

## Lecture Notes – Insert Table

### Section 1: Lecture Summary

DML (Data Manipulation Language) covers commands to manipulate data in tables: **insert** for adding rows, **delete** for removing rows, and **update** for modifying rows. The focus is on the **insert** command with syntax options for all columns, specific columns, and multiple rows at once.

### Section 2: Key Concepts and Explanations

**Insert** adds new rows to a table. Basic syntax for all columns: `INSERT INTO table_name VALUES (value1, value2, ...);`. Values must match column order and data types: strings and dates in single quotes, numerics without quotes. For specific columns: `INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);`. Column order must match values provided; omitted columns get **NULL**. Multiple rows use comma-separated value sets: `INSERT INTO table_name (columns...) VALUES (row1), (row2), ...;`. Foreign key values like **DeptID** must reference valid existing IDs.

### Section 3: Example Code and Use Cases

Using **companyDB** **Employees** table (EmpID int, FirstName varchar, LastName varchar, JobTitle varchar, DeptID int, HireDate date, Salary decimal):

Insert one row into all columns:

```
INSERT INTO Employees VALUES (1, 'John', 'Doe', 'Manager', 1, '2023-01-15', 50000);
```

Insert into specific columns (skipping DeptID, results in NULL):

```
INSERT INTO Employees (EmpID, FirstName, LastName, JobTitle, HireDate, Salary)
VALUES (2, 'Jane', 'Smith', 'Analyst', '2023-02-20', 45000);
```

Insert multiple rows:

```
INSERT INTO Employees (EmpID, FirstName, LastName, JobTitle, HireDate,
Salary, DeptID)
VALUES
(3, 'Bob', 'Johnson', 'Developer', '2023-03-10', 60000, 1),
(4, 'Alice', 'Brown', 'Tester', '2023-04-05', 40000, NULL);
```

Verify with ``SELECT * FROM Employees;``.

#### Section 4: Key Takeaways

Match values to column order and types. Specify columns for selective inserts or reordering. Insert multiple rows in one statement with commas. Omitted columns default to `**NULL**`. Use valid `**DeptID**` from `**Departments**` for referential integrity.