Experiment No. 7

Aim: To implement the concept of SQL working with dates, including operations such as retrieving formatting & performing calculations on the date and time values using SQL queries.

Aim : To implement the concept of SQL working with dates including operations such as retrieving, formatting & performing calculations on date & time values using SQL queries.

Objective : To understand how to use data & time functions in SQL.

To retrieve & manipulate date-related data from a database.

To perform arithmetic operations with dates

To use built-in SQL functions for date formatting, difference & comparison.

Theory : Dates & times are crucial data types in DBMS. SQL provides various built in functions for handeling and manipulating date/time values. These functions allow formatting, extraction, difference calculations & conditional checks.

Common SQL Date functions:

i) CURRENT_DATE, CURRENT_TIME, NOW () → Retreive the current date/time.

ii) DATE (), TIME () → Extract date on time from datetime.

iii) DATEDIFF (date1, date2) → Returns the difference in dates.

iv) DATE_ADD (date, interval n UNIT) → Adds interval to the date.

v) DATE_SUB (date, INTERVAL n UNIT) → Subtracts interval from date.

vi) YEAR (), MONTH (), DAY () → Extract year, month or day.

vii) STR_TO_DATE () / DATE_FORMAT () → Convert / format date string.

Procedure :

i) Create a sample table with date & time feilds.

ii) Insert sample records including various records.

iii) Write & exeute SQL queries using different date functions.

iv) Observe & analyze the outputs.

Codes
Experimental :

i) What is the SQL query to display the current system date ?

ii) How do you retrieve the records of employees who joined after January 1, 2020 ?

iii) Write an SQL query to calculate the age of each employee using their date of birth DOB ?

Conclusion: The experiment successfully demonstrated how SQL handles date & time values. We practiced retrieving current dates, filtering data based on date conditions & using built in functions for calculation & formatting. These skills are essential in real-world DB systems.

Input

Run SQL

```sql
-- Table Creation
CREATE TABLE Employees (
    EmpID INTEGER PRIMARY KEY,
    Name TEXT,
    DateOfJoining DATE,
    DateOfBirth DATE
);

-- Data Insertion
INSERT INTO Employees VALUES (1, 'Alice', '2021-06-15', '1995-03-22');
INSERT INTO Employees VALUES (2, 'Bob', '2020-01-10', '1990-07-14');
INSERT INTO Employees VALUES (3, 'Charlie', '2019-11-05', '1988-12-01');

-- Query 1: Current Date
SELECT DATE('now') AS Current_Date;

-- Query 2: Employees joined after 2020
SELECT * FROM Employees
WHERE DateOfJoining > '2020-01-01';

-- Query 3: Calculate age (approximate, by subtracting years)
SELECT Name,
    CAST(strftime('%Y', 'now') AS INTEGER) - CAST(strftime('%Y', DateOfBirth) AS INTEGER) AS Age
FROM Employees;

-- Query 4: Days since joining
SELECT Name,
    ROUND(julianday('now') - julianday(DateOfJoining)) AS Days_Since_Joining
FROM Employees;

-- Query 5: Add 1 year to joining date
SELECT Name,
    DATE(DateOfJoining, '+1 year') AS Next_Anniversary
FROM Employees;
```

Output   **Available Tables**

Current_Date
2025-04-19

EmpID	Name	DateOfJoining	DateOfBirth
1	Alice	2021-06-15	1995-03-22
2	Bob	2020-01-10	1990-07-14

Name	Age
Alice	30
Bob	35
Charlie	37

Name	Days_Since_Joining
Alice	1405
Bob	1927
Charlie	1993

Name	Next_Anniversary
Alice	2022-06-15
Bob	2021-01-10
Charlie	2020-11-05

Name	Formatted_DoB

iv) How can you find the number of days each employee has been working since their joining date?

v) Write a query to add 1 year to the joining date of each employee?

vi) What is the SQL query to format the date of birth as "DD-MM-YYYY"?

Conclusion: The experiment was successfully demonstrated how SQL handles date & time values. We practiced retaining current dates, filtering data based on the date conditions & using built in functions for calculations & formatting. These skills are essential in real world DB systems.

(A+) Rudhula

24/4/25