Lecture Notes – Range Queries

Section 1: Lecture Summary

Range queries use the **BETWEEN** operator in the **WHERE** clause to filter rows where a column value falls within a specified range, inclusive of both endpoints. The syntax is `column_name BETWEEN value1 AND value2`, applied to numeric columns like salary or ID for conditional selection. Examples demonstrate selecting employees by salary range and projects by ID range from the companyDB, with optional sorting using **ORDER BY**.

Section 2: Key Concepts and Explanations

BETWEEN operator checks if a value is greater than or equal to the first value and less than or equal to the second value in a range. It includes the boundary values, so `Salary BETWEEN 60000 AND 90000` returns rows where Salary >= 60000 and Salary <= 90000. This works on columns like **Salary** (decimal) or **ProjectID** (int) from the provided schemas. Multiple columns can be selected explicitly or using `*` for all columns. **ORDER BY** can sort results, defaulting to ascending order.

Section 3: Example Code and Use Cases

Display employees whose salaries are between 60000 and 90000:

```
SELECT EmpID, FirstName, Salary
FROM Employees
WHERE Salary BETWEEN 60000 AND 90000;
```

This returns employees with Salary in the inclusive range from the **Employees** table.

Show projects with IDs between 202 and 208, sorted by project name:

```
SELECT *
FROM Projects
WHERE ProjectID BETWEEN 202 AND 208
ORDER BY ProjectName;
```

This retrieves all columns from **Projects** where **ProjectID** is within 202 to 208 inclusive, sorted ascending by **ProjectName**.

Section 4: Key Takeaways

BETWEEN simplifies range conditions in **WHERE** clauses for inclusive filtering. Specify exact column names, start value, and end value. Combine with **SELECT** specific columns or `*`, and use **ORDER BY** for sorted output. Applies directly to numeric fields like **Salary** and **ProjectID** in companyDB.