CSE-3104 (Database Lab)
SQL Query Exercise

1. Find out the ID and salary of the instructors.
2. Find out the ID and salary of the instructor who gets more than $85,000.
3. Find out the department names and their budget at the university.
4. List out the names of the instructors from Computer Science who have more than $70,000.
5. For all instructors in the university who have taught some course, find their names and the course ID of all courses they taught.
6. Find the names of all instructors whose salary is greater than at least one instructor in the Biology department.
7. Find the advisor of the student with ID 12345
8. Find the average salary of all instructors.
9. Find the names of all departments whose building name includes the substring 'Watson'.
10. Find the names of instructors with salary amounts between $90,000 and $100,000.
11. Find the instructor names and the courses they taught for all instructors in the Biology department who have taught some course.
12. Find the courses taught in Fall-2009 semester.
13. Find the set of all courses taught either in Fall-2009 or in Spring-2010.
14. Find the set of all courses taught in the Fall-2009 as well as in Spring-2010.
15. Find all courses taught in the Fall-2009 semester but not in the Spring-2010 semester.
16. Find all instructors who appear in the *instructor* relation with null values for *salary*.
17. Find the average salary of instructors in the Finance department.
18. Find the total number of instructors who teach a course in the Spring-2010 semester.
19. Find the average salary in each department.
20. Find the number of instructors in each department who teach a course in the Spring-2010 semester.
21. List out the departments where the average salary of the instructors is more than $42,000.
22. 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.
23. Find all the courses taught in both the Fall-2009 and Spring-2010 semesters.
24. Find all the courses taught in the Fall-2009 semester but not in the Spring-2010 semester.
25. Select the names of instructors whose names are neither "Mozart" nor "Einstein".
26. Find the total number of (distinct) students who have taken course sections taught by the instructor with *ID* 110011.
27. Find the ID and names of all instructors whose salary is greater than at least one instructor in the History department.
28. Find the names of all instructors that have a salary value greater than that of each instructor in the Biology department.
29. Find the departments that have the highest average salary.
30. Find all courses taught in both the Fall 2009 semester and in the Spring-2010 semester.
31. Find all students who have taken all the courses offered in the Biology department.
32. Find all courses that were offered at most once in 2009.
33. Find all courses that were offered at least twice in 2009.
34. Find the average instructors' salaries of those departments where the average salary is greater than $42,000.
35. Find the maximum across all departments of the total salary at each department.
36. List all departments along with the number of instructors in each department.

37. Find the titles of courses in the Comp. Sci. department that has 3 credits.
38. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.
39. Find the highest salary of any instructor.
40. Find all instructors earning the highest salary (there may be more than one with the same salary).
41. Find the enrollment of each section that was offered in Autumn-2009.
42. Find the maximum enrollment, across all sections, in Autumn-2009.
43. Find the salaries after the following operation: Increase the salary of each instructor in the Comp. Sci. department by 10%.
44. Find all students who have not taken a course.
45. List all course sections offered by the Physics department in the Fall-2009 semester, with the building and room number of each section.
46. Find the student names who take courses in Spring-2010 semester at Watson Building.
47. List the students who take courses teaches by 'Brandt'.
48. Find out the average salary of the instructor in each department.
49. Find the number of students who take the course titled 'Intro. To Computer Science'.
50. Find out the total salary of the instructors of the Computer Science department who take a course(s) in Watson building.
51. Find out the course titles which starts between 10:00 to 12:00.
52. List the course names where 'CS-101' is the pre-requisite course.
53. List the student names who get more than B+ grades in their respective courses.
54. Find the student who takes the maximum credit from each department.
55. Find out the student ID and grades who take a course(s) in Spring-2009 semester.
56. Find the building(s) where the student takes the course titled 'Image Processing'.
57. Find the room no. and the building where the student from Fall-2009 semester can take a course(s)

CSE-3104 (Database Lab)
Answers of the SQL Queries

```
1.  select id, salary from instructor
2.  select id, salary from instructor where salary > 85000
3.  select dept_name, budget from department
4.  select name from instructor where salary > 70000 and dept_name = ‘comp. sci.’
5.  select name, course_id from instructor, teaches where instructor.id= teaches.id;
6.  select distinct t.name from instructor as t, instructor as s
    where t.salary > s.salary and s.dept_name = ’biology’;
7.  select instructor.name from instructor
    inner join advisor on instructor.id = advisor.i_id
    inner join student on advisor.s_id = student.id
    where student.id = ‘12345’
8.  select avg (salary) as avg_salary from instructor
9.  select dept_name from department where building like ’%watson%’;
10. select name from instructor where salary between 90000 and 100000;
11. select name, course_id from instructor, teaches
    where instructor.id= teaches.id and dept_name = ’biology’;
12. select course.title from course
    inner join section on section.course_id = course.course_id
    where section.semester = ‘fall’ and section.year = 2009
13. (select course_id from section where semester = ’fall’ and year= 2009)
    union
    (select course_id from section where semester = ’spring’ and year= 2010);
14. (select course_id from section where semester = ’fall’ and year= 2009)
    intersect
    (select course_id from section where semester = ’spring’ and year= 2010);
15. (select course_id from section where semester = ’fall’ and year= 2009)
    except
    (select course_id from section where semester = ’spring’ and year= 2010);
16. select name from instructor where salary is null;
17. select avg (salary) as avg_salary from instructor where dept_name= ’comp. sci.’;
18. select count (distinct id) from teaches where semester = ’spring’ and year = 2010;
19. select dept_name, avg (salary) as avg_salary from instructor group by dept_name;
20. select dept_name, count (distinct instructor.id) as instr_count
    from instructor inner join teaches on instructor.ID = teaches.ID
    where semester = ’spring’ and year = 2010 group by dept_name;
21. select dept_name, avg (salary) as avg_salar from instructor
    group by dept_name having avg (salary) > 42000;
22. select course_id, semester, year, sec_id, avg (tot_cred)
    from takes inner join student on takes.ID = student.ID
    where year = 2009 group by course_id, semester, year, sec_id having count (student.id) >= 2;
23. select course_id from section where semester = ’spring’ and year= 2010
24. (select course_id from section where semester = ’fall’ and year= 2009)
    except
    (select course_id from section where semester = ’spring’ and year= 2010);
```

```
25. select distinct name from instructor where name not in (’mozart’, ’einstein’);
26. select count (distinct id) from takes
    where (course_id, sec_id, semester, year) in (select course_id, sec_id, semester, year
    from teaches where teaches.id= 10101);
27. select distinct t.id, t.name from instructor as t, instructor as s
    where t.salary > s.salary and s.dept_name = ’history’;
28. select name from instructor where salary > all (select salary from instructor
    where dept_name = ’biology’);
29. select dept_name from instructor group by dept_name
    having avg (salary) >= all (select avg (salary) from instructor group by dept_name);
30. select course_id from section as s where semester = ’fall’ and year= 2009 and exists (select * from
    section as t where semester = ’spring’ and year= 2010 and s.course_id= t.course_id);
31. select distinct s.id, s.name from student as s where not exists ((select course_id
    from course where dept_name = ’biology’) except (select t.course_id from takes as t where s.id =
    t.id));
32. select t.course_id from course as t where unique (select r.course_id from section as r where
    t.course_id= r.course_id and r.year = 2009);
33. select t.course_id from course as t where not unique (select r.course_id from section as r
    where t.course_id= r.course_id and r.year = 2009);
34. select dept_name, avg salary from (select dept_name, avg (salary) as avg salary from instructor group
    by dept_name) where avg salary > 42000;
35. select max (tot salary) from (select dept_name, sum(salary) from instructor group by dept_name) as
    dept total (dept_name, tot salary);
36. select dept_name, (select count(*) from instructor where department.dept_name =
    instructor.dept_name) as num_instructors from department;
37. select title from course where dept_name = 'comp. sci.' and credits = 3
38. select distinct student.id from (student join takes using(id)) join (instructor join teaches using(id))
    using(course_id, sec_id, semester, year) where instructor.name = 'einstein'
39. select max(salary) from instructor
40. select id, name from instructor where salary = (select max(salary) from instructor)
41. select section.course_id, section.sec_id, count(id) from section
    inner join takes on takes.course_id = section.course_id
    where section.semester = 'autumn' and section.year = 2009
    group by section.course_id, section.sec_id
42. select max(enrollment) from (select count(id) as enrollment from section
    natural join takes where semester = 'autumn' and year = 2009
    group by course_id, sec_id)
43. select ID, salary + salary * 1.10 from instructor where dept_name = 'comp. sci.'
44. select ID from student natural left outer join takes where course_id is null;
45. select course.course_id, sec_id, building, room_number from course
    inner join section on course.course_id = section.course_id
    where course.dept_name = ’physics’ and section.semester = ’fall’ and section.year = ’2009’;
46. select student.id, student.name from student
    inner join takes on student.id = takes.id
    inner join section on takes.course_id = section.course_id
    where building = ‘watson’
47. select student.id, student.name from student
    inner join takes on student.id = takes.id
    inner join section on takes.course_id = section.course_id
    inner join teaches on section.course_id = teaches.course_id
    inner join instructor on teaches.id = instructor.id where instructor.name = ‘brandt’
48. select avg(salary) as avg_salary from instructor group by dept_name
49. select count(takes.ID) as total_stud from takes
    inner join section on section.course_id = takes.course_id
    inner join course on course.course_id = section.course_id
    where title = ‘intro. to comp. sci.’
50. select sum(salary) from instructor
    inner join department on instructor.dept_name = department.dept_name
    where department.building = ‘watson’ and department.dept_name = ‘comp. sci’
51. select course.title from course
    inner join section on course.course_id = section.course_id and time_slot_id in (select time_slot_id from
    timeslot where start_time between ‘10:00’ and ‘12:00’)
52. select course.title from course
    inner join prereq on course.course_id = prereq.course_id
    where prereq.prereq_id = ‘CS-101’
53. select student.id, student.name from student
    inner join takes on student.id = takes.id
    where takes.grade = ‘a-‘ or takes.grade = ‘a’ or takes.grade = ‘a+’
54. select max(tot_cred) as max_credit from student group by dept_name
55. select student.id, student.name from student
    inner join takes on student.id = takes.id
    where takes.semester = ‘spring’ and takes.year = ‘2009’
56. select building from classroom
    inner join section on classroom.building = section.building
    inner join course on course.course_id = section.course_id
    where course.title = ‘image processing’
57. select classroom.building, classroom.room_number from classroom
    inner join section on classroom.building = section.building
    where section.semester = ‘fall’ and section.year = ‘2009;
```

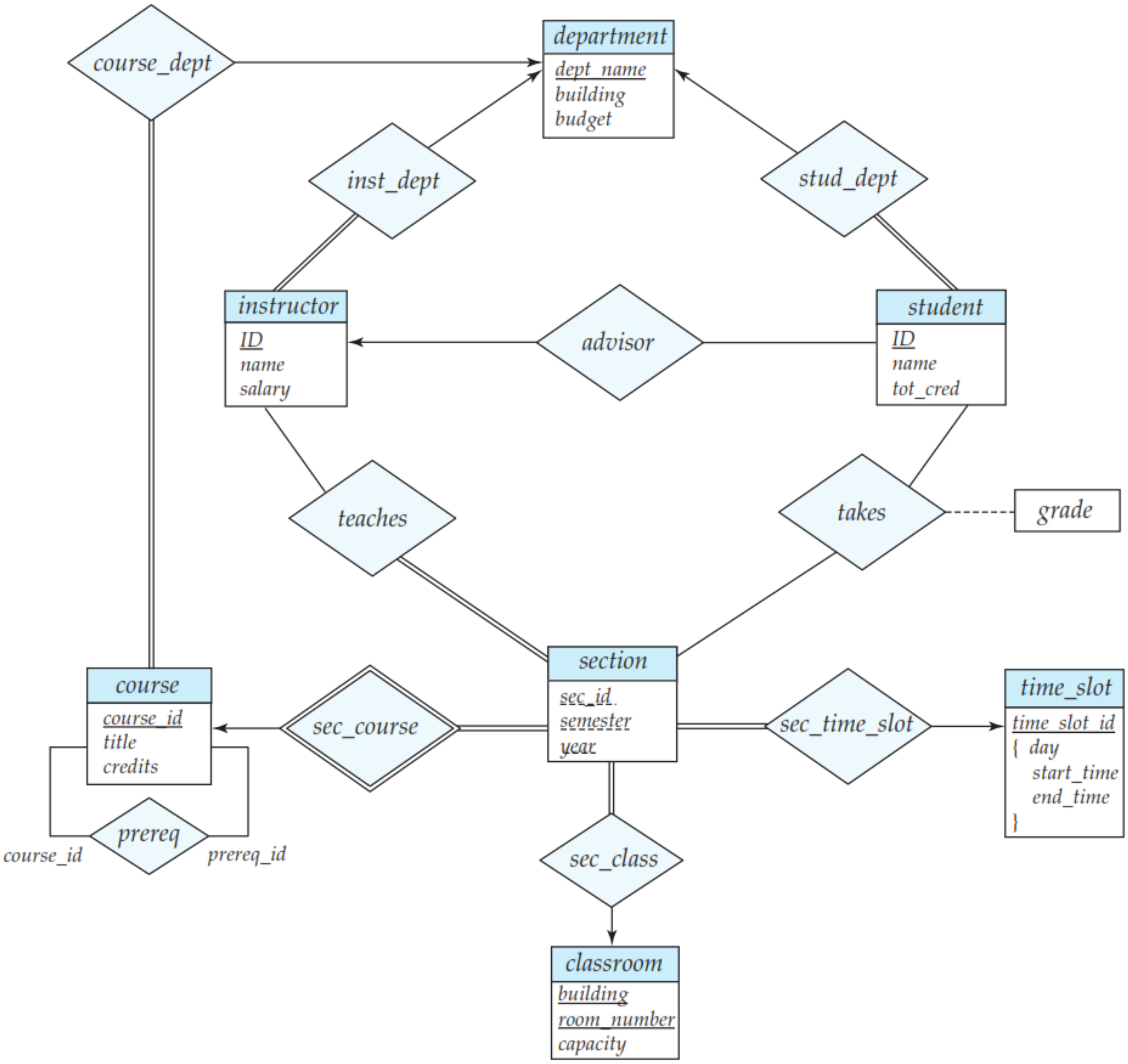


Fig-1: E-R diagram of University System

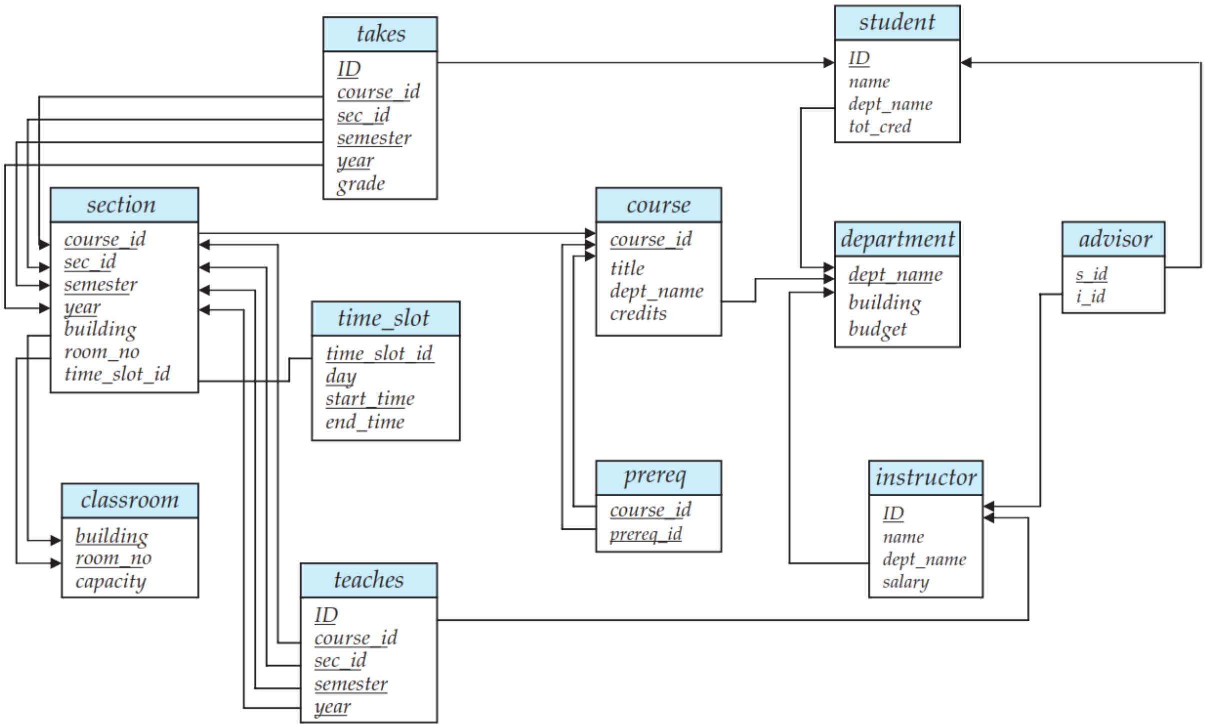
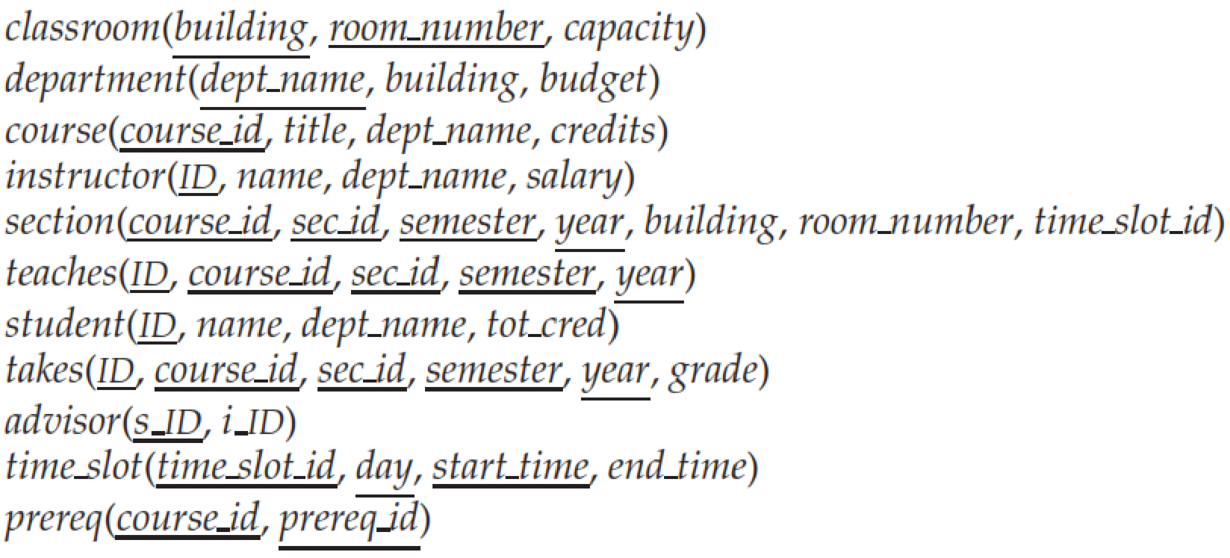
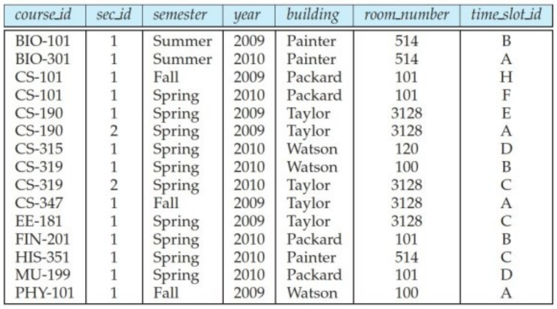


Fig-2: Schema diagram of University System



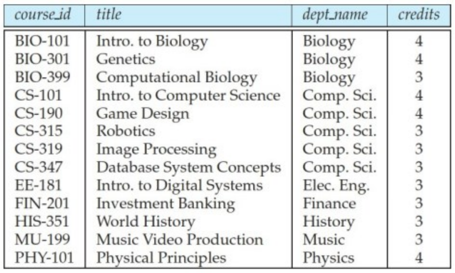
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, 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)

Fig-3: Full Schema of University System



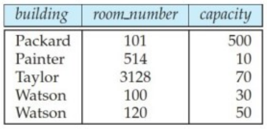
course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Figure A.7 The *section* relation.



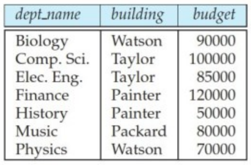
course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

Figure A.5 The *course* relation.



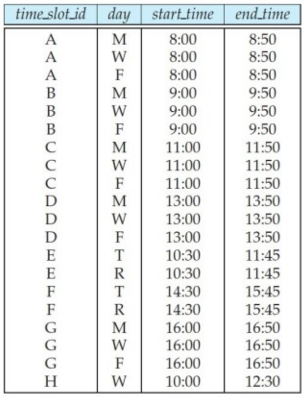
building	room_number	capacity
Packard	101	500
Painter	514	10
Taylor	3128	70
Watson	100	30
Watson	120	50

Figure A.3 The *classroom* relation.



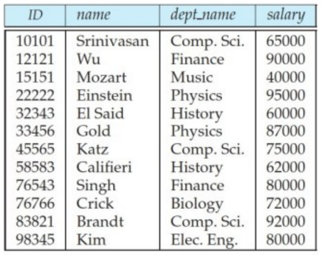
dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Figure A.4 The *department* relation.



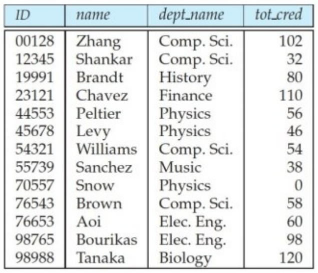
time_slot_id	day	start_time	end_time
A	M	8:00	8:50
A	W	8:00	8:50
A	F	8:00	8:50
B	M	9:00	9:50
B	W	9:00	9:50
B	F	9:00	9:50
C	M	11:00	11:50
C	W	11:00	11:50
C	F	11:00	11:50
D	M	13:00	13:50
D	W	13:00	13:50
D	F	13:00	13:50
E	T	10:30	11:45
E	R	10:30	11:45
F	T	14:30	15:45
F	R	14:30	15:45
G	M	16:00	16:50
G	W	16:00	16:50
G	F	16:00	16:50
H	W	10:00	12:30

Figure A.12 The *time_slot* relation.



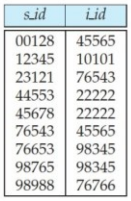
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure A.6 The *instructor* relation.



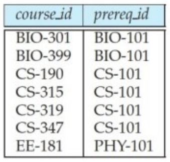
ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

Figure A.9 The *student* relation.



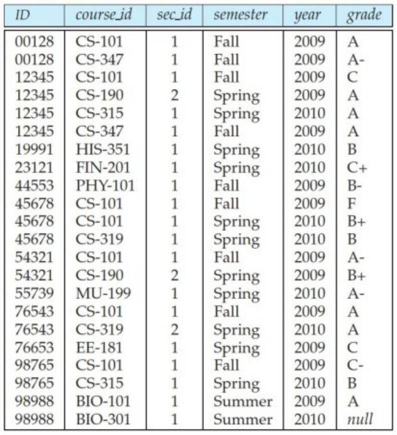
s_id	i_id
00128	45565
12345	10101
23121	76543
44553	22222
45678	22222
76543	45565
76653	98345
98765	98345
98988	76766

Figure A.11 The *advisor* relation.



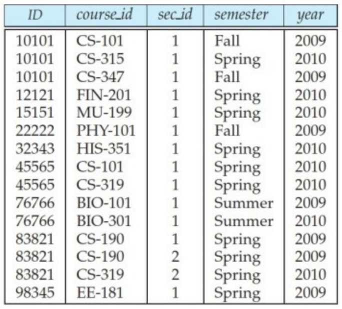
course_id	prereq_id
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

Figure A.13 The *prereq* relation.



ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	*null*

Figure A.10 The *takes* relation.



ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Figure A.8 The *teaches* relation.