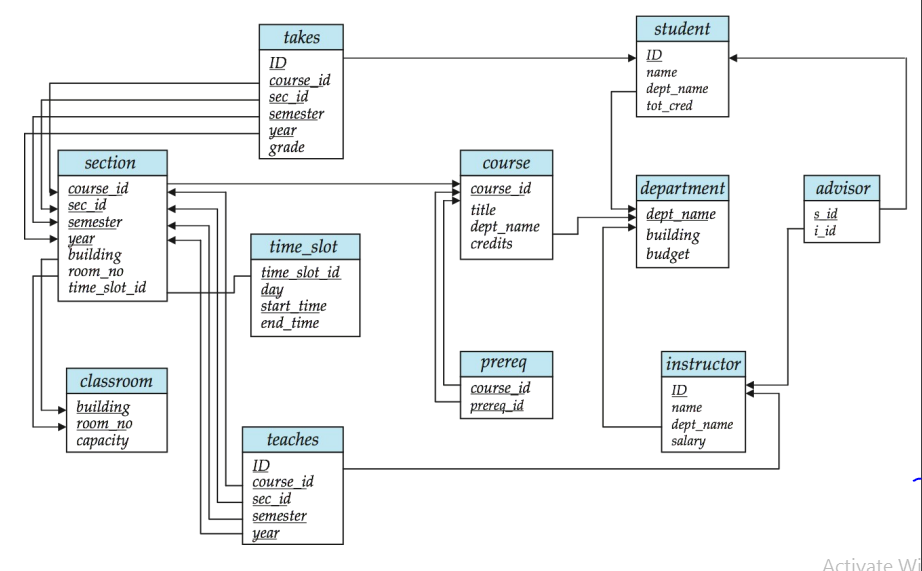
Experiment-5 University DataBase



create table classroom

(building varchar(15),

room\_number varchar(7),

capacity numeric(4,0),

primary key (building, room\_number)

);

create table department

(dept\_name varchar(20),

building varchar(15),

budget numeric(12,2) check (budget > 0),

primary key (dept\_name)

);

create table course

(course\_id varchar(8),

title varchar(50),

dept\_name varchar(20),

credits numeric(2,0) check (credits > 0),

primary key (course\_id),

foreign key (dept\_name) references department

on delete set null

);

create table instructor

(ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

salary numeric(8,2) check (salary > 29000),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

create table section

(course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6)

check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),

year numeric(4,0) check (year > 1701 and year < 2100),

building varchar(15),

room\_number varchar(7),

time\_slot\_id varchar(4),

primary key (course\_id, sec\_id, semester, year),

foreign key (course\_id) references course

on delete cascade,

foreign key (building, room\_number) references classroom

on delete set null

);

create table teaches

(ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references instructor

on delete cascade

);

create table student

(

ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

tot\_cred numeric(3,0) check (tot\_cred >= 0),

primary key (ID),

foreign key (dept\_name) references department

on delete set null

);

create table takes

(

ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

grade varchar(2),

primary key (ID, course\_id, sec\_id, semester, year),

foreign key (course\_id,sec\_id, semester, year) references section

on delete cascade,

foreign key (ID) references student

on delete cascade

);

create table advisor

(

s\_ID varchar(5),

i\_ID varchar(5),

primary key (s\_ID),

foreign key (i\_ID) references instructor (ID)

on delete set null,

foreign key (s\_ID) references student (ID)

on delete cascade

);

create table time\_slot

(

time\_slot\_id varchar(4),

day varchar(1),

start\_hr numeric(2) check (start\_hr >= 0 and start\_hr < 24),

start\_min numeric(2) check (start\_min >= 0 and start\_min < 60),

end\_hr numeric(2) check (end\_hr >= 0 and end\_hr < 24),

end\_min numeric(2) check (end\_min >= 0 and end\_min < 60),

primary key (time\_slot\_id, day, start\_hr, start\_min)

);

create table prereq

(

course\_id varchar(8),

prereq\_id varchar(8),

primary key (course\_id, prereq\_id),

foreign key (course\_id) references course

on delete cascade,

foreign key (prereq\_id) references course

);

delete from prereq;

delete from time\_slot;

delete from advisor;

delete from takes;

delete from student;

delete from teaches;

delete from section;

delete from instructor;

delete from course;

delete from department;

delete from classroom;

insert into classroom values ('Packard', '101', '500');

insert into classroom values ('Painter', '514', '10');

insert into classroom values ('Taylor', '3128', '70');

insert into classroom values ('Watson', '100', '30');

insert into classroom values ('Watson', '120', '50');

insert into department values ('Biology', 'Watson', '90000');

insert into department values ('Comp. Sci.', 'Taylor', '100000');

insert into department values ('Elec. Eng.', 'Taylor', '85000');

insert into department values ('Finance', 'Painter', '120000');

insert into department values ('History', 'Painter', '50000');

insert into department values ('Music', 'Packard', '80000');

insert into department values ('Physics', 'Watson', '70000');

insert into course values ('BIO-101', 'Intro. to Biology', 'Biology', '4');

insert into course values ('BIO-301', 'Genetics', 'Biology', '4');

insert into course values ('BIO-399', 'Computational Biology', 'Biology', '3');

insert into course values ('CS-101', 'Intro. to Computer Science', 'Comp. Sci.', '4');

insert into course values ('CS-190', 'Game Design', 'Comp. Sci.', '4');

insert into course values ('CS-315', 'Robotics', 'Comp. Sci.', '3');

insert into course values ('CS-319', 'Image Processing', 'Comp. Sci.', '3');

insert into course values ('CS-347', 'Database System Concepts', 'Comp. Sci.', '3');

insert into course values ('EE-181', 'Intro. to Digital Systems', 'Elec. Eng.', '3');

insert into course values ('FIN-201', 'Investment Banking', 'Finance', '3');

insert into course values ('HIS-351', 'World History', 'History', '3');

insert into course values ('MU-199', 'Music Video Production', 'Music', '3');

insert into course values ('PHY-101', 'Physical Principles', 'Physics', '4');

insert into instructor values ('10101', 'Srinivasan', 'Comp. Sci.', '65000');

insert into instructor values ('12121', 'Wu', 'Finance', '90000');

insert into instructor values ('15151', 'Mozart', 'Music', '40000');

insert into instructor values ('22222', 'Einstein', 'Physics', '95000');

insert into instructor values ('32343', 'El Said', 'History', '60000');

insert into instructor values ('33456', 'Gold', 'Physics', '87000');

insert into instructor values ('45565', 'Katz', 'Comp. Sci.', '75000');

insert into instructor values ('58583', 'Califieri', 'History', '62000');

insert into instructor values ('76543', 'Singh', 'Finance', '80000');

insert into instructor values ('76766', 'Crick', 'Biology', '72000');

insert into instructor values ('83821', 'Brandt', 'Comp. Sci.', '92000');

insert into instructor values ('98345', 'Kim', 'Elec. Eng.', '80000');

insert into section values ('BIO-101', '1', 'Summer', '2009', 'Painter', '514', 'B');

insert into section values ('BIO-301', '1', 'Summer', '2010', 'Painter', '514', 'A');

insert into section values ('CS-101', '1', 'Fall', '2009', 'Packard', '101', 'H');

insert into section values ('CS-101', '1', 'Spring', '2010', 'Packard', '101', 'F');

insert into section values ('CS-190', '1', 'Spring', '2009', 'Taylor', '3128', 'E');

insert into section values ('CS-190', '2', 'Spring', '2009', 'Taylor', '3128', 'A');

insert into section values ('CS-315', '1', 'Spring', '2010', 'Watson', '120', 'D');

insert into section values ('CS-319', '1', 'Spring', '2010', 'Watson', '100', 'B');

insert into section values ('CS-319', '2', 'Spring', '2010', 'Taylor', '3128', 'C');

insert into section values ('CS-347', '1', 'Fall', '2009', 'Taylor', '3128', 'A');

insert into section values ('EE-181', '1', 'Spring', '2009', 'Taylor', '3128', 'C');

insert into section values ('FIN-201', '1', 'Spring', '2010', 'Packard', '101', 'B');

insert into section values ('HIS-351', '1', 'Spring', '2010', 'Painter', '514', 'C');

insert into section values ('MU-199', '1', 'Spring', '2010', 'Packard', '101', 'D');

insert into section values ('PHY-101', '1', 'Fall', '2009', 'Watson', '100', 'A');

insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2009');

insert into teaches values ('10101', 'CS-315', '1', 'Spring', '2010');

insert into teaches values ('10101', 'CS-347', '1', 'Fall', '2009');

insert into teaches values ('12121', 'FIN-201', '1', 'Spring', '2010');

insert into teaches values ('15151', 'MU-199', '1', 'Spring', '2010');

insert into teaches values ('22222', 'PHY-101', '1', 'Fall', '2009');

insert into teaches values ('32343', 'HIS-351', '1', 'Spring', '2010');

insert into teaches values ('45565', 'CS-101', '1', 'Spring', '2010');

insert into teaches values ('45565', 'CS-319', '1', 'Spring', '2010');

insert into teaches values ('76766', 'BIO-101', '1', 'Summer', '2009');

insert into teaches values ('76766', 'BIO-301', '1', 'Summer', '2010');

insert into teaches values ('83821', 'CS-190', '1', 'Spring', '2009');

insert into teaches values ('83821', 'CS-190', '2', 'Spring', '2009');

insert into teaches values ('83821', 'CS-319', '2', 'Spring', '2010');

insert into teaches values ('98345', 'EE-181', '1', 'Spring', '2009');

insert into student values ('00128', 'Zhang', 'Comp. Sci.', '102');

insert into student values ('12345', 'Shankar', 'Comp. Sci.', '32');

insert into student values ('19991', 'Brandt', 'History', '80');

insert into student values ('23121', 'Chavez', 'Finance', '110');

insert into student values ('44553', 'Peltier', 'Physics', '56');

insert into student values ('45678', 'Levy', 'Physics', '46');

insert into student values ('54321', 'Williams', 'Comp. Sci.', '54');

insert into student values ('55739', 'Sanchez', 'Music', '38');

insert into student values ('70557', 'Snow', 'Physics', '0');

insert into student values ('76543', 'Brown', 'Comp. Sci.', '58');

insert into student values ('76653', 'Aoi', 'Elec. Eng.', '60');

insert into student values ('98765', 'Bourikas', 'Elec. Eng.', '98');

insert into student values ('98988', 'Tanaka', 'Biology', '120');

insert into takes values ('00128', 'CS-101', '1', 'Fall', '2009', 'A');

insert into takes values ('00128', 'CS-347', '1', 'Fall', '2009', 'A-');

insert into takes values ('12345', 'CS-101', '1', 'Fall', '2009', 'C');

insert into takes values ('12345', 'CS-190', '2', 'Spring', '2009', 'A');

insert into takes values ('12345', 'CS-315', '1', 'Spring', '2010', 'A');

insert into takes values ('12345', 'CS-347', '1', 'Fall', '2009', 'A');

insert into takes values ('19991', 'HIS-351', '1', 'Spring', '2010', 'B');

insert into takes values ('23121', 'FIN-201', '1', 'Spring', '2010', 'C+');

insert into takes values ('44553', 'PHY-101', '1', 'Fall', '2009', 'B-');

insert into takes values ('45678', 'CS-101', '1', 'Fall', '2009', 'F');

insert into takes values ('45678', 'CS-101', '1', 'Spring', '2010', 'B+');

insert into takes values ('45678', 'CS-319', '1', 'Spring', '2010', 'B');

insert into takes values ('54321', 'CS-101', '1', 'Fall', '2009', 'A-');

insert into takes values ('54321', 'CS-190', '2', 'Spring', '2009', 'B+');

insert into takes values ('55739', 'MU-199', '1', 'Spring', '2010', 'A-');

insert into takes values ('76543', 'CS-101', '1', 'Fall', '2009', 'A');

insert into takes values ('76543', 'CS-319', '2', 'Spring', '2010', 'A');

insert into takes values ('76653', 'EE-181', '1', 'Spring', '2009', 'C');

insert into takes values ('98765', 'CS-101', '1', 'Fall', '2009', 'C-');

insert into takes values ('98765', 'CS-315', '1', 'Spring', '2010', 'B');

insert into takes values ('98988', 'BIO-101', '1', 'Summer', '2009', 'A');

insert into takes values ('98988', 'BIO-301', '1', 'Summer', '2010', null);

insert into advisor values ('00128', '45565');

insert into advisor values ('12345', '10101');

insert into advisor values ('23121', '76543');

insert into advisor values ('44553', '22222');

insert into advisor values ('45678', '22222');

insert into advisor values ('76543', '45565');

insert into advisor values ('76653', '98345');

insert into advisor values ('98765', '98345');

insert into advisor values ('98988', '76766');

insert into time\_slot values ('A', 'M', '8', '0', '8', '50');

insert into time\_slot values ('A', 'W', '8', '0', '8', '50');

insert into time\_slot values ('A', 'F', '8', '0', '8', '50');

insert into time\_slot values ('B', 'M', '9', '0', '9', '50');

insert into time\_slot values ('B', 'W', '9', '0', '9', '50');

insert into time\_slot values ('B', 'F', '9', '0', '9', '50');

insert into time\_slot values ('C', 'M', '11', '0', '11', '50');

insert into time\_slot values ('C', 'W', '11', '0', '11', '50');

insert into time\_slot values ('C', 'F', '11', '0', '11', '50');

insert into time\_slot values ('D', 'M', '13', '0', '13', '50');

insert into time\_slot values ('D', 'W', '13', '0', '13', '50');

insert into time\_slot values ('D', 'F', '13', '0', '13', '50');

insert into time\_slot values ('E', 'T', '10', '30', '11', '45 ');

insert into time\_slot values ('E', 'R', '10', '30', '11', '45 ');

insert into time\_slot values ('F', 'T', '14', '30', '15', '45 ');

insert into time\_slot values ('F', 'R', '14', '30', '15', '45 ');

insert into time\_slot values ('G', 'M', '16', '0', '16', '50');

insert into time\_slot values ('G', 'W', '16', '0', '16', '50');

insert into time\_slot values ('G', 'F', '16', '0', '16', '50');

insert into time\_slot values ('H', 'W', '10', '0', '12', '30');

insert into prereq values ('BIO-301', 'BIO-101');

insert into prereq values ('BIO-399', 'BIO-101');

insert into prereq values ('CS-190', 'CS-101');

insert into prereq values ('CS-315', 'CS-101');

insert into prereq values ('CS-319', 'CS-101');

insert into prereq values ('CS-347', 'CS-101');

insert into prereq values ('EE-181', 'PHY-101');

WRITE QUERIES:

--Find the names of all the students whose total credits are greater than 100

select name from student where tot\_cred>100

--Find the course id and grades of all courses taken by any student named 'Tanaka'

select t.course\_id,t.grade from takes t,student s

where t.id=s.id and s.name='Tanaka'

--Find the ID and name of instructors who have taught a course in the Comp. Sci. department, even if they are themselves not from the Comp. Sci. department. To test this query, make sure you add appropriate data, and include the corresponding insert statements along with your query.

select distinct i.dept\_name,i.ID,i.name from instructor i,course c,teaches t

where t.course\_id=c.course\_id and i.id=t.id and c.dept\_name='Comp. Sci.'

--Find the courses which are offered in both 'Fall' and 'Spring' semester (not necessarily in the same year)

select distinct c.title from course c,section s

where c.course\_id=s.course\_id and

c.course\_id in

(select s.course\_id from section s where s.semester='Fall'

intersect

select s.course\_id from section s where s.semester='Spring')

--Find the names of all the instructors from Comp. Sci. department

select name from instructor where dept\_name ='Comp. Sci.'

--Find the course id and titles of all courses taught by an instructor named 'Srinivasan'

select c.course\_id,c.title from course c,instructor i,teaches t

where t.course\_id=c.course\_id and t.ID=i.ID and i.name='Srinivasan'

--Find names of instructors who have taught at least one course in Spring 2009

select name from instructor where id in

(select i.id from instructor i,teaches t where t.id=i.id and t.semester='Spring' and t.year='2009'

Group by i.id having count(\*)>=1)

Aggregate Function:

* Find the number of instructors who have never taught any course. If the result of your query is empty, add appropriate data (and include corresponding insert statements) to ensure the result is not empty.

select count (\*) from instructor

where id not in (select id from teaches)

* Find the total capacity of every building in the university

select sum(capacity) from classroom group by building

* Find the maximum number of teachers for any single course section.  Your output should be a single number.  For example if CS-101 section 1 in Spring 2012 had 3 instructors teaching the course, and no other section had more instructors teaching the section, your answer would be 3.

select max(c1) as maximum\_instructors from v1

* Find all departments that have at least one instructor, and list the names of the departments along with the number of instructors;   order the result in descending order of number of instructors.

select dept\_name,count(\*)as c1 from instructor

group by dept\_name having count(\*)>=1

order by c1 desc

select department.dept\_name

from department left join instructor

on department.dept\_name=instructor.dept\_name

insert into department

values('psychology','xyz','5000')

* As in the previous question, but this time you shouold include departments even if they do not have any instructor, with the count as 0

select department.dept\_name,count(id)

from department left join instructor

on department.dept\_name=instructor.dept\_name

group by department.dept\_name

* For each student, compute the total credits they have successfully completed, i.e. total credits of courses they have taken, for which they have a non-null grade other than 'F'. Do NOT use the tot\_creds attribute of student.
* Find the number of students who have been taught (at any time) by an instructor named 'Srinivasan'. Make sure you count a student only once even if the student has taken more than one course from Srinivasan.
* Find the name of all instructors who get the highest salary in their department.
* Find all students who have taken all courses taken by instructor 'Srinivasan'. (This is the division operation of relational algebra.) You can implement it by counting the number of courses taught by Srinivasan, and for each student (i.e. group by student), find the number of courses taken by that student, which were taught by Srinivasan. Make sure to count each course ID only once.
* Find the total money spent by each department for salaries of instructors of that department.
* Find the names of all students whose advisor has taught the maximum number of courses (multiple offerings of a course count as only 1).

Nested Queries

* Find the id and title of all courses which do not require any prerequisites.
* Find the names of students who have not taken any biology dept courses
* Write SQL update queries to perform the following (queries 2 and 4 are pretty meangless, but still fun to write):
* Give a 10% hike to all instructors
* Increase the tot\_creds of all students who have taken the course titled "Genetics" by the number of credits associated with that course.
* For all instructors who are advisors of atleast 2 students, increase their salary by 50000.
* Set the  credits to 2 for all courses which have less than 5 students taking them (across all sections for the course, across all years/semesters).