King Fahd University of Petroleum & Minerals
Information and Computer Science Department

ICS324 (Database Systems)

Term: 172

Recommendation Letters Database System

Phase III

Team #9

Doctor in Charge:

Dr. Yahya Garout

Team Members:

Name	ID
Yousef Majeed	201568070
Yasir Alabas	201593610
Omar Lajam	201475920

Friday, May 4, 2018

"9th Team" ® - *"Success is a journey not a destination!"*

Table of Contents

Recommendation Letters Database System

1. Problem Statement

Write the database requirements and the draw the ER diagram for the following described system:

Instructors frequently need to write a recommendation letters to their students. It hard for an instructor to find details of the courses taught to the recommended student. Develop a database system to help the instructors in recommendations. The data requirements are summarized as follows:

- Students are identified by a unique student id, their first and last names, and a major.
- The instructor teaches certain courses each term. The courses are uniquely identified by a course number, a section number, and the term in which they are taught. The instructor also assigns grade cutoffs for letter grades for each course he teaches.
- Students are enrolled in courses taught by the instructor.
- Each course being taught by the instructor has a number of grading components (such as mid-term, final exam, project, etc.). Each grading component has a maximum number of points (such as 100 or 50) and a weight (such as 20% or 10%). The weights of all the grading components of a course usually add up to 100.
- Finally, the instructor records the points earned by each student in each of the grading components in each of the courses. For example, student with id=201112340 earns 84 points for the grading component mid-term for the course ICS324 section 2 in the 142 term. The mid-term grading component may have been defined to have a maximum of 100 points and a weight of 20% of the course grade.

2. Database Requirements

2.1 Entities, Their Attributes and Constraints:

The following table shows the entities with their attributes of the conceptual model:

Entities	Attributes
Instructor	instructorID, fistName, lastName
Student	studentID, firstName, lastName, major
Course	courseNumber, sectionNumber
Grading_Component	componentName, maxPoints, weight
Grades_Cutoffs	LetterGrade, Score

Constrains of the model:

- INSTRUCTOR have Instructor ID (Key), Instructor First Name, Last Name.
- STUDENT have Student ID (Key), Student First Name, Last Name, Major.
- COURSE have Course Number (Key), Section Number.
- GRADING COMPONENT have Component Name (Key), Maximum Point, Weight.
- GRADE CUTOFFS have Letter Grade (Key), Score.

2.2 Relationships, Their Attributes and Constraints:

1. INSTRUCTOR **TEACH** a COURSE
   Attributes: Term (xxx) (e.g. 172)
   Constraints: AN INSTRUCTOR can teach in more than one COURSE and a COURSE can have more than one INSTUCTOR.
   (many to many cardinality and partial participation of both entities)

2. STUDENT **ENROLL** in a COURSE
   Constraints: A STUDENT can enroll in more than one COURSE and a COURSE can be enrolled by more than one STUDENT.
   (many to many cardinality and partial participation of both entities)

3. INSTRUCTOR **ASSIGN** GRADE_CUTOFFS for a COURSE
   Attributes: Term (xxx) (e.g. 172)
   Constraints: Constraints are one to many to one cardinality and, total participation of GRADE_CUTOFFS and partial for the rest.
   (one to many to one cardinality and, total participation of GRADECUTOFFS and partial for the rest)

4. COURSE **HAS** GRADING_COMPONENT
   Constraints: A COURSE can have more than one GRADING_COMPONENT, but every GRADING_COMPONENT must belong only to one COURSE.
   (one to many cardinality and, partial participation of COURSE, total for GRADING_COMPONENT)

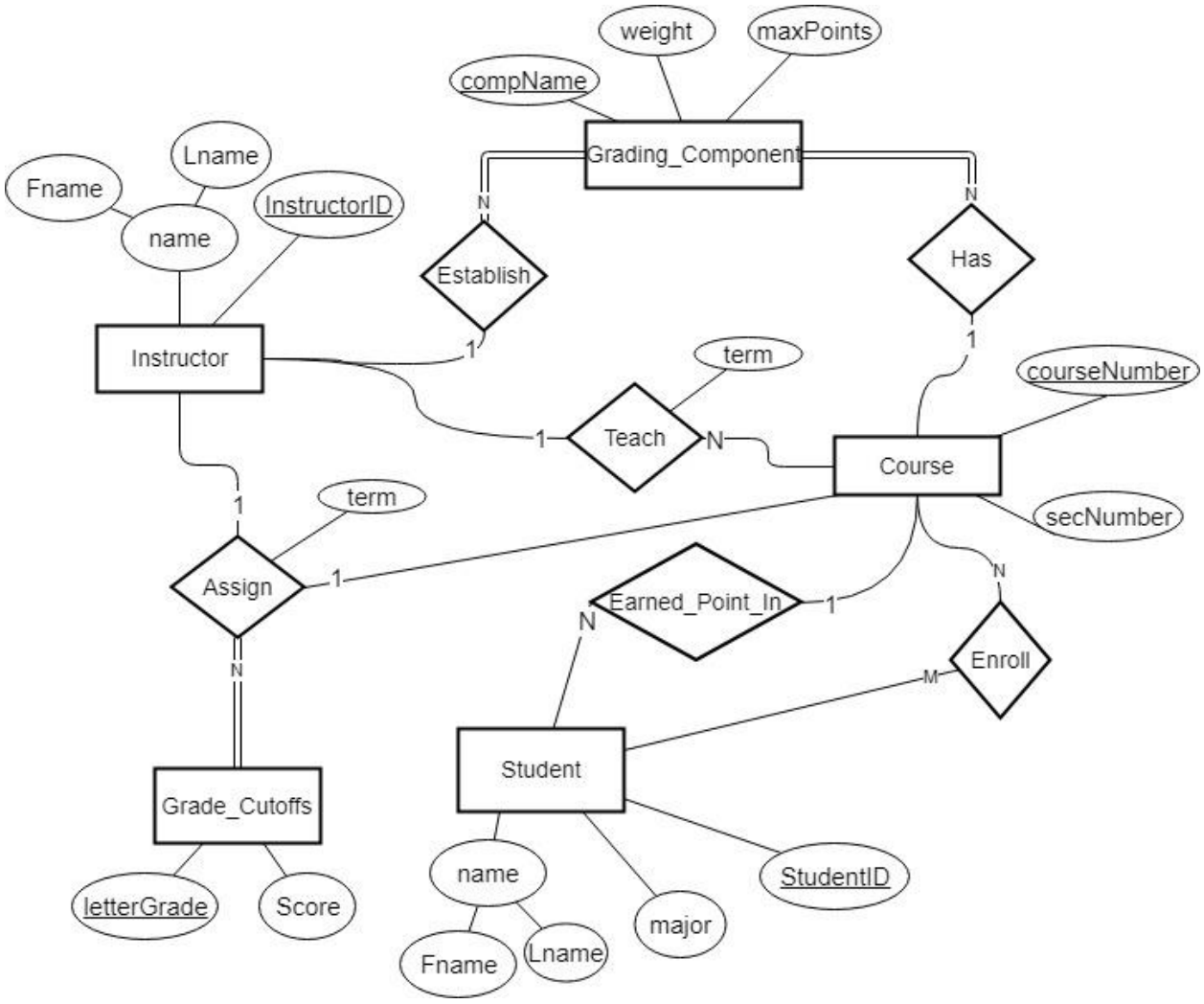
5. INSTRUCTOR **ESTABLISH** a GRADING_COMPONENT
   Constraints: AN INSTRUCTOR can establish more than one GREADING_COMPONENT and a GREADING_COMPONENT can be established by only one INSTRUCTOR.
   (one to many cardinality and total participation for GREADING_COMPONENT, partial for the rest)

6. STUDENT **EARNED_POINT_IN** a specific COURSE
   Constraints: A STUDENT can earn point only in one specific COURSE and points can be earned in a specific COURSE by more than one STUDENT.
   (many to one cardinality and, partial participation of both entities).

Relation	Between	Type	Participation	Attributes
Teach	Instructor and Course	M: N (many-to-many)	Both are partial	term
Enroll	Student and Course	M: N (many-to-many)	Both are partial	-
Assign	Instructor, Grade_Cutoffs and Course	1: N: 1 (one-to-many-to-one)	Total for G_Cut, partial for the rest	term
Has	Course and Grading_Component	1: N (one-to-many)	Total for G_Co, partial for Course	-
Establish	Instructor and Grading_Component	1: N (one-to-many)	Total for G_Co, partial for Instructor	-
Earend_Point_In	Student and Course	N: 1 (many-to-one)	Both are partial	-

3. ER/EER Model

The following figure shows the ER diagram of database:



4. Relational Schema (all relations and constraints)

- INSTRUCTOR (InstructorID , Fname, Lname)
- STUDENT (StudentID, Fname, Lname, major)
- COURSE (courseNumber, secNumber)
- GRADING COMPONENT (compName, maxPoints, weight)
- GRADE CUTOFFS (letterGrade, Score)
- TEACH (InstructorID, courseNumber, term)
- ASSIGN (letterGrade, courseNumber, InstructorID, term)

5. DDL Statements to Create Database Tables

```
CREATE TABLE INSTRUCTOR (
InstructorID NUMBER (9) NOT NULL,
Fname VARCHAR (10) NOT NULL,
Lname VARCHAR (10) NOT NULL,
CONSTRAINT IPK PRIMARY KEY (InstructorID));
```

```
CREATE TABLE STUDENT (
StudentID NUMBER (9)    NOT NULL,
Fname VARCHAR(10)       NOT NULL,
Lname VARCHAR(10)       NOT NULL,
major VARCHAR(4),
CONSTRAINT SPM PRIMARY KEY(StudentID));
```

```
CREATE TABLE COURSE (
courseNumber NUMBER (4) NOT NULL,
secNumber NUMBER (3),
CONSTRAINT CPM PRIMARY KEY(courseNumber));
```

```
CREATE TABLE GRADING_COMPONENT (
compName VARCHAR(10)   NOT NULL,
maxPoint NUMBER (4),
weight NUMBER (3),
CONSTRAINT GCOPM PRIMARY KEY(compName));
```

```
CREATE TABLE GRADE_CUTOFFS (
letterGrade VARCHAR (2)   NOT NULL,
Score NUMBER (4),
CONSTRAINT GCUPM PRIMARY KEY(letterGrade));
```

```
CREATE TABLE TEACH (
InstructorID NUMBER (9)   NOT NULL,
courseNumber NUMBER (4),
term NUMBER (3),
CONSTRAINT TPM PRIMARY KEY(InstructorID, courseNumber));
```

```
CREATE TABLE ASSIGN (
letterGrade VARCHAR (2)   NOT NULL,
courseNumber NUMBER (4),
InstructorID NUMBER (9),
term NUMBER (3),
CONSTRAINT APM PRIMARY KEY(InstructorID, courseNumber, letterGrade));
```

```
ALTER TABLE TEACH
ADD CONSTRAINT AFK FOREIGN KEY (InstructorID) REFERENCES
INSTRUCTOR(InstructorID)
ON DELETE CASCADE;
```

```
ALTER TABLE ASSIGN
ADD CONSTRAINT AFK FOREIGN KEY (letterGrade) REFERENCES
GRADE_CUTOFFS (letterGrade)
ON DELETE CASCADE;
```

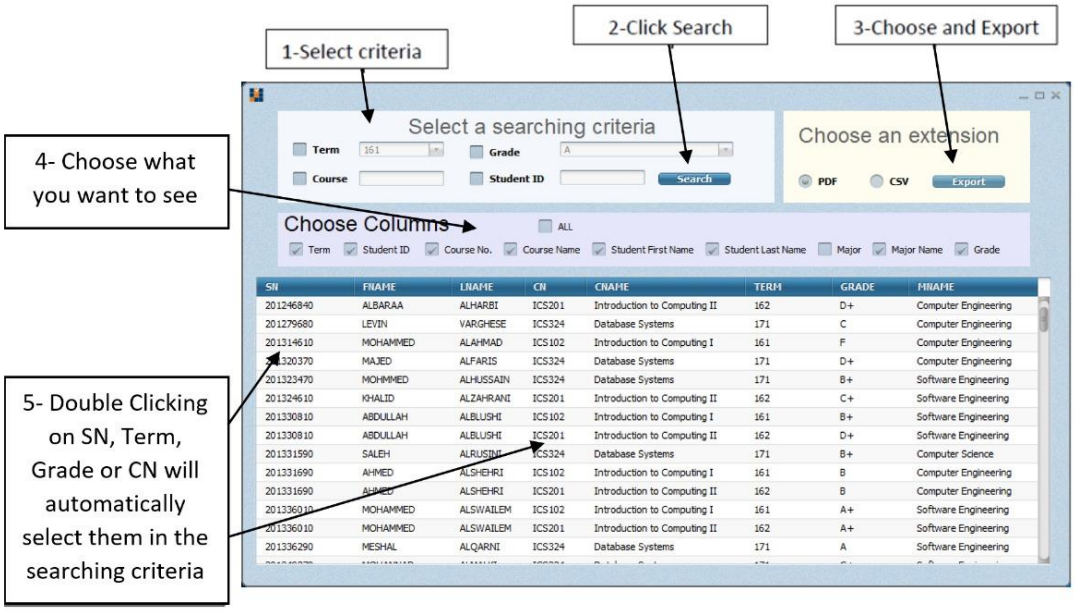
6. Tools and Language Used

As for the implementation of the front-end, we used Java programming language and the following tools:

- Eclipse IDE
- Window Builder

7. The User Manual

The following figure illustrates how to use the main functionalities of the system:



SN	FNAME	LNAME	CN	CNAME	TERM	GRADE	MNAME
201246840	ALBARAA	ALHARBI	ICS201	Introduction to Computing II	162	D+	Computer Engineering
201279680	LEVIN	VARGHESE	ICS324	Database Systems	171	C	Computer Engineering
201314610	MOHAMMED	ALAHMAD	ICS102	Introduction to Computing I	161	F	Computer Engineering
201320370	MAJED	ALFARIS	ICS324	Database Systems	171	D+	Computer Engineering
201323470	MOHMMED	ALHUSSAIN	ICS324	Database Systems	171	B+	Software Engineering
201324610	KHALID	ALZAHRANI	ICS201	Introduction to Computing II	162	C+	Software Engineering
201330810	ABDULLAH	ALBLUSHI	ICS102	Introduction to Computing I	161	B+	Software Engineering
201330810	ABDULLAH	ALBLUSHI	ICS201	Introduction to Computing II	162	D+	Software Engineering
201331590	SALEH	ALRUSINI	ICS324	Database Systems	171	B+	Computer Science
201331690	AHMED	ALSHEHRI	ICS102	Introduction to Computing I	161	B	Computer Engineering
201331690	AHMED	ALSHEHRI	ICS201	Introduction to Computing II	162	B	Computer Engineering
201336010	MOHAMMED	ALSWAILEM	ICS102	Introduction to Computing I	161	A+	Software Engineering
201336010	MOHAMMED	ALSWAILEM	ICS201	Introduction to Computing II	162	A+	Software Engineering
201336290	MESHAL	ALQARNI	ICS324	Database Systems	171	A	Software Engineering

8. Conclusion

The IT solutions always solve main problems that people face every day. In this project, we have solved the problem of finding information related to student who are taught by instructors in universities in purpose of writing recommendations for them. During the implementation of this project, we have learned many skills and we widen our knowledge in the fields of programming and databases.

9. Distribution of Work

The following table shows the percentage of contribution in the work done by the team members during the execution of the project*:

Team Member	Phase I	Phase II	Phase III
Yousef Majeed	33.3%	96%	85%
Yasir Alabas	33.3%	2%	5%
Omar Lajam	33.3%	2%	10%

*Note: all team members were aware of what is happing and verify the work.