**Ayush Goyal**

**190905522 CSE D 62**

**DBS Lab-4 (Week 4) – Complex Queries on SQL**

**Group By:**

1. **Find the number of students in each course.**

select title, count(distinct takes.ID) no\_of\_students

from student,takes,course

where student.ID=takes.ID and takes.course\_id=course.course\_id and student.dept\_name=course.dept\_name group by title;

Text

Description automatically generated

1. **Find those departments where the number of students are greater than 10. (Taken as 1 instead of 10 for this question).**

select dept\_name,count(ID) as total\_students from student group by dept\_name having count(ID) > 1;

Text

Description automatically generated

1. **Find the total number of courses in each department.**

select dept\_name,count(course\_id) from course group by dept\_name;

Text

Description automatically generated

1. **Find the names and average salaries of all departments whose average salary is greater than 42000.**

select dept\_name,avg(salary) from instructor group by dept\_name having avg(salary)>42000;

Text

Description automatically generated

1. **Find the enrolment of each section that was offered in Spring 2009.Ordering the display of Tuples**

select sec\_id,count(ID) from takes where semester='Spring' and year=2009 group by sec\_id;

A picture containing text

Description automatically generated

**(Use ORDER BY ASC/DESC):**

1. **List all the courses with prerequisite courses, then display course id in increasing order.**

select \* from prereq order by course\_id asc;

A picture containing text

Description automatically generated

1. **Display the details of instructors sorting the salary in decreasing order.**

select \* from instructor order by salary desc;

Text

Description automatically generated

**Derived Relations: (means using sub-queries in the from clause)**

1. **Find the maximum total salary across the departments.**

select max(total\_salary) from (select dept\_name, sum(salary) as total\_salary from instructor group by dept\_name);

Text

Description automatically generated

1. **Find the average instructors’ salaries of those departments where the average salary is greater than 42000.**

select dept\_name, avg\_salary from(select dept\_name,avg(salary) as avg\_salary from instructor group by dept\_name) where avg\_salary>42000;

Text

Description automatically generated

1. **Find the sections that had the maximum enrolment in Spring 2010**

select sec\_id, enroll from(select sec\_id,count(ID) as enroll from takes where semester='Spring' and year=2010 group by sec\_id) where enroll >=all(select count(ID) as enroll from takes where semester='Spring' and year=2010 group by sec\_id);

Text

Description automatically generated

1. **Find the names of all instructors who teach all students that belong to ‘CSE’ department.**

select distinct name from instructor natural join teaches where course\_id in (select distinct course\_id from student natural join takes where dept\_name='Comp. Sci.');

Text

Description automatically generated

1. **Find the average salary of those department where the average salary is greater than 50000 and total number of instructors in the department are more than 5.**

select dept\_name,total,average\_salary from(select dept\_name,count(\*) as total,avg(salary) as average\_salary from instructor group by dept\_name having count(ID)>2 and avg(salary)>5000);

Text

Description automatically generated

**With Clause:**

1. **Find all departments with the maximum budget.**

with budg(val) as (select max(budget) from department)

select dept\_name,budget from budg,department where budg.val=department.budget;

Text

Description automatically generated

1. **Find all departments where the total salary is greater than the average of the total salary at all departments.**

with tot(dept\_name,total) as (select dept\_name,sum(salary) as tot from instructor group by dept\_name), avge(val) as (select avg(total) from tot) select dept\_name, total from tot, avge where total>val;

Text

Description automatically generated

1. **Find the sections that had the maximum enrolment in Fall 2009.**

with totl(sec\_id, cnt) as (select sec\_id, count(distinct ID) from takes where semester='Fall' and year=2009 group by sec\_id), mx(val) as (select max(cnt) from totl) select sec\_id, cnt from totl, mx where cnt=val;

Text

Description automatically generated

1. **Select the names of those departments where the total credits earned by all the students is greater than the total credits earned by all the students in the Finance Department.**

with t1(dept\_name,total\_cred) as

(select dept\_name,sum(tot\_cred) from student group by dept\_name),

t2(value) as (select total\_cred from t1

where dept\_name='Finance')

select dept\_name from t1,t2

where total\_cred>value;

Text

Description automatically generated

**(Use ROLLBACK (and SAVEPOINT) to undo the effect of any modification on database before COMMIT)**

1. **Delete all the instructors of Finance department.**

savepoint Q17;

delete from instructor where dept\_name = 'Finance';

Text

Description automatically generated

1. **Delete all courses in CSE department.**

savepoint Q18;

delete from course where dept\_name='Comp. Sci.';

A picture containing text

Description automatically generated

1. **Transfer all the students from CSE department to IT department.**

update student set dept\_name=’IT' where dept\_name='Comp.Sci.';

Text

Description automatically generated

1. **Increase salaries of instructors whose salary is over $100,000 by 3%, and all others receive a 5% raise.**

update instructor

set salary=case

when salary>100000

then salary\*1.03

else

salary\*1.05

end;

rollback;

Text

Description automatically generated

**THE END**