Lecture Notes – Update Table

Section 1: Lecture Summary

The lecture covers the **UPDATE** statement, a core Data Manipulation Language (DML) command used to modify existing data in tables. The instructor demonstrates the syntax, structure, and practical application of UPDATE with multiple examples using the Employees table from the companyDB database. Emphasis is placed on the critical importance of the WHERE clause to ensure modifications affect only the intended rows.

Section 2: Key Concepts and Explanations

UPDATE Statement Syntax

The basic structure for updating table data is:

UPDATE table_name SET column_name = new_value WHERE condition;

For multiple column updates:

UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

Core Components:

- **UPDATE**: Keyword indicating a modification operation
- **table_name**: The table containing data to modify
- **SET**: Keyword specifying which columns and values to change

- **WHERE**: Condition that identifies which rows to update (critical to avoid unintended modifications)

The WHERE Clause: Essential for Precision

The WHERE clause is fundamental to safe UPDATE operations. Without it or with an improperly constructed condition, the UPDATE statement will modify all rows in the table. Always identify the exact row(s) to modify, typically using the **primary key** to ensure single-row precision.

Section 3: Example Code and Use Cases

Example 1: Updating Multiple Columns for a Single Employee

Update the salary and department ID for employee ID 4 (both previously null):

```
UPDATE Employees
SET DeptID = 3, Salary = 70000
WHERE EmpID = 4;
```

This modifies only the row where EmpID equals 4, setting DeptID to 3 and Salary to 70000.

Example 2: Updating a Single Column

Update the department ID for employee ID 2:

```
UPDATE Employees
SET DeptID = 2
WHERE EmpID = 2;
```

Example 3: Updating a Boolean Status Field

Change the active status for employee ID 3 from true (1) to false (0):

```
UPDATE Employees
SET IsActive = 0
WHERE EmpID = 3;
```

Example 4: Updating Multiple Employees (Multi-Row Update)

Update the salary for all employees in department 1:

```
UPDATE Employees
SET Salary = 65000
WHERE DeptID = 1;
```

Note: This condition matches multiple rows. Use this pattern only when intentionally modifying multiple records.

Example 5: Updating with Calculated Values

Increase the salary of all employees in the Sales department by 10%:

```
UPDATE Employees
SET Salary = Salary * 1.10
WHERE DeptID = (SELECT DeptID FROM Departments WHERE DeptName = 'Sales');
```

Section 4: Key Takeaways

- The UPDATE statement modifies existing data in specific columns and rows

- Always use a WHERE clause with proper conditions to target exact rows

- Use the **primary key** (like EmpID) for precise single-row updates

- Multiple columns can be updated in a single statement by separating assignments with commas

- Careless UPDATE statements without proper WHERE conditions risk losing data

- Practice UPDATE operations on test databases rather than production data

- Subsequent DML operations include DELETE for removing records