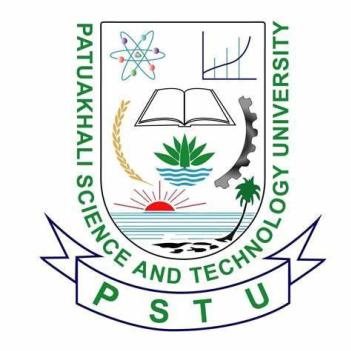
**PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY**

****

**COURSE CODE CCE-224**

**SUBMITTED TO:**

**Prof. Dr. Md Samsuzzaman sobuz**

**Department of Computer and Communication**

**Engineering**

**Faculty of Computer Science and Engineering**

****

**SUBMITTED BY:**

**Abul Basar**

ID: **2102036**,

Registration No: **10163**

**Faculty of Computer Science and Engineering**

****

Date of submission: **14-05-2025**

**Lab Problem: 02**

1. **Find out the ID and salary of the instructors.**

SELECT ID, salary FROM instructor;

1. **Find out the ID and salary of the instructor who gets more than $85,000.**

SELECT ID, salary FROM instructor WHERE salary > 85000;

1. **Find out the department names and their budget at the university.**

SELECT dept\_name, budget FROM department;

1. **List out the names of the instructors from Computer Science who have more than $70,000.**

SELECT name FROM instructor

WHERE dept\_name = 'Computer Science' AND salary > 70000;

1. **For all instructors in the university who have taught some course, find their names and the course ID of all courses they taught.**

SELECT i.name, t.course\_id

FROM instructor i JOIN teaches t ON i.ID = t.ID;

1. **Find the names of all instructors whose salary is greater than at least one instructor in the Biology department.**

SELECT name FROM instructor

WHERE salary > ANY (SELECT salary FROM instructor WHERE dept\_name = 'Biology');

1. **Find the advisor of the student with ID 12345**

SELECT i.name FROM advisor a JOIN instructor i ON a.i\_ID = i.ID

WHERE a.s\_ID = '12345';

1. **Find the average salary of all instructors.**

SELECT AVG(salary) FROM instructor;

1. **Find the names of all departments whose building name includes the substring Watson.**

SELECT dept\_name FROM department WHERE building LIKE '%Watson%';

1. **Find the names of instructors with salary amounts between 90,000and90,000*and*100,000.**

SELECT name FROM instructor WHERE salary BETWEEN 90000 AND 100000;

1. **Find the instructor names and the courses they taught for all instructors in the Biology department who have taught some course.**

SELECT i.name, t.course\_id

FROM instructor i JOIN teaches t ON i.ID = t.ID

WHERE i.dept\_name = 'Biology';

1. **Find the courses taught in Fall-2009 semester.**

SELECT DISTINCT course\_id FROM teaches WHERE semester = 'Fall' AND year = 2009;

1. **Find the set of all courses taught either in Fall-2009 or in Spring-2010.**

SELECT DISTINCT course\_id FROM teaches

WHERE (semester = 'Fall' AND year = 2009)

OR (semester = 'Spring' AND year = 2010);

1. **Find the set of all courses taught in the Fall-2009 as well as in Spring-2010.**

SELECT DISTINCT t1.course\_id

FROM teaches t1 JOIN teaches t2 ON t1.course\_id = t2.course\_id

WHERE t1.semester = 'Fall' AND t1.year = 2009

AND t2.semester = 'Spring' AND t2.year = 2010;

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

SELECT DISTINCT course\_id FROM teaches

WHERE semester = 'Fall' AND year = 2009

AND course\_id NOT IN (

SELECT course\_id FROM teaches

WHERE semester = 'Spring' AND year = 2010

);

1. **Find all instructors who appear in the instructor relation with null values for salary.**

SELECT \* FROM instructor WHERE salary IS NULL;

1. **Find the average salary of instructors in the Finance department.**

SELECT AVG(salary) FROM instructor WHERE dept\_name = 'Finance';

1. **Find the total number of instructors who teach a course in the Spring-2010 semester.**

SELECT COUNT(DISTINCT ID) FROM teaches

WHERE semester = 'Spring' AND year = 2010;

1. **Find the average salary in each department.**

SELECT dept\_name, AVG(salary)

FROM instructor GROUP BY dept\_name;

1. **Find the number of instructors in each department who teach a course in the Spring-2010 semester.**

SELECT i.dept\_name, COUNT(DISTINCT i.ID)

FROM instructor i JOIN teaches t ON i.ID = t.ID

WHERE t.semester = 'Spring' AND t.year = 2010

GROUP BY i.dept\_name;

1. **List out the departments where the average salary of the instructors is more than $42,000.**

SELECT dept\_name FROM instructor

GROUP BY dept\_name HAVING AVG(salary) > 42000;

1. **For each course section offered in 2009, find the average total credits (tot cred) of all students enrolled in the section, if the section had at least 2 students.**

SELECT s.course\_id, s.sec\_id, AVG(st.tot\_cred)

FROM section s JOIN takes t ON s.course\_id = t.course\_id AND s.sec\_id = t.sec\_id

JOIN student st ON t.ID = st.ID

WHERE s.year = 2009

GROUP BY s.course\_id, s.sec\_id

HAVING COUNT(t.ID) >= 2;

1. **Find all the courses taught in both the Fall-2009 and Spring-2010 semesters.**

SELECT DISTINCT t1.course\_id

FROM teaches t1 JOIN teaches t2 ON t1.course\_id = t2.course\_id

WHERE t1.semester = 'Fall' AND t1.year = 2009

AND t2.semester = 'Spring' AND t2.year = 2010;

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

SELECT DISTINCT course\_id FROM teaches

WHERE semester = 'Fall' AND year = 2009

AND course\_id NOT IN (

SELECT course\_id FROM teaches

WHERE semester = 'Spring' AND year = 2010

);

1. **Select the names of instructors whose names are neither "Mozart" nor "Einstein".**

SELECT name FROM instructor

WHERE name NOT IN ('Mozart', 'Einstein');

1. **Find the total number of (distinct) students who have taken course sections taught by the instructor with ID 110011.**

SELECT COUNT(DISTINCT t.ID)

FROM takes t JOIN teaches te ON t.course\_id = te.course\_id

AND t.sec\_id = te.sec\_id AND t.semester = te.semester AND t.year = te.year

WHERE te.ID = '110011';

Intermediate Queries (27-52)

1. **Find the ID and names of all instructors whose salary is greater than at least one instructor in the History department.**

SELECT ID, name FROM instructor

WHERE salary > ANY (SELECT salary FROM instructor WHERE dept\_name = 'History');

1. **Find the names of all instructors that have a salary value greater than that of each instructor in the Biology department.**

SELECT name FROM instructor

WHERE salary > ALL (SELECT salary FROM instructor WHERE dept\_name = 'Biology');

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

SELECT dept\_name FROM instructor

GROUP BY dept\_name

HAVING AVG(salary) >= ALL (

SELECT AVG(salary) FROM instructor GROUP BY dept\_name

);

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

SELECT DISTINCT t1.course\_id

FROM teaches t1 JOIN teaches t2 ON t1.course\_id = t2.course\_id

WHERE t1.semester = 'Fall' AND t1.year = 2009

AND t2.semester = 'Spring' AND t2.year = 2010;

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

SELECT s.ID, s.name

FROM student s

WHERE NOT EXISTS (

SELECT c.course\_id FROM course c

WHERE c.dept\_name = 'Biology'

EXCEPT

SELECT t.course\_id FROM takes t

WHERE t.ID = s.ID

);

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

SELECT course\_id FROM teaches

WHERE year = 2009

GROUP BY course\_id

HAVING COUNT(\*) <= 1;

1. **Find all courses that were offered at least twice in 2009.**

SELECT course\_id FROM teaches

WHERE year = 2009

GROUP BY course\_id

HAVING COUNT(\*) >= 2;

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

SELECT dept\_name, AVG(salary)

FROM instructor

GROUP BY dept\_name

HAVING AVG(salary) > 42000;

1. **Find the maximum across all departments of the total salary at each department.**

SELECT MAX(total\_salary)

FROM (SELECT SUM(salary) as total\_salary

FROM instructor GROUP BY dept\_name) AS dept\_salaries;

1. **List all departments along with the number of instructors in each department.**

SELECT dept\_name, COUNT(ID)

FROM instructor

GROUP BY dept\_name;

1. **Find the titles of courses in the Comp. Sci. department that has 3 credits.**

SELECT title FROM course

WHERE dept\_name = 'Comp. Sci.' AND credits = 3;

1. **Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.**

SELECT DISTINCT t.ID

FROM takes t JOIN teaches te ON t.course\_id = te.course\_id

AND t.sec\_id = te.sec\_id AND t.semester = te.semester AND t.year = te.year

JOIN instructor i ON te.ID = i.ID

WHERE i.name = 'Einstein';

1. **Find the highest salary of any instructor.**

SELECT MAX(salary) FROM instructor;

1. **Find all instructors earning the highest salary (there may be more than one with the same salary).**

SELECT \* FROM instructor

WHERE salary = (SELECT MAX(salary) FROM instructor);

1. **Find the enrollment of each section that was offered in Autumn-2009.**

SELECT s.course\_id, s.sec\_id, COUNT(t.ID) as enrollment

FROM section s LEFT JOIN takes t ON s.course\_id = t.course\_id

AND s.sec\_id = t.sec\_id AND s.semester = t.semester AND s.year = t.year

WHERE s.semester = 'Fall' AND s.year = 2009

GROUP BY s.course\_id, s.sec\_id;

1. **Find the maximum enrollment, across all sections, in Autumn-2009.**

SELECT MAX(enrollment)

FROM (SELECT COUNT(t.ID) as enrollment

FROM section s JOIN takes t ON s.course\_id = t.course\_id

AND s.sec\_id = t.sec\_id AND s.semester = t.semester AND s.year = t.year

WHERE s.semester = 'Fall' AND s.year = 2009

GROUP BY s.course\_id, s.sec\_id) AS enrollments;

1. **Find the salaries after the following operation: Increase the salary of each instructor in the Comp. Sci. department by 10%.**

SELECT ID, name, salary\*1.1 as new\_salary

FROM instructor

WHERE dept\_name = 'Comp. Sci.';

1. **Find all students who have not taken a course.**

SELECT ID, name FROM student

WHERE ID NOT IN (SELECT DISTINCT ID FROM takes);

1. **List all course sections offered by the Physics department in the Fall-2009 semester, with the building and room number of each section.**

SELECT s.course\_id, s.sec\_id, s.building, s.room\_number

FROM section s JOIN course c ON s.course\_id = c.course\_id

WHERE c.dept\_name = 'Physics' AND s.semester = 'Fall' AND s.year = 2009;

1. **Find the student names who take courses in Spring-2010 semester at Watson Building.**

SELECT DISTINCT st.name

FROM student st JOIN takes t ON st.ID = t.ID

JOIN section s ON t.course\_id = s.course\_id AND t.sec\_id = s.sec\_id

AND t.semester = s.semester AND t.year = s.year

WHERE t.semester = 'Spring' AND t.year = 2010 AND s.building = 'Watson';

1. **List the students who take courses teaches by Brandt.**

SELECT DISTINCT st.name

FROM student st JOIN takes t ON st.ID = t.ID

JOIN teaches te ON t.course\_id = te.course\_id

AND t.sec\_id = te.sec\_id AND t.semester = te.semester AND t.year = te.year

JOIN instructor i ON te.ID = i.ID

WHERE i.name = 'Brandt';

1. **Find out the average salary of the instructor in each department.**

SELECT dept\_name, AVG(salary)

FROM instructor

GROUP BY dept\_name;

1. **Find the number of students who take the course titled "Intro. To Computer Science".**

SELECT COUNT(DISTINCT t.ID)

FROM takes t JOIN course c ON t.course\_id = c.course\_id

WHERE c.title = 'Intro. To Computer Science';

1. **Find out the total salary of the instructors of the Computer Science department who take a course(s) in Watson building.**

SELECT SUM(i.salary)

FROM instructor i JOIN teaches te ON i.ID = te.ID

JOIN section s ON te.course\_id = s.course\_id AND te.sec\_id = s.sec\_id

AND te.semester = s.semester AND te.year = s.year

WHERE i.dept\_name = 'Computer Science' AND s.building = 'Watson';

1. **Find out the course titles which starts between 10:00 to 12:00.**

SELECT DISTINCT c.title

FROM course c JOIN section s ON c.course\_id = s.course\_id

WHERE s.time\_slot\_id IN (

SELECT time\_slot\_id FROM time\_slot

WHERE start\_hr BETWEEN 10 AND 11

OR (start\_hr = 12 AND start\_min = 0)

);

1. **List the course names where CS-101 is the pre-requisite course.**

SELECT c.title

FROM course c JOIN prereq p ON c.course\_id = p.course\_id

WHERE p.prereq\_id = 'CS-101';

Advanced Queries (53-70)

1. **List the student names who get more than B+ grades in their respective courses.**

SELECT DISTINCT s.name

FROM student s JOIN takes t ON s.ID = t.ID

WHERE t.grade IN ('A', 'A-', 'A+', 'B', 'B+');

1. **Find the student who takes the maximum credit from each department.**

WITH dept\_max AS (

SELECT dept\_name, MAX(tot\_cred) as max\_cred

FROM student

GROUP BY dept\_name

)

SELECT s.dept\_name, s.ID, s.name, s.tot\_cred

FROM student s JOIN dept\_max d ON s.dept\_name = d.dept\_name AND s.tot\_cred = d.max\_cred;

1. **Find out the student ID and grades who take a course(s) in Spring-2009 semester.**

SELECT ID, grade FROM takes

WHERE semester = 'Spring' AND year = 2009;

1. **Find the building(s) where the student takes the course titled "Image Processing".**

SELECT DISTINCT s.building

FROM section s JOIN takes t ON s.course\_id = t.course\_id AND s.sec\_id = t.sec\_id

AND s.semester = t.semester AND s.year = t.year

JOIN course c ON s.course\_id = c.course\_id

WHERE c.title = 'Image Processing';

1. **Find the room no. and the building where the student from Fall-2009 semester can take a course(s).**

SELECT DISTINCT building, room\_number

FROM section

WHERE semester = 'Fall' AND year = 2009;

1. **Find the names of those departments whose budget is higher than that of Astronomy. List them in alphabetic order.**

SELECT d1.dept\_name

FROM department d1, department d2

WHERE d2.dept\_name = 'Astronomy' AND d1.budget > d2.budget

ORDER BY d1.dept\_name;

1. **Display a list of all instructors, showing each instructor's ID and the number of sections taught. Make sure to show the number of sections as 0 for instructors who have not taught any section.**

SELECT i.ID, COUNT(t.course\_id) as num\_sections

FROM instructor i LEFT JOIN teaches t ON i.ID = t.ID

GROUP BY i.ID;

1. **For each student who has retaken a course at least twice (i.e., the student has taken the course at least three times), show the course ID and the student's ID. Please display your results in order of course ID and do not display duplicate rows.**

SELECT DISTINCT t1.ID, t1.course\_id

FROM takes t1

WHERE (SELECT COUNT(\*) FROM takes t2

WHERE t2.ID = t1.ID AND t2.course\_id = t1.course\_id) >= 3

ORDER BY t1.course\_id;

1. **Find the names of Biology students who have taken at least 3 Accounting courses.**

SELECT s.name

FROM student s

WHERE s.dept\_name = 'Biology' AND (

SELECT COUNT(DISTINCT t.course\_id)

FROM takes t JOIN course c ON t.course\_id = c.course\_id

WHERE t.ID = s.ID AND c.dept\_name = 'Accounting'

) >= 3;

1. **Find the sections that had maximum enrollment in Fall 2010.**

WITH enrollments AS (

SELECT course\_id, sec\_id, COUNT(ID) as num\_students

FROM takes

WHERE semester = 'Fall' AND year = 2010

GROUP BY course\_id, sec\_id

)

SELECT course\_id, sec\_id, num\_students

FROM enrollments

WHERE num\_students = (SELECT MAX(num\_students) FROM enrollments);

1. **Find student names and the number of law courses taken for students who have taken at least half of the available law courses. (These courses are named things like 'Tort Law' or 'Environmental Law').**

WITH law\_courses AS (

SELECT course\_id FROM course

WHERE title LIKE '%Law%'

),

student\_law\_counts AS (

SELECT t.ID, COUNT(DISTINCT t.course\_id) as num\_law\_courses

FROM takes t JOIN law\_courses l ON t.course\_id = l.course\_id

GROUP BY t.ID

),

total\_law\_courses AS (

SELECT COUNT(\*) as total FROM law\_courses

)

SELECT s.name, slc.num\_law\_courses

FROM student s JOIN student\_law\_counts slc ON s.ID = slc.ID

CROSS JOIN total\_law\_courses tlc

WHERE slc.num\_law\_courses >= tlc.total / 2;

1. **Find the rank and name of the 10 students who earned the most A grades (A-, A, A+). Use alphabetical order by name to break ties.**

SELECT RANK() OVER (ORDER BY COUNT(\*) DESC, s.name) as rank, s.name, COUNT(\*) as num\_a\_grades

FROM student s JOIN takes t ON s.ID = t.ID

WHERE t.grade IN ('A', 'A-', 'A+')

GROUP BY s.ID, s.name

ORDER BY num\_a\_grades DESC, s.name

LIMIT 10;

1. **Find the titles of courses in the Comp. Sci. department that have 3 credits.**

SELECT title FROM course

WHERE dept\_name = 'Comp. Sci.' AND credits = 3;

1. **Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.**

SELECT DISTINCT t.ID

FROM takes t JOIN teaches te ON t.course\_id = te.course\_id

AND t.sec\_id = te.sec\_id AND t.semester = te.semester AND t.year = te.year

JOIN instructor i ON te.ID = i.ID

WHERE i.name = 'Einstein';

1. **Find the ID and name of each student who has taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.**

SELECT DISTINCT s.ID, s.name

FROM student s JOIN takes t ON s.ID = t.ID

JOIN course c ON t.course\_id = c.course\_id

WHERE c.dept\_name = 'Comp. Sci.';

1. **Find the course id, section id, and building for each section of a Biology course.**

SELECT s.course\_id, s.sec\_id, s.building

FROM section s JOIN course c ON s.course\_id = c.course\_id

WHERE c.dept\_name = 'Biology';

1. **Output instructor names sorted by the ratio of their salary to their department's budget (in ascending order).**

SELECT i.name, i.salary/d.budget as ratio

FROM instructor i JOIN department d ON i.dept\_name = d.dept\_name

ORDER BY ratio ASC;

1. **Output instructor names and buildings for each building an instructor has taught in. Include instructor names who have not taught any classes (the building name should be NULL in this case).**

SELECT DISTINCT i.name, s.building

FROM instructor i LEFT JOIN teaches t ON i.ID = t.ID

LEFT JOIN section s ON t.course\_id = s.course\_id AND t.sec\_id = s.sec\_id

AND t.semester = s.semester AND t.year = s.year;