**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;

```
SQL> select course_id from section where semester='Fall' and year=2009 union select course_id from section where semeste
r='Spring' and year=2010;

COURSE_I
--------
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

8 rows selected.

SQL>
```

**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;

```
SQL> select course_id from section where semester='Fall' and year=2009 intersect select course_id from section where sem
ester='Spring' and year=2010;

COURSE_I
--------
CS-101

SQL>
```

**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;

```
SQL> select course_id from section where semester='Fall' and year=2009 minus select course_id from section where semeste
r='Spring' and year=2010;

COURSE_I
--------
CS-347
PHY-101

SQL>
```

**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);

```
SQL> select title from course where course_id NOT IN(select takes.course_id from takes);

TITLE
--------------------------------------------------
Computational Biology

SQL>
```

**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);

```
SQL> 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);

COURSE_I
--------
CS-101

SQL>
```

**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');

```
SQL> select count(takes.id) from takes where course_id in(select course_id from teaches where id='10101');

COUNT(TAKES.ID)
---------------
             11

SQL>
```

**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);

```
SQL> select distinct course_id from section where semester='Fall' and year=2009 and course_id not in(select course_id fr
om section where semester='Spring' and year=2010);

COURSE_I
--------
CS-347
PHY-101

SQL>
```

**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);

```
SQL> select name from student where name in(select name from instructor);

NAME
--------------------
Brandt

SQL>
```

**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');

```
SQL> select name from instructor where salary > some(select salary from instructor where dept_name='Biology');

NAME
--------------------
Einstein
Brandt
Wu
Gold
Kim
Singh
Katz

7 rows selected.

SQL>
```

**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');

```
SQL> select name from instructor where salary > all(select salary from instructor where dept_name='Biology');

NAME
--------------------
Katz
Singh
Kim
Gold
Wu
Brandt
Einstein

7 rows selected.

SQL>
```

**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));

```
SQL> 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));

DEPT_NAME
--------------------
Physics

SQL>
```

**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);

```
SQL> select dept_name from department where budget < (select avg(salary) from instructor);

DEPT_NAME
--------------------
History
Physics

SQL>
```

**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);
```

```
SQL> select course_id
  2  from section S
  3  where semester = 'Spring'
  4      and year = 2010
  5      and exists(
  6          select course_id
  7          from section T
  8          where semester = 'Fall'
  9              and year = 2009
 10              and T.course_id = S.course_id
 11      );

COURSE_I
--------
CS-101

SQL>
```

**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
      )
   );
```

```
SQL> select distinct S.ID,
  2      S.name
  3  from student S
  4  where not exists(
  5          (
  6              select course_id
  7              from course
  8              where dept_name = 'Biology'
  9          )
 10          minus
 11          (
 12              select T.course_id
 13              from takes T
 14              where S.ID = T.ID
 15          )
 16      );

no rows selected

SQL>
```

**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;

```
SQL> select course_id from(select course_id, count(*)count from section where section.year=2009 group by course_id) wher
e count=1;

COURSE_I
--------
BIO-101
CS-101
CS-347
EE-181
PHY-101

SQL>
```

**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;

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

ID
-----
00128
98765
12345
54321
76543
45678

6 rows selected.

SQL>
```

## 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;

```
SQL> select dept_name, average from(select dept_name, avg(salary) as average from instructor group by dept_name) where a
verage > 42000;

DEPT_NAME              AVERAGE
-------------------- ----------
Elec. Eng.                80000
Physics                   91000
Comp. Sci.            77333.3333
Finance                   85000
Biology                   72000
History                   61000

6 rows selected.

SQL>
```

## 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);

```
SQL> 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);

View created.

SQL>
```

**19. Select all the courses from all_courses view.**

select * from all_courses;

```
SQL> select * from all_courses;

COURSE_I SEC_ID   BUILDING        ROOM_NU
-------- -------- --------------- -------
PHY-101  1        Watson          100

SQL>
```

**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;

```
SQL> create view department_total_salary as (select dept_name,sum(salary) as total_sal from instructor group by dept_name);


View created.

SQL> select * from department_total_salary;

DEPT_NAME            TOTAL_SAL
-------------------- ----------
Elec. Eng.                80000
Physics                  182000
Comp. Sci.               232000
Finance                  170000
Biology                   72000
Music                     40000
History                  122000

7 rows selected.

SQL>
```

**THE END**