1. Problem Statement

You are asked to implement a grade book to keep track student grades for several courses that a professor teaches. Courses should have the information of department, course number, course name, semester, and year. For each course, the grade is calculated on various categories, including course participation, homework, tests, projects, etc. The total percentages of the categories should add to 100% and the total perfect grade should be 100. The number of assignments from each category is unspecified, and can change at any time. For example, a course may be graded by the distribution: 10% participation, 20% homework, 50% tests, 20% projects. Please note that if there are 5 homework, each homework is worth 20%/5=4% of the grade.

2.Tasks

Design the ER diagram;

Write the commands for creating tables and inserting values;

Show the tables with the contents that you have inserted;

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

List all of the students in a given course;

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

Add an assignment to a course;

Change the percentages of the categories for a course;

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

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

Compute the grade for a student;

Compute the grade for a student, where the lowest score for a given category is dropped.

Instructions

The system is implemented in SQL. Please read below to compile and execute the program:

Prerequisites

Install an SQL DBMS (e.g., MySQL) and run on your local machine/remote server.

Creating Tables

Connect to your DBMS using a SQL client (e.g., MySQL Workbench, pgAdmin).

Use the following commands to create their respective tables:

Course table: CREATE TABLE Course (

course\_id INT PRIMARY KEY,

department VARCHAR(255),

course\_number INT,

course\_name VARCHAR(255),

semester VARCHAR(255),

year INT

);

Student table:

CREATE TABLE Student (

student\_id INT PRIMARY KEY,

first\_name VARCHAR(255),

last\_name VARCHAR(255),

course\_id INT,

FOREIGN KEY (course\_id) REFERENCES Course(course\_id)

);

Assignment table:

CREATE TABLE Assignment (

assignment\_id INT PRIMARY KEY,

category VARCHAR(255),

percentage DECIMAL(5,2),

course\_id INT,

FOREIGN KEY (course\_id) REFERENCES Course(course\_id)

);

Grades table:

CREATE TABLE Grades (

grade\_id INT PRIMARY KEY,

assignment\_id INT,

student\_id INT,

score DECIMAL(5,2),

FOREIGN KEY (assignment\_id) REFERENCES

Assignment(assignment\_id),

FOREIGN KEY (student\_id) REFERENCES Student(student\_id)

);

Inserting Values

Use the following commands to insert values into the tables using the following examples:

Insertions for Course table:

INSERT INTO Course (course\_id, department, course\_number, course\_name, semester, year)

VALUES (1, 'CSCI', 432, 'Database Systems', 'Spring', 2024);

| course\_id | department | course\_number | course\_name | semester | year |
| --- | --- | --- | --- | --- | --- |
| 1 | CSCI | 432 | Database Systems | Spring | 2024 |

Insertions for Student table:

INSERT INTO Student (student\_id, first\_name, last\_name, course\_id)

VALUES (1, 'Jamie', 'Stevens', 1),

(2, 'Monae', 'Adams', 1),

(3, 'Alliston', 'Dunn', 1);

| student\_id | first\_name | last\_name | course\_id |
| --- | --- | --- | --- |
| 1 | Jamie | Stevens | 1 |
| 2 | Monae | Adams | 1 |
| 3 | Alliston | Dunn | 1 |
| 4 | Sam | McQueen | 1 |

Insertions for Assignment table:

INSERT INTO Assignment (assignment\_id, category, percentage, course\_id)

VALUES (1, 'Assignments', 15.0, 1),

(2, 'Midterm', 30.0, 1),

(3, 'Project', 15.0, 1),

(4, 'Final Exam', 40.0, 1);

| assignment\_id | category | percentage | course\_id |
| --- | --- | --- | --- |
| 1 | Assignments | 15 | 1 |
| 2 | Midterm | 20 | 1 |
| 3 | Project | 15 | 1 |
| 4 | Final Exam | 40 | 1 |

Insertions for Grades table:

INSERT INTO Grades (grade\_id, assignment\_id, student\_id, score)

VALUES (1, 1, 1, 6.5),

(2, 2, 1, 92.0),

(3, 3, 1, 75.0),

(4, 4, 1, 88.0),

(5, 1, 2, 9.0),

(6, 2, 2, 84.5),

(7, 3, 2, 82.0),

(8, 4, 2, 92.0),

(9, 1, 3, 7.0),

(10, 2, 3, 78.0),

(11, 3, 3, 90.0),

(12, 4, 3, 95.0);

| grade\_id | assignment\_id | studnet\_id | score |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 6.5 |
| 2 | 2 | 1 | 92.0 |
| 3 | 3 | 1 | 75.0 |
| 4 | 4 | 1 | 88.0 |
| 5 | 1 | 2 | 9.0 |
| 6 | 2 | 2 | 84.5 |
| 7 | 3 | 2 | 82.0 |
| 8 | 4 | 2 | 92.0 |
| 9 | 1 | 3 | 7.0 |
| 10 | 2 | 3 | 78.0 |
| 11 | 3 | 3 | 90.0 |
| 12 | 4 | 3 | 95.0 |

Test cases and Result:

1. Compute the average/highest/lowest score of an assignment;

Test case:

assignment\_id=2,

Output:

average\_score=83.125

highest\_score=92

lowest\_score=78

1. List all of the students in a given course;

Test Case: course\_id=1

Output:

| student\_id | first\_name | last\_name | course\_id |
| --- | --- | --- | --- |
| 1 | Jamie | Stevens | 1 |
| 2 | Monae | Adams | 1 |
| 3 | Alliston | Dunn | 1 |
| 4 | Sam | McQueen | 1 |

1. List all of the students in a course and all of their scores on every assignment;

Test Case:

Course\_id=1

Output:

| first\_name | last\_name | assignment\_id | category | score |
| --- | --- | --- | --- | --- |
| Monae | Adams | 1 | Assignments | 9 |
| Monae | Adams | 2 | Midterm | 84.5 |
| Monae | Adams | 3 | Project | 82 |
| Monae | Adams | 4 | Final | 92 |
| Alliston | Dunn | 1 | Assignments | 7 |
| Alliston | Dunn | 2 | Midterm | 78 |
| Alliston | Dunn | 3 | Project | 90 |
| Alliston | Dunn | 4 | Final | 95 |
| Sam | McQueen | 2 | Midterm | 78 |
| Sam | McQueen | 3 | Project | 70 |
| Sam | McQueen | 4 | Final | 85 |
| Jamie | Stevens | 1 | Assignments | 6.5 |
| Jamie | Stevens | 2 | Midterm | 92 |
| Jamie | Stevens | 3 | Project | 75 |
| Jamie | Stevens | 4 | Final | 88 |

1. Add an assignment to a course;

Testcase

course\_id=1

Output

| assignment\_id | category | percentage | course\_id |
| --- | --- | --- | --- |
| 1 | Assignments | 15 | 1 |
| 2 | Midterm | 30 | 1 |
| 3 | Project | 15 | 1 |
| 4 | Final Exam | 40 | 1 |
| 5 | Extra Credit | 10 | 1 |

1. Change the percentages of the categories for a course;

Test case

category = 'Assignments', course\_id = 1

Output

| assignment\_id | category | percentage | course\_id |
| --- | --- | --- | --- |
| 1 | Assignments | 30 | 1 |
| 2 | Midterm | 30 | 1 |
| 3 | Project | 15 | 1 |
| 4 | Final Exam | 40 | 1 |
| 5 | Extra Credit | 10 | 1 |

1. Add 2 points to the score of each student on an assignment;

Testcase

assignment\_id=3

Output

| first\_name | last\_name | assignment\_id | category | score |
| --- | --- | --- | --- | --- |
| Monae | Adams | 1 | Assignments | 9 |
| Monae | Adams | 2 | Midterm | 84.5 |
| Monae | Adams | 3 | Project | 84 |
| Monae | Adams | 4 | Final | 92 |
| Alliston | Dunn | 1 | Assignments | 7 |
| Alliston | Dunn | 2 | Midterm | 78 |
| Alliston | Dunn | 3 | Project | 92 |
| Alliston | Dunn | 4 | Final | 95 |
| Sam | McQueen | 2 | Midterm | 78 |
| Sam | McQueen | 3 | Project | 72 |
| Sam | McQueen | 4 | Final | 85 |
| Jamie | Stevens | 1 | Assignments | 6.5 |
| Jamie | Stevens | 2 | Midterm | 92 |
| Jamie | Stevens | 3 | Project | 77 |
| Jamie | Stevens | 4 | Final | 88 |

1. Add 2 points just to those students whose last name contains a ‘Q’.

Testcase

assignment\_id=1

Output

1. Compute the grade for a student;

Testcase

student\_id=1

Output

| first\_name | last\_name | grade |
| --- | --- | --- |
| Jamie | Stevens | 74.95 |

1. Compute the grade for a student, where the lowest score for a given category is dropped.

Testcase

student\_id=2

Output

| first\_name | last\_name | grade |
| --- | --- | --- |
| Monae | Adams | 5.7 |