**Date Functions**

**current\_date()**

* It return current date of the system in MySQL date format.
* Example:

SELECT current\_date();

**current\_time()**

* It return current time of the system in MySQL date format.
* Example:

SELECT current\_time();

**current\_** **timestamp () / now()**

* It return current date and time of the system in MySQL date format.
* Example:

SELECT current\_ timestamp ();

**DATE\_ADD()**

* It is used add months/ years/days to the specified date.
* Example

SELECT current\_date(), date\_add(current\_date(), INTERVAL 1 YEAR);

SELECT current\_date(), date\_add(current\_date(), INTERVAL -1 YEAR);

SELECT current\_date(), date\_add(current\_date(), INTERVAL 1 MONTH);

SELECT current\_date(), date\_add(current\_date(), INTERVAL -1 MONTH);

SELECT current\_date(), date\_add(current\_date(), INTERVAL 1 DAY);

SELECT current\_date(), date\_add(current\_date(), INTERVAL -1 DAY);

Assignments

* Write a query to add 5 days into the given date ’14-02-2024’
* Write a query to display the employees who are joining in the month of Dec.
* Write a query to display the employees who are joining in the year of 2023.

Solutions

* SELECT date\_add('12-02-2024', INTERVAL 5 DAY); # Not Work
* SELECT date\_add('2024-02-12', INTERVAL 5 DAY); #implicit conversion
* SELECT date\_add(str\_to\_date('12-02-2024','%d-%m-%Y'), INTERVAL 5 DAY);
* SELECT \* FROM employee WHERE date\_format(hire\_date,'%m') = '12';
* SELECT \* FROM employee WHERE month(hire\_date) = '12';
* SELECT \* FROM employee WHERE date\_format(hire\_date,'%Y') = '2023';
* SELECT \* FROM employee WHERE year(hire\_date) = '2023';

**Inserting Date Values**

INSERT INTO employee value(6,'Tushar','IT',35000,'2024-02-20');

INSERT INTO employee value(7,'Tarun','IT',35000,str\_to\_date('2024-02-20','%Y-%m-%d'));

**Date Difference**

* DATEDIFF() functions calculates the number of days between two dates.
* Syntax:

DATEDIFF(end\_date, start\_date)

* Example:

SELECT DATEDIFF('2024-02-15', '2024-02-10');

SELECT hire\_date, DATEDIFF(current\_date(), hire\_date) from employee;

Assignment

* Write a query to display employees having more than 1 year of experience.
* Write a query to display employees who have completed 1-year service on 31 DEC 2022.
* Write a query to display employee who will complete 1 year of service on 31 DEC 2025.

Solutions

SELECT \* FROM employee WHERE DATEDIFF(CURRENT\_DATE(), hire\_date) > 365;

SELECT \* FROM employee WHERE DATEDIFF('2022-12-31', hire\_date) > 365;

SELECT hire\_date, DATEDIFF('2025-12-31', hire\_date) from employee;

**Group Functions**

* Group functions are operates over number of values in a table column and returns single values.
* Example
  + Max()
  + Min()
  + Avg()
  + Sum()
  + Count(\*)
  + Count(column name)

**Max()**

* It returns maximum value from a table column.
* Example:

SELECT MAX(salary) FROM employee;

SELECT MAX(hire\_date) FROM employee;

SELECT MAX(name) FROM employee;

**Min()**

* It returns minimum value from a table column.
* Example:

SELECT MIN(salary) FROM employee;

SELECT MIN(hire\_date) FROM employee;

SELECT MIN(name) FROM employee;

**Note**

* We are not allowed to use group function in where clause.
* Example

SELECT \* FROM employee WHERE salary=MIN(salary);

**Avg()**

* It returns average from number data type column.
* Example:

SELECT avg(salary) FROM employee;

**Sum()**

* It returns sum of number data type column.
* Example:

SELECT sum(salary) FROM employee;

**Count(\*)**

* It returns number of rows in a table.
* Example:

SELECT count(\*) FROM employee;

**Count(column name)**

* It counts number of not null values from a table column.
* Example:

SELECT count(name) FROM employee;

**Handling NULL Values**

ALTER TABLE employee ADD comm decimal(8,2);

SET SQL\_SAFE\_UPDATES=0;

UPDATE employee SET comm=100 WHERE employee\_id=2;

SELECT name, salary, comm, salary+comm as total\_salary FROM employee;

SELECT null+100;

**COALESCE Function**

* COALESCE() function is used to handle null values.
* It returns the first non-NULL value.
* Example:

SELECT coalesce(null,0);

SELECT coalesce(null,1);

SELECT coalesce(2,1,3);

Approach1:

SELECT name, salary, comm, salary+ coalesce(comm,0) as total\_salary FROM employee;

Approach2:

SELECT name, salary, comm, salary+ ifnull(comm,0) as total\_salary FROM employee;