

**King Fahd University of Petroleum and Minerals**

**Information and Computer Science Department**

**ICS 324**

**Term 143**

**Phase 1: Conceptual and Logical Design**

|  |  |
| --- | --- |
| Member | ID |
| Ibrahim BinAlshikh | 201139750 |
| Ibrahim | 201224780 |
| Aqeel | 201236760 |

Table of Contents

[1. Problem Statement 2](#_Toc426292783)

[2. Assumptions 2](#_Toc426292784)

[3. Database Requirements (defines the entities, attributes, relationships and constraints). 2](#_Toc426292785)

[4. Conceptual schema (ER/EER Model) 3](#_Toc426292786)

[5. Relational schema (all relations and constraints) 3](#_Toc426292787)

[6. DDL statements to create database tables 4](#_Toc426292788)

# Problem Statement

We have to design and implement a database that can be used by academic institutions to record the grades of students in specific course. To design this, we have to consider 3 main actors: student, instructor and a course. Each of those will affect the final product.

# Assumptions

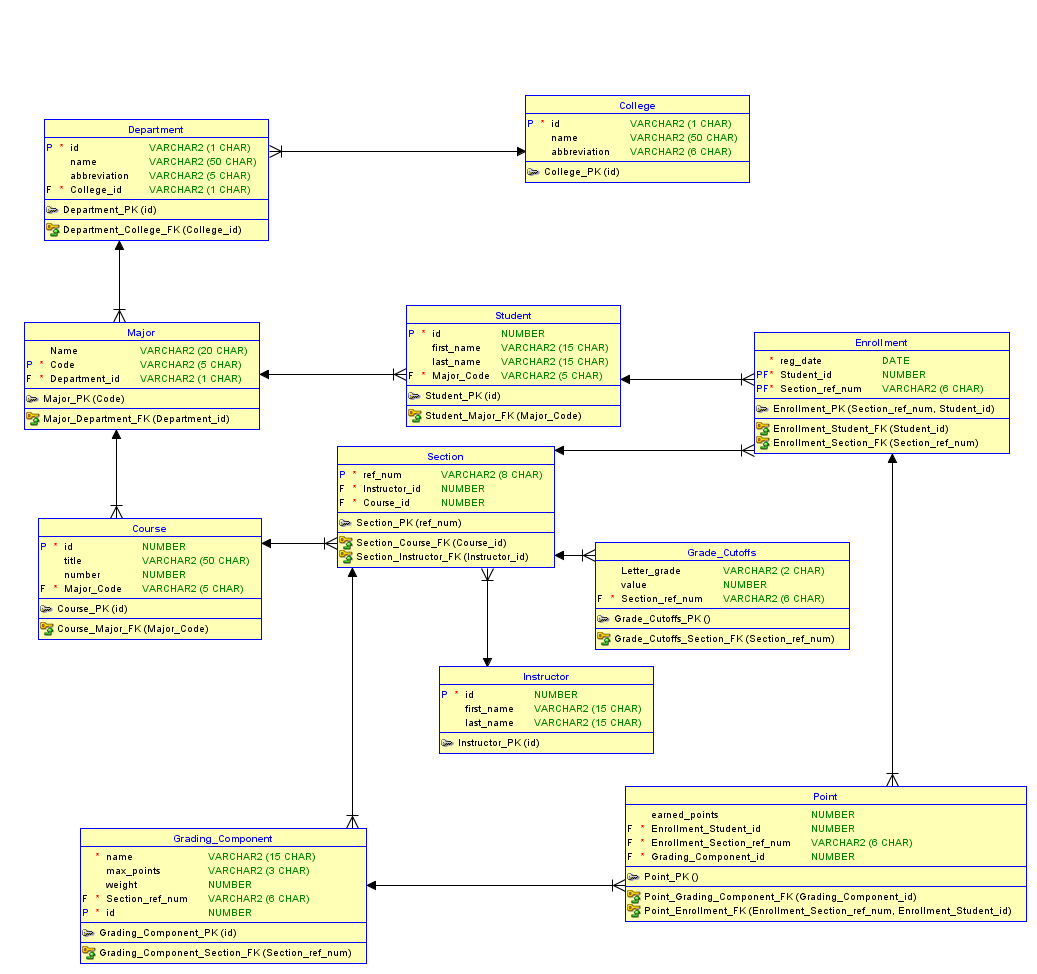
* 1. We will assume that the database will be made for KFUPM use.
  2. We will assume that the length of names will not exceed 15 characters.
  3. We will assume that the condition for most relations on delete will be ‘set null’.
  4. When a section is removed, all students will be removed from that section.
  5. Assume that each course will have at least 2 grading component.
  6. Assume that the number of grading component for all courses won’t exceed 1000.
  7. Assume that each college and department has an alphabetic unique ID that represents it.
  8. Assume that the section reference number will be formed as follows: College ID + Department ID + course number + section number + term number(e.g. AA1111-53-1)
  9. Assume that every year the section reference number is recycled.
  10. Each section will be taught by one instructor.
  11. Assume that the sum of all points on a section can be less than 100 or more.

# Database Requirements (defines the entities, attributes, relationships and constraints).

We can summarize the database requirements as follows:

1. Students’ first and last names and majors shall be recorded and be identified with a unique student id
2. Courses’ number shall be recorded and it shall be identified with a unique course id
3. Each section must belong to a course in a specific term and shall be identified by a unique section number and taught by an instructor
4. The section is made up from several grading components where each has a maximum points and an overall weight
5. Sections must have enrolled students
6. The instructor will be able to record points for students
7. Instructors shall be able to assign letter grades and grade cutoffs for each of his section

# Conceptual schema (ER/EER Model)



# Relational schema (all relations and constraints)

1. Every college will have many departments
2. Each student will have a major
3. Each student can register in many sections
4. One instructor can teaches many sections
5. The instructor can create grading components.
6. Each course will have many grading components.
7. Each enrollment will have many grading components.

# DDL statements to create database tables

-- Generated by Oracle SQL Developer Data Modeler 4.1.1.888

-- at: 2015-08-03 23:37:09 AST

-- site: Oracle Database 12c

-- type: Oracle Database 12c

CREATE TABLE College

(

id VARCHAR(1) NOT NULL ,

name VARCHAR (50) ,

abbreviation VARCHAR(5)

) ;

ALTER TABLE College ADD CONSTRAINT College\_PK PRIMARY KEY ( id ) ;

CREATE TABLE Course

(

"number" numeric(4) NOT NULL ,

title VARCHAR (50) ,

"level" numeric(3) ,

Major\_code VARCHAR (4) NOT NULL

) ;

ALTER TABLE Course ADD CONSTRAINT Course\_PK PRIMARY KEY ( "number" ) ;

CREATE TABLE Department

(

id VARCHAR (1) NOT NULL ,

name VARCHAR (50) ,

abbreviation VARCHAR (5 ) ,

College\_id VARCHAR (1 ) NOT NULL

) ;

ALTER TABLE Department ADD CONSTRAINT Department\_PK PRIMARY KEY ( id ) ;

CREATE TABLE Enrollment

(

reg\_date DATE ,

Student\_id numeric(9) NOT NULL ,

Section\_refrence\_number VARCHAR (11) NOT NULL

) ;

ALTER TABLE Enrollment ADD CONSTRAINT Enrollment\_PK PRIMARY KEY ( Student\_id, Section\_refrence\_number ) ;

CREATE TABLE Grading\_Component

(

id numeric(3) NOT NULL ,

name VARCHAR (15) ,

max\_points numeric(3) ,

weight numeric(2) ,

Course\_number numeric(4) NOT NULL ,

Instructor\_id numeric(9) NOT NULL

) ;

ALTER TABLE Grading\_Component ADD CONSTRAINT Grading\_Component\_PK PRIMARY KEY ( id ) ;

CREATE TABLE Instructor

(

id numeric(9) NOT NULL ,

first\_name VARCHAR (15) ,

last\_name VARCHAR (15)

) ;

ALTER TABLE Instructor ADD CONSTRAINT Instructor\_PK PRIMARY KEY ( id ) ;

CREATE TABLE Major

(

name VARCHAR(50) ,

code VARCHAR(4) NOT NULL ,

Department\_id VARCHAR (1) NOT NULL

) ;

ALTER TABLE Major ADD CONSTRAINT Major\_PK PRIMARY KEY ( code ) ;

CREATE TABLE Point

(

earned\_Points numeric(3) ,

Grading\_Component\_id numeric(3) NOT NULL ,

Enrollment\_Student\_id numeric(9) NOT NULL ,

Enrollment\_refrence\_number VARCHAR (11) NOT NULL

) ;

CREATE TABLE Section

(

refrence\_number VARCHAR (11) NOT NULL ,

"number" numeric(2) ,

Instructor\_id numeric(9) NOT NULL

) ;

ALTER TABLE Section ADD CONSTRAINT Section\_PK PRIMARY KEY ( refrence\_number ) ;

CREATE TABLE Student

(

id numeric(9) NOT NULL ,

first\_name VARCHAR (15) ,

last\_name VARCHAR (15) ,

Major\_code VARCHAR (4) NOT NULL

) ;

ALTER TABLE Student ADD CONSTRAINT Student\_PK PRIMARY KEY ( id ) ;

ALTER TABLE Course ADD CONSTRAINT Course\_Major\_FK FOREIGN KEY ( Major\_code ) REFERENCES Major ( code ) ON

DELETE CASCADE ;

ALTER TABLE Department ADD CONSTRAINT Department\_College\_FK FOREIGN KEY ( College\_id ) REFERENCES College ( id ) ;

ALTER TABLE Enrollment ADD CONSTRAINT Enroll\_Section\_FK FOREIGN KEY ( Section\_refrence\_number ) REFERENCES Section ( refrence\_number ) ON

DELETE CASCADE ;

ALTER TABLE Enrollment ADD CONSTRAINT Enroll\_Student\_FK FOREIGN KEY ( Student\_id ) REFERENCES Student ( id ) ;

ALTER TABLE Grading\_Component ADD CONSTRAINT Grading\_Comp\_Course\_FK FOREIGN KEY ( Course\_number ) REFERENCES Course ( "number" ) ON

DELETE CASCADE ;

ALTER TABLE Grading\_Component ADD CONSTRAINT Grading\_Comp\_Inst\_FK FOREIGN KEY ( Instructor\_id ) REFERENCES Instructor ( id ) ON

DELETE CASCADE ;

ALTER TABLE Major ADD CONSTRAINT Major\_Department\_FK FOREIGN KEY ( Department\_id ) REFERENCES Department ( id ) ;

ALTER TABLE Point ADD CONSTRAINT Point\_Enrollment\_FK FOREIGN KEY ( Enrollment\_Student\_id, Enrollment\_refrence\_number ) REFERENCES Enrollment ( Student\_id, Section\_refrence\_number ) ON

DELETE CASCADE ;

ALTER TABLE Point ADD CONSTRAINT Point\_Grading\_Comp\_FK FOREIGN KEY ( Grading\_Component\_id ) REFERENCES Grading\_Component ( id ) ON

DELETE CASCADE ;

ALTER TABLE Section ADD CONSTRAINT Section\_Instructor\_FK FOREIGN KEY ( Instructor\_id ) REFERENCES Instructor ( id ) ;

ALTER TABLE Student ADD CONSTRAINT Student\_Major\_FK FOREIGN KEY ( Major\_code ) REFERENCES Major ( code ) ;