Lecture Notes – Aliasing Columns

Section 1: Lecture Summary

The lecture covers **aliasing columns** in SQL SELECT statements, explaining how to assign new names or nicknames to columns in query results. It demonstrates using aliases for single columns, calculated expressions like annual salary, and concatenated columns for full names, emphasizing meaningful column headers in output.

Section 2: Key Concepts and Explanations

Aliasing renames columns in the SELECT clause for clearer, shorter, or more descriptive output headers. Syntax uses **AS** keyword after the column or expression: `column_name AS alias_name`, where alias can be quoted or unquoted. Aliases are useful for shortening long column names (e.g., **FirstName** to **FName**), labeling calculated fields (e.g., **Salary * 12 AS AnnualSalary**), or naming results of expressions like concatenation (e.g., **CONCAT(FirstName, ' ', LastName) AS FullName**). The original table column name does not change; only the result set header does.

Section 3: Example Code and Use Cases

```
SELECT FirstName AS FName, LastName AS LName
FROM Employees;
```

Displays employee first and last names with shortened column headers **FName** and **LName**.

```
SELECT Salary * 12 AS AnnualSalary
FROM Employees;
```

Calculates yearly salary from the **Salary** column (monthly) and labels the result **AnnualSalary** for clarity.

```
SELECT CONCAT(FirstName, ' ', LastName) AS FullName
FROM Employees;
```

Joins **FirstName** and **LastName** with a space, displaying as a single **FullName** column.

Section 4: Key Takeaways

- Use **AS alias_name** in SELECT for readable result headers on columns, formulas, or functions.

- Ideal for calculations (e.g., annual salary) or concatenations where default names are unclear.

- Alias applies only to query output, not the underlying table structure.

- Practice on **Employees** table for single columns, expressions, and combinations.