Chapter: Mastering Dates and Times

Topic: Date and Time Functions in SQL

Section 1: Learn

1.1 Introduction to Date and Time in SQL

In SQL, handling **date and time** data accurately is crucial for real-world applications such as:

- Tracking orders and deliveries
- Calculating age or tenure
- Creating monthly or yearly reports
- Formatting timestamps for reports and dashboards

Most relational databases support date/time data types like:

- DATE – stores year, month, and day
- TIME – stores hour, minute, second
- DATETIME or TIMESTAMP – stores both date and time

SQL provides powerful **built-in functions** to manipulate these values for both retrieval and computation.

1.2 Getting the Current Date and Time

Functions:

- CURRENT_DATE – returns today's date.
- CURRENT_TIME – returns current time.
- NOW() – returns current date and time.

Example:

```
SELECT CURRENT_DATE, CURRENT_TIME, NOW();
```

Use Cases:

- Logging activities
- Tracking when a user was last active
- Inserting default date/time in tables

1.3 EXTRACT(): Extracting Parts of a Date

Purpose:

To isolate specific parts from a date such as year, month, day, hour, minute, etc.

Syntax:

```
EXTRACT(part FROM date_column)
```

Common Parts:

- YEAR, MONTH, DAY
- HOUR, MINUTE, SECOND (when using DATETIME)

Example:

```
SELECT name,
    EXTRACT(YEAR FROM join_date) AS join_year,
    EXTRACT(MONTH FROM join_date) AS join_month
FROM employees;
```

Use Cases:

- Analyzing trends by year/month
- Building time-based filters
- Creating derived columns for reports

1.4 DATE_ADD() and DATE_SUB(): Date Arithmetic

These functions let you **add or subtract time intervals** (days, months, years, etc.) from dates.

Syntax:

```
DATE_ADD(date, INTERVAL value unit)
DATE_SUB(date, INTERVAL value unit)
```

Examples:

```
-- Add 15 days to a hire date
SELECT DATE_ADD(hire_date, INTERVAL 15 DAY) AS probation_end
FROM employees;

-- Subtract 2 months from an order date
SELECT DATE_SUB(order_date, INTERVAL 2 MONTH) AS
estimated_production
FROM orders;
```

Use Cases:

- Estimate future or past events
- Automate renewal/expiry reminders

- Calculate deadlines or grace periods

1.5 DATEDIFF(): Finding the Number of Days Between Two Dates

Purpose:

To calculate the **duration** in days between two dates.

Syntax:

```
DATEDIFF(end_date, start_date)
```

Example:

```
SELECT student_id,
    DATEDIFF(check_out, check_in) AS stay_duration
FROM hostel_records;
```

> Returns the number of days the student stayed in the hostel.

Use Cases:

- Track customer retention
- Calculate service duration
- Determine number of days overdue

1.6 DATE_FORMAT(): Custom Formatting of Dates

Purpose:

To display a date in a **specific, human-readable format**.

Syntax (MySQL):

```
DATE_FORMAT(date_column, 'format_string')
```

Common Format Specifiers:

- %d – day (2 digits)
- %m – month (2 digits)
- %Y – year (4 digits)
- %H:%i:%s – time in HH:MM:SS

Example:

```
SELECT DATE_FORMAT(birth_date, '%d-%m-%Y') AS formatted_dob
FROM students;
```

> Outputs birth dates like 31-12-2000.

Use Cases:

- Create regional-friendly date displays
- Export date data in specific formats
- Format for front-end applications

1.7 Working with TIME Values

When dealing with time, you can:

- Extract hour, minute, second using HOUR(), MINUTE(), SECOND()
- Add/subtract time using ADDTIME() or SUBTIME()

Example:

```
SELECT HOUR(CURRENT_TIME) AS current_hour;
```

> Returns the hour part of the current time.

Section 2: Practise

Exercise 1: Display Today's Date and Time

```
SELECT CURRENT_DATE, CURRENT_TIME, NOW();
```

Exercise 2: Extract Year, Month, and Day

```
SELECT name,
    EXTRACT(YEAR FROM join_date) AS year_joined,
    EXTRACT(MONTH FROM join_date) AS month_joined,
    EXTRACT(DAY FROM join_date) AS day_joined
FROM employees;
```

Exercise 3: Calculate Warranty Expiry

Assume each product has a 2-year warranty.

```
SELECT product_id,
    DATE_ADD(purchase_date, INTERVAL 2 YEAR) AS warranty_expires
FROM products;
```

Exercise 4: Find Overdue Submissions

```
SELECT submission_id,
    DATEDIFF(due_date, submission_date) AS days_late
FROM submissions;
```

Exercise 5: Format Join Date

```
SELECT name,
    DATE_FORMAT(join_date, '%d-%M-%Y') AS formatted_join_date
FROM employees;
```

> This will show 12-April-2023

Exercise 6: Calculate Age

```
SELECT name,
    TIMESTAMPDIFF(YEAR, birth_date, CURDATE()) AS age
FROM students;
```

Section 3: FAQ – Know More

Q1. What is the difference between NOW() and CURRENT_TIMESTAMP?

Both return the **current date and time**.

In most systems like MySQL and PostgreSQL, they are equivalent.

Q2. Can we filter records using date ranges?

Yes, using BETWEEN or comparison operators:

```
SELECT * FROM orders
WHERE order_date BETWEEN '2024-01-01' AND '2024-03-31';
```

Q3. How to find the last day of a month?

Use LAST_DAY():

```
SELECT LAST_DAY('2024-02-15');  -- Output: 2024-02-29
```

Q4. How to find the weekday of a given date?

Use DAYNAME() or WEEKDAY():

```
SELECT DAYNAME('2024-04-11');  -- Output: Thursday
```

Q5. Is there a way to round dates to the nearest month or year?

Not directly, but you can use DATE_TRUNC() in PostgreSQL:

```
SELECT DATE_TRUNC('month', NOW());
```

> In MySQL, use string manipulation or custom formatting to achieve similar effects.

End of Notes for Chapter: Mastering Dates and Times