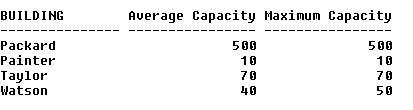
**WEEK 4:**

Write SQL Select Statements using Aggregate Functions, Group By and Having clauses for the following queries that retrieve data from university database:

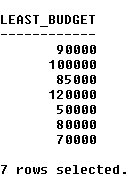
1. Find the maximum and average capacity of buildings in the university.

select building, avg(capacity) as "Average Capacity", max(capacity) as "Maximum Capacity" from classroom group by building;



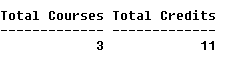
2. Display the least budget of the departments.

select min(budget) as Least\_Budget from department group by dept\_name;



3. Find the total number of courses and credits offered by Biology department.

select count(course\_id) as "Total Courses", sum(credits) as "Total Credits" from course where dept\_name like 'Biology';



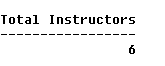
4. Find the average salary of instructors in the Computer Science department.

select avg(salary) as "Average Salary" from instructor where dept\_name like 'Comp. Sci.';



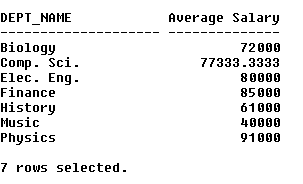
5. Find the total number of instructors who teach a course in the Spring 2010 semester.

select count(distinct(id)) as "Total Instructors" from teaches where semester like 'Spring' and year like 2010;



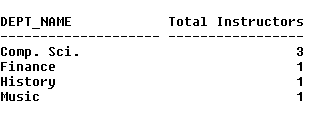
6. Find the average salary in each department.

select dept\_name, avg(salary) as "Average Salary" from instructor group by dept\_name;



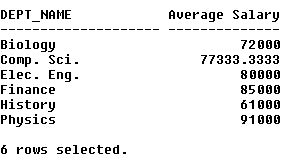
7. Find the number of instructors in each department who teach a course in the Spring 2010 semester.

select i.dept\_name, count(distinct(t.id)) as "Total Instructors" from instructor i, teaches t where semester like 'Spring' and year like 2010 and i.id = t.id group by i.dept\_name;



8. Find the department name and average salary of the department for only those departments where the average salary of the instructors is more than $42,000.

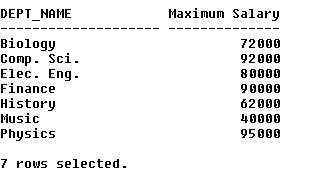
select dept\_name, avg(salary) as "Average Salary" from instructor group by dept\_name having avg(salary) >= 42000;



9. For each course section offered in 2009, find the average total credits (tot\_cred) of all students enrolled in the section, if the section had at least 2 students.

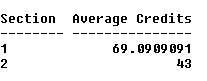
10. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.

select dept\_name, max(salary) as "Maximum Salary" from instructor group by dept\_name;



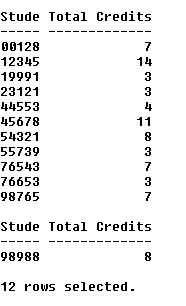
11. For the student with ID 12345 (or any other value), show the total number of credits scored for all courses (taken by that student). Don't display the tot\_creds value from the student table, you should use SQL aggregation on courses taken by the student.

select t.sec\_id as "Section ID", avg(tot\_cred) as "Average Credits" from takes t, student s where t.id = s.id and year = 2009 group by t.sec\_id having count(t.sec\_id) > 1;



12. Display the total credits for each of the students, along with the ID of the student; don't bother about the name of the student. (Don't display the tot\_creds value from the student table, you should use SQL aggregation on courses taken by the student. For students who have not registered for any course, tot\_creds should be 0)

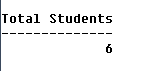
select s.id as "Student ID", sum(credits) as "Total Credits" from student s, course c, takes t where t.id = s.id and c.course\_id = t.course\_id group by s.id;



Write nested queries for answering the following queries that retrieve data from university database:

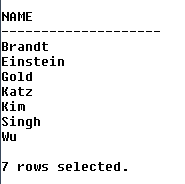
1. Find the total number of (distinct) students who have taken course sections taught by the instructor with ID 10101

select count(distinct(t.id)) as "Total Students" from takes t, teaches th where th.id = 10101 and t.course\_id = th.course\_id and t.sec\_id = th.sec\_id;



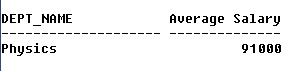
2. Find the names of all instructors whose salary is greater than at least one instructor in the Biology department.

select distinct(name) from instructor where salary > some(select salary from instructor where dept\_name like 'Biology');



3. Find the department that has the highest average salary.

select dept\_name, avg(salary) as "Average Salary" from instructor group by dept\_name HAVING avg(salary) >= all(select avg(salary) from instructor group by dept\_name);



4. Find all the courses taught in the both the Fall 2009 and Spring 2010 semesters.

select distinct(course\_id) from teaches where semester = 'Fall' and year = 2009 and course\_id in(select distinct(course\_id) from teaches where semester = 'Spring' and year = 2010);



5. Find all the courses taught in the Fall 2009 semester but not in the Spring 2010 semester.

select distinct(course\_id) from teaches where semester = 'Fall' and year = 2009 and course\_id not in(select distinct(course\_id) from teaches where semester = 'Spring' and year = 2010);



6. Find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester. (Write correlated nested Query)

select course\_id from teaches t1 where semester = 'Fall' and year = 2009 and exists(select course\_id from teaches t2 where semester = 'Spring' and year = 2010 and t1.course\_id = t2.course\_id);



7. Find all students who have taken all courses offered in the Biology department. (Write Correlated nested Query)

select distinct(s.id) from student s where not exists((select course\_id from course where dept\_name like 'Biology') minus (select t.course\_id from takes t where t.id = s.id));



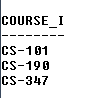
8. Find all courses that were offered at most once in 2009.

select distinct(course\_id) from takes where year = 2009 group by course\_id having count(course\_id) <= 1;



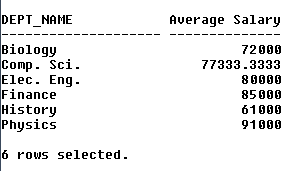
9. Find all courses that were offered at least twice in 2009.

select distinct(course\_id) from takes where year = 2009 group by course\_id having count(course\_id) >= 2;



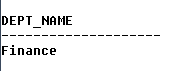
10. Find the average instructors salaries of those departments where the average salary is greater than $42000.

select dept\_name, avg(salary) as "Average Salary" from instructor group by dept\_name having avg(salary) >= 42000;



11. Find the department with the maximum budget.

select dept\_name from department where budget = (select max(budget) from department);



12. Find the names of instructors who have not taught any course.

select i.id from instructor i where not exists(select t.id from teaches t where t.id = i.id);



13. Find the IDs and names of all students who have not taken any course offering before Spring 2009.

14. Find the lowest, across all departments, of the per-department maximum salary computed.

15. Display the IDs and names of the instructors who have taught all Comp. Sci. courses.