



INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

CS315

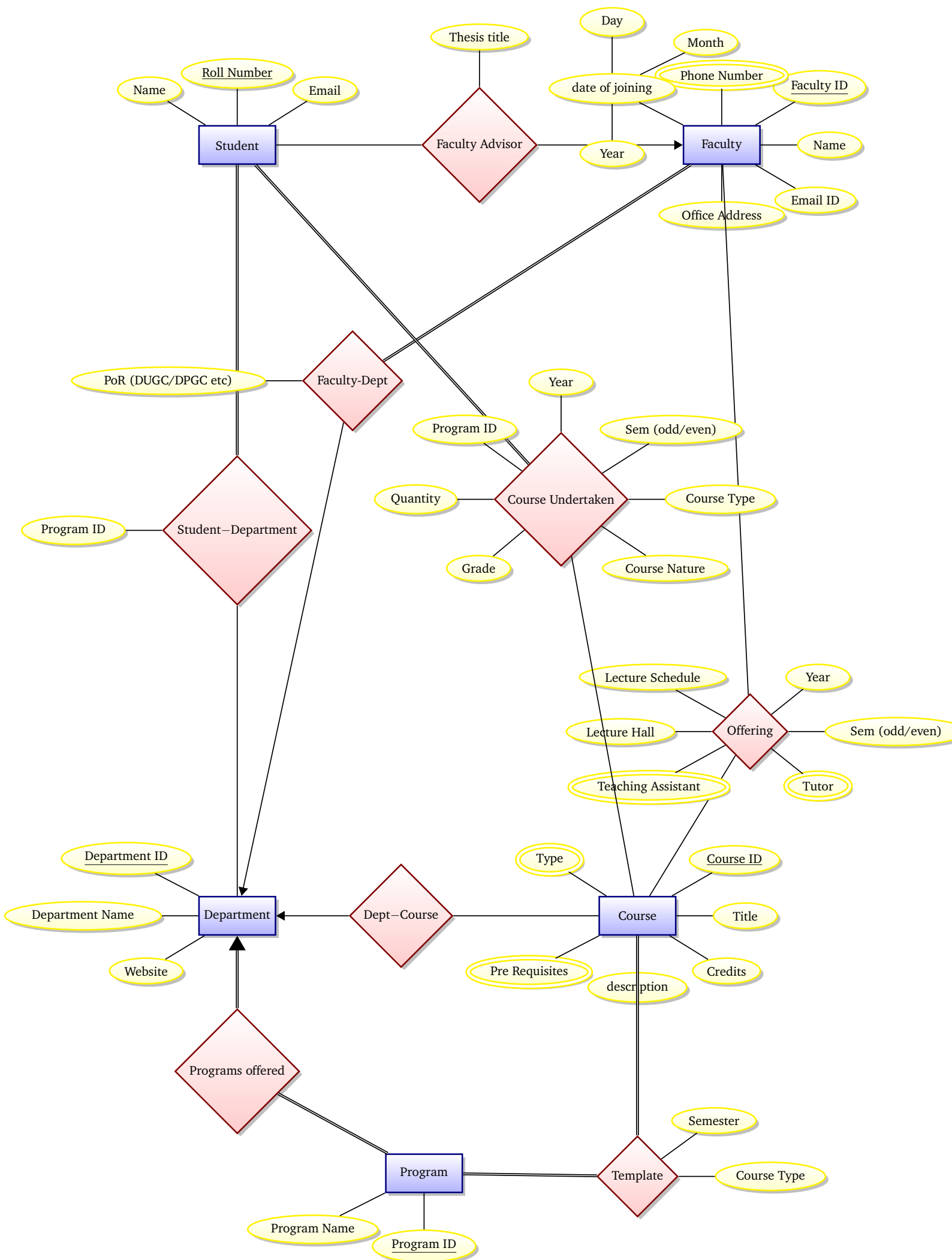
Principles Of Database Systems

Submitted To:

Arnab Bhattacharya
Professor
Computer Science
Department

Submitted By :

Raghukul Raman
160538
Computer Science
Department



Entity Relation Model for Academic system of IITK

Above ER Model describes the academic system of IIT Kanpur, let's discuss non-trivial part/assumptions for the attributes and relationships in more detail.

Entities

- **Student:** The student entity has a unique Roll Number, with other attributes being name, emailID. We can derive other things like the batch of student, year of joining from the roll number.
- **Faculty:** Faculty has a unique code (faculty ID) which acts as the primary key. Faculty entity also has office address, phone number/s etc.
- **Department:** Department entity has a unique department ID, website and the name. We can store other things like department address etc, but that won't be relevant for academic prespective.
- **Course:** This entity represent the courses available in IITK, with couse ID being the primary key. Courses have a mutli valued attribute type, which represent the different categories in which this course can be taken, for eg DE, OE, HSS etc. Note that pre requisites is also a multi valued attribute, since there can be more than one pre-reqs.
- **Program:** This entity represent the different programs offered by a particular department. This can be BTech, MTech, PhD or minor in Algorithms etc. This entity is uniquely identified by the program ID.

Relationships

- **Faculty Advisor:** This relationship is useful for MTech/PhD students, it represents the faculty advisor for that student. Note that we can also store the UGP supervisors for the UG students here. Assuming that a student has only one main faculty advisor, this is a many(student) to one(student) relationship. Participation on either side is not total since a student might not have a faculty advisor, similarly a faculty might not have any student.
- **Student-Department:** This relationship define how a student is related to a particular department. It has attribute program ID to represent this relation. For example student A is doing minors in Algorithms, then he is related to CSE with program ID = program_ID(algorithms). Since a student might be involved in more that one department (major in one, minor in other), this is a many to many relationship, but note that a student is related to atleast one department, so it has a total participation from the student side. I am assuming that a department can exist without any student. Hence not a total participation from dept side.
- **Faculty-Dept:** It represents the department which a faculty belongs to. It has an attribute PoR to represent if he is holding some position in the academic committee of department, eg Prof. Anil Seth is DUGC of CSE, so this attribute will be DUGC for him. Since a faculty is related to only one department, this is a many(department) to one(faculty) relationship. Note that each faculty member is involved with one department, so this is total participation from the faculty side. I am assuming that a department might exist without any faculty, so the participation from department side is not total.
- **Course Undertaken:** This relationship is to define the courses a student has done in past or doing currently. It has an attribute grade, which can be used in SPI/CPI calculation. Course Nature is fresh/repeat etc. Course type is used to represent the category in which the student is doing the course, as student A might do CS315 as DE, while other might do it as an OE. Program ID is maintained to know that this student is doing this course in order to complete which program. For example student A might be doing ESO207 in order to complete his BTech in CSE, while B might be doing ESO207 to complete a minor in algorithms. Quantity attribute is used to encorporate thesis course. For normal courses this value is 1. But for MTech/PhD thesis course this can be > 1. Let's say that student A is taking 4 thesis credits, this would mean that he is taking course CS699 (or other equivalant) 4 times, for which he might get different grades.

Also note that grade should have been multi valued, but since there are more UG, we keep it comma separated to save memory.

Assuming that a student has to take at least one course, this relation is total from the student side. But note that course might be offered but no one took it, so course is not in total participation.

- **Offering:** This relationship between course and faculty represents the offering of this course by this particular faculty. This offering has year, semester as attribute since an instructor might offer same course multiple times. This relationship also contains some other relevant information like lecture hall, tutors, TA, and course timings etc.
- **Dept-Course:** This relationship is necessary since a course is offered by faculty members of a particular department only. For example MTH101, would only be offered by faculty members of mathematics department.
It's a many (course) to one(dept) relationship, since a course can only belong to one department.
- **Programs offered** This relationship incorporates which all programs are offered by a particular department. For example CSE offers minor in 5-6 fields, major in CS, BTech in CS, Dual etc.
Many(programs) to one(department), since a program can only belong to only one department. Here a program is like BTech in Economics and not just BTech. There is total participation from both sides since, none can exist without the other.
- **Template** Relationship between program and course is needed to store which all courses one needs to complete in order to get this program's recognition. This relationship has an attribute semester, which would signify in which semester one should do this course. For example, CSE BTechs need to do LIF101 in 2nd semester. Course type is similar to what we discussed in course entity. ie how one should take this course in order to complete the template. For eg LIF101 should be done as IC by UnderGrads.
Total participation from both sides is needed, since none can be without other.

Note: Grade is a single valued attribute, but to incorporate for multiple thesis credit, we can keep the grade comma separated. Note that this would work fine since the grades are only S/X type in these courses, hence not required in CPI/SPI calculation.

Problem 2: Conversion to Relational Model

Using the scheme discussed in lecture, converted ER model is given by (dashed underline: Foreign key, solid underline: Primary key):

- **Student** (Roll Number, Name, Email ID)
- **Faculty** (Faculty ID, Name, Email ID, Office Address, Year, Month, Day)
- **Faculty_Contact** (Faculty ID, Phone Number) // *Faculty ID is a foreign key in Faculty*
- **Department** (Department ID, Department Name, Website)
- **Course** (Course ID, Title, Credits, Description)
- **Pre_Requisites** (Course ID, Pre Req Course ID) // *Second course ID represent ID for pre req.*
- **Course_Type** (Course ID, type) // *Type can be DC, IC, OE etc.*
- **Program** (Program ID, Program Name) // *can be BTech in CS, MTech in CS, minor in algo. etc.*
- **Template** (Program ID, Course ID, Semester, Course Type)
- **Programs_Offered** (Department ID, Program ID)
- **Dept_Course** (Course ID, Department ID)
- **Offering** (Course ID, Faculty ID, Year, Sem, Lecture Hall, Lecture Schedule)
- **Tutors** (Course ID, Faculty ID, Year, Sem, Tutor ID) // *Since a tutor can be either a student or a faculty, this tutor ID is either roll number or faculty ID*
- **Teaching_Assistants** (Course ID, Faculty ID, Year, Sem, Roll Number) // *Since TA are always student*
- **Course_Undertaken** (Roll Number, Course ID, Year, Sem, Course Type, Course Nature, Grade, Quantity, Program ID)
- **Student_Department** (Roll Number, Department ID, Program ID)
- **Faculty Advisor** (Roll Number, Faculty ID, Thesis title)
- **Faculty_Department** (Faculty ID, Department ID, PoR)