Lecture Notes – Date & Time Conversion

Section 1: Lecture Summary

The lecture covers MySQL datetime conversion functions: **STR_TO_DATE** for converting strings to date objects using specified formats, **MAKEDATE** for creating dates from year and day-of-year numbers, **MAKETIME** for building time objects from hour, minute, and second values, and **FROM_DAYS** for converting total days (from AD start) into dates, typically used relatively with functions like **TO_DAYS** on current dates plus intervals.

Section 2: Key Concepts and Explanations

STR_TO_DATE takes a string and format specifier (e.g., %d for day, %m for month, %Y for four-digit year) to parse into a standard SQL date object shown as YYYY-MM-DD. Formats match those from prior extraction functions like DAYNAME or MONTHNAME.

MAKEDATE requires year and day-of-year (1-366); it outputs the corresponding date, e.g., MAKEDATE(2025, 100) gives the 100th day of 2025.

MAKETIME directly constructs time from hour (0-23), minute (0-59), and second (0-59), producing a database time object like 'HH:MM:SS'.

FROM_DAYS converts a total day count from the AD epoch (year 0000-00-00) to a date, accounting for Julian-to-Gregorian calendar shifts, so absolute large values like 2000*365 yield unexpected years (e.g., 1998); use relatively by converting current date with **TO_DAYS** (e.g., TO_DAYS(CURDATE() + INTERVAL 30 DAY)) then applying **FROM_DAYS** to add/subtract periods.

Section 3: Example Code and Use Cases

Using **companyDB** for hire date conversions:

```
USE companyDB;

-- STR_TO_DATE: Convert "01/12/2025" string to date
```

```
SELECT STR_TO_DATE('01/12/2025', '%d/%m/%Y') AS converted_date;
-- Result: 2025-12-01

-- MAKEDATE: 100th day of 2025 for employee hires
SELECT MAKEDATE(2025, 100) AS hire_date_example;
-- Result: 2025-04-10

-- MAKETIME: Example shift time
SELECT MAKETIME(14, 30, 0) AS shift_time;
-- Result: 14:30:00

-- FROM_DAYS with relative calculation: 30 days after an employee's
HireDate
SELECT
    EmpID, FirstName, HireDate,
    FROM_DAYS(TO_DAYS(HireDate + INTERVAL 30 DAY)) AS thirty_days_later
FROM Employees
WHERE DeptID = 10;
-- Adds 30 days to each HireDate and converts back to date
```

Using **eCommerceDB** for order dates:

```
USE eCommerceDB;

-- STR_TO_DATE on JoinDate string simulation
SELECT STR_TO_DATE('15/02/2025', '%d/%m/%Y') AS parsed_join_date;

-- MAKEDATE for projected stock review date
SELECT MAKEDATE(2026, 56) AS review_date;

-- FROM_DAYS: 30 days after OrderDate
SELECT
    OrderID, CustomerID, OrderDate,
    FROM_DAYS(TO_DAYS(OrderDate + INTERVAL 30 DAY)) AS delivery_estimate
FROM Orders
WHERE OrderStatus = 'Pending';
```

Section 4: Key Takeaways

STR_TO_DATE enables date creation from formatted strings; specify order with %d/%m/%Y etc. **MAKEDATE** and **MAKETIME** build dates/times from numeric

parts. **FROM_DAYS**/**TO_DAYS** pair for relative day arithmetic from AD epoch, avoiding calendar discrepancies with direct large inputs; prefer INTERVAL for simple additions. Practice these for string/date conversions in queries.