MySQL Database Unit 06 - Aggregating Queries

Chapter 06 - How to Code Summary Queries

SUMMARIZING: (EACH PRODUCES ONLY ONE ROW)

SELECT

COUNT(\*) 🡪 gives number of records in the Faculty table

SUM(salary) 🡪 gives sum of all faculty members salary

AVG(salary) 🡪 gives the average salary

MIN(salary) 🡪 gives the minimum salary

MAX(salary) 🡪 gives maximum salary

FROM faculty;

GROUP BY:

SELECT Department.Name AS Department,

Faculty.LastName

FROM Faculty

INNER JOIN

Department ON faculty.DepartmentID = department.ID ;

How to get Statistics/Info about faculty members, BUT BY DEPARTMENT

* With a one-to-many relationship:
* GROUP on the one side (Department)
* COUNT (sum, avg, min, max) on the many side (Faculty)
* **Must add ‘GROUP BY Department.Name ; ’**
* This will result in errors bc it only wants to produce one row per department, but the

‘Faculty.LastName’ will produce multiple rows per department

* **Replace that statement**:

SELECT Department.Name AS Department,

**COUNT(\*) AS ‘# of Faculty’**  **🡪 Counts on the many side (calculates the statistics)**

FROM Faculty

INNER JOIN

Department ON faculty.DepartmentID = department.ID ;

**GROUP BY Department.Name ; 🡪 Groups on the one side**

The Above Query:

* Produces one row per department
* Shows number of faculty in each department

THE HAVING CLAUSE: filters records (similar to the WHERE)

GROUP BY Department.Name

HAVING COUNT(\*) > 2 ; 🡪 shows departments with more than 2 faculty members

MULTIPLE TABLE JOINS WITH GROUP BY:

SELECT

Semester.Name AS Semester,

LastName,

FirstName,

COUNT(Section.ID) AS 'Classes Taken', 🡪 # of classes the student took

SUM(Course.CreditHours) AS 'Credit Hours', 🡪 Sum of credit hours

FORMAT(AVG(Registration.Grade), 2) AS GPA 🡪 Average grade - GPA

FROM

Student

INNER JOIN

Registration ON Registration.StudentID = Student.ID

INNER JOIN

Section ON Registration.SectionID = Section.ID

INNER JOIN

Course ON Section.CourseID = Course.ID

INNER JOIN

Semester ON Section.SemesterID = Semester.ID

**WHERE Semester.ID = 2 🡪 Filters on Individual Records**

GROUP BY Semester.ID, Student.ID

**HAVING AVG(Registration.Grade) >=3.5 🡪** Filters **on Statistics**

ORDER BY Semester.ID, LastName;

\*\*Method for writing complex queries like this\*\*

* SELECT individual records & JOIN all tables
* Add the STATISICS
* Add GROUP BY
* Filter using WHERE & HAVING