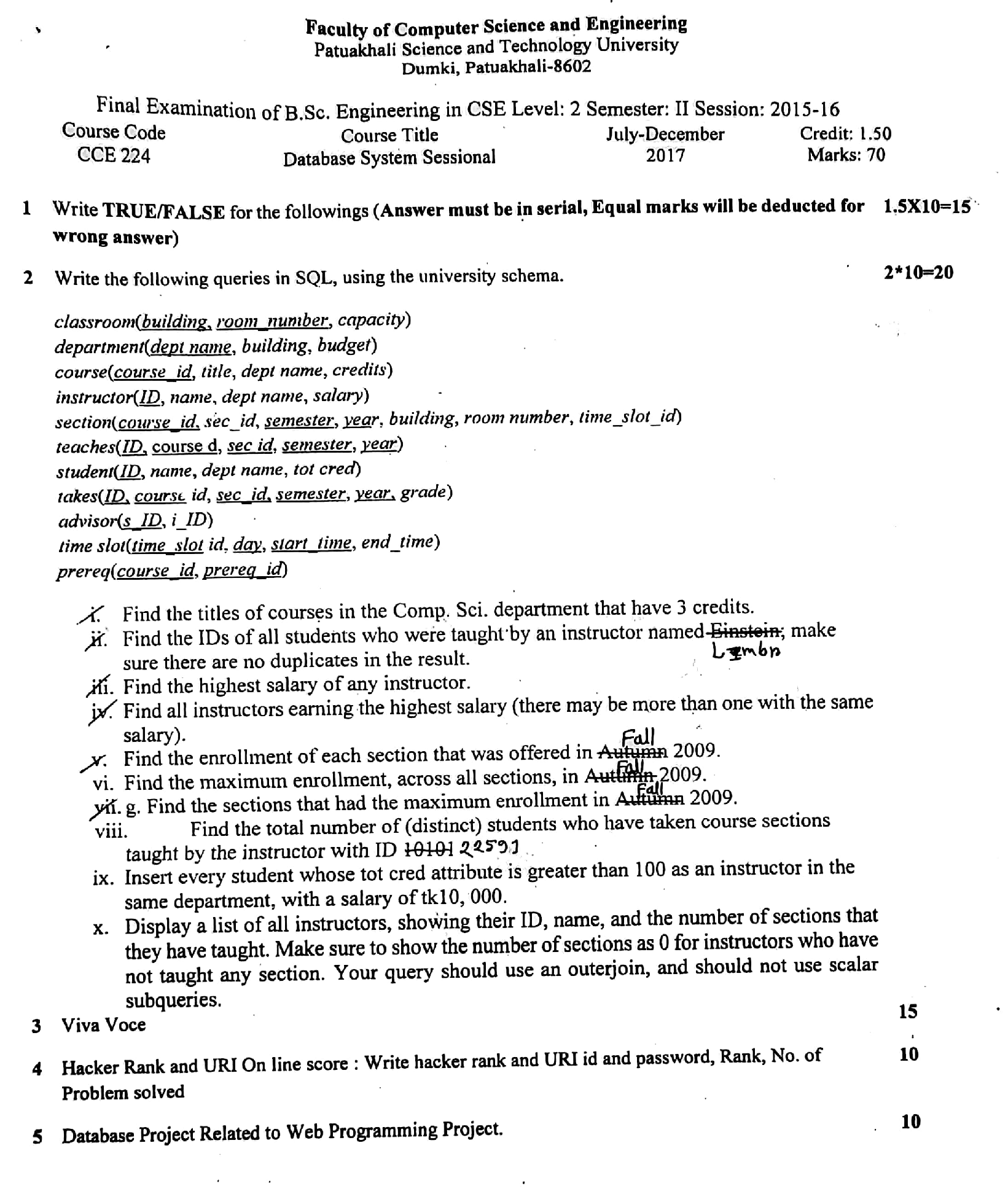
Database System Sessional (CCE 224) - SQL Queries



Schema reminder:

* classroom(building, room\_number, capacity)
* department(dept\_name, building, budget)
* course(course\_id, title, dept\_name, credits)
* instructor(ID, name, dept\_name, salary)
* section(course\_id, sec\_id, semester, year, building, room\_number, time\_slot\_id)
* teaches(ID, course\_id, sec\_id, semester, year)
* student(ID, name, dept\_name, tot\_cred)
* takes(ID, course\_id, sec\_id, semester, year, grade)
* advisor(s\_ID, i\_ID)
* time\_slot(time\_slot\_id, day, start\_time, end\_time)
* prereq(course\_id, prereq\_id)

# Questions and Answers

## 1. Titles of courses in the Comp. Sci. department that have 3 credits:

SELECT title  
FROM course  
WHERE dept\_name = 'Comp. Sci.' AND credits = 3;

## 2. IDs of all students who were taught by an instructor named 'Einstein' (no duplicates):

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';

## 3. The highest salary of any instructor:

SELECT MAX(salary) AS highest\_salary  
FROM instructor;

## 4. All instructors earning the highest salary:

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

## 5. The enrollment of each section that was offered in Autumn 2009:

SELECT course\_id, sec\_id, COUNT(ID) AS enrollment  
FROM takes  
WHERE semester = 'Autumn' AND year = 2009  
GROUP BY course\_id, sec\_id;

## 6. Maximum enrollment, across all sections, in Autumn 2009:

SELECT MAX(student\_count) AS max\_enrollment  
FROM (  
 SELECT COUNT(ID) AS student\_count  
 FROM takes  
 WHERE semester = 'Autumn' AND year = 2009  
 GROUP BY course\_id, sec\_id  
) AS sub;

## 7. Sections that had the maximum enrollment in Autumn 2009:

SELECT course\_id, sec\_id  
FROM takes  
WHERE semester = 'Autumn' AND year = 2009  
GROUP BY course\_id, sec\_id  
HAVING COUNT(ID) = (  
 SELECT MAX(student\_count)  
 FROM (  
 SELECT COUNT(ID) AS student\_count  
 FROM takes  
 WHERE semester = 'Autumn' AND year = 2009  
 GROUP BY course\_id, sec\_id  
 ) AS sub  
);

## 8. Total number of distinct students taught by the instructor with ID '10101':

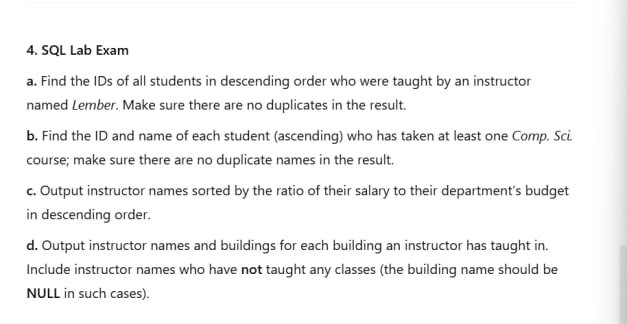
SELECT COUNT(DISTINCT t.ID) AS total\_students  
FROM takes t  
JOIN teaches te ON t.course\_id = te.course\_id AND t.sec\_id = te.sec\_id   
 AND t.semester = te.semester AND t.year = te.year  
WHERE te.ID = '10101';

## 9. Insert students with tot\_cred > 100 as instructors with salary 10000:

INSERT INTO instructor(ID, name, dept\_name, salary)  
SELECT ID, name, dept\_name, 10000  
FROM student  
WHERE tot\_cred > 100;

## 10. All instructors with their ID, name, and number of sections taught (including 0):

SELECT i.ID, i.name, COUNT(te.course\_id) AS num\_sections  
FROM instructor i  
LEFT JOIN teaches te ON i.ID = te.ID  
GROUP BY i.ID, i.name;



## 1. IDs of students in descending order who were taught by instructor named 'Lember':

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 = 'Lember'  
ORDER BY t.ID DESC;

## 2. ID and name of each student (ascending) who has taken at least one Comp. Sci. course:

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.'  
ORDER BY s.ID ASC;

## 3. Instructor names sorted by the ratio of their salary to department's budget (descending):

SELECT i.name  
FROM instructor i  
JOIN department d ON i.dept\_name = d.dept\_name  
ORDER BY (i.salary \* 1.0) / d.budget DESC;

## 4. Instructor names and buildings for each building they have taught in (NULL if none):

SELECT i.name, b.building  
FROM instructor i  
LEFT JOIN teaches te ON i.ID = te.ID  
LEFT 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  
LEFT JOIN classroom b ON s.building = b.building  
GROUP BY i.name, b.building;