Lecture Notes – Introduction Date & Time Fuctions

Section 1: Lecture Summary

This lecture introduces **date and time functions** in SQL, which are essential tools for processing temporal data in complex queries. The content covers the categorization of date and time functions and explains where these functions can be applied within SQL statements.

Section 2: Key Concepts and Explanations

Date and time functions in SQL are organized into several categories:

Current Date and Time Functions retrieve the present date and time values from the system.

Universal Current Date and Time Functions provide date and time values in a standardized format.

Date and Time Extraction Functions allow you to extract specific components such as year, month, day, hour, minute, or second from a date or time value.

Arithmetic Operations on Dates enable you to add or subtract days, months, or years from a given date, useful for calculating future or past dates.

Time Difference Functions calculate the interval between two dates or times, providing results in various units.

Conversion Functions transform dates into strings or strings into dates, facilitating data type conversions for storage and display.

Applicable SQL Clauses Date and time functions can be used across multiple SQL clauses including SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY, providing flexibility in query construction.

Section 3: Example Code and Use Cases

Using the eCommerceDB schema, here are practical applications of date and time functions:

Extracting Order Information by Date

```
SELECT OrderID, CustomerID, OrderDate
FROM Orders
WHERE OrderDate >= '2025-01-01'
ORDER BY OrderDate DESC;
```

This query retrieves orders placed after a specific date, demonstrating date filtering in the WHERE clause.

Calculating Days Since Customer Joined

```
SELECT CustomerID, FirstName, JoinDate,
       DATEDIFF(CURDATE(), JoinDate) AS DaysSinceJoined
FROM Customers
ORDER BY DaysSinceJoined DESC;
```

This example uses a date arithmetic function to calculate the number of days since each customer joined.

Grouping Orders by Month

```
SELECT YEAR(OrderDate) AS Year,
       MONTH(OrderDate) AS Month,
       COUNT(*) AS TotalOrders
FROM Orders
GROUP BY YEAR(OrderDate), MONTH(OrderDate)
ORDER BY Year, Month;
```

This demonstrates extraction functions within the GROUP BY clause to analyze order trends over time.

Section 4: Key Takeaways

Date and time functions are powerful tools for temporal data manipulation in SQL. Understanding their categories helps you select the appropriate function for your specific use case. These functions work seamlessly across all major SQL clauses, allowing you to filter, group, sort, and transform temporal data effectively in your queries.