East West University
Department of Computer Science and Engineering

CSE 302: LAB 05 (Handout)
Course Instructor: Mahmuda Rawnak Jahan

Introducing Nested Subqueries and Outer Joins in SQL

1. Course (course_id, title, dept_name, credits)
2. Section (course_id, sec_id, semester, year, building, room_number, time_slot_id)
3. Teaches (ID, course_id, sec_id, semester, year)
4. Instructor (ID, name, dept_name, salary)
5. Student (ID, name, dept_name, tot_cred)
6. Takes (ID, course_id, sec_id, semester, year, grade)
7. Department (dept_name, building, budget)

Subqueries in the WHERE clause

A. IN / NOT IN

- Find courses offered in Fall 2009 and in Spring 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
);
```

- Find courses offered in Fall 2009 but not in Spring 2010

```sql
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
);
```

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

```sql
SELECT COUNT(DISTINCT student_id) AS total_students
FROM takes
WHERE course_id IN ( SELECT course_id FROM section
WHERE instructor_id = 10101 );
```

B. SOME / ALL

- **Find names of instructors with salaries greater than that of some instructor in the Biology department**

```
SELECT name FROM instructor WHERE salary > SOME ( SELECT
  salary FROM instructor WHERE dept_name = 'Biology');
```

- **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' );
```

C. EXISTS/NOT EXISTS

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

```
SELECT DISTINCT course_id FROM section s1
WHERE semester = 'Fall' AND year = 2009 AND
EXISTS ( SELECT 1 FROM section s2 WHERE s1.course_id =
s2.course_id AND semester = 'Spring' AND year = 2010 );
```

- **Find all courses taught in Fall 2009 semester but not in the Spring 2010 semester**

```
SELECT DISTINCT course_id FROM section s1
WHERE semester = 'Fall' AND year = 2009 AND NOT EXISTS (
SELECT 1 FROM section s2 WHERE s1.course_id =
s2.course_id AND semester = 'Spring' AND year = 2010 );
```

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

```
  SELECT ID FROM Student s
WHERE NOT EXISTS ( SELECT course_id FROM Course
    WHERE dept_name = 'Biology'
    AND course_id NOT IN ( SELECT course_id FROM Takes
    WHERE ID = s.ID
    )
);
```

➔ **Subqueries in the FROM clause**

- **Find the average instructors' salaries of those departments where the average salary is greater than $42,000**

```
  SELECT dept_name, AVG(salary) AS avg_salary
FROM Instructor
GROUP BY dept_name
HAVING AVG(salary) > 42000;
```

➔ **Complex Queries using WITH clause**

- **Find all departments with the maximum budget**

```
  WITH DeptBudget AS (
    SELECT dept_name, MAX(budget) AS max_budget
    FROM Department )
SELECT dept_name
FROM DeptBudget
WHERE budget = max_budget;
```

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

```
  WITH DeptSalary AS (
    SELECT dept_name, SUM(salary) AS total_salary
    FROM Instructor
    GROUP BY dept_name
)
SELECT dept_name
FROM DeptSalary
WHERE total_salary > (SELECT AVG(total_salary) FROM
  DeptSalary);
```

➔ **Subqueries in the SELECT clause (Scalar Subquery)**

- **Find number of instructors for each department**

```
SELECT dept_name,
       (SELECT COUNT(*)
        FROM Instructor i
        WHERE i.dept_name = d.dept_name) AS num_instructors
FROM Department d;
```

➔ Performing Outer Joins

- **Left Outer Join Example**

```
     SELECT d.dept_name, i.name
FROM Department d
LEFT   OUTER   JOIN   Instructor   i   ON
d.dept_name = i.dept_name;
```

- **Right Outer Join Example**

```
     SELECT i.name, d.dept_name
FROM Instructor i
RIGHT   OUTER   JOIN   Department   d   ON
i.dept_name = d.dept_name;
```

- **Full Outer Join Example**

```
     SELECT d.dept_name, i.name
FROM Department d
FULL   OUTER   JOIN   Instructor   i   ON
d.dept_name = i.dept_name;
```

- **Find the number of instructors for each department, including departments with no instructor**

```
     SELECT  d.dept_name,  COUNT(i.ID)
     AS num_instructors
FROM Department d
LEFT   OUTER   JOIN   Instructor   i   ON
d.dept_name = i.dept_name
GROUP BY d.dept_name;
```