Kevin Nolasco

MCIS 510

Milestone 2

Conceptual, Logical, and Physical Models

Introduction:

For this project, I will be using the narrative provided in the course description for the School of Natural Sciences and Allied Health at Cabrini University. The narrative is provided as follows (any added business rules will be in bold):

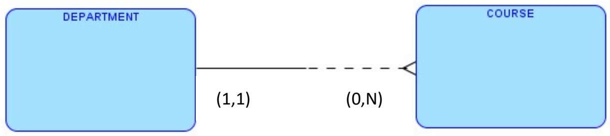
"School of Natural Sciences and Allied Health at Cabrini University contains many departments. Each department may offer any number of courses. ***A course must belong to exactly one department.*** *A department may have many instructors, but an instructor can work only in one department. For each department, there is a head, and an instructor can be head of only one department. Each instructor can teach any number of courses, and a course can be taken by only one instructor.* ***Each course can generate a class. Not all classes are offered every semester.*** *A student can enroll for any number of courses and each course can have any number of students.* ***Each department can have any number of students. A student can be a part of only one department****.* ***Each department can offer any number of majors. Many students can declare any combination of degree type and major. Each class is held in one room. Each room can hold many classes. Each room belongs to one building. An instructor has their office in one room. A department is in one room.****"*

Construction of the Conceptual Model:

We will consider one business rule at a time to construct the conceptual model.

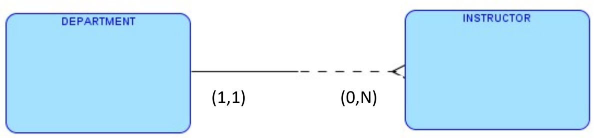
1) "*School of Natural Sciences and Allied Health at Cabrini University contains many departments. Each department may offer any number of courses.* ***A course must belong to exactly one department.***"

   This rule requires two entities: DEPARTMENT and COURSE. There is a 1:M relationship between DEPARTMENT and COURSE. COURSE is optional to DEPARTMENT because not every department will offer courses. The cardinality from DEPARTMENT to COURSE is (0,N). The cardinality from COURSE to DEPARTMENT is (1,1). (NOTE: In this model, an optional entity is connected by a dashed line).



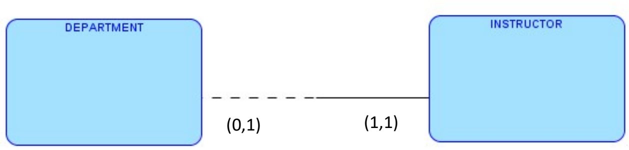
2) *" A department may have many instructors, but an instructor can work only in one department."*

This rule requires two entities: DEPARTMENT and INSTRUCTOR. There is a 1:M relationship between DEPARTMENT and INSTRUCTOR. INSTRUCTOR is optional to DEPARTMENT because some departments may not have any instructors. The cardinality from DEPARTMENT to INSTRUCTOR is (0,N). The cardinality from INSTRUCTOR to DEPARTMENT is (1,1).



3) *"For each department, there is a head, and an instructor can be head of only one department."*

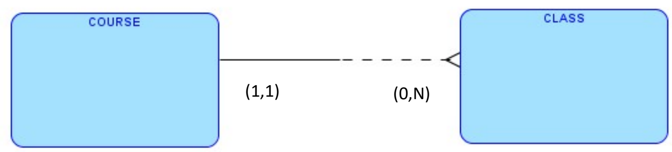
This rule also consists of DEPARTMENT and INSTRUCTOR. There is a 1:1 relationship between DEPARTMENT and INSTRUCTOR. DEPARTMENT is optional to INSTRUCTOR because not all instructors are the head of a department. The cardinality from DEPARTMENT to INSTRUCTOR is (1,1). The cardinality from INSTRUCTOR to DEPARTMENT is (0,1).



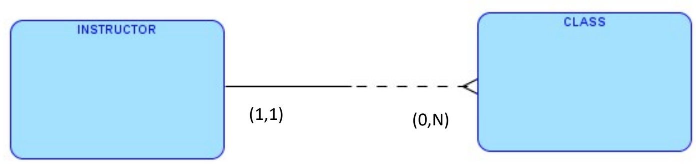
4) *"Each instructor can teach any number of courses, and a course can be taken by only one instructor.* ***Each course can generate a class. Not all classes are offered every semester.****"*

This rule requires four entities: INSTRUCTOR, COURSE, CLASS, SEMESTER. First, we consider the relationship between COURSE and CLASS. There is a 1:M relationship between COURSE and CLASS. Each course may create any number of classes. Each class is a part of exactly one course.

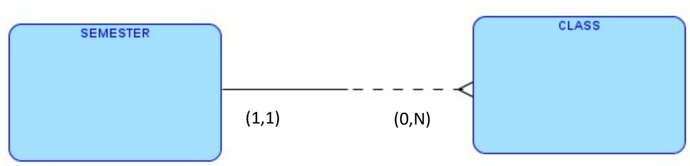
The cardinality between COURSE and CLASS is (0,N). The cardinality between CLASS and COURSE is (1,1).



Next, we consider the relationship between INSTRUCTOR and CLASS. I am making the assumption that the business rule is meant to say "Each instructor can teach any number of ***classes***…" There is a 1:M relationship between INSTRUCTOR and CLASS. The cardinality between INSTRUCTOR and CLASS is (0,N). The cardinality between CLASS and INSTRUCTOR is (1,1).

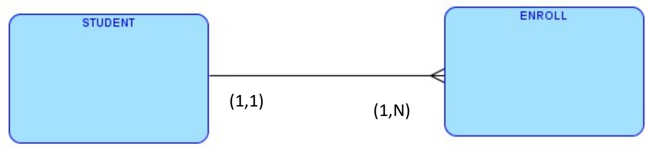


Finally, we consider the relationship between SEMESTER and CLASS. There is a 1:M relationship between SEMESTER and CLASS. The cardinality between SEMESTER and CLASS is (0,N). The cardinality between CLASS and SEMESTER is (1,1).

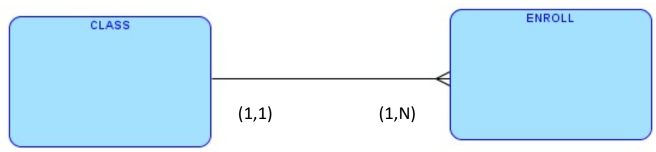


5) *"A student can enroll for any number of courses and each course can have any number of students."* This rule requires three entities because two of the entities have a M:N relationship (A student can enroll into many classes. A class can have many students enrolled in it). To bridge this M:N relationship, we will consider the STUDENT, ENROLL, and CLASS entities. First, we consider the relationship between STUDENT and ENROLL. There is a 1:M relationship between STUDENT and

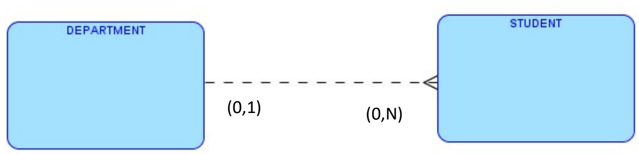
ENROLL. The cardinality of STUDENT to ENROLL is (1,N). The cardinality of ENROLL to STUDENT is (1,1).



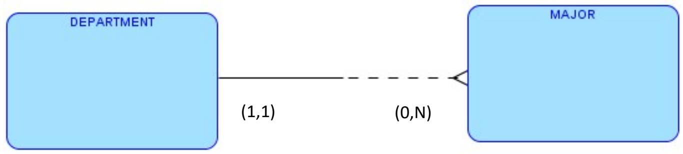
Now, lets consider the relationship between CLASS and ENROLL. There is a 1:M relationship between CLASS and ENROLL. The cardinality of CLASS to ENROLL is (1,N). The cardinality of ENROLL to CLASS is (1,1).



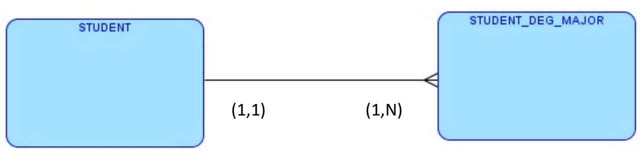
6) "*Each department can have any number of students. A student can be a part of only one department.*" This rule requires two entities: DEPARTMENT and STUDENT. There is a 1:M relationship between DEPARTMENT and STUDENT. The cardinality for DEPARTMENT to STUDENT is (0,N) - this is because a department may or may not deal with any students. The cardinality for STUDENT to DEPARTMENT is (0,1) – a student may belong to one or no department.



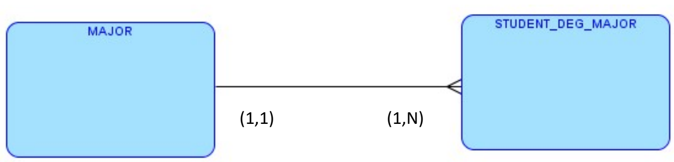
7) *"Each department can offer any number of majors."* This rule requires two entities: DEPARTMENT and MAJOR. There is a 1:M relationship between DEPARTMENT and MAJOR. The cardinality for DEPARTMENT to MAJOR is (0,N) – a department may offer many majors or no majors. The cardinality for MAJOR to DEPARTMENT is (1,1) – each major is run by exactly one department.



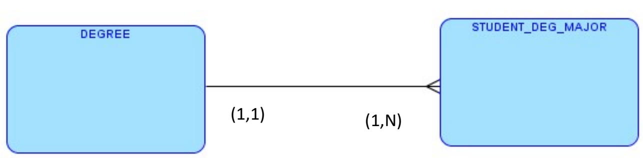
8) *"Many students can declare any combination of degree type and major."* This rule requires four entities because each entity has an M:N relationship with another. The four entities used in this relationship will be: STUDENT, MAJOR, DEGREE, STUDENT_DEG_MAJOR. First, let's consider the relationship between STUDENT and STUDENT_DEG_MAJOR. There is a 1:M relationship between STUDENT and STUDENT_DEG_MAJOR. The cardinality for STUDENT to STUDENT_DEG_MAJOR is (1,N) – a student must have a major and degree type and they may declare more than one. The cardinality for STUDENT_DEG_MAJOR and STUDENT is (1,1).



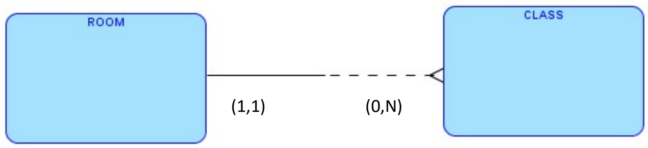
Next, we consider the relationship between MAJOR and STUDENT_DEG_MAJOR. There is a 1:M relationship between MAJOR and STUDENT_DEG_MAJOR. The cardinality for MAJOR to STUDENT_DEG_MAJOR is (1,N). The cardinality for STUDENT_DEG_MAJOR and MAJOR is (1,1).



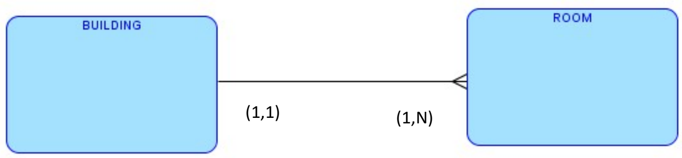
Finally, we consider the relationship between DEGREE and STUDENT_DEG_MAJOR. There is a 1:M relationship between DEGREE and STUDENT_DEG_MAJOR. The cardinality for DEGREE to STUDENT_DEG_MAJOR is (1,N). The cardinality for STUDENT_DEG_MAJOR and DEGREE is (1,1).



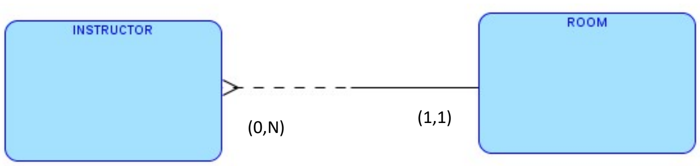
9) *"Each class is held in one room. Each room can hold many classes."* This rule requires two entities: CLASS and ROOM. There is a 1:M relationship between CLASS and ROOM. The cardinality for CLASS to ROOM is (1,1). The cardinality for ROOM to CLASS is (0,N).



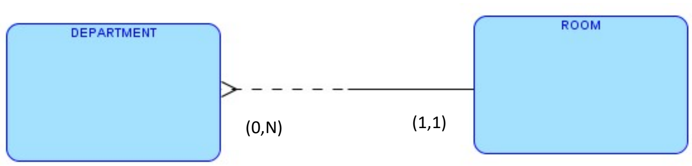
10) *"Each room belongs to one building."* This rule requires two entities: ROOM and BUILDING. There is a 1:M relationship between BUILDING and ROOM. The cardinality of BUILDING to ROOM is (1,N) – each building has at least one room. The cardinality of ROOM to BUILDING is (1,1).



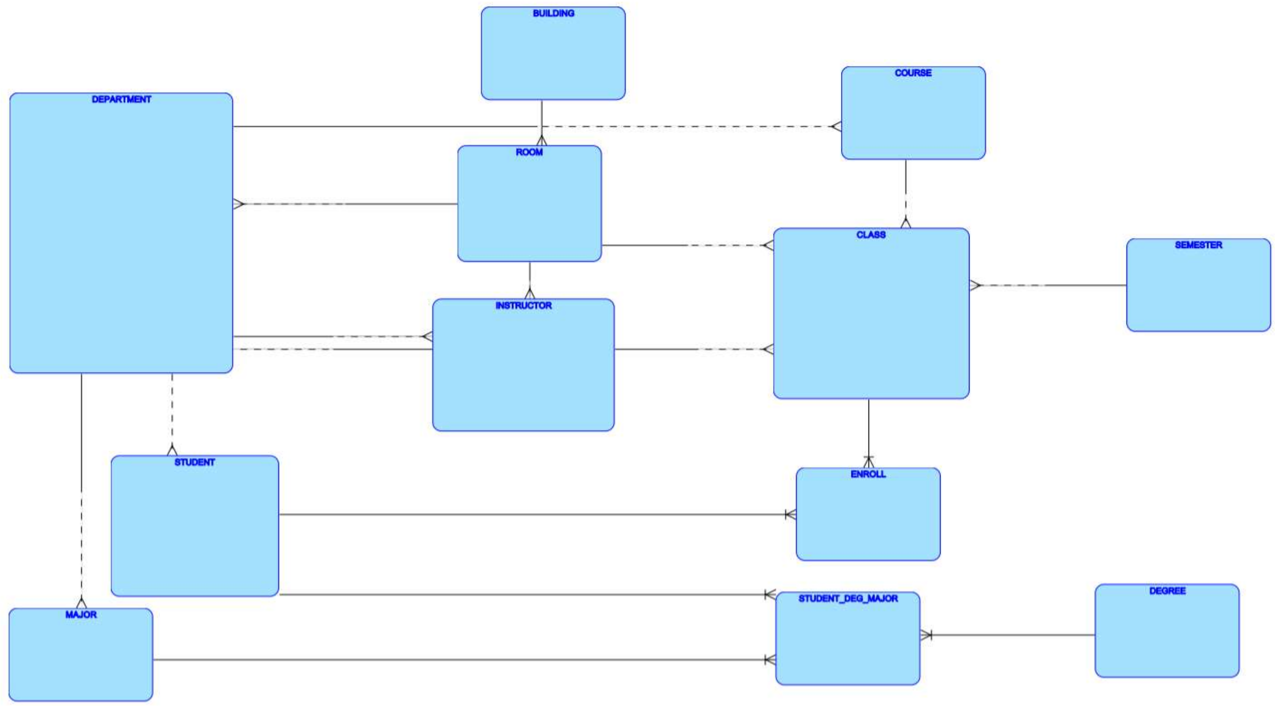
11) *"An instructor has their office in one room."* This rule requires two entities: INSTRUCTOR and ROOM. There is a 1:M relationship between ROOM and INSTRUCTOR. The cardinality of INSTRUCTOR to ROOM is (1,1). The cardinality of ROOM to INSTRUCTOR is (0,N) – each room can have many instructors sharing an office or no instructors.



12) *"A department is in one room."* This rule requires two entities: DEPARTMENT and ROOM. There is a 1:M relationship between ROOM and DEPARTMENT. The cardinality for ROOM to DEPARTMENT is (0,N). The cardinality for DEPARTMENT to ROOM is (1,1).



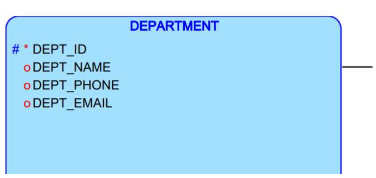
By putting all these entities and relationships together, we get the conceptual model!



Construction of the Logical Model:

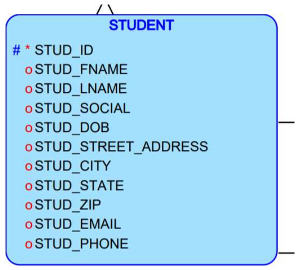
At this stage of the modeling design process, we wish to use the conceptual model and extend it so that it includes the relevant attributes for each entity. Each attribute that is included in an entity must describe the entity.

1) DEPARTMENT: To uniquely identify a department, we will use a department ID called DEPT_ID. It is important for a department to have a name, phone number, and email address. These attributes are called DEPT_NAME, DEPT_PHONE, DEPT_EMAIL.

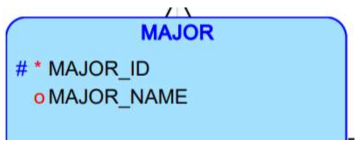


2) STUDENT: To uniquely identify a student, we will use a student ID called STUD_ID. It is important to save a student's personal information such as first name, last name, social security number, date of birth, address, email, and phone number. These attributes are called STUD_FNAME,

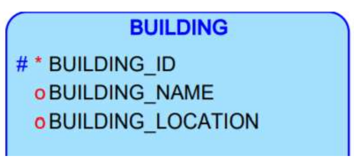
STUD_LNAME, STUD_SOCIAL, STUD_DOB, STUD_STREET_ADDRESS, STUD_CITY, STUD_STATE, STUD_ZIP, STUD_EMAIL , STUD_PHONE.



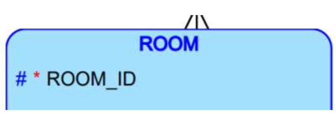
3) MAJOR: To uniquely identify a major, we will use a major ID called MAJOR_ID. Another way to describe a major is by its name – MAJOR_NAME.



4) BUILDING: To uniquely identify a building, we will use a building ID called BUILDING_ID. A building can be described by its name and where it is located – BUILDING_NAME, BUILDING_LOCATION.

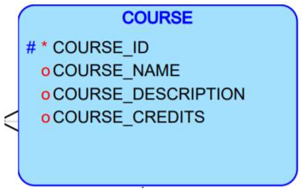


5) ROOM: To uniquely identify a room, we will use a room ID called ROOM_ID.

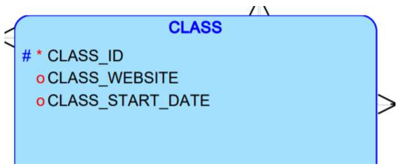


6) INSTRUCTOR: To uniquely identify an instructor, we will use an instructor ID called INST_ID. It is important to save an instructor’s personal information such as first name, last name, email, social security number, date of birth, address, salary, and phone number. These attributes are called INST_FNAME, INST_LNAME, INST_EMAIL, INST_SOCIAL, INST_PHONE, INST_DOB, INST_SALARY.

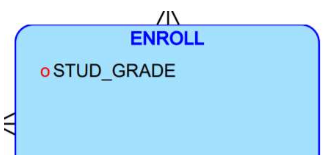
7) COURSE: To uniquely identify a course, we will use a course ID called COURSE_ID. The course should have a name, credit amount, and description. These attributes are called COURSE_NAME, COURSE_CREDITS, COURSE_DESCRIPTION.



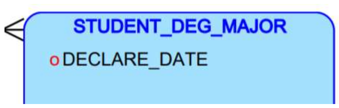
8) CLASS: To uniquely identify a class, we will use a class ID called CLASS_ID. The class should have a class website and a class start date. These attributes are called CLASS_WEBSITE, CLASS_START_DATE.



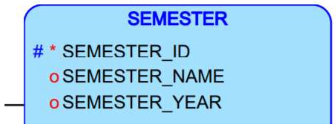
9) ENROLL: The enroll entity should have the student's grade – STUD_GRADE. This entity will inherit the unique identifiers from CLASS and STUDENT.



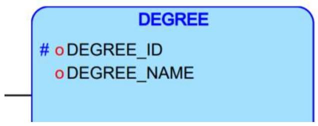
10) STUDENT_DEG_MAJOR: This entity can have an attribute that notes when a student declared their major/degree combo – DECLARE_DATE. This entity will inherit the unique identifiers from STUDENT, MAJOR, and DEGREE.



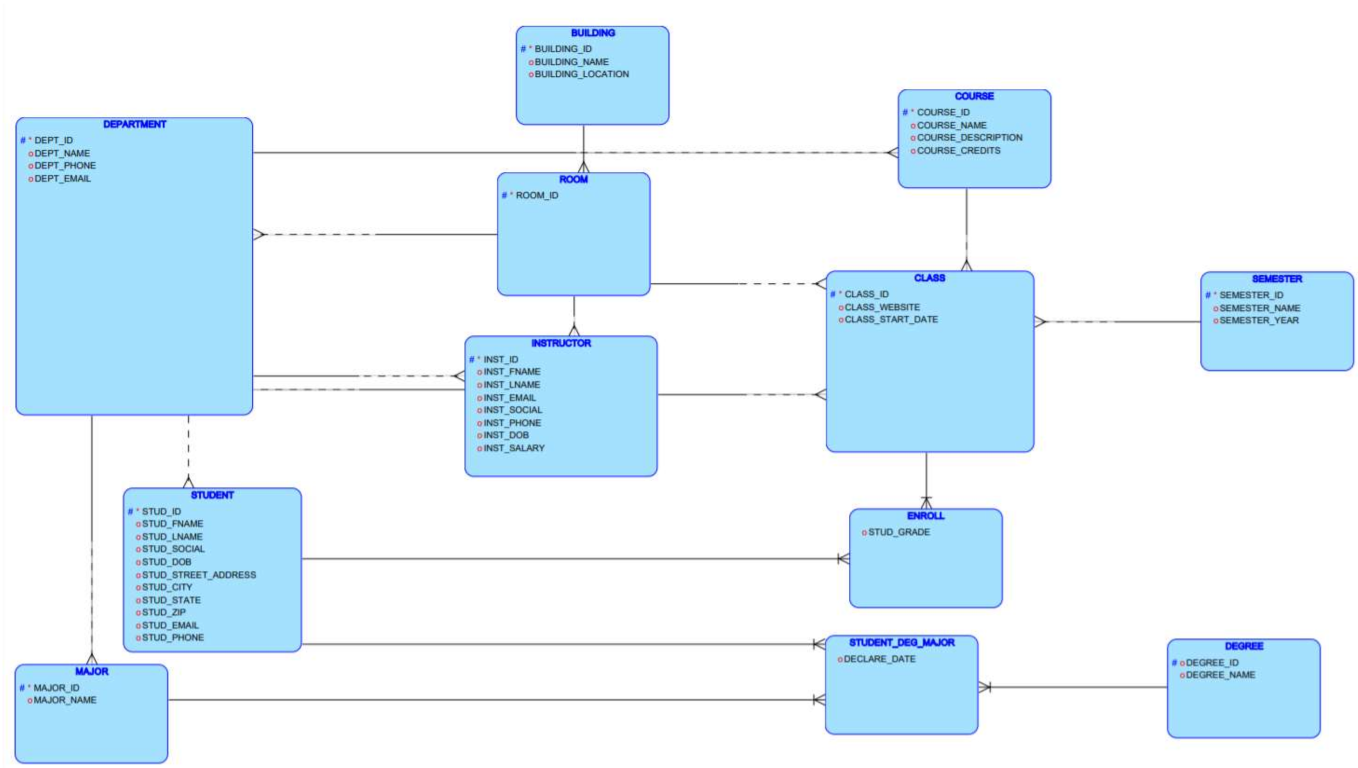
11) SEMESTER: To uniquely identify a semester, we will use a semester ID called SEMESTER_ID. A semester also has a name and the year that it is taking place in – SEMESTER_NAME, SEMESTER_YEAR.



12) DEGREE: To uniquely identify a degree, we will use a degree ID called DEGREE_ID. A degree also has a name – DEGREE_NAME.



Putting all these attributes together, we create our logical model!



Construction of the Physical Model:

The next step of the data model design process is to describe the data types of the attributes and identify and primary/foreign keys for each entity.

1) DEPARTMENT: Since DEPT_ID uniquely identifies the department, then DEPT_ID is the primary key. The DEPT_ID will be a 5 character ID number. Each department has a head, who is also an

instructor. Therefore, department contains HEAD_INST_ID as a foreign key. This attribute consists of 6 characters. The final foreign key for department is ROOM_ID, since each department is contained in one room. This attribute consists of 8 characters. The data types of the remaining attributes are summarized in the table below:

		DEPARTMENT	
P	*	DEPT_ID	VARCHAR2 (5 CHAR)
F		HEAD_INST_ID	VARCHAR2 (6 CHAR)
F		ROOM_ID	VARCHAR2 (8 CHAR)
		DEPT_NAME	VARCHAR2 (20 CHAR)
		DEPT_PHONE	VARCHAR2 (12 CHAR)
		DEPT_EMAIL	VARCHAR2 (20 CHAR)

2) STUDENT: Since STUD_ID uniquely identifies the student, then STUD_ID is the primary key. The STUD_ID will be a 7 character ID number. Each student is part of one department, so DEPT_ID is a foreign key. This attribute consists of 5 characters. The data types of the remaining attributes are summarized in the table below:

		STUDENT	
P	*	STUD_ID	VARCHAR2 (7 CHAR)
F		DEPT_ID	VARCHAR2 (5 CHAR)
		STUD_FNAME	VARCHAR2 (10 CHAR)
		STUD_LNAME	VARCHAR2 (20 CHAR)
		STUD_SOCIAL	VARCHAR2 (11 CHAR)
		STUD_DOB	DATE
		STUD_STREET_ADDRESS	VARCHAR2 (20 CHAR)
		STUD_CITY	VARCHAR2 (20 CHAR)
		STUD_STATE	VARCHAR2 (2 CHAR)
		STUD_ZIP	VARCHAR2 (10 CHAR)
		STUD_EMAIL	VARCHAR2 (20 CHAR)
		STUD_PHONE	VARCHAR2 (12 CHAR)

3) MAJOR: Since MAJOR_ID uniquely identifies the major, then MAJOR_ID is the primary key. The MAJOR_ID will be an integer. Each major is part of one department, so MAJOR_ID is a foreign key. This attribute consists of 5 characters. The data types of the remaining attributes are summarized in the table below:

		MAJOR	
P	*	MAJOR_ID	INTEGER
F		DEPT_ID	VARCHAR2 (5 CHAR)
		MAJOR_NAME	VARCHAR2 (20 CHAR)

4) BUILDING: Since BUILDING_ID uniquely identifies the building, then BUILDING_ID is the primary key. The BUILDING_ID will consist of 6 characters. This entity contains no foreign keys. The data types of the remaining attributes are summarized in the table below:

		BUILDING	
P	*	BUILDING_ID	VARCHAR2 (6 CHAR)
		BUILDING_NAME	VARCHAR2 (20 CHAR)
		BUILDING_LOCATION	VARCHAR2 (4000)

5) ROOM: Since ROOM_ID uniquely identifies the room, then ROOM_ID is the primary key. The ROOM_ID will consist of 8 characters. Each room is part of one building, so BUILDING_ID is a foreign key. This attribute consists of 6 characters. The data types of the remaining attributes are summarized in the table below:

		ROOM	
P	*	ROOM_ID	VARCHAR2 (8 CHAR)
F	*	BUILDING_ID	VARCHAR2 (6 CHAR)

6) INSTRUCTOR: Since INST_ID uniquely identifies the instructor, then INST_ID is the primary key. The INST_ID will consist of 6 characters. Each instructor is part of one department, so DEPT_ID is a foreign key. This attribute consists of 5 characters. Each instructor has an office in one room, so ROOM_ID is a foreign key. This attribute consists of 8 characters. The data types of the remaining attributes are summarized in the table below:

		INSTRUCTOR	
P	*	INST_ID	VARCHAR2 (6 CHAR)
F		DEPT_ID	VARCHAR2 (5 CHAR)
F		ROOM_ID	VARCHAR2 (8 CHAR)
		INST_FNAME	VARCHAR2 (10 CHAR)
		INST_LNAME	VARCHAR2 (15)
		INST_EMAIL	VARCHAR2 (20)
		INST_SOCIAL	VARCHAR2 (11)
		INST_PHONE	VARCHAR2 (12 CHAR)
		INST_DOB	DATE
		INST_SALARY	NUMBER (9,2)

7) COURSE: Since COURSE_ID uniquely identifies the course, then COURSE_ID is the primary key. The COURSE_ID will consist of 7 characters. Each course is offered by one department, so DEPT_ID is a foreign key. This attribute consists of 5 characters. The data types of the remaining attributes are summarized in the table below:

COURSE		
P *	COURSE_ID	VARCHAR2 (7 CHAR)
F	DEPT_ID	VARCHAR2 (5 CHAR)
	COURSE_NAME	VARCHAR2 (20 CHAR)
	COURSE_DESCRIPTION	VARCHAR2 (4000)
	COURSE_CREDITS	NUMBER (2,1)

8) SEMESTER: Since SEMESTER_ID uniquely identifies the semester, then SEMESTER_ID is the primary key. The SEMESTER_ID will consist of 8 characters. The data types of the remaining attributes are summarized in the table below:

SEMESTER		
P *	SEMESTER_ID	VARCHAR2 (8 CHAR)
	SEMESTER_NAME	VARCHAR2 (7 CHAR)
	SEMESTER_YEAR	INTEGER

9) DEGREE: Since DEGREE_ID uniquely identifies the degree, then DEGREE_ID is the primary key. The DEGREE_ID will be an integer. The data types of the remaining attributes are summarized in the table below:

DEGREE		
P *	DEGREE_ID	INTEGER
	DEGREE_NAME	VARCHAR2 (20 CHAR)

10) CLASS: Since CLASS_ID uniquely identifies the class, then CLASS_ID is the primary key. The CLASS_ID will consist of 8 characters. Each class is generated from a course, so COURSE_ID is a foreign key. COURSE_ID consists of 7 characters. Each class is taught by one instructor, so INST_ID is a foreign key. INST_ID consists of 6 characters. Each class is taken in a room, so ROOM_ID, is a foreign key. ROOM_ID consists of 8 characters. Each class is offered in a semester, so SEMESTER_ID is a foreign key. SEMESTER_ID consists of 8 characters. The data types of the remaining attributes are summarized in the table below:

CLASS		
P *	CLASS_ID	VARCHAR2 (8 CHAR)
F	COURSE_ID	VARCHAR2 (7 CHAR)
F	INST_ID	VARCHAR2 (6 CHAR)
F	ROOM_ID	VARCHAR2 (8 CHAR)
F	SEMESTER_ID	VARCHAR2 (8 CHAR)
	CLASS_WEBSITE	VARCHAR2 (4000)
	CLASS_START_DATE	DATE

11) ENROLL: The ENROLL entity is a composite entity. Therefore, the primary and foreign keys are dependent on the entities that are related to ENROLL. The primary/foreign keys are STUD_ID and CLASS_ID. The data types of the remaining attributes are summarized in the table below:

ENROLL		
PF*	STUD_ID	VARCHAR2 (7 CHAR)
PF*	CLASS_ID	VARCHAR2 (8 CHAR)
	STUD_GRADE	VARCHAR2 (1 CHAR)

12) STUDENT_DEG_MAJOR: The STUDENT_DEG_MAJOR entity is a composite entity. Therefore, the primary and foreign keys are dependent on the entities that are related to STUDENT_DEGREE_MAJOR. The primary/foreign keys are STUD_ID , DEGREE_ID, and MAJOR_ID. The data types of the remaining attributes are summarized in the table below:

STUDENT_DEG_MAJOR		
PF*	STUD_ID	VARCHAR2 (7 CHAR)
PF*	DEGREE_ID	INTEGER
PF*	MAJOR_ID	INTEGER
	DECLARE_DATE	DATE

Putting all these together results in our physical model!