**Lab Tasks:**

**➔ Subqueries in the WHERE clause**

**A. IN / NOT IN**

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

SELECT course\_id, title

FROM course

WHERE course\_id IN (SELECT course\_id

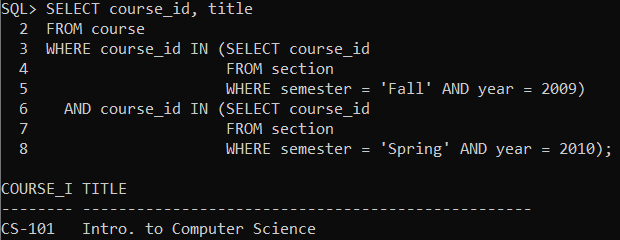
FROM section

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**

SELECT course\_id, title

FROM course

WHERE course\_id IN (SELECT course\_id

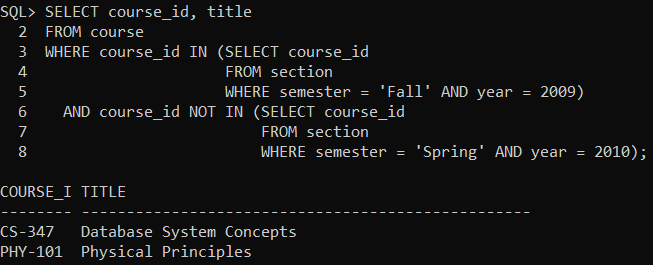
FROM section

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**

SELECT COUNT(DISTINCT T.ID) AS num\_students

FROM takes T

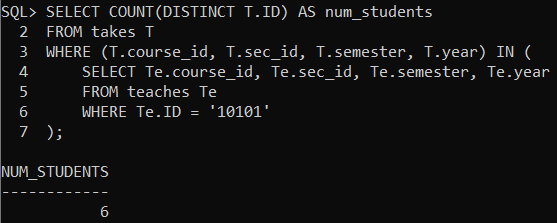
WHERE (T.course\_id, T.sec\_id, T.semester, T.year) IN (

SELECT Te.course\_id, Te.sec\_id, Te.semester, Te.year

FROM teaches Te

WHERE Te.ID = '10101'

);



**B. SOME / ALL**

* **Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.**

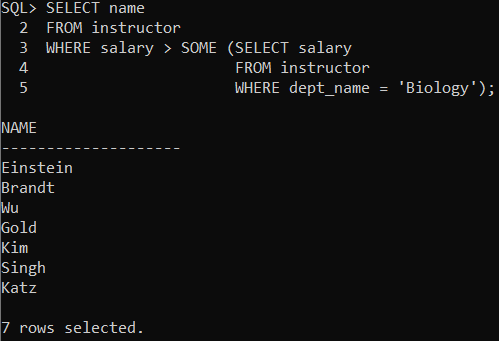
SELECT name

FROM instructor

WHERE salary > SOME (SELECT salary

FROM instructor

WHERE dept\_name = 'Biology');



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

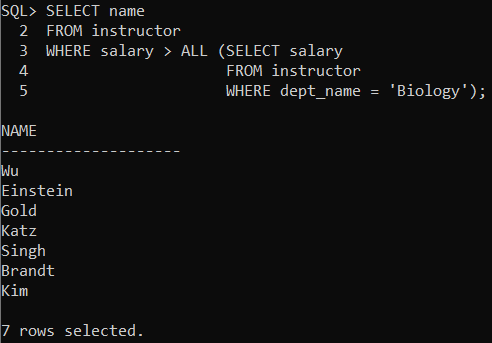
SELECT name

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 C.course\_id, C.title

FROM course C

WHERE EXISTS (SELECT S1.course\_id

FROM section S1

WHERE S1.course\_id = C.course\_id

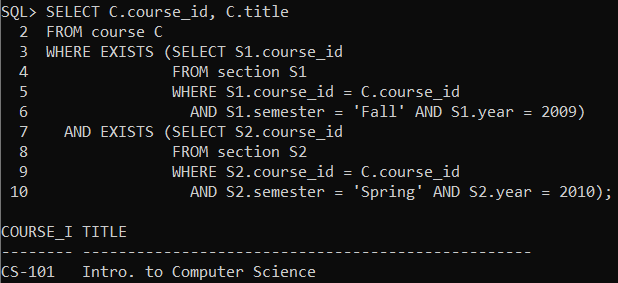
AND S1.semester = 'Fall' AND S1.year = 2009)

AND EXISTS (SELECT S2.course\_id

FROM section S2

WHERE S2.course\_id = C.course\_id

AND S2.semester = 'Spring' AND S2.year = 2010);



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

SELECT C.course\_id, C.title

FROM course C

WHERE EXISTS (SELECT S1.course\_id

FROM section S1

WHERE S1.course\_id = C.course\_id

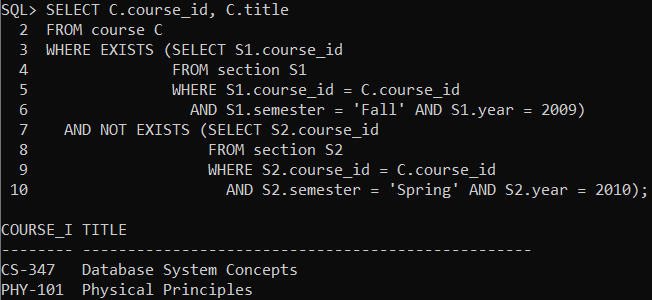
AND S1.semester = 'Fall' AND S1.year = 2009)

AND NOT EXISTS (SELECT S2.course\_id

FROM section S2

WHERE S2.course\_id = C.course\_id

AND S2.semester = 'Spring' AND S2.year = 2010);



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

SELECT S.ID, S.name

FROM student S

WHERE EXISTS (

-- Find Biology courses...

SELECT C.course\_id

FROM course C

WHERE C.dept\_name = 'Biology'

AND EXISTS (

-- ...that the student S has NOT taken

SELECT T.course\_id

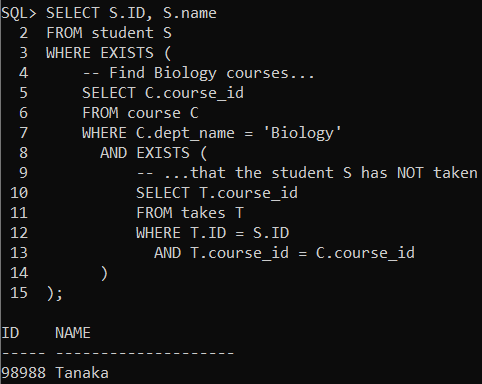
FROM takes T

WHERE T.ID = S.ID

AND T.course\_id = C.course\_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

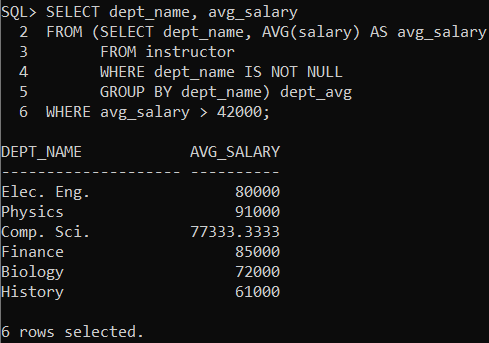
FROM (SELECT dept\_name, AVG(salary) AS avg\_salary

FROM instructor

WHERE dept\_name IS NOT NULL

GROUP BY dept\_name) dept\_avg

WHERE avg\_salary > 42000;



**➔ Complex Queries using WITH clause**

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

WITH MaxBudget AS ( -- Removed the column alias list (value) from here

SELECT MAX(budget) AS value

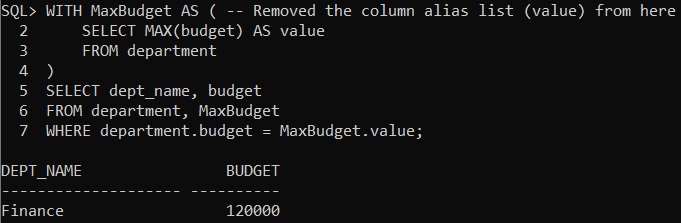
FROM department

)

SELECT dept\_name, budget

FROM department, MaxBudget

WHERE department.budget = MaxBudget.value;



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

WITH DepTotalSalary AS ( -- Removed (dept\_name, total\_salary)

SELECT dept\_name, SUM(salary) AS total\_salary -- Added alias AS total\_salary here

FROM instructor

WHERE dept\_name IS NOT NULL

GROUP BY dept\_name

),

AvgTotalSalary AS ( -- Removed (avg\_value)

SELECT AVG(total\_salary) AS avg\_value -- Added alias AS avg\_value here

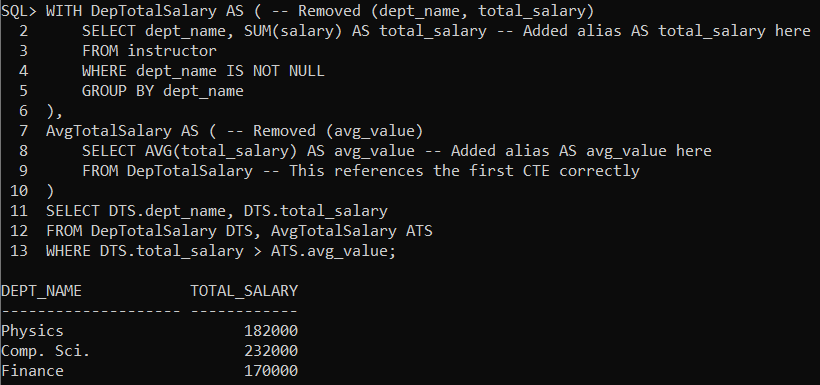
FROM DepTotalSalary -- This references the first CTE correctly

)

SELECT DTS.dept\_name, DTS.total\_salary

FROM DepTotalSalary DTS, AvgTotalSalary ATS

WHERE DTS.total\_salary > ATS.avg\_value;



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

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

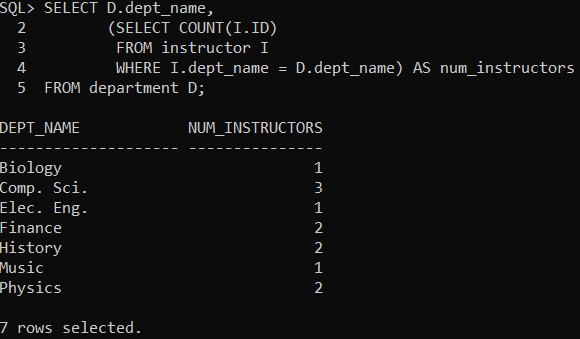
SELECT D.dept\_name,

(SELECT COUNT(I.ID)

FROM instructor I

WHERE I.dept\_name = D.dept\_name) AS num\_instructors

FROM department D;



**➔ Performing Outer Joins**

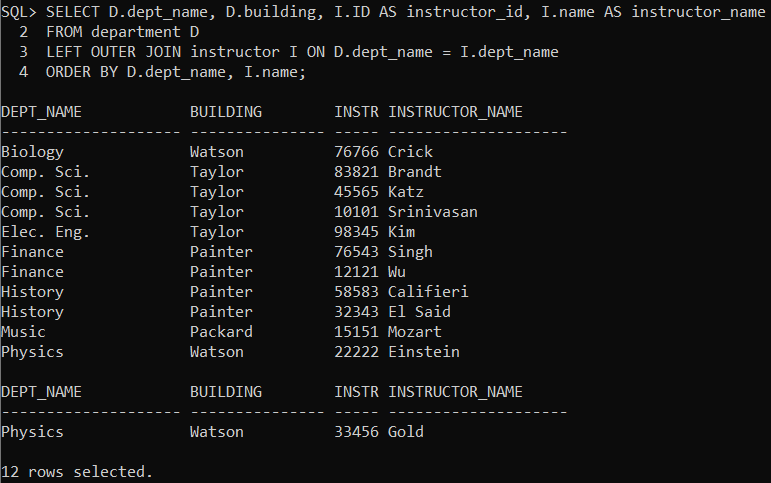
* **Left Outer Join**

SELECT D.dept\_name, D.building, I.ID AS instructor\_id, I.name AS instructor\_name

FROM department D

LEFT OUTER JOIN instructor I ON D.dept\_name = I.dept\_name

ORDER BY D.dept\_name, I.name;



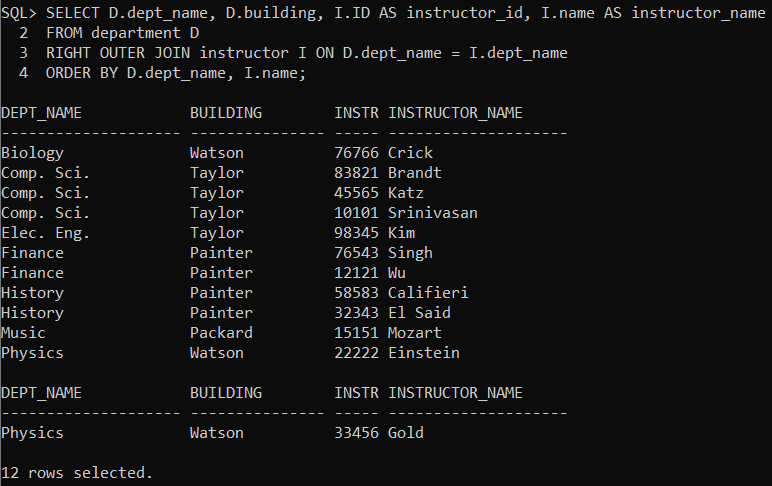
* **Right Outer Join**

SELECT D.dept\_name, D.building, I.ID AS instructor\_id, I.name AS instructor\_name

FROM department D

RIGHT OUTER JOIN instructor I ON D.dept\_name = I.dept\_name

ORDER BY D.dept\_name, I.name;



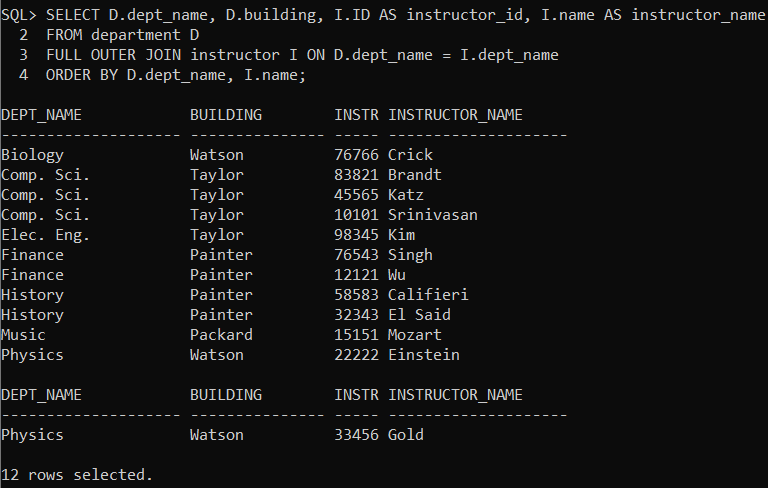
* **Full Outer Join**

SELECT D.dept\_name, D.building, I.ID AS instructor\_id, I.name AS instructor\_name

FROM department D

FULL OUTER JOIN instructor I ON D.dept\_name = I.dept\_name

ORDER BY D.dept\_name, I.name;



* **Find number of instructors for each department. You must include 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

ORDER BY D.dept\_name;

