Philosophy

[Name of the Writer]

[Name of the Institution]

Part 1:

Conceptual Design:

A **conceptual model** is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the **model** represents.

* Entities:  
  Admin  
   Professor  
   Staff  
   Department  
   Faculty  
   Course  
   Student  
   Room  
   Building
* Supertype:  
   Employee  
  Subtype:  
   Staff  
   Admin  
   Professor

# Primary key:

Employees:  
 id

Student  
 student

Course  
 course\_id

Faculty  
 faculty

Department  
 Deptt\_id

# Candidate Key:

Primary keys are also used as a candidate key. There can be more than one candidate key. Candidate key is used to identify uniquely data from database

# Surrogate key:

Room:

Room\_id

# Relationships and Cardinalities:

One to One relationship between Student and Department because one student has only one department.

One to many relationships between Student and Course because one student has enrolled in many courses.

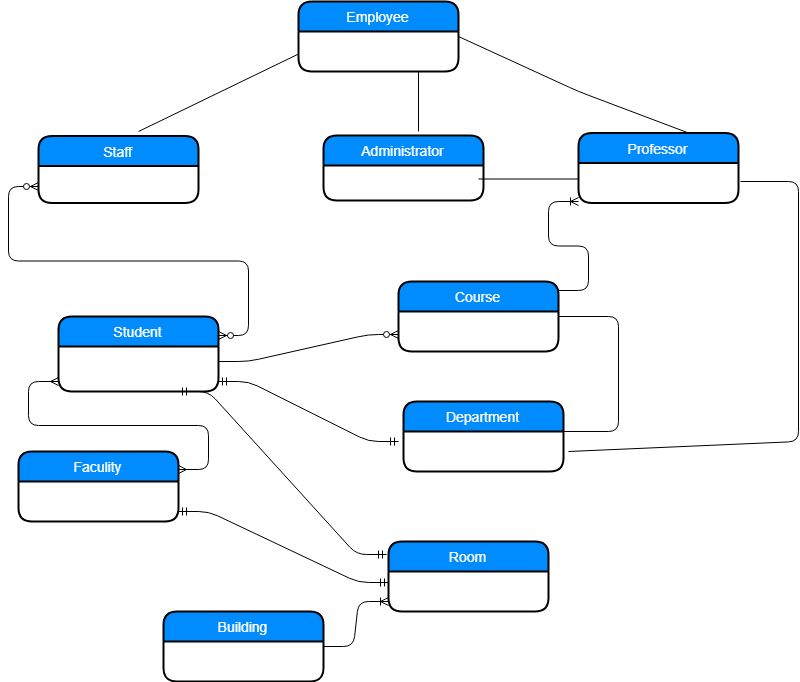
One to one relationship between Building and Room because each room is associated with each building.

One to one relationship between faculty and Department because every faculty member has one department.

Many to Many relationships between Course and professor because one professor has taught many courses at a time.

One to many relationships between faculty and Student because one faculty member has advised many students at a time.

# Conceptual ER Diagram:



# Part 2:

# **Logical Design:**

# A **logical** data **model** describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a **logical** data **model** include: Includes all entities and relationships among them. All attributes for each entity are specified.

# Describe Data Attributes and their types for each entity

* Employee:

+Id: int (PK)  
+First\_Name: Varchar  
+Last\_Name: Varchar  
+Hire\_date: Date

* Admin

+Admin\_position

* Professor

+Rank: int  
 +Specialty: Varchar

* Staff:

+level: int

* Student

+Stud\_id: Varchar (PK)  
 +First\_Name: Varchar  
 +Last\_Name: Varchar  
 +Email: Varchar

* Department

+Deptt\_id: int  
 +Deptt\_Name: Varchar

* Faculty

+Faculty\_id: int (PK)  
+Faculty\_Name: Varchar

* Room:

+Room\_id: int  
+type: char

# Identify Foreign keys for each identity:

In student class stud\_id is the foreign key for courses class and the course\_id in Courses class is the foreign key for student class.

In Professor class Emp\_id is the foreign key for Faculty class and the Faculty\_id in faculty class is the foreign key for professor class.

In Department class Deptt\_id is the foreign key for Course class and the Course\_id in Course class is the foreign key for Department class.

# If there are any one to one relationship then describe:

One to One relationship between Student and Department because one student has only one department.

One to one relationship between Building and Room because each room is associated with each building.

One to one relationship between faculty and Department because every faculty member has one department.

# Logical Model Diagram:

