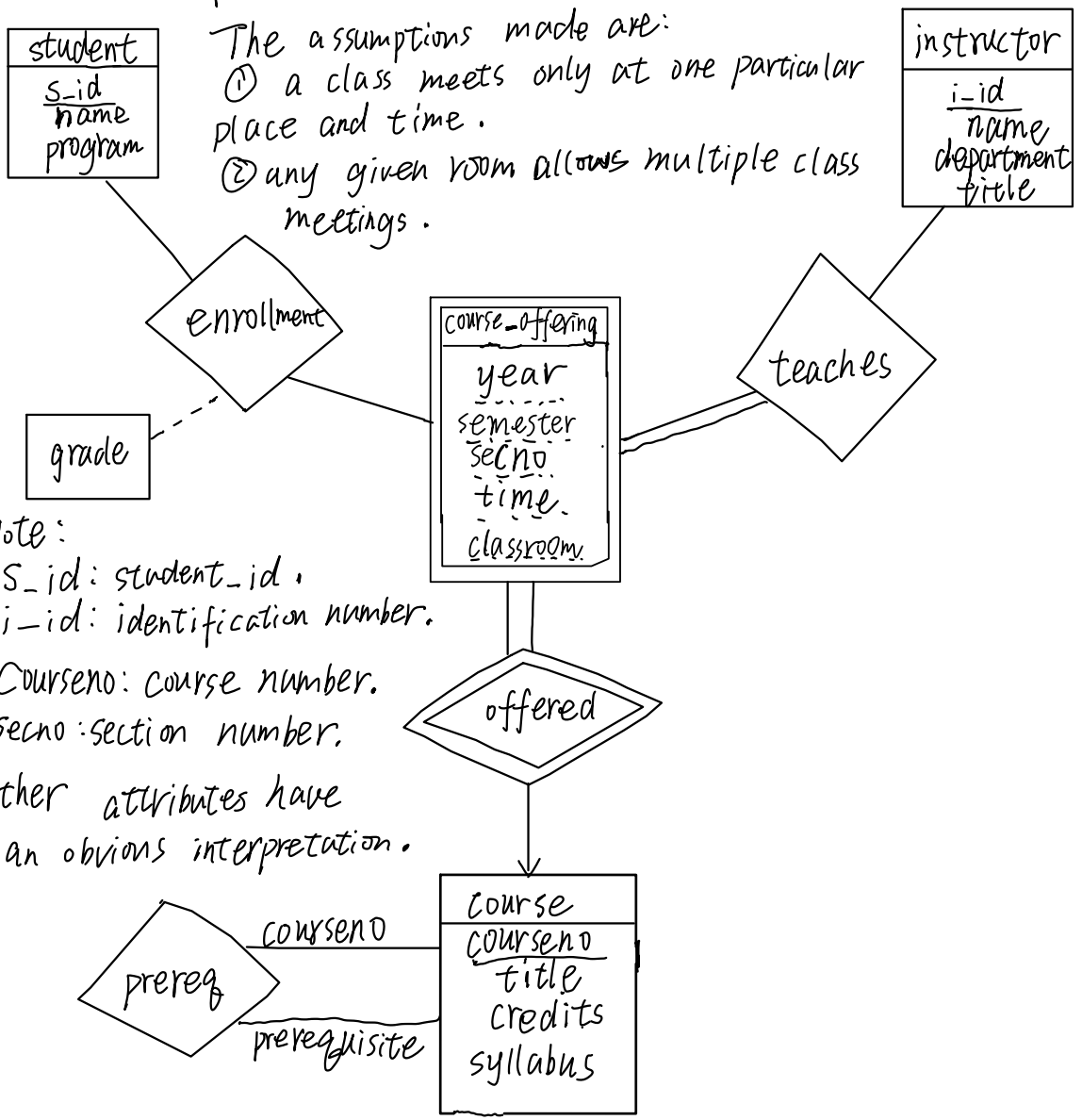
#1. <a> The entity set course_offering is a weak entity set dependent on course.

The assumptions made are:
① a class meets only at one particular place and time.
② any given room allows multiple class meetings.



Note:
S_id: student_id.
i_id: identification number.
Courseno: course number.
secno: section number.
other attributes have an obvious interpretation.

<b> Course_offering (courseno, year, semester, secno, time, classroom)
teaches (courseno, i_id, year, semester, secno, time, classroom)
enrollment (courseno, s_id, year, semester, secno, time, classroom, grade)
prereq (courseno, prerequisite)

student (s_id, name, program)
instructor (i_id, name, department, title)
course (courseno, title, credits, syllabus)

#2. <a> the following columns would be changed.

schema	column(s)
STUDENT	Major
COURSE	Course_number, Department
SECTION	Course_number
PREREQUISITE	Course_number, Prerequisite_number

<b> split the following columns into two columns.

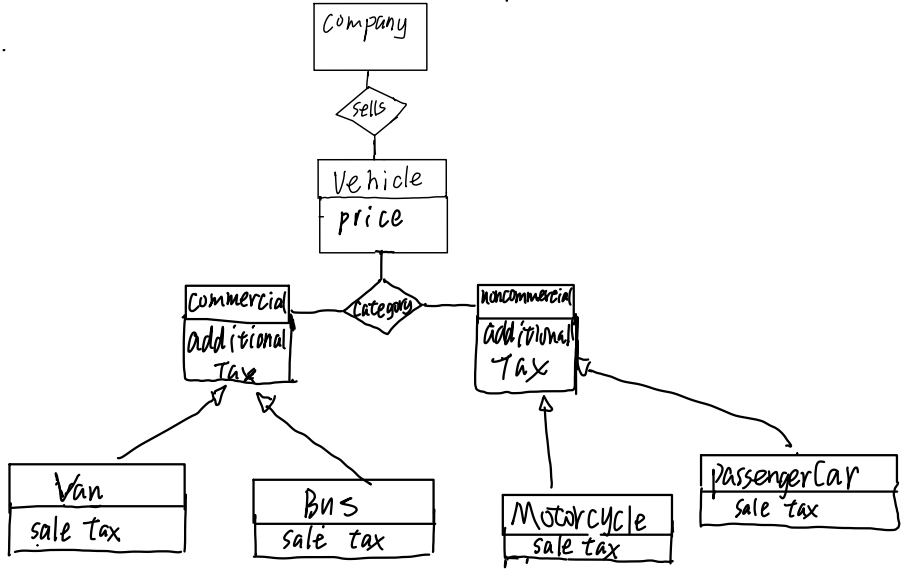
① Course_number → CourseDept and CourseNum

② Prerequisite_number → PrereqDept and PrereqNum

Note: in COURSE schema, the new splited column CourseNum would not be needed since the column Department already exists.

schema	column	split column(s)
COURSE	Course_number	CourseNum
SECTION		CourseDept, CourseNum
PREREQUISITE		CourseDept, CourseNum
PREREQUISITE	Prerequisite_number	PrereqDept, PrereqNum

#3.



#4.

$$\pi_{Fname, Minit, Lname, Address}(\sigma_{Dno = Dnumber \wedge Dname = \text{"Research"}}(EMPLOYEE \times DEPARTMENT))$$

#5.

$$\pi_{Pnumber, Lname, Address, Bdate}($$
$$EMPLOYEE \bowtie_{Mgr\_ssn = Ssn} ($$
$$\sigma_{Plocation = \text{"Stafford"}}(PROJECT \bowtie_{Dnum = Dnumber} DEPARTMENT)))$$

#6.

$$SMITH\_AS\_WORKER \leftarrow \pi_{Pnumber}(WORKS\_ON \bowtie_{Essn = Ssn} \sigma_{Lname = \text{"Smith"}}(EMPLOYEE))$$
$$MGRS \leftarrow \pi_{Lname, Dnumber}(EMPLOYEE \bowtie_{Ssn = Mgr\_ssn} DEPARTMENT)$$
$$SMITH\_AS\_MGR \leftarrow \pi_{Pnumber}(PROJECT \bowtie_{Dnumber = Dnum} \sigma_{Lname = \text{"Smith"}}(MGRS))$$
$$RESULT \leftarrow SMITH\_AS\_WORKER \cup SMITH\_AS\_MGR$$

As a single in-line expression, the query becomes:

$$\pi_{Pnumber}(WORKS\_ON \bowtie_{Essn = Ssn} \sigma_{Lname = \text{"Smith"}}(EMPLOYEE))$$
$$\cup$$
$$\pi_{Pnumber}(PROJECT \bowtie_{Dnumber = Dnum} \sigma_{Lname = \text{"Smith"}}(EMPLOYEE \bowtie_{Ssn = Mgr\_ssn} DEPARTMENT))$$

#7.

$$\pi_{Fname, Minit, Lname}(EMPLOYEE \bowtie (\pi_{Ssn}(EMPLOYEE) - \rho_{Ssn}(\pi_{Essn}(DEPENDENT))))$$

#8.

$$\pi_{Fname, Minit, Lname}(EMPLOYEE \bowtie (\rho_{Ssn}(\pi_{Essn}(DEPENDENT)) \cap \rho_{Ssn}(\pi_{Mgr\_ssn}(DEPARTMENT))))$$

#9. $\sigma(EMPLOYEE \bowtie_{Super\_ssn = Ssn} \sigma_{Fname \neq \text{"James"} \wedge Lname = \text{"Borg"}}(EMPLOYEE))$

#10 <1>

$$\sigma(EMPLOYEE \bowtie_{Super\_ssn = Ssn} (EMPLOYEE \bowtie_{Super\_ssn = Ssn} \sigma_{Fname = \text{"James"} \wedge Lname = \text{"Borg"}}(EMPLOYEE)))$$

(2) Yes, it is possible, but needs to know the maximum number of levels.

By using recursive closure.

like the steps in question 9, suppose the name of the given employee is X

$$LEVEL_1 = \sigma(EMPLOYEE \bowtie_{super\_ssn = ssn} \sigma_{name: X}(EMPLOYEE))$$

$$LEVEL_{(i+1)} = \sigma(EMPLOYEE \bowtie_{super\_ssn = ssn} LEVEL_i) \quad (*)$$

keep doing (*), when reaches to the maximum number of level, then apply the $\cup$ operation as follows:

$$RESULT \leftarrow LEVEL_1 \cup LEVEL_2 \cup LEVEL_3 \cdots \cup LEVEL_N$$

note:

Although it is possible to recover the employees supervised at each level, if not known the maximum number of levels, the query can't be written.