

# GradeBook Project

By David Daniels

# Task 1 - ERD Diagrams

Students	
PK	<u>StudentID</u>
	First Name Last Name Major

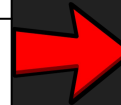
Enrollment	
PK, FK2 PK,FK1	StudentID CourseID

Courses	
PK	<u>CourseID</u>
	Department CourseNumber CourseName Semester Year

StudentScores	
PK, FK1 PK2,FK2	<u>StudentID</u> <u>Assignment ID</u>
	Points

Assignments	
PK	<u>AssignmentID</u>
FK1	DistributionID Instance PointsPossible

Distribution	
PK	<u>DistributionID</u>
FK1	CourseID Category Percent



# Task 2

# Write the commands for creating tables and inserting values

```
CREATE TABLE INSTRUCTOR (Fname VARCHAR(20), Lname VARCHAR(20), Location VARCHAR(20), IEaddr VARCHAR(50),  
Inst_ID INT(4) primary key);
```

```
CREATE TABLE STUDENT (Student_ID INT(4) primary key, Major VARCHAR(20), Lname VARCHAR(20), Fname VARCHAR(20),  
Eaddr VARCHAR(20));
```

```
CREATE TABLE ASSIGNMENT (Aname VARCHAR(20), Avg_Points FLOAT(3,2), MAX_POINTS FLOAT(3,2), Weight INT(3),  
Asgn_ID INT(4) primary key, CRN INT(4) );
```

```
CREATE TABLE COURSE (Cname VARCHAR(20), Term VARCHAR(20), Year VARCHAR(4), CRN INT(4) foreign key, Section_ID  
INT(5) primary key, Cnum INT(4));
```

```
INSERT INTO INSTRUCTOR ("JOHN","LITZ","USA", "101 North King St", 1456);
```

```
INSERT INTO <TABLE NAME> (<COLUMN VALUES SEPARATED BY COMMA ACCORDING TO DATATYPE>);
```

# Task 3

# Show the tables with the contents that you have inserted

```
SELECT * FROM STUDENT;
```

```
SELECT * FROM SCORE;
```

```
SELECT * FROM ASSIGNMENT;
```

```
SELECT * FROM DISTRIBUTION;
```

```
SELECT * FROM ENROLLMENT;
```

```
SELECT * FROM COURSE;
```

# Task 4



# Compute the average/highest/lowest score of an assignment

```
SELECT Avg_Points AS AVERAGE, MAX_POINTS AS MAXIMUM , MIN(Weight) AS MINIMUM  
FROM  
ASSIGNMENT;
```

# Task 5

# List all of the students in a given course

```
SELECT s.Student_ID from STUDENT s, COURSE c, GRADE g
```

where

```
c.CRN = g.CRN
```

```
and s.Student_ID = g.Student_ID
```

# Task 6

# List all of the students in a course and all of their scores on every assignment

```
SELECT DISTINCT pt.StudentID, st.FirstName, st.LastName, pt.CourseID, pt.AssignmentID, pt.CategoryName, pt.Points, pt.PointsPossible, pt.Percentage
FROM (
    SELECT STUDENT.StudentID, AssignmentID, FirstName, LastName, CourseID, Points
    FROM STUDENT JOIN ENROLLMENT JOIN SCORE
    WHERE STUDENT.StudentID = ENROLLMENT.StudentID
    AND STUDENT.StudentID = SCORE.StudentID) st
JOIN
(SELECT StudentID, CourseID, CategoryName, ASSIGNMENT.AssignmentID, Points, ASSIGNMENT.PointsPossible, DISTRIBUTION.Percentage
FROM DISTRIBUTION JOIN ASSIGNMENT JOIN SCORE
WHERE DISTRIBUTION.DistributionID = ASSIGNMENT.DistributionID
AND ASSIGNMENT.AssignmentID = SCORE.AssignmentID) pt
WHERE st.AssignmentID = pt.AssignmentID
AND st.Points = pt.Points AND pt.Courseid = 85675;
```

# Task 8

## Change the percentages of the categories for a course

Update Distribution

Set Percentage = Percentage + 7

where CourseID = 56738 And CategoryName = 'MidTerm';

Update Distribution

Set Percentage = Percentage - 7

where CourseID = 56738 And CategoryName = 'HW';

# Task 9



## Add 2 points to the score of each student on an assignment

```
UPDATE Score
```

```
SET POINTS=Points + 2
```

```
WHERE AssignmentID= 2 AND POINTS <=98;
```

```
SELECT DISTINCT st.StudentID, st.FirstName, st.LastName, pt.CourseID,pt.AssignmentID, pt.CategoryName, pt.Points,  
pt.PointsPossible, pt.Percentage
```

```
FROM (
```

```
SELECT STUDENT.StudentID, AssignmentID, FirstName, LastName, CourseID, Points
```

```
FROM STUDENT JOIN ENROLLMENT JOIN SCORE
```

```
WHERE STUDENT.StudentID = ENROLLMENT.StudentID
```

```
AND STUDENT.StudentID = SCORE.StudentID) st
```

## Add 2 points to the score of each student on an assignment (contd)

JOIN

(SELECT StudentID, CourseID, CategoryName, ASSIGNMENT.AssignmentID, Points, ASSIGNMENT.PointsPossible, DISTRIBUTION.Percentage

FROM DISTRIBUTION JOIN ASSIGNMENT JOIN SCORE

WHERE DISTRIBUTION.DistributionID = ASSIGNMENT.DistributionID

AND ASSIGNMENT.AssignmentID = SCORE.AssignmentID) pt

WHERE st.AssignmentID = pt.AssignmentID

AND st.Points = pt.Points AND pt.AssignmentID = 2;

# Task 10

Add 2 points just to those students whose last name contains a 'Q'.

```
SELECT* from SCORE
WHERE SCORE.studentid IN( SELECT STUDENT.studentid From STUDENT WHERE
    CHARINDEX('Q',lastname)
        or CHARINDEX('Q',firstname)
        or CHARINDEX('q',firstname)
        or CHARINDEX('q',lastname)
    )
```

And SCORE.assignmentid=9;

```
UPDATE SCORE
```

# Add 2 points just to those students whose last name contains a 'Q' (contd)

SET

points = points+2

WHERE SCORE.studentid IN( SELECT STUDENT.studentid From STUDENT WHERE

CHARINDEX('Q',lastname)

or CHARINDEX('Q',firstname)

or CHARINDEX('q',firstname)

or CHARINDEX('q',lastname));

SELECT\* from SCORE

WHERE SCORE.studentid IN( SELECT STUDENT.studentid From STUDENT WHERE

CHARINDEX('Q',lastname)

or CHARINDEX('Q',firstname)

or CHARINDEX('q',firstname)

or CHARINDEX('q',lastname))

And SCORE.assignmentid=9;

# Task 11

## Compute the grade for a student

```
SELECT DISTINCT pt.StudentID, st.FirstName, st.LastName, pt.CourseID, pt.AssignmentID, pt.CategoryName, pt.Points
FROM (
    SELECT STUDENT.StudentID, AssignmentID, FirstName, LastName, CourseID, Points
    FROM STUDENT JOIN ENROLLMENT JOIN SCORE
    WHERE STUDENT.StudentID = ENROLLMENT.StudentID
    AND STUDENT.StudentID = SCORE.StudentID) st
```

## Compute the grade for a student (contd)

JOIN

```
(SELECT StudentID, CourseID, CategoryName, ASSIGNMENT.AssignmentID, Points
```

```
FROM DISTRIBUTION JOIN ASSIGNMENT JOIN SCORE
```

```
WHERE DISTRIBUTION.DistributionID = ASSIGNMENT.DistributionID
```

```
AND ASSIGNMENT.AssignmentID = SCORE.AssignmentID) pt
```

```
WHERE st.AssignmentID = pt.AssignmentID
```

```
AND st.Points = pt.Points AND st.StudentID=1234;
```



# Task 12

## Compute the grade for a student

```
select    sum( distinct pt.Points*pt.Percentage) *1/(1 -( select zt.Percentage From
((SELECT StudentID, CourseID,  CategoryName, ASSIGNMENT.AssignmentID, Points, Percentage
FROM DISTRIBUTION JOIN ASSIGNMENT JOIN SCORE
WHERE DISTRIBUTION.DistributionID = ASSIGNMENT.DistributionID
AND ASSIGNMENT.AssignmentID = SCORE.AssignmentID And CategoryName = 'Final') zt)))
FROM (
SELECT STUDENT.StudentID, AssignmentID, FirstName, LastName, CourseID, Points
FROM STUDENT JOIN ENROLLMENT JOIN SCORE
WHERE STUDENT.StudentID = ENROLLMENT.StudentID
AND STUDENT.StudentID = SCORE.StudentID) st
```

## Compute the grade for a student (cont'd)

```
JOIN  
  
(SELECT StudentID, CourseID, CategoryName, ASSIGNMENT.AssignmentID, Points, Percentage  
FROM DISTRIBUTION JOIN ASSIGNMENT JOIN SCORE  
WHERE DISTRIBUTION.DistributionID = ASSIGNMENT.DistributionID  
AND ASSIGNMENT.AssignmentID = SCORE.AssignmentID) pt  
WHERE st.AssignmentID = pt.AssignmentID  
AND st.Points = pt.Points AND st.StudentID=1234 AND pt.CourseID = 85675 AND Not CategoryName = 'Final';
```

End of Presentation