数据库原理实验报告

(2019-2020年度第一学期)

学院：­­ 计算机科学与技术

专业：­­ 计算机科学与技术

班级：­­ 25班

学号：­­ 21172501

姓名： 孙博文

|  |  |
| --- | --- |
| 实验环境 | Windows 10 1903 64bit，MySQL 8.0  注：考虑到数据的关联性，对部分题目的题干进行了适量更改，以删除线的方式展现修改后的结果。 |
| 实验题目1 | 从课程表（course）中查询所有课程信息。 |
| 答案 | select \* from course; |
| 效果截图 |  |
| 实验题目2 | 从课程表中（course）查询课程名。 |
| 答案 | select title from course; |
| 效果截图 |  |
| 实验题目3 | 从课程段（section）表中查询课程名称，要求消除值相同的那些行。 |
| 答案 | select distinct title from course where course\_id in (select course\_id from section); |
| 效果截图 |  |
| 实验题目4 | 从学生（student）表中查询所有的信息，要求只显示查询结果的前6行数据。 |
| 答案 | select \* from student limit 6; |
| 效果截图 |  |
| 实验题目5 | 查询选了所有计算机学院开设课程的学生的姓名。 |
| 答案 | select S.name from student as S where not exists (select course\_id from course where dept\_name = 'Comp. Sci.' and course\_id not in(select T.course\_id from takes as T where S.ID = T.ID)); |
| 效果截图 |  |
| 实验题目6 | 查询~~2019~~ 2010年春季开课，但~~2018~~ 2009年不开课的课程的编号。 |
| 答案 | select distinct course\_id from section  where semester = 'Spring' and year = '2010'  and course\_id in  (select course\_id from section  where course\_id not in  (select course\_id from section where year = '2009')); |
| 效果截图 |  |
| 实验题目7 | 假设毕业要求为修够80学分，请统计计算机学院内学生距离毕业要求还差多少学分，并按所差分数的升序排列。 |
| 答案 | SELECT ID, name, dept\_name, (80-tot\_cred) as A FROM student where dept\_name = 'Comp. Sci.' and tot\_cred <= 80 order by A desc; |
| 效果截图 |  |
| 实验题目8 | 统计~~2019~~ 2009年春季所开课程段选课人数的最大值。 |
| 答案 | with a(course\_id, val) as  (select takes.course\_id, count(\*) from course, takes  where course.course\_id = takes.course\_id  and takes.year = '2009' and takes.semester = 'Spring' group by course.course\_id)  select course\_id, max(val) from a; |
| 效果截图 |  |
| 实验题目9 | 统计各个学院老师的平均年薪。 |
| 答案 | select dept\_name, avg(salary) as avg\_salary from instructor group by dept\_name; |
| 效果截图 |  |
| 实验题目10 | 请输出没有选择~~2019~~ 2009年春季开课的课程段的学生的姓名。 |
| 答案 | with choose(ID, name) as  (select student.ID, student.name from student, takes  where year = '2009' and semester = 'Spring' and student.ID = takes.ID)  select distinct student.ID, student.name  from student left join choose  on (student.ID = choose.ID and student.name = choose.name)  where choose.ID is null; |
| 效果截图 |  |
| 实验题目11 | 请输出“~~张三~~ Srinivasan”指导的学生在~~2019~~ 2009年春季开设课程段中所获得的总学分。（我们默认只要选择某个course的一个section，就可以获得这个course的学分；若选择同一个course的多个section，也只获得1次这个course的学分）。 |
| 答案 | select S.ID, S.name, sum(credits) from student as S, course as C, takes as T, instructor as I where I.ID in (select ID from instructor where name = 'Srinivasan') and (S.ID, I.ID) in (select A.s\_ID, A.i\_ID from advisor as A) and year = '2009' and semester = 'Spring' and T.ID = S.ID and T.course\_id = C.course\_id group by S.ID; |
| 效果截图 |  |
| 实验题目11 | 请输出“~~数据库系统原理~~ Database System Concepts”和“~~离散数学~~Robotics”的共同的先修课程的ID。 |
| 答案 | select prereq\_id from prereq where course\_id in (select course\_id from course where title = 'Database System Concepts') and prereq\_id in (select prereq\_id from prereq where course\_id in (select course\_id from course where title = 'Robotics')); |
| 效果截图 |  |
| 实验题目12 | 请统计~~2016-2018~~ 2009-2010年，计算机学院每年开设的课程段的数量。 |
| 答案 | select year, count(course\_id) from section where course\_id in (select course\_id from course where dept\_name = 'Comp. Sci.') and (year = '2009' or year = '2010') group by year; |
| 效果截图 |  |
| 实验题目13 | 查询课程编号不为“004”、“007”、“013”的课程编号和课程名称。 |
| 答案 | select course\_id, title from course where course\_id not like '%004%' or course\_id not like '%007%' or course\_id not like '%013%'; |
| 效果截图 |  |
| 实验题目14 | 查询课程名以字母~~D~~ C开始，以 ~~e~~  y结尾的课程信息。 |
| 答案 | select \* from course where title like 'C%y'; |
| 效果截图 |  |
| 实验题目15 | 查询课程名以“~~制作~~ to”两字作为中间字的课程信息。（要求“~~制作~~to”不做开头和结尾） |
| 答案 | select \* from course where title like '%to%' and title not like 'to%' and title not like '%to'; |
| 效果截图 |  |
| 实验题目16 | 查询姓名第二个字为“~~宝~~ a”的学生信息。 |
| 答案 | select \* from student where name like '\_a%'; |
| 效果截图 |  |
| 实验题目17 | 查询不姓“~~刘~~ B”的学生信息。 |
| 答案 | select \* from student where name not like 'B%'; |
| 效果截图 |  |
| 实验题目18 | 查询那些在~~2018~~ 2009年有至少两个课程段的课程的ID。 |
| 答案 | select T.course\_id from course as T where 1 < (select count(R.course\_id) from course, section as R where T.course\_id = R.course\_id and R.year = '2009'); |
| 效果截图 |  |
| 实验题目19 | 查询计算机学院学生选了非本学院老师开设的属于本学院的课程的情况，统计这些学生的ID,姓名，总学分。 |
| 答案 | select ID, name, tot\_cred from student where dept\_name = 'Comp. Sci.' and ID in (select ID from takes where course\_id in (select course\_id from course where dept\_name = 'Comp. Sci.') and (course\_id, sec\_id, semester, year) in (select course\_id, sec\_id, semester, year from teaches where ID in (select ID from instructor where dept\_name <> 'Comp. Sci.'))); |
| 效果截图 |  |
| 实验题目20 | 查询~~2019~~ 2009年春季开设的，选课人数少于~~25~~ 3并且多于~~15~~ 1人的课程段信息。 |
| 答案 | select \* from section as S where semester = 'Spring' and year = '2009' and 1 < (select count(ID) from takes as T where T.course\_id = S.course\_id) and 3 > (select count(ID) from takes as T where T.course\_id = S.course\_id); |
| 效果截图 |  |
| 实验题目21 | 查询~~2018~~ 2009年由非本学院开设的课程段的总数。如果没有，输出结果为0。 |
| 答案 | select count(course\_id) from section where year = '2009' and course\_id in (select course\_id from teaches where year = '2009' and ID not in (select ID from instructor where dept\_name = 'English')) and course\_id in (select course\_id from course where dept\_name = 'English'); |
| 效果截图 |  |
| 实验题目22 | 查询报名人数大于25或者少于15人的课程信息，要求查询结果按照报名人数降序排列。 |
| 答案 | with A(course\_id, sec\_id, semester, year ,sum) as (select course\_id, sec\_id, semester, year ,count(\*) from takes group by course\_id, sec\_id, semester, year)  select \* from section natural join A where A.sum < 15 or A.sum > 25 order by A.sum desc; |
| 效果截图 |  |
| 实验题目23 | 查询与计算机学院同处于一座大楼的其他学院的老师的平均工资。 |
| 答案 | select dept\_name,avg(salary) from instructor as I where (I.dept\_name in  (select D.dept\_name from department as D where D.dept\_name <> 'Comp. Sci.' and D.building in (select D.building from department as D where D.dept\_name = 'Comp. Sci.'))) group by I.dept\_name; |
| 效果截图 |  |
| 实验题目24 | 给在“~~匡亚明~~ Taylor”大楼内办公的老师的工资增加到原来工资的1.5倍。 |
| 答案 | update instructor I set I.salary = I.salary \* 1.5 where I.dept\_name in (select D.dept\_name from department as D where D.building = ' Taylor '); |
| 效果截图 | 原有：    现在： |
| 实验题目25 | 统计各个学院的学生的数量。 |
| 答案 | select dept\_name, count(ID) from student group by dept\_name; |
| 效果截图 |  |
| 实验题目26 | 统计计算机学院中所获总学分排名前~~10~~ 2位的学生的信息。  （数据量不够，见谅） |
| 答案 | select \* from student where dept\_name = 'Comp. Sci.' order by tot\_cred desc limit 2; |
| 效果截图 |  |
| 实验题目27 | 查询“~~李四~~ Brandt”老师在~~2017-2019~~ 2009-2010年开设课程段的数量。 |
| 答案 | select count(T.ID) from teaches as T where (T.year = '2009' or T.year = '2010') and T.ID in (select I.ID from instructor as I where I.name = 'Brandt'); |
| 效果截图 |  |
| 实验题目28 | 查询计算机学院老师中比~~生物~~物理学院工资最高的老师工资低，但是比~~生物~~物理学院最低工资高的老师的信息。 |
| 答案 | select \* from instructor where dept\_name = 'Comp. Sci.' and salary > (select min(salary) from instructor where dept\_name = 'Physics') and salary < (select max(salary) from instructor where dept\_name = 'Physics'); |
| 效果截图 |  |
| 实验题目29 | 请统计~~2019~~ 2009年计算机学院比生物学院多开设了几个课程段。 |
| 答案 | select A.id - B.id from (select count(course\_id) as id from section where year = '2009' and course\_id in (select course\_id from course where dept\_name = 'Comp. Sci.')) as A, (select count(course\_id) as id from section where year = '2009' and course\_id in (select course\_id from course where dept\_name = 'Biology')) as B; |
| 效果截图 |  |
| 实验题目30 | 请统计~~2019~~ 2009年春季，没有学生选课的课程段的数量。若没有，请输出0。 |
| 答案 | select count(S.course\_id) from section as S where S.year = '2009' and S.semester = 'Spring' and 1 > (select count(T.ID) from takes as T where S.course\_id = T.course\_id); |
| 效果截图 |  |
| 实验题目31 | 请输出计算机学院学生的指导老师的姓名，去掉重复信息。 |
| 答案 | select distinct name from instructor as I where I.ID in (select A.i\_ID from advisor as A where A.s\_ID in (select S.ID from student as S where S.dept\_name = 'Comp. Sci.')); |
| 效果截图 |  |
| 实验题目32 | 将低于整个学校的平均预算的各个学院的预算提高到原来的1.2倍。 |
| 答案 | update department set budget = budget \* 1.2 where budget < (select \* from (select avg(budget) from department) a ); |
| 效果截图 | 原有：    现在： |
| 实验题目33 | 统计“~~李四~~ Srinivasan”老师指导的学生数量。若没有，请输出0。 |
| 答案 | select count(A.s\_ID) from advisor as A where A.i\_ID in (select I.ID from instructor as I where I.name = 'Srinivasan'); |
| 效果截图 |  |
| 实验题目34 | 查询已经选了“~~数据库原理~~ Database System Concepts”这门课的所有直接先修课程的学生的信息 |
| 答案 | select S.\* from student as S where S.ID in  (select T.ID from takes as T where T.course\_id in  (select P.prereq\_id from prereq as P where P.course\_id in  (select C.course\_id from course as C where C.title = 'Database System Concepts'))); |
| 效果截图 |  |
| 实验题目35 | 查询~~2019~~ 2009年春季选了自己指导教师开设的课程段的学生的姓名、指导老师的姓名和课程段的ID。 |
| 答案 | select distinct S.name as student\_name, I.name as instructor\_name, TA.course\_id, TA.sec\_id from student as S, instructor as I, takes as TA, teaches as TE where TA.year = '2009' and TA.semester = 'Spring' and TE.year = TA.year and TE.semester = TA.semester and (TA.ID, TE.ID) in (select s\_ID, i\_ID from advisor) and (TA.course\_id, TA.sec\_id, TA.semester, TA.year) in (select course\_id, sec\_id, semester, year from teaches) and S.ID = TA.ID and I.ID = TE.ID; |
| 效果截图 |  |

附件1：建立表目的语句

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(dept\_name)

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(dept\_name)

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(course\_id)

on delete cascade,

foreign key (building, room\_number) references classroom(building, room\_number)

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

on delete cascade,

foreign key (ID) references instructor(ID)

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 (dept\_name)

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

on delete cascade,

foreign key (ID) references student(ID)

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(course\_id)

on delete cascade,

foreign key (prereq\_id) references course(course\_id)

);

附件2：插入数据的语句

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 classroom values ('Gates', 707, 65);

insert into classroom values ('Gates', 314, 10);

insert into classroom values ('Main', 45, 30);

insert into classroom values ('Main', 425, 22);

insert into classroom values ('Fairchild', 145, 27);

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 department values ('Math', 'Brodhead', 110000.00);

insert into department values ('Astronomy', 'Taylor', 150000.00);

insert into department values ('English', 'Palmer', 60000.00);

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 course values ('MA-101', 'Computability Theory', 'Math', '4');

insert into course values ('AS-101', 'Sensor Networks', 'Astronomy', '4');

insert into course values ('EN-201', 'Religion', 'English', '3');

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 instructor values ('28097', 'Kean', 'English', '35000');

insert into instructor values ('43779', 'Romero', 'Astronomy', '75000');

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 section values ('EN-201', '1', 'Fall', '2009', 'Main', '45', 'A');

insert into section values ('CS-319', 1, 'Fall', 2011, 'Fairchild', 145, '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 teaches values ('32343','EN-201',1,'Fall',2009);

insert into teaches values ('43779', 'CS-319', 1, 'Fall', 2011);

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-319', '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 takes values ('76543', 'CS-319', 1, 'Fall', 2011, 'A');

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 advisor values ('54321', '10101');

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');

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

insert into prereq values ('EN-201', 'HIS-351');

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