**Test Date: 28 September 2012**

**Aim:**

Implement the following schema and solve the queries.

**Schema:**

**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\_HR START\_MIN END\_HR END\_MIN

**PREREQ**

COURSE\_ID PREREQ\_ID

**SQL:**

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

);

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');

1. ***Find course\_id and title of all the courses offered by 'Comp. Sci.' department.***

> select course\_id,title from course where dept\_name='Comp. Sci.';

+-----------+----------------------------+

| course\_id | title |

+-----------+----------------------------+

| CS-101 | Intro. to Computer Science |

| CS-190 | Game Design |

| CS-315 | Robotics |

| CS-319 | Image Processing |

| CS-347 | Database System Concepts |

+-----------+----------------------------+

1. ***Find names of all students who have a class starting at 10.30 on Tuesdays in Spring 2010.***

> select student.name from student, section, time\_slot, takes where section.course\_id=takes.course\_id and takes.id=student.id and section.time\_slot\_id=time\_slot.time\_slot\_id and section.semester='spring' and section.year=2010 and time\_slot.day='T' and time\_slot.start\_hr=10 and time\_slot.start\_min=30;

Empty set

1. ***Find id and title of all the courses offered in Spring 2010, which have no pre-requisite.***

> select unique section.course\_id,title from course,prereq,section where section.course\_id=course.course\_id and semester='Spring' and year=2010 and section.course\_id not in (select prereq.course\_id from prereq);

+-----------+----------------------------+

| course\_id | title |

+-----------+----------------------------+

| CS-101 | Intro. to Computer Science |

| FIN-201 | Investment Banking |

| HIS-351 | World History |

| MU-199 | Music Video Production |

+-----------+----------------------------+

1. ***Find the departments (without duplicates) of courses that have the maximum credits.***

>select unique dept\_name from course where credits=(select max(credits) from course);

+------------+

| dept\_name |

+------------+

| Biology |

| Comp. Sci. |

| Physics |

+------------+

1. ***Create a view rich\_instructors which shows ID, name, dept\_name and salary of all instructors with salary > 50000. Also, show the effect of inserting a tuple in rich\_instructor with salary of 100000, and another with salary of 10000.***

> create view rich\_instructors as select id,name,dept\_name,salary from instructor where salary>50000;

> select \* from rich\_instructors;

+-------+------------+------------+----------+

| id | name | dept\_name | salary |

+-------+------------+------------+----------+

| 10101 | Srinivasan | Comp. Sci. | 65000.00 |

| 12121 | Wu | Finance | 90000.00 |

| 22222 | Einstein | Physics | 95000.00 |

| 32343 | El Said | History | 60000.00 |

| 33456 | Gold | Physics | 87000.00 |

| 45565 | Katz | Comp. Sci. | 75000.00 |

| 58583 | Califieri | History | 62000.00 |

| 76543 | Singh | Finance | 80000.00 |

| 76766 | Crick | Biology | 72000.00 |

| 83821 | Brandt | Comp. Sci. | 92000.00 |

| 98345 | Kim | Elec. Eng. | 80000.00 |

+-------+------------+------------+----------+

1. ***Show a list of all instructors (ID and name), along with the course\_id of courses they have taught. If they have not taught any course, show the ID and name, with null value for course\_id.***

> select instructor.id,instructor.name,teaches.course\_id from instructor left outer join teaches on instructor.id=teaches.id;

+-------+------------+-----------+

| id | name | course\_id |

+-------+------------+-----------+

| 10101 | Srinivasan | CS-101 |

| 10101 | Srinivasan | CS-315 |

| 10101 | Srinivasan | CS-347 |

| 12121 | Wu | FIN-201 |

| 15151 | Mozart | MU-199 |

| 22222 | Einstein | PHY-101 |

| 32343 | El Said | HIS-351 |

| 33456 | Gold | NULL |

| 45565 | Katz | CS-101 |

| 45565 | Katz | CS-319 |

| 58583 | Califieri | NULL |

| 76543 | Singh | NULL |

| 76766 | Crick | BIO-101 |

| 76766 | Crick | BIO-301 |

| 83821 | Brandt | CS-190 |

| 83821 | Brandt | CS-190 |

| 83821 | Brandt | CS-319 |

| 98345 | Kim | EE-181 |

+-------+------------+-----------+

1. ***Find IDs and names of all students who have not enrolled in any course in Spring 2010.(using outer join)***

> select y.id,y.name from (select student.id,name,takes.course\_id from student left outer join takes on takes.ID=student.ID and semester='Spring' and year='2010') y where y.course\_id is NULL;

+-------+----------+

| id | name |

+-------+----------+

| 00128 | Zhang |

| 44553 | Peltier |

| 54321 | Williams |

| 70557 | Snow |

| 76653 | Aoi |

| 98988 | Tanaka |

+-------+----------+

1. ***Create a schema to represent the following information. Make appropriate assumptions about types and string/numeric lengths.***

***\* item(Iid, name, price)***

***\* customer(Cid, name)***

***\* bill(Bid, Cid, totalprice)***

***\* billitem(Bid, lineno, Iid, quantity)***

***Here, a bill can have multiple lines, and billitem has a tuple for each line in the bill. Make sure to define the primary key and foreign keys.***

>

1. ***Suppose we wish to change the slot for CS 101 in Spring 2010. Find slots in which all students currently registered for the course are free.***

>

1. ***Find the IDs of all students who have taken all the prerequisites of the course CS 317, using the counting method***

>