Numeric Functions in Oracle

ABS() Function

This Numeric Function in Oracle is used to convert (-VE) value into (+VE) value. That means the ABS function in Oracle returns the absolute value of n. This function takes as an argument any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. The function returns the same datatype as the numeric datatype of the argument.

```
SELECT ABS(-12) FROM DUAL;
```

CEIL() Function

This Numeric Function in Oracle is used to return a value that is greater than or equal to the given value. That means the CEIL function in oracle returns the smallest integer greater than or equal to n. This function takes as an argument any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. The function returns the same datatype as the numeric datatype of the argument.

```
SELECT CEIL(9.0) FROM DUAL;
```

```
SELECT CEIL(9.3) FROM DUAL;
```

FLOOR() Function

This numeric function is used to return the largest integer equal to or less than n. This function takes as an argument any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. The function returns the same datatype as the numeric datatype of the argument.

```
SELECT FLOOR(9.0) FROM DUAL;
```

```
SELECT FLOOR(9.8) FROM DUAL;
```

MOD() Function

The Oracle MOD() function is used to return the remainder value. That means this Numeric Function in Oracle is used to return the remainder of a dividend divided by a divisor. This function takes as arguments any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. Oracle

determines the argument with the highest numeric precedence, implicitly converts the remaining arguments to that datatype and returns that datatype.

```
SELECT MOD(10,2) FROM DUAL;
```

```
SELECT MOD(15,4) FROM DUAL;
```

POWER() Function

This numeric function is used to return the power of a given expression. The POWER Function in Oracle returns n2 raised to the n1 power. The base n2 and the exponent n1 can be any numbers, but if n2 is negative, then n1 must be an integer. This POWER function takes as arguments any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. If any argument is BINARY_FLOAT or BINARY_DOUBLE, then the function returns BINARY_DOUBLE. Otherwise, the function returns NUMBER.

```
SELECT POWER(2, 3) FROM DUAL;
```

ROUND() Function

This numeric function returns the nearest value of the given expression. The ROUND function in oracle returns n rounded to integer places to the right of the decimal point. If you omit integer, then n is rounded to 0 places. The argument integer can be negative to round off digits left of the decimal point.

```
SELECT ROUND(15.253, 1) FROM DUAL;
```

```
SELECT ROUND(15.253,-1) FROM DUAL;
```

```
SELECT ROUND(2.5), ROUND(3.5) FROM DUAL;
```

```
SELECT ROUND(2.5f), ROUND(3.5f) FROM DUAL;
```

```
SELECT ROUND(32.456,2) FROM DUAL;
```

TRUNC() Function

The TRUNC (number) function in Oracle is used to return n1 truncated to n2 decimal places. If n2 is omitted, then n1 is truncated to 0 places. n2 can be negative to truncate (make zero) n2 digits left of the decimal point. This function takes as an argument any numeric datatype or any non-numeric data type that can be implicitly converted to a numeric datatype. If you omit n2, then the function returns the same datatype as the numeric datatype of the argument. If you include n2, then the function returns NUMBER.

```
SELECT TRUNC(5.50) FROM DUAL;
```

```
SELECT TRUNC(32.456,2) FROM DUAL;
SELECT TRUNC(32.456,-1) FROM DUAL;
```