**HW3**

**Name: Will Boland**

**Part 1: Creating Tables**

**Professor Creation**

CREATE TABLE Professor

(

Prof\_ID int(10) NOT NULL,

Prof\_Name varchar(50) NOT NULL,

Prof\_Email varchar(50) NULL,

Prof\_Office varchar(100) NULL

);

**Student Creation**

CREATE TABLE Student

(

Student\_ID int(10) NOT NULL,

Student\_Name varchar(50) NOT NULL,

Student\_Email varchar(50) NULL,

Student\_Year varchar(10) NULL

);

**Course Creation**

CREATE TABLE Course

(

Course\_ID int(10) NOT NULL,

Course\_Name varchar(50) NOT NULL,

Department varchar(50) NULL

);

**Enrollment Creation**

CREATE TABLE Enrollment

(

Student\_ID int(10) NOT NULL,

Prof\_ID int(10) NOT NULL,

Course\_ID int(10) NOT NULL,

Semester varchar(25) NOT NULL,

Grade varchar(3) NULL

);

## INSERTION STATEMENTS

**Professor Insertion Statements**

INSERT INTO Professor

(

Prof\_ID,

Prof\_Name,

Prof\_Email,

Prof\_Office

)

VALUES

(

0003438284,

'Steve Smith',

'ssmith@college.edu',

'Flagler Building Room 201'

);

INSERT INTO Professor

(

Prof\_ID,

Prof\_Name,

Prof\_Email,

Prof\_Office

)

VALUES

(

0003040029,

'Fred Stanza',

'stanf@college.edu',

'Library Room 104'

);

INSERT INTO Professor

(

Prof\_ID,

Prof\_Name,

Prof\_Email,

Prof\_Office

)

VALUES

(

0002938729,

'Brandy Queens',

'queenb@college.edu',

'Union Starbucks'

);

INSERT INTO Professor

(

Prof\_ID,

Prof\_Name,

Prof\_Email,

Prof\_Office

)

VALUES

(

0003848738,

'Brad Bradleson',

'bradbrad@college.edu',

'Flagler Building Room 222'

);

INSERT INTO Professor

(

Prof\_ID,

Prof\_Name,

Prof\_Email,

Prof\_Office

)

VALUES

(

0004938293,

'Estabon Espenoza',

'eespenoza@college.edu',

'Library Starbucks'

);

**Student Insertion Statements**

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0005993049,

'Brent Bartles',

'bartleb@college.edu',

'Sophomore'

);

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0005586949,

'Riley Moo',

'moomoo@college.edu',

'Freshman'

);

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(0003849283,'Baxter Morris','baxtm@college.edu','Graduate');

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0005839299,

'Georgio Estobane',

'geoest@college.edu',

'Freshman'

);

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0003929483,

'Colin Farkone',

'farkone@college.edu',

'Junior'

);

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0003929283,

'Jorge Bonametchio',

'bonametchio@college.edu',

'Senior'

);

INSERT INTO Student

(

Student\_ID,

Student\_Name,

Student\_Email,

Student\_Year

)

VALUES

(

0004838293,

'Alphonso Barone',

'baronea@college.edu',

'Junior'

);

**Course Insertion Statements**

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(

129182829,

'Physics 101',

'Physics'

);

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(193929304,'Calc 212’,'Math');

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(194958392,'Calc 213’,'Math');

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(198483728,'Database Concepts’,'Csci');

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(138493948,'English 400’,’Education');

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(103858390,'Intro to Telecom’,'Telecom');

INSERT INTO Course

(

Course\_ID,

Course\_Name,

Department

)

VALUES

(193948392,’Stats','Math');

**Course Insertion Statements**

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005993049,0003438284,129182829,'Spring 2014’,'A');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003849283,0003848738,138493948,'Fall 2013’,'C');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005586949,0002938729,129182829,'Fall 2012’,'B+');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003929483,0003040029,103858390,'Summer 2011’,'A');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003929283,0002938729,129182829,'Spring 2014’,'B-');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0004838293,0004938293,138493948,'Fall 2012’,'A');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005993049,0002938729,129182829,'Fall 2014’,'B+');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003929283,0004938293,198483728,'Spring 2014’,'C');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005993049,0004938293,198483728,'Spring 2014’,'A-');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005839299,0004938293,198483728,'Spring 2014’,'F');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0004838293,0004938293,198483728,'Spring 2014’,'A');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005586949,0002938729,198483728,'Fall 2012’,’A');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0005993049,0003848738,138493948,'Summer 2011’,'D+');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003929483,0003438284,193948392,'Fall 2012’,'C-');

INSERT INTO Enrollment

(

Student\_ID,

Prof\_Id,

Course\_ID,

Semester,

Grade

)

VALUES

(0003929483,0002938729,129182829,'Summer 2013’,'D-');

**Part 2: SQL Statements/Queries**

1. Question/Task: List all student names who have received some form of an 'A'.

• Your SQL Statement:

SELECT

Student.Student\_Name, Course.Course\_Name, Enrollment.Semester, Enrollment.Grade

FROM

Student, Enrollment, Course

WHERE

Course.Course\_ID = Enrollment.Course\_ID AND Student.Student\_ID = Enrollment.Student\_ID AND Enrollment.Grade LIKE 'A%'

ORDER BY

Student.Student\_Name;

1. Question/Task: List all student names who took 'Database Concepts' in 'Spring 2014' and their corresponding grades.

• Your SQL Statement:

SELECT

Student.Student\_Name, Course.Course\_Name, Enrollment.Semester, Enrollment.Grade

FROM

Student, Course, Enrollment

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Enrollment.Course\_ID = Course.Course\_ID AND Course.Course\_Name = 'Database Concepts' AND Enrollment.Semester = 'Spring 2014'

ORDER BY

Student.Student\_Name;

1. Question/Task: List all professors who teach a 'Math' class.

• Your SQL Statement:

SELECT

Professor.Prof\_Name, Course.Department, Course.Course\_Name

FROM

Professor, Course, Enrollment

WHERE

Professor.Prof\_ID = Enrollment.Prof\_ID AND Course.Course\_ID = Enrollment.Course\_ID AND Course.Department = 'Math';

1. Question/Task: List all students who have not failed any courses. (HINT: A **passing** grade is one that is C- or better)

• Your SQL Statement:

SELECT

Student.Student\_Name, Course.Course\_Name, Enrollment.Semester, Enrollment.Grade

FROM

Student, Course, Enrollment

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Course.Course\_ID = Enrollment.Course\_ID AND Enrollment.Grade BETWEEN 'A' AND 'C-'

ORDER BY

Student.Student\_Name, Enrollment.Grade;

1. Question/Task: What classes has 'Riley Moo' taken and list the corresponding semester and grade for each class.

• Your SQL Statement:

SELECT

Student.Student\_Name, Course.Course\_Name, Enrollment.Semester, Enrollment.Grade

FROM

Student, Enrollment, Course

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Course.Course\_ID = Enrollment.Course\_ID AND Student.Student\_Name = 'Riley Moo'

ORDER BY

Enrollment.Grade, Enrollment.Semester DESC;

1. Question/Task: List all student names and corresponding course who have retaken a course.

• Your SQL Statement:

SELECT

Student.Student\_Name, Course.Course\_Name, COUNT(Enrollment.Course\_ID) AS Times\_Taken

FROM

Student, Course, Enrollment

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Enrollment.Course\_ID = Course.Course\_ID

GROUP BY

Student.Student\_Name, Enrollment.Course\_ID

HAVING

Times\_Taken >= 2;

1. Question/Task: List all professors who teach more than one course in a semester.

• Your SQL Statement:

SELECT

Professor.Prof\_Name, COUNT(Enrollment.Semester) AS Courses\_Taught

FROM

Professor, Course, Enrollment

WHERE

Professor.Prof\_ID = Enrollment.Prof\_ID AND Course.Course\_ID = Enrollment.Course\_ID

GROUP BY

Professor.Prof\_Name, Enrollment.Course\_ID

HAVING

Courses\_Taught = 2;

1. Question/Task: List all student names and the corresponding number of courses they have taken, in ascending order.

• Your SQL Statement:

SELECT Student.Student\_Name, COUNT(Student.Student\_ID) AS Classes\_Taken

FROM

Student, Enrollment, Course

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Course.Course\_ID = Enrollment.Course\_ID

GROUP BY

Student.Student\_ID

ORDER BY

Classes\_Taken;

1. Question/Task: What courses (display course names) were offered in 'Spring 2014'?

• Your SQL Statement:

SELECT

DISTINCT Course.Course\_Name, Enrollment.Semester

FROM

Course, Enrollment

WHERE

Course.Course\_ID = Enrollment.Course\_ID AND Enrollment.Semester = 'Spring 2014';

1. Question/Task: List all student names with corresponding professors (by name) that have taught them.

• Your SQL Statement:

SELECT

Student.Student\_Name, Professor.Prof\_Name, Course.Course\_Name, Enrollment.Semester

FROM

Student, Professor, Course, Enrollment

WHERE

Student.Student\_ID = Enrollment.Student\_ID AND Professor.Prof\_ID = Enrollment.Prof\_ID AND Course.Course\_ID = Enrollment.Course\_ID

ORDER BY

Student.Student\_Name, Enrollment.Semester, Course.Course\_Name;