

/* String functions*/

select ascii('CB'); /*return leftmost ascii value*/

select char(66); /*return ascii value to character*/

select len('Microsift sql');/*return length*/

select lower('JHON');/*convert to lowercase*/

select replace('Microsoft sql','sql','server');/*replace*/

select reverse('python');/*reverse the string */

select upper('aparna');/*converts to upper*/

select str(136.564,8,4);/*STR(number, length, decimals)*/

	(No column name)
1	67
	(No column name)
1	B
	(No column name)
1	13
	(No column name)
1	jhon
	(No column name)
1	Microsoft server
	(No column name)
1	nohtyp
	(No column name)
1	APARNA
	(No column name)
1	136.5640

/* Date Functions*/

select getdate ();/*get current date and time*/

select dateadd (mm, 2, '2023-12-07');/*add months to existed date */

select datediff (year, convert (datetime, '2006-05-06'), convert (datetime, '2009-01-01'))/*it will return the difference of date,months,years also*/

select datepart (mm, '2008-5-22');/*return months value*/

select day ('2023-05-30');/*return value of date of that particular day*/

select month ('2023-05-31');/*return month value*/

select year ('2023-05-3');/*return year value*/

	(No column name)
1	2024-11-08 09:11:30.410
	(No column name)
1	2024-02-07 00:00:00.000
	(No column name)
1	3
	(No column name)
1	5
	(No column name)
1	30
	(No column name)
1	5
	(No column name)
1	2023

/*Mathematical Functions*/

select abs(-101);/*returns absolute value*/

select sin(1.5);/*returns angle in radians*/

select ceiling(14.01);/*returns the smallest or greater to the specified value*/

select exp(4.5);/*returns the exponential value*/

select floor(14.75);

select log(5.4);/*return logarithmic value*/

	(No column name)
1	101
	(No column name)
1	0.997494986604054
	(No column name)
1	15
	(No column name)
1	90.0171313005218
	(No column name)
1	14
	(No column name)
1	1.68639895357023

/*Rankig Functions*/

/* row()_number-giving consecutive numbers to rank*/

select * from employee;

select id,name,salary,ROW_NUMBER() over(order by salary desc) as rownumber
from employee;

select * from employee;

	id	name	salary	rownumber
1	105	nani	50000	1
2	107	nani	50000	2
3	106	stella	35000	3
4	103	sunny	35000	4
5	104	stella	35000	5
6	102	bob	30000	6
7	101	alex	20000	7

/*rank()-used to give rank if duplicates allowed ranking will be changed based on duplicates */

select id,name,salary,rank() over(order by salary) as rank
from employee;

	id	name	salary	rank
1	101	alex	20000	1
2	102	bob	30000	2
3	103	sunny	35000	3
4	104	stella	35000	3
5	106	stella	35000	3
6	107	nani	50000	6
7	105	nani	50000	6

/*dense_rank()-used to give ranks consecutively even if duplicates are allowed*/

select id,name,salary,dense_rank() over(order by salary desc) as rank
from employee;

	id	name	salary	rank
1	105	nani	50000	1
2	107	nani	50000	1
3	106	stella	35000	2
4	103	sunny	35000	2
5	104	stella	35000	2
6	102	bob	30000	3
7	101	alex	20000	4

/*ntile() function- it will divide give the rank in groups*/

```
select id,name,salary,ntile(2) over(order by salary) as rank  
from employee; /*without condition*/
```

	id	name	salary	rank
1	101	alex	20000	1
2	102	bob	30000	1
3	103	sunny	35000	1
4	104	stella	35000	1
5	106	stella	35000	2
6	107	nani	50000	2
7	105	nani	50000	2

```
select name,salary,ntile(4) over(order by salary) as rank  
from employee where salary>10000; /*with condition*/
```

	name	salary	rank
1	alex	20000	1
2	bob	30000	1
3	sunny	35000	2
4	stella	35000	2
5	stella	35000	3
6	nani	50000	3
7	nani	50000	4