Database Fundamentals

Cybersecurity Department

Course Code: ***CBS 213***

Practical Lecture 6: Advanced SQL – Aggregation & Grouping

Halal Abdulrahman Ahmed

Outlines

- Creating a sample database for aggregation
- Aggregate functions: COUNT, SUM, AVG, MIN, MAX
- Using DISTINCT with aggregates
- GROUP BY for summarizing data
- HAVING for filtering grouped results
- ORDER BY with aggregates

Learning Outcomes

By the end of this lecture, students will be able to:

- Use SQL aggregate functions to summarize large datasets
- Apply COUNT(), SUM(), AVG(), MIN(), MAX()
- Group rows using GROUP BY
- Filter grouped results using HAVING
- Combine GROUP BY with ORDER BY for reports
- Analyze university-style datasets effectively

What Are Aggregate Functions?

Aggregate functions summarize large sets of data into meaningful results. They:

- Perform calculations on **multiple rows** and returns a **single value**
- Ignore **NULL values** (except COUNT(*))
- Commonly used with **GROUP BY**, **HAVING**, **ORDER BY**

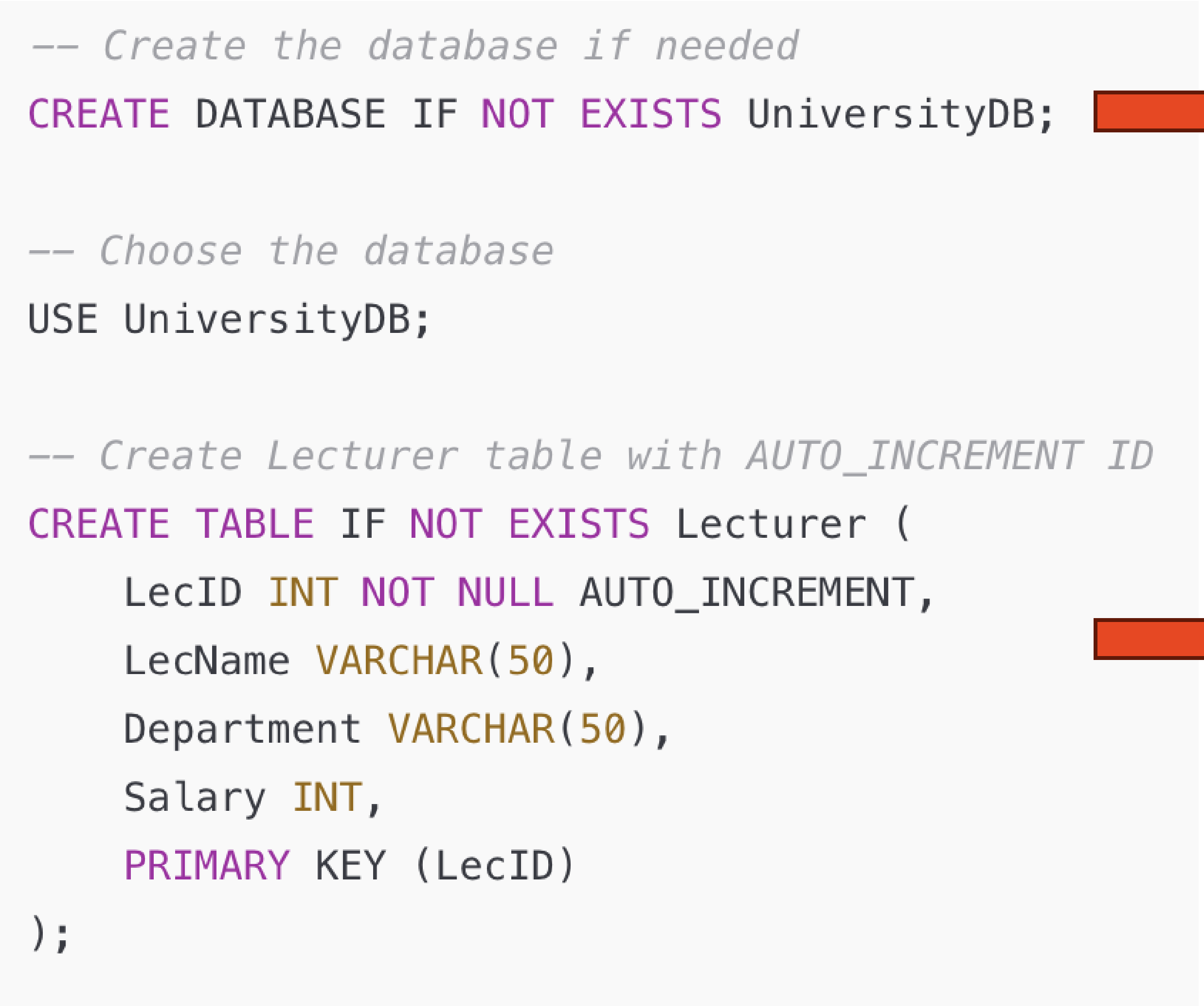
Examples of operations:

- Total salary
- Average grades
- Number of students
- Minimum / Maximum values

Common aggregate functions:

- COUNT()
- SUM()
- AVG()
- MIN()
- MAX()

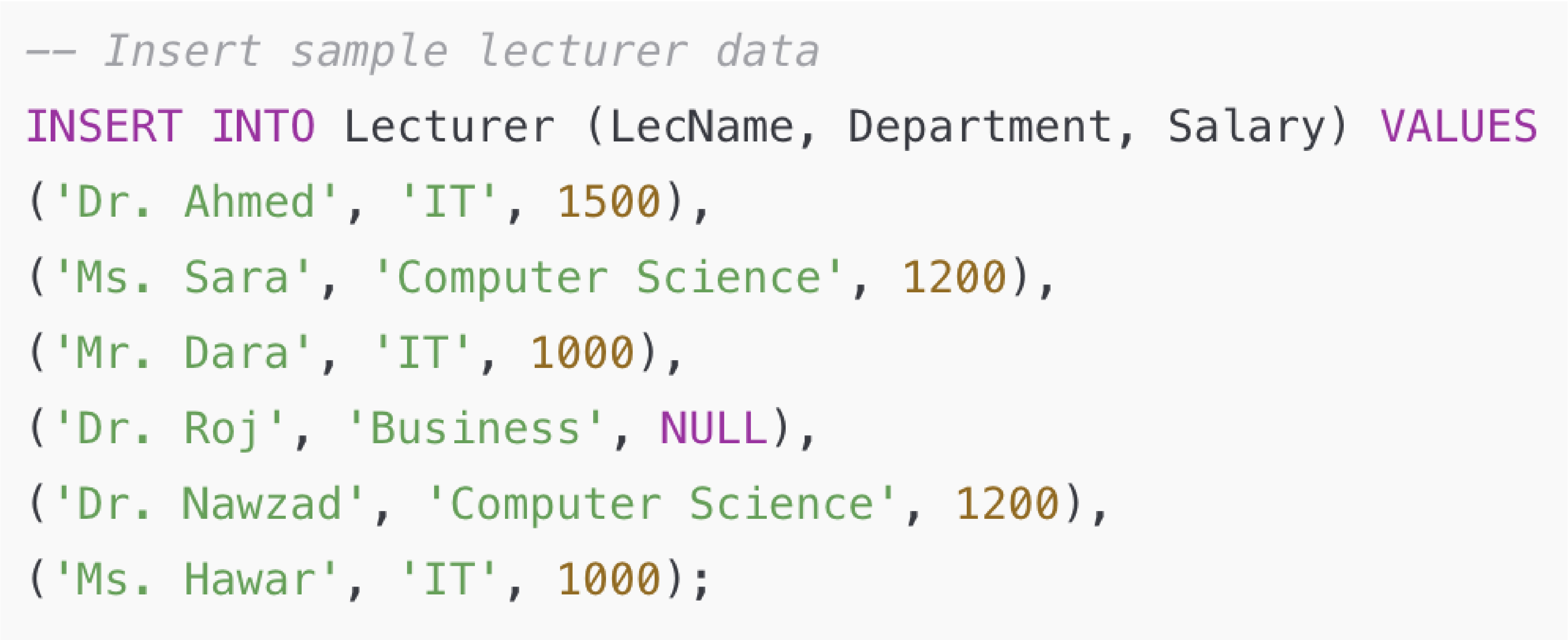
Creating a Database (University Example)



```sql
-- Create the database if needed
CREATE DATABASE IF NOT EXISTS UniversityDB;

-- Choose the database
USE UniversityDB;

-- Create Lecturer table with AUTO_INCREMENT ID
CREATE TABLE IF NOT EXISTS Lecturer (
    LecID INT NOT NULL AUTO_INCREMENT,
    LecName VARCHAR(50),
    Department VARCHAR(50),
    Salary INT,
    PRIMARY KEY (LecID)
);
```

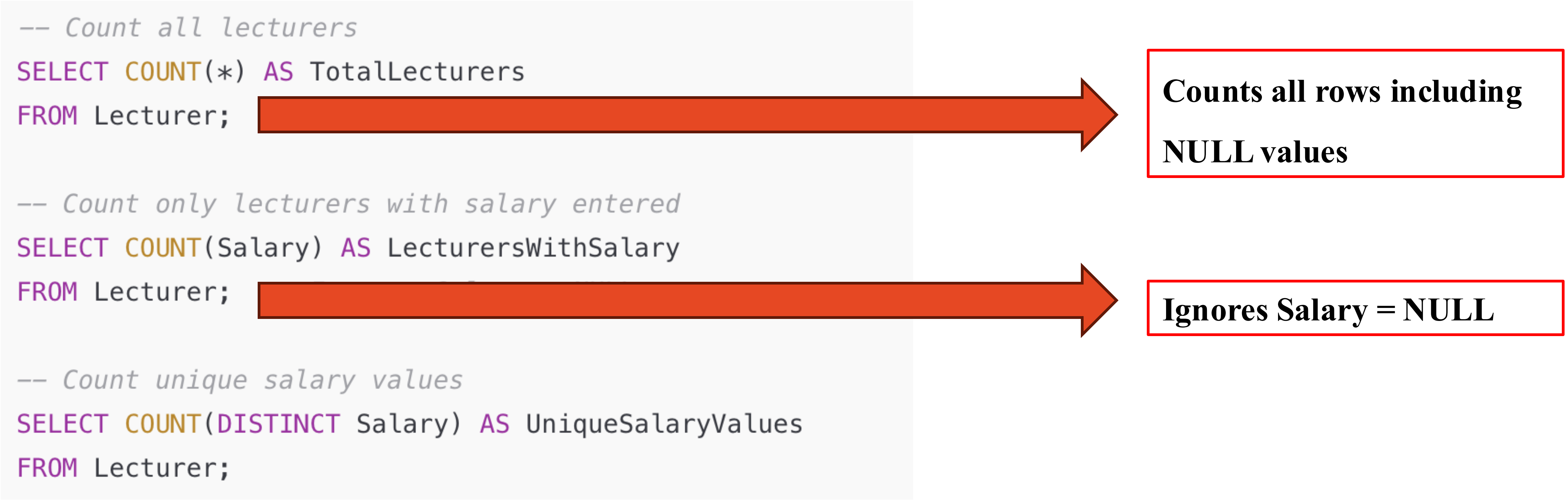
Create database

Create Lecturer table



```sql
-- Insert sample lecturer data
INSERT INTO Lecturer (LecName, Department, Salary) VALUES
('Dr. Ahmed', 'IT', 1500),
('Ms. Sara', 'Computer Science', 1200),
('Mr. Dara', 'IT', 1000),
('Dr. Roj', 'Business', NULL),
('Dr. Nawzad', 'Computer Science', 1200),
('Ms. Hawar', 'IT', 1000);
```

COUNT() - count rows

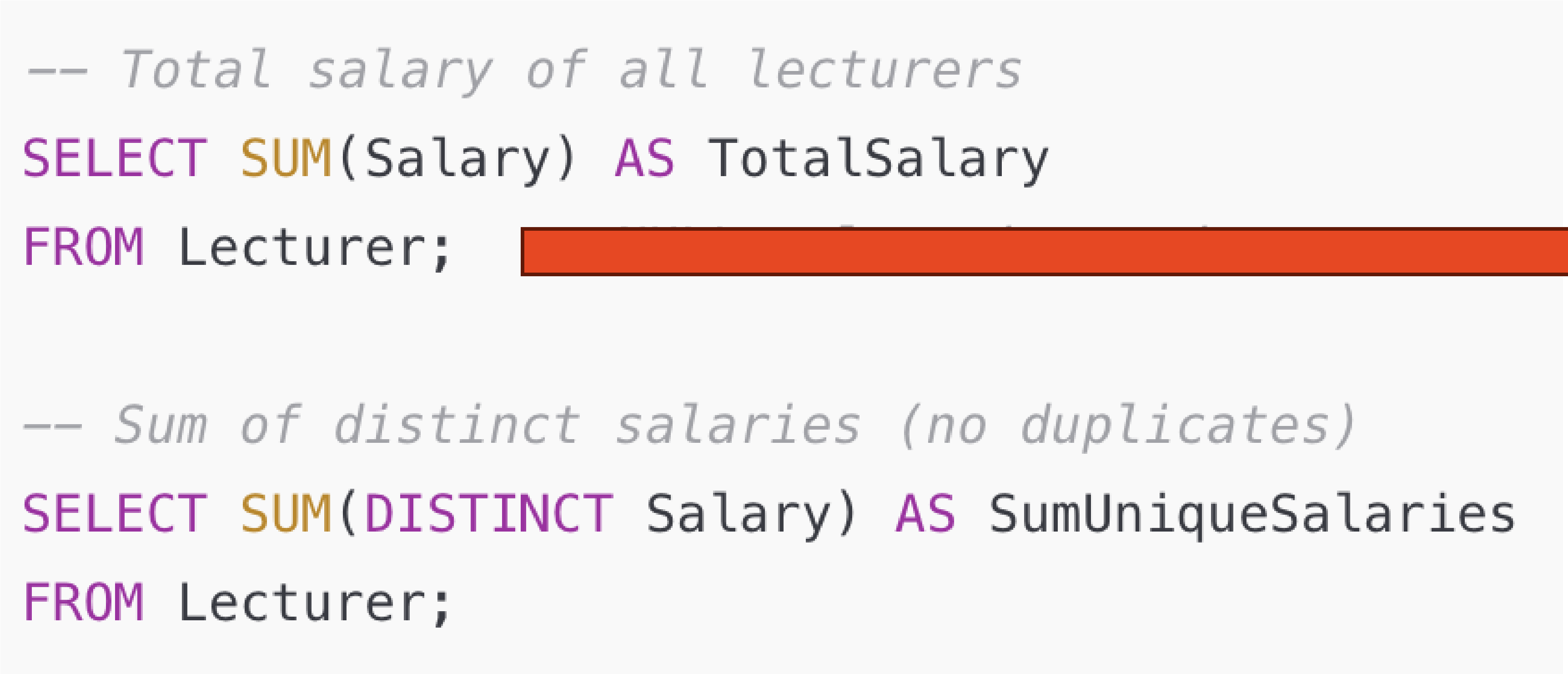


```
-- Count all lecturers
SELECT COUNT(*) AS TotalLecturers
FROM Lecturer;

-- Count only lecturers with salary entered
SELECT COUNT(Salary) AS LecturersWithSalary
FROM Lecturer;

-- Count unique salary values
SELECT COUNT(DISTINCT Salary) AS UniqueSalaryValues
FROM Lecturer;
```

Counts all rows including NULL values

Ignores Salary = NULL

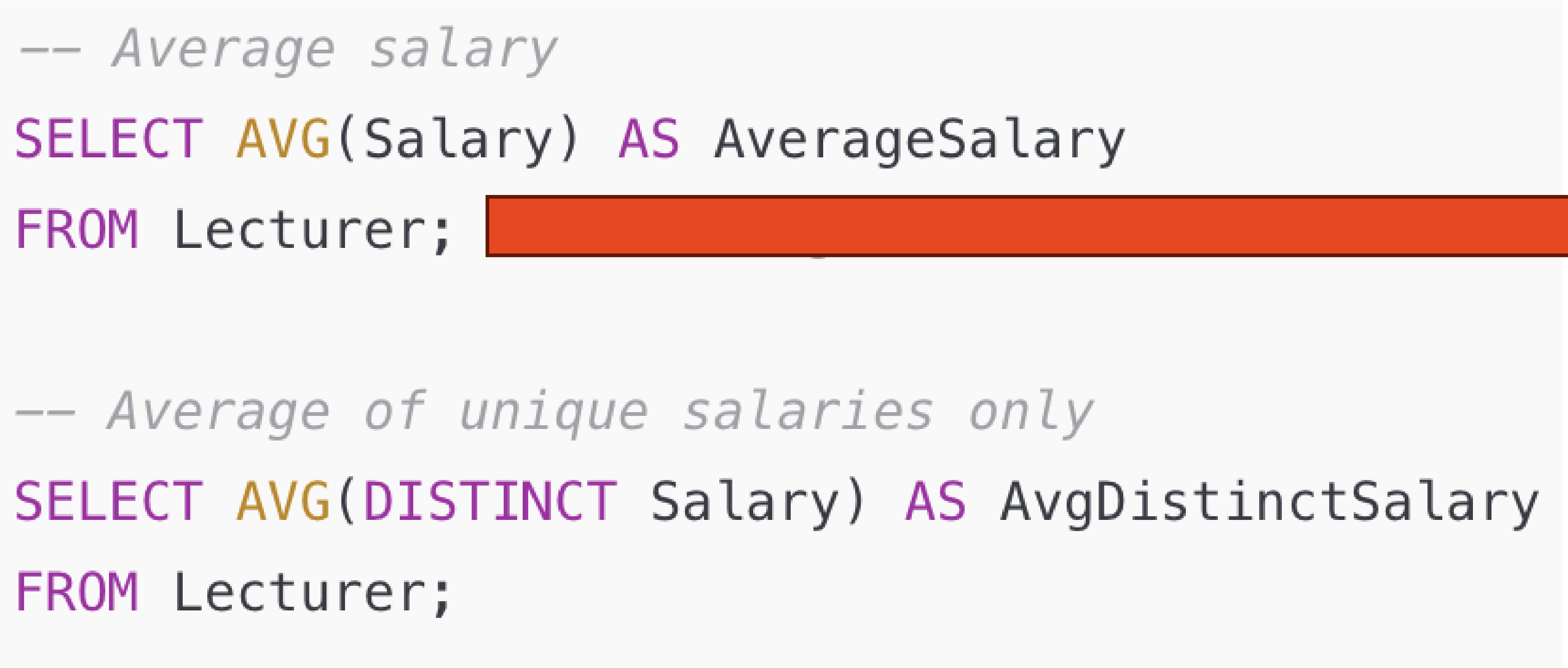
SUM() - total of numbers



```
-- Total salary of all lecturers
SELECT SUM(Salary) AS TotalSalary
FROM Lecturer;

-- Sum of distinct salaries (no duplicates)
SELECT SUM(DISTINCT Salary) AS SumUniqueSalaries
FROM Lecturer;
```

NULL values ignored

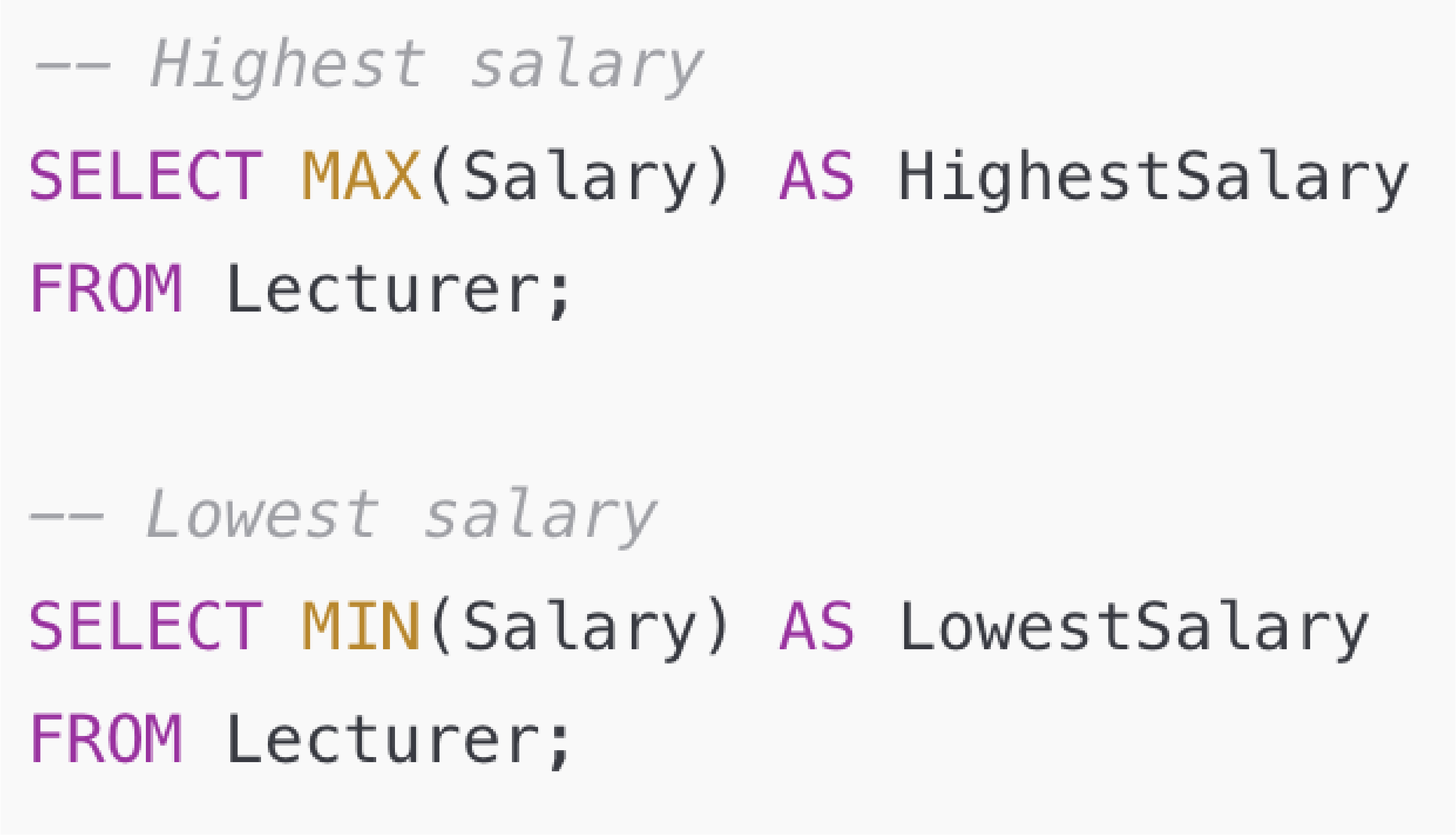
AVG() - average of numbers



```sql
-- Average salary
SELECT AVG(Salary) AS AverageSalary
FROM Lecturer;

-- Average of unique salaries only
SELECT AVG(DISTINCT Salary) AS AvgDistinctSalary
FROM Lecturer;
```

NULL values ignored

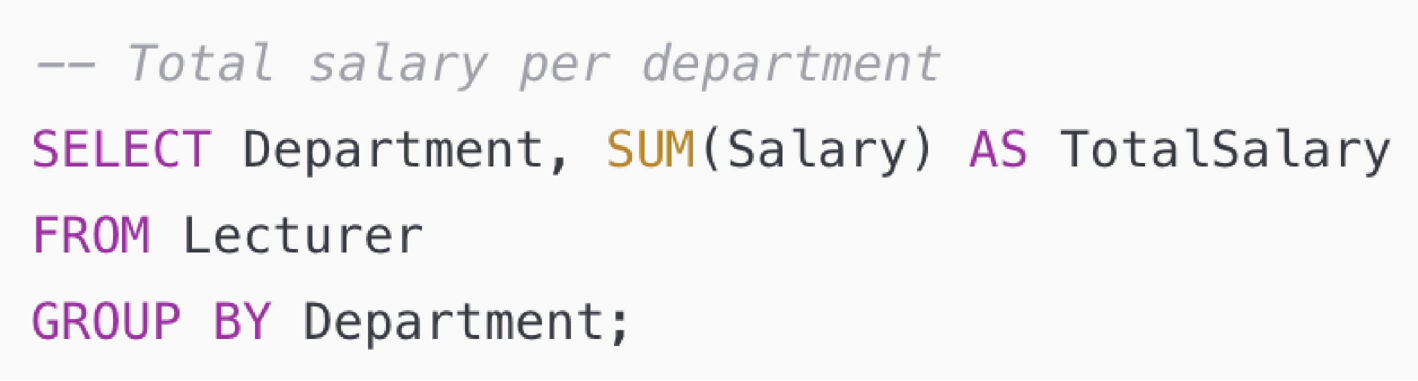
MIN() & MAX()



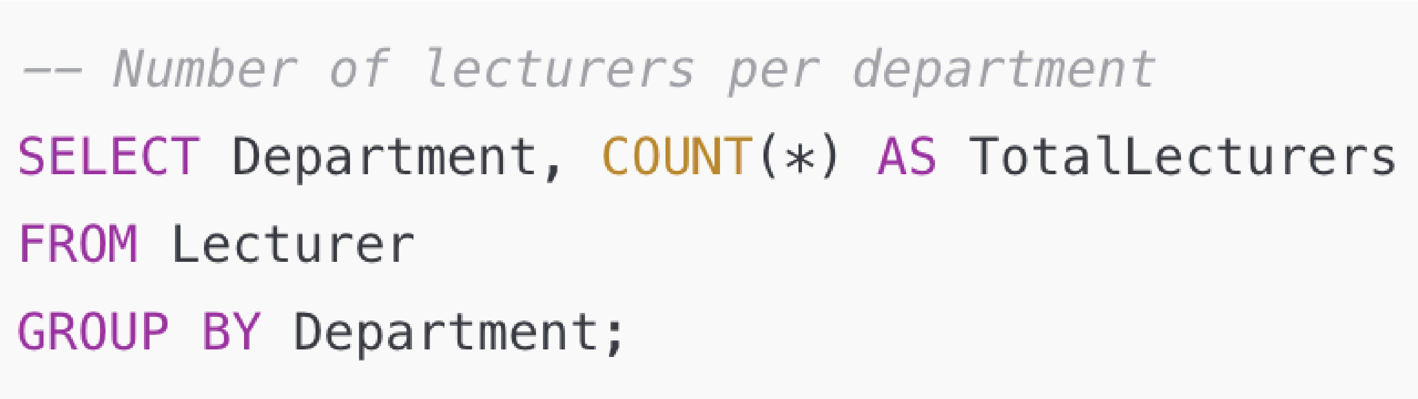
```sql
-- Highest salary
SELECT MAX(Salary) AS HighestSalary
FROM Lecturer;

-- Lowest salary
SELECT MIN(Salary) AS LowestSalary
FROM Lecturer;
```

- **MIN() - smallest value**
- **MAX() - largest value**

GROUP BY



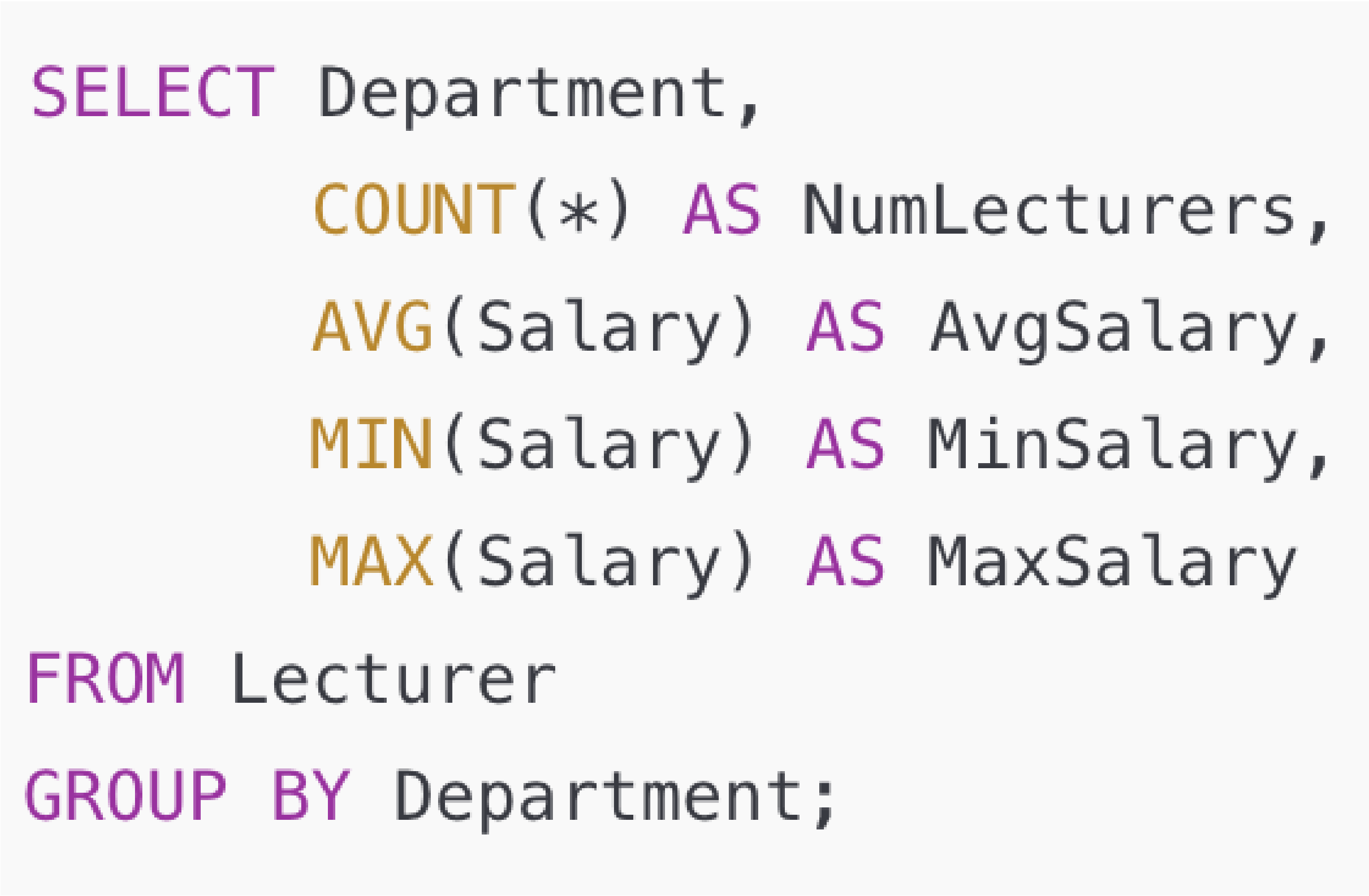
```sql
-- Total salary per department
SELECT Department, SUM(Salary) AS TotalSalary
FROM Lecturer
GROUP BY Department;
```



```sql
-- Number of lecturers per department
SELECT Department, COUNT(*) AS TotalLecturers
FROM Lecturer
GROUP BY Department;
```

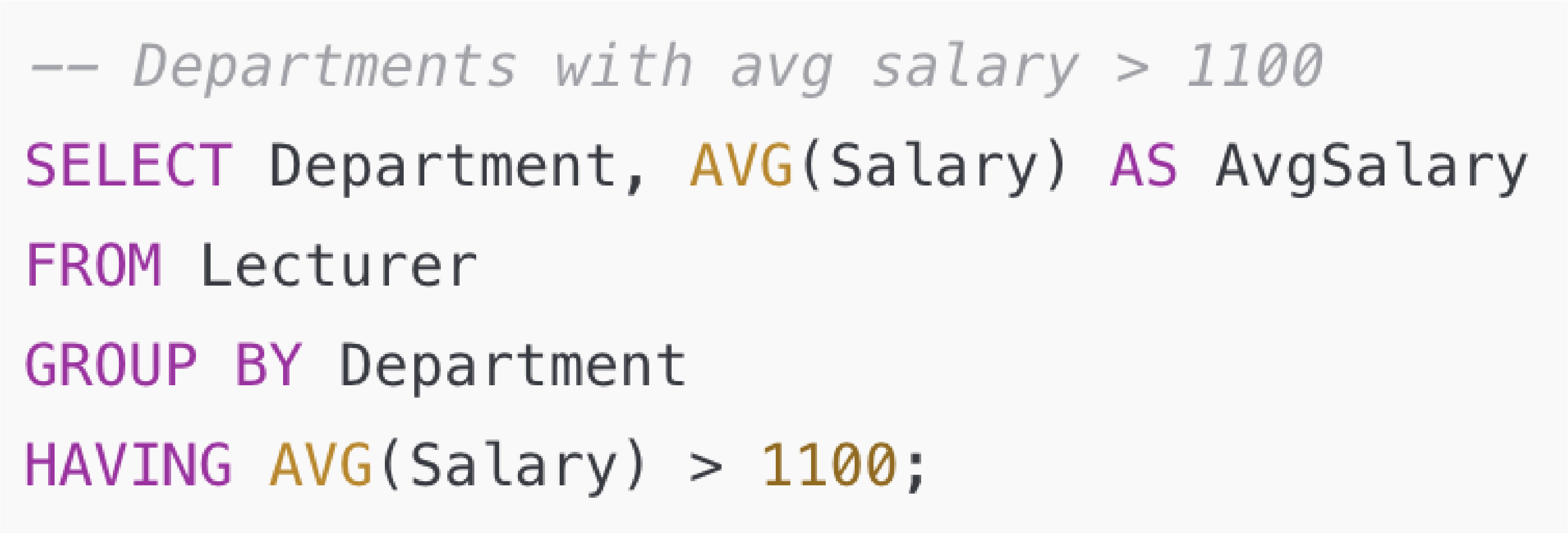
Group rows with the same department

GROUP BY with Multiple Functions



```sql
SELECT Department,
       COUNT(*) AS NumLecturers,
       AVG(Salary) AS AvgSalary,
       MIN(Salary) AS MinSalary,
       MAX(Salary) AS MaxSalary
FROM Lecturer
GROUP BY Department;
```

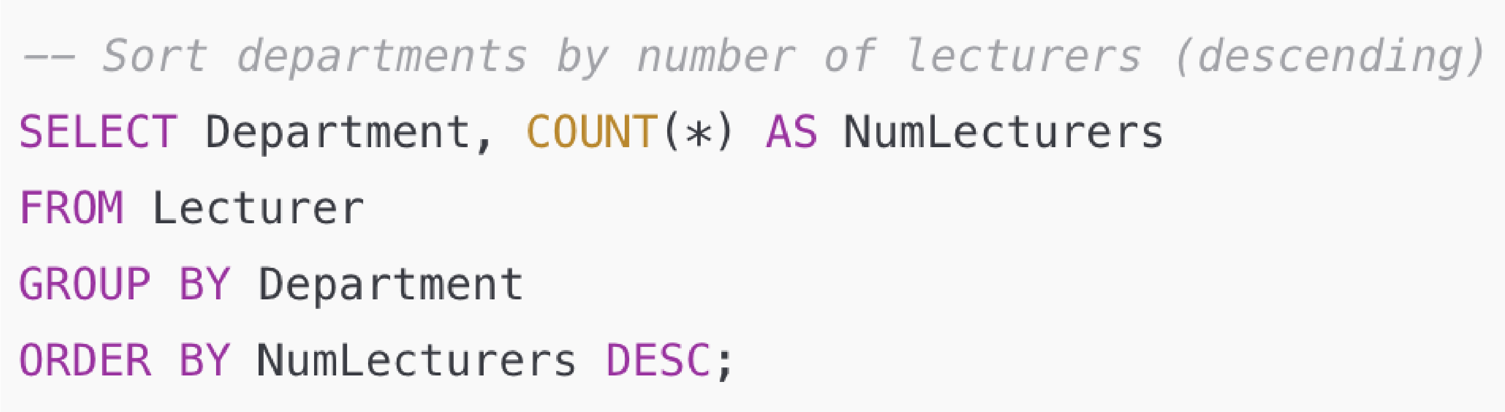
HAVING Clause



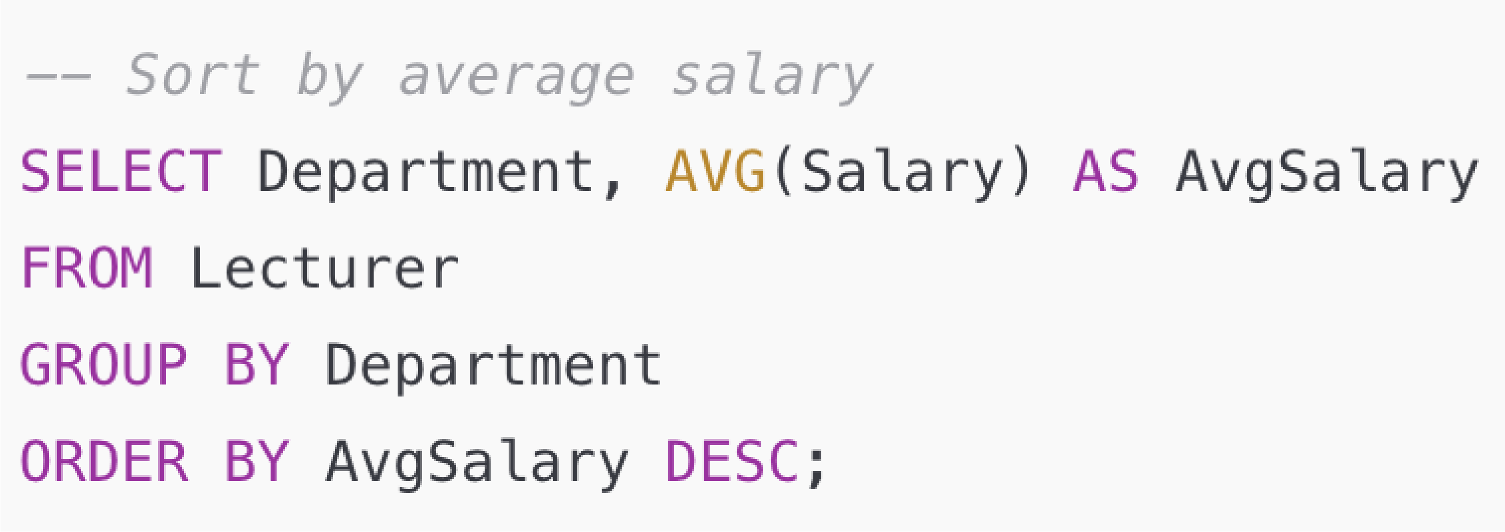
```
-- Departments with avg salary > 1100
SELECT Department, AVG(Salary) AS AvgSalary
FROM Lecturer
GROUP BY Department
HAVING AVG(Salary) > 1100;
```

- WHERE filters rows.
- HAVING filters groups.

GROUP BY + ORDER BY

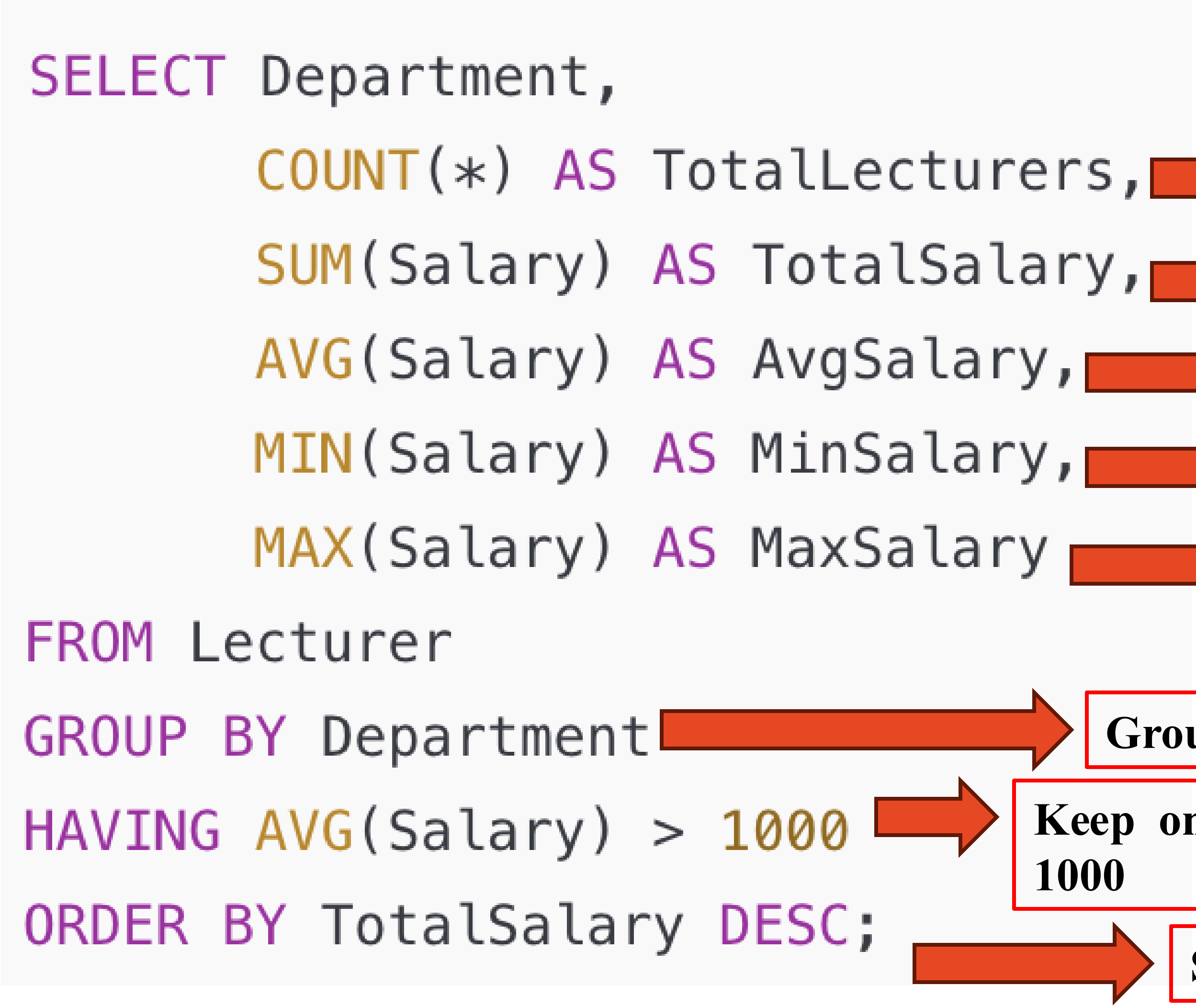


```sql
-- Sort departments by number of lecturers (descending)
SELECT Department, COUNT(*) AS NumLecturers
FROM Lecturer
GROUP BY Department
ORDER BY NumLecturers DESC;
```



```sql
-- Sort by average salary
SELECT Department, AVG(Salary) AS AvgSalary
FROM Lecturer
GROUP BY Department
ORDER BY AvgSalary DESC;
```

Full Department Summary Report



```sql
SELECT Department,
       COUNT(*) AS TotalLecturers,
       SUM(Salary) AS TotalSalary,
       AVG(Salary) AS AvgSalary,
       MIN(Salary) AS MinSalary,
       MAX(Salary) AS MaxSalary
FROM Lecturer
GROUP BY Department
HAVING AVG(Salary) > 1000
ORDER BY TotalSalary DESC;
```

Count how many lecturers belong to each department

Add all salaries in each department

Calculate average salary in each dept.

Find the lowest salary in each dept.

Find the highest salary in each department

Group rows by department (one result per department)

Keep only groups where the average salary is more than 1000

Sort the final results by total salary, highest first

References

- Budy, S., Reese, G., & Tahaghoghi, S. M. M. (Year unknown). *Learning MySQL*. O’Reilly Media.

- Beaulieu, A. (Year unknown). *MySQL Cookbook* (4th ed.). O’Reilly Media.

Any
Question