Numeric Functions

ROUND() Function:

The ROUND() function rounds a number to a specified number of decimal places.

Syntax:
ROUND(number, decimals)

1. Write a query to round off the number 56.3436 at second decimal place

```
SELECT ROUND(56.3436,2);
```

Output:

ROUND(56.3436)
56.34

If you observe from the above output the value is been rounded off to two decimal places

If you don’t specify the number of decimal values that time by default it will be considered as 0.

Now let’s understand how the precision works for negative numbers

round(45.65, 1) gives result = 45.7
round(45.65, -1) gives result = 50

because the precision in this case is calculated from the decimal point. If positive then it'll consider the right side number and round it upwards if it's >= 5, and if <=4 then round is downwards. and similarly if it's negative then the precision is calculated for the left hand side of the decimal point. If it's >= 5

For example, round(44.65, -1) gives 40 but round(45.65, -1) gives 50… (Here it is checking with left hand side)

2. Predict the output ROUND(563.3436, -1)

```
SELECT ROUND(563.3436,-1);
```

Output:

ROUND(563.3436)
560

As you can see the left side value is less than 4, precision is decremented

3. Predict the output ROUND(-16.56,-1)

```
SELECT ROUND(-16.56,-1);
```

Output:

ROUND(-16.56,-1)
-20

TRUNCATE() Function:

The TRUNCATE() function truncates a number to the specified number of decimal places.

Syntax:

TRUNCATE(number, decimals)

4. Write a query to truncate 456.432 to 2nd decimal places

```
SELECT TRUNCATE(456.432,2);
```

Output:

TRUNCATE(456.432,2)
456.43

5. Guess the output of TRUNCATE(456.556, -1)

```
SELECT TRUNCATE(456.556, -1);
```

Output:

TRUNCATE(456.556,-1)
450

MOD Function:
The MOD() function returns the remainder of a number divided by another number.

Syntax:
MOD(x, y)

1. Write a query to display reminder of a number 241 when divided by 2

```
SELECT MOD(241, 2);
```

Output:

MOD(241, 2)
1

2. Write a query to display the details of even number of rows in employee table

```
SELECT
    emp_id, first_name
FROM
    employee
WHERE
MOD(emp_id,2) = 0; //here we are checking whether the
number perfectly divided by 2
```

Output:

emp_id	first_name
2	Tom
4	andy
6	ram

8	john

3. Write a query to display the details of odd number of rows in employee table

```
SELECT
   emp_id, first_name
FROM
   employee
WHERE
NOT MOD(emp_id,2) = 0;
```

Output:

emp_id	first_name
1	kelly
3	mike
5	anjel
7	rohan

CEIL() Function:
CEIL() function is used to get the smallest integer which is greater than, or equal to, the specified numeric expression.

Syntax:
CEIL(number)

1. Write a query to get the ceil of 2.88

```
SELECT CEIL(2.88);
```

Output:

CEIL(2.88)
3

Here it is rounded off to the next greater number which is 3 here.

2. Write a query to get the ceil of 2

```
SELECT CEIL(2);
```

Output:

CEIL(2)
2

3. Write a query to get the ceil of -2.88

```
SELECT CEIL(-2.88);
```

Output:

CEIL(-2.88)
2

4. Query the salary of employees by incrementing it to 23.33% in integer format

```
SELECT
  CEIL(salary + salary * 23.33/100) as ceil
FROM
  employee;
```

Output:

ceil
96198
103844
121111
52046
52046
79178
103844
153176

FLOOR Function:

The FLOOR() function returns the largest integer value that is smaller than or equal to a number.

Syntax:
FLOOR(number)

1. Write a query to get the floor of 2.33

```
SELECT FLOOR(2.33);
```

Output:

FLOOR(2.33)
2

2. Write a query to get the floor of -2.33

```
SELECT FLOOR(-2.33);
```

Output:

FLOOR(2.33)
-3

POWER() Function:

The POWER() function returns the value of a number raised to the power of another number.

Syntax:
POWER(a, b)

1. Write a query to find cube of a number 3

```
SELECT POWER(3,3);
```

Output:

POWER(3,3)
27

SQRT() Function:
The SQRT() function returns the square root of a number.

Syntax:
SQRT(number)

1. Write a query to to find the square root of a number 16

```
SELECT SQRT(16);
```

Output:

SQRT(16)
4