

Lecture Notes – Group By & Having

Section 1: Lecture Summary

Group by groups data based on column values, allowing aggregate functions like count, sum, average, max to be applied to each group individually. Having filters groups based on conditions using aggregate functions, such as count greater than five.

Section 2: Key Concepts and Explanations

Group by divides rows into groups by unique values in specified columns; aggregate functions then compute results per group. Columns in **group by** must appear in the **select** clause alongside aggregates. **Having** applies conditions to grouped results, typically with aggregates, to filter groups after grouping occurs. Without grouping columns in **select**, results lack context on which group each aggregate belongs to. Aggregates previously applied to entire tables; **group by** enables per-group aggregation.

Section 3: Example Code and Use Cases

Count employees per department:

```
SELECT DeptID, COUNT(*)
FROM Employees
GROUP BY DeptID;
```

This groups Employees by **DeptID**, counts per group (e.g., DeptID 1: 4, DeptID 2: 6, DeptID 3: 5, DeptID 4: 5).

Filter groups with more than 5 employees:

```
SELECT DeptID, COUNT(*)
FROM Employees
GROUP BY DeptID
HAVING COUNT(*) > 5;
```

This returns only DeptID 2 with count 6.

Section 4: Key Takeaways

Use `group by` for per-group aggregates on columns like `DeptID`; include grouping columns in `select`. Apply `having` with aggregates to filter groups. Practice combines with count, sum, average, min, max for summarizing data.