Lecture Notes – 2 CreatingDatabase-DDL

Section 1: Lecture Summary

The lecture covers creating tables in the companyDB database using **DDL** (Data Definition Language) with the **CREATE TABLE** statement. It explains the script used to build tables like Departments, Employees, Projects, and EmployeeProjects, including column definitions, data types, **primary keys**, **foreign keys**, and **composite primary keys**.

Section 2: Key Concepts and Explanations

CREATE TABLE syntax starts with **CREATE TABLE** followed by the table name and brackets containing column definitions. Each column requires a name and **data type**, such as **int** for numeric values without decimals, **decimal** for values with decimals (e.g., decimal(12,2) for 10 digits before and 2 after the decimal point), **varchar(50)** for variable-length strings up to 50 characters, and **date** for dates.

Primary key is defined on a column like **DeptID int PRIMARY KEY** to uniquely identify rows. **Foreign key** links tables, specified at the end as **FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)**, referencing the primary key in another table.

Composite primary key combines multiple columns, as in **PRIMARY KEY (EmpID, ProjectID)** for the EmployeeProjects table handling many-to-many relationships. Both EmpID and ProjectID in EmployeeProjects are also foreign keys referencing Employees and Projects.

Section 3: Example Code and Use Cases

Departments table:

```
CREATE TABLE Departments (
  DeptID int PRIMARY KEY,
  DeptName varchar(50),
  Location varchar(50)
);
```

Employees table:

```
CREATE TABLE Employees (
  EmpID int PRIMARY KEY,
  FirstName varchar(50),
  LastName varchar(50),
  JobTitle varchar(50),
  DeptID int,
  HireDate date,
  Salary decimal(12,2),
  FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
);
```

Projects table:

```
CREATE TABLE Projects (
  ProjectID int PRIMARY KEY,
  ProjectName varchar(50),
  DeptID int,
  FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
);
```

EmployeeProjects table (many-to-many junction):

```
CREATE TABLE EmployeeProjects (
  EmpID int,
  ProjectID int,
  PRIMARY KEY (EmpID, ProjectID),
  FOREIGN KEY (EmpID) REFERENCES Employees(EmpID),
  FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID)
);
```

Section 4: Key Takeaways

Use **CREATE TABLE** with columns, data types, and constraints like **PRIMARY KEY** and **FOREIGN KEY**. Define foreign keys at the end referencing primary keys in other tables. Composite primary keys handle many-to-many relationships in junction tables like EmployeeProjects. Common data types: **int**, **decimal**, **varchar(n)**, **date**.