Database Design CS 6360

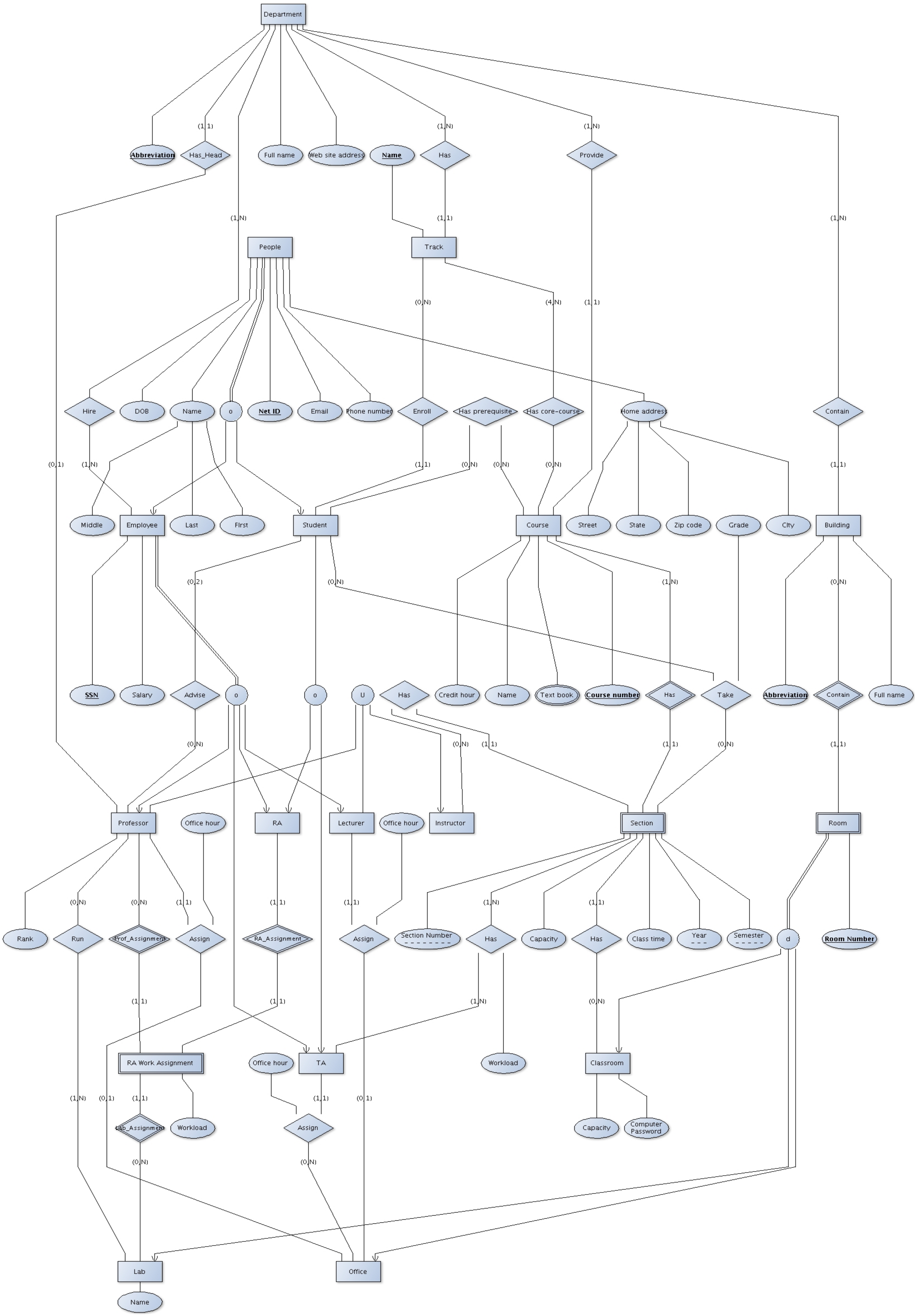
Project Phase II

Team Member

Yuchuan Liu	2021184144
Xian Shi	2021187621
Jiaming Fan	2021225346

Table of Contents

I. Updated EER Diagram



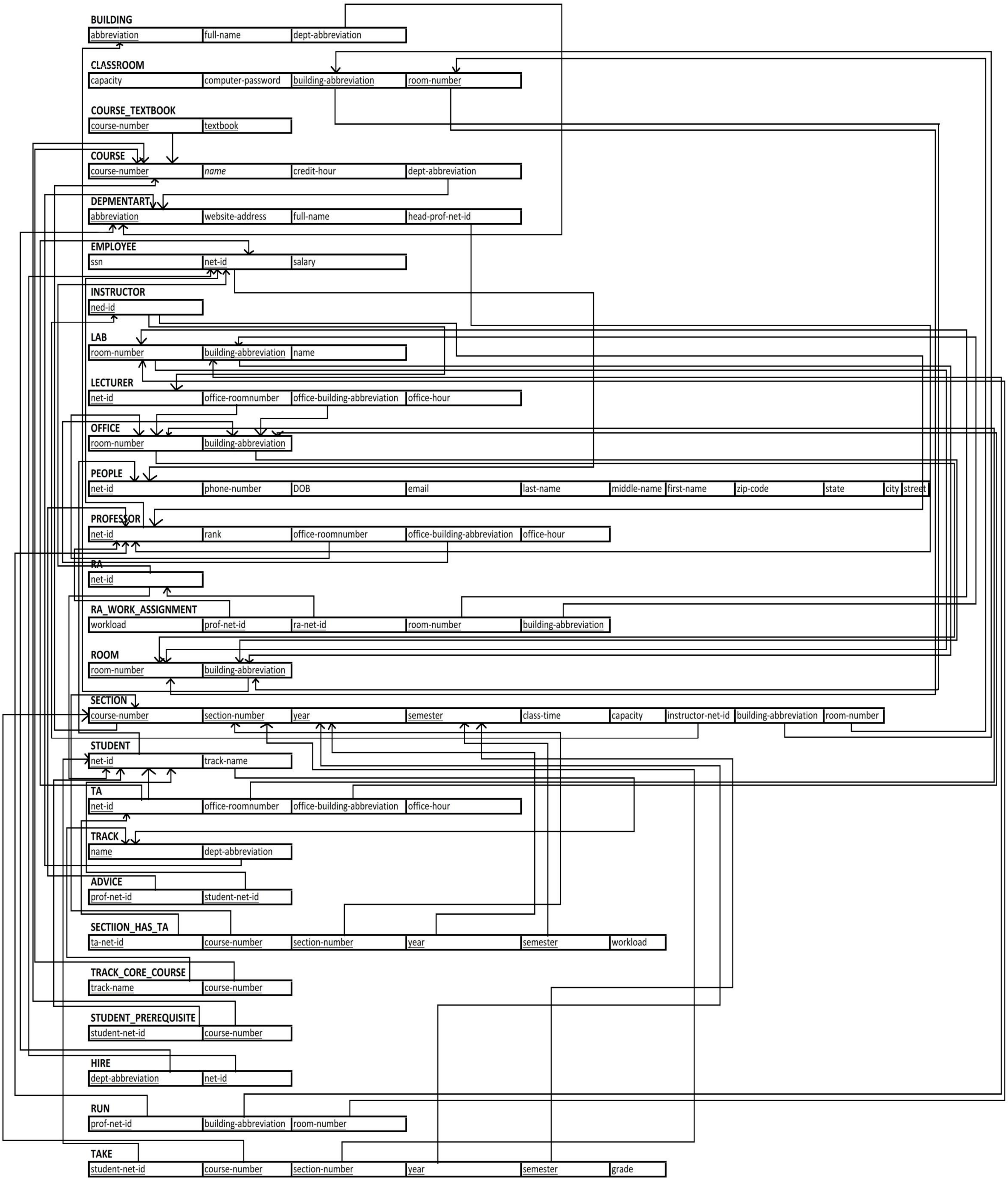
II. EER Diagram Update Log

1. Remove entity type “Graduate School” and related relations. (The removal of entity type below will also remove related relation)
2. Change entity type “Room” to weak entity type and change relation type “Contain” among “Room” and “Building” to Identifying relationship type.
3. Remove entity type “Associate Prof.”, “Assistant Prof.”, and “Full Prof”, and add attribute “Rank” to entity type “Professor”.
4. Change the Specialization of entity type “Employee” to overlapping.
5. Change the partial key of weak entity type “Section” to combination of number, semester, and year. (Since the tool we choose doesn’t support partial key notation, we add “- - - - -“ beneath the attribute name)
6. Change the two minimal of cardinality constrains of relationship between entity types “Professor” and “Student” to 0.
7. Change cardinality constrains among entity type “RA”, “Professor”, “Lab”, and “RA Work Assignment”.
8. Change cardinality constrain for “Has Prerequisite” relation between entity type “Course” and “Student”. They are now partial participate.
9. Entity type “Instructor” becomes a union of “Lecturer” and “Professor”.
10. Entity type “Office” no longer has total participation in relationships “Assign” between “Lecturer”/”TA”/”Professor” and “Office”.
11. We change layout of EER model. Since previous version may cause difficulties in reading min-max constraints on edges.

III. Relation Schema

INSTRUCTOR is a union of LECTURER and PROFESSOR. The primary key “net id” of INSTRUCTOR is foreign key refer to “net id” in thoes two relations. (We have two lines in diagram.)

RA/TA is both STUDENT and EMPLOYEE, The foreign key “net id” of TA/RA refer to STUDENT and EMPLOYEE at same time. (Due to complexity representing it in diagram, we just use two lines.)



BUILDING

abbreviation	full-name	dept-abbreviation

CLASSROOM

capacity	computer-password	building-abbreviation	room-number

COURSE_TEXTBOOK

course-number	textbook

COURSE

course-number	*name*	credit-hour	dept-abbreviation

DEPMENTART

abbreviation	website-address	full-name	head-prof-net-id

EMPLOYEE

ssn	net-id	salary

INSTRUCTOR

ned-id

LAB

room-number	building-abbreviation	name

LECTURER

net-id	office-roomnumber	office-building-abbreviation	office-hour

OFFICE

room-number	building-abbreviation

PEOPLE

net-id	phone-number	DOB	email	last-name	middle-name	first-name	zip-code	state	city	street

PROFESSOR

net-id	rank	office-roomnumber	office-building-abbreviation	office-hour

RA

net-id

RA_WORK_ASSIGNMENT

workload	prof-net-id	ra-net-id	room-number	building-abbreviation

ROOM

room-number	building-abbreviation

SECTION

course-number	section-number	year	semester	class-time	capacity	instructor-net-id	building-abbreviation	room-number

STUDENT

net-id	track-name

TA

net-id	office-roomnumber	office-building-abbreviation	office-hour

TRACK

name	dept-abbreviation

ADVICE

prof-net-id	student-net-id

SECTIION_HAS_TA

ta-net-id	course-number	section-number	year	semester	workload

TRACK_CORE_COURSE

track-name	course-number

STUDENT_PREREQUISITE

student-net-id	course-number

HIRE

dept-abbreviation	net-id

RUN

prof-net-id	building-abbreviation	room-number

TAKE

student-net-id	course-number	section-number	year	semester	grade

IV. Mapping Steps

We convert EER model to Relational Model based on following Steps:

- Map entity type “Building” to relation “BUILDING”. (Related attributes added as column header. Also apply to following if not specified)
- Map entity type “Classroom: to relation “CLASSROOM”. We use Option 8A, multiple relations with superclass and subclasses. We choice this option for all rest superclass and subclasses, if not specified.
- Map entity type “Course” to relation “COURSE”.
- Map multi-value attribute textbook and attribute course-number of entity type “COURSE” to relation “COURSE_TEXTBOOK”.
- Map entity type “Department” to relation “Department”.
- Map entity type “Employee” to relation “EMPLOYEE”.
- Map entity type “Instructor” to relation “INSTRUCTOR”. It is a Union of Professor and Lecturer. We use their common key “net-id” as INSTRUCTOR's key.
- Map entity type “lab” to relation “LAB”.
- Map entity type “Lecturer” to relation “LECTURER”.
- Map entity type “Office” to relation “OFFICE”.
- Map entity type “People” to relation “PEOPLE”.
- Map entity type “Professor” to relation “PROFESSOR”.
- Map entity type “RA” to relation “RA”.
- Map entity type “RA Work Assignment” to relation “RA_WORK_ASSIGNMENT”.
- Map entity type “Room” to relation “ROOM”.
- Map entity type “Section” to relation “SECTION”.
- Map entity type “Student” to relation “STUDENT”.
- Map entity type “TA” to relation “TA”.
- Map entity type “Track” to relation “TRACK”.
- Map relationship type “Advice” between Student and Professor to relation “ADVICE”.
- Map relationship type “Assign” between Professor and Office as attributes to relation “PROFESSOR”. (We put OFFICE’s Key as attributes to relation PROFESSOR. We will do similar below when we map relationship type as attributes.)
- Map relationship type “Assign” between Lecturer and Office as attributes to relation “LECTURER”.
- Map relationship type “Assign” between TA and Office as attributes to relation “TA”.
- Map relationship type “Contain” between Department and Building as attributes to relation “BUILDING”.
- Map relationship type “Enroll” between Track and Student as attributes to relation “STUDENT”.
- Map relationship type “Has” between Instructor and Section as attributes to relation “SECTION”.
- Map relationship type “Has” between Section and TA to relation “SECTION_HAS_TA”.
- Map relationship type “Has” between Section and Classroom as attributes to relation “SECTION”.
- Map relationship type “Has” between Department and Track as attributes to relation “TRACK”.
- Map relationship type “Has core-course” between Track and Course to relation

"TRACK_CORE_COURSE".

- Map relationship type "Has prerequisite" between Student and Course to relation "STUDENT_PREREQUISITE".
- Map relationship type "Has Head" between Department and Professor as attributes to relation "DEPARTMENT".
- Map relationship type "Hire" between Department and Professor to relation "HIRE".
- Map relationship type "Provide" between Department and Course as attributes to relation "COURSE".
- Map relationship type "Run" between Professor and Lab to relation "RUN".
- Map relationship type "Take" between Student and Section to relation "TAKE".

V. Constrains of Relation Schema

There is only one candidate key for our relation schema. "ssn" could be candidate key for relation "EMPLOYEE". However, we choose "net id" as its primary key since all other people related relations use "net id" as primary key.

We have following diagram for our constraints on relation schema. The constraint for attribute is right beneath.

BUILDING

abbreviation	full-name	dept-abbreviation
non-null;string	non-null;string	non-null;string

CLASSROOM

capacity	computer-password	building-abbreviation	room-number
non-null;integer	string	non-null;string	non-null;integer(1000-9999)

COURSE_TEXTBOOK

course-number	textbook
non-null;integer(1000-9999)	non-null;string

COURSE

course-number	*name*	credit-hour	dept-abbreviation
non-null;integer(1000-9999)	non-null;string	non-null;integer(1-6)	non-null;string

DEPMENTART

abbreviation	website-address	full-name	head-prof-net-id
non-null;string	string	non-null;string	non-null;string

EMPLOYEE

ssn	net-id	salary
non-null;9 digit integer	non-null;string	non-null;decimal

INSTRUCTOR

ned-id
non-null;string

LAB

room-number	building-abbreviation	name
non-null;integer(1000-9999)	non-null;string	non-null;string

LECTURER

net-id	office-roomnumber	office-building-abbreviation	office-hour
non-null;string	non-null;integer(1000-9999)	non-null;string	non-null;decimal

OFFICE

room-number	building-abbreviation
non-null;integer(1000-9999)	non-null;string

PEOPLE

net-id	phone-number	DOB	email	last-name	middle-name	first-name	zip-code	state	city	street
non-null;string	10 digit integer	date	string	non-null;string	sting	non-null;string	5 digit integer	string	string	string

PROFESSOR

net-id	rank	office-roomnumber	office-building-abbreviation	office-hour
non-null;string	1 of (assistant,associate,full)	non-null;integer(1000-9999)	non-null;string	non-null;decimal

RA

net-id
non-null;string

RA_WORK_ASSIGNMENT

workload	prof-net-id	ra-net-id	room-number	building-abbreviation
non-null;decimal	non-null;string	non-null;string	non-null;integer(1000-9999)	non-null;string

ROOM

room-number	building-abbreviation
non-null;integer(1000-9999)	non-null;string

SECTION

course-number	section-number	year	semester	class-time	capacity	instructor-net-id	building-abbreviation	room-number
non-null;integer(1000-9999)	non-null;integer(0-999)	non-null;4 digit integer	non-null;string	non-null;decimal	non-null;integer	non-null;string	non-null;string	non-null;integer(1000-9999)

STUDENT

net-id	track-name
non-null;string	non-null;string

TA

net-id	office-roomnumber	office-building-abbreviation	office-hour
non-null;string	non-null;integer(1000-9999)	non-null;string	non-null;decimal

TRACK

name	dept-abbreviation
non-null;string	non-null;string

ADVICE

prof-net-id	student-net-id
non-null;string	non-null;string

SECTIION_HAS_TA

ta-net-id	course-number	section-number	year	semester	workload
non-null;string	non-null;integer(1000-9999)	non-null;integer(0-999)	non-null;4 digit integer	non-null;string	non-null;decimal

TRACK_CORE_COURSE

track-name	course-number
non-null;string	non-null;integer(1000-9999)

STUDENT_PREREQUISITE

student-net-id	course-number
non-null;string	non-null;integer(1000-9999)

HIRE

dept-abbreviation	net-id
non-null;string	non-null;string

RUN

prof-net-id	building-abbreviation	room-number
non-null;string	non-null;string	non-null;integer(1000-9999)

TAKE

student-net-id	course-number	section-number	year	semester	grade
non-null;string	non-null;integer(1000-9999)	non-null;integer(0-999)	non-null;4 digit integer	non-null;string	decimal(0.00-4.00)