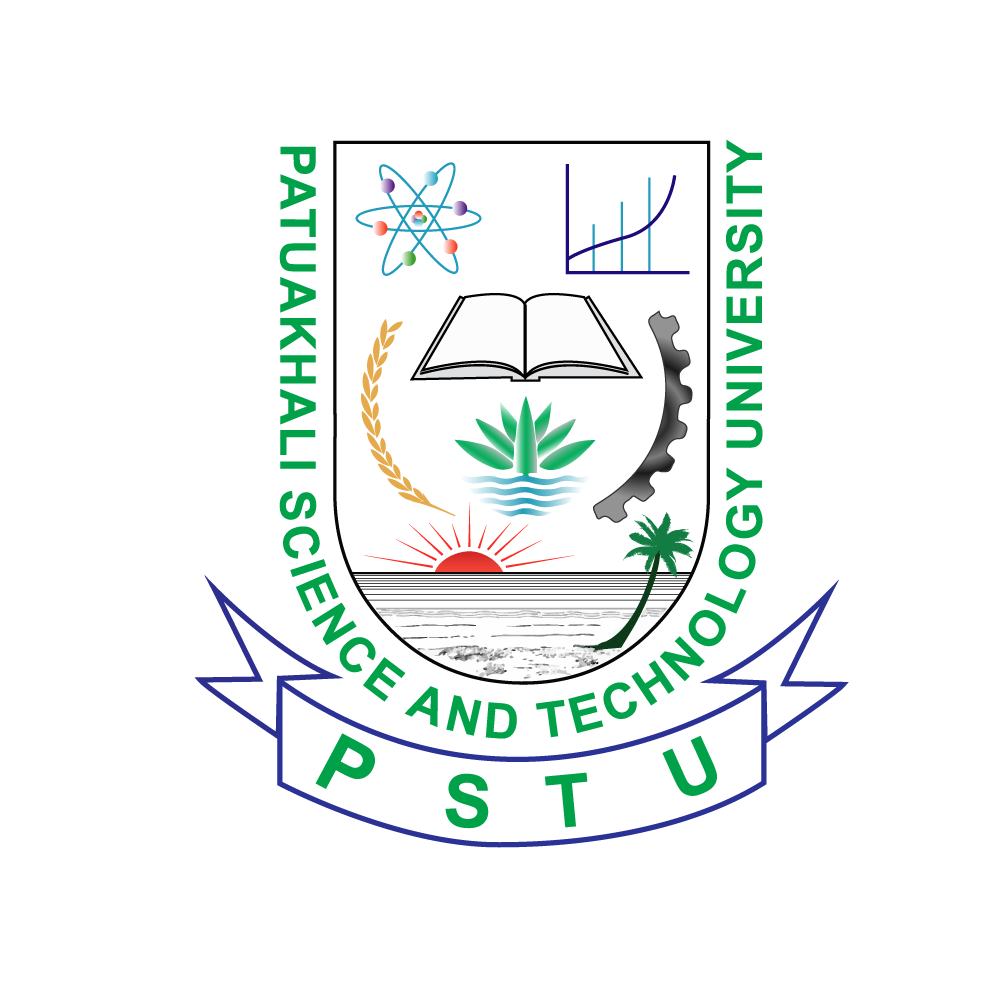
****

**Course Title :** Database System Sessional

**Course Code : CCE -** 224

**Lab Problem : 04**

Patuakhali Science & Technology University

Dumki, Patuakhali - 8602

**Submitted By**

**Name :**

**Id :**

**Reg. No :**

**Session :**

Md Mohidul Alam

1902016

08722

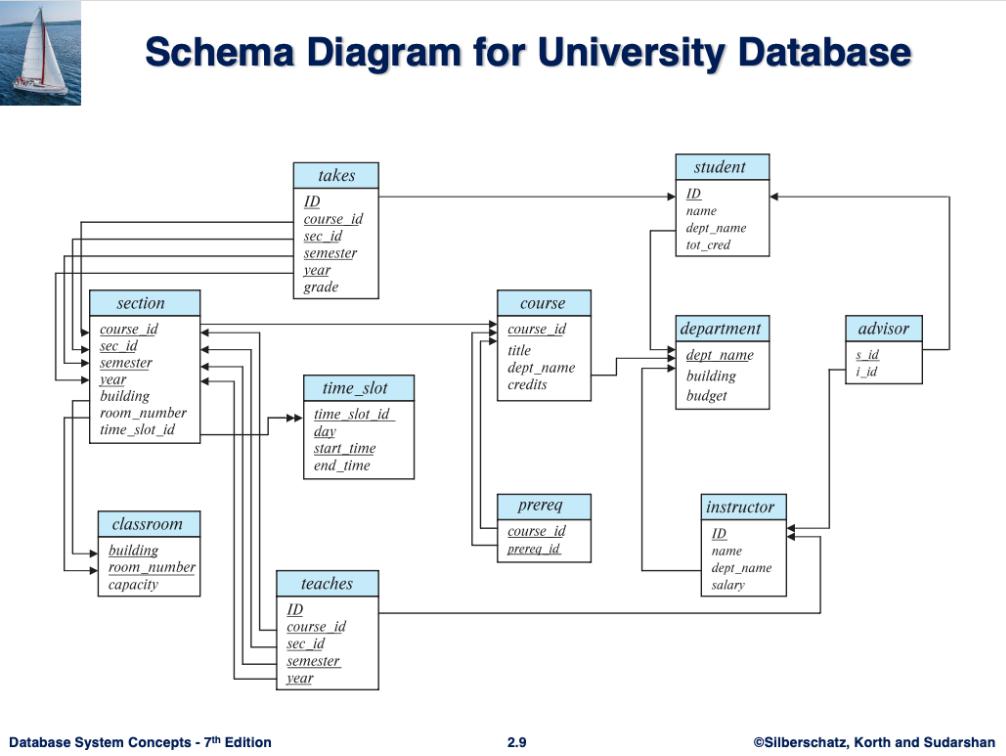
2019 - 20

**Submitted To**

**Professor** **Dr. Md. Samsuzzaman**

Department of Computer and Communication Engineering

Faculty Of Computer Science & Engineering.



**01\_Find the titles of courses in the Comp. Sci. department that have 3 credits.**

**Answer :**

SELECT title FROM course

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

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

**Answer :**

SELECT DISTINCT takes.id FROM takes

JOIN section ON takes.course\_id = section.course\_id AND takes.sec\_id = section.sec\_id AND takes.semester = section.semester AND takes.year = section.year

JOIN teaches ON section.course\_id = teaches.course\_id AND section.sec\_id = teaches.sec\_id AND section.semester = teaches.semester AND section.year = teaches.year

JOIN instructor ON teaches.id = instructor.id AND instructor.name = 'Einstein';

**3\_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**.

**Answer :**

SELECT DISTINCT student.ID, student.name FROM student

JOIN takes ON student.ID = takes.ID

JOIN course ON takes.course\_id = course.course\_id AND course.dept\_name = 'Comp. Sci.';

**4\_Find the course id, section id, and building for each section of a Biology course.**

**Answer :**

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

FROM section

JOIN course ON section.course\_id = course.course\_id AND course.dept\_name = 'Biology';

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

**Answer :**

SELECT instructor.name, (instructor.salary / department.budget) AS salary\_budget\_ratio

FROM instructor

JOIN department ON instructor.dept\_name = department.dept\_name

ORDER BY salary\_budget\_ratio ASC;

**6\_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).**

**Answer :**

SELECT instructor.name, section.building

FROM instructor

LEFT JOIN teaches ON instructor.ID = teaches.ID

LEFT JOIN section ON teaches.course\_id = section.course\_id AND teaches.sec\_id = section.sec\_id

ORDER BY instructor.name ASC;

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

**Answer :**

SELECT dept\_name

FROM department

WHERE budget > (SELECT budget FROM department WHERE dept\_name = 'Astronomy')

ORDER BY dept\_name ASC;

8\_**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).**

**Answer :**

SELECT instructor.name, section.building

FROM instructor

LEFT JOIN teaches ON instructor.ID = teaches.ID

LEFT JOIN section ON teaches.course\_id = section.course\_id AND teaches.sec\_id = section.sec\_id

ORDER BY instructor.name ASC;

**9\_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.**

**Answer :**

SELECT DISTINCT takes.course\_id, takes.id

FROM takes

INNER JOIN (

SELECT takes.course\_id, takes.id, COUNT(\*) as num\_takes

FROM takes

GROUP BY takes.course\_id, takes.id

HAVING COUNT(\*) >= 3

) AS retakes ON takes.course\_id = retakes.course\_id AND takes.id = retakes.id

ORDER BY takes.course\_id ASC;

**10\_Find the names of Biology students who have taken at least 3 Accounting**

**Answer :**

SELECT DISTINCT s.name

FROM student AS s

INNER JOIN takes AS t ON s.id = t.id

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

WHERE s.dept\_name = 'Biology' AND c.dept\_name = 'Accounting'

GROUP BY s.id

HAVING COUNT(DISTINCT c.course\_id) >= 3;

**Answer 11:**

SELECT RANK() OVER (ORDER BY num\_A\_grades DESC, name ASC) AS rank, name

FROM (

SELECT s.name, COUNT(\*) AS num\_A\_grades

FROM student AS s

INNER JOIN takes AS t ON s.id = t.id

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

GROUP BY s.id

ORDER BY num\_A\_grades DESC, s.name ASC

)

LIMIT 10;