

**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 | 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

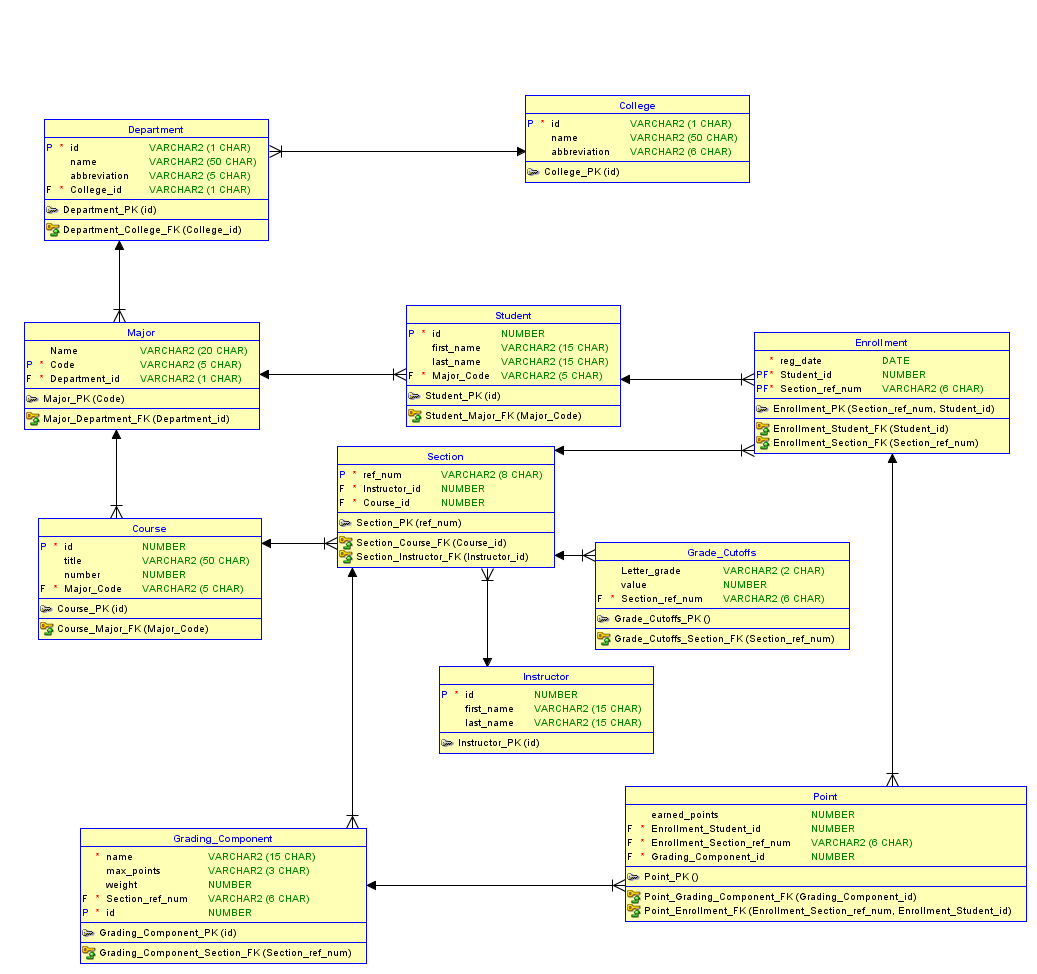
* 1. We will assume that the database will be made for KFUPM use.
  2. We will assume that the length of names will not exceed 15 characters.
  3. We will assume that the condition on delete will be ‘set null’.
  4. Assume that each course will have at least 2 grading component.
  5. Assume that the number of grading component for all courses won’t exceed 1000.
  6. Assume that each college and department has an alphabetic unique ID that represents it.
  7. Assume that the section reference number will be formed as follows: College ID + Department ID + course number + term number + section number.
  8. Assume that every year the section reference number is recycled.
  9. Each section will be taught by one instructor.
  10. 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)

# DDL statements to create database tables

CREATE TABLE College

(

id VARCHAR2 (1 CHAR) NOT NULL ,

name VARCHAR2 (50 CHAR) ,

abbreviation VARCHAR2 (6 CHAR)

) ;

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

CREATE TABLE Course

(

id NUMBER NOT NULL ,

title VARCHAR2 (50 CHAR) ,

"number" NUMBER ,

Major\_Code VARCHAR2 (5 CHAR) NOT NULL

) ;

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

CREATE TABLE Department

(

id VARCHAR2 (1 CHAR) NOT NULL ,

name VARCHAR2 (50 CHAR) ,

abbreviation VARCHAR2 (5 CHAR) ,

College\_id VARCHAR2 (1 CHAR) NOT NULL

) ;

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

CREATE TABLE Enrollment

(

reg\_date DATE NOT NULL ,

Student\_id NUMBER NOT NULL ,

Section\_ref\_num VARCHAR2 (6 CHAR) NOT NULL

) ;

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

CREATE TABLE Grade\_Cutoffs

(

Letter\_grade VARCHAR2 (2 CHAR) ,

value NUMBER ,

Section\_ref\_num VARCHAR2 (6 CHAR) NOT NULL

) ;

CREATE TABLE Grading\_Component

(

name VARCHAR2 (15 CHAR) NOT NULL ,

max\_points VARCHAR2 (3 CHAR) ,

weight NUMBER ,

Section\_ref\_num VARCHAR2 (6 CHAR) NOT NULL ,

id NUMBER NOT NULL

) ;

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

CREATE TABLE Instructor

(

id NUMBER NOT NULL ,

first\_name VARCHAR2 (15 CHAR) ,

last\_name VARCHAR2 (15 CHAR)

) ;

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

CREATE TABLE Major

(

Name VARCHAR2 (20 CHAR) ,

Code VARCHAR2 (5 CHAR) NOT NULL ,

Department\_id VARCHAR2 (1 CHAR) NOT NULL

) ;

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

CREATE TABLE Point

(

earned\_points NUMBER ,

Enrollment\_Student\_id NUMBER NOT NULL ,

Enrollment\_Section\_ref\_num VARCHAR2 (6 CHAR) NOT NULL ,

Grading\_Component\_id NUMBER NOT NULL

) ;

CREATE TABLE Section

(

ref\_num VARCHAR2 (8 CHAR) NOT NULL ,

Instructor\_id NUMBER NOT NULL ,

Course\_id NUMBER NOT NULL

) ;

ALTER TABLE Section ADD CONSTRAINT Section\_PK PRIMARY KEY ( ref\_num ) ;

CREATE TABLE Student

(

id NUMBER NOT NULL ,

first\_name VARCHAR2 (15 CHAR) ,

last\_name VARCHAR2 (15 CHAR) ,

Major\_Code VARCHAR2 (5 CHAR) 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 ) ;

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

ALTER TABLE Enrollment ADD CONSTRAINT Enrollment\_Section\_FK FOREIGN KEY ( Section\_ref\_num ) REFERENCES Section ( ref\_num ) ;

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

ALTER TABLE Grade\_Cutoffs ADD CONSTRAINT Grade\_Cutoffs\_Section\_FK FOREIGN KEY ( Section\_ref\_num ) REFERENCES Section ( ref\_num ) ;

ALTER TABLE Grading\_Component ADD CONSTRAINT Grading\_Component\_Section\_FK FOREIGN KEY ( Section\_ref\_num ) REFERENCES Section ( ref\_num ) ;

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\_Section\_ref\_num, Enrollment\_Student\_id ) REFERENCES Enrollment ( Section\_ref\_num, Student\_id ) ;

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

ALTER TABLE Section ADD CONSTRAINT Section\_Course\_FK FOREIGN KEY ( Course\_id ) REFERENCES Course ( id ) ;

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 ) ;