

6. CEIL(num)

It returns the smallest integer greater than or equal to the number.

- Rounds num up to nearest integer.
- Works even if the number is already an integer(no change)

Ex. Ceil(12.3);

```
mysql> select ceil(12.3);
```

```
+-----+
| ceil(12.3) |
+-----+
|          13 |
+-----+
```

Select ceil(12.0);

```
mysql> Select ceil(12.0);
```

```
+-----+
| ceil(12.0) |
+-----+
|          12 |
+-----+
```

Ex. `select ceil(-12.3);`

```
+-----+
| ceil(-12.3) |
+-----+
|           -12 |
+-----+
```

7. FLOOR(num)

Returns the largest integer less than or equal to the number.

Ex. `mysql> Select floor(12.3);`

```
+-----+
| floor(12.3) |
+-----+
|           12 |
+-----+
```

Ex. mysql> select floor(-12.3);

```
+-----+
| floor(-12.3) |
+-----+
|           -13 |
+-----+
```

8. ASCII(str)

Returns the ascii value of the first character.

Ex. Select ascii('A');

mysql> Select ascii('A');

```
+-----+
| ascii('A') |
+-----+
|          65 |
+-----+
```

Ex. Select ASCII('Flynaut');

```
mysql> Select ASCII('Flynnaut');
```

```
+-----+
| ASCII('Flynnaut') |
+-----+
|                  70 |
+-----+
```

9. Substr(str, start, length)

Extract the substring starting at a specified position

Ex. SELECT substr('database' , 2, 4);

```
mysql> SELECT substr(' database' , 2, 4);
```

```
+-----+
| substr(' database' , 2, 4) |
+-----+
| atab                      |
+-----+
```

Ex. Select substr('Hello' , 1, 2); -> He

10. UPPER(str)

Convert a string into upper case

Ex. SELECT UPPER('FlyNaUt');

mysql> SELECT UPPER('FlyNaUt');

+	-----	+
	UPPER('FlyNaUt')	
+	-----	+
	FLYNAUT	
+	-----	+

11. LOWER(str)

Convert a string into lower case

Ex. SELECT LOWER('FlyNaUt');

mysql> SELECT LOWER('FlyNaUt');

+	-----	+
	LOWER('FlyNaUt')	
+	-----	+
	flynaut	
+	-----	+

12. INSTR(str, substr)

It returns the position of first occurrence of substr

Ex. Select instr('Database' , ' base');

mysql> Select instr('Database','base');

```
+-----+
| instr('Database','base') |
+-----+
|                        5 |
+-----+
```

13. TRIM(str)

It Removes all the leading and trailing spaces from a string.

Ex. Select trim(' Hello ');

mysql> Select trim(' Hello ');

```
+-----+
| trim('      Hello      ') |
+-----+
| Hello                      |
+-----+
```

14. lpad(str, length, padstr)

Left pads a string with specified characters to a given length.

Ex. Select lpad('Hello' ,8, ' *');

mysql> Select lpad('Hello',8,'*');

```
+-----+
| lpad('Hello',8,'*') |
+-----+
| ***Hello           |
+-----+
```

15. rpad(str, length, padstr)

Right pads a string with specified characters to a given length.

Select rpad('Hello' ,8, ' *');

mysql> Select rpad('Hello',8, '*');

```
+-----+
| rpad('Hello',8, '*') |
+-----+
| Hello***             |
+-----+
```

16. length(str)

Select length('flynaut');

mysql> Select length('flynaut');

```
+-----+
| length('flynaut') |
+-----+
|                    7 |
+-----+
```

17. Format(num, d)

Formats a number to d decimal places, rounds the number to the specified decimal places.

Assignment Solving

Write a query to find employees whose name contains the letter 'A' anywhere in the name, and order the result by their hire date.

```
SELECT ename, hiredate  
From emp  
WHERE ename like '%A%'  
Order by hiredate;
```

Write a query to find employees whose name starts with 'M' and whose job is 'SALESMAN'.

```
SELECT ename, job  
From emp  
Where ename like 'M%' AND job =  
'SALESMAN' ;
```

Write a query to find employees whose name ends with 'R', whose job is either 'SALESMAN' or 'MANAGER', and display their salaries.

```
Select ename, job, sal  
From emp  
WHERE ENAME like '%R'  
And  
Job in ( 'salesman' , ' manager' );
```

Write a query to find employees whose name has exactly 6 letters, and display their names, jobs, and salaries.

```
SELECT ename, job, sal
From emp
Where ename like '_____' ;
                (6 underscores)
```

Write a query to find employees whose name contains the letter 'S' in the second position and display their names, jobs, and hire dates.

```
SELECT ename, job, hiredate
From emp
Where ename like '_S%' ;
```

Write a query to find employees ordered by their hire date in ascending order, but if two employees have the same hire date, order by their salary in descending order.

```
SELECT * FROM EMP
order by hiredate asc, sal desc;
```

Write a query to find employees ordered by their commission in descending order, and then by their job.

```
SELECT * FROM EMP  
Order by comm desc, job;
```

Write a query to find employees in department 30, ordered by their job title in ascending order and then by salary in descending order.

```
Select * from emp  
Where deptno = 30  
Order by job asc, sal desc;
```

Write a query to find employees whose salary is between 1000 and 3000, ordered by their hire date and then by their job in descending order.

```
Select * from emp  
Where sal between 1000 and 3000  
Order by hiredate asc, job desc;
```

Write a query to find employees in departments 10 and 20, ordered by their hire date, and if the hire date is the same, order by their name in ascending order.

```
Select * from emp
Where deptno in (10,20)
Order by hiredate asc, ename asc;
```

Write a query to display the top 5 employees with the highest commission who work as a SALESMAN.

```
Select ename, comm
From emp
Where job = 'salesman'
Order by comm desc
Limit 5;
```

Write a query to display the top 3 employees with the earliest hire dates in department 30.

```
Select ename,hiredate
From emp
Where deptno = 30
Order by hiredate asc
Limit 3;
```

Write a query to find the top 5 employees who have the lowest salary and display their names, jobs, and salaries.

```
Select ename, job, sal  
From emp  
Order by sal asc  
Limit 5;
```

Write a query to display the first 3 employees in terms of salary from department 20, ordered by salary in ascending order.

```
Select ename, sal  
From emp  
Where deptno=20  
Order by sal asc  
Limit 3;
```

Write a query to display the bottom 3 employees (by salary) in departments 10 and 20, ordered by salary in descending order.

```
Select ename, sal  
From emp  
Where deptno in (10, 20)  
Order by sal desc  
Limit 3;
```

- HAVING CLAUSE

It is used to filter records based on aggregate functions.

Purpose:

- It filter a grouped data based on a condition
- While where clause filters the rows before grouping
- Having clause filters the group after aggregation
- It is used in conjunction with group by clause to filter grouped data
- Having clause is used after the group by clause
- We can use aggregate functions in having clause

Ex. Get the departments whose total salary is more than 9000;

```
Select deptno, sum(sal) As total_salary  
From emp  
Group by deptno
```

```
Having sum(sal) > 9000;
```

Explanation:

- The data is grouped by department
- The sum(sal) for each department is calculated
- Only groups where the total salary is more than 9000 will be in a result table.

EX> To get the departments with more than 5 employees

Ex> to get the departments where total salary is more than 9200 and maximum salary is less than 10000.