

If Month value is between July to December then it will consider first jan of next year.

Examples:


```
SQL> select round(to_date('16-May-21'),'year') from dual;

ROUND(TO_
---------
01-JAN-21

SQL> select round(to_date('16-sep-21'),'year') from dual;

ROUND(TO_
---------
01-JAN-22
```

7) trunc()
We can truncate month value and year value

i) trunc-month

For trunc-month Trunc function always return first day of same month.

Example: select trunc(to_date('10-sep-21'), 'MONTH') from dual;

o/p=> 1-SEP-21

ii) For trunc-year

For trunc-year, trunc function always retrun First January of same year

Example: select trunc(to_date('10-sep-21'), 'year') from dual;

o/p=> 1-Jan-21