**Ayush Goyal**

**190905522 CSE D 62**

**DBS Lab 3(Week 3) – Intermediate SQL**

**Set Operations:**

**UNION (Use union all to retain duplicates):**

**1. Find courses that ran in Fall 2009 or in Spring 2010**

select course\_id from section where semester='Fall' and year=2009 union select course\_id from section where semester='Spring' and year=2010;

Text

Description automatically generated

**INTERSECT (Use intersect all to retain duplicates):**

**2. Find courses that ran in Fall 2009 and in spring 2010**

select course\_id from section where semester='Fall' and year=2009 intersect select course\_id from section where semester='Spring' and year=2010;

Text

Description automatically generated with low confidence

**MINUS:**

**3. Find courses that ran in Fall 2009 but not in Spring 2010**

select course\_id from section where semester='Fall' and year=2009 minus select course\_id from section where semester='Spring' and year=2010;

Text

Description automatically generated

**Null values**

**4. Find the name of the course for which none of the students registered.**

select title from course where course\_id NOT IN(select takes.course\_id from takes);

Text

Description automatically generated

**Nested Subqueries**

**Set Membership (in / not in):**

**5. Find courses offered in Fall 2009 and in Spring 2010.**

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

Text

Description automatically generated

**6. Find the total number of students who have taken course taught by the instructor with ID 10101.**

select count(takes.id) from takes where course\_id in(select course\_id from teaches where id='10101');

Text

Description automatically generated

**7. Find courses offered in Fall 2009 but not in Spring 2010.**

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

Text

Description automatically generated

**8. Find the names of all students whose name is same as the instructor’s name.**

select name from student where name in(select name from instructor);

Text

Description automatically generated

**Set Comparison (>=some/all)**

**9. Find names of instructors with salary greater than that of some (at least one) instructor**

**in the Biology department.**

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

Text

Description automatically generated

**10. Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.**

select name from instructor where salary > all(select salary from instructor where dept\_name='Biology');

Text

Description automatically generated

**11. Find the departments that have the highest average salary.**

select dept\_name from(select dept\_name,avg(salary) avg\_salary from instructor group by dept\_name) where avg\_salary = (select max(avg\_salary) from (select dept\_name,avg(salary) avg\_salary from instructor group by dept\_name));

Text

Description automatically generated

**12. Find the names of those departments whose budget is lesser than the average salary of all instructors.**

select dept\_name from department where budget < (select avg(salary) from instructor);

Text

Description automatically generated

**Test for Empty Relations (exists/ not exists)**

**13. Find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester.**

select course\_id

from section S

where semester = 'Spring'

and year = 2010

and exists(

select course\_id

from section T

where semester = 'Fall'

and year = 2009

and T.course\_id = S.course\_id);

Text

Description automatically generated

**14. Find all students who have taken all courses offered in the Biology department.**

select distinct S.ID,

S.name

from student S

where not exists(

(

select course\_id

from course

where dept\_name = 'Biology'

)

minus

(

select T.course\_id

from takes T

where S.ID = T.ID

)

);

**Text

Description automatically generated**

**Test for Absence of Duplicate Tuples**

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

select course\_id from(select course\_id, count(\*)count from section where section.year=2009 group by course\_id) where count=1;

**Text

Description automatically generated**

**16. Find all the students who have opted at least two courses offered by CSE department.**

select ID from (select ID, count(\*) as c from takes natural join course where dept\_name = 'Comp. Sci.' group by ID) where c >= 2;

Text

Description automatically generated

**Subqueries in the From Clause**

**17. Find the average instructors salary of those departments where the average salary is greater than 42000**

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

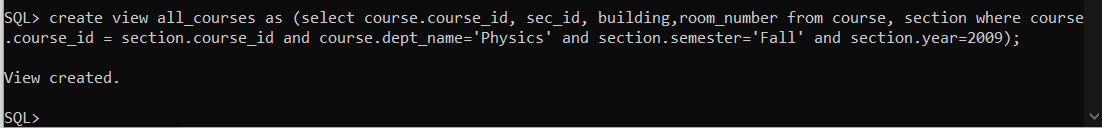
**Text

Description automatically generated**

**Views**

**18. Create a view all\_courses consisting of course sections offered by Physics department in the Fall 2009, with the building and room number of each section.**

create view all\_courses as (select course.course\_id, sec\_id, building,room\_number from course, section where course.course\_id = section.course\_id and course.dept\_name='Physics' and section.semester='Fall' and section.year=2009);

****

**19. Select all the courses from all\_courses view.**

select \* from all\_courses;

Shape

Description automatically generated with medium confidence

**20. Create a view department\_total\_salary consisting of department name and total salary of that department.**

create view department\_total\_salary as (select dept\_name,sum(salary) as total\_sal from instructor group by dept\_name);

select \* from department\_total\_salary;

Text

Description automatically generated

**THE END**