23111 - Ex2-S-Basic SQL

Anna McCartain

10362766

Documentation For The Physical Design Of The University Database

--For this part I created a table for each table in the physical
--design, Values I assigned according to the layout of values in
--the insert document, ie VarChar is for values with char and
--mixed char and ints, like course subject names, foreign keys
--represent the arrows on the physical design

Student		
Field	**Data Type**	**Constraints**
id	Integer	Unsigned, 0-4294967295, primary key
name	VarChar	255 characters, not null
dept_name	VarChar	100 characters, foreign key
tot_cred	SmallInt	Unsigned, 0-65535
Takes		
Field	**Data Type**	**Constraints**
id	Integer	Unsigned, 0-4294967295, primary key, foreign key
course_id	VarChar	100 characters, primary key, foreign key
sec_id	SmallInt	Unsigned, 0-65535, primary key, foreign key
semester	Char	100 characters, primary key, foreign key
year	SmallInt	Unsigned, 0-65535, not null, primary key, foreign key
grade	VarChar	15 characters, not null
Section		
Field	**Data Type**	**Constraints**
course_id	VarChar	100 characters, primary key, foreign key
sec_id	SmallInt	Unsigned, 0-65535, primary key
semester	Char	100 characters, primary key
year	SmallInt	Unsigned, 0-65535, primary key
building	Char	100 characters, foreign key
room_no	SmallInt	Unsigned, 0-65535, foreign key

time_slot_id	Char	15 characters, foreign key
Time_slot		
Field	**Data Type**	**Constraints**
time_slot_id	Char	15 characters, not null, primary key
day	Char	15 characters, not null, primary key
start_hour	SmallInt	Unsigned, 0-65535, not null, primary key
start_min	SmallInt	Unsigned, 0-65535, not null, primary key
end_hour	SmallInt	Unsigned, 0-65535, not null
end_min	SmallInt	Unsigned, 0-65535, not null
Classroom		
Field	**Data Type**	**Constraints**
building	Char	100 characters, primary key
room_no	SmallInt	Unsigned, 0-65535, primary key
capacity	SmallInt	Unsigned, 0-65535, not null
Teaches		
Field	**Data Type**	**Constraints**
id	Integer	Unsigned, 0-4294967295, primary key, foreign key
course_id	VarChar	100 characters, primary key, foreign key
sec_id	SmallInt	Unsigned, 0-65535, primary key, foreign key
semester	Char	100 characters, primary key, foreign key
year	SmallInt	Unsigned, 0-65535, primary key, foreign key
Course		
Field	**Data Type**	**Constraints**
course_id	VarChar	15 characters, primary key
title	Char	100 characters, not null
dept_name	VarChar	100 characters, foreign key
credits	SmallInt	Unsigned, 0-65535
Prereq		
Field	**Data Type**	**Constraints**
course_id	VarChar	100 characters, primary key, foreign key
prereq_id	VarChar	100 characters, primary key, foreign key

Instructor		
Field	**Data Type**	**Constraints**
id	Integer	Unsigned, 0-4294967295, primary key
name	Char	100 characters, not null
dept_name	VarChar	100 characters, foreign key
salary	Integer	Unsigned, 0-4294967295
Advisor		
Field	**Data Type**	**Constraints**
s_id	Integer	Unsigned, 0-4294967295, primary key, foreign key
i_id	Integer	Unsigned, 0-4294967295, foreign key
Department		
Field	**Data Type**	**Constraints**
dept_name	VarChar	100 characters, primary key
building	Char	100 characters, not null
budget	Integer	Unsigned, 0-65535, not null

Implement The University Database Using MySQL Statements

```
--this section written in sql, primary keys are the underlined values,
--there can be multiple, foreign keys I have split up according to if they
--point to a different table (see advisor). Not null assigned to values
--which cannot be null.


CREATE TABLE classroom (
                        building    CHAR(100),
                        room_no     SMALLINT UNSIGNED,
                        capacity    SMALLINT UNSIGNED NOT NULL,
                        PRIMARY KEY (building, room_no)
                        );

CREATE TABLE department (
                        dept_name   VARCHAR(100),
                        building    CHAR(100) NOT NULL,
```

```
                        budget              INT UNSIGNED NOT NULL,
                        PRIMARY KEY (dept_name)
                        );

CREATE TABLE instructor (
                        id          INT UNSIGNED,
                        name        CHAR(100) NOT NULL,
                        dept_name   VARCHAR(100),
                        salary              INT UNSIGNED,
                        PRIMARY KEY (id),
                        FOREIGN KEY (dept_name)
                               REFERENCES Department(dept_name)
                        );

CREATE TABLE course (
                        course_id   VARCHAR(15),
                        title       CHAR(100) NOT NULL,
                        dept_name   VARCHAR(100),
                        credits             SMALLINT UNSIGNED,
                        PRIMARY KEY (course_id),
                        FOREIGN KEY (dept_name)
                              REFERENCES Department(dept_name)
                        );

CREATE TABLE student (
                        Id          INT UNSIGNED,
                        name        VARCHAR(255) NOT NULL,
                        dept_name   VARCHAR(100),
                        tot_cred    SMALLINT UNSIGNED,
                        PRIMARY KEY (id),
                        FOREIGN KEY (dept_name)
                              REFERENCES Department(dept_name)
                        );

CREATE TABLE time_slot (
                        time_slot_id      CHAR(15) NOT NULL,
                        Day         CHAR(15) NOT NULL,
                        start_hour  SMALLINT UNSIGNED NOT NULL,
                        start_min   SMALLINT UNSIGNED NOT NULL,
                        end_hour    SMALLINT UNSIGNED NOT NULL,
                        end_min     SMALLINT UNSIGNED NOT NULL,
                        PRIMARY KEY (time_slot_id, day, start_hour,
                        start_min)
                        );

CREATE TABLE advisor (
                        s_id        INT UNSIGNED,
                        i_id        INT UNSIGNED,
                        PRIMARY KEY (s_id),
                        FOREIGN KEY (s_id)
                              REFERENCES Student(s_id),
```

```
                        FOREIGN KEY (i_id)
                              REFERENCES Instructor(i_id)
                        );

CREATE TABLE prereq (
                        course_id   VARCHAR(100),
                        prereq_id   VARCHAR(100),
                        PRIMARY KEY (course_id, prereq_id),
                        FOREIGN KEY (course_id)
                              REFERENCES Course(course_id),
                        FOREIGN KEY (prereq_id)
                              REFERENCES Course(prereq_id)
                        );

CREATE TABLE section (
                        course_id   VARCHAR(100),
                        sec_id            SMALLINT UNSIGNED,
                        semester    CHAR(100),
                        year        SMALLINT UNSIGNED,
                        building    CHAR(100),
                        room_no     SMALLINT UNSIGNED,
                        time_slot_id      CHAR(15),
                        PRIMARY KEY(course_id, sec_id, semester, year),
                        FOREIGN KEY (course_id)
                              REFERENCES Course(course_id),
                        FOREIGN KEY (building, room_no)
                              REFERENCES Classroom(building, room_no),
                        FOREIGN KEY (time_slot_id)
                              REFERENCES Time_slot(time_slot_id)
                        );

CREATE TABLE teaches (
                        id          INT UNSIGNED,
                        course_id   VARCHAR(100),
                        sec_id            SMALLINT UNSIGNED,
                        semester    CHAR(100),
                        year        SMALLINT UNSIGNED,
                        PRIMARY KEY(id, course_id, sec_id, semester, year),
                        FOREIGN KEY (id)
                              REFERENCES Instructor(id),
                        FOREIGN KEY (course_id, sec_id, semester, year)
                              REFERENCES Section(course_id, sec_id,
                        semester, year)
                        );

CREATE TABLE takes (
                        id          INT UNSIGNED,
                        course_id   VARCHAR(100),
                        sec_id            SMALLINT UNSIGNED,
                        semester    CHAR(100),
                        year        SMALLINT UNSIGNED NOT NULL,
```

```
grade       VARCHAR(15),
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),
FOREIGN KEY (id)
      REFERENCES Student(id)
);
```

Add Test Records For Each Of The Tables

```
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');
```

TASK 5

```
--Find the names of all students who have taken at least one
--computer science course, making sure there are no duplicate
--names in the result.

SELECT DISTINCT student.name FROM student, takes
     WHERE student.ID = takes.ID
     AND takes.course_id LIKE '%CS%';
```

TASK 6

```
--Find the IDs and names of all students with a fail grade.

SELECT student.id, student.name FROM student, takes
     WHERE student.ID = takes.ID
     AND   takes.grade = 'F';
```

TASK 7

```
--For each department, find the maximum salary of instructors in
--that department. Every department should have at least one
--instructor.

SELECT department.dept_name, instructor.salary
     FROM instructor, department
     WHERE (department.dept_name, instructor.salary) IN
     (SELECT department.dept_name, MAX(instructor.salary)
     FROM instructor, department GROUP BY department.dept_name)
```