

Course Title : Database System Sessional
Course Code : CCE - 224

Lab Problem : 04

Submitted To

Professor Dr. Md. Samsuzzaman

Department of Computer and Communication Engineering
Faculty Of Computer Science & Engineering.

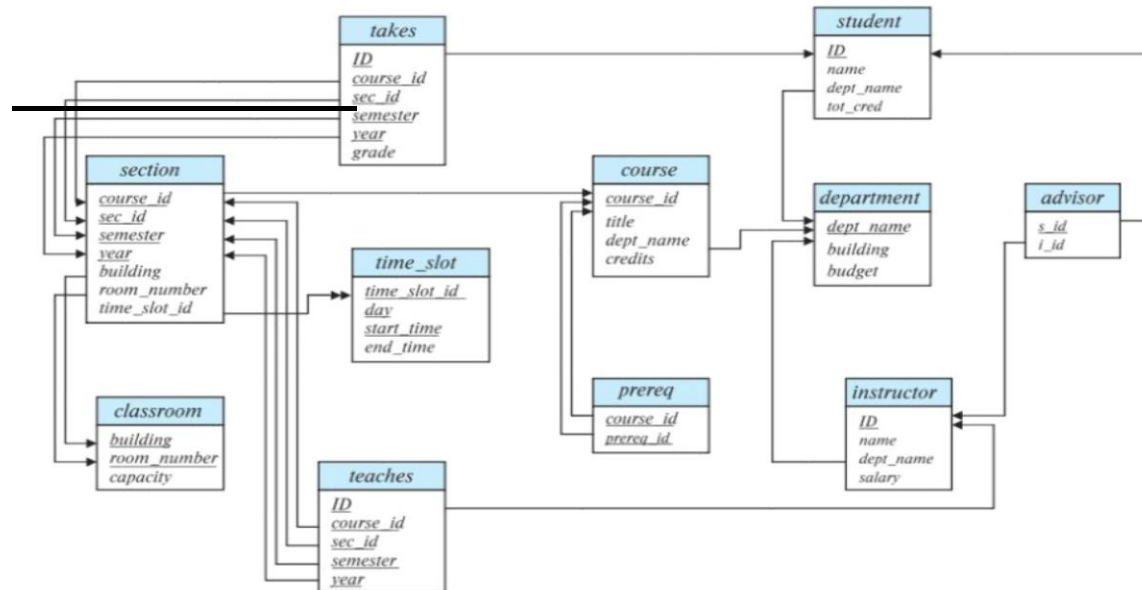
Submitted By

Name : Md Mohidul Alam
Id : 1902016
Reg. No : 08722
Session : 2019 - 20

Patuakhali Science & Technology University
Dumki, Patuakhali - 8602



Schema Diagram for University Database



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;
```